An introduction to fractional calculus

Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

May 2, 2022

The original idea

The concept of differentiation and integration to noninteger order goes as far back as the concept we are used to work with. Leibniz mentions it in a letter to L'Hôspital in 1695:

> *"John Bernoulli seems to have told you of my having mentioned to him a marvelous analogy which makes it possible to say in a way that successive differentials are in geometric progression.* ***One can ask what would be a differential having as its exponent a fraction****. You see that the result can be expressed by an infinite series. Although this seems removed from Geometry, which does not yet know of such fractional exponents,* ***it appears that one day these paradoxes will yield useful consequences, since there is hardly a paradox without utility****. Thoughts that mattered little in themselves may give occasion to more beautiful ones."*

(Leibniz, 1646-1716)

Who cares?

Derivatives of non integer order help

- modeling of *viscoelastic* phenomena, *e.g.*, (Bagley and Torvik 1986; Müller et al. 2011)
- restate fundamental model from physics [gravity (Giusti, Garrappa, and Vachon 2020), Schrödinger (Laskin 2002), waves (Luchko 2013), ...],
- modeling of heterogeneous cardiac tissues (Cusimano et al. 2015),
- describing phenomena with *memory* and *non locality* aspects, *e.g.*, (Benzi et al. 2020; Riascos and Mateos 2014)

⋮

This is a **booming topic**, and many new applications frequently arise.

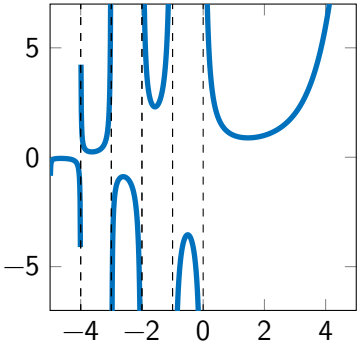
Fractional integrals

Euler Γ-function

The Γ **function** $\Gamma(z)$ is defined for complex numbers with a positive real part via the convergent improper integral:

$$\Gamma(z) = \int_0^{+\infty} x^{z-1} e^{-x}\,\mathrm{d}x, \qquad \Re(z) > 0,$$

and then extended by **analyitic continuation** to a *meromorphic* function that is holomorphic in the whole complex plane except zero and the negative integers, where the function has simple poles.



$\Gamma(z+1) = z\,\Gamma(z)$

Bounded in:

$S = \{z \in \mathbb{C} : \Re z \in [1, 2)\}$

A formula for repeated integration

Swapping Integrals

If $G(x,t)$ is jointly continuous on $[c,b] \times [c,b]$:

$$\int_c^x \mathrm{d}x_1 \int_c^{x_1} G(x_1,x_2)\mathrm{d}x_2 = \int_c^x \mathrm{d}x_2 \int_{x_2}^x G(x_1,x_2)\mathrm{d}x_1.$$

A formula for repeated integration

Swapping Integrals

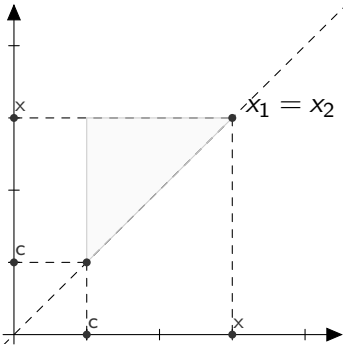
If $G(x,t)$ is jointly continuous on $[c,b]\times[c,b]$:

$$\int_c^x \mathrm{d}x_1 \int_c^{x_1} G(x_1,x_2)\mathrm{d}x_2 = \int_c^x \mathrm{d}x_2 \int_{x_2}^x G(x_1,x_2)\mathrm{d}x_1.$$

Fubini's Theorem

Given $(X,\mathfrak{S}_X,\mu_x)$, $(Y,\mathfrak{S}_Y,\mu_y)$ measure spaces with σ-finite complete measures μ_x, μ_y on the σ-algebras $\mathfrak{S}_X$, and $\mathfrak{S}_Y$. If the function $f(x,y)$ is integrable on the product $X\times Y$ w.r.t. the product measure $\mu=\mu_x\times\mu_y$, then the following equality holds true

$$\int_{X\times Y} f(x,y)\,\mathrm{d}\mu = \int_Y \mathrm{d}\mu_y \int_X f(x,y)\mathrm{d}\mu_x.$$



A formula for repeated integration

Swapping Integrals

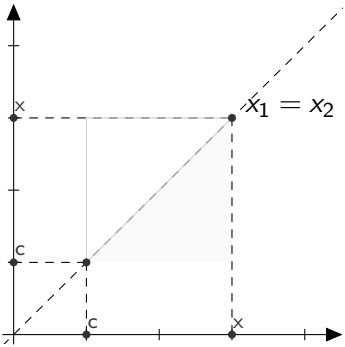
If $G(x,t)$ is jointly continuous on $[c,b]\times[c,b]$:

$$\int_c^x \mathrm{d}x_1 \int_c^{x_1} G(x_1,x_2)\mathrm{d}x_2 = \int_c^x \mathrm{d}x_2 \int_{x_2}^x G(x_1,x_2)\mathrm{d}x_1.$$

Fubini's Theorem

Given $(X,\mathfrak{S}_X,\mu_x)$, $(Y,\mathfrak{S}_Y,\mu_y)$ measure spaces with σ-finite complete measures μ_x, μ_y on the σ-algebras $\mathfrak{S}_X$, and $\mathfrak{S}_Y$. If the function $f(x,y)$ is integrable on the product $X\times Y$ w.r.t. the product measure $\mu=\mu_x\times\mu_y$, then the following equality holds true

$$\int_{X\times Y} f(x,y)\,\mathrm{d}\mu = \int_Y \mathrm{d}\mu_y \int_X f(x,y)\mathrm{d}\mu_x.$$



A formula for repeated integration

Cauchy's formula

The indefinite integral of order $n \in \mathbb{N}$ of function $f(t)$ is given by

$$I^n_{c,t} f(t) = \int_c^t \cdots \int_c^t f(t)\,\mathrm{d}t \cdots \mathrm{d}t = \frac{1}{(n-1)!} \int_c^t (t-\tau)^{n-1} f(\tau)\,\mathrm{d}\tau,$$

$$I^n_{t,c} f(t) = \int_t^c \cdots \int_t^c f(t)\,\mathrm{d}t \cdots \mathrm{d}t = \frac{1}{(n-1)!} \int_c^t (\tau-t)^{n-1} f(\tau)\,\mathrm{d}\tau.$$

- Can be proved **by induction** using Fubini's Theorem/the previous formula,

A formula for repeated integration

Cauchy's formula

The indefinite integral of order $n \in \mathbb{N}$ of function $f(t)$ is given by

$$I^n_{c,t} f(t) = \int_c^t \cdots \int_c^t f(t)\,\mathrm{d}t \cdots \mathrm{d}t = \frac{1}{\Gamma(n)} \int_c^t (t-\tau)^{n-1} f(\tau)\,\mathrm{d}\tau,$$

$$I^n_{t,c} f(t) = \int_t^c \cdots \int_t^c f(t)\,\mathrm{d}t \cdots \mathrm{d}t = \frac{1}{\Gamma(n)} \int_c^t (\tau-t)^{n-1} f(\tau)\,\mathrm{d}\tau.$$

- Can be proved **by induction** using Fubini's Theorem/the previous formula,
- We have introduced the Γ function so let's use it,

A formula for repeated integration

Cauchy's formula

The indefinite integral of order $n \in \mathbb{N}$ of function $f(t)$ is given by

$$I^n_{c,t} f(t) = \int_c^t \cdots \int_c^t f(t)\,\mathrm{d}t \cdots \mathrm{d}t = \frac{1}{\Gamma(n)} \int_c^t (t-\tau)^{n-1} f(\tau)\,\mathrm{d}\tau,$$

$$I^n_{t,c} f(t) = \int_t^c \cdots \int_t^c f(t)\,\mathrm{d}t \cdots \mathrm{d}t = \frac{1}{\Gamma(n)} \int_c^t (\tau-t)^{n-1} f(\tau)\,\mathrm{d}\tau.$$

- Can be proved **by induction** using Fubini's Theorem/the previous formula,
- We have introduced the Γ function so let's use it,
- Now we use it to move from the integer case to the **real one**.

Riemann–Liouville Fractional Integrals

Riemann–Liouville Fractional Integral

Let $\Re\alpha > 0$, and let $f \in \mathbb{L}^1([a,b])$. Then for $t \in [a,b]$ we call

$$I^{\alpha}_{[a,t]} f(t) = {}_aD_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} f(\tau)\,\mathrm{d}\tau,$$

$$I^{\alpha}_{[t,b]} f(t) = {}_aD_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_t^b (\tau-t)^{\alpha-1} f(\tau)\,\mathrm{d}\tau.$$

the **Riemann–Liouville** fractional integrals of f of order α, we set it to be the identity operator whenever $\alpha = 0$.

- **the idea** is that we have substituted the integer number n of repetition of the integral with the real order α,
- but does this makes sense?

RL Fractional Integrals: properties - I

Theorem (Existence).

Lef $f \in \mathbb{L}^1[a,b]$, and $\alpha > 0$. Then, the integral $I^{\alpha}_{[a,t]} f(t)$ exists for almost every $t \in [a,b]$. Moreover, the function $I^{\alpha}_{[a,t]} f$ itself is also an element of $\mathbb{L}^1[a,b]$.

Proof. It is sufficient to recognize that we can write the integral in question as a convolution on $\mathbb{R}$, indeed:

$$\int_0^t (t-\tau)^{\alpha-1} f(\tau) \mathrm{d}\tau = \int_{-\infty}^{+\infty} \Phi_1(t-\tau)\Phi_2(\tau)\,\mathrm{d}\tau,$$

where

$$\Phi_1(u) = \begin{cases} u^{\alpha-1}, & \text{for } 0 < t \leq b-a, \\ 0, & \text{otherwise,} \end{cases} \quad \text{and } \Phi_2(u) = \begin{cases} f(u), & \text{for } u \in [a,b], \\ 0, & \text{otherwise.} \end{cases}$$

By construction both the Φ_j, $j = 1, 2$, are in $\mathbb{L}^1(\mathbb{R})$, and thus the integral exists and is a member of $\mathbb{L}^1$ as a convolution of $\mathbb{L}^1$ functions (We are using again *Fubini's Theorem*).

RL Fractional Integrals: properties - II

Theorem (Semigroup property).

The RL fractional integral operators $\{I_c^\alpha : \mathbb{L}^1[a,b] \to \mathbb{L}^1[a,b],\ \alpha \geq 0\}$ form a commutative semigroup with respect to the concatenation operation, that is

$$I_c^\alpha(I_c^\beta f(t)) = I_c^{\alpha+\beta} f(t)), \text{ and } I_c^\beta(I_c^\alpha f(t)) = I_c^{\alpha+\beta} f(t)).$$

The neutral element of this semigroup is the I_c^0 operator.

Proof. We prove it for one side, the other is analogous.

RL Fractional Integrals: properties - II

Theorem (Semigroup property).

The RL fractional integral operators $\{I_c^\alpha : \mathbb{L}^1[a,b] \to \mathbb{L}^1[a,b],\ \alpha \geq 0\}$ form a commutative semigroup with respect to the concatenation operation, that is

$$I_c^\alpha(I_c^\beta f(t)) = I_c^{\alpha+\beta} f(t)), \text{ and } I_c^\beta(I_c^\alpha f(t)) = I_c^{\alpha+\beta} f(t)).$$

The neutral element of this semigroup is the I_c^0 operator.

Proof. We have just proved that the integral exists, then by using *Fubini's theorem* we can interchange the order of integration:

$$I_{[a,t]}^\alpha I_{[a,t]}^\beta f(x) = \frac{1}{\Gamma(\alpha)\Gamma(\beta)} \int_a^x \int_\tau^x (x-t)^{\alpha-1}(t-\tau)^{\beta-1} f(\tau)\,\mathrm{d}\tau\,\mathrm{d}t$$

RL Fractional Integrals: properties - II

Theorem (Semigroup property).

The RL fractional integral operators $\{I_c^\alpha : \mathbb{L}^1[a,b] \to \mathbb{L}^1[a,b],\ \alpha \geq 0\}$ form a commutative semigroup with respect to the concatenation operation, that is

$$I_c^\alpha(I_c^\beta f(t)) = I_c^{\alpha+\beta} f(t)), \text{ and } I_c^\beta(I_c^\alpha f(t)) = I_c^{\alpha+\beta} f(t)).$$

The neutral element of this semigroup is the I_c^0 operator.

Proof. We have just proved that the integral exists, then by using *Fubini's theorem* we can interchange the order of integration:

$$I_{[a,t]}^\alpha I_{[a,t]}^\beta f(x) = \frac{1}{\Gamma(\alpha)\Gamma(\beta)} \int_a^x f(\tau) \int_\tau^x (x-t)^{\alpha-1}(t-\tau)^{\beta-1}\,\mathrm{d}t\,\mathrm{d}\tau.$$

RL Fractional Integrals: properties - II

Theorem (Semigroup property).

The RL fractional integral operators $\{I_c^\alpha : \mathbb{L}^1[a,b] \to \mathbb{L}^1[a,b],\ \alpha \geq 0\}$ form a commutative semigroup with respect to the concatenation operation, that is

$$I_c^\alpha(I_c^\beta f(t)) = I_c^{\alpha+\beta} f(t)), \text{ and } I_c^\beta(I_c^\alpha f(t)) = I_c^{\alpha+\beta} f(t)).$$

The neutral element of this semigroup is the I_c^0 operator.

Proof. We have just proved that the integral exists, then by using *Fubini's theorem* we can interchange the order of integration:

$$I_{[a,t]}^\alpha I_{[a,t]}^\beta f(x) = \frac{1}{\Gamma(\alpha)\Gamma(\beta)} \int_a^x f(\tau) \int_\tau^x (x-t)^{\alpha-1}(t-\tau)^{\beta-1}\, \mathrm{d}t\, \mathrm{d}\tau.$$

We now use the substitution $t = \tau + s(x-\tau)$, $\mathrm{d}t = (x-\tau)\mathrm{d}s$.

RL Fractional Integrals: properties - II

Theorem (Semigroup property).

The RL fractional integral operators $\{I_c^\alpha : \mathbb{L}^1[a,b] \to \mathbb{L}^1[a,b],\ \alpha \geq 0\}$ form a commutative semigroup with respect to the concatenation operation, that is

$$I_c^\alpha(I_c^\beta f(t)) = I_c^{\alpha+\beta} f(t)), \text{ and } I_c^\beta(I_c^\alpha f(t)) = I_c^{\alpha+\beta} f(t)).$$

The neutral element of this semigroup is the I_c^0 operator.

Proof. We have just proved that the integral exists, then by using *Fubini's theorem* we can interchange the order of integration. We now use the substitution $t = \tau + s(x-\tau)$, $\mathrm{d}t = (x-\tau)\mathrm{d}s$. We obtain:

$$I_{[a,t]}^\alpha I_{[a,t]}^\beta f(x) = \frac{1}{\Gamma(\alpha)\Gamma(\beta)} \int_a^x f(\tau) \int_0^1 [(x-\tau)(1-s)]^{\alpha-1} [s(x-\tau)]^{\beta-1} (x-\tau)\, \mathrm{d}s\, \mathrm{d}\tau$$

RL Fractional Integrals: properties - II

Theorem (Semigroup property).

The RL fractional integral operators $\{I_c^\alpha : \mathbb{L}^1[a,b] \to \mathbb{L}^1[a,b],\ \alpha \geq 0\}$ form a commutative semigroup with respect to the concatenation operation, that is

$$I_c^\alpha(I_c^\beta f(t)) = I_c^{\alpha+\beta} f(t)), \text{ and } I_c^\beta(I_c^\alpha f(t)) = I_c^{\alpha+\beta} f(t)).$$

The neutral element of this semigroup is the I_c^0 operator.

Proof. We have just proved that the integral exists, then by using *Fubini's theorem* we can interchange the order of integration. We now use the substitution $t = \tau + s(x-\tau)$, $\mathrm{d}t = (x-\tau)\mathrm{d}s$. We obtain:

$$I_{[a,t]}^\alpha I_{[a,t]}^\beta f(x) = \frac{1}{\Gamma(\alpha)\Gamma(\beta)} \int_a^x f(\tau)(x-\tau)^{\alpha+\beta-1} \int_0^1 (1-s)^{\alpha-1} s^{\beta-1}\, \mathrm{d}s\, \mathrm{d}\tau.$$

RL Fractional Integrals: properties - II

Euler's β-function

The Euler's β-function is defined as:

$$\beta(x,y) \triangleq \int_0^1 u^{x-1}(1-u)^{y-1}\,du = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)} \quad \Re x > 0, \Re y > 0,$$

Proof. We have just proved that the integral exists, then by using *Fubini's theorem* we can interchange the order of integration. We now use the substitution $t = \tau + s(x-\tau)$, $\mathrm{d}t = (x-\tau)\mathrm{d}s$. We obtain:

$$I^{\alpha}_{[a,t]} I^{\beta}_{[a,t]} f(x) = \frac{1}{\Gamma(\alpha)\Gamma(\beta)} \int_a^x f(\tau)(x-\tau)^{\alpha+\beta-1} \int_0^1 (1-s)^{\alpha-1} s^{\beta-1}\,\mathrm{d}s\,\mathrm{d}\tau.$$

RL Fractional Integrals: properties - II

Theorem (Semigroup property).

The RL fractional integral operators $\{I_c^\alpha : \mathbb{L}^1[a,b] \to \mathbb{L}^1[a,b],\ \alpha \geq 0\}$ form a commutative semigroup with respect to the concatenation operation, that is

$$I_c^\alpha(I_c^\beta f(t)) = I_c^{\alpha+\beta} f(t)), \text{ and } I_c^\beta(I_c^\alpha f(t)) = I_c^{\alpha+\beta} f(t)).$$

The neutral element of this semigroup is the I_c^0 operator.

Proof. We have just proved that the integral exists, then by using *Fubini's theorem* we can interchange the order of integration. We now use the substitution $t = \tau + s(x-\tau)$, $\mathrm{d}t = (x-\tau)\mathrm{d}s$. We obtain:

$$I_{[a,t]}^\alpha I_{[a,t]}^\beta f(x) = \frac{1}{\Gamma(\alpha+\beta)} \int_a^x (x-\tau)^{\alpha+\beta-1} f(\tau)\,\mathrm{d}\tau = I_{[a,t]}^{\alpha+\beta} f(x), \quad \text{a.e. on } [a,b].$$

The same works also if we exchange α and β, while we have the 0th order operator being the neutral element by definition.

RL Fractional Integrals: properties - III

A note on regularity.

Observe that in the proof we could say something more on the regularity of the resulting functions. Indeed if f is a continuous function on $[a, b]$, then also $I^{\alpha}_{[a,t]} f$ is continuous.

Therefore we have that also the concatenation $I^{\alpha}_{[a,t]} I^{\beta}_{[a,t]}$ and $I^{\alpha+\beta}_{[a,t]}$ are continuous. Then what we have proved is that we have two continuous function that are **almost everywhere equal**, and therefore they most coincide everywhere. Furthermore, if $f \in \mathbb{L}^1[a, b]$ and $\alpha + \beta \geq 1$ we can use Semigroup property to write

$$I^{\alpha}_{[a,t]} I^{\beta}_{[a,t]} f = I^{\alpha+\beta}_{[a,t]} f = I^{\alpha+\beta-1}_{[a,t]} I^{1}_{[a,t]} f, \ \ a.e.$$

Now, since $I^{1}_{[a,t]} f$ is continuos, we also get that the other two way of writing it are continuous, and thus we can conclude the equality everywhere by the same argument as before.

Computing a Riemann–Liouville fractional integral.

$$I^{\alpha}_{[0,t]} t^{\mu} = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} \tau^{\mu} \, d\tau,$$

This should be the simplest possible example, and indeed it is as simple as using again the **Euler β Function**:

$$\beta(x,y) \triangleq \int_0^1 u^{x-1}(1-u)^{y-1} \, du = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)} \quad \Re x > 0, \Re y > 0.$$

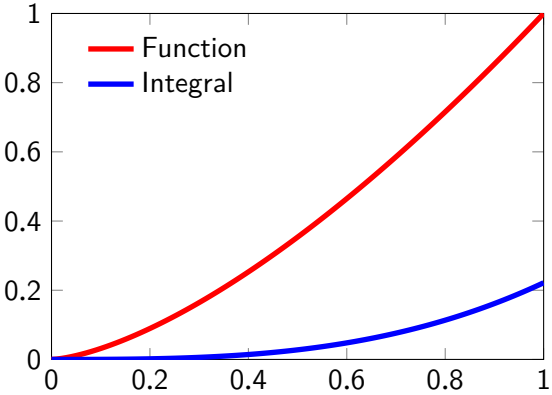
To obtain it, we do the substitution for $u = \frac{\tau}{t}$, then

$$\begin{aligned} I^{\alpha}_{[0,t]} t^{\mu} =& \frac{t^{\alpha+\mu}}{\Gamma(\alpha)} \int_0^1 u^{\mu}(1-u)^{\alpha-1} \, du \\ =& \frac{t^{\alpha+\mu}}{\Gamma(\alpha)} \frac{\Gamma(\mu+1)\Gamma(\alpha)}{\Gamma(\alpha+\mu+1)} = \frac{\Gamma(\mu+1)}{\Gamma(\alpha+\mu+1)} t^{\alpha+\mu}. \end{aligned}$$

Computing a Riemann–Liouville fractional integral.

$$I^{\alpha}_{[0,t]} t^{\mu} = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} \tau^{\mu} \, d\tau = \frac{\Gamma(\mu+1)}{\Gamma(\alpha+\mu+1)} t^{\alpha+\mu},$$

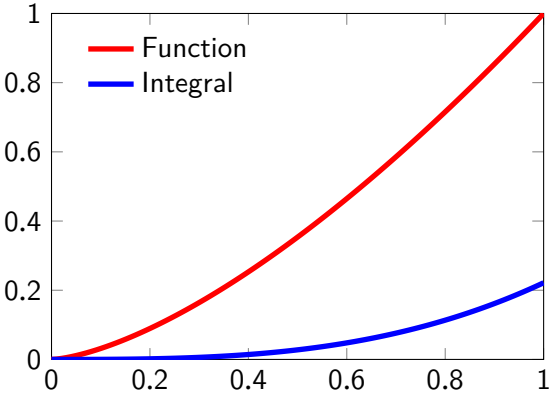
```
t = linspace(0,1,100);
I = @(alpha,mu,t)
↪  gamma(mu+1)*t.^(alpha+mu)/
↪  gamma(alpha+mu+1);
mu = 1.5;
alpha = 1.5;
plot(t,t.^mu,'r-',t,I(alpha,mu,t),
↪  'b-','Linewidth',2);
legend('Function','Integral');
```



Computing a Riemann–Liouville fractional integral.

$$I^{\alpha}_{[0,t]} t^{\mu} = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} \tau^{\mu} \, d\tau = \frac{\Gamma(\mu+1)}{\Gamma(\alpha+\mu+1)} t^{\alpha+\mu},$$

```
t = linspace(0,1,100);
I = @(alpha,mu,t)
↪  gamma(mu+1)*t.^(alpha+mu)/
↪  gamma(alpha+mu+1);
mu = 1.5;
alpha = 1.5;
plot(t,t.^mu,'r-',t,I(alpha,mu,t),
↪  'b-','Linewidth',2);
legend('Function','Integral');
```



☹ They are hard to compute!

Quadratures for Fractional Integrals

Quadrature idea

Let us assume that $f(t)$ is suitably smooth on an interval (a, b). Let

$$h = \frac{b-a}{N}, \quad t_k = a + kh, \quad \text{with } k = 0, 1, 2, \ldots, N, \quad N \in \mathbb{N}$$

then we can approximate for $t = t_N$ the fractional integral as

$${}_aD_b^{-\alpha} f(t)\big|_{t=t_N} = \frac{1}{\Gamma(\alpha)} \int_a^{t_N} (t_N - \tau)^{\alpha-1} f(\tau)\,\mathrm{d}\tau = \frac{1}{\Gamma(\alpha)} \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} (t_k - \tau)^{\alpha-1} f(\tau)\,\mathrm{d}\tau.$$

We approximate $f(x)$ with a polynomial $p(x)$ such that we can compute exactly the involved integrals, this yields quadratures by the usual look

$${}_aD_b^{-\alpha} f(t)\big|_{t=t_N} \approx \sum_{k=0}^{N-1} \omega_k f(t_k).$$

Piecewise constant approximation

We approximate $f(t)$ on the intervals $[t_k, t_k+1)$, $k = 0, \ldots, N-1$, selecting

$$f(t) \approx p(t) \equiv p(t_k), \quad t \in [t_k, t_k+1), \; k = 0, 1, \ldots, N-1,$$

from which we get the formula

$$\begin{aligned}
{}_aD_b^{-\alpha} f(t)\big|_{t=t_N} &\approx \frac{1}{\Gamma(\alpha)} \sum_{k=0}^{N-1} f(t_k) \int_{t_k}^{t_k+1} (t_N - \tau)^{\alpha-1} \, \mathrm{d}\tau = \frac{1}{\Gamma(\alpha)} \sum_{k=0}^{N-1} f(t_k) \left[-\frac{1}{\alpha} (t_N - \tau)^{\alpha} \right]_{t_k}^{t_{k+1}} \\
&= \sum_{k=0}^{N-1} f(t_k) \frac{1}{\alpha\Gamma(\alpha)} \left[(t_N - t_k)^{\alpha} - (t_N - t_{k+1})^{\alpha} \right] \\
&= \sum_{k=0}^{N-1} f(t_k) \frac{1}{\alpha\Gamma(\alpha)} \left[(a + hn - a - kh)^{\alpha} - (a + hn - a - (k+1)h)^{\alpha} \right] \\
&= \sum_{k=0}^{N-1} f(t_k) \frac{h^{\alpha}}{\Gamma(\alpha+1)} \left[(n-k)^{\alpha} - (N-k-1)^{\alpha} \right] = \sum_{k=0}^{N-1} b_{N-k-1} f(t_k),
\end{aligned}$$

Piecewise constant approximation

We approximate $f(t)$ on the intervals $[t_k, t_k+1)$, $k = 0, \ldots, N-1$, selecting

$$f(t) \approx p(t) \equiv p(t_k), \quad t \in [t_k, t_k+1), \; k = 0, 1, \ldots, N-1,$$

from which we get the formula

$${}_aD_b^{-\alpha} f(t)\big|_{t=t_N} \approx \sum_{k=0}^{N-1} b_{N-k-1} f(t_k),$$

where we have defined

$$b_k = \frac{h^\alpha}{\Gamma(\alpha+1)}[(k+1)^\alpha - k^\alpha], \qquad 0 \le k \le N-1.$$

Piecewise constant approximation

We approximate $f(t)$ on the intervals $[t_k, t_k+1)$, $k = 0, \ldots, N-1$, selecting

$$f(t) \approx p(t) \equiv p(t_k), \quad t \in [t_k, t_k+1), \ k = 0, 1, \ldots, N-1,$$

from which we get the formula

$${}_aD_b^{-\alpha} f(t)\big|_{t=t_N} \approx \sum_{k=0}^{N-1} b_{N-k-1} f(t_k), \qquad b_k = \frac{h^\alpha}{\Gamma(\alpha+1)}[(k+1)^\alpha - k^\alpha].$$

Analogously we get the case in which we select the **right approximation**

$$f(t) \approx p(t) \equiv p(t_{k+1}), \quad t \in [t_k, t_k+1), \ k = 0, 1, \ldots, N-1,$$

and, more generally, for the **weighted formula** in which we select

$$f(t) \approx p(t) \equiv \lambda p(t_k) + (1-\lambda) p(t_{k+1}), \quad t \in [t_k, t_k+1), \ k = 0, 1, \ldots, N-1, \ \lambda \in [0, 1].$$

Piecewise constant approximation

We approximate $f(t)$ on the intervals $[t_k, t_k+1)$, $k = 0, \ldots, N-1$, selecting

$$f(t) \approx p(t) \equiv p(t_k), \quad t \in [t_k, t_k+1), \; k = 0, 1, \ldots, N-1,$$

from which we get the formula

$${}_aD_b^{-\alpha} f(t)\big|_{t=t_N} \approx \sum_{k=0}^{N-1} b_{N-k-1} f(t_k), \qquad b_k = \frac{h^\alpha}{\Gamma(\alpha+1)}[(k+1)^\alpha - k^\alpha].$$

The general **weighted formula** is then given by

$${}_aD_b^{-\alpha} f(t)\big|_{t=t_N} \approx \sum_{k=0}^{N-1} b_{N-k-1} \left[\lambda p(t_k) + (1-\lambda) p(t_{k+1})\right], \quad \lambda \in [0,1].$$

Implementation

This is a simple procedure to implement

```
function I = constfracint(f,a,t,alpha,N,lambda)
%CONSTFRACINT computes the fractional integral with the weighted piecewise
%constant approximation of the function f between a and t, over N uniformly
%distributed intervals.
h  = (t-a)/N;
tk = (a:h:t)';
b  = zeros(N,1);
for k=0:N-1
   b(k+1) = (k+1)^alpha - k^alpha;
end
b = h^alpha*b/gamma(alpha+1);
p = f(tk);
I = flipud(b)'*(lambda*p(1:N) + (1-lambda)*p(2:N+1));
end
```

Implementation - II

And we can **test the results** using the fractional integral we have computed by hand

```
f     = @(t,mu) t.^mu;
Itrue = @(alpha,mu,t) gamma(mu+1)*t.^(alpha+mu)/ gamma(alpha+mu+1);
mu    = 1;
alpha = 1.5;

N = 100;
lambda = 1;
I = constfracint(@(t) f(t,mu),0,1,alpha,N,1);
fprintf('Relative error is: %e\n',abs(I-Itrue(alpha,mu,1))./abs(Itrue(alpha,mu,1)));
```

That returns us

```
Relative error is: 1.246939e-02
```

But what about convergence?

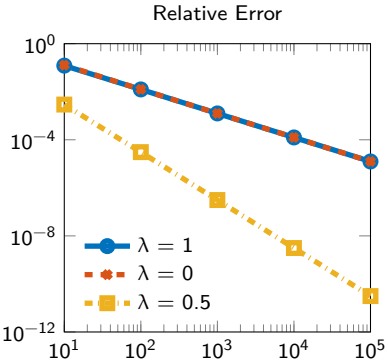
Convergence

Fractional Newton-Cotes formula

Lef $f(t)$ be approximated by a polynomial $p_{k,r}(t)$ of degree r on the grid points $\{t_k = t_0^{(k)}, \ldots, t_r^{(k)} = t_{k+1}\}$. Then the error estimate for an $f \in \mathcal{C}^{r+1}([a,b])$ on each sub-interval $[t_k, t_{k+1}]$ is given by

$$f(t) - p_{k,r}(t) = \frac{f^{(r+1)}(\tau_k)}{(r+1)!} \prod_{j=0}^{r} (t - t_j^{(k)}),$$

for $r \in \mathbb{N}$, $t, \tau_k \in [t_k, t_{k+1}]$, i.e., the formula is of order $O(h^{r+1})$.



N/λ	0	1	1/2
1e+01	1.3e-01	1.2e-01	2.9e-03
1e+02	1.3e-02	1.2e-02	3.1e-05
1e+03	1.3e-03	1.2e-03	3.1e-07
1e+04	1.3e-04	1.2e-04	3.1e-09
1e+05	1.3e-05	1.2e-05	3.1e-11

Convergence

Proof. The interpolating polynomial can be expressed in the **Lagrange basis**

$$p_{k,r}(t) = \sum_{i=0}^{r} l_{k,i}(t) f(t_i^{(k)}), \quad l_{k,i}(t) = \prod_{\substack{j=0 \\ j\neq i}}^{r} \frac{t - t_j^{(k)}}{t_i^{(k)} - t_j^{(k)}}, \quad 0 \leq i \leq r, \ t \in [t_k, t_{k+1}].$$

Then the fractional Newton-Coates formula si given by

$${}_aD_b^{-\alpha} f(t)\big|_{t=t_N} \approx {}_aD_b^{-\alpha} p_{k,r}(t)\big|_{t=t_N} = \sum_{k=0}^{N-1} \sum_{i=0}^{r} C_{i,N}^{(k)} f(t_i^{(k)}),$$

for

$$C_{i,N}^{(k)} = \frac{1}{\Gamma(\alpha)} \int_{t_k}^{t_{k+1}} (t_N - \tau)^{\alpha-1} l_{k,i}(\tau)\, \mathrm{d}\tau.$$

Convergence

Proof. Then the fractional Newton-Coates formula si given by

$$ {}_aD_b^{-\alpha} f(t)\big|_{t=t_N} \approx {}_aD_b^{-\alpha} p_{k,r}(t)\big|_{t=t_N} = \sum_{k=0}^{N-1} \sum_{i=0}^{r} C_{i,N}^{(k)} f(t_i^{(k)}), $$

from which we obtain the error estimate as

$$ \begin{aligned} \left| {}_aD_b^{-\alpha} f(t) - {}_aD_b^{-\alpha} p_{k,r}(t) \right| &\leq \frac{1}{\Gamma(\alpha)} \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} (t_N - \tau)^{\alpha-1} \left| f(\tau) - p_{k,r}(\tau) \right| \, \mathrm{d}\tau \\ &\leq \max_{t \in [a,t_N]} \frac{|f^{(r+1)}(t)|}{(r+1)!\Gamma(\alpha)} \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} (t_N - \tau)^{\alpha-1} \prod_{j=0}^{r} |\tau - t_j^{(k)}| \, \mathrm{d}\tau \\ &\leq \max_{t \in [a,t_N]} \left| f^{(r+1)}(t) \right| \frac{h^{r+1}}{(r+1)!\Gamma(\alpha)} \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} (t_N - \tau)^{\alpha-1} \, \mathrm{d}\tau. \end{aligned} $$

Convergence

Proof. Then the fractional Newton-Coates formula si given by

$$ {}_aD_b^{-\alpha} f(t)\big|_{t=t_N} \approx {}_aD_b^{-\alpha} p_{k,r}(t)\big|_{t=t_N} = \sum_{k=0}^{N-1} \sum_{i=0}^{r} C_{i,N}^{(k)} f(t_i^{(k)}), $$

from which we obtain the error estimate as

$$ \begin{aligned} \left| {}_aD_b^{-\alpha} f(t) - {}_aD_b^{-\alpha} p_{k,r}(t) \right| &\leq \frac{1}{\Gamma(\alpha)} \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} (t_N - \tau)^{\alpha-1} \left| f(\tau) - p_{k,r}(\tau) \right| \, \mathrm{d}\tau \\ &\leq \max_{t\in[a,t_N]} \frac{|f^{(r+1)}(t)|}{(r+1)!\Gamma(\alpha)} \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} (t_N - \tau)^{\alpha-1} \prod_{j=0}^{r} |\tau - t_j^{(k)}| \, \mathrm{d}\tau \\ &\leq \max_{t\in[a,t_N]} \left| f^{(r+1)}(t) \right| \frac{h^{r+1}}{(r+1)!\Gamma(\alpha+1)} (t_N - t_0)^{\alpha}. \end{aligned} $$

Convergence

Proof. Then the fractional Newton-Coates formula si given by

$$ {}_aD_b^{-\alpha} f(t)\Big|_{t=t_N} \approx {}_aD_b^{-\alpha} p_{k,r}(t)\Big|_{t=t_N} = \sum_{k=0}^{N-1} \sum_{i=0}^{r} C_{i,N}^{(k)} f(t_i^{(k)}), $$

from which we obtain the error estimate as

$$ \left| {}_aD_b^{-\alpha} f(t) - {}_aD_b^{-\alpha} p_{k,r}(t) \right| \in O(h^{r+1}). $$

Remark

The error estimate does not coincide completely with the classical one for Newton-Coates formulas, this is due to the nonsymmetry of the integral kernel $(t_N' - t)^{\alpha-1}$.

Suggested exercises, and some extensions

(i) Rewrite (and implement) the fractional weighted constant approximation for the *other-sided* Riemann-Liuoville fractional integral,

(ii) Denote with $t_{k+1/2} = {}^{t_k+t_{k+1}}/2$ on each sub-interval $[t_k, t_{k+1}]$, approximate $f(t)$ with a *piecewise quadratic polynomial*, derive and implement the fractional Simpson's formula
   – ⚠️ The closed form of the coefficients for this case is cumbersome...

Extensions

By mimicking the usual procedure for deriving collocation/spectral type quadrature formulas, we could approximate $f(t)$ by using, e.g., Jacobi polynomials to obtain the related quadrature formulas (when you have obtained formulas for Jacobi, then *Chebyshev* and *Legendre* follow with relative "ease").

Riemann–Liouville Fractional Derivatives

Now that we've gotten a little bit of familiarity with Riemann–Liouville integral operators, we can finally **introduce the corresponding differential operators**.

The key idea

Let f be a function having a continuous nth derivative on the interval $[a, b]$, and let $m \in \mathbb{N}$ be such that $m > n$, then

$$\frac{d^n}{dt^n} f(t) = \frac{1}{(m-n-1)!} \frac{d^m}{dt^m} \int_a^t (t-\tau)^{m-n-1} f(\tau)\, \mathrm{d}\tau = \frac{d^m}{dt^m} I_a^{m-n} f,$$

simply by employing the Fundamental Theorem of (Classical) Calculus

$$f = \frac{d^{m-n}}{dt^{m-n}} I_a^{m-n} f,$$

and applying the operator $\frac{d^n}{dt^n}$ to both side of it.

Riemann–Liouville Fractional Derivatives

Now that we've gotten a little bit of familiarity with Riemann–Liouville integral operators, we can finally **introduce the corresponding differential operators**.

The key idea now we go from integers to real numbers!

Let f be a function having a continuous nth derivative on the interval $[a, b]$, and let $m \in \mathbb{N}$ be such that $m > n$, then

$$\frac{d^n}{dt^n} f(t) = \frac{1}{(m-n-1)!} \frac{d^m}{dt^m} \int_a^t (t-\tau)^{m-n-1} f(\tau)\, \mathrm{d}\tau = \frac{d^m}{dt^m} I_a^{m-n} f,$$

simply by employing the Fundamental Theorem of (Classical) Calculus

$$f = \frac{d^{m-n}}{dt^{m-n}} I_a^{m-n} f,$$

and applying the operator $\frac{d^n}{dt^n}$ to both side of it.

Riemann–Liouville Fractional Derivatives

Substitute the integer n with a real positive number α and select an $m \in \mathbb{N}$ s.t. $m > \alpha$.

RL Derivative

Let $\alpha \in \mathbb{R}_+$ and $m = \lceil \alpha \rceil$, we define the Riemann-Liouville operator ${}_{\mathrm{RL}}D_a^{\alpha}$ as

$$ {}_{\mathrm{RL}}D_a^{\alpha} f(t) \triangleq \frac{d^m}{dt^m} I_a^{m-\alpha} f(t), $$

and we set ${}_{\mathrm{RL}}D_a^{0}$ to the identity operator.

⚠ The right-hand side of our definition remains valid, **but** now the resulting operator depends on the choice of the point a.

Riemann–Liouville Fractional Derivatives

Substitute the integer n with a real positive number α and select an $m \in \mathbb{N}$ s.t. $m > \alpha$.

RL Derivative

Let $\alpha \in \mathbb{R}_+$ and $m = \lceil \alpha \rceil$, we define the Riemann-Liouville operator ${}_{\mathrm{RL}}D_a^\alpha$ as

$$ {}_{\mathrm{RL}}D_a^\alpha f(t) \triangleq \frac{d^m}{dt^m} I_a^{m-\alpha} f(t), $$

and we set ${}_{\mathrm{RL}}D_a^0$ to the identity operator.

⚠ The right-hand side of our definition remains valid, **but** now the resulting operator depends on the choice of the point a.

? for what functions f does this definition make sense?

Riemann–Liouville Fractional Derivatives Existence

The $\mathbb{A}^n$ functions

We call $\mathbb{A}^n[a,b]$, or simply $\mathbb{A}^n$ when the interval is clear from the context, the space of function with an **absolutely continuous** $(n-1)$st derivative, i.e., the functions f for which there exists almost everywhere a (generalized) nth derivative function $g \in \mathbb{L}^1[a,b]$ for which holds

$$f^{(n-1)}(t) = f^{(n-1)}(a) + \int_a^t g(\tau)\,\mathrm{d}\tau.$$

Remind: For a compact interval:

continuously differentiable $\subseteq$ Lipschitz continuous $\subseteq$ absolutely continuous $\subseteq$ bounded variation $\subseteq$ differentiable almost everywhere

Example: $f(t) = \sqrt[3]{t}$ is absolutely continuous on any bounded interval I **but** not Lipschitz continuous on any interval I such that $0 \in I$.

Riemann–Liouville Fractional Derivatives Existence

Theorem (Existence)

Lef $f \in \mathbb{A}^1[a,b]$, and $0 < \alpha < 1$. Then ${}_{\mathrm{RL}}D_a^{\alpha} f(t)$ exists almost everywhere in $[a,b]$. Moreover, ${}_{\mathrm{RL}}D_a^{\alpha} f(t) \in \mathbb{L}^p$ for $1 \leq p < \alpha^{-1}$ and

$$ {}_{\mathrm{RL}}D_a^{\alpha} f(t) = \frac{1}{\Gamma(1-\alpha)} \left(\frac{f(a)}{(t-a)^{\alpha}} + \int_a^t f'(\tau)(t-\tau)\,\mathrm{d}\tau \right). $$

Proof. We use directly the two definitions

$$ \begin{aligned} {}_{\mathrm{RL}}D_a^{\alpha} f(t) &= \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_a^t f(\tau)(t-\tau)^{-\alpha}\,\mathrm{d}\tau \\ &= \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_a^t \left(f(a) + \int_a^{\tau} f'(s)\,\mathrm{d}s \right) (t-\tau)^{-\alpha}\,\mathrm{d}\tau \end{aligned} $$

Riemann–Liouville Fractional Derivatives Existence

Theorem (Existence)

Lef $f \in \mathbb{A}^1[a,b]$, and $0 < \alpha < 1$. Then ${}_{\mathrm{RL}}D_a^{\alpha} f(t)$ exists almost everywhere in $[a,b]$. Moreover, ${}_{\mathrm{RL}}D_a^{\alpha} f(t) \in \mathbb{L}^p$ for $1 \leq p < \alpha^{-1}$ and

$$ {}_{\mathrm{RL}}D_a^{\alpha} f(t) = \frac{1}{\Gamma(1-\alpha)} \left(\frac{f(a)}{(t-a)^{\alpha}} + \int_a^t f'(\tau)(t-\tau)\,\mathrm{d}\tau \right). $$

Proof. We use directly the two definitions

$$ \begin{aligned} {}_{\mathrm{RL}}D_a^{\alpha} f(t) &= \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_a^t \left(f(a) + \int_a^{\tau} f'(s)\,\mathrm{d}s \right) (t-\tau)^{-\alpha}\,\mathrm{d}\tau \\ &= \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \left(f(a) \int_a^t \frac{\mathrm{d}t}{(x-t)^{\alpha}} + \int_a^t \int_a^{\tau} f'(s)(t-\tau)^{-\alpha}\,\mathrm{d}s\,\mathrm{d}\tau \right) \end{aligned} $$

Riemann–Liouville Fractional Derivatives Existence

Theorem (Existence)

Lef $f \in \mathbb{A}^1[a,b]$, and $0 < \alpha < 1$. Then ${}_{\mathrm{RL}}D_a^{\alpha} f(t)$ exists almost everywhere in $[a,b]$. Moreover, ${}_{\mathrm{RL}}D_a^{\alpha} f(t) \in \mathbb{L}^p$ for $1 \leq p < \alpha^{-1}$ and

$$ {}_{\mathrm{RL}}D_a^{\alpha} f(t) = \frac{1}{\Gamma(1-\alpha)} \left(\frac{f(a)}{(t-a)^{\alpha}} + \int_a^t f'(\tau)(t-\tau)\,\mathrm{d}\tau \right). $$

Proof. We use directly the two definitions, and apply again *Fubini's Theorem*

$$ \begin{aligned} {}_{\mathrm{RL}}D_a^{\alpha} f(t) &= \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \left(f(a) \int_a^t \frac{\mathrm{d}t}{(x-t)^{\alpha}} + \int_a^{\tau} \int_a^t f'(s)(t-\tau)^{-\alpha}\,\mathrm{d}s\,\mathrm{d}\tau \right) \\ &= \frac{1}{\Gamma(1-\alpha)} \left(\frac{f(a)}{(t-a)^{\alpha}} + \frac{d}{dt} \int_a^{\tau} \int_a^t f'(s)(t-\tau)^{-\alpha}\,\mathrm{d}s\,\mathrm{d}\tau \right) \\ \text{(Fubini)} \quad &= \frac{1}{\Gamma(1-\alpha)} \left(\frac{f(a)}{(t-a)^{\alpha}} + \frac{d}{dt} \int_a^t f'(s) \frac{(t-s)^{1-\alpha}}{1-\alpha}\,\mathrm{d}s \right), \end{aligned} $$

Riemann–Liouville Fractional Derivatives Existence

Theorem (Existence)

Lef $f \in \mathbb{A}^1[a,b]$, and $0 < \alpha < 1$. Then ${}_{\mathrm{RL}}D_a^\alpha f(t)$ exists almost everywhere in $[a,b]$. Moreover, ${}_{\mathrm{RL}}D_a^\alpha f(t) \in \mathbb{L}^p$ for $1 \leq p < \alpha^{-1}$ and

$$ {}_{\mathrm{RL}}D_a^\alpha f(t) = \frac{1}{\Gamma(1-\alpha)} \left(\frac{f(a)}{(t-a)^\alpha} + \int_a^t f'(\tau)(t-\tau)\,\mathrm{d}\tau \right). $$

Proof. We use directly the two definitions, and finally Leibniz rule for the derivative of integral functions,

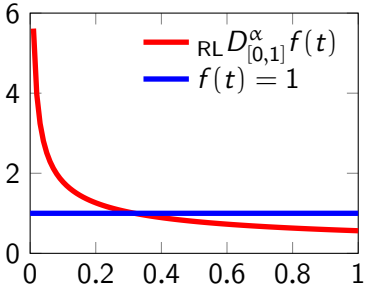
$$ \begin{aligned} {}_{\mathrm{RL}}D_a^\alpha f(t) &= \frac{1}{\Gamma(1-\alpha)} \left(\frac{f(a)}{(t-a)^\alpha} + \frac{d}{dt} \int_a^t f'(s) \frac{(t-s)^{1-\alpha}}{1-\alpha}\,\mathrm{d}s \right), \\ \text{(Leibniz)} \quad &= \frac{1}{\Gamma(1-\alpha)} \left(\frac{f(a)}{(t-a)^\alpha} + \int_a^t f'(\tau)(t-\tau)\,\mathrm{d}\tau \right). \end{aligned} $$

Computing our first RL derivatives

To keep things simple we can compute, first of all, the fractional derivative of order $\alpha \in (0,1)$ of the **constant function** $f(t) = 1$ in $[0, t]$:

We simply apply the previous representation theorem, and thus:

$$
\begin{aligned}
{}_{\mathrm{RL}}D^{\alpha}_{[0,1]} f(t) &= \frac{1}{\Gamma(1-\alpha)} \left(\frac{f(0)}{(t-0)^{\alpha}} + \int_0^t f'(\tau)(t-\tau)\,\mathrm{d}\tau \right) = \\
&= \frac{1}{\Gamma(1-\alpha)} \frac{1}{(t-0)^{\alpha}} = \frac{t^{-\alpha}}{\Gamma(1-\alpha)}
\end{aligned}
$$

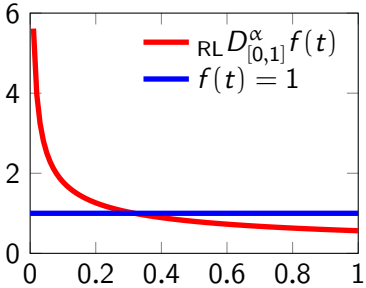


Computing our first RL derivatives

To keep things simple we can compute, first of all, the fractional derivative of order $\alpha \in (0,1)$ of the **constant function** $f(t) = 1$ in $[0,t]$:

We simply apply the previous representation theorem, and thus:

$$
\begin{aligned}
{}_{\mathrm{RL}}D^{\alpha}_{[0,1]} f(t) &= \frac{1}{\Gamma(1-\alpha)} \left(\frac{f(0)}{(t-0)^{\alpha}} + \int_0^t f'(\tau)(t-\tau)\,\mathrm{d}\tau \right) = \\
&= \frac{1}{\Gamma(1-\alpha)} \frac{1}{(t-0)^{\alpha}} = \frac{t^{-\alpha}}{\Gamma(1-\alpha)}
\end{aligned}
$$



The RL derivative of a constant is not zero!

Computing our first RL derivatives

Let $f(t) = (t-a)^\beta$ for some $\beta > -1$ and compute its RL derivative of order $\alpha > 0$ on an interval $[a, b]$.
First we compute the **fractional integral** part of the definition:

$$
\begin{aligned}
I^\alpha_{[a,t]} f(t) =& \frac{1}{\Gamma(\alpha)} \int_a^t (\tau - a)^\beta (t-\tau)^{\alpha-1} \, \mathrm{d}\tau = \\
=& \frac{1}{\Gamma(\alpha)} \int_0^{t-a} s^\beta (t-a-s)^{\alpha-1} \, \mathrm{d}s = \leftarrow \left(\int_0^x s^{\beta-1}(x-s)^{\alpha-1} \mathrm{d}s = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} x^{\alpha+\beta-1} \right) \\
=& \frac{\Gamma(\beta+1)}{\Gamma(\alpha+\beta+1)} (t-a)^{\alpha+\beta},
\end{aligned}
$$

Computing our first RL derivatives

Let $f(t) = (t-a)^\beta$ for some $\beta > -1$ and compute its RL derivative of order $\alpha > 0$ on an interval $[a, b]$.

First we compute the **fractional integral** part of the definition:

$$I^\alpha_{[a,t]} f(t) = \frac{\Gamma(\beta+1)}{\Gamma(\alpha+\beta+1)}(t-a)^{\alpha+\beta},$$

Then we just have to compute the derivative with the correct indexes

$${}_{\mathrm{RL}}D^\alpha_{[0,1]} f(t) = \frac{d^{\lceil\alpha\rceil}}{dt^{\lceil\alpha\rceil}} I^{\lceil\alpha\rceil-\alpha}_{[a,t]} f(t) = \frac{\Gamma(\beta+1)}{\Gamma(\lceil\alpha\rceil-\alpha+\beta+1)} \frac{d^{\lceil\alpha\rceil}}{dt^{\lceil\alpha\rceil}}(\cdot - a)^{\lceil\alpha\rceil-\alpha+\beta}\Big|_t,$$

now, if $\alpha - \beta \in \mathbb{N}$ the right-hand side vanishes ($\lceil\alpha\rceil$-derivative of a polynomial of lower degree), if $\alpha - \beta \notin \mathbb{N}$, we find

$${}_{\mathrm{RL}}D^\alpha_{[0,1]} f(t) = \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\alpha)}(t-a)^{\beta-\alpha}.$$

Summary and anticipations

We did

- Definition and properties of Riemann–Liouville Integrals,
- Some examples of Fractional Newton-Cotes formulas for RL integral computations,
- Definition and existence of Riemann–Liouville Derivatives,
- A couple of by-hand computations of RL derivatives of simple functions.

Next up

- Properties and interactions between Riemann–Liouville Integrals and Derivatives,
- The Caputo fractional derivative,
- An introduction to Fractional Differential Equations.

Bibliography I

- Bagley, R. L. and P. J. Torvik (1986). "On the Fractional Calculus Model of Viscoelastic Behavior". In: *Journal of Rheology* 30.1, pp. 133–155. DOI: 10.1122/1.549887.
- Benzi, M. et al. (2020). "Non-local network dynamics via fractional graph Laplacians". In: *J. Complex Netw.* 8.3, cnaa017, 29. ISSN: 2051-1310. DOI: 10.1093/comnet/cnaa017. URL: https://doi.org/10.1093/comnet/cnaa017.
- Cusimano, N. et al. (2015). "On the Order of the Fractional Laplacian in Determining the Spatio-Temporal Evolution of a Space-Fractional Model of Cardiac Electrophysiology". In: *PLoS ONE* 10.12. cited By 30. DOI: 10.1371/journal.pone.0143938. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84955438668&doi=10.1371%2fjournal.pone.0143938&partnerID=40&md5=0b18e4be5403a7316bb63f69a0eb5f80.
- Giusti, A., R. Garrappa, and G. Vachon (Oct. 2020). "On the Kuzmin model in fractional Newtonian gravity". In: *The European Physical Journal Plus* 135.10, p. 798. ISSN: 2190-5444. DOI: 10.1140/epjp/s13360-020-00831-9. URL: https://doi.org/10.1140/epjp/s13360-020-00831-9.

Bibliography II

- Laskin, N. (Nov. 2002). "Fractional Schrödinger equation". In: *Phys. Rev. E* 66 (5), p. 056108. DOI: 10.1103/PhysRevE.66.056108. URL: https://link.aps.org/doi/10.1103/PhysRevE.66.056108.
- Luchko, Y. (2013). "Fractional wave equation and damped waves". In: *Journal of Mathematical Physics* 54.3, p. 031505. DOI: 10.1063/1.4794076. eprint: https://doi.org/10.1063/1.4794076. URL: https://doi.org/10.1063/1.4794076.
- Müller, S. et al. (2011). "A nonlinear fractional viscoelastic material model for polymers". In: *Computational Materials Science* 50.10, pp. 2938–2949. ISSN: 0927-0256. DOI: https://doi.org/10.1016/j.commatsci.2011.05.011.
- Riascos, A. and J. Mateos (2014). "Fractional dynamics on networks: Emergence of anomalous diffusion and Lévy flights". In: *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 90.3. cited By 49. DOI: 10.1103/PhysRevE.90.032809. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84907266357&doi=10.1103%2fPhysRevE.90.032809&partnerID=40&md5=be06b3148ba7bc17a50f52854beb9fac.

An introduction to fractional calculus

Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

May, 2022

RL Fractional Integrals and Derivatives

Riemann–Liouville Fractional Integral

Let $\Re\alpha > 0$, and let $f \in \mathbb{L}^1([a,b])$. Then for $t \in [a,b]$ we define

$$I^{\alpha}_{[a,t]} f(t) = {}_aD_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} f(\tau)\,\mathrm{d}\tau,$$

$$I^{\alpha}_{[t,b]} f(t) = {}_aD_t^{-\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_t^b (\tau-t)^{\alpha-1} f(\tau)\,\mathrm{d}\tau.$$

Riemann–Liouville Fractional Derivative

Let $\Re\alpha > 0$, $m = \lceil \alpha \rceil$, and $f \in \mathbb{A}^m([a,b])$, Then for $t \in [a,b]$ we define

$${}_{RL}D^{\alpha}_{[a,t]} f(t) = \frac{1}{\Gamma(m-\alpha)} \frac{d^m}{dt^m} \int_a^t (t-\tau)^{m-\alpha-1} f(\tau)\,\mathrm{d}\tau,$$

$${}_{RL}D^{\alpha}_{[t,b]} f(t) = \frac{(-1)^m}{\Gamma(m-\alpha)} \frac{d^m}{dt^m} \int_t^b (\tau-t)^{m-\alpha-1} f(\tau)\,\mathrm{d}\tau.$$

RL Derivatives Properties

RL integrals have a semigroup property, $^{d}/_{dt}$ has it, so what about RL Derivatives?

Theorem

Assume that $\alpha_1, \alpha_2 \geq 0$. Moreover let $\phi \in \mathbb{L}^1([a,b])$, and $f = I_{[a,b]}^{\alpha_1+\alpha_2}\phi$. Then,

$$ {}_{RL}D_{[a,t]}^{\alpha_1}\,{}_{RL}D_{[a,t]}^{\alpha_2} f = {}_{RL}D_{[a,t]}^{\alpha_1+\alpha_2}. $$

Proof. We use the definition and the assumption on f,

$$ {}_{RL}D_{[a,t]}^{\alpha_1}\,{}_{RL}D_{[a,t]}^{\alpha_2} f = {}_{RL}D_{[a,t]}^{\alpha_1}\,{}_{RL}D_{[a,t]}^{\alpha_2} I_{[a,b]}^{\alpha_1+\alpha_2}\phi = \frac{d^{\lceil \alpha_1 \rceil}}{dt^{\lceil \alpha_1 \rceil}} I_{[a,b]}^{\lceil \alpha_1 \rceil - \alpha_1} \frac{d^{\lceil \alpha_2 \rceil}}{dt^{\lceil \alpha_2 \rceil}} I_{[a,b]}^{\lceil \alpha_2 \rceil - \alpha_2} I_{[a,b]}^{\alpha_1+\alpha_2}\phi $$

RL Derivatives Properties

RL integrals have a semigroup property, $^{d}/_{dt}$ has it, so what about RL Derivatives?

Theorem

Assume that $\alpha_1, \alpha_2 \geq 0$. Moreover let $\phi \in \mathbb{L}^1([a,b])$, and $f = I_{[a,b]}^{\alpha_1+\alpha_2}\phi$. Then,

$$ {}_{RL}D_{[a,t]}^{\alpha_1}\,{}_{RL}D_{[a,t]}^{\alpha_2} f = {}_{RL}D_{[a,t]}^{\alpha_1+\alpha_2}. $$

Proof. We use the definition and the assumption on f, then we use the *semigroup property* for integrals

$$ \begin{aligned} {}_{RL}D_{[a,t]}^{\alpha_1}\,{}_{RL}D_{[a,t]}^{\alpha_2} f =& {}_{RL}D_{[a,t]}^{\alpha_1}\,{}_{RL}D_{[a,t]}^{\alpha_2} I_{[a,b]}^{\alpha_1+\alpha_2}\phi = \frac{d^{\lceil\alpha_1\rceil}}{dt^{\lceil\alpha_1\rceil}} I_{[a,b]}^{\lceil\alpha_1\rceil-\alpha_1} \frac{d^{\lceil\alpha_2\rceil}}{dt^{\lceil\alpha_2\rceil}} I_{[a,b]}^{\lceil\alpha_2\rceil-\alpha_2} I_{[a,b]}^{\alpha_1+\alpha_2}\phi \\ =& \frac{d^{\lceil\alpha_1\rceil}}{dt^{\lceil\alpha_1\rceil}} I_{[a,b]}^{\lceil\alpha_1\rceil-\alpha_1} \frac{d^{\lceil\alpha_2\rceil}}{dt^{\lceil\alpha_2\rceil}} I_{[a,b]}^{\lceil\alpha_2\rceil+\alpha_1}\phi = \frac{d^{\lceil\alpha_1\rceil}}{dt^{\lceil\alpha_1\rceil}} I_{[a,b]}^{\lceil\alpha_1\rceil-\alpha_1} \frac{d^{\lceil\alpha_2\rceil}}{dt^{\lceil\alpha_2\rceil}} I_{[a,b]}^{\lceil\alpha_2\rceil} I_{[a,b]}^{\alpha_1}\phi \end{aligned} $$

RL Derivatives Properties

RL integrals have a semigroup property, d/dt has it, so what about RL Derivatives?

Theorem

Assume that $\alpha_1, \alpha_2 \geq 0$. Moreover let $\phi \in \mathbb{L}^1([a,b])$, and $f = I_{[a,b]}^{\alpha_1+\alpha_2}\phi$. Then,

$$ {}_{RL}D_{[a,t]}^{\alpha_1}\,{}_{RL}D_{[a,t]}^{\alpha_2} f = {}_{RL}D_{[a,t]}^{\alpha_1+\alpha_2}. $$

Proof. We use the definition and the assumption on f, then we use the *semigroup property* for integrals, and since orders of the integral and differential operators involved are in $\mathbb{N}$

$$ \begin{aligned} {}_{RL}D_{[a,t]}^{\alpha_1}\,{}_{RL}D_{[a,t]}^{\alpha_2} f =& \frac{d^{\lceil \alpha_1 \rceil}}{dt^{\lceil \alpha_1 \rceil}} I_{[a,b]}^{\lceil \alpha_1 \rceil - \alpha_1} \frac{d^{\lceil \alpha_2 \rceil}}{dt^{\lceil \alpha_2 \rceil}} I_{[a,b]}^{\lceil \alpha_2 \rceil} I_{[a,b]}^{\alpha_1}\phi = \frac{d^{\lceil \alpha_1 \rceil}}{dt^{\lceil \alpha_1 \rceil}} I_{[a,b]}^{\lceil \alpha_1 \rceil - \alpha_1} I_{[a,b]}^{\alpha_1}\phi = \frac{d^{\lceil \alpha_1 \rceil}}{dt^{\lceil \alpha_1 \rceil}} I_{[a,b]}^{\lceil \alpha_1 \rceil}\phi \\ =& \phi. \end{aligned} $$

RL Derivatives Properties

RL integrals have a semigroup property, $^{d}/_{dt}$ has it, so what about RL Derivatives?

Theorem

Assume that $\alpha_1, \alpha_2 \geq 0$. Moreover let $\phi \in \mathbb{L}^1([a,b])$, and $f = I_{[a,b]}^{\alpha_1+\alpha_2}\phi$. Then,

$$ {}_{RL}D_{[a,t]}^{\alpha_1}\,{}_{RL}D_{[a,t]}^{\alpha_2} f = {}_{RL}D_{[a,t]}^{\alpha_1+\alpha_2}. $$

Proof. We use the definition and the assumption on f, then we use the *semigroup property* for integrals, and since orders of the integral and differential operators involved are in $\mathbb{N}$. This way we proved that: ${}_{RL}D_{[a,t]}^{\alpha_1}\,{}_{RL}D_{[a,t]}^{\alpha_2} f = \phi$. Now we work on the other part, that is analogous:

$$ {}_{RL}D_{[a,t]}^{\alpha_1+\alpha_2} f = \frac{d^{\lceil \alpha_1+\alpha_2 \rceil}}{dt^{\lceil \alpha_1+\alpha_2 \rceil}} I_{[a,b]}^{\lceil \alpha_1+\alpha_2 \rceil-\alpha_1-\alpha_2} f = \frac{d^{\lceil \alpha_1+\alpha_2 \rceil}}{dt^{\lceil \alpha_1+\alpha_2 \rceil}} I_{[a,b]}^{\lceil \alpha_1+\alpha_2 \rceil} I_{[a,b]}^{-\alpha_1-\alpha_2} I_{[a,b]}^{\alpha_1+\alpha_2}\phi = \phi. $$

RL Derivatives Properties

Theorem

Assume that $\alpha_1, \alpha_2 \geq 0$. Moreover let $\phi \in \mathbb{L}^1([a,b])$, and $f = I_{[a,b]}^{\alpha_1+\alpha_2}\phi$. Then,

$$ {}_{RL}D_{[a,t]}^{\alpha_1}\, {}_{RL}D_{[a,t]}^{\alpha_2} f = {}_{RL}D_{[a,t]}^{\alpha_1+\alpha_2}. $$

An observation on the hypothesis

The crucial hypothesis for the proof has been having $f = I_{[a,b]}^{\alpha_1+\alpha_2}\phi$. This is **not technical**, consider $f(t) = \sqrt{t}$, and $\alpha_1 = \alpha_2 = 1/2$, then we have computed in the last lecture

$$ {}_{RL}D_{[0,t]}^{1/2}\sqrt{t} = 0, \Rightarrow {}_{RL}D_{[0,t]}^{1/2}\, {}_{RL}D_{[0,t]}^{1/2}\sqrt{t} = 0, $$

but ${}_{RL}D_{[0,t]}^{1} = \frac{d}{dt}\sqrt{t} = 1/2\sqrt{t} \neq 0$. The condition on f implies both the needed regularity, and regulates how $f(t) \to 0$ as $t \to a$. **Other example.** Consider the same function with $\alpha_1 = 1/2$, $\alpha_2 = 3/2$.

RL Derivatives Properties - II

Theorem

Let $\alpha \geq 0$. Then, for every $f \in \mathbb{L}^1([a,b])$

$$ {}_{RL}D^{\alpha}_{[a,t]} I^{\alpha}_{[a,t]} f = f \qquad \text{a.e.} $$

Proof. The case $\alpha = 0$ descend from the definitions, both operators are the identity. For $\alpha > 0$, let $m = \lceil \alpha \rceil$, then we use the definition of ${}_{RL}D^{\alpha}_{[a,t]}$ and the semigroup property of fractional integration

$$ {}_{RL}D^{\alpha}_{[a,t]} I^{\alpha}_{[a,t]} f = \frac{d^m}{dt^m} I^{m-\alpha}_{[a,t]} I^{\alpha}_{[a,t]} f = \frac{d^m}{dt^m} I^{m}_{[a,t]} f = f(t). $$

RL Derivatives Properties - II

Theorem

Let $\alpha \geq 0$. Then, for every $f \in \mathbb{L}^1([a,b])$

$$ {}_{RL}D^{\alpha}_{[a,t]} I^{\alpha}_{[a,t]} f = f \qquad \text{a.e.} $$

Thus we have proved that the RL derivative is a **left inverse** of the RL integral, unfortunately we cannot claim that it is the right inverse.

Theorem

Let $\alpha > 0$. If there exists some $\phi \in \mathbb{L}^1([a,b])$ such that $f = I^{\alpha}_{[a,t]}\phi$ then

$$ I^{\alpha}_{[a,t]} {}_{RL}D^{\alpha}_{[a,t]} f = f. $$

Proof. This is an immediate consequence of the left-inverse property, since

$$ I^{\alpha}_{[a,t]} {}_{RL}D^{\alpha}_{[a,t]} f = I^{\alpha}_{[a,t]} {}_{RL}D^{\alpha}_{[a,t]} I^{\alpha}_{[a,t]}\phi = I^{\alpha}_{[a,t]}\phi = f. $$

RL Derivatives Properties - II

Theorem

Let $\alpha \geq 0$. Then, for every $f \in \mathbb{L}^1([a,b])$

$$ {}_{RL}D^{\alpha}_{[a,t]} I^{\alpha}_{[a,t]} f = f \qquad \text{a.e.} $$

Thus we have proved that the RL derivative is a **left inverse** of the RL integral, unfortunately we cannot claim that it is the right inverse.

Theorem

Let $\alpha > 0$. If there exists some $\phi \in \mathbb{L}^1([a,b])$ such that $f = I^{\alpha}_{[a,t]}\phi$ then

$$ I^{\alpha}_{[a,t]} {}_{RL}D^{\alpha}_{[a,t]} f = f. $$

What happens in the general case?

RL Derivatives Properties - III

Theorem

Let $\alpha > 0$, and $m = \lfloor \alpha \rfloor + 1$. Assume that f is such that $I^{m-\alpha}_{[a,t]} f \in \mathbb{A}^m([a,b])$. Then,

$$I^{\alpha}_{[a,t]}\, {}_{RL}D^{\alpha}_{[a,t]} f = f(t) - \sum_{k=0}^{m-1} \frac{(t-a)^{\alpha-k-1}}{\Gamma(\alpha-k)} \lim_{z \to a^+} \frac{d^{m-k-1}}{dz} I^{m-\alpha}_{[a,z]} f(z).$$

That reduces to

$$I^{\alpha}_{[a,t]}\, {}_{RL}D^{\alpha}_{[a,t]} f = f(t) - \frac{(t-a)^{\alpha-1}}{\Gamma(\alpha)} \lim_{z \to a^+} I^{1-\alpha}_{[a,z]} f(z), \text{ for } 0 < \alpha < 1.$$

- As for the semigroup property this is an issue of regularity and of going rapidly enough to zero at the beginning of the interval,
- The analogous property can be written also for the *other-sided* RL derivatives.

RL - Combinations, products and compositions

Linear combination descend easily from the definition.

Theorem

Let $f_1, f_2 : [a, b] \to \mathbb{R}$ such that ${}_{RL}D^{\alpha}_{[a,t]} f_1$, and ${}_{RL}D^{\alpha}_{[a,t]} f_1$ exist almost everywhere. Then, for $c_1, c_2 \in \mathbb{R}$ we have ${}_{RL}D^{\alpha}_{[a,t]}(c_1 f_1 + c_2 f_2)$ exists almost everywhere, and

$$ {}_{RL}D^{\alpha}_{[a,t]}(c_1 f_1 + c_2 f_2) = c_1 \, {}_{RL}D^{\alpha}_{[a,t]} f_1 + c_2 \, {}_{RL}D^{\alpha}_{[a,t]} f_2. $$

RL - Combinations, products and compositions

Linear combination descend easily from the definition.

Theorem

Let $f_1, f_2 : [a, b] \to \mathbb{R}$ such that ${}_{RL}D^{\alpha}_{[a,t]} f_1$, and ${}_{RL}D^{\alpha}_{[a,t]} f_1$ exist almost everywhere. Then, for $c_1, c_2 \in \mathbb{R}$ we have ${}_{RL}D^{\alpha}_{[a,t]}(c_1 f_1 + c_2 f_2)$ exists almost everywhere, and

$$ {}_{RL}D^{\alpha}_{[a,t]}(c_1 f_1 + c_2 f_2) = c_1 {}_{RL}D^{\alpha}_{[a,t]} f_1 + c_2 {}_{RL}D^{\alpha}_{[a,t]} f_2. $$

Leibniz' formula for Riemann–Liouville operators, doesn't come so easily

Theorem (Leibniz' formula for Riemann–Liouville operators)

Let $\alpha > 0$, and assume f and g analytic on $(a - h, a + h)$ for some $h > 0$. Then,

$$ {}_{RL}D^{\alpha}_{[a,t]}[fg](t) = \sum_{k=0}^{\lfloor \alpha \rfloor} \binom{\alpha}{k} {}_{RL}D^{k}_{[a,t]} f(t) {}_{RL}D^{\alpha-k}_{[a,t]} g(t) + \sum_{k=\lfloor \alpha \rfloor + 1}^{+\infty} \binom{\alpha}{k} {}_{RL}D^{k}_{[a,t]} f(t) I^{k-\alpha}_{[a,t]} g(t), $$

for $t \in (a, a + h/2)$.

RL - Combinations, products and compositions - II

For compositions we need to recall first a result for integer-order derivatives

Francesco da Paola Virginio Secondo Maria Faà di Bruno's Lemma

If g and f are functions with a sufficient number of derivatives and $n \in \mathbb{N}$, then

$$\frac{d^n}{dt^n}[g(f(\cdot))](t) = \sum \left(\frac{d^k}{dt^k} g\right)(f(t)) \prod_{\mu=1}^{n} \left(\frac{d^\mu}{dt^\mu} f(t)\right)^{b_\mu},$$

where the sum is over all partitions of $\{1, 2, \ldots, n\}$, and for each partition k is its number of blocks and b_j is the number of blocks with exactly j elements.

For a proof (and the history) see (Johnson 2002).

RL - Combinations, products and compositions - II

For compositions we need to recall first a result for integer-order derivatives, then we can look at its extension

Faà di Bruno's formula for RL operators

If f and g are *regular enough* we have

$$
{}_{RL}D^{\alpha}_{[a,t]}[fg](t) = \sum_{k=1}^{+\infty} \binom{\alpha}{k} \frac{k!(t-a)^{k-\alpha}}{\Gamma(k-\alpha+1)} \sum_{\ell=1}^{k} \left({}_{RL}D^{\ell}_{[a,t]} f \right)(g(t))
$$

$$
\sum_{(a_1,\cdots,a_k)\in A_{k,\ell}} \prod_{r=1}^{k} \frac{1}{a_r!} \left(\frac{\frac{d^r}{dt^r} g(t)}{r} \right)^{a_r} + \frac{(t-a)^{-\alpha}}{\Gamma(1-\alpha)} f(g(t)),
$$

where $(a_1, \ldots, a_k) \in A_{k,\ell}$ means that

$$
a_1, \ldots, a_k \in \mathbb{N}_0, \sum_{r=1}^{k} r a_r = k \text{ and } \sum_{r=1}^{k} a_r = \ell.
$$

What now?

We have put together **all the analogues of the instruments of classical calculus**, but what do we do with them now?

What now?

We have put together **all the analogues of the instruments of classical calculus**, but what do we do with them now?

What we would like to solve is:

$$ {}_{RL}D^{\alpha}_{[0,t]}\mathbf{y}(t) = f(t, \mathbf{y}(t)), \qquad \mathbf{y} : [0, T] \to \mathbb{R}^d, \ f : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d. $$

Nevertheless, we have a problem! What we would like to solve is a Cauchy problem, so we need to put **initial conditions**, but last time we observed that

$$ {}_{RL}D^{\alpha}_{[0,t]}\mathbf{c} \neq 0, \qquad \mathbf{c} \in \mathbb{R}^d. $$

Therefore, we should equip the system with the **following initial conditions** instead

$$ {}_{RL}D^{\alpha-k}_{[0,t]}\mathbf{y}(0) = \mathbf{b}_k, \quad k = 1, 2, \ldots, \lceil \alpha \rceil - 1, \ \lim_{z \to 0^+} I^{\lceil \alpha \rceil - \alpha}_{[0,t]}\mathbf{y}(z) = b_{\lceil \alpha \rceil}. $$

What now?

We have put together **all the analogues of the instruments of classical calculus**, but what do we do with them now?

What we would like to solve is:

$$ {}_{RL}D^{\alpha}_{[0,t]}\mathbf{y}(t) = f(t, \mathbf{y}(t)), \qquad \mathbf{y} : [0, T] \to \mathbb{R}^d, \ f : [0, T] \times \mathbb{R}^d \to \mathbb{R}^d. $$

Nevertheless, we have a problem! What we would like to solve is a Cauchy problem, so we need to put **initial conditions**, but last time we observed that

$$ {}_{RL}D^{\alpha}_{[0,t]}\mathbf{c} \neq 0, \qquad \mathbf{c} \in \mathbb{R}^d. $$

Therefore, we should equip the system with the **following initial conditions** instead

$$ {}_{RL}D^{\alpha-k}_{[0,t]}\mathbf{y}(0) = \mathbf{b}_k, \quad k = 1, 2, \ldots, \lceil \alpha \rceil - 1, \ \lim_{z \to 0^+} I^{\lceil \alpha \rceil - \alpha}_{[0,t]}\mathbf{y}(z) = b_{\lceil \alpha \rceil}. $$

We could develop a theory for this, but **these conditions are physically difficult** to use, we don't get this type of initial data from the applications.

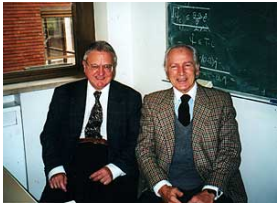
Caputo fractional derivatives

Caputo fractional derivative (Caputo 2008)

Let $\alpha \geq 0$, and $m = \lceil \alpha \rceil$. Then, we define the operator

$$ {}_C D^{\alpha}_{[a,t]} f = I^{m-\alpha}_{[a,t]} \frac{d^m}{dt^m} f, $$

whenever $\frac{d^m}{dt^m} f \in \mathbb{L}^1([a,b])$.



(R. Gorenflo, M. Caputo, Bologna 2000, source: fracalmo.org)

- We have exchanged the order of the derivative and fractional integral operators.

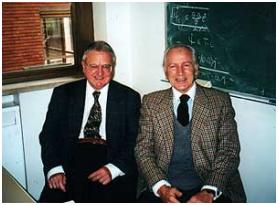
Caputo fractional derivatives

Caputo fractional derivative (Caputo 2008)

Let $\alpha \geq 0$, and $m = \lceil \alpha \rceil$. Then, we define the operator

$$ {}_{C}D^{\alpha}_{[a,t]} f = I^{m-\alpha}_{[a,t]} \frac{d^m}{dt^m} f, $$

whenever $\frac{d^m}{dt^m} f \in \mathbb{L}^1([a,b])$.



(R. Gorenflo, M. Caputo, Bologna 2000, source: fracalmo.org)

- We have exchanged the order of the derivative and fractional integral operators.

"Chi cerca trova, chi ricerca ritrova." - E. De Giorgi

The concept occurred a certain number of times: (Džrbašjan and Nersesjan 1968; Gerasimov 1948; Gross 1947; Liouville 1832; Rabotnov et al. 1969).

So, what is the difference?

First of all, we have the result we wanted on constants $c \in \mathbb{R}$:

$$ {}_{C}D^{\alpha}_{[a,t]}c = I^{m-\alpha}_{[a,t]}\frac{d^m}{dt^m}c = I^{m-\alpha}_{[a,t]}0 = 0. $$

So, what is the difference?

First of all, we have the result we wanted on constants $c \in \mathbb{R}$:

$$ {}_{C}D^{\alpha}_{[a,t]}c = I^{m-\alpha}_{[a,t]}\frac{d^m}{dt^m}c = I^{m-\alpha}_{[a,t]}0 = 0. $$

We can put in relation the two operators with the following result

Theorem

Let $\alpha > 0$ and $m = \lceil \alpha \rceil$. Moreover, assume that $f \in \mathbb{A}^m([a,b])$. Then,

$$ {}_{C}D^{\alpha}_{[a,t]}f = {}_{RL}D^{\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] \text{ a.e. on } [a,b], $$

for $T_{m-1}[f;a]$ the Taylor polynomial of degree $m-1$ for the function f centered at a, with $T_{-1}[f;a] = 0$.

So, what is the difference?

Proof. In the case $\alpha \in \mathbb{N}$ the result follows easily, since both quantities reduces to the integer order αth derivative.

So, what is the difference?

Proof. In the case $\alpha \in \mathbb{N}$ the result follows easily, since both quantities reduces to the integer order αth derivative. Therefore, we consider the case $\alpha \notin \mathbb{N}$ and $m = \lceil \alpha \rceil > \alpha$

$$
\begin{aligned}
{}_{RL}D^{\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] =& \frac{d^m}{dt^m} I^{m-\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] \\
=& \frac{d^m}{dt^m} \int_a^t \frac{(t-\tau)^{m-\alpha-1}}{\Gamma(m-\alpha)} \left(f(\tau) - T_{m-1}[f;a](\tau)\right) \, \mathrm{d}\tau,
\end{aligned}
$$

So, what is the difference?

Proof. In the case $\alpha \in \mathbb{N}$ the result follows easily, since both quantities reduces to the integer order αth derivative. Therefore, we consider the case $\alpha \notin \mathbb{N}$ and $m = \lceil \alpha \rceil > \alpha$

$$
\begin{aligned}
{}_{RL}D^{\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] &= \frac{d^m}{dt^m} I^{m-\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] \\
&= \frac{d^m}{dt^m} \int_a^t \frac{(t-\tau)^{m-\alpha-1}}{\Gamma(m-\alpha)} \left(f(\tau) - T_{m-1}[f;a](\tau)\right) \mathrm{d}\tau,
\end{aligned}
$$

We apply a **partial integration**

$$
\begin{aligned}
* = &-\frac{1}{\Gamma(m-\alpha+1)} \left[(f(\tau) - T_{m-1}[f,a](\tau))(t-\tau)^{m-\alpha}\right]\Big|_{\tau=a}^{\tau=t} \\
&+ \frac{1}{\Gamma(m-\alpha+1)} \int_a^t (f'(\tau) - (T_{m-1}[f,a](\tau))')(t-\tau)^{m-\alpha} \mathrm{d}\tau.
\end{aligned}
$$

So, what is the difference?

Proof. In the case $\alpha \in \mathbb{N}$ the result follows easily, since both quantities reduces to the integer order αth derivative. Therefore, we consider the case $\alpha \notin \mathbb{N}$ and $m = \lceil \alpha \rceil > \alpha$

$$
\begin{aligned}
{}_{RL}D^{\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] &= \frac{d^m}{dt^m} I^{m-\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] \\
&= \frac{d^m}{dt^m} \int_a^t \frac{(t-\tau)^{m-\alpha-1}}{\Gamma(m-\alpha)} \left(f(\tau) - T_{m-1}[f;a](\tau)\right) \mathrm{d}\tau,
\end{aligned}
$$

We apply a **partial integration**

$$
\begin{aligned}
* = &-\frac{1}{\Gamma(m-\alpha+1)} \left[(f(\tau) - T_{m-1}[f,a](\tau))(t-\tau)^{m-\alpha}\right]\Big|_{\tau=a}^{\tau=t} \\
&+ \frac{1}{\Gamma(m-\alpha+1)} \int_a^t (f'(\tau) - (T_{m-1}[f,a](\tau))')(t-\tau)^{m-\alpha} \mathrm{d}\tau.
\end{aligned}
$$

The terms in red are zero, and only the integral terms remain.

So, what is the difference?

Proof. In the case $\alpha \in \mathbb{N}$ the result follows easily, since both quantities reduces to the integer order αth derivative. Therefore, we consider the case $\alpha \notin \mathbb{N}$ and $m = \lceil \alpha \rceil > \alpha$

$$
\begin{aligned}
{}_{RL}D^{\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] =& \frac{d^m}{dt^m} I^{m-\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] \\
=& \frac{d^m}{dt^m} \int_a^t \frac{(t-\tau)^{m-\alpha-1}}{\Gamma(m-\alpha)} \left(f(\tau) - T_{m-1}[f;a](\tau)\right) \mathrm{d}\tau,
\end{aligned}
$$

We apply a **partial integration** m times since $f \in \mathbb{A}^m([a,b])$:

$$
I^{m-\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] = I^{2m-\alpha}_{[a,t]} \frac{d^m}{dt^m}\left[f - T_{m-1}[f;a]\right] = I^{m}_{[a,t]} I^{m-\alpha}_{[a,t]} \frac{d^m}{dt^m}\left[f - T_{m-1}[f;a]\right],
$$

So, what is the difference?

Proof. In the case $\alpha \in \mathbb{N}$ the result follows easily, since both quantities reduces to the integer order αth derivative. Therefore, we consider the case $\alpha \notin \mathbb{N}$ and $m = \lceil \alpha \rceil > \alpha$

$$
\begin{aligned}
{}_{RL}D^{\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] &= \frac{d^m}{dt^m} I^{m-\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] \\
&= \frac{d^m}{dt^m} \int_a^t \frac{(t-\tau)^{m-\alpha-1}}{\Gamma(m-\alpha)} \left(f(\tau) - T_{m-1}[f;a](\tau)\right) \mathrm{d}\tau,
\end{aligned}
$$

We apply a **partial integration** m times since $f \in \mathbb{A}^m([a,b])$:

$$
I^{m-\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] = I^{2m-\alpha}_{[a,t]} \frac{d^m}{dt^m}\left[f - T_{m-1}[f;a]\right] = I^{m}_{[a,t]} I^{m-\alpha}_{[a,t]} \frac{d^m}{dt^m}\left[f - T_{m-1}[f;a]\right],
$$

the mth derivative of the Taylor polynomial is zero (degree $m-1$).

So, what is the difference?

Proof. In the case $\alpha \in \mathbb{N}$ the result follows easily, since both quantities reduces to the integer order αth derivative. Therefore, we consider the case $\alpha \notin \mathbb{N}$ and $m = \lceil \alpha \rceil > \alpha$

$$
\begin{aligned}
{}_{RL}D^{\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] =& \frac{d^m}{dt^m} I^{m-\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] \\
=& \frac{d^m}{dt^m} \int_a^t \frac{(t-\tau)^{m-\alpha-1}}{\Gamma(m-\alpha)} \left(f(\tau) - T_{m-1}[f;a](\tau)\right) \mathrm{d}\tau,
\end{aligned}
$$

We apply a **partial integration** m times since $f \in \mathbb{A}^m([a,b])$ and obtain the expression

$$
I^{m-\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] = I^{m}_{[a,t]} I^{m-\alpha}_{[a,t]} \frac{d^m}{dt^m} f.
$$

So, what is the difference?

Proof. In the case $\alpha \in \mathbb{N}$ the result follows easily, since both quantities reduces to the integer order αth derivative. Therefore, we consider the case $\alpha \notin \mathbb{N}$ and $m = \lceil \alpha \rceil > \alpha$

$$
\begin{aligned}
{}_{RL}D^{\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] &= \frac{d^m}{dt^m} I^{m-\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] \\
&= \frac{d^m}{dt^m} \int_a^t \frac{(t-\tau)^{m-\alpha-1}}{\Gamma(m-\alpha)} \left(f(\tau) - T_{m-1}[f;a](\tau)\right) \, \mathrm{d}\tau,
\end{aligned}
$$

We apply a **partial integration** m times since $f \in \mathbb{A}^m([a,b])$ and obtain the expression

$$
I^{m-\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] = I^{m}_{[a,t]} I^{m-\alpha}_{[a,t]} \frac{d^m}{dt^m} f.
$$

We reapply the mth derivative to the simplified expression:

$$
{}_{RL}D^{\alpha}_{[a,t]}\left[f - T_{m-1}[f;a]\right] = \frac{d^m}{dt^m} I^{m}_{[a,t]} I^{m-\alpha}_{[a,t]} \frac{d^m}{dt^m} f = I^{m-\alpha}_{[a,t]} \frac{d^m}{dt^m} f = {}_{C}D^{\alpha}_{[a,t]} f.
$$

An example of computation

Let $f(t) = (t-a)^\beta$ for some $\beta \geq 0$, then

$$
{}_{CA}D^{\alpha}_{[a,t]} f(t) = \begin{cases} 0, & \beta \in \{0, 1, 2, \dots, \lceil \alpha \rceil - 1\}, \\ \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\alpha)}(t-a)^{\beta-\alpha}, & (\beta \in \mathbb{N} \wedge \beta \geq \lceil \alpha \rceil) \\ & \vee (\beta \notin \mathbb{N} \wedge \beta > \lceil \alpha \rceil - 1). \end{cases}
$$

Let us compare it with the Riemann-Liouville case:

$$
{}_{RL}D^{\alpha}_{[0,1]} f(t) = \begin{cases} 0, & \alpha - \beta \in \mathbb{N}, \\ \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\alpha)}(t-a)^{\beta-\alpha}, & \alpha - \beta \notin \mathbb{N}. \end{cases}
$$

! The two operators have different kernels and domain.

Caputo fractional derivatives - Properties

We can rewrite all the properties we have seen for RL derivatives for the Caputo version.

Theorem. (Caputo Derivatives Properties)

Let $\alpha \geq 0$ and $m = \lceil \alpha \rceil$

(i) ${}_C D^{\alpha}_{[a,t]} f = {}_{RL} D^{\alpha}_{[a,t]} f - \sum_{k=0}^{m-1} f^{(k)}(a)/\Gamma(k-\alpha+1)(t-a)^{k-\alpha}$,

(ii) ${}_C D^{\alpha}_{[a,t]} f = {}_{RL} D^{\alpha}_{[a,t]} f$ iff f has a zero of order m at a,

(iii) If f is continuous, ${}_C D^{\alpha}_{[a,t]} I^{\alpha}_{[a,t]} f = f$,

(iv) If $f \in \mathbb{A}^m([a,b])$ then $I^{\alpha}_{[a,t]} {}_C D^{\alpha}_{[a,t]} = f(t) - \sum_{k=0}^{m-1} f^{(k)}(a)/k!(x-a)^k$,

(v) If $f \in \mathcal{C}^k([a,b])$, $\alpha, \beta > 0$ s.t. $\exists \ell \in \mathbb{N}$ $\ell \leq k$ and $\alpha, \alpha+\beta \in [\ell-1, \ell]$ then ${}_C D^{\alpha}_{[a,t]} {}_C D^{\beta}_{[a,t]} f = {}_C D^{\alpha+\beta}_{[a,t]} f$.

(vi) $f \in \mathcal{C}^{\mu}([a,b])$, $\alpha \in [0, \mu]$, then ${}_{RL} D^{\mu-\alpha}_{[a,t]} {}_C D^{\alpha}_{[a,t]} f = f^{(\mu)}$.

Caputo fractional derivatives - Properties

Theorem. (Caputo Derivatives Properties)

(vii) For $f_1, f_2 : [a,b] \to \mathbb{R}$, $c_1, c_2 \in \mathbb{R}$ then

$$ {}_C D^{\alpha}_{[a,t]}(c_1 f_1 + c_2 f_2) = c_1 \, {}_C D^{\alpha}_{[a,t]} f_1 + c_2 \, {}_C D^{\alpha}_{[a,t]} f_2 \text{ a.e. on } [a,b], $$

if ${}_C D^{\alpha}_{[a,t]} f_1$, ${}_C D^{\alpha}_{[a,t]} f_2$ exist a.e. on $[a,b]$,

(viii) (Leibniz) let $\alpha \in (0,1)$, f, g analytic on $(a-h, a+h)$, then

$$ {}_C D^{\alpha}_{[a,t]}[fg](t) = \frac{(t-a)^{-\alpha}}{\Gamma(1-\alpha)} g(a)(f(t) - f(a)) + \left({}_C D^{\alpha}_{[a,t]}(g(t)) \right) f(t) + \sum_{k=1}^{\infty} \binom{\alpha}{k} \left(I^{k-\alpha}_{[a,t]} g(t) \right) {}_C D^{k}_{[a,t]} f(t). $$

They can all be proved by mimicking the proofs for the RL derivative.

Fractional ODEs with Caputo Derivatives

Let's restart with the differential equation, but now written in terms of Caputo Derivatives

$$\alpha > 0, \quad m = \lceil \alpha \rceil, \qquad \begin{cases} {}_C D^{\alpha}_{[0,t]} \mathbf{y}(t) = f(t, \mathbf{y}(t)), & t \in [0, T], \\ \dfrac{d^k \mathbf{y}(0)}{dt^k} = \mathbf{y}_0^{(k)}, & k = 0, 1, \ldots, m-1. \end{cases} \tag{FODE}$$

And we are now faced with the usual questions

- Is there any solution?
- If there is at least one, then how many there are?
- When it is all said and proved, how can we approximate it?

Fractional ODEs with Caputo Derivatives

Let's restart with the differential equation, but now written in terms of Caputo Derivatives

$$\alpha > 0, \quad m = \lceil \alpha \rceil, \qquad \begin{cases} {}_C D^{\alpha}_{[0,t]} \mathbf{y}(t) = f(t, \mathbf{y}(t)), & t \in [0, T], \\ \dfrac{d^k \mathbf{y}(0)}{dt^k} = \mathbf{y}_0^{(k)}, & k = 0, 1, \ldots, m-1. \end{cases} \tag{FODE}$$

And we are now faced with the usual questions

- Is there any solution? $\longrightarrow$ This lecture
- If there is at least one, then how many there are? $\longrightarrow$ This lecture
- When it is all said and proved, how can we approximate it? $\rightsquigarrow$ The next one

A Peano existence theorem for first order equations

Theorem (Diethelm and Ford 2002, Theorem 2.1, 2.2)

Let $0 < \alpha$ and $m = \lceil \alpha \rceil$. Moreover let $\{y_0^{(k)} \in \mathbb{R}\}_{k=0}^{m-1}$, $K > 0$, and $h^* > 0$. We define

$$G = \left\{ (t, y) \,:\, t \in [0, h^*] \,:\, \left| y - \sum_{k=0}^{m-1} t^k y_0^{(k)}/k! \right| \leq K \right\},$$

and let the function $f : G \to \mathbb{R}$ be continuous. Furthermore, define

$$M = \sup_{(t,z)\in G} |f(t,z)|, \; h = \begin{cases} h^*, & \text{if } M = 0, \\ \min\{h^*, (K^{\Gamma(\alpha+1)}/M)^{1/n}\}, & \text{else.} \end{cases}$$

Then, there exists a function $y \in \mathcal{C}([0, h])$ solving (FODE).

A Peano existence theorem for first order equations

Theorem (Diethelm and Ford 2002, Theorem 2.1, 2.2)

Let $0 < \alpha$ and $m = \lceil \alpha \rceil$. Moreover let $\{y_0^{(k)} \in \mathbb{R}\}_{k=0}^{m-1}$, $K > 0$, and $h^* > 0$. We define

$$G = \left\{ (t, y) \,:\, t \in [0, h^*] \,:\, \left| y - \sum_{k=0}^{m-1} t^k y_0^{(k)}/k! \right| \leq K \right\},$$

and let the function $f : G \to \mathbb{R}$ be continuous. Furthermore, define

$$M = \sup_{(t,z)\in G} |f(t,z)|, \; h = \begin{cases} h^*, & \text{if } M = 0, \\ \min\{h^*, (K^{\Gamma(\alpha+1)}/M)^{1/n}\}, & \text{else.} \end{cases}$$

Then, there exists a function $y \in \mathcal{C}([0, h])$ solving (FODE).

To prove it we need a Lemma...and a bit of work.

A Peano existence theorem for first order equations

Lemma

Under the same hypotheses of the previous Theorem. A function $y \in \mathcal{C}([0,h])$ is a solution of the initial value problem (FODE) *if and only if* it is a solution of the nonlinear Volterra integral equation of the second kind

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Proof. We need to prove both the implications.

A Peano existence theorem for first order equations

Lemma

Under the same hypotheses of the previous Theorem. A function $y \in \mathcal{C}([0,h])$ is a solution of the initial value problem (FODE) *if and only if* it is a solution of the nonlinear Volterra integral equation of the second kind

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Proof. We need to prove both the implications. $(\Rightarrow)$ First of all we have $y(t)$ being a continuous solution of the nonlinear Volterra equation. We apply on both side the Caputo derivative of order α

$$ {}_C D^{\alpha}_{[0,t]} y(t) = \underbrace{{}_C D^{\alpha}_{[0,t]} \left[\sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} \right]}_{=0 \ \lceil \alpha \rceil > m-1} + {}_C D^{\alpha}_{[0,t]} \left[\int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau \right],$$

A Peano existence theorem for first order equations

Lemma

Under the same hypotheses of the previous Theorem. A function $y \in \mathcal{C}([0,h])$ is a solution of the initial value problem (FODE) *if and only if* it is a solution of the nonlinear Volterra integral equation of the second kind

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Proof. We need to prove both the implications.$(\Rightarrow)$ First of all we have $y(t)$ being a continuous solution of the nonlinear Volterra equation. We apply on both side the Caputo derivative of order α

$${}_C D^{\alpha}_{[0,t]} y(t) = {}_C D^{\alpha}_{[0,t]} I^{\alpha}_{[0,t]} f(t, y(t)) = f(t, y(t)),\ f \text{ is continuous.}$$

A Peano existence theorem for first order equations

Lemma

Under the same hypotheses of the previous Theorem. A function $y \in \mathcal{C}([0,h])$ is a solution of the initial value problem (FODE) *if and only if* it is a solution of the nonlinear Volterra integral equation of the second kind

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Proof. We need to prove both the implications. $(\Rightarrow)$ follows by direct computation. $(\Leftarrow)$ Is a bit more laborious. Let us define $z(t) = f(t, y(t)) \in \mathcal{C}[0,h]$, we can rewrite (FODE) as:

$$z(t) = f(t, y(t)) = {}_C D^{\alpha}_{[0,t]} y(t) = {}_{RL} D^{\alpha}_{[0,t]} (y - T_{m-1}[y;0](t)) = \frac{d^m}{dt^m} I_0^{m-\alpha} (y - T_{m-1}[y;0])(t),$$

A Peano existence theorem for first order equations

Lemma

Under the same hypotheses of the previous Theorem. A function $y \in \mathcal{C}([0,h])$ is a solution of the initial value problem (FODE) *if and only if* it is a solution of the nonlinear Volterra integral equation of the second kind

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Proof. We need to prove both the implications. $(\Rightarrow)$ follows by direct computation. $(\Leftarrow)$ Is a bit more laborious. Let us define $z(t) = f(t, y(t)) \in \mathcal{C}[0,h]$, we can rewrite (FODE) as: we have an equality between continuous function, so we can apply $I^m_{[0,t]}$ to both sides!

A Peano existence theorem for first order equations

Lemma

Under the same hypotheses of the previous Theorem. A function $y \in \mathcal{C}([0,h])$ is a solution of the initial value problem (FODE) *if and only if* it is a solution of the nonlinear Volterra integral equation of the second kind

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Proof. We need to prove both the implications. $(\Rightarrow)$ follows by direct computation. $(\Leftarrow)$ Is a bit more laborious. Let us define $z(t) = f(t, y(t)) \in \mathcal{C}[0,h]$, we can rewrite (FODE) as:

$$I^m_{[0,t]} z(t) = I^m_{[0,t]} \frac{d^m}{dt^m} I_0^{m-\alpha}(y - T_{m-1}[y;0])(t) = I_0^{m-\alpha}(y - T_{m-1}[y;0])(t) + q(t),$$

for a polynomial $q(t) \in \mathbb{P}_{\leq m-1}[t]$.

A Peano existence theorem for first order equations

Lemma

Under the same hypotheses of the previous Theorem. A function $y \in \mathcal{C}([0,h])$ is a solution of the initial value problem (FODE) *if and only if* it is a solution of the nonlinear Volterra integral equation of the second kind

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Proof. We need to prove both the implications. $(\Rightarrow)$ follows by direct computation. $(\Leftarrow)$ Is a bit more laborious. Let us define $z(t) = f(t, y(t)) \in \mathcal{C}[0,h]$, we can rewrite (FODE) as:

$$I^m_{[0,t]} z(t) = I^{m-\alpha}_0 (y - T_{m-1}[y;0])(t) + q(t) \Rightarrow$$

- $I^m_{[0,t]} z(t) = 0$ (at least) $m$times for $t = 0$,
- $y - T_{m-1}[y;0] = 0$ (at least) $m$times for $t = 0$,
- $I^{m-\alpha}_0 (y - T_{m-1}[y;0])(t) = 0$ (at least) $m$times for $t = 0$,

A Peano existence theorem for first order equations

Lemma

Under the same hypotheses of the previous Theorem. A function $y \in \mathcal{C}([0,h])$ is a solution of the initial value problem (FODE) *if and only if* it is a solution of the nonlinear Volterra integral equation of the second kind

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Proof. We need to prove both the implications. $(\Rightarrow)$ follows by direct computation. $(\Leftarrow)$ Is a bit more laborious. Let us define $z(t) = f(t, y(t)) \in \mathcal{C}[0,h]$, we can rewrite (FODE) as:

$$I^m_{[0,t]} z(t) = I_0^{m-\alpha}(y - T_{m-1}[y;0])(t) + q(t)$$

Therefore $q(t) = 0$ (at least) $m$times for $t = 0$, but $\deg(q) \leq m-1 \Rightarrow q \equiv 0$.

A Peano existence theorem for first order equations

Lemma

Under the same hypotheses of the previous Theorem. A function $y \in \mathcal{C}([0,h])$ is a solution of the initial value problem (FODE) *if and only if* it is a solution of the nonlinear Volterra integral equation of the second kind

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Proof. We need to prove both the implications. $(\Rightarrow)$ follows by direct computation. $(\Leftarrow)$ Is a bit more laborious. Let us define $z(t) = f(t, y(t)) \in \mathcal{C}[0,h]$, we can rewrite (FODE) as:

$$I_{[0,t]}^m z(t) = I_0^{m-\alpha}(y - T_{m-1}[y;0])(t)$$

and apply ${}_{RL}D_{[0,t]}^{m-\alpha}$ to both side of the equation.

A Peano existence theorem for first order equations

Lemma

Under the same hypotheses of the previous Theorem. A function $y \in \mathcal{C}([0,h])$ is a solution of the initial value problem (FODE) *if and only if* it is a solution of the nonlinear Volterra integral equation of the second kind

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Proof. We need to prove both the implications. $(\Rightarrow)$ follows by direct computation. $(\Leftarrow)$ Is a bit more laborious. Let us define $z(t) = f(t, y(t)) \in \mathcal{C}[0,h]$, we can rewrite (FODE) as:

$${}_{RL}D_{[0,t]}^{m-\alpha} I_0^{m-\alpha} (y - T_{m-1}[y;0])(t) = {}_{RL}D_{[0,t]}^{m-\alpha} I_{[0,t]}^{m} z(t)$$

A Peano existence theorem for first order equations

Lemma

Under the same hypotheses of the previous Theorem. A function $y \in \mathcal{C}([0,h])$ is a solution of the initial value problem (FODE) *if and only if* it is a solution of the nonlinear Volterra integral equation of the second kind

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Proof. We need to prove both the implications. $(\Rightarrow)$ follows by direct computation. $(\Leftarrow)$ Is a bit more laborious. Let us define $z(t) = f(t, y(t)) \in \mathcal{C}[0,h]$, we can rewrite (FODE) as:

$$(y - T_{m-1}[y;0])(t) = {}_{RL}D_{[0,t]}^{m-\alpha} I_{[0,t]}^{m} z(t) = \frac{d}{dt} I_{[0,t]}^{1+\alpha-m} I_{[0,t]}^{m} z(t) = \frac{d}{dt} I_{[0,t]}^{1+\alpha} z(t) = I_0^{\alpha} z(t).$$

A Peano existence theorem for first order equations

Lemma

Under the same hypotheses of the previous Theorem. A function $y \in \mathcal{C}([0,h])$ is a solution of the initial value problem (FODE) *if and only if* it is a solution of the nonlinear Volterra integral equation of the second kind

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Proof. We need to prove both the implications. $(\Rightarrow)$ follows by direct computation. $(\Leftarrow)$ Is a bit more laborious. Let us define $z(t) = f(t, y(t)) \in \mathcal{C}[0,h]$, we can rewrite (FODE) as:

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau.$$

by recalling the definitions of $T_{m-1}[y, 0]$ and the RL-integral.

A Peano existence theorem for first order equations

The other two results we will need (and that we are not going to prove) are

Theorem (Ascoli-Arzelà)

Lef $F \subset \mathcal{C}([a,b])$ for some $a < b$, and assume the sets to be equipped with the supremum norm. Then F is *relatively compact*[1] in $\mathcal{C}([a,b])$ if F is

- *uniformly bounded*, $\exists\, C > 0$ s.t. $\|f\|_\infty \leq C \;\forall\, f \in F$,
- *equicontinuous* $\forall\, \varepsilon > 0\; \exists\, \delta > 0$ such that $\forall\, f \in F$ and all $x, x* \in [a,b]$ with $|x - x^*| < \delta$ we have $|f(x) - f(x^*)| < \varepsilon$.

Schauder's Fixed Point Theorem

Lef (E,d) be a complete metric space, let U be a closed convex subset of E, and let $A : U \to U$ be a mapping such that the set $\{Au \,:\, u \in U\}$ is *relatively compact*[1] in E. Then A has at least one fixed point.

[1] A subset whose closure is compact.

A Peano existence theorem for first order equations

Let us look again at the statement of the Theorem.

Theorem (Diethelm and Ford 2002, Theorem 2.1, 2.2)

Let $0 < \alpha$ and $m = \lceil \alpha \rceil$. Moreover let $\{y_0^{(k)} \in \mathbb{R}\}_{k=0}^{m-1}$, $K > 0$, and $h^* > 0$. We define

$$G = \left\{ (t, y) \,:\, t \in [0, h^*] \,:\, \left| y - \sum_{k=0}^{m-1} t^k y_0^{(k)}/k! \right| \leq K \right\},$$

and let the function $f : G \to \mathbb{R}$ be continuous. Furthermore, define

$$M = \sup_{(t,z)\in G} |f(t,z)|, \;\; h = \begin{cases} h^*, & \text{if } M = 0, \\ \min\{h^*, (K^{\Gamma(\alpha+1)}/M^{1/n}\}, & \text{else.} \end{cases}$$

Then, there exists a function $y \in \mathcal{C}([0, h])$ solving (FODE).

A Peano existence theorem for first order equations

Proof. If $M = 0$, then $f(x, y) = 0$ for all $(x, y) \in G$, then we can explicitly write the solution as

$$y : [0, h] \to \mathbb{R} \qquad y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)},$$

therefore a solution exists.

A Peano existence theorem for first order equations

Proof. If $M > 0$, let us apply the **Lemma** and rewrite our problem as a Volterra equation:

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil,$$

and introduce the polynomial T satisfying the boundary condition and the space U

$$T(t) = \sum_{k=0}^{m-1} \frac{x^k}{k!} y_0^{(k)}, \quad U = \{y \in \mathcal{C}([0,h]) \,:\, \|y - T\|_\infty \leq K\}.$$

A Peano existence theorem for first order equations

Proof. If $M > 0$, let us apply the **Lemma** and rewrite our problem as a Volterra equation:

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil,$$

and introduce the polynomial T satisfying the boundary condition and the space U

$$T(t) = \sum_{k=0}^{m-1} \frac{x^k}{k!} y_0^{(k)}, \quad U = \{y \in \mathcal{C}([0,h]) \,:\, \|y - T\|_\infty \leq K\}.$$

- U is closed and convex,
- $U \subset \mathcal{C}([0,h])$,

$\Rightarrow$ U is a non empty Banach space (at least $T \in U$).

A Peano existence theorem for first order equations

Proof. If $M > 0$, let us apply the **Lemma** and rewrite our problem as a Volterra equation:

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil,$$

and introduce the polynomial T satisfying the boundary condition and the space U

$$T(t) = \sum_{k=0}^{m-1} \frac{x^k}{k!} y_0^{(k)}, \quad U = \{y \in \mathcal{C}([0,h]) \,:\, \|y - T\|_\infty \leq K\}.$$

Let us define the operator:

$$(Ay)(t) = T(t) + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau.$$

A Peano existence theorem for first order equations

Proof. If $M > 0$, let us apply the **Lemma** and rewrite our problem as a Volterra equation:

$$y = Ay, \quad (Ay)(t) = T(t) + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau.$$

we have to prove that A has a fixed point by the following steps:

1. proving that $Ay \in U$,
2. showing that $A(U) = \{Au : u \in U\}$ is *relatively compact* (Ascoli-Arzelà),
3. apply Schauder's Fixed Point Theorem for the victory .

A Peano existence theorem for first order equations

Proof. *Step 1.* Let us take $0 \leq t_1 \leq t_2 \leq h$

$$
\begin{aligned}
|(Ay)(t_1) - (Ay)(t_2)| =& \frac{1}{\Gamma(\alpha)} \left| \int_0^{t_1} (t_1 - \tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau - \int_0^{t_2} (t_2 - \tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau \right| \\
=& \frac{1}{\Gamma(\alpha)} \left| \int_0^{t_1} \left[(t_1 - \tau)^{\alpha-1} - (t_2 - \tau)^{\alpha-1} \right] f(\tau, y(\tau))\, \mathrm{d}\tau \right. \\
& \left. - \int_{t_1}^{t_2} (t_2 - \tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau \right| \\
\leq& \frac{M}{\Gamma(\alpha)} \left(\int_0^{t_1} \left| (t_1 - \tau)^{\alpha-1} - (t_2 - \tau)^{\alpha-1} \right| \mathrm{d}\tau + \int_{t_1}^{t_2} (t_2 - \tau)^{\alpha-1}\, \mathrm{d}\tau \right)
\end{aligned}
$$

A Peano existence theorem for first order equations

Proof. *Step 1.* Let us take $0 \leq t_1 \leq t_2 \leq h$

$$
\begin{aligned}
|(Ay)(t_1) - (Ay)(t_2)| =& \frac{1}{\Gamma(\alpha)} \left| \int_0^{t_1} (t_1 - \tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau - \int_0^{t_2} (t_2 - \tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau \right| \\
=& \frac{1}{\Gamma(\alpha)} \left| \int_0^{t_1} \left[(t_1 - \tau)^{\alpha-1} - (t_2 - \tau)^{\alpha-1} \right] f(\tau, y(\tau))\, \mathrm{d}\tau \right. \\
& \left. - \int_{t_1}^{t_2} (t_2 - \tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau \right| \\
\leq& \frac{M}{\Gamma(\alpha)} \left(\int_0^{t_1} \left| (t_1 - \tau)^{\alpha-1} - (t_2 - \tau)^{\alpha-1} \right| \mathrm{d}\tau + \frac{(t_2 - t_1)^{\alpha}}{\alpha} \right)
\end{aligned}
$$

A Peano existence theorem for first order equations

Proof. *Step 1.* Let us take $0 \le t_1 \le t_2 \le h$

$$|(Ay)(t_1) - (Ay)(t_2)| \le \frac{M}{\Gamma(\alpha)} \left(\int_0^{t_1} \left| (t_1 - \tau)^{\alpha-1} - (t_2 - \tau)^{\alpha-1} \right| \, \mathrm{d}\tau + \frac{(t_2 - t_1)^\alpha}{\alpha} \right).$$

If $\alpha = 1$ the first integral vanishes.
If $\alpha < 1$, $\alpha - 1 < 0$, and hence $(t_1 - \tau)^{\alpha-1} \ge (t_2 - \tau)^{\alpha-1}$, thus we remove the $|\cdot|$ and

$$\int_0^{t_1} \left| (t_1 - \tau)^{\alpha-1} - (t_2 - \tau)^{\alpha-1} \right| = \frac{1}{\alpha}(t_1^\alpha - t_2^\alpha + (t_2 - t_1)^\alpha) \le \frac{1}{\alpha}(t_2 - t_1)^\alpha.$$

If $\alpha > 1$ we have $(t_1 - \tau)^{\alpha-1} \le (t_2 - \tau)^{\alpha-1}$

$$\int_0^{t_1} \left| (t_1 - \tau)^{\alpha-1} - (t_2 - \tau)^{\alpha-1} \right| = \frac{1}{\alpha}(t_2^\alpha - t_1^\alpha - (t_2 - t_1)^\alpha) \le \frac{1}{\alpha}(t_2^\alpha - t_1^\alpha).$$

A Peano existence theorem for first order equations

Proof. *Step 1.* Let us take $0 \leq t_1 \leq t_2 \leq h$

$$|(Ay)(t_1) - (Ay)(t_2)| \leq \begin{cases} 2M/\Gamma(\alpha+1)(t_2 - t_1)^\alpha, & \alpha \leq 1, \\ M/\Gamma(\alpha+1)((t_2 - t_1)^\alpha + t_2^\alpha - t_1^\alpha), & \alpha > 1. \end{cases}$$

Therefore,

A Peano existence theorem for first order equations

Proof. *Step 1.* Let us take $0 \leq t_1 \leq t_2 \leq h$

$$|(Ay)(t_1) - (Ay)(t_2)| \leq \begin{cases} 2M/\Gamma(\alpha+1)(t_2 - t_1)^\alpha, & \alpha \leq 1, \\ M/\Gamma(\alpha+1)((t_2 - t_1)^\alpha + t_2^\alpha - t_1^\alpha), & \alpha > 1. \end{cases}$$

Therefore,

- Ay is continuous since $|(Ay)(t_1) - (Ay)(t_2)| \to 0$ for $t_2 \to t_1$,

A Peano existence theorem for first order equations

Proof. *Step 1.* Let us take $0 \le t_1 \le t_2 \le h$

$$|(Ay)(t_1) - (Ay)(t_2)| \le \begin{cases} {}^{2M}\!/_{\Gamma(\alpha+1)}(t_2 - t_1)^\alpha, & \alpha \le 1, \\ {}^{M}\!/_{\Gamma(\alpha+1)}((t_2 - t_1)^\alpha + t_2^\alpha - t_1^\alpha), & \alpha > 1. \end{cases}$$

Therefore,

- Ay is continuous since $|(Ay)(t_1) - (Ay)(t_2)| \to 0$ for $t_2 \to t_1$,
- for $y \in U$ and $t \in [0, h]$ we find

$$|(Ay)(t) - T(t)| = \frac{1}{\Gamma(\alpha)} \left| \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau)) \right| \le \frac{1}{\Gamma(\alpha+1)} M t^\alpha \le \frac{1}{\Gamma(\alpha+1)} M h^\alpha$$

$$\left(\text{Hp: } h < K \frac{\Gamma(\alpha+1)}{M}^{1/n}\right) \le \frac{1}{\Gamma(\alpha+1)} M \frac{K\Gamma(\alpha+1)}{M} = K.$$

A Peano existence theorem for first order equations

Proof. *Step 1.* Let us take $0 \leq t_1 \leq t_2 \leq h$

$$|(Ay)(t_1) - (Ay)(t_2)| \leq \begin{cases} 2M/\Gamma(\alpha+1)(t_2 - t_1)^\alpha, & \alpha \leq 1, \\ M/\Gamma(\alpha+1)((t_2 - t_1)^\alpha + t_2^\alpha - t_1^\alpha), & \alpha > 1. \end{cases}$$

Therefore,

- Ay is continuous since $|(Ay)(t_1) - (Ay)(t_2)| \to 0$ for $t_2 \to t_1$,
- for $y \in U$ and $t \in [0, h]$ we find $|(Ay)(t) - T(t)| \leq K$

$\Rightarrow$ $Ay \in U$ if $y \in U$.

A Peano existence theorem for first order equations

Proof. Our plan:

✔ proving that $Ay \in U$,

2. showing that $A(U) = \{Au : u \in U\}$ is *relatively compact* (Ascoli-Arzelà),
3. apply Schauder's Fixed Point Theorem for the victory ✋.

Step 2. First we prove that the set is *bounded*, let $z \in A(U)$ and $t \in [0, h]$

$$
\begin{aligned}
|z(t)| = |(Ay)(t)| &\leq \| T \|_\infty + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} |f(\tau, y(\tau))| \, \mathrm{d}\tau \\
&\leq \| T \|_\infty + \frac{1}{\Gamma(\alpha+1)} M h^\alpha \leq \| T \|_\infty + K.
\end{aligned}
$$

A Peano existence theorem for first order equations

Proof. Our plan:

✔ proving that $Ay \in U$,

2. showing that $A(U) = \{Au : u \in U\}$ is *relatively compact* (Ascoli-Arzelà),
3. apply Schauder's Fixed Point Theorem for the victory 👆.

Step 2. First we prove that the set is *bounded*, let $z \in A(U)$ and $t \in [0, h]$

$$|z(t)| \leq \|T\|_\infty + K.$$

For the *emicontinuity*, let $0 \leq t_1 \leq t_2 \leq h$ we found (for $\alpha \leq 1$)

$$|(Ay)(t_1) - (Ay)(t_2)| \leq \frac{2M}{\Gamma(\alpha+1)}(t_2 - t_1)^\alpha \leq \frac{2M}{\Gamma(\alpha+1)}\delta^\alpha, \quad \text{if } |t_2 - t_1| < \delta.$$

the expression on the right is independent of y, t_1, and t_2.

A Peano existence theorem for first order equations

Proof. Our plan:

✔ proving that $Ay \in U$,

2. showing that $A(U) = \{Au : u \in U\}$ is *relatively compact* (Ascoli-Arzelà),
3. apply Schauder's Fixed Point Theorem for the victory 👆.

Step 2. First we prove that the set is *bounded*, let $z \in A(U)$ and $t \in [0, h]$

$$|z(t)| \leq \|T\|_\infty + K.$$

For the *emicontinuity*, let $0 \leq t_1 \leq t_2 \leq h$ we found (for $\alpha > 1$)

$$|(Ay)(t_1) - (Ay)(t_2)| \leq \frac{M}{\Gamma(\alpha+1)}((t_2 - t_1)^\alpha + t_2^\alpha - t_1^\alpha),$$

$$\text{(Mean Value Theorem)} \quad = \frac{M}{\Gamma(\alpha+1)}((t_2 - t_1)^\alpha + \alpha(t_2 - t_1)\tau^{\alpha-1}), \quad \tau \in [t_1, t_2] \subseteq [0, h]$$

A Peano existence theorem for first order equations

Proof. Our plan:

✔ proving that $Ay \in U$,

2. showing that $A(U) = \{Au : u \in U\}$ is *relatively compact* (Ascoli-Arzelà),
3. apply Schauder's Fixed Point Theorem for the victory ✌.

Step 2. First we prove that the set is *bounded*, let $z \in A(U)$ and $t \in [0, h]$

$$|z(t)| \leq \|T\|_\infty + K.$$

For the *emicontinuity*, let $0 \leq t_1 \leq t_2 \leq h$ we found (for $\alpha > 1$)

$$\begin{aligned} |(Ay)(t_1) - (Ay)(t_2)| \leq & \frac{M}{\Gamma(\alpha+1)}((t_2 - t_1)^\alpha + t_2^\alpha - t_1^\alpha), \\ \leq & \frac{M}{\Gamma(\alpha+1)}((t_2 - t_1)^\alpha + \alpha(t_2 - t_1)h^{\alpha-1}) \end{aligned}$$

A Peano existence theorem for first order equations

Proof. Our plan:

✔ proving that $Ay \in U$,

2. showing that $A(U) = \{Au : u \in U\}$ is *relatively compact* (Ascoli-Arzelà),
3. apply Schauder's Fixed Point Theorem for the victory 👆.

Step 2. First we prove that the set is *bounded*, let $z \in A(U)$ and $t \in [0, h]$

$$|z(t)| \leq \|T\|_\infty + K.$$

For the *emicontinuity*, let $0 \leq t_1 \leq t_2 \leq h$ we found (for $\alpha > 1$)

$$|(Ay)(t_1) - (Ay)(t_2)| \leq \frac{M}{\Gamma(\alpha+1)}(\delta^\alpha + \alpha\delta h^\alpha - 1), \qquad \text{if } |t_2 - t_1| < \delta,$$

the expression on the right is again independent of y, t_1, and t_2.

A Peano existence theorem for first order equations

Proof. Our plan:

- ✔ proving that $Ay \in U$,
- ✔ showing that $A(U) = \{Au : u \in U\}$ is *relatively compact* (Ascoli-Arzelà),
- 3. apply Schauder's Fixed Point Theorem for the victory ✋.

Finally we have all the ingredients:

- $E = \mathcal{C}([0,h])$, $U = \{y \in \mathcal{C}([0,h]) : \|y - T\|_\infty \leq K\}$ is a closed, convex subset of E.
- We have proved that the operator A is such that $\{Au : u \in U\}$ is relatively compact in E.

$\Rightarrow$ By Schauder's Fixed Point Theorem we have the existence of *at least* a solution. $\square$

A Peano existence theorem for first order equations

Proof. Our plan:

- ✔ proving that $Ay \in U$,
- ✔ showing that $A(U) = \{Au : u \in U\}$ is *relatively compact* (Ascoli-Arzelà),
- ✔ apply Schauder's Fixed Point Theorem for the victory ✋.

Finally we have all the ingredients:

- $E = \mathcal{C}([0,h])$, $U = \{y \in \mathcal{C}([0,h]) : \|y - T\|_\infty \leq K\}$ is a closed, convex subset of E.
- We have proved that the operator A is such that $\{Au : u \in U\}$ is relatively compact in E.

$\Rightarrow$ By Schauder's Fixed Point Theorem we have the existence of *at least* a solution. □

At last...

We have proved existence: what about uniqueness?

</> A programming idea

We could use the fixed-point iteration as an algorithm for obtaining a solution.

Uniqueness of the solution à-la-Picard-Lindelöf

As for the classical calculus case, to prove *uniqueness* we need Lipschitzianity of the system dynamics w.r.t. to the second component.

Uniqueness of the solution à-la-Picard-Lindelöf

As for the classical calculus case, to prove *uniqueness* we need Lipschitzianity of the system dynamics w.r.t. to the second component.

Weissinger's Fixed Point Theorem

Assume (U, d) to be a nonempty complete metric space, and let $\beta_j \geq 0$ for every $j \in \mathbb{N}_0$ and such that $\sum_{j=0}^{\infty} \beta_j$ converges. Furthermore, let the mapping $A : U \to U$ satisfy the inequality

$$d(A^j u, A^j v) \leq \beta_j d(u, v), \quad \forall j \in \mathbb{N}, \quad \forall\, u, v \in U.$$

Then A has a *uniquely determined fixed point* u^*. Moreover, for any $u_0 \in U$, the sequence $(A^j u_0)_{j=1}^{\infty}$ converge to this fixed point.

Uniqueness of the solution à-la-Picard-Lindelöf

As for the classical calculus case, to prove *uniqueness* we need Lipschitzianity of the system dynamics w.r.t. to the second component.

Weissinger's Fixed Point Theorem

Assume (U, d) to be a nonempty complete metric space, and let $\beta_j \geq 0$ for every $j \in \mathbb{N}_0$ and such that $\sum_{j=0}^{\infty} \beta_j$ converges. Furthermore, let the mapping $A : U \to U$ satisfy the inequality

$$d(A^j u, A^j v) \leq \beta_j d(u, v), \quad \forall j \in \mathbb{N}, \quad \forall\, u, v \in U.$$

Then A has a *uniquely determined fixed point* u^*. Moreover, for any $u_0 \in U$, the sequence $(A^j u_0)_{j=1}^{\infty}$ converge to this fixed point.

The plan

- ♻ Reuse the same set U, and map A from the existence proof,
- ≤ Prove the inequality and give an expression of the α_j in term of the Lipschitz constant.

Uniqueness of the solution à-la-Picard-Lindelöf

Theorem

Let $0 < \alpha$ and $m = \lceil \alpha \rceil$. Moreover, let $y_0^{(0)}, \ldots, y_0^{(m-1)} \in \mathbb{R}$, $K > 0$, and $h^* > 0$. We define the same set G:

$$G = \left\{ (t, y) \,:\, t \in [0, h^*] \,:\, \left| y - \sum_{k=0}^{m-1} t^k y_0^{(k)} / k! \right| \leq K \right\},$$

and let the function $f : G \to \mathbb{R}$ be continuous and Lipschitz w.r.t. the second entry

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2|,$$

for some $L > 0$ independently of t, y_1, and y_2. Then, for h such that

$$M = \sup_{(t,z) \in G} |f(t, z)|, \;\; h = \begin{cases} h^*, & \text{if } M = 0, \\ \min\{h^*, (K\Gamma(\alpha+1)/M^{1/n}\}, & \text{else.} \end{cases}$$

there exist a uniquely defined $y \in \mathcal{C}[0, h]$ solving (FODE).

Uniqueness of the solution à-la-Picard-Lindelöf

Proof. We are under the same hypotheses of the **Existence Theorem**, thus (FODE) has a solution.
We prove **by induction** on j that

$$\|A^j y - A^j \tilde{y}\|_\infty \leq \frac{(Lt^\alpha)^j}{\Gamma(1+\alpha j)} \|y - \tilde{y}\|_\infty.$$

Uniqueness of the solution à-la-Picard-Lindelöf

Proof. We are under the same hypotheses of the **Existence Theorem**, thus (FODE) has a solution.
We prove **by induction** on j that

$$\|A^j y - A^j \tilde{y}\|_\infty \leq \frac{(Lt^\alpha)^j}{\Gamma(1+\alpha j)} \|y - \tilde{y}\|_\infty.$$

Base case: $j = 0$ follows by the definition.

Uniqueness of the solution à-la-Picard-Lindelöf

Proof. We are under the same hypotheses of the **Existence Theorem**, thus (FODE) has a solution.
We prove **by induction** on j that

$$\|A^j y - A^j \tilde{y}\|_\infty \leq \frac{(Lt^\alpha)^j}{\Gamma(1+\alpha j)} \|y - \tilde{y}\|_\infty.$$

Inductive hypothesis: we assume it true for $j-1$ and prove it for j.
Inductive step:

$$\begin{aligned}
\|A^j y - A^j \tilde{y}\|_\infty =& \|A(A^j - 1y) - A(A^{j-1}\tilde{y})\|_\infty \\
=& \frac{1}{\Gamma(\alpha)} \sup_{0 \leq w \leq t} \left| \int_0^w (w-\tau)^{\alpha-1} \left[f(\tau, A^{j-1}y(\tau)) - f(\tau, A^{j-1}\tilde{y}(\tau)) \right] \, \mathrm{d}\tau \right| \\
\text{(Lipschitz)} \quad \leq& \frac{L}{\Gamma(\alpha)} \sup_{0 \leq w \leq t} \int_0^w (w-\tau)^{\alpha-1} \left| A^{j-1}y(\tau) - A^{j-1}\tilde{y}(\tau) \right| \, \mathrm{d}\tau
\end{aligned}$$

Uniqueness of the solution à-la-Picard-Lindelöf

Proof. We are under the same hypotheses of the **Existence Theorem**, thus (FODE) has a solution.
We prove **by induction** on j that

$$\|A^j y - A^j \tilde{y}\|_\infty \leq \frac{(Lt^\alpha)^j}{\Gamma(1+\alpha j)} \|y - \tilde{y}\|_\infty.$$

Inductive hypothesis: we assume it true for $j-1$ and prove it for j.
Inductive step:

$$\begin{aligned}
\|A^j y - A^j \tilde{y}\|_\infty =& \|A(A^j - 1y) - A(A^{j-1}\tilde{y})\|_\infty \\
\text{(Lipschitz)} \quad \leq& \frac{L}{\Gamma(\alpha)} \sup_{0 \leq w \leq t} \int_0^w (w-\tau)^{\alpha-1} \left| A^{j-1} y(\tau) - A^{j-1}\tilde{y}(\tau) \right| \, \mathrm{d}\tau \\
\leq& \frac{L}{\Gamma(\alpha)} \int_0^w (w-\tau)^{\alpha-1} \sup_{0 \leq w \leq \tau} \left| A^{j-1} y(w) - A^{j-1}\tilde{y}(w) \right| \, \mathrm{d}\tau
\end{aligned}$$

Uniqueness of the solution à-la-Picard-Lindelöf

Proof. We are under the same hypotheses of the **Existence Theorem**, thus (FODE) has a solution.
We prove **by induction** on j that

$$\|A^j y - A^j \tilde{y}\|_\infty \leq \frac{(Lt^\alpha)^j}{\Gamma(1+\alpha j)} \|y - \tilde{y}\|_\infty.$$

Inductive hypothesis: we assume it true for $j-1$ and prove it for j.
Inductive step:

$$\begin{aligned}
\|A^j y - A^j \tilde{y}\|_\infty =& \|A(A^j - 1y) - A(A^{j-1}\tilde{y})\|_\infty \\
\leq& \frac{L}{\Gamma(\alpha)} \int_0^w (w-\tau)^{\alpha-1} \sup_{0\leq w\leq \tau} \left|A^{j-1}y(w) - A^{j-1}\tilde{y}(w)\right| \, \mathrm{d}\tau \\
\text{(I.H.)} \quad \leq& \frac{L^j}{\Gamma(\alpha)\Gamma(1+\alpha(j-1))} \int_0^t (t-\tau)^{\alpha-1} t^{\alpha(j-1)} \sup_{0\leq w\leq \tau} |y(w) - \tilde{y}(w)| \, \mathrm{d}\tau
\end{aligned}$$

Uniqueness of the solution à-la-Picard-Lindelöf

Proof. We are under the same hypotheses of the **Existence Theorem**, thus (FODE) has a solution.
We prove **by induction** on j that

$$\|A^j y - A^j \tilde{y}\|_\infty \leq \frac{(Lt^\alpha)^j}{\Gamma(1+\alpha j)} \|y - \tilde{y}\|_\infty.$$

Inductive hypothesis: we assume it true for $j-1$ and prove it for j.
Inductive step:

$$\begin{aligned}
\|A^j y - A^j \tilde{y}\|_\infty =& \|A(A^j - 1y) - A(A^{j-1}\tilde{y})\|_\infty \\
\text{(I.H.)} \quad \leq& \frac{L^j}{\Gamma(\alpha)\Gamma(1+\alpha(j-1))} \int_0^t (t-\tau)^{\alpha-1} t^{\alpha(j-1)} \sup_{0\leq w\leq \tau} |y(w) - \tilde{y}(w)| \, \mathrm{d}\tau \\
\leq& \frac{L^j}{\Gamma(\alpha)\Gamma(1+\alpha(j-1))} \sup_{0\leq w\leq t} |y(w) - \tilde{y}(w)| \int_0^t (t-\tau)^{\alpha-1} t^{\alpha(j-1)} \, \mathrm{d}\tau
\end{aligned}$$

Uniqueness of the solution à-la-Picard-Lindelöf

Proof. We are under the same hypotheses of the **Existence Theorem**, thus (FODE) has a solution.
We prove **by induction** on j that

$$\|A^j y - A^j \tilde{y}\|_\infty \leq \frac{(Lt^\alpha)^j}{\Gamma(1+\alpha j)} \|y - \tilde{y}\|_\infty.$$

Inductive hypothesis: we assume it true for $j-1$ and prove it for j.
Inductive step:

$$\begin{aligned}
\|A^j y - A^j \tilde{y}\|_\infty =& \|A(A^j - 1y) - A(A^{j-1}\tilde{y})\|_\infty \\
\leq& \frac{L^j}{\Gamma(\alpha)\Gamma(1+\alpha(j-1))} \sup_{0 \leq w \leq t} |y(w) - \tilde{y}(w)| \int_0^t (t-\tau)^{\alpha-1} t^{\alpha(j-1)} \, \mathrm{d}\tau \\
\leq& \frac{L^j}{\Gamma(\alpha)\Gamma(1+\alpha(j-1))} \|y - \tilde{y}\|_\infty \frac{\Gamma(\alpha)\Gamma(1+\alpha(j-1))}{\Gamma(1+\alpha j)} t^{\alpha j}
\end{aligned}$$

Uniqueness of the solution à-la-Picard-Lindelöf

Proof. We are under the same hypotheses of the **Existence Theorem**, thus (FODE) has a solution.
We prove **by induction** on j that

$$\|A^j y - A^j \tilde{y}\|_\infty \leq \frac{(Lt^\alpha)^j}{\Gamma(1+\alpha j)} \|y - \tilde{y}\|_\infty.$$

Inductive hypothesis: we assume it true for $j-1$ and prove it for j.
Inductive step:

$$\begin{aligned}\|A^j y - A^j \tilde{y}\|_\infty =& \|A(A^j - 1y) - A(A^{j-1}\tilde{y})\|_\infty \\ \leq& \frac{L^j}{\Gamma(\alpha)\Gamma(1+\alpha(j-1))} \|y - \tilde{y}\|_\infty \frac{\Gamma(\alpha)\Gamma(1+\alpha(j-1))}{\Gamma(1+\alpha j)} t^{\alpha j}\end{aligned}$$

Uniqueness of the solution à-la-Picard-Lindelöf

Proof. We are under the same hypotheses of the **Existence Theorem**, thus (FODE) has a solution.
We prove **by induction** on j that

$$\|A^j y - A^j \tilde{y}\|_\infty \leq \frac{(Lt^\alpha)^j}{\Gamma(1+\alpha j)} \|y - \tilde{y}\|_\infty.$$

Inductive hypothesis: we assume it true for $j-1$ and prove it for j.
Inductive step:

$$\|A^j y - A^j \tilde{y}\|_\infty \leq \frac{(Lt^\alpha)^j}{\Gamma(1+\alpha j)} \|y - \tilde{y}\|_\infty.$$

Uniqueness of the solution à-la-Picard-Lindelöf

Proof. We are under the same hypotheses of the **Existence Theorem**, thus (FODE) has a solution.
We proved **by induction** on j that

$$\|A^j y - A^j \tilde{y}\|_\infty \leq \frac{(Lt^\alpha)^j}{\Gamma(1+\alpha j)}\|y-\tilde{y}\|_\infty = \alpha_j \|y-\tilde{y}\|_\infty, \qquad \alpha_j = \frac{(Lh)^\alpha}{\Gamma(1+\alpha j)}.$$

To apply Weissinger's Fixed Point Theorem we need to prove that the series $\displaystyle\sum_{j=0}^{+\infty} \alpha_j = \sum_{j=0}^{+\infty} \frac{(Lh)^\alpha}{\Gamma(1+\alpha j)}$ converges.

Mittag-Leffler

$$E_\alpha(z) = \sum_{k=0}^{+\infty} \frac{z^\alpha}{\Gamma(\alpha k+1)}, \quad \alpha > 0 \qquad \text{is an entire function.}$$

State of the art

We have proved that the *Cauchy problem*

$$\alpha > 0, \quad m = \lceil \alpha \rceil, \qquad \begin{cases} {}_C D^{\alpha}_{[0,t]} \mathbf{y}(t) = f(t, \mathbf{y}(t)), & t \in [0, T], \\ \dfrac{d^k \mathbf{y}(0)}{dt^k} = \mathbf{y}_0^{(k)}, & k = 0, 1, \ldots, m-1. \end{cases}$$

admits

- for f continuous a *local* solution in $\mathcal{C}([0, h])$, $h < h^*$,
- for f continuous and Lipschitz in the second entry a *local* and *unique* solution in $\mathcal{C}([0, h])$, $h < h^*$.

State of the art

We have proved that the *Cauchy problem*

$$\alpha > 0, \quad m = \lceil \alpha \rceil, \qquad \begin{cases} {}_C D^{\alpha}_{[0,t]} \mathbf{y}(t) = f(t, \mathbf{y}(t)), & t \in [0, T], \\ \dfrac{d^k \mathbf{y}(0)}{dt^k} = \mathbf{y}_0^{(k)}, & k = 0, 1, \ldots, m-1. \end{cases}$$

admits

- for f continuous a *local* solution in $\mathcal{C}([0, h])$, $h < h^*$,
- for f continuous and Lipschitz in the second entry a *local* and *unique* solution in $\mathcal{C}([0, h])$, $h < h^*$.

For classical ODEs this is the point in which one starts proving *extension results* for the solutions. They exist also for the Fractional case. We are going to state them without proof.

Extension results

Corollary

Assume the hypotheses of the existence Theorem, but substitute G with the domain of definition of f, i.e., $G = [0, h^*] \times \mathbb{R}$. Moreover, assume that f is continuous and that there exist constants $c_1 \geq 0, c_2 \geq 0$, $0 \leq \mu < 1$ such that

$$f(t, y) \leq c_1 + c_2 |y|^{\mu}, \quad \forall (t, y) \in G.$$

Then, there exists a function $y \in \mathcal{C}([0, h^*])$ solving (FODE).

- Since G is no longer compact we need to demand a suitable bound explicitly, Weierstrasse Theorem no longer applies,
- A sufficient condition on f to imply the decay we need is for f to be continuous and bounded on G,

⚠ Our condition is violated already by linear equations!

Extension results

Theorem

Let $0 < \alpha$ and $m = \lceil \alpha \rceil$. Moreover, let $y_0^{(0)}, \ldots, y_0^{(m-1)} \in \mathbb{R}$ and $h^* > 0$. We define the set $G = [0, h^*] \times \mathbb{R}$ and let $f : G \to \mathbb{R}$ be continuous and fulfill a Lipschitz condition with respect to the second variable with a Lipischitz constant L that is independent of t, y_1, and y_2. Then there exist a uniquely defined function $y \in \mathcal{C}([0, h^*])$ solving the (FODE).

- For a **proof** see the proof of Theorem 6.8 from (Diethelm 2010, pp 96-102) that is inspired by the proof for Volterra integral equations in (Linz 1985, Theorem 4.8).
- ☺ We can now solve linear equations

$$ {}_{CA}D^{\alpha}_{[0,t]} y(t) = f(t)y(t) + g(t), \qquad f, g \in \mathcal{C}([0, h^*]), \quad L = \|f\|_\infty < \infty. $$

Extension results

Theorem

Let $0 < \alpha$ and $m = \lceil \alpha \rceil$. Moreover, let $y_0^{(0)}, \ldots, y_0^{(m-1)} \in \mathbb{R}$ and $h^* > 0$. We define the set $G = [0, h^*] \times \mathbb{R}$ and let $f : G \to \mathbb{R}$ be continuous and fulfill a Lipschitz condition with respect to the second variable with a Lipischitz constant L that is independent of t, y_1, and y_2. Then there exist a uniquely defined function $y \in \mathcal{C}([0, h^*])$ solving the (FODE).

- For a **proof** see the proof of Theorem 6.8 from (Diethelm 2010, pp 96-102) that is inspired by the proof for Volterra integral equations in (Linz 1985, Theorem 4.8).
- ☺ We can now solve linear equations

$$ {}_{CA}D^{\alpha}_{[0,t]} y(t) = f(t)y(t) + g(t), \qquad f, g \in \mathcal{C}([0, h^*]), \quad L = \|f\|_\infty < \infty. $$

- ? Do we know hot to solve by hand any simple FODE?

Simple cases and representation formulas

The simplest ODE we know ho to solve is the *relaxation equation*

$$\mathbb{R} \ni \lambda < 0, \qquad \begin{cases} y'(t) = \lambda y(t), & t \in [0, T], \\ y(0) = y_0, \end{cases} \qquad y(t) = y_0 \exp(\lambda t).$$

Simple cases and representation formulas

The simplest ODE we know ho to solve is the *relaxation equation*

$$\mathbb{R} \ni \lambda < 0, \quad \begin{cases} y'(t) = \lambda y(t), & t \in [0, T], \\ y(0) = y_0, \end{cases} \qquad y(t) = y_0 \exp(\lambda t).$$

Relaxation FODE

Let $\alpha > 0$, $m = \lceil \alpha \rceil$ and $\lambda \in \mathbb{R}$. The solution of the Cauchy problem

$${}_{CA}D_{[0,t]} y(t) = \lambda y(t), \quad y(0) = y_0, \quad y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1,$$

is given by

$$y(t) = y_0 E_\alpha(\lambda t^\alpha), \quad t \geq 0.$$

Simple cases and representation formulas

The simplest ODE we know ho to solve is the *relaxation equation*

$$\mathbb{R} \ni \lambda < 0, \quad \begin{cases} y'(t) = \lambda y(t), & t \in [0, T], \\ y(0) = y_0, \end{cases} \qquad y(t) = y_0 \exp(\lambda t).$$

Relaxation FODE

Let $\alpha > 0$, $m = \lceil \alpha \rceil$ and $\lambda \in \mathbb{R}$. The solution of the Cauchy problem

$$_{CA}D_{[0,t]} y(t) = \lambda y(t), \quad y(0) = y_0, \quad y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1,$$

is given by

$$y(t) = y_0 E_\alpha(\lambda t^\alpha), \quad t \geq 0.$$

- The previous existence result tells us that the problem has indeed a *unique solution*.

Solution of the relaxation FODE

Two parameters Mittag-Leffler

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{+\infty} \frac{z^\alpha}{\Gamma(\alpha k + \beta)}, \quad \alpha, \beta > 0 \qquad \text{is an entire function.}$$

To see that this is the case we can use **Stirling formula** and **root test**

Stirling: $\Gamma(x+1) = (x/e)^x \sqrt{2\pi x}(1 + o(1))$ for $x \to +\infty$,

Root test: $\sum_{n=1}^{+\infty} a_n$ converge absolutely if $C = \limsup_{n\to+\infty} \sqrt[n]{|a_n|} < 1$.

We write

$$a_j^{1/j} = \left(\frac{e}{j\alpha + \beta}\right)^{\alpha + \beta/j} (2\pi(\alpha j + \beta))^{-1/2j}(1 + o(1)) \to 0 \text{ for } j \to \infty.$$

Thus the radius of convergence is infinite.

Solution of the relaxation FODE

$$\alpha > 0,\ m = \lceil \alpha \rceil,\ {}_{CA}D_{[0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1,$$

1. $y(0) = y_0 E_\alpha(0) = y_0$ since

$$E_\alpha(z) = 1 + \frac{z}{\Gamma(\alpha+1)} + \frac{z^2}{\Gamma(2\alpha+1)} + \ldots,$$

Solution of the relaxation FODE

$$\alpha > 0,\ m = \lceil \alpha \rceil,\ {}_{CA}D_{[0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1,$$

1. $y(0) = y_0 E_\alpha(0) = y_0$ since

$$E_\alpha(z) = 1 + \frac{z}{\Gamma(\alpha+1)} + \frac{z^2}{\Gamma(2\alpha+1)} + \ldots,$$

2. If $\alpha > 1,\ m \geq 2,\ y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1$

$$y(t) = 1 + \frac{\lambda t^\alpha}{\Gamma(\alpha+1)} + \frac{\lambda^2 t^{2\alpha}}{\Gamma(2\alpha+1)} + \ldots,$$

imposing the condition on the derivatives implies

$$y^{(k)}(t) = \frac{\lambda t^{\alpha-k}}{\Gamma(\alpha+1-k)} + \frac{\lambda^2 t^{2\alpha-k}}{\Gamma(2\alpha+1-k)} + \ldots, \quad k = 1, 2, \ldots, m-1.$$

Solution of the relaxation FODE

$$\alpha > 0,\ m = \lceil \alpha \rceil,\ {}_{CA}D_{[0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1,$$

Let $p_k(t) = t^k$

$$ {}_{CA}D_{[0,t]}y(t) = {}_{CA}D_{[0,t]}\left[\sum_{j=0}^{+\infty} \frac{(\lambda p_\alpha)^j}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\frac{d^m}{dt^m}\left[\sum_{j=0}^{+\infty} \frac{\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right]$$

Solution of the relaxation FODE

$$\alpha > 0,\ m = \lceil \alpha \rceil,\ {}_{CA}D_{[0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1,$$

Let $p_k(t) = t^k$

$$\begin{aligned} {}_{CA}D_{[0,t]}y(t) &= {}_{CA}D_{[0,t]}\left[\sum_{j=0}^{+\infty} \frac{(\lambda p_\alpha)^j}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\frac{d^m}{dt^m}\left[\sum_{j=0}^{+\infty} \frac{\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] \\ &= I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty} \frac{\frac{d^m}{dt^m}\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] \end{aligned}$$

Solution of the relaxation FODE

$$\alpha > 0,\ m = \lceil \alpha \rceil,\ {}_{CA}D_{[0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1,$$

Let $p_k(t) = t^k$

$$\begin{aligned} {}_{CA}D_{[0,t]}y(t) &= {}_{CA}D_{[0,t]}\left[\sum_{j=0}^{+\infty} \frac{(\lambda p_\alpha)^j}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\frac{d^m}{dt^m}\left[\sum_{j=0}^{+\infty} \frac{\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] \\ &= I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty} \frac{\frac{d^m}{dt^m}\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty} \frac{\lambda^j p_{j\alpha-m}}{\Gamma(j\alpha+1-m)}\right] \end{aligned}$$

Solution of the relaxation FODE

$$\alpha > 0,\ m = \lceil \alpha \rceil,\ {}_{CA}D_{[0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1,$$

Let $p_k(t) = t^k$

$$\begin{aligned}
{}_{CA}D_{[0,t]}y(t) &= {}_{CA}D_{[0,t]}\left[\sum_{j=0}^{+\infty} \frac{(\lambda p_\alpha)^j}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\frac{d^m}{dt^m}\left[\sum_{j=0}^{+\infty} \frac{\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] \\
&= I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty} \frac{\frac{d^m}{dt^m}\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty} \frac{\lambda^j p_{j\alpha-m}}{\Gamma(j\alpha+1-m)}\right] \\
&= \sum_{j=1}^{+\infty} \frac{\lambda^j I_0^{m-\alpha} p_{j\alpha-m}}{\Gamma(j\alpha+1-m)}
\end{aligned}$$

Solution of the relaxation FODE

$$\alpha > 0,\ m = \lceil \alpha \rceil,\ {}_{CA}D_{[0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1,$$

Let $p_k(t) = t^k$

$$\begin{aligned}
{}_{CA}D_{[0,t]}y(t) &= {}_{CA}D_{[0,t]}\left[\sum_{j=0}^{+\infty}\frac{(\lambda p_\alpha)^j}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\frac{d^m}{dt^m}\left[\sum_{j=0}^{+\infty}\frac{\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] \\
&= I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty}\frac{\frac{d^m}{dt^m}\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty}\frac{\lambda^j p_{j\alpha-m}}{\Gamma(j\alpha+1-m)}\right] \\
&= \sum_{j=1}^{+\infty}\frac{\lambda^j I_0^{m-\alpha} p_{j\alpha-m}}{\Gamma(j\alpha+1-m)} = \sum_{j=1}^{+\infty}\frac{\lambda^j p_{j\alpha-\alpha}}{\Gamma(j\alpha+1-\alpha)}
\end{aligned}$$

Solution of the relaxation FODE

$$\alpha > 0,\ m = \lceil \alpha \rceil,\ {}_{CA}D_{[0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1,$$

Let $p_k(t) = t^k$

$$\begin{aligned}
{}_{CA}D_{[0,t]}y(t) &= {}_{CA}D_{[0,t]}\left[\sum_{j=0}^{+\infty} \frac{(\lambda p_\alpha)^j}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\frac{d^m}{dt^m}\left[\sum_{j=0}^{+\infty} \frac{\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] \\
&= I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty} \frac{\frac{d^m}{dt^m}\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty} \frac{\lambda^j p_{j\alpha-m}}{\Gamma(j\alpha+1-m)}\right] \\
&= \sum_{j=1}^{+\infty} \frac{\lambda^j I_0^{m-\alpha} p_{j\alpha-m}}{\Gamma(j\alpha+1-m)} = \sum_{j=1}^{+\infty} \frac{\lambda^j t^{j\alpha-\alpha}}{\Gamma(j\alpha+1-\alpha)}
\end{aligned}$$

Solution of the relaxation FODE

$$\alpha > 0,\ m = \lceil \alpha \rceil,\ {}_{CA}D_{[0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1,$$

Let $p_k(t) = t^k$

$$\begin{aligned}
{}_{CA}D_{[0,t]}y(t) &= {}_{CA}D_{[0,t]}\left[\sum_{j=0}^{+\infty} \frac{(\lambda p_\alpha)^j}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\frac{d^m}{dt^m}\left[\sum_{j=0}^{+\infty} \frac{\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] \\
&= I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty} \frac{\frac{d^m}{dt^m}\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty} \frac{\lambda^j p_{j\alpha-m}}{\Gamma(j\alpha+1-m)}\right] \\
&= \sum_{j=1}^{+\infty} \frac{\lambda^j t^{j\alpha-\alpha}}{\Gamma(j\alpha+1-\alpha)} \sum_{j=0}^{+\infty} \frac{\lambda^{j+1} t^{\alpha j}}{\Gamma(\alpha j+1)}
\end{aligned}$$

Solution of the relaxation FODE

$$\alpha > 0,\ m = \lceil \alpha \rceil,\ {}_{CA}D_{[0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad y^{(k)}(0) = 0, \quad k = 1, 2, \ldots, m-1,$$

Let $p_k(t) = t^k$

$$\begin{aligned}
{}_{CA}D_{[0,t]}y(t) &= {}_{CA}D_{[0,t]}\left[\sum_{j=0}^{+\infty} \frac{(\lambda p_\alpha)^j}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\frac{d^m}{dt^m}\left[\sum_{j=0}^{+\infty} \frac{\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] \\
&= I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty} \frac{\frac{d^m}{dt^m}\lambda^j p_{j\alpha}}{\Gamma(j\alpha+1)}\right] = I_0^{m-\alpha}\left[\sum_{j=1}^{+\infty} \frac{\lambda^j p_{j\alpha-m}}{\Gamma(j\alpha+1-m)}\right] \\
&= \sum_{j=1}^{+\infty} \frac{\lambda^j t^{j\alpha-\alpha}}{\Gamma(j\alpha+1-\alpha)} = \lambda y(t).
\end{aligned}$$

Why only continuous solutions?

The existence theorems we have seen give us a solution $y(t) \in \mathcal{C}([0, h])$, can we have more?

Why only continuous solutions?

The existence theorems we have seen give us a solution $y(t) \in \mathcal{C}([0, h])$, can we have more?

Regularity for ODEs

$$k \in \mathbb{N},\ f \in \mathcal{C}^{k-1}([y_0 - K, y_0 + k] \times \mathbb{R}),\ \begin{cases} y'(t) = f(t, y(t)), \\ y(0) = y_0 \end{cases} \quad \Rightarrow\ y(t) \in \mathcal{C}^k.$$

Why only continuous solutions?

The existence theorems we have seen give us a solution $y(t) \in \mathcal{C}([0,h])$, can we have more?

Regularity for ODEs

$$k \in \mathbb{N},\ f \in \mathcal{C}^{k-1}([y_0 - K, y_0 + k] \times \mathbb{R}),\ \begin{cases} y'(t) = f(t, y(t)), \\ y(0) = y_0 \end{cases} \Rightarrow\ y(t) \in \mathcal{C}^k.$$

We can reuse our example computation:

$$f(t) = (t-a)^{\beta},\ \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\alpha)}(t-a)^{\beta-\alpha},\ \beta \notin \mathbb{N} \wedge \beta > \lceil \alpha \rceil - 1$$

If we select $a = 0,\ \alpha = 1/2,\ \beta = 1/2$, then

$$\begin{cases} {}_{CA}D_{[0,t]}y(t) = \Gamma(3/2), \\ y(0) = 0, \end{cases} \Rightarrow\ y(t) = \sqrt{x}.$$

From an analytic right-hand side we got a non differentiable solution.

Why only continuous solutions?

Take-home message

Regularity of the right-hand side of the (FODE) is not sufficient to ensure regularity of the solution.

- Some more restrictive conditions under which regularity can be ensured can be found in (Diethelm 2007), to give an idea, one have to further ensure conditions for the zeros of $z(t) = f(t, y(t))$.
- Furthermore, if the solution of (FODE) is analytic, but not a polynomial of degree $\lceil \alpha \rceil - 1$, then f is not analytic.

Why only continuous solutions?

Take-home message

Regularity of the right-hand side of the (FODE) is not sufficient to ensure regularity of the solution.

- Some more restrictive conditions under which regularity can be ensured can be found in (Diethelm 2007), to give an idea, one have to further ensure conditions for the zeros of $z(t) = f(t, y(t))$.
- Furthermore, if the solution of (FODE) is analytic, but not a polynomial of degree $\lceil \alpha \rceil - 1$, then f is not analytic.
- This will be important when we try do design *numerical methods*, since many results on convergence order usually rely on the regularity of the solution. Going high-order in the fractional settings is not in general an easy task!

The Mittag-Leffler Function

The $E_{\alpha,\beta}(z)$ takes the role of the exponential function when moving from ODEs to FODEs.

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{+\infty} \frac{z^{\alpha}}{\Gamma(\alpha k + \beta)}, \quad \alpha, \beta > 0.$$

❓ How can we compute it?

The Mittag-Leffler Function

The $E_{\alpha,\beta}(z)$ takes the role of the exponential function when moving from ODEs to FODEs.

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{+\infty} \frac{z^{\alpha}}{\Gamma(\alpha k + \beta)}, \quad \alpha, \beta > 0.$$

- How can we compute it?
  - Using the series representation,

The Mittag-Leffler Function

The $E_{\alpha,\beta}(z)$ takes the role of the exponential function when moving from ODEs to FODEs.

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{+\infty} \frac{z^{\alpha}}{\Gamma(\alpha k + \beta)}, \quad \alpha, \beta > 0.$$

- How can we compute it?
  - Using the series representation,
  - A quadrature formula applied to an integral representation,

The Mittag-Leffler Function

The $E_{\alpha,\beta}(z)$ takes the role of the exponential function when moving from ODEs to FODEs.

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{+\infty} \frac{z^{\alpha}}{\Gamma(\alpha k + \beta)}, \quad \alpha, \beta > 0.$$

- How can we compute it?
  - Using the series representation,
  - A quadrature formula applied to an integral representation,
  - Inversion of the Laplace transform.

The Mittag-Leffler Function

The $E_{\alpha,\beta}(z)$ takes the role of the exponential function when moving from ODEs to FODEs.

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{+\infty} \frac{z^{\alpha}}{\Gamma(\alpha k + \beta)}, \quad \alpha, \beta > 0.$$

How can we compute it?

- Using the series representation,
- A quadrature formula applied to an integral representation,
- Inversion of the Laplace transform.

Laplace Transform

For a real- or complex-valued function $f(t)$ of the real variable t defined on $\mathbb{R}$ the (two-sided) Laplace transform is defined as

$$F(s) = \mathcal{L}\{f\}(s) = \int_{-\infty}^{+\infty} e^{-st} f(t)\,\mathrm{d}t.$$

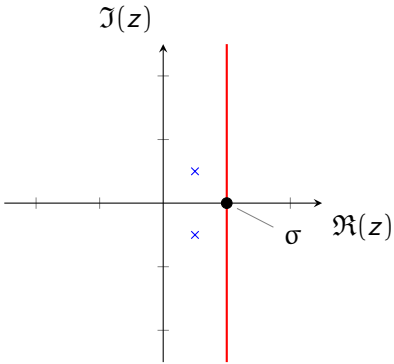
Inverting the Laplace Transform

If we want fo compute $f(t)$ and have access to $F(s) = \mathcal{L}\{f\}(s)$ we can perform a *numerical inversion*, that is

$$f(t) = \frac{1}{2\pi i} \int_{\sigma - i\infty}^{\sigma + \infty} e^{st} F(s) \, \mathrm{d}s.$$

where

- $(\sigma - i\infty, \sigma + i\infty)$ is called the Bromwich line,
- σ is such that all the singularities of $F(s)$ lies to the left $\Re(s) = \sigma$.



Branch lines

If $F(s)$ is a *multivalued function* we need to add a branch-cut to make the integrand single-valued.

Inverting the Laplace Transform

To numerically approximate the integral

$$f(t) = \frac{1}{2\pi i} \int_{\sigma - i\infty}^{\sigma + i\infty} e^{st} F(s) \, \mathrm{d}s.$$

we **always need a change of variable**, the exponential term *oscillates wildly* and *decays slowly* along the Bromwich line.
We have to change the countour of integration to something more suitable, i.e., we change

$$s = s(u) \mapsto f(t) = \frac{1}{2\pi i} \int_{-\infty}^{+\infty} e^{s(u)t} F(s(u)) s'(u) \, \mathrm{d}u,$$

and then approximate the integral with the *trapezoidal rule* with spacing h

$$f_{h,N}(t) = \frac{h}{2\pi i} \sum_{k=-N}^{N} e^{s(u_k)t} F(s(u_k)) s'(u_k), \quad u_k = kh.$$

Inverting the Laplace Transform

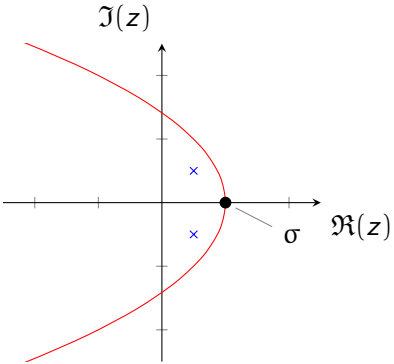
❓ What is the **best contour**?

Inverting the Laplace Transform

❓ What is the **best contour**?

- Parabolic contour:

$$s = \mu(iu+1)^2, \qquad -\infty < u < \infty$$



Inverting the Laplace Transform

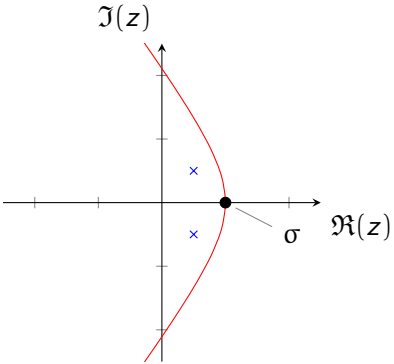
❓ What is the **best contour**?

- Parabolic contour:

$$s = \mu(iu+1)^2, \qquad -\infty < u < \infty$$

- Hyperbolic contour:

$$s = \mu(1+\sin(iu-\alpha)), \qquad -\infty < u < \infty$$



Inverting the Laplace Transform

❓ What is the **best contour**?

- Parabolic contour:

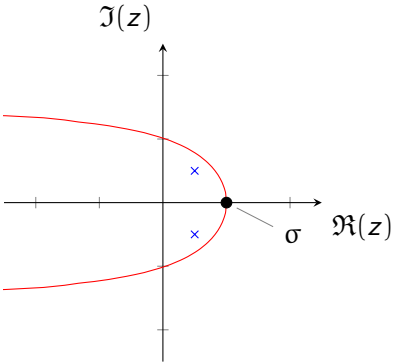
$$s = \mu(iu+1)^2, \qquad -\infty < u < \infty$$

- Hyperbolic contour:

$$s = \mu(1+\sin(iu-\alpha)), \qquad -\infty < u < \infty$$

- Talbot contour:

$$s = -\sigma + \mu\theta\cot(\alpha\theta) + i\theta\nu, \qquad -\pi \le \theta \le \pi$$



Inverting the Laplace Transform

❓ What is the **best contour**?

- Parabolic contour:

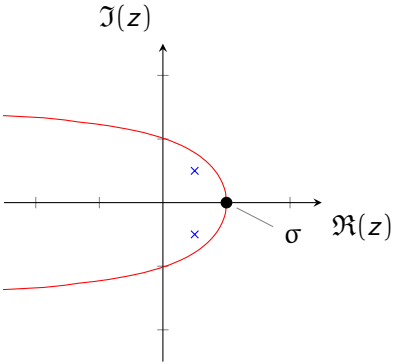
$$s = \mu(iu+1)^2, \qquad -\infty < u < \infty$$

- Hyperbolic contour:

$$s = \mu(1+\sin(iu-\alpha)), \qquad -\infty < u < \infty$$

- Talbot contour:

$$s = -\sigma + \mu\theta\cot(\alpha\theta) + i\theta\nu, \qquad -\pi \le \theta \le \pi$$



Selecting **contour** and **parameters** depends on the **error analysis**.

Inverting the Laplace Transform

- All the contours exploit the fact that e^{st} decays rapidly as $\Re(s) \to -\infty$,
  - **Trapezoidal rule** for integral on the real line for which the integrand decay sufficiently rapidly is **exponential**:

Theorem (Trefethen and Weideman 2014, Theorem 5.1)

Suppose that w is analytic in the strip $|\Im(x)| < a$ for some $a > 0$. Suppose further that $w(x) \to 0$ uniformly as $|x| \to +\infty$ in the strip, and that for some M it satisfies

$$\int_{-\infty}^{+\infty} |w(x + ib)| \, \mathrm{d}x \leq M, \quad \forall b \in (-a, a),$$

then for any $h > 0$, the trapezoidal rule $w_{h,N}$ with step-size h exists and satisfies

$$|w_h - \int_{-\infty}^{+\infty} w(x) \, \mathrm{d}x| \leq \frac{2M}{\exp(2\pi a/h) - 1},$$

and the quantity $2M$ on the numerator is as small as possible.

Inverting the Laplace Transform

- All the contours exploit the fact that e^{st} decays rapidly as $\Re(s) \to -\infty$,
  - **Trapezoidal rule** for integral on the real line for which the integrand decay sufficiently rapidly is **exponential**:

Steepest descent contours

For some functions it is possible to use a technique called "saddle point technique" from complex analysis to estimate the asymptotic of complex integrals. This determines the optimal steepest descent contour.

References for the general problem are:

Talbot: Dingfelder and Weideman 2015; Trefethen, Weideman, and Schmelzer 2006; Weideman 2006,

Parabolic & Hyperbolic: Weideman and Trefethen 2007.

Our case: we've got poles and a branch cut

In our case the function for which we can compute the *Laplace transform* is

$$e_{\alpha,\beta}(t;\lambda) = t^{\beta-1}E_{\alpha,\beta}(t^{\alpha}\lambda), \quad t \in \mathbb{R}_+, \qquad \lambda \in \mathcal{C}.$$

That is given by

$$\mathcal{E}_{\alpha,\beta}(t;\lambda) = \frac{s^{\alpha-\beta}}{s^{\alpha}-\lambda}, \quad \Re(s) > 0, \qquad |\lambda s^{-\alpha}| < 1.$$

Our case: we've got poles and a branch cut

In our case the function for which we can compute the *Laplace transform* is

$$e_{\alpha,\beta}(t;\lambda) = t^{\beta-1}E_{\alpha,\beta}(t^{\alpha}\lambda), \quad t \in \mathbb{R}_+, \qquad \lambda \in \mathcal{C}.$$

That is given by

$$\mathcal{E}_{\alpha,\beta}(t;\lambda) = \frac{s^{\alpha-\beta}}{s^{\alpha}-\lambda}, \quad \Re(s) > 0, \qquad |\lambda s^{-\alpha}| < 1.$$

- There are non-integer powers $\Rightarrow \mathcal{E}_{\alpha,\beta}$ is a multivalued function and a branch-cut on the real negative semi-axis is needed,

Our case: we've got poles and a branch cut

In our case the function for which we can compute the *Laplace transform* is

$$e_{\alpha,\beta}(t;\lambda) = t^{\beta-1}E_{\alpha,\beta}(t^{\alpha}\lambda), \quad t \in \mathbb{R}_+, \qquad \lambda \in \mathcal{C}.$$

That is given by

$$\mathcal{E}_{\alpha,\beta}(t;\lambda) = \frac{s^{\alpha-\beta}}{s^{\alpha}-\lambda}, \quad \Re(s) > 0, \qquad |\lambda s^{-\alpha}| < 1.$$

- There are non-integer powers $\Rightarrow \mathcal{E}_{\alpha,\beta}$ is a multivalued function and a branch-cut on the real negative semi-axis is needed,
- We have also the **poles** for $\theta = \arg(\lambda)$

$$\overline{s}_j^* = \lambda^{1/\alpha} = |\lambda|^{1/\alpha} e^{i\frac{\theta+2\pi j}{\alpha}}, \quad \left\{ j \in \mathbb{Z} \,\middle|\, -\frac{\alpha}{2} - \frac{\theta}{2\pi} < j \leq \frac{\alpha}{2} - \frac{\theta}{2\pi} \right\},$$

Our case: we've got poles and a branch cut

In our case the function for which we can compute the *Laplace transform* is

$$e_{\alpha,\beta}(t;\lambda) = t^{\beta-1}E_{\alpha,\beta}(t^{\alpha}\lambda), \quad t \in \mathbb{R}_+, \qquad \lambda \in \mathcal{C}.$$

That is given by

$$\mathcal{E}_{\alpha,\beta}(t;\lambda) = \frac{s^{\alpha-\beta}}{s^{\alpha}-\lambda}, \quad \Re(s) > 0, \qquad |\lambda s^{-\alpha}| < 1.$$

- There are non-integer powers $\Rightarrow \mathcal{E}_{\alpha,\beta}$ is a multivalued function and a branch-cut on the real negative semi-axis is needed,
- We have also the **poles** for $\theta = \arg(\lambda)$

$$\overline{s}_j^* = \lambda^{1/\alpha} = |\lambda|^{1/\alpha} e^{i\frac{\theta+2\pi j}{\alpha}}, \quad \left\{ j \in \mathbb{Z} \,\middle|\, -\frac{\alpha}{2} - \frac{\theta}{2\pi} < j \leq \frac{\alpha}{2} - \frac{\theta}{2\pi} \right\},$$

⚠ There could be lots of poles! Finding suitable contours is difficult.

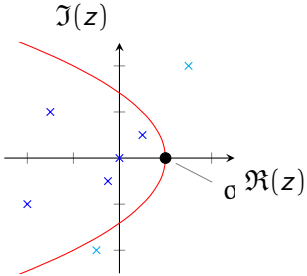
Cauchy's residue theorem to the rescue

We can use Cauchy's residue theorem if we have too many poles

$$e_{\alpha,\beta}(t;\lambda) = \sum_{s^* \in \mathcal{S}_{\mathcal{C}}^*} \mathrm{Res}(e^{st}\mathcal{E}_{\alpha,\beta}(s;\lambda), s^*) + \frac{1}{2\pi i}\int_{\mathcal{C}} e^{st}\mathcal{E}_{\alpha,\beta}(s;\lambda)\,\mathrm{d}s.$$

- $\mathcal{S}_{\mathcal{C}}^*$ is the set of all singularities lying on the rightmost part of the complex plane delimited by $\mathcal{C}$,
- We can compute the residual in close form:

$$\mathrm{Res}(e^{st}\mathcal{E}_{\alpha,\beta}(s;\lambda), s^*) = \frac{1}{\alpha}(s^*)^{1-\beta}e^{s^*t}.$$



The full algorithm (Garrappa 2015)

To build the full algorithm few technical steps are needed:

1. Finding an ordering of the poles,

$$\phi(s) = \frac{\Re(s) + |s|}{2}, \qquad 0 = \phi(s_0^*) < \phi(s_1^*) < \cdots < \phi(s_J^*),$$

The full algorithm (Garrappa 2015)

To build the full algorithm few technical steps are needed:

1. Finding an ordering of the poles,

$$\phi(s) = \frac{\Re(s) + |s|}{2}, \qquad 0 = \phi(s_0^*) < \phi(s_1^*) < \cdots < \phi(s_J^*),$$

2. Consider $J+1$ parabolas $s = \phi(s_j^*)(u+1)^2$ and the relevant $J+1$ plane regions R_j,

The full algorithm (Garrappa 2015)

To build the full algorithm few technical steps are needed:

1. Finding an ordering of the poles,

$$\phi(s) = \frac{\Re(s) + |s|}{2}, \qquad 0 = \phi(s_0^*) < \phi(s_1^*) < \cdots < \phi(s_J^*),$$

2. Consider $J+1$ parabolas $s = \phi(s_j^*)(u+1)^2$ and the relevant $J+1$ plane regions R_j,
3. The regions R_j are the analyticity regions to use in the Trefethen and Weideman result,

The full algorithm (Garrappa 2015)

To build the full algorithm few technical steps are needed:

1. Finding an ordering of the poles,

$$\phi(s) = \frac{\Re(s) + |s|}{2}, \qquad 0 = \phi(s_0^*) < \phi(s_1^*) < \cdots < \phi(s_J^*),$$

2. Consider $J+1$ parabolas $s = \phi(s_j^*)(u+1)^2$ and the relevant $J+1$ plane regions R_j,
3. The regions R_j are the analyticity regions to use in the Trefethen and Weideman result,
4. Obtain bounds on the discretization error and use it to determine optimal μ_j, step-size h_j and number of quadrature nodes N_k,

The full algorithm (Garrappa 2015)

To build the full algorithm few technical steps are needed:

1. Finding an ordering of the poles,

$$\phi(s) = \frac{\Re(s) + |s|}{2}, \qquad 0 = \phi(s_0^*) < \phi(s_1^*) < \cdots < \phi(s_J^*),$$

2. Consider $J+1$ parabolas $s = \phi(s_j^*)(u+1)^2$ and the relevant $J+1$ plane regions R_j,
3. The regions R_j are the analyticity regions to use in the Trefethen and Weideman result,
4. Obtain bounds on the discretization error and use it to determine optimal μ_j, step-size h_j and number of quadrature nodes N_k,
5. Select the best region R_j w.r.t. the lowest computation and reduction of round-off errors.

The full algorithm (Garrappa 2015)

To build the full algorithm few technical steps are needed:

1. Finding an ordering of the poles,

$$\phi(s) = \frac{\Re(s) + |s|}{2}, \qquad 0 = \phi(s_0^*) < \phi(s_1^*) < \cdots < \phi(s_J^*),$$

2. Consider $J+1$ parabolas $s = \phi(s_j^*)(u+1)^2$ and the relevant $J+1$ plane regions R_j,
3. The regions R_j are the analyticity regions to use in the Trefethen and Weideman result,
4. Obtain bounds on the discretization error and use it to determine optimal μ_j, step-size h_j and number of quadrature nodes N_k,
5. Select the best region R_j w.r.t. the lowest computation and reduction of round-off errors.

</> it.mathworks.com/matlabcentral/fileexchange/48154-the-mittag-leffler-function

Summary and anticipations

We did

- ✅ Uncovered properties of Riemann-Liouville Derivatives,
- ✅ Introduced the Caputo Derivative,
- ✅ Formulation, existence and uniqueness results for FODEs,
- ✅ The Mittag-Leffler function and its computation.

Next up

- 📋 Numerical methods for the integration of FODEs.

Bibliography I

- Caputo, M. (2008). "Linear models of dissipation whose Q is almost frequency independent. II". In: *Fract. Calc. Appl. Anal.* 11.1. Reprinted from Geophys. J. R. Astr. Soc. **13** (1967), no. 5, 529–539, pp. 4–14. ISSN: 1311-0454.
- Diethelm, K. (2007). "Smoothness properties of solutions of Caputo-type fractional differential equations". In: *Fract. Calc. Appl. Anal.* 10.2, pp. 151–160. ISSN: 1311-0454.
- — (2010). *The analysis of fractional differential equations*. Vol. 2004. Lecture Notes in Mathematics. An application-oriented exposition using differential operators of Caputo type. Springer-Verlag, Berlin, pp. viii+247. ISBN: 978-3-642-14573-5. DOI: 10.1007/978-3-642-14574-2. URL: https://doi.org/10.1007/978-3-642-14574-2.
- Diethelm, K. and N. J. Ford (2002). "Analysis of fractional differential equations". In: *J. Math. Anal. Appl.* 265.2, pp. 229–248. ISSN: 0022-247X. DOI: 10.1006/jmaa.2000.7194. URL: https://doi.org/10.1006/jmaa.2000.7194.
- Dingfelder, B. and J. A. C. Weideman (2015). "An improved Talbot method for numerical Laplace transform inversion". In: *Numer. Algorithms* 68.1, pp. 167–183. ISSN: 1017-1398. DOI: 10.1007/s11075-014-9895-z. URL: https://doi.org/10.1007/s11075-014-9895-z.

Bibliography II

- Džrbašjan, M. M. and A. B. Nersesjan (1968). "Fractional derivatives and the Cauchy problem for differential equations of fractional order". In: *Izv. Akad. Nauk Armjan. SSR Ser. Mat.* 3.1, pp. 3–29. ISSN: 0002-3043.
- Garrappa, R. (2015). "Numerical evaluation of two and three parameter Mittag-Leffler functions". In: *SIAM J. Numer. Anal.* 53.3, pp. 1350–1369. ISSN: 0036-1429. DOI: 10.1137/140971191. URL: https://doi.org/10.1137/140971191.
- Gerasimov, A. N. (1948). "A generalization of linear laws of deformation and its application to problems of internal friction". In: *Akad. Nauk SSSR. Prikl. Mat. Meh.* 12, pp. 251–260.
- Gross, B. (1947). "On creep and relaxation". In: *J. Appl. Phys.* 18, pp. 212–221. ISSN: 0021-8979.
- Johnson, W. P. (2002). "The curious history of Faà di Bruno's formula". In: *Amer. Math. Monthly* 109.3, pp. 217–234. ISSN: 0002-9890. DOI: 10.2307/2695352. URL: https://doi.org/10.2307/2695352.

Bibliography III

- Linz, P. (1985). *Analytical and numerical methods for Volterra equations*. Vol. 7. SIAM Studies in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, pp. xiii+227. ISBN: 0-89871-198-3. DOI: 10.1137/1.9781611970852. URL: https://doi.org/10.1137/1.9781611970852.
- Liouville, J. (1832). *Mémoire sur quelques questions de géométrie et de mécanique, et sur un nouveau genre de calcul pour résoudre ces questions*.
- Rabotnov, I. N. et al. (1969). *Creep problems in structural members*. Vol. 7. North-Holland Publishing Company.
- Trefethen, L. N., J. A. C. Weideman, and T. Schmelzer (2006). "Talbot quadratures and rational approximations". In: *BIT* 46.3, pp. 653–670. ISSN: 0006-3835. DOI: 10.1007/s10543-006-0077-9. URL: https://doi.org/10.1007/s10543-006-0077-9.
- Trefethen, L. N. and J. A. C. Weideman (2014). "The exponentially convergent trapezoidal rule". In: *SIAM Rev.* 56.3, pp. 385–458. ISSN: 0036-1445. DOI: 10.1137/130932132. URL: https://doi.org/10.1137/130932132.

Bibliography IV

Weideman, J. A. C. (2006). "Optimizing Talbot's contours for the inversion of the Laplace transform". In: *SIAM J. Numer. Anal.* 44.6, pp. 2342–2362. ISSN: 0036-1429. DOI: 10.1137/050625837. URL: https://doi.org/10.1137/050625837.

Weideman, J. A. C. and L. N. Trefethen (2007). "Parabolic and hyperbolic contours for computing the Bromwich integral". In: *Math. Comp.* 76.259, pp. 1341–1356. ISSN: 0025-5718. DOI: 10.1090/S0025-5718-07-01945-X. URL: https://doi.org/10.1090/S0025-5718-07-01945-X.

An introduction to fractional calculus

Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

May, 2022

The Numerical Integration of FODEs

We want to find a **numerical solution** of the differential equation written in terms of Caputo Derivatives

$$\alpha > 0, \quad m = \lceil \alpha \rceil, \qquad \begin{cases} {}_C D^{\alpha}_{[0,t]} \mathbf{y}(t) = f(t, \mathbf{y}(t)), & t \in [0, T], \\ \dfrac{d^k \mathbf{y}(0)}{dt^k} = \mathbf{y}_0^{(k)}, & k = 0, 1, \ldots, m-1. \end{cases} \tag{FODE}$$

Caputo fractional derivative (Caputo 2008)

Let $\alpha \geq 0$, and $m = \lceil \alpha \rceil$. Then, we define the operator

$$ {}_C D^{\alpha}_{[a,t]} y = I^{m-\alpha}_{[a,t]} \frac{d^m}{dt^m} y,$$

whenever $\frac{d^m}{dt^m} y \in \mathbb{L}^1([a, b])$.

The Numerical Integration of ODEs

What methods do we know for ODEs?

The Numerical Integration of ODEs

What methods do we know for ODEs?
Given a grid $\{t_j = j\tau\}_{j=1}^{N}$ and $\tau = T/N$, approximating $\mathbf{y}(t_j) \approx y^{(j)}$.

The Numerical Integration of ODEs

What methods do we know for ODEs?
Given a grid $\{t_j = j\tau\}_{j=1}^{N}$ and $\tau = T/N$, approximating $\mathbf{y}(t_j) \approx y^{(j)}$.

Explicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k < j$

The Numerical Integration of ODEs

What methods do we know for ODEs?
Given a grid $\{t_j = j\tau\}_{j=1}^{N}$ and $\tau = T/N$, approximating $\mathbf{y}(t_j) \approx y^{(j)}$.

Explicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k < j$

Implicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k \leq j$

The Numerical Integration of ODEs

What methods do we know for ODEs?
Given a grid $\{t_j = j\tau\}_{j=1}^{N}$ and $\tau = T/N$, approximating $\mathbf{y}(t_j) \approx y^{(j)}$.

Explicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k < j$

Implicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k \leq j$

Exponential integrators: Compute directly $y^{(N)}$ without any $y^{(j)}$ for $j < N$.

The Numerical Integration of ODEs

What methods do we know for ODEs?
Given a grid $\{t_j = j\tau\}_{j=1}^{N}$ and $\tau = T/N$, approximating $\mathbf{y}(t_j) \approx y^{(j)}$.

Explicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k < j$

Implicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k \leq j$

Exponential integrators: Compute directly $y^{(N)}$ without any $y^{(j)}$ for $j < N$.

The Numerical Integration of ODEs

What methods do we know for ODEs?
Given a grid $\{t_j = j\tau\}_{j=1}^{N}$ and $\tau = T/N$, approximating $\mathbf{y}(t_j) \approx y^{(j)}$.

Explicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k < j$

- One Step Methods: Explicit Runge-Kutta Methods (ERK)
- Linear Multistep Methods: Adams–Bashforth, Predictor-Corrector

Implicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k \leq j$

Exponential integrators: Compute directly $y^{(N)}$ without any $y^{(j)}$ for $j < N$.

The Numerical Integration of ODEs

What methods do we know for ODEs?
Given a grid $\{t_j = j\tau\}_{j=1}^{N}$ and $\tau = T/N$, approximating $\mathbf{y}(t_j) \approx y^{(j)}$.

Explicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k < j$

- One Step Methods: Explicit Runge-Kutta Methods (ERK)
- Linear Multistep Methods: Adams–Bashforth, Predictor-Corrector

Implicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k \leq j$

- One Step Methods: Implicit Runge-Kutta Methods (IRK,DIRK,SDIRK)
- Linear Multistep Methods: Adams–Moulton, Backward Differentiation Formulas (BDFs), Numerical Differentiation Formulas (NDFs),...

Exponential integrators: Compute directly $y^{(N)}$ without any $y^{(j)}$ for $j < N$.

The Numerical Integration of ODEs

What methods do we know for ODEs?
Given a grid $\{t_j = j\tau\}_{j=1}^{N}$ and $\tau = T/N$, approximating $\mathbf{y}(t_j) \approx y^{(j)}$.

Explicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k < j$

- One Step Methods: Explicit Runge-Kutta Methods (ERK)
- Linear Multistep Methods: Adams–Bashforth, Predictor-Corrector

Implicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k \leq j$

- One Step Methods: Implicit Runge-Kutta Methods (IRK,DIRK,SDIRK)
- Linear Multistep Methods: Adams–Moulton, Backward Differentiation Formulas (BDFs), Numerical Differentiation Formulas (NDFs),...

Exponential integrators: Compute directly $y^{(N)}$ without any $y^{(j)}$ for $j < N$.

For both *implicit* and *explicit* methods we have also all-at-once formulations, and a middle-ground represented by Implicit-Explicit (IMEX) methods.

The Numerical Integration of ODEs

What methods do we know for ODEs?
Given a grid $\{t_j = j\tau\}_{j=1}^{N}$ and $\tau = T/N$, approximating $\mathbf{y}(t_j) \approx y^{(j)}$.

Explicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k < j$

- One Step Methods: Explicit Runge-Kutta Methods (ERK)
- Linear Multistep Methods: Adams–Bashforth, Predictor-Corrector

Implicit methods: Compute $y^{(j)}$ using only values $y^{(k)}$ for $k \leq j$

- One Step Methods: Implicit Runge-Kutta Methods (IRK,DIRK,SDIRK)
- Linear Multistep Methods: Adams–Moulton, Backward Differentiation Formulas (BDFs), Numerical Differentiation Formulas (NDFs),...

Exponential integrators: Compute directly $y^{(N)}$ without any $y^{(j)}$ for $j < N$.

For both *implicit* and *explicit* methods we have also all-at-once formulations, and a middle-ground represented by Implicit-Explicit (IMEX) methods.
Our *objective* is to transport what we can for the solution of (FODE).

Product Integration Rules

Product **I**ntegration rules were introduced in the work (Young 1954) for Integral Equations.

Product Integration Rules

Product **I**ntegration rules were introduced in the work (Young 1954) for Integral Equations. From the *existence results* we know that a solution to (FODE) is a solution to the Integral Equation

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

Product Integration Rules

Product **I**ntegration rules were introduced in the work (Young 1954) for Integral Equations. From the *existence results* we know that a solution to (FODE) is a solution to the Integral Equation

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

- Adams-Bashforth-Moulton methods are obtained by applying a *quadrature formula to the integral*,

Product Integration Rules

Product **I**ntegration rules were introduced in the work (Young 1954) for Integral Equations. From the *existence results* we know that a solution to (FODE) is a solution to the Integral Equation

$$y(t) = \sum_{k=0}^{m-1} \frac{t^k}{k!} y_0^{(k)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau, \quad m = \lceil \alpha \rceil.$$

- Adams-Bashforth-Moulton methods are obtained by applying a *quadrature formula to the integral*,
- We can use, e.g.,
  - the **fractional rectangular formula** with nodes $\{t_j = j\tau\}_{j=1}^{n-1}$,
  - or the **product trapezoidal quadrature formula** with nodes $\{t_j = j\tau\}_{j=1}^{n}$.

  To obtain a predictor-corrector method.

Product Integral Rules

The main idea behind PI rules is to approximate the integral

$$\int_0^t (t-\tau)^{\alpha-1} f(s, y(s))\,\mathrm{d}s$$

by approximating the vector field f with suitable polynomials..

Product Integral Rules

The main idea behind PI rules is to approximate the integral

$$\int_0^t (t-\tau)^{\alpha-1} f(s, y(s))\,\mathrm{d}s$$

by approximating the vector field f with suitable polynomials. We build an Adams-type method. If we consider a grid $\{t_0, t_1, \ldots, t_N\}$ on the whole $[t_0, T]$ we can decompose the integral as

$$y(t) = T_{m-1}(t) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} (t-s)^{\alpha-1} f(\tau, y(\tau))\,\mathrm{d}\tau, \qquad t \geq t_n.$$

Product Integral Rules

The main idea behind PI rules is to approximate the integral

$$\int_0^t (t-\tau)^{\alpha-1} f(s, y(s))\,\mathrm{d}s$$

by approximating the vector field f with suitable polynomials. We build an Adams-type method. If we consider a grid $\{t_0, t_1, \ldots, t_N\}$ on the whole $[t_0, T]$ we can decompose the integral as

$$y(t) = T_{m-1}(t) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} (t-s)^{\alpha-1} f(\tau, y(\tau))\,\mathrm{d}\tau, \qquad t \geq t_n.$$

- Replace f in each sub-interval by the first–degree polynomial interpolant

$$p_j(\tau) = f_{j+1} + \frac{s - t_{j+1}}{\tau_j}, \quad s \in [t_j, tj+1], \quad \tau_j = t_{j+1} - t_j, \quad f_j = f(t_j, y_j).$$

Product Integral Rules

The main idea behind PI rules is to approximate the integral

$$\int_0^t (t-\tau)^{\alpha-1} f(s, y(s))\,\mathrm{d}s$$

by approximating the vector field f with suitable polynomials. We build an Adams-type method. If we consider a grid $\{t_0, t_1, \ldots, t_N\}$ on the whole $[t_0, T]$ we can decompose the integral as

$$y(t) = T_{m-1}(t) + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} (t-s)^{\alpha-1} f(\tau, y(\tau))\,\mathrm{d}\tau, \qquad t \geq t_n.$$

- Replace f in each sub-interval by the first–degree polynomial interpolant
- These produce the usual fractional integral that we now how to solve

$$I_{n,j}^{(k)} = \frac{1}{\Gamma(\alpha)} \int_{t_j}^{t_n} (t_n-\tau)^{\alpha-1} (\tau - t_j)^k\,\mathrm{d}\tau = \frac{(t_n-t_j)^{\alpha+k}}{\Gamma(\alpha+k+1)}.$$

Product Integral Rules

The main idea behind PI rules is to approximate the integral

$$\int_0^t (t-\tau)^{\alpha-1} f(s, y(s))\, \mathrm{d}s$$

by approximating the vector field f with suitable polynomials. We build an Adams-type method. If we consider a grid $\{t_0, t_1, \ldots, t_N\}$ on the whole $[t_0, T]$ we can decompose the integral as

$$y^{(n)} = T_{m-1}(t_n) + w_n f^{(0)} + \sum_{j=1}^{n} b_{n,j} f^{(j)},$$

- Replace f in each sub-interval by the first–degree polynomial interpolant
- These produce the usual fractional integral that we now how to solve
- We plug everything in our expression using that:

$$w_n = I_{n,0}^{(0)} - \frac{I_{n,0}^{(1)}}{\tau_0} + \frac{I_{n,1}^{(1)}}{\tau_0}, \quad b_{n_j} = \frac{I_{n,j-1}^{(1)} - I_{n,j}^{(1)}}{\tau_{j-1}} - \frac{I_{n,j}^{(1)} - I_{n,j+1}^{(1)}}{\tau_j}, j \leq n-1, \quad b_{n,n} = \frac{I_{n,n-1}^{(1)}}{\tau_{n-1}}.$$

Product Integral Rules - Convergence

To discuss **convergence properties** we can piggyback on the theory of Abel's and Volterra's fractional integral equations.

Product Integral Rules - Convergence

To discuss **convergence properties** we can piggyback on the theory of Abel's and Volterra's fractional integral equations.

$$\int_0^t (t-s)^{-\alpha} K(t,s) y(s)\,\mathrm{d}s = f(t), \quad 0 < \alpha < 1 \qquad \text{(Volterra's Integral Eq.)}$$

Product Integral Rules - Convergence

To discuss **convergence properties** we can piggyback on the theory of Abel's and Volterra's fractional integral equations.

$$\int_0^t (t-s)^{-\alpha} y(s)\,\mathrm{d}s = f(t), \quad 0 < \alpha < 1 \qquad \text{(Abel's Integral Eq.)}$$

If we discretize everything as before we get

$$[B_N \odot K_N]\mathbf{y} = \mathbf{g}, \quad B_N = \tau^{1-\alpha}[b_{i,j}], \quad K_N = [k(t_i, t_j)], \quad \odot \text{ Hadamard product.}$$

where $\mathbf{y} = (y_0, \ldots, y_N)^T$ and g contains the **initial conditions** and the **evaluations** of f.

Convergence analysis for (Cameron and McKee 1985)

"[Consistency of order p] demands that $f(t) \in \mathcal{C}^{1-\alpha}[0, T]$ which is necessary in any case for $y(t)$ to be a smooth function ... $|y(t_i) - y_i| \leq C\tau^p$, $i = 0, 1, \ldots, m-1$."

Product Integral Rules - Convergence

The requirements from the standard theory are **far too strong** for what we can reasonably expect from the analysis on the solution regularity we did in the last lecture.

Theorem (Dixon 1985)

Let f be Lipschitz continuous with respect to the second variable and y_n be the numerical approximation obtained by applying the PI trapezoidal rule on the interval $[t_0, T]$. There exist a constant $C = C_1(T - t_0)$, which does not depend on h, such that

$$\|y(t_n) - y_n\| \leq C(t_n^{\alpha-1}\tau^{1+\alpha} + \tau^2), \qquad \tau = \max_{j=0,\dots,n-1} \tau_j.$$

Product Integral Rules - Convergence

The requirements from the standard theory are **far too strong** for what we can reasonably expect from the analysis on the solution regularity we did in the last lecture.

Theorem (Dixon 1985)

Let f be Lipschitz continuous with respect to the second variable and y_n be the numerical approximation obtained by applying the PI trapezoidal rule on the interval $[t_0, T]$. There exist a constant $C = C_1(T - t_0)$, which does not depend on h, such that

$$\|y(t_n) - y_n\| \leq C(t_n^{\alpha-1}\tau^{1+\alpha} + \tau^2), \qquad \tau = \max_{j=0,\dots,n-1} \tau_j.$$

- The same drop in the convergence order occurs also when higher degree polynomials are employed,

Product Integral Rules - Convergence

The requirements from the standard theory are **far too strong** for what we can reasonably expect from the analysis on the solution regularity we did in the last lecture.

Theorem (Dixon 1985)

Let f be Lipschitz continuous with respect to the second variable and y_n be the numerical approximation obtained by applying the PI trapezoidal rule on the interval $[t_0, T]$. There exist a constant $C = C_1(T - t_0)$, which does not depend on h, such that

$$\|y(t_n) - y_n\| \leq C(t_n^{\alpha-1}\tau^{1+\alpha} + \tau^2), \qquad \tau = \max_{j=0,\dots,n-1} \tau_j.$$

- The same drop in the convergence order occurs also when higher degree polynomials are employed,
- When $\alpha > 1$ convergence order 2 is obtained.

Product Integral Rules - Convergence

The requirements from the standard theory are **far too strong** for what we can reasonably expect from the analysis on the solution regularity we did in the last lecture.

Theorem (Dixon 1985)

Let f be Lipschitz continuous with respect to the second variable and y_n be the numerical approximation obtained by applying the PI trapezoidal rule on the interval $[t_0, T]$. There exist a constant $C = C_1(T - t_0)$, which does not depend on h, such that

$$\|y(t_n) - y_n\| \leq C(t_n^{\alpha-1}\tau^{1+\alpha} + \tau^2), \qquad \tau = \max_{j=0,\dots,n-1} \tau_j.$$

- The same drop in the convergence order occurs also when higher degree polynomials are employed,
- When $\alpha > 1$ convergence order 2 is obtained.

It doesn't make much sense to use higher-degree PI rules if $0 < \alpha < 1$.

The Fractional Rectangular Formula

Let us reduce to the case with $\alpha \in (0,1)$, $m = 1$, and a *uniform mesh*.
To build it we need to *approximate the integral* with the rectangule rule

$$\int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\,\mathrm{d}\tau$$

on the grid $\{t_j = t_0 + j\tau\}_{j=1}^N$ with *uniform* grid spacing τ, we denote

$$f^{(j)} = f(t_j, y^{(j)}) \text{ for } y^{(j)} \approx y(t_j),$$

and write it as

$$y^{(n)} = y_0 + \frac{\tau^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n-1} b_{n-j-1} f^{(j)}, \quad b_n = [(n+1)^\alpha - n^\alpha]/\alpha, \quad n = 1, \ldots, N.$$

The Fractional Rectangular Formula

Let us reduce to the case with $\alpha \in (0,1)$, $m = 1$, and a *uniform mesh*.
To build it we need to *approximate the integral* with the rectangule rule

$$\int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\,\mathrm{d}\tau$$

on the grid $\{t_j = t_0 + j\tau\}_{j=1}^N$ with *uniform* grid spacing τ, we denote

$$f^{(j)} = f(t_j, y^{(j)}) \text{ for } y^{(j)} \approx y(t_j),$$

and write it as

$$y^{(n)} = y_0 + \frac{\tau^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n-1} b_{n-j-1} f^{(j)}, \quad b_n = [(n+1)^\alpha - n^\alpha]/\alpha, \quad n = 1, \ldots, N.$$

- This is an explicit method,

The Fractional Rectangular Formula

Let us reduce to the case with $\alpha \in (0,1)$, $m = 1$, and a *uniform mesh*.
To build it we need to *approximate the integral* with the rectangule rule

$$\int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau$$

on the grid $\{t_j = t_0 + j\tau\}_{j=1}^N$ with *uniform* grid spacing τ, we denote

$$f^{(j)} = f(t_j, y^{(j)}) \text{ for } y^{(j)} \approx y(t_j),$$

and write it as

$$y^{(n)} = y_0 + \frac{\tau^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n-1} b_{n-j-1} f^{(j)}, \quad b_n = [(n+1)^\alpha - n^\alpha]/\alpha, \quad n = 1, \ldots, N.$$

- This is an explicit method,
- By construction, this is a 1-step method...

The Fractional Rectangular Formula

Let us reduce to the case with $\alpha \in (0,1)$, $m = 1$, and a *uniform mesh*.
To build it we need to *approximate the integral* with the rectangule rule

$$\int_0^t (t-\tau)^{\alpha-1} f(\tau, y(\tau))\, \mathrm{d}\tau$$

on the grid $\{t_j = t_0 + j\tau\}_{j=1}^N$ with *uniform* grid spacing τ, we denote

$$f^{(j)} = f(t_j, y^{(j)}) \text{ for } y^{(j)} \approx y(t_j),$$

and write it as

$$y^{(n)} = y_0 + \frac{\tau^\alpha}{\Gamma(\alpha)} \left(b_0 f^{(n-1)} + \sum_{j=0}^{n-2} b_{n-j-1} f^{(j)} \right) \qquad b_n = [(n+1)^\alpha - n^\alpha]/\alpha, \quad n = 1, \ldots, N.$$

- This is an explicit method,
- By construction, this is a 1-step method... but in reality we need all the previous steps!

Some observations

- To build the solution we have to keep in memory either the **previous solutions** or the **function evaluations**,

Some observations

- To build the solution we have to keep in memory either the **previous solutions** or the **function evaluations**,
- Using a uniform mesh the evaluation of the weights just involve the computation of real powers of integer numbers, we can simplify also the fractional trapezoidal formula

$$
\begin{aligned}
y^{(n)} =& T_{m-1}(t_n) + \frac{\tau^\alpha}{\Gamma(\alpha+2)}\left(w_n f^{(0)} + \sum_{j=1}^{n} b_{n-j} f^{(j)}\right), \\
w_n =& (\alpha+1-n)n^\alpha + (n-1)^{\alpha+1}, \\
b_0 =& 1,\ b_n = (n-1)^{\alpha+1} - 2n^{\alpha+1} + (n+1)^{\alpha+1},\ n \geq 1.
\end{aligned}
$$

Some observations

- To build the solution we have to keep in memory either the **previous solutions** or the **function evaluations**,
- Using a uniform mesh the evaluation of the weights just involve the computation of real powers of integer numbers,
- We have to compute:

$$\sum_{j=0}^{n-2} b_{n-j-1} f^{(j)},$$

this is a quadratic cost with N, but there is a **convolution structure**, so we can expect that some FFT-based trick could come to the rescue.

Some observations

- To build the solution we have to keep in memory either the **previous solutions** or the **function evaluations**,
- Using a uniform mesh the evaluation of the weights just involve the computation of real powers of integer numbers,
- We have to compute:

$$\sum_{j=0}^{n-2} b_{n-j-1} f^{(j)},$$

  this is a quadratic cost with N, but there is a **convolution structure**, so we can expect that some FFT-based trick could come to the rescue.
- The 1-step name is related to the number of initial values to start the computation.

Some observations

- To build the solution we have to keep in memory either the **previous solutions** or the **function evaluations**,
- Using a uniform mesh the evaluation of the weights just involve the computation of real powers of integer numbers,
- We have to compute:

$$\sum_{j=0}^{n-2} b_{n-j-1} f^{(j)},$$

  this is a quadratic cost with N, but there is a **convolution structure**, so we can expect that some FFT-based trick could come to the rescue.
- The 1-step name is related to the number of initial values to start the computation.

Predictor-Corrector algorithms

Now that we have two schemes we can think of using them together to build a **predictor-corrector** algorithm.

Fractional Predictor-Corrector Scheme (Diethelm 1997)

We are going to write it again for $0 < \alpha < 1$ on a uniform mesh

1. In the *prediction step* we use the fractional rectangular formula

$$y_P^{(n+1)} = y^{(0)} + \frac{\tau^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} b_{j,n+1} f(t_j, y^{(j)}), \quad b_{j,n+1} = \frac{(n+1-j)^\alpha - (n-j)^\alpha}{\alpha}$$

Fractional Predictor-Corrector Scheme (Diethelm 1997)

We are going to write it again for $0 < \alpha < 1$ on a uniform mesh

1. In the *prediction step* we use the fractional rectangular formula

$$y_P^{(n+1)} = y^{(0)} + \frac{\tau^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} b_{j,n+1} f(t_j, y^{(j)}), \quad b_{j,n+1} = \frac{(n+1-j)^\alpha - (n-j)^\alpha}{\alpha}$$

2. In the *correction step* we use the fractional trapezoidal formula

$$y^{(n+1)} = y^{(0)} + \frac{\tau^\alpha}{\Gamma(\alpha)} \left(\sum_{j=0}^{n} a_{j,n+1} f(t_j, y^{(j)}) + a_{n+1,n+1} f(t_{n+1}, y_P^{(n+1)}) \right)$$

where

$$a_{j,n+1} = \begin{cases} {(n^{\alpha+1} - (n-\alpha)(n+1)^\alpha}/{\alpha(\alpha+1)}, & j = 0, \\ {(n-j+2)^{\alpha+1} - 2(n-j+1)^{\alpha+1} + (n-j)^{\alpha+1}}/{\alpha(\alpha+1)}, & j = 1, 2, \ldots, n, \\ {1}/{\alpha(\alpha+1)}, & j = n+1. \end{cases}$$

A Fractional Predictor-Corrector Scheme

- Predictor-Corrector schemes are of interest because they represent a good **compromise** between **accuracy** and **ease of implementation**.

A Fractional Predictor-Corrector Scheme

- Predictor-Corrector schemes are of interest because they represent a good **compromise** between **accuracy** and **ease of implementation**.
- To investigate the convergence we need to look deeper into the convergence results of the two PI integral rules (Diethelm, Ford, and Freed 2004).

Theorem (Diethelm, Ford, and Freed 2004, Theorem 2.4)

(a) Let $z \in \mathcal{C}^1([0,T])$. Then

$$\left| \int_0^{t_{k+1}} (t_{k+1}-t)^{\alpha-1} z(t)\,\mathrm{d}t - \sum_{j=0}^{k} b_{j,k+1} z(t_j) \right| \leq \frac{1}{\alpha} \|z'\|_\infty t_{k+1}^{\alpha} \tau.$$

(b) Let $z(t) = t^p$ for some $p \in (0,1)$. Then,

$$\left| \int_0^{t_{k+1}} (t_{k+1}-t)^{\alpha-1} z(t)\,\mathrm{d}t - \sum_{j=0}^{k} b_{j,k+1} z(t_j) \right| \leq C_{\alpha,p}^{Re} t_{k+1}^{\alpha+p-1} \tau.$$

A Fractional Predictor-Corrector Scheme

And analogously for the product trapezoidal formula.

Theorem (Diethelm, Ford, and Freed 2004, Theorem 2.5).

(a) If $z \in \mathcal{C}^2([0,T])$, then there exist a constant C_α^{Tr} depending only on α such that

$$\left| \int_0^{t_{k+1}} (t_{k+1}-t)^{\alpha-1} z(t)\,\mathrm{d}t - \sum_{j=0}^{k+1} a_{j,k+1} z(t_j) \right| \leq C_\alpha^{Tr} \|z''\|_\infty t_{k+1}^\alpha \tau^2.$$

(b) Let $z \in \mathcal{C}^1([0,T])$ and assume that z' fulfills a Lipschitz condition of order $\mu \in (0,1)$. Then, there exists positive constants $B_{\alpha,\mu}^{Tr}$ and $M_{z,\mu}$ such that

$$\left| \int_0^{t_{k+1}} (t_{k+1}-t)^{\alpha-1} z(t)\,\mathrm{d}t - \sum_{j=0}^{k+1} a_{j,k+1} z(t_j) \right| \leq B_{\alpha,\mu}^{Tr} M_{z,\mu} t_{k+1}^\alpha \tau^{1+\mu}.$$

A Fractional Predictor-Corrector Scheme

And analogously for the product trapezoidal formula.

Theorem (Diethelm, Ford, and Freed 2004, Theorem 2.5).

(a) Let $z \in \mathcal{C}^1([0,T])$ and assume that z' fulfills a Lipschitz condition of order $\mu \in (0,1)$. Then, there exists positive constants $B^{Tr}_{\alpha,\mu}$ and $M_{z,\mu}$ such that

$$\left| \int_0^{t_{k+1}} (t_{k+1}-t)^{\alpha-1} z(t)\,\mathrm{d}t - \sum_{j=0}^{k+1} a_{j,k+1} z(t_j) \right| \leq B^{Tr}_{\alpha,\mu} M_{z,\mu} t_{k+1}^{\alpha} \tau^{1+\mu}.$$

(c) Let $z(t) = t^p$ for some $p \in (0,2)$ and $\rho = \min(2, p+1)$. Then

$$\left| \int_0^{t_{k+1}} (t_{k+1}-t)^{\alpha-1} z(t)\,\mathrm{d}t - \sum_{j=0}^{k+1} a_{j,k+1} z(t_j) \right| \leq C^{Tr}_{\alpha,p} t_{k+1}^{\alpha+p-\rho} \tau^{\rho}.$$

A Fractional Predictor-Corrector Scheme

Observe that for the fractional rectangular case (b) the bound contains

$$t_{k+1}^{\alpha+p-1},$$

if $\alpha + p < 1$ then we get that the overall integration error becomes larger if the size of the interval of integration becomes smaller!
Similarly for the case (c) for the fractional trapezoidal rule

$$\alpha < 1,\ p < 1,\ \rho = p + 1, \quad t_{k+1}^{\alpha+p-\rho},$$

has the same explosive behavior.

Smaller intervals for harder integrals

By making t_{k+1} smaller we have two effects

1. We reduce the length of the integration interval,
2. We change the weight function in a way that makes the integral more difficult.

A Fractional Predictor-Corrector Scheme

Lemma (Diethelm, Ford, and Freed 2004, Lemma 3.1)

Assume that the solution y of the initial value problem is such that

$$\left| \int_0^{t_{k+1}} (t_{k+1} - t)^{\alpha-1} {}_{CA}D^{\alpha}_{[0,t]} y(t)\,\mathrm{d}t - \sum_{j=0}^{k} b_{j,k+1}\, {}_{CA}D^{\alpha}_{[0,t]} y(t) \right| \leq C_1 t_{k+1}^{\gamma_1} \tau^{\delta_1},$$

and

$$\left| \int_0^{t_{k+1}} (t_{k+1} - t)^{\alpha-1} {}_{CA}D^{\alpha}_{[0,t]} y(t)\,\mathrm{d}t - \sum_{j=0}^{k+1} a_{j,k+1}\, {}_{CA}D^{\alpha}_{[0,t]} y(t) \right| \leq C_2 t_{k+1}^{\gamma_2} \tau^{\delta_2},$$

with some $\gamma_1, \gamma_2 \geq 0$ and $\delta_1, \delta_2 > 0$. Then, for some suitably chosen $T > 0$, we have

$$\max_{0 \leq j \leq N} |y(t_j) - y^{(j)}| = O(\tau^q), \quad q = \min\{\delta_1 + \alpha, \delta_2\}, \quad N = \lceil T/\tau \rceil.$$

Error bounds

Theorem (Diethelm, Ford, and Freed 2004, Theorem 3.2)

Let $0 < \alpha$ and assume ${}_{CA}D^{\alpha}_{[0,t]}y(t) \in \mathcal{C}^2([O, T])$ for some suitable T. Then,

$$\max_{0 \le j \le N} |y(t_j) - y^{(j)}| = \begin{cases} O(\tau^2), & \text{if } \alpha \ge 1, \\ O(\tau^{1+\alpha}), & \text{if } \alpha < 1. \end{cases}$$

Proof. In view of the two bounds for the Fractional Rectangular and Trapezoidal forms we can apply the previous Lemma with $\gamma_1 = \gamma_2 = \alpha > 0$, $\delta_1 = 1$, $\delta_2 = 2$. Therefore we find a bound of order $O(\tau^q)$ where

$$q = \min\{1 + \alpha, 2\} = \begin{cases} 2, & \text{if } \alpha \ge 1, \\ 1 + \alpha, & \text{if } \alpha < 1. \end{cases}$$

Error bounds

Theorem (Diethelm, Ford, and Freed 2004, Theorem 3.2)

Let $0 < \alpha$ and assume ${}_{CA}D^{\alpha}_{[0,t]}y(t) \in \mathcal{C}^2([O, T])$ for some suitable T. Then,

$$\max_{0 \le j \le N} |y(t_j) - y^{(j)}| = \begin{cases} O(\tau^2), & \text{if } \alpha \ge 1, \\ O(\tau^{1+\alpha}), & \text{if } \alpha < 1. \end{cases}$$

- Order of convergence is a non-decreasing function of α,
- Hypotheses are stated in terms of the αth Caputo derivative of the solution,

Error bounds

Theorem (Diethelm, Ford, and Freed 2004, Theorem 3.2)

Let $0 < \alpha$ and assume ${}_{CA}D^{\alpha}_{[0,t]}y(t) \in \mathcal{C}^2([O, T])$ for some suitable T. Then,

$$\max_{0 \le j \le N} |y(t_j) - y^{(j)}| = \begin{cases} O(\tau^2), & \text{if } \alpha \ge 1, \\ O(\tau^{1+\alpha}), & \text{if } \alpha < 1. \end{cases}$$

- Order of convergence is a non-decreasing function of α,
- Hypotheses are stated in terms of the αth Caputo derivative of the solution,
- Can we replace them by similar assumptions on y itself?

Theorem Diethelm, Ford, and Freed 2004, Theorem 3.3

Let $\alpha > 1$ and assume $y \in \mathcal{C}^{1+\lceil\alpha\rceil}([0, T])$ for some suitable T, then

$$\max_{0 \le j \le N} |y(t_j) - y^{(j)}| = O(\tau^{1+\lceil\alpha\rceil-\alpha}).$$

Error bounds

Theorem (Diethelm, Ford, and Freed 2004, Theorem 3.2)

Let $0 < \alpha$ and assume ${}_{CA}D^{\alpha}_{[0,t]}y(t) \in \mathcal{C}^2([O,T])$ for some suitable T. Then,

$$\max_{0\le j\le N} |y(t_j) - y^{(j)}| = \begin{cases} O(\tau^2), & \text{if } \alpha \ge 1, \\ O(\tau^{1+\alpha}), & \text{if } \alpha < 1. \end{cases}$$

Theorem Diethelm, Ford, and Freed 2004, Theorem 3.3

Let $\alpha > 1$ and assume $y \in \mathcal{C}^{1+\lceil\alpha\rceil}([0,T])$ for some suitable T, then

$$\max_{0\le j\le N} |y(t_j) - y^{(j)}| = O(\tau^{1+\lceil\alpha\rceil-\alpha}).$$

Proof. We need to use the characterization of Caputo's derivative

$${}_{CA}D^{\alpha}_{[0,t]}y(t) = \sum_{\ell=0}^{m-\lceil\alpha\rceil-1} \frac{y^{(l+\lceil\alpha\rceil)}(0)}{\Gamma(\lceil\alpha\rceil-\alpha+\ell+1)} t^{\lceil\alpha\rceil-\alpha+\ell} + g(t), \qquad \begin{array}{l} g \in \mathcal{C}^{m-\lceil\alpha\rceil}([O,T]), \\ g^{(m-\lceil\alpha\rceil)} \in \mathrm{Lip}(\lceil\alpha\rceil-\alpha). \end{array}$$

Error bounds

Theorem (Diethelm, Ford, and Freed 2004, Theorem 3.2)

Let $0 < \alpha$ and assume ${}_{CA}D^{\alpha}_{[0,t]}y(t) \in \mathcal{C}^2([O, T])$ for some suitable T. Then,

$$\max_{0\le j\le N} |y(t_j) - y^{(j)}| = \begin{cases} O(\tau^2), & \text{if } \alpha \ge 1, \\ O(\tau^{1+\alpha}), & \text{if } \alpha < 1. \end{cases}$$

Theorem Diethelm, Ford, and Freed 2004, Theorem 3.3

Let $\alpha > 1$ and assume $y \in \mathcal{C}^{1+\lceil\alpha\rceil}([0, T])$ for some suitable T, then

$$\max_{0\le j\le N} |y(t_j) - y^{(j)}| = O(\tau^{1+\lceil\alpha\rceil-\alpha}).$$

Proof. Then for $\alpha > 1$, we can apply the Lemma with $\gamma_1 = 0$, $\gamma_2 = \alpha - 1 > 0$, $\delta_1 = 1$, $\delta_2 = 1 + \lceil\alpha\rceil - \alpha$ and thus $\delta_1 + \alpha = 1 + \alpha > 2 > \delta_2$, $\min\{\delta_1 + \alpha, \delta_2\} = \delta_2$. The overall order is then $O(\tau^{\delta_2}) = O(\tau^{1+\lceil\alpha\rceil-\alpha})$.

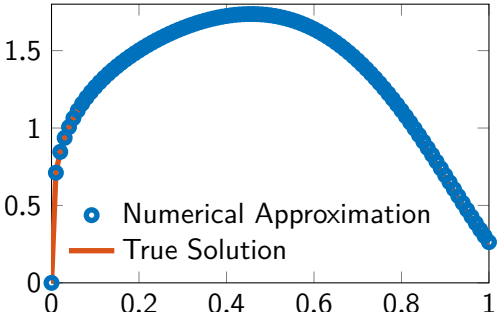
An example

Example

$$\begin{cases} {}_{CA}D^{\alpha}_{[0,t]}y(t) = \frac{40320}{\Gamma(9-\alpha)}t^{8-\alpha} - 3\frac{\Gamma(5+\alpha/2)}{\Gamma(5-\alpha/2)}t^{4-\alpha/2} + \frac{9}{4}\Gamma(\alpha+1) + \left(3t^{\alpha/2}/2 - t^4\right)^3 - y(t)^{3/2}, \\ y(0) = 0. \end{cases}$$

Solution: $y(t) = t^8 - 3t^{4+\alpha/2} + \frac{9}{4}t^{\alpha}$.

```
tauval = 2.^(-(1:6));
for i=1:length(hval)
 tau  = tauval(i);
 t0 = 0; T = 1;
 alpha   = 0.25;
 [T, Y] = fde_pi1_ex(alpha, f_fun, t0,
 ↪  T, y0, tau);
 err(i) = norm(Y - ye(T),'inf');
end
```



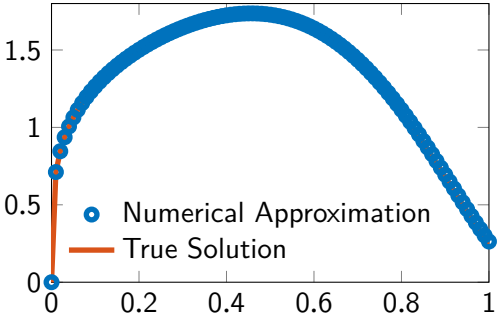
An example

Example

$$\begin{cases} {}_{CA}D^{\alpha}_{[0,t]}y(t) = \frac{40320}{\Gamma(9-\alpha)}t^{8-\alpha} - 3\frac{\Gamma(5+\alpha/2)}{\Gamma(5-\alpha/2)}t^{4-\alpha/2} + \frac{9}{4}\Gamma(\alpha+1) + \left(3t^{\alpha/2}/2 - t^4\right)^3 - y(t)^{3/2}, \\ y(0) = 0. \end{cases}$$

Solution: $y(t) = t^8 - 3t^{4+\alpha/2} + \frac{9}{4}t^{\alpha}$.

α	τ	E	q
0.25	5.00e-01	1.42e+00	
	2.50e-01	4.17e-01	1.77
	1.25e-01	2.13e-01	0.97
	6.25e-02	1.03e-01	1.05
	3.12e-02	5.04e-02	1.03
	1.56e-02	2.44e-02	1.05



An example

Example

$$\begin{cases} {}_{CA}D^{\alpha}_{[0,t]}y(t) = \frac{40320}{\Gamma(9-\alpha)}t^{8-\alpha} - 3\frac{\Gamma(5+\alpha/2)}{\Gamma(5-\alpha/2)}t^{4-\alpha/2} + \frac{9}{4}\Gamma(\alpha+1) + \left(3t^{\alpha/2}/2 - t^4\right)^3 - y(t)^{3/2}, \\ y(0) = 0. \end{cases}$$

Solution: $y(t) = t^8 - 3t^{4+\alpha/2} + \frac{9}{4}t^{\alpha}$.

α	τ	E	q
0.25	5.00e-01	1.42e+00	
	2.50e-01	4.17e-01	1.77
	1.25e-01	2.13e-01	0.97
	6.25e-02	1.03e-01	1.05
	3.12e-02	5.04e-02	1.03
	1.56e-02	2.44e-02	1.05

```
hval = 2.^(-(1:6));
for i=1:length(hval)
 h = hval(i);
 t0 = 0; T = 1;
 [T, Y] = fde_pi12_pc(alpha, f_fun,
 ↪  t0, T, y0, h, [], 1);
 err(i) = norm(Y - ye(T),'inf');
end
```

An example

Example

$$\begin{cases} {}_{CA}D^{\alpha}_{[0,t]}y(t) = \frac{40320}{\Gamma(9-\alpha)}t^{8-\alpha} - 3\frac{\Gamma(5+\alpha/2)}{\Gamma(5-\alpha/2)}t^{4-\alpha/2} + \frac{9}{4}\Gamma(\alpha+1) + \left(3t^{\alpha/2}/2 - t^4\right)^3 - y(t)^{3/2}, \\ y(0) = 0. \end{cases}$$

Solution: $y(t) = t^8 - 3t^{4+\alpha/2} + \frac{9}{4}t^{\alpha}$.

α	τ	E	q
0.25	5.00e-01	1.42e+00	
	2.50e-01	4.17e-01	1.77
	1.25e-01	2.13e-01	0.97
	6.25e-02	1.03e-01	1.05
	3.12e-02	5.04e-02	1.03
	1.56e-02	2.44e-02	1.05

α	τ	E	q
0.25	5.00e-01	2.75e+00	
	2.50e-01	1.80e+00	0.61
	1.25e-01	8.37e-01	1.10
	6.25e-02	2.45e-01	1.77
	3.12e-02	6.57e-02	1.90
	1.56e-02	2.02e-02	1.70

An example

Example

$$\begin{cases} {}_{CA}D^{\alpha}_{[0,t]}y(t) = \frac{40320}{\Gamma(9-\alpha)}t^{8-\alpha} - 3\frac{\Gamma(5+\alpha/2)}{\Gamma(5-\alpha/2)}t^{4-\alpha/2} + \frac{9}{4}\Gamma(\alpha+1) + \left(3t^{\alpha/2}/2 - t^4\right)^3 - y(t)^{3/2}, \\ y(0) = 0. \end{cases}$$

Solution: $y(t) = t^8 - 3t^{4+\alpha/2} + \frac{9}{4}t^{\alpha}$.

α	τ	E	q
0.25	5.00e-01	1.42e+00	
	2.50e-01	4.17e-01	1.77
	1.25e-01	2.13e-01	0.97
	6.25e-02	1.03e-01	1.05
	3.12e-02	5.04e-02	1.03
	1.56e-02	2.44e-02	1.05

α	τ	E	q
	1.95e-03	9.33e-04	1.42
	9.77e-04	3.58e-04	1.38
	4.88e-04	1.40e-04	1.35
	2.44e-04	5.56e-05	1.33
	1.22e-04	2.23e-05	1.32
	6.10e-05	9.00e-06	1.31

More than one correction step

One can think of improving convergence by performing **more than one correction step** in the algorithm (Diethelm, Ford, and Freed 2002).
Let us call $\mu \in \mathbb{N}$ the number of correction steps:

$$\begin{cases} y_{[0]}^{(n+1)} = y^{(0)} + \frac{\tau^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} b_{j,n+1} f(t_j, y^{(j)}), & \text{Prediction step,} \\ y_{[\ell]}^{(n+1)} = y^{(0)} + \frac{\tau^\alpha}{\Gamma(\alpha)} \left(\sum_{j=0}^{n} a_{j,n+1} f(t_j, y^{(j)}) + a_{n+1,n+1} f(t_{n+1}, y_{[\ell-1]}^{(n+1)}) \right), & \text{Correction steps} \\ & \ell = 1, \dots, \mu. \\ y^{(n+1)} \equiv y_{[\mu]}^{(n+1)}. \end{cases}$$

More than one correction step

One can think of improving convergence by performing **more than one correction step** in the algorithm (Diethelm, Ford, and Freed 2002).
Let us call $\mu \in \mathbb{N}$ the number of correction steps:

$$\begin{cases} y_{[0]}^{(n+1)} = y^{(0)} + \frac{\tau^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} b_{j,n+1} f(t_j, y^{(j)}), & \text{Prediction step,} \\ y_{[\ell]}^{(n+1)} = y^{(0)} + \frac{\tau^\alpha}{\Gamma(\alpha)} \left(\sum_{j=0}^{n} a_{j,n+1} f(t_j, y^{(j)}) + a_{n+1,n+1} f(t_{n+1}, y_{[\ell-1]}^{(n+1)}) \right), & \text{Correction steps} \\ & \ell = 1, \ldots, \mu. \\ y^{(n+1)} \equiv y_{[\mu]}^{(n+1)}. \end{cases}$$

- Each iteration is expected to increase the order of convergence of a fraction α from order 1 ($\mu = 0$) representing the fractional rectangular rule,

More than one correction step

One can think of improving convergence by performing **more than one correction step** in the algorithm (Diethelm, Ford, and Freed 2002).
Let us call $\mu \in \mathbb{N}$ the number of correction steps:

$$\begin{cases} y_{[0]}^{(n+1)} = y^{(0)} + \frac{\tau^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} b_{j,n+1} f(t_j, y^{(j)}), & \text{Prediction step,} \\ y_{[\ell]}^{(n+1)} = y^{(0)} + \frac{\tau^\alpha}{\Gamma(\alpha)} \left(\sum_{j=0}^{n} a_{j,n+1} f(t_j, y^{(j)}) + a_{n+1,n+1} f(t_{n+1}, y_{[\ell-1]}^{(n+1)}) \right), & \text{Correction steps} \\ & \ell = 1, \ldots, \mu. \\ y^{(n+1)} \equiv y_{[\mu]}^{(n+1)}. \end{cases}$$

- Each iteration is expected to increase the order of convergence of a fraction α from order 1 ($\mu = 0$) representing the fractional rectangular rule,
- The standard predictor corrector method is obtained for $\mu = 1$.

Convergence behavior

The convergence behavior can be described by using repeatedly the result from (Diethelm, Ford, and Freed 2004, Lemma 3.1) that we have used to obtain the other convergence bounds.

Corollary

$$\max_{0\le n\le N} |y(t_n) - y^{(n)}| = \begin{cases} O(\tau^{\min(1+\mu\alpha,2)}), & \text{if } {}_{CA}D^{\alpha}_{[t_0,t]}y(t) \in \mathcal{C}^2([0,T]), \\ O(\tau^{\min(1+\mu\alpha,2-\alpha)}), & \text{if } y(t) \in \mathcal{C}^2([0,T]), \\ O(\tau^{1+\alpha}), & \text{if } f(t,y) \in \mathcal{C}^2([O,T]\times\mathbb{D}). \end{cases}$$

Convergence behavior

The convergence behavior can be described by using repeatedly the result from (Diethelm, Ford, and Freed 2004, Lemma 3.1) that we have used to obtain the other convergence bounds.

Corollary

$$\max_{0\le n\le N} |y(t_n) - y^{(n)}| = \begin{cases} O(\tau^{\min(1+\mu\alpha,2)}), & \text{if } {}_{CA}D^{\alpha}_{[t_0,t]}y(t) \in \mathcal{C}^2([0,T]), \\ O(\tau^{\min(1+\mu\alpha,2-\alpha)}), & \text{if } y(t) \in \mathcal{C}^2([0,T]), \\ O(\tau^{1+\alpha}), & \text{if } f(t,y) \in \mathcal{C}^2([O,T]\times\mathbb{D}). \end{cases}$$

- The maximum order of convergence for ${}_{CA}D^{\alpha}_{[t_0,t]}y(t) \in \mathcal{C}^2([0,T])$ is obtained for $\mu = \lceil 1/\alpha \rceil$,

Convergence behavior

The convergence behavior can be described by using repeatedly the result from (Diethelm, Ford, and Freed 2004, Lemma 3.1) that we have used to obtain the other convergence bounds.

Corollary

$$\max_{0\le n\le N} |y(t_n) - y^{(n)}| = \begin{cases} O(\tau^{\min(1+\mu\alpha,2)}), & \text{if } {}_{CA}D^{\alpha}_{[t_0,t]}y(t) \in \mathcal{C}^2([0,T]), \\ O(\tau^{\min(1+\mu\alpha,2-\alpha)}), & \text{if } y(t) \in \mathcal{C}^2([0,T]), \\ O(\tau^{1+\alpha}), & \text{if } f(t,y) \in \mathcal{C}^2([O,T]\times\mathbb{D}). \end{cases}$$

- The maximum order of convergence for ${}_{CA}D^{\alpha}_{[t_0,t]}y(t) \in \mathcal{C}^2([0,T])$ is obtained for $\mu = \lceil 1/\alpha \rceil$,
- The maximum order of convergence for $y(t) \in \mathcal{C}^2([0,T])$ is obtained for $\mu = \lceil 1-\alpha/\alpha \rceil$,

Convergence behavior

The convergence behavior can be described by using repeatedly the result from (Diethelm, Ford, and Freed 2004, Lemma 3.1) that we have used to obtain the other convergence bounds.

Corollary

$$\max_{0\le n\le N} |y(t_n) - y^{(n)}| = \begin{cases} O(\tau^{\min(1+\mu\alpha,2)}), & \text{if } {}_{CA}D^{\alpha}_{[t_0,t]}y(t) \in \mathcal{C}^2([0,T]), \\ O(\tau^{\min(1+\mu\alpha,2-\alpha)}), & \text{if } y(t) \in \mathcal{C}^2([0,T]), \\ O(\tau^{1+\alpha}), & \text{if } f(t,y) \in \mathcal{C}^2([O,T]\times\mathbb{D}). \end{cases}$$

- The maximum order of convergence for ${}_{CA}D^{\alpha}_{[t_0,t]}y(t) \in \mathcal{C}^2([0,T])$ is obtained for $\mu = \lceil 1/\alpha \rceil$,
- The maximum order of convergence for $y(t) \in \mathcal{C}^2([0,T])$ is obtained for $\mu = \lceil 1-\alpha/\alpha \rceil$,
- In the third case with a single corrector step, and no improvement is possible.

Convergence behavior

The convergence behavior can be described by using repeatedly the result from (Diethelm, Ford, and Freed 2004, Lemma 3.1) that we have used to obtain the other convergence bounds.

Corollary

$$\max_{0\le n\le N} |y(t_n) - y^{(n)}| = \begin{cases} O(\tau^{\min(1+\mu\alpha,2)}), & \text{if } {}_{CA}D^{\alpha}_{[t_0,t]}y(t) \in \mathcal{C}^2([0,T]), \\ O(\tau^{\min(1+\mu\alpha,2-\alpha)}), & \text{if } y(t) \in \mathcal{C}^2([0,T]), \\ O(\tau^{1+\alpha}), & \text{if } f(t,y) \in \mathcal{C}^2([O,T]\times\mathbb{D}). \end{cases}$$

- The maximum order of convergence for ${}_{CA}D^{\alpha}_{[t_0,t]}y(t) \in \mathcal{C}^2([0,T])$ is obtained for $\mu = \lceil 1/\alpha \rceil$,
- The maximum order of convergence for $y(t) \in \mathcal{C}^2([0,T])$ is obtained for $\mu = \lceil 1-\alpha/\alpha \rceil$,
- In the third case with a single corrector step, and no improvement is possible.
- In general we could fix a maximum number of steps μ and halt the procedure when the error is under a certain tolerance.

Absolute stability

Let us focus on the **test problem**

$$ {}_{CA}D^{\alpha}_{[t_0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad \lambda \in \mathbb{C}, \quad 0 < \alpha < 1. $$

In the last lecture we have seen that the solution of this problem can be expressed as

$$ y(t) = E_{\alpha}(\lambda(t - t_0)^{\alpha})y_0. $$

Absolute stability

Let us focus on the **test problem**

$$ {}_{CA}D^{\alpha}_{[t_0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad \lambda \in \mathbb{C}, \quad 0 < \alpha < 1. $$

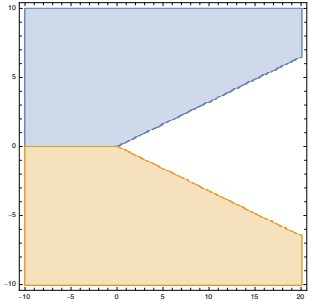
In the last lecture we have seen that the solution of this problem can be expressed as

$$ y(t) = E_{\alpha}(\lambda(t-t_0)^{\alpha})y_0. $$

Asymptotic behavior

The solution $y(t)$ asymptotically vanishes as $t \to +\infty$ for

$$ \lambda \in S^* = \{z \in \mathbb{C} : |\arg(z) - \pi| < (1 - \alpha/2)\pi.\} $$



Absolute stability

Let us focus on the **test problem**

$$ {}_{CA}D^{\alpha}_{[t_0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad \lambda \in \mathbb{C}, \quad 0 < \alpha < 1. $$

In the last lecture we have seen that the solution of this problem can be expressed as

$$ y(t) = E_{\alpha}(\lambda(t - t_0)^{\alpha})y_0. $$

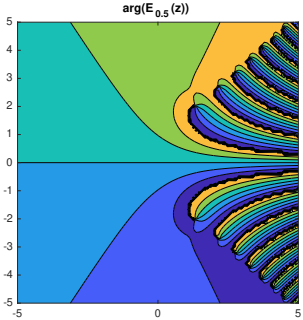
Asymptotic behavior

The solution $y(t)$ asymptotically vanishes as $t \to +\infty$ for

$$ \lambda \in S^* = \{z \in \mathbb{C} : |\arg(z) - \pi| < (1 - \alpha/2)\pi.\} $$

The application of PI rule leads to a non-homogeneous difference equation

$$ y^{(n)} = g^{(n)} + \sum_{j=k}^{n} c_{n-j} y^{(j)}, \quad n \geq k, $$



Absolute stability

Let us focus on the **test problem**

$$ {}_{CA}D^{\alpha}_{[t_0,t]}y(t) = \lambda y(t), \quad y(0) = y_0, \quad \lambda \in \mathbb{C}, \quad 0 < \alpha < 1. $$

In the last lecture we have seen that the solution of this problem can be expressed as

$$ y(t) = E_{\alpha}(\lambda(t-t_0)^{\alpha})y_0. $$

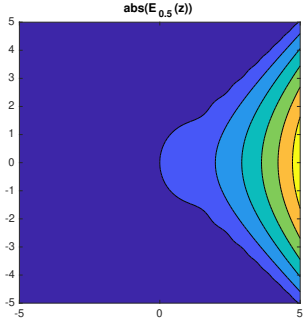
Asymptotic behavior

The solution $y(t)$ asymptotically vanishes as $t \to +\infty$ for

$$ \lambda \in S^* = \{z \in \mathbb{C} : |\arg(z) - \pi| < (1 - \alpha/2)\pi.\} $$

The application of PI rule leads to a non-homogeneous difference equation

$$ y^{(n)} = g^{(n)} + \sum_{j=k}^{n} c_{n-j} y^{(j)}, \quad n \geq k, $$



Absolute stability

Informally

The stability region of the various PI formulas can be described as the set of all $z = \tau^\alpha \lambda$ for which the numerical solution $\{y^{(n)}\}_n$ behaves as the true solution and tends to 0 as $n \to +\infty$.

Absolute stability

Informally

The stability region of the various PI formulas can be described as the set of all $z = \tau^\alpha \lambda$ for which the numerical solution $\{y^{(n)}\}_n$ behaves as the true solution and tends to 0 as $n \to +\infty$.

As for the other theoretical result we are going to leverage information on the associated Volterra integral equation (Lubich 1986a).

- First we rewrite our non-homogeneous difference equation (in which we simplify the notation assuming to work with scalars) as

$$\begin{cases} y_n = f_n + \tau^\alpha \sum_{j=0}^{n} \omega_{n-j} g(y_j), & n \geq 0 \\ f_n = f(t_n) + \tau^\alpha \sum_{j=-m}^{-1} w_{n,j} g(y_j), & t_n = t_0 + n\tau, \quad t_0 = mh. \end{cases}$$

- Then we assume that $h^\alpha w_{n,j} g(y_j) = O((n\tau)^{\alpha-1}\tau g(y_j))$, i.e., $w_{n,j} = O(n^{\alpha-1})$ as $n \to +\infty$, $j = -M, \ldots, -1$.

Absolute stability

A connection to the classical theory

In the classical case $\alpha = 1$, if we can express the term

$$\sum_{n=0}^{+\infty} \omega_n \zeta^n = \frac{\sigma(\zeta^{-1})}{\rho(\zeta^{-1})}$$

as a rational function, then we have found a standard Linear Multistep Method.

Absolute stability

A connection to the classical theory

In the classical case $\alpha = 1$, if we can express the term

$$\sum_{n=0}^{+\infty} \omega_n \zeta^n = \frac{\sigma(\zeta^{-1})}{\rho(\zeta^{-1})}$$

as a rational function, then we have found a standard Linear Multistep Method.

A-stable method

A convolution quadrature $\{\omega\}_n$ for the Abel equation

$$y(t) = f(t) + \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} g[y(s)]\,\mathrm{d}s, \quad t \geq 0,\ 0 < \alpha \leq 1,$$

is called A-stable if the solution $\{y_n\}_n$ given by the convolution quadrature satisfies

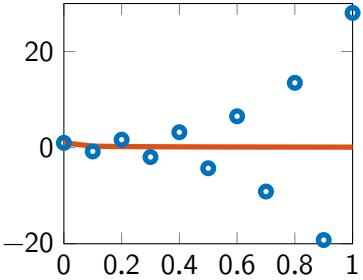
$$y_n \to 0 \text{ as } n \to +\infty \text{ whenever } \{f_n\}_n \text{ has a finite limit } \forall \tau > 0,\ \forall \lambda \in S^*.$$

Stability region

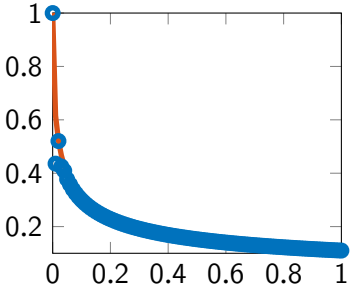
In general we cannot expect to have stability for every $\lambda \in S^*$, consider, e.g.

$$ {}_{CA}D^{\alpha}_{[t_0,t]}y(t) = -5y(t), \quad y(0) = 1, \quad T = 1. $$

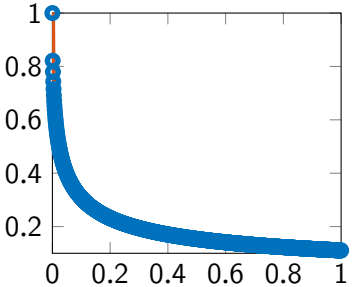
integrated with the explicit fractional rectangular rule



$h = 10^{-1}$



$h = 10^{-2}$



$h = 10^{-3}$

Stability region

Stability region

The *stability region* S of a convolution quadrature $\{\omega_m\}$ is the set of all complex $z = \tau^\alpha \lambda$ for which the numerical solution $\{y_n\}_n$ satisfies

$$y_n \to 0 \text{ as } n \to +\infty \text{ whenever } \{f_n\}_n \text{ has a finite limit.}$$

The method is called *strongly stable*, if for any $\lambda \in S^*$ there exists $\tau_0(\lambda) > 0$ such that $\tau^\alpha \lambda \in S$ for all $0 < \tau < \tau_0(\lambda)$. The method is called $A(\theta)$-stable if S contains the sector $|\arg(z) - \pi| < \theta$.

Stability region

Stability region

The *stability region* S of a convolution quadrature $\{\omega_m\}$ is the set of all complex $z = \tau^\alpha \lambda$ for which the numerical solution $\{y_n\}_n$ satisfies

$$y_n \to 0 \text{ as } n \to +\infty \text{ whenever } \{f_n\}_n \text{ has a finite limit.}$$

The method is called *strongly stable*, if for any $\lambda \in S^*$ there exists $\tau_0(\lambda) > 0$ such that $\tau^\alpha \lambda \in S$ for all $0 < \tau < \tau_0(\lambda)$. The method is called $A(\theta)$-stable if S contains the sector $|\arg(z) - \pi| < \theta$.

To obtain the characterization we need, we consider weights

$$\omega_n = (-1)^n \binom{-\alpha}{n} + v_n, \quad n \geq 0, \{v_n\}_n \in \ell^1, \tag{H$_1$}$$

to which corresponds

$$\omega(\zeta) = (1-\zeta)^{-\alpha} + v(\zeta) \text{ continuous in } \{\zeta \in \mathbb{C} : |\zeta| \leq 1,\ \zeta \neq 1\},\ \lim_{\zeta \to 1^-} w(\zeta) = +\infty.$$

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}\,.$$

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Proof. Let $z = \tau^\alpha \lambda$. Since 0 is neither contained in S^* nor in S, we can assume $z \neq 0$. We can rewrite our difference equation as

$$y(\zeta) = f(\zeta) + z\omega(\zeta)y(\zeta) \;\Leftrightarrow\; y(\zeta) = \frac{f(\zeta)}{1 - z\omega(\zeta)} = \frac{(1-\zeta)^\alpha f(\zeta)}{(1-\zeta)^\alpha[1 - z\omega(\zeta)]}.$$

We first prove that $S \subseteq S^*$.

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Proof. Let $z = \tau^\alpha \lambda$. Since 0 is neither contained in S^* nor in S, we can assume $z \neq 0$. We can rewrite our difference equation as

$$y(\zeta) = f(\zeta) + z\omega(\zeta)y(\zeta) \;\Leftrightarrow\; y(\zeta) = \frac{f(\zeta)}{1 - z\omega(\zeta)} = \frac{(1-\zeta)^\alpha f(\zeta)}{(1-\zeta)^\alpha[1 - z\omega(\zeta)]}.$$

We first prove that $S \subseteq S^*$.

- The coefficient sequence $(1-\zeta)^\alpha[1 - z\omega(\zeta)]$ is in ℓ^1, indeed $v(\zeta)$ and $(1-\zeta)^\alpha$ are in ℓ^1 by using (H_1) (for the first one with $-\alpha$ instead of α), hence also $1 + (1-\zeta)^\alpha v(\zeta) = (1-\zeta)^\alpha \omega(\zeta)$, since for any two sequences in ℓ^1 we have $\sum_n |\sum_i a_{n-i} b_i| \leq \sum |a_i||b_i|$.

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Proof. Let $z = \tau^\alpha \lambda$. Since 0 is neither contained in S^* nor in S, we can assume $z \neq 0$. We can rewrite our difference equation as

$$y(\zeta) = f(\zeta) + z\omega(\zeta)y(\zeta) \;\Leftrightarrow\; y(\zeta) = \frac{f(\zeta)}{1 - z\omega(\zeta)} = \frac{(1-\zeta)^\alpha f(\zeta)}{(1-\zeta)^\alpha[1 - z\omega(\zeta)]}.$$

We first prove that $S \subseteq S^*$.

- The coefficient sequence $(1-\zeta)^\alpha[1 - z\omega(\zeta)]$ is in ℓ^1,
- If $z \in S$ then $1 - z\omega(\zeta) \neq 0$ for $|\zeta| \leq 1$ with $\zeta \neq 1$.

Stability region

Wiener inversion Theorem

$f(\zeta) = \sum_{n=0}^{+\infty} a_n \zeta^n$ with $\|f\|_1 < +\infty$, $\zeta = e^{in\theta}$, then $1/f(\theta) \in \ell^1$ iff $f(\theta) \neq 0$ for all θ.

Proof. Let $z = \tau^\alpha \lambda$. Since 0 is neither contained in S^* nor in S, we can assume $z \neq 0$. We can rewrite our difference equation as

$$y(\zeta) = f(\zeta) + z\omega(\zeta)y(\zeta) \;\Leftrightarrow\; y(\zeta) = \frac{f(\zeta)}{1 - z\omega(\zeta)} = \frac{(1-\zeta)^\alpha f(\zeta)}{(1-\zeta)^\alpha[1 - z\omega(\zeta)]}.$$

We first prove that $S \subseteq S^*$.

- The coefficient sequence $(1-\zeta)^\alpha[1 - z\omega(\zeta)]$ is in ℓ^1,
- If $z \in S$ then $1 - z\omega(\zeta) \neq 0$ for $|\zeta| \leq 1$ with $\zeta \neq 1$.

(H_1) $(1-\zeta)^\alpha[1 - z\omega(\zeta)] = (1-\zeta)^\alpha[1 - zv(\zeta)] - z$ and thus

$$(1-\zeta)^\alpha[1 - z\omega(\zeta)] \neq 0 \text{ for } |\zeta| \leq 1$$

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Proof. Let $z = \tau^\alpha \lambda$. Since 0 is neither contained in S^* nor in S, we can assume $z \neq 0$. We can rewrite our difference equation as

$$y(\zeta) = f(\zeta) + z\omega(\zeta)y(\zeta) \;\Leftrightarrow\; y(\zeta) = \frac{f(\zeta)}{1 - z\omega(\zeta)} = \frac{(1-\zeta)^\alpha f(\zeta)}{(1-\zeta)^\alpha[1 - z\omega(\zeta)]}.$$

We first prove that $S \subseteq S^*$.

- The coefficient sequence $(1-\zeta)^\alpha[1 - z\omega(\zeta)]$ is in ℓ^1,
- If $z \in S$ then $1 - z\omega(\zeta) \neq 0$ for $|\zeta| \leq 1$ with $\zeta \neq 1$.

(H_1) $(1-\zeta)^\alpha[1 - z\omega(\zeta)] = (1-\zeta)^\alpha[1 - zv(\zeta)] - z$ and thus

$$(1-\zeta)^\alpha[1 - z\omega(\zeta)] \neq 0 \text{ for } |\zeta| \leq 1 \;\Rightarrow\; 1/(1-\zeta)^\alpha[1 - z\omega(\zeta)] \in \ell^1.$$

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Proof. We first prove that $S \subseteq S^*$. Let $\tilde{f}_n = f_n - f_\infty \overset{n\to+\infty}{\quad 0}$ so that we can write

$$f(\zeta) = \frac{f_\infty}{1-\zeta} + \tilde{f}(\zeta) \;\Rightarrow\; (1-\zeta)^\alpha f(\zeta) = (1-\zeta)^{\alpha-1} f_\infty + (1-\zeta)^\alpha \tilde{f}(\zeta) \text{ has coefficients } \to 0.$$

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Proof. We first prove that $S \subseteq S^*$. Let $\tilde{f}_n = f_n - f_\infty \overset{n\to+\infty}{\quad 0 \quad}$ so that we can write

$$f(\zeta) = \frac{f_\infty}{1-\zeta} + \tilde{f}(\zeta) \;\Rightarrow\; (1-\zeta)^\alpha f(\zeta) = (1-\zeta)^{\alpha-1} f_\infty + (1-\zeta)^\alpha \tilde{f}(\zeta) \text{ has coefficients } \to 0.$$

By (H_1) the coefficient sequence of $(1-\zeta)^{\alpha-1} \to 0$.

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Proof. We first prove that $S \subseteq S^*$. Let $\tilde{f}_n = f_n - f_\infty \overset{n\to+\infty}{} 0$ so that we can write

$$f(\zeta) = \frac{f_\infty}{1-\zeta} + \tilde{f}(\zeta) \;\Rightarrow\; (1-\zeta)^\alpha f(\zeta) = (1-\zeta)^{\alpha-1} f_\infty + (1-\zeta)^\alpha \tilde{f}(\zeta) \text{ has coefficients } \to 0.$$

By (H_1) the coefficient sequence of $(1-\zeta)^{\alpha-1} \to 0$. The coefficient sequence of $(1-\zeta)^\alpha \tilde{f}(\zeta) \to 0$ since $(1-\zeta)^\alpha \in \ell_1$ and $\ell_1 * c_0 \subset c_0$ for $*$ the convolution operator, and c_0 the space of zero sequences

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Proof. We first prove that $S \subseteq S^*$. Let $\tilde{f}_n = f_n - f_\infty \overset{n\to+\infty}{\quad} 0$ so that we can write

$$f(\zeta) = \frac{f_\infty}{1-\zeta} + \tilde{f}(\zeta) \;\Rightarrow\; (1-\zeta)^\alpha f(\zeta) = (1-\zeta)^{\alpha-1} f_\infty + (1-\zeta)^\alpha \tilde{f}(\zeta) \text{ has coefficients } \to 0.$$

By (H_1) the coefficient sequence of $(1-\zeta)^{\alpha-1} \to 0$. The coefficient sequence of $(1-\zeta)^\alpha \tilde{f}(\zeta) \to 0$ since $(1-\zeta)^\alpha \in \ell_1$ and $\ell_1 * c_0 \subset c_0$ for $*$ the convolution operator, and c_0 the space of zero sequences $\Rightarrow$ the sequence $\{y_n\}_n$ of $y(\zeta)$ is in c_0. Hence we have proved that if $z \in S$ then $z \in S^*$.

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Proof. To conclude we need to prove that S^* is exhausted by S, we assume that

$$1 - z\omega(\zeta_0) = 0 \text{ for some } |\zeta_0| \leq 1 \text{ and by } (\mathrm{H}_1)\ \zeta_0 \neq 1,$$

and show that then $z \notin S^*$.

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Proof. To conclude we need to prove that S^* is exhausted by S, we assume that

$$1 - z\omega(\zeta_0) = 0 \text{ for some } |\zeta_0| \leq 1 \text{ and by } (\mathrm{H}_1)\ \zeta_0 \neq 1,$$

and show that then $z \notin S^*$.We select

$$y(\zeta) = \frac{(1-\zeta)^\alpha}{\zeta - \zeta_0} = \frac{(1-\zeta)^\alpha - (1-\zeta_0)^\alpha}{\zeta - \zeta_0} + (1-\zeta_0)^\alpha \frac{1}{\zeta - \zeta_0}.$$

Stability region

Lemma (Lubich 1986a, Lemma 2.1)

Assume that the coefficient sequence of $a(\zeta)$ is in ℓ^1. Let $|\zeta_0| \leq 1$. Then the coefficient sequence of

$$\frac{a(\zeta) - a(\zeta_0)}{\zeta - \zeta_0} \text{ converges to zero.}$$

Proof. To conclude we need to prove that S^* is exhausted by S, we assume that

$$1 - z\omega(\zeta_0) = 0 \text{ for some } |\zeta_0| \leq 1 \text{ and by } (\mathrm{H}_1)\ \zeta_0 \neq 1,$$

and show that then $z \notin S^*$.We select

$$y(\zeta) = \frac{(1-\zeta)^\alpha}{\zeta - \zeta_0} = \underbrace{\frac{(1-\zeta)^\alpha - (1-\zeta_0)^\alpha}{\zeta - \zeta_0}}_{=0} + (1-\zeta_0)^\alpha \frac{1}{\zeta - \zeta_0}.$$

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Proof. To conclude we need to prove that S^* is exhausted by S, we assume that

$$1 - z\omega(\zeta_0) = 0 \text{ for some } |\zeta_0| \leq 1 \text{ and by } (\mathrm{H}_1)\ \zeta_0 \neq 1,$$

and show that then $z \notin S^*$.We select

$$y(\zeta) = \frac{(1-\zeta)^\alpha}{\zeta - \zeta_0} = +(1-\zeta_0)^\alpha \frac{1}{\zeta - \zeta_0}.$$

On the other hand, $1/\zeta - \zeta_0 = -\sum_{n=0}^{+\infty} \zeta_0^{-n-1}\zeta^n$ diverges! Hence also the sequence associated to $y(\zeta)$ diverges.

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Proof. We can now collect the various parts together

$$\begin{aligned} f(\zeta) = [1 - z\omega(\zeta)]y(\zeta) =& (1-\zeta)^{\alpha}[1 - z\omega(\zeta)](1-\zeta)^{-\alpha}y(\zeta) \\ =& \frac{(1-\zeta)^{\alpha}(1 - z\omega(\zeta)) - (1-\zeta_0)(1 - z\omega(\zeta_0))}{\zeta - \zeta_0} \end{aligned}$$

using again the lemma we get that $\{f_n\}_n$ goes to zero, but, $\{y_n\}_n$ does not, hence $z \notin S^*$.

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Corollary

If a convolution quadrature satisfying (H_1) is applied to the Volterra equation and if $\tau^\alpha \lambda \in S$, then $\{y_n\}_n$ is bounded whenever $\{f_n\}_n$ is bounded. Conversely, if $\{y_n\}_n$ is bounded whenever $\{f_n\}_n$ is bounded then $\tau^\alpha \lambda \in \overline{S}$.

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Corollary

The stability region of an explicit convolution quadrature ($\omega_0 = 0$) satisfying (H_1) is bounded.

Proof. By the open mapping theorem $\omega(\zeta)$ maps neighborhood of 0 into neighborhood of 0. Hence S^* is a neighborhood of ∞, and the result follows from the Theorem.

Stability region

Theorem (Lubich 1986a, Theorem 2.1)

The stability region of a convolution quadrature under the condition (H_1) is

$$S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}.$$

Corollary

The stability region of an explicit convolution quadrature ($\omega_0 = 0$) satisfying (H_1) is bounded.

Corollary

Every convolution quadrature satisfying (H_1) is strongly stable.

- Using these results we can recover the stability regions for the different methods,
- Often PI rules do not possess analytical representation of $\omega(\zeta)$ we can just use numerical approximations.

Stability region: predictor corrector method

For the Predictor-Corrector method we have

$$\begin{cases} y_P^{(n+1)} = y^{(0)} + \frac{\tau^\alpha}{\Gamma(\alpha)} \sum_{j=0}^{n} b_{j,n+1} f(t_j, y^{(j)}), \\ y^{(n+1)} = y^{(0)} + \frac{\tau^\alpha}{\Gamma(\alpha)} \left(\sum_{j=0}^{n} a_{j,n+1} f(t_j, y^{(j)}) + a_{n+1,n+1} f(t_{n+1}, y_P^{(n+1)}) \right) \end{cases}$$

where

$$b_{j,n+1} = \frac{(n+1-j)^\alpha - (n-j)^\alpha}{\alpha}$$

$$a_{j,n+1} = \begin{cases} {(n^{\alpha+1} - (n-\alpha)(n+1)^\alpha}/{\alpha(\alpha+1)}, & j = 0, \\ {(n-j+2)^{\alpha+1} - 2(n-j+1)^{\alpha+1} + (n-j)^{\alpha+1}}/{\alpha(\alpha+1)}, & j = 1, 2, \ldots, n, \\ {1}/{\alpha(\alpha+1)}, & j = n+1. \end{cases}$$

Stability region: predictor corrector method

For the Predictor-Corrector method we have

$$\begin{cases} y_P^{(n+1)} = y^{(0)} + \tau^\alpha \sum_{j=0}^{n} b_{n-j-1} f(t_j, y^{(j)}), \\ y^{(n+1)} = y^{(0)} + \tau^\alpha a_{n,0} f^{(0)} + \tau^\alpha \sum_{j=1}^{n} a_{n-j} f(t_n, y_P^{(n+1)}) \end{cases}$$

where

$$\begin{aligned} b_n &= \frac{(n+1)^\alpha - n^\alpha}{\Gamma(\alpha+1)} \\ a_{n,0} &= {}^{(n-1)^{\alpha+1} - n^\alpha (n-\alpha-1)}/_{\Gamma(\alpha+2)}, \\ a_n &= \begin{cases} {}^{1}/_{\Gamma(\alpha+2)}, & n = 0, \\ {}^{(n-1)^{\alpha+1} - 2n^{\alpha+1} + (n+1)^{\alpha+1}}/_{\Gamma(\alpha+2)}, & n \geq 1. \end{cases} \end{aligned}$$

Stability region: predictor corrector method

For the Predictor-Corrector method we have

$$y^{(n)} = g^{(n)} + \sum_{j=k}^{n} c_{n-j} y^{(j)}, \quad n \geq k,$$

where

$$\begin{cases} g^{(n)} = (1 + za_{n,0} + za_0 + z^2 a_0 b_{n-1}) y^{(0)}, \\ c_0 = 0, \ c_n = za_n + z^2 a_0 b_{n-1}, \qquad\qquad n \geq 1. \end{cases}$$

Stability region: predictor corrector method

For the Predictor-Corrector method we have

$$y^{(n)} = g^{(n)} + \sum_{j=k}^{n} c_{n-j} y^{(j)}, \quad n \geq k,$$

where

$$\begin{cases} g^{(n)} = (1 + za_{n,0} + za_0 + z^2 a_0 b_{n-1}) y^{(0)}, \\ c_0 = 0, \ c_n = za_n + z^2 a_0 b_{n-1}, \qquad\qquad n \geq 1. \end{cases}$$

✿ To apply the stability region Theorem we have then to investigate the quantity $1 - c(\zeta)$ for $|\zeta| \leq 1$, and $c(\zeta) = \sum_{n=0}^{+\infty} c_n \zeta^n$.

Stability region: predictor corrector method

For the Predictor-Corrector method we have

$$y^{(n)} = g^{(n)} + \sum_{j=k}^{n} c_{n-j} y^{(j)}, \quad n \geq k,$$

where

$$\begin{cases} g^{(n)} = (1 + z a_{n,0} + z a_0 + z^2 a_0 b_{n-1}) y^{(0)}, \\ c_0 = 0, \ c_n = z a_n + z^2 a_0 b_{n-1}, \qquad\qquad n \geq 1. \end{cases}$$

⚙ To apply the stability region Theorem we have then to investigate the quantity $1 - c(\zeta)$ for $|\zeta| \leq 1$, and $c(\zeta) = \sum_{n=0}^{+\infty} c_n \zeta^n$.

Proposition

The stability region of the Predictor-Corrector method is

$$S = \{z \in \mathbb{C} \,|\, 1 - z(\alpha(\zeta) - a_0) - z^2 a_0 \zeta b(\zeta) \neq 0 \ : \ |\zeta| \leq 1\}.$$

Stability region: predictor corrector method

Proposition

The stability region of the Predictor-Corrector method is

$$S = \{z \in \mathbb{C} \mid 1 - z(\alpha(\zeta) - a_0) - z^2 a_0 \zeta b(\zeta) \neq 0 \ : \ |\zeta| \leq 1\}.$$

Proof. To apply the Theorem we need to prove (H_1), we use the binomial series to write

$$(n-1)^p = n^p - pn^{p-1} + \frac{p(p-1)}{2} n^{p-2} + \frac{p(p-1)(p-2)}{6} n^{p-3} + O(n^{p-4}),$$

and similarly for $(n+1)^p$, from which we obtain

$$b_n = \frac{1}{\Gamma(\alpha)} n^{\alpha-1} + O(n^{\alpha-2}), \ a_{n,0} = \frac{1}{2\Gamma(\alpha)} n^{\alpha-1} + O(n^{\alpha-2}), \ \alpha_n = \frac{1}{\Gamma(\alpha)} n^{\alpha-1} + O(n^{\alpha-3}),$$

and the expression we need for $c(\zeta)$ as

$$c(\zeta) = z(\alpha(\zeta) - \alpha_0) + z^2 \alpha_0 \zeta b(\zeta). \qquad \square$$

Stability region: predictor corrector method

Proposition

The stability region of the Predictor-Corrector method is

$$S = \{z \in \mathbb{C} \mid 1 - z(\alpha(\zeta) - a_0) - z^2 a_0 \zeta b(\zeta) \neq 0 \ : \ |\zeta| \leq 1\}.$$

Proof. To apply the Theorem we need to prove (H_1), we use the binomial series to write

$$(n-1)^p = n^p - pn^{p-1} + \frac{p(p-1)}{2} n^{p-2} + \frac{p(p-1)(p-2)}{6} n^{p-3} + O(n^{p-4}),$$

and similarly for $(n+1)^p$, from which we obtain

$$b_n = \frac{1}{\Gamma(\alpha)} n^{\alpha-1} + O(n^{\alpha-2}), \ a_{n,0} = \frac{1}{2\Gamma(\alpha)} n^{\alpha-1} + O(n^{\alpha-2}), \ \alpha_n = \frac{1}{\Gamma(\alpha)} n^{\alpha-1} + O(n^{\alpha-3}),$$

and the expression we need for $c(\zeta)$ as

$$c(\zeta) = z(\alpha(\zeta) - \alpha_0) + z^2 \alpha_0 \zeta b(\zeta). \qquad \square$$

⚙ The expression can be evaluated only numerically.

A research idea?

We have written a predictor-method in an explicit form, we can write and analyze in a similar way also a predictor-corrector made of two *implicit methods*.

- We have now to solve a (possibly) non-linear problem at each step, thus things don't seem to good...
- But we can expect better stability and convergence properties.

A research idea?

We have written a predictor-method in an explicit form, we can write and analyze in a similar way also a predictor-corrector made of two *implicit methods*.

- We have now to solve a (possibly) non-linear problem at each step, thus things don't seem to good...
- But we can expect better stability and convergence properties.
- What if we decide to solve the nonlinear problem in reduced precision?

A research idea?

We have written a predictor-method in an explicit form, we can write and analyze in a similar way also a predictor-corrector made of two *implicit methods*.

- We have now to solve a (possibly) non-linear problem at each step, thus things don't seem to good...
- But we can expect better stability and convergence properties.
- What if we decide to solve the nonlinear problem in reduced precision?

Multiprecision algorithms on specialized hardware can give both an acceleration and maintain the overall accuracy. This idea has already been partially explored for the ODE case, but not yet for FODEs:

B. Burnett et al. (2021). "Performance Evaluation of Mixed-Precision Runge-Kutta Methods". In: *2021 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, pp. 1–6

A research idea?

We have written a predictor-method in an explicit form, we can write and analyze in a similar way also a predictor-corrector made of two *implicit methods*.

- We have now to solve a (possibly) non-linear problem at each step, thus things don't seem to good...
- But we can expect better stability and convergence properties.
- What if we decide to solve the nonlinear problem in reduced precision?

Multiprecision algorithms on specialized hardware can give both an acceleration and maintain the overall accuracy. This idea has already been partially explored for the ODE case, but not yet for FODEs:

B. Burnett et al. (2021). "Performance Evaluation of Mixed-Precision Runge-Kutta Methods". In: *2021 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, pp. 1–6

Further analyses

One can investigate also stability regions, effects of multiple correction steps, tolerances and step-size selections...

Fractional Linear Multistep Method

To obtain methods that can be analyzed we can move to *Linear Multistep Methods*.

Fractional Linear Multistep Method

To obtain methods that can be analyzed we can move to *Linear Multistep Methods*.

- For an ODE a FLMM with k step is a method of the form:

$$\sum_{j=0}^{k} a_j y_{n+j} = \tau \sum_{j=0}^{k} b_j f_{n+j}, \quad n = 0, \ldots, s. \tag{1}$$

for $t_j = t_0 + j\tau$, for $j = 0, \ldots, N$, $\tau = (T - t_0)/N$,

Fractional Linear Multistep Method

To obtain methods that can be analyzed we can move to *Linear Multistep Methods*.

- For an ODE a FLMM with k step is a method of the form:

$$\sum_{j=0}^{k} a_j y_{n+j} = \tau \sum_{j=0}^{k} b_j f_{n+j}, \quad n = 0, \ldots, s. \tag{1}$$

  for $t_j = t_0 + j\tau$, for $j = 0, \ldots, N$, $\tau = (T - t_0)/N$,
- They are associated with the polynomials $\rho(z) = \sum_{j=0}^{k} a_j z^j$, $\sigma(z) = \sum_{j=0}^{k} b_j z^j$,

Fractional Linear Multistep Method

To obtain methods that can be analyzed we can move to *Linear Multistep Methods*.

- For an ODE a FLMM with k step is a method of the form:

$$\sum_{j=0}^{k} a_j y_{n+j} = \tau \sum_{j=0}^{k} b_j f_{n+j}, \quad n = 0, \dots, s. \tag{1}$$

  for $t_j = t_0 + j\tau$, for $j = 0, \dots, N$, $\tau = (T - t_0)/N$,
- They are associated with the polynomials $\rho(z) = \sum_{j=0}^{k} a_j z^j$, $\sigma(z) = \sum_{j=0}^{k} b_j z^j$,
- The fractional version has been introduced in the pioneering work (Lubich 1986b)

Theorem (Lubich 1986b, Theorem 2.6)

Let (ρ, σ) denote an implicit linear multistep method which is stable and consistent of order p. Assume that the zeros of $\sigma(\zeta)$ have absolute values less than 1. Let $w(\zeta) = \sigma(\zeta^{-1})/\rho(\zeta^{-1})$ denote the generating power series of the corresponding convolution quadrature ω. We define $\omega^\alpha = \{\omega_n^{(\alpha)}\}_{n=0}^{+\infty}$ by $\omega^\alpha(\zeta) = \omega(\zeta)^\alpha$, then the convolution quadrature ω^α is convergent of order p.

Fractional Linear Multistep Method

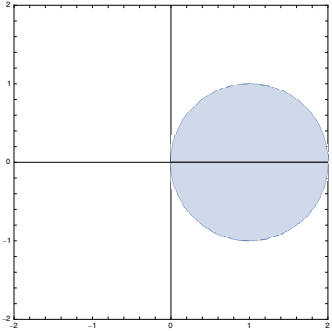
An example is represented by **B**ackward **D**ifferentiation **F**ormula**s**, for which we have

p	$\omega^{\alpha}(\zeta)$
1	$(1-\zeta)^{-\alpha}$
2	$(3/2 - 2\zeta + 1/2\zeta^2)^{-\alpha}$
3	$(11/6 - 3\zeta + 3/2\zeta^2 - 1/3\zeta^3)^{-\alpha}$
4	$(25/12 - 4\zeta + 4\zeta^2 - 4/3\zeta^3 + 1/4\zeta^4)^{-\alpha}$
5	$(137/60 - 5\zeta + 5\zeta^2 - 10/3\zeta^3 + 5/4\zeta^4 - 1/5\zeta^5)^{-\alpha}$
6	$(147/60 - 6\zeta + 15/2\zeta^2 - 20/3\zeta^3 + 15/4\zeta^4 - 6/5\zeta^5 + 1/6\zeta^6)^{-\alpha}$

Fractional Linear Multistep Method

An example is represented by **B**ackward **D**ifferentiation **F**ormula**s**, for which we have

p	$\omega^\alpha(\zeta)$
1	$(1-\zeta)^{-\alpha}$
2	$(3/2 - 2\zeta + 1/2\zeta^2)^{-\alpha}$
3	$(11/6 - 3\zeta + 3/2\zeta^2 - 1/3\zeta^3)^{-\alpha}$
4	$(25/12 - 4\zeta + 4\zeta^2 - 4/3\zeta^3 + 1/4\zeta^4)^{-\alpha}$
5	$(137/60 - 5\zeta + 5\zeta^2 - 10/3\zeta^3 + 5/4\zeta^4 - 1/5\zeta^5)^{-\alpha}$
6	$(147/60 - 6\zeta + 15/2\zeta^2 - 20/3\zeta^3 + 15/4\zeta^4 - 6/5\zeta^5 + 1/6\zeta^6)^{-\alpha}$



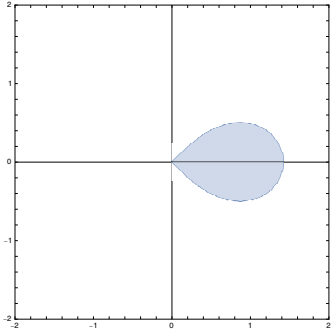
$\alpha = 1$

For $p = 1$ we can consider the stability region $S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}$ and plot the part of the $\mathbb{C}$-plane we have to remove.

Fractional Linear Multistep Method

An example is represented by **B**ackward **D**ifferentiation **F**ormula**s**, for which we have

p	$\omega^\alpha(\zeta)$
1	$(1-\zeta)^{-\alpha}$
2	$(3/2 - 2\zeta + 1/2\zeta^2)^{-\alpha}$
3	$(11/6 - 3\zeta + 3/2\zeta^2 - 1/3\zeta^3)^{-\alpha}$
4	$(25/12 - 4\zeta + 4\zeta^2 - 4/3\zeta^3 + 1/4\zeta^4)^{-\alpha}$
5	$(137/60 - 5\zeta + 5\zeta^2 - 10/3\zeta^3 + 5/4\zeta^4 - 1/5\zeta^5)^{-\alpha}$
6	$(147/60 - 6\zeta + 15/2\zeta^2 - 20/3\zeta^3 + 15/4\zeta^4 - 6/5\zeta^5 + 1/6\zeta^6)^{-\alpha}$



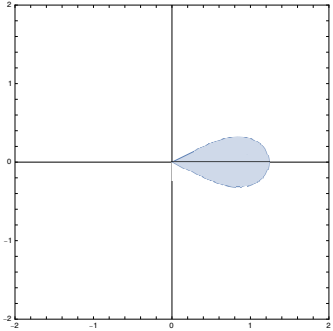
$\alpha = 0.5$

For $p = 1$ we can consider the stability region $S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}$ and plot the part of the $\mathbb{C}$-plane we have to remove.

Fractional Linear Multistep Method

An example is represented by **B**ackward **D**ifferentiation **F**ormula**s**, for which we have

p	$\omega^\alpha(\zeta)$
1	$(1-\zeta)^{-\alpha}$
2	$(3/2 - 2\zeta + 1/2\zeta^2)^{-\alpha}$
3	$(11/6 - 3\zeta + 3/2\zeta^2 - 1/3\zeta^3)^{-\alpha}$
4	$(25/12 - 4\zeta + 4\zeta^2 - 4/3\zeta^3 + 1/4\zeta^4)^{-\alpha}$
5	$(137/60 - 5\zeta + 5\zeta^2 - 10/3\zeta^3 + 5/4\zeta^4 - 1/5\zeta^5)^{-\alpha}$
6	$(147/60 - 6\zeta + 15/2\zeta^2 - 20/3\zeta^3 + 15/4\zeta^4 - 6/5\zeta^5 + 1/6\zeta^6)^{-\alpha}$



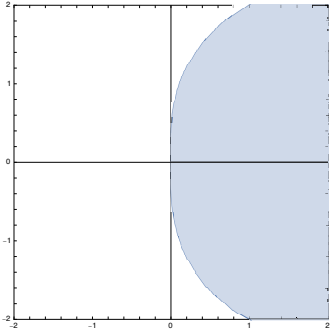
$\alpha = 0.3$

For $p = 1$ we can consider the stability region $S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}$ and plot the part of the $\mathbb{C}$-plane we have to remove.

Fractional Linear Multistep Method

An example is represented by **B**ackward **D**ifferentiation **F**ormula**s**, for which we have

p	$\omega^\alpha(\zeta)$
1	$(1-\zeta)^{-\alpha}$
2	$(3/2 - 2\zeta + 1/2\zeta^2)^{-\alpha}$
3	$(11/6 - 3\zeta + 3/2\zeta^2 - 1/3\zeta^3)^{-\alpha}$
4	$(25/12 - 4\zeta + 4\zeta^2 - 4/3\zeta^3 + 1/4\zeta^4)^{-\alpha}$
5	$(137/60 - 5\zeta + 5\zeta^2 - 10/3\zeta^3 + 5/4\zeta^4 - 1/5\zeta^5)^{-\alpha}$
6	$(147/60 - 6\zeta + 15/2\zeta^2 - 20/3\zeta^3 + 15/4\zeta^4 - 6/5\zeta^5 + 1/6\zeta^6)^{-\alpha}$



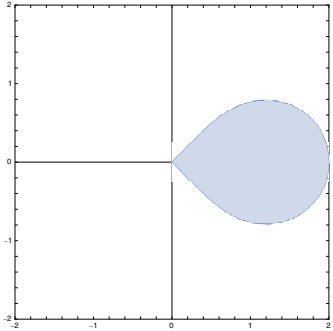
$\alpha = 1$

For $p = 1$ we can consider the stability region $S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}$ and plot the part of the $\mathbb{C}$-plane we have to remove.

Fractional Linear Multistep Method

An example is represented by **B**ackward **D**ifferentiation **F**ormula**s**, for which we have

p	$\omega^\alpha(\zeta)$
1	$(1-\zeta)^{-\alpha}$
2	$(3/2 - 2\zeta + 1/2\zeta^2)^{-\alpha}$
3	$(11/6 - 3\zeta + 3/2\zeta^2 - 1/3\zeta^3)^{-\alpha}$
4	$(25/12 - 4\zeta + 4\zeta^2 - 4/3\zeta^3 + 1/4\zeta^4)^{-\alpha}$
5	$(137/60 - 5\zeta + 5\zeta^2 - 10/3\zeta^3 + 5/4\zeta^4 - 1/5\zeta^5)^{-\alpha}$
6	$(147/60 - 6\zeta + 15/2\zeta^2 - 20/3\zeta^3 + 15/4\zeta^4 - 6/5\zeta^5 + 1/6\zeta^6)^{-\alpha}$



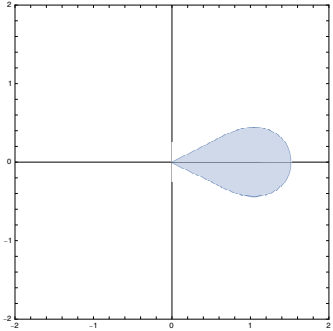
$\alpha = 0.5$

For $p = 1$ we can consider the stability region $S = \mathbb{C} \setminus \{1/\omega(\zeta) \,:\, |\zeta| \leq 1\}$ and plot the part of the $\mathbb{C}$-plane we have to remove.

Fractional Linear Multistep Method

An example is represented by **B**ackward **D**ifferentiation **F**ormula**s**, for which we have

p	$\omega^\alpha(\zeta)$
1	$(1-\zeta)^{-\alpha}$
2	$(3/2 - 2\zeta + 1/2\zeta^2)^{-\alpha}$
3	$(11/6 - 3\zeta + 3/2\zeta^2 - 1/3\zeta^3)^{-\alpha}$
4	$(25/12 - 4\zeta + 4\zeta^2 - 4/3\zeta^3 + 1/4\zeta^4)^{-\alpha}$
5	$(137/60 - 5\zeta + 5\zeta^2 - 10/3\zeta^3 + 5/4\zeta^4 - 1/5\zeta^5)^{-\alpha}$
6	$(147/60 - 6\zeta + 15/2\zeta^2 - 20/3\zeta^3 + 15/4\zeta^4 - 6/5\zeta^5 + 1/6\zeta^6)^{-\alpha}$



$\alpha = 0.3$

For $p = 1$ we can consider the stability region $S = \mathbb{C} \setminus \{1/\omega(\zeta) : |\zeta| \leq 1\}$ and plot the part of the $\mathbb{C}$-plane we have to remove.

Fractional Linear Multistep Method

An example is represented by **B**ackward **D**ifferentiation **F**ormula**s**, for which we have

p	$\omega^\alpha(\zeta)$
1	$(1-\zeta)^{-\alpha}$
2	$(3/2 - 2\zeta + 1/2\zeta^2)^{-\alpha}$
3	$(11/6 - 3\zeta + 3/2\zeta^2 - 1/3\zeta^3)^{-\alpha}$
4	$(25/12 - 4\zeta + 4\zeta^2 - 4/3\zeta^3 + 1/4\zeta^4)^{-\alpha}$
5	$(137/60 - 5\zeta + 5\zeta^2 - 10/3\zeta^3 + 5/4\zeta^4 - 1/5\zeta^5)^{-\alpha}$
6	$(147/60 - 6\zeta + 15/2\zeta^2 - 20/3\zeta^3 + 15/4\zeta^4 - 6/5\zeta^5 + 1/6\zeta^6)^{-\alpha}$

? How do we obtain the coefficients?

How can we obtain the coefficient describing the method?

Computing the FLMM coefficients

We have now the converse of the previous problem, we have a closed expression for $\omega(\zeta)$, and now we need the coefficients to write

$$I_\tau^\alpha g(t_n) = \tau^\alpha \sum_{j=0}^{n} \omega_{n-j} g(t_j) + \tau^\beta \sum_{j=0}^{s} w_{n,j} g(t_j),$$

- $\{\omega_j\}_{j=0}^n$ convolution coefficients from $\omega(\zeta)$,
- $\{w_{n,j}\}_{j=0}^k$ starting quadrature weights.

Computing the FLMM coefficients

We have now the converse of the previous problem, we have a closed expression for $\omega(\zeta)$, and now we need the coefficients to write

$$I_\tau^\alpha g(t_n) = \tau^\alpha \sum_{j=0}^{n} \omega_{n-j} g(t_j) + \tau^\beta \sum_{j=0}^{s} w_{n,j} g(t_j),$$

- $\{\omega_j\}_{j=0}^n$ convolution coefficients from $\omega(\zeta)$,
- $\{w_{n,j}\}_{j=0}^k$ starting quadrature weights.
- For the convolution coefficients we can use:

Computing the FLMM coefficients

We have now the converse of the previous problem, we have a closed expression for $\omega(\zeta)$, and now we need the coefficients to write

$$I_\tau^\alpha g(t_n) = \tau^\alpha \sum_{j=0}^{n} \omega_{n-j} g(t_j) + \tau^\beta \sum_{j=0}^{s} w_{n,j} g(t_j),$$

- $\{\omega_j\}_{j=0}^n$ convolution coefficients from $\omega(\zeta)$,
- $\{w_{n,j}\}_{j=0}^k$ starting quadrature weights.
- For the convolution coefficients we can use:
  - Fast Fourier Transform (FFT) techniques for formal power series,

Computing the FLMM coefficients

We have now the converse of the previous problem, we have a closed expression for $\omega(\zeta)$, and now we need the coefficients to write

$$I_\tau^\alpha g(t_n) = \tau^\alpha \sum_{j=0}^{n} \omega_{n-j} g(t_j) + \tau^\beta \sum_{j=0}^{s} w_{n,j} g(t_j),$$

- $\{\omega_j\}_{j=0}^n$ convolution coefficients from $\omega(\zeta)$,
- $\{w_{n,j}\}_{j=0}^k$ starting quadrature weights.
- For the convolution coefficients we can use:
  - Fast Fourier Transform (FFT) techniques for formal power series,
  - A recursion technique for complex binomial series.

Computing the FLMM coefficients

We have now the converse of the previous problem, we have a closed expression for $\omega(\zeta)$, and now we need the coefficients to write

$$I_\tau^\alpha g(t_n) = \tau^\alpha \sum_{j=0}^{n} \omega_{n-j} g(t_j) + \tau^\beta \sum_{j=0}^{s} w_{n,j} g(t_j),$$

- $\{\omega_j\}_{j=0}^n$ convolution coefficients from $\omega(\zeta)$,
- $\{w_{n,j}\}_{j=0}^k$ starting quadrature weights.
- For the convolution coefficients we can use:
  - Fast Fourier Transform (FFT) techniques for formal power series,
  - A recursion technique for complex binomial series.
- Solving a small $k \times k$ Vandermonde system.

The Newton Method for Power Series (Henrici 1979)

Let us suppose that $\alpha = 1/2$ and that we have a power series of the form

$$\omega(\zeta) = \sum_{j=0}^{+\infty} \omega_j \zeta^j,$$

for which we want to compute for a generic pth degree BDF

$$\omega(\zeta)^{-2} = q(\zeta) \text{ with } q(\zeta) = \sum_{k=1}^{p} \frac{1}{k}(1-\zeta)^k,$$

The Newton Method for Power Series (Henrici 1979)

Let us suppose that $\alpha = 1/2$ and that we have a power series of the form

$$\omega(\zeta) = \sum_{j=0}^{+\infty} \omega_j \zeta^j,$$

for which we want to compute for a generic pth degree BDF

$$F(\omega(\zeta)) = 0 \text{ with } F(w) = w^{-2} - q(\zeta).$$

The Newton Method for Power Series (Henrici 1979)

Let us suppose that $\alpha = 1/2$ and that we have a power series of the form

$$\omega(\zeta) = \sum_{j=0}^{+\infty} \omega_j \zeta^j,$$

for which we want to compute for a generic pth degree BDF
To which we can apply the Newton's method for power series

$$\begin{cases} \omega^{(0)}(\zeta) = \omega_0, \\ \omega^{(m+1)}(\zeta) = \left[\omega^{(m)}(\zeta) - F'(\omega^{(m)}(\zeta))^{-1} F(\omega^{(m)}(\zeta))\right]_{2^{m+1}}, \end{cases}$$

for $[\cdot]_k$ the truncation operator for a power series, i.e., $\left[\sum_{j=0}^{+\infty} a_j \zeta^j\right]_k = \sum_{j=0}^{k} a_j \zeta^j$, and ω_0 the solution of $[F(\omega_0)]_1 = 0$.

The Newton Method for Power Series (Henrici 1979)

Let us suppose that $\alpha = 1/2$ and that we have a power series of the form

$$\omega(\zeta) = \sum_{j=0}^{+\infty} \omega_j \zeta^j,$$

for which we want to compute for a generic pth degree BDF
To which we can apply the Newton's method for power series

$$\begin{cases} \omega^{(0)}(\zeta) = \omega_0 = q(0)^{-1/2}, \\ \omega^{(m+1)}(\zeta) = \left[3/2\omega^{(m)}(\zeta) - 1/2 \left(\omega^{(m)}(\zeta)\right)^3 q(\zeta) \right]_{2^{m+1}}, \end{cases}$$

for $[\cdot]_k$ the truncation operator for a power series, i.e., $\left[\sum_{j=0}^{+\infty} a_j \zeta^j\right]_k = \sum_{j=0}^{k} a_j \zeta^j$.
After m step we have that

$$\omega^{(m)}(\zeta) = [\omega(\zeta)]_{2^m} = \sum_{j=0}^{2^m-1} \omega_j \zeta^j \quad \forall m \geq 0 \text{ and cost } O(2^m \log(2^m)).$$

Recurrence relation

Theorem Henrici 1974, Theorem 1.6c, p. 42

Let $\phi(\zeta) = 1 + \sum_{n=1}^{+\infty} a_n \zeta^n$ be a formal power series. Then for any $\alpha \in \mathcal{C}$, we have

$$(\phi(\zeta))^\alpha = \sum_{n=0}^{+\infty} v_n^{(\alpha)} \zeta^n,$$

where coefficients $v_n^{(\alpha)}$ can be evaluated recursively as

$$v_0^{(\alpha)} = 1, \qquad v_n^{(\alpha)} = \sum_{j=1}^{n} \left(\frac{(\alpha+1)j}{n} - 1 \right) a_j v_{n-j}^{(\alpha)}$$

- This approach costs an $O(N^2)$ in general, but can be simplified, e.g., when $a_1 = \pm 1$, and $a_i > 0$ for $i > 1$ it involves only $2N$ multiplications and N additions.

Computing the starting weights

The starting weights $w_{n,j}$ in

$$I_\tau^\alpha g(t_n) = \tau^\alpha \sum_{j=0}^{n} \omega_{n-j} g(t_j) + \tau^\beta \sum_{j=0}^{s} w_{n,j} g(t_j),$$

are introduced to deal with the singular behavior of the solution close to the left endpoint of the integration interval.

Computing the starting weights

The starting weights $w_{n,j}$ in

$$I_\tau^\alpha g(t_n) = \tau^\alpha \sum_{j=0}^{n} \omega_{n-j} g(t_j) + \tau^\beta \sum_{j=0}^{s} w_{n,j} g(t_j),$$

are introduced to deal with the singular behavior of the solution close to the left endpoint of the integration interval.

Starting weight selection

We fix them by imposing that $I_\tau^\alpha t^\nu$ is exact for $\nu \in \mathcal{A} = \mathcal{A}_{p-1} \cup \{p-1\}$ with p the order of convergence of the FLMM, and $\mathcal{A}_{p-1} = \{\nu \in \mathbb{R} \mid \nu = i + j\alpha, \quad i, j \in \mathbb{N}, \nu < p-1\}$.

Computing the starting weights

The starting weights $w_{n,j}$ in

$$I_\tau^\alpha g(t_n) = \tau^\alpha \sum_{j=0}^{n} \omega_{n-j} g(t_j) + \tau^\beta \sum_{j=0}^{s} w_{n,j} g(t_j),$$

are introduced to deal with the singular behavior of the solution close to the left endpoint of the integration interval.

Starting weight selection

We fix them by imposing that $I_\tau^\alpha t^\nu$ is exact for $\nu \in \mathcal{A} = \mathcal{A}_{p-1} \cup \{p-1\}$ with p the order of convergence of the FLMM, and $\mathcal{A}_{p-1} = \{\nu \in \mathbb{R} \,|\, \nu = i + j\alpha, \quad i, j \in \mathbb{N}, \nu < p-1\}$.

$$\tau^\alpha \sum_{j=0}^{s} w_{n,j} (jh)^\nu = \frac{1}{\Gamma(\alpha)} \int_0^{n\tau} (n\tau - \xi)^{\alpha-1} \chi^\nu \, \mathrm{d}\chi - \tau^\alpha \sum_{j=0}^{n} \omega_{n-j} (jh)^\nu, \quad \nu \in \mathcal{A}.$$

Solving the Vandermonde system

The resulting linear system is of "real" Vandermonde type, i.e.,

$$(A)^s_{j,\nu_i=1} = (jh)^{\nu_i}, \qquad \nu_i \in A, \quad s = |\mathcal{A}|.$$

- The condition number depends on α!

Solving the Vandermonde system

The resulting linear system is of "real" Vandermonde type, i.e.,

$$(A)^s_{j,\nu_i=1} = (jh)^{\nu_i}, \qquad \nu_i \in A, \quad s = |\mathcal{A}|.$$

- The condition number depends on α!
- If $\alpha = 1/M$ for some integer M then we can rewrite the system in the "integer" Vandermonde form, thus *mildly* ill-conditioned,

Solving the Vandermonde system

The resulting linear system is of "real" Vandermonde type, i.e.,

$$(A)^s_{j,\nu_i=1} = (jh)^{\nu_i}, \qquad \nu_i \in A, \quad s = |\mathcal{A}|.$$

- The condition number depends on α!
- If $\alpha = 1/M$ for some integer M then we can rewrite the system in the "integer" Vandermonde form, thus *mildly* ill-conditioned,
- If $\alpha = 1/M - \epsilon$ and $p \geq 2$, then $\mathcal{A}$ will contain 1 and $M\alpha = 1 - M\epsilon$, hence the matrix will have two almost identical columns, thus a *bad* ill-conditioning.

Solving the Vandermonde system

The resulting linear system is of "real" Vandermonde type, i.e.,

$$(A)_{j,\nu_i=1}^{s} = (jh)^{\nu_i}, \qquad \nu_i \in A, \quad s = |\mathcal{A}|.$$

- The condition number depends on α!
- If $\alpha = 1/M$ for some integer M then we can rewrite the system in the "integer" Vandermonde form, thus *mildly* ill-conditioned,
- If $\alpha = 1/M - \epsilon$ and $p \geq 2$, then $\mathcal{A}$ will contain 1 and $M\alpha = 1 - M\epsilon$, hence the matrix will have two almost identical columns, thus a *bad* ill-conditioning.
- The right-hand side

$$\frac{1}{\Gamma(\alpha)} \int_0^{n\tau} (n\tau - \xi)^{\alpha-1} \chi^{\nu} \, \mathrm{d}\chi - \tau^{\alpha} \sum_{j=0}^{n} \omega_{n-j} (jh)^{\nu}$$

can suffer from cancellation of digits!

Where are we?

We know a general way to obtain FLMM methods of the form

$$y^{(n)} = T_{m-1}(t_n) + \tau^{\beta} \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^{\alpha} \sum_{j=0}^{n} \omega_{n-j} f(t_j, y^{(j)}),$$

- starting from the polynomials (ρ, σ) of an implicit order p method,

Where are we?

We know a general way to obtain FLMM methods of the form

$$y^{(n)} = T_{m-1}(t_n) + \tau^{\beta} \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^{\alpha} \sum_{j=0}^{n} \omega_{n-j} f(t_j, y^{(j)}),$$

- starting from the polynomials (ρ, σ) of an implicit order p method,
- we have seen how to compute the convolution coefficients ω_n,

Where are we?

We know a general way to obtain FLMM methods of the form

$$y^{(n)} = T_{m-1}(t_n) + \tau^{\beta} \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^{\alpha} \sum_{j=0}^{n} \omega_{n-j} f(t_j, y^{(j)}),$$

- starting from the polynomials (ρ, σ) of an implicit order p method,
- we have seen how to compute the convolution coefficients ω_n,
- we have seen how to compute the starting nodes $w_{n,j}$,

Where are we?

We know a general way to obtain FLMM methods of the form

$$y^{(n)} = T_{m-1}(t_n) + \tau^{\beta} \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^{\alpha} \sum_{j=0}^{n} \omega_{n-j} f(t_j, y^{(j)}),$$

- starting from the polynomials (ρ, σ) of an implicit order p method,
- we have seen how to compute the convolution coefficients ω_n,
- we have seen how to compute the starting nodes $w_{n,j}$,
- we need to discuss how we compute the starting values for a multi-step method,

Where are we?

We know a general way to obtain FLMM methods of the form

$$y^{(n)} = T_{m-1}(t_n) + \tau^{\beta} \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^{\alpha} \sum_{j=0}^{n} \omega_{n-j} f(t_j, y^{(j)}),$$

- ✅ starting from the polynomials (ρ, σ) of an implicit order p method,
- ✅ we have seen how to compute the convolution coefficients ω_n,
- ✅ we have seen how to compute the starting nodes $w_{n,j}$,
- 📋 we need to discuss how we compute the starting values for a multi-step method,
- 📋 we still need to discuss how we can efficiently treat the memory term.

Bibliography I

- Burnett, B. et al. (2021). "Performance Evaluation of Mixed-Precision Runge-Kutta Methods". In: *2021 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, pp. 1–6.
- Cameron, R. F. and S. McKee (July 1985). "The Analysis of Product Integration Methods for Abel's Equation using Discrete Fractional Differentiation". In: *IMA Journal of Numerical Analysis* 5.3, pp. 339–353. ISSN: 0272-4979. DOI: 10.1093/imanum/5.3.339. eprint: https://academic.oup.com/imajna/article-pdf/5/3/339/2612709/5-3-339.pdf. URL: https://doi.org/10.1093/imanum/5.3.339.
- Caputo, M. (2008). "Linear models of dissipation whose Q is almost frequency independent. II". In: *Fract. Calc. Appl. Anal.* 11.1. Reprinted from Geophys. J. R. Astr. Soc. **1**3 (1967), no. 5, 529–539, pp. 4–14. ISSN: 1311-0454.
- Diethelm, K. (1997). "An algorithm for the numerical solution of differential equations of fractional order". In: *Electron. Trans. Numer. Anal.* 5.Mar. Pp. 1–6.

Bibliography II

- Diethelm, K., N. J. Ford, and A. D. Freed (2002). "A predictor-corrector approach for the numerical solution of fractional differential equations". In: vol. 29. 1-4. Fractional order calculus and its applications, pp. 3–22. DOI: 10.1023/A:1016592219341. URL: https://doi.org/10.1023/A:1016592219341.
- — (2004). "Detailed error analysis for a fractional Adams method". In: *Numer. Algorithms* 36.1, pp. 31–52. ISSN: 1017-1398. DOI: 10.1023/B:NUMA.0000027736.85078.be. URL: https://doi.org/10.1023/B:NUMA.0000027736.85078.be.
- Dixon, J. (1985). "On the order of the error in discretization methods for weakly singular second kind Volterra integral equations with nonsmooth solutions". In: *BIT* 25.4, pp. 624–634. ISSN: 0006-3835. DOI: 10.1007/BF01936141. URL: https://doi.org/10.1007/BF01936141.
- Henrici, P. (1974). *Applied and computational complex analysis*. Pure and Applied Mathematics. Volume 1: Power series—integration—conformal mapping—location of zeros. Wiley-Interscience [John Wiley & Sons], New York-London-Sydney, pp. xv+682.

Bibliography III

- Henrici, P. (1979). "Fast Fourier methods in computational complex analysis". In: *SIAM Rev.* 21.4, pp. 481–527. ISSN: 0036-1445. DOI: 10.1137/1021093. URL: https://doi.org/10.1137/1021093.
- Lubich, C. (1986a). "A stability analysis of convolution quadratures for Abel-Volterra integral equations". In: *IMA J. Numer. Anal.* 6.1, pp. 87–101. ISSN: 0272-4979. DOI: 10.1093/imanum/6.1.87. URL: https://doi.org/10.1093/imanum/6.1.87.
- — (1986b). "Discretized fractional calculus". In: *SIAM J. Math. Anal.* 17.3, pp. 704–719. ISSN: 0036-1410. DOI: 10.1137/0517050. URL: https://doi.org/10.1137/0517050.
- Young, A. (1954). "The application of approximate product integration to the numerical solution of integral equations". In: *Proc. Roy. Soc. London Ser. A* 224, pp. 561–573. ISSN: 0962-8444. DOI: 10.1098/rspa.1954.0180. URL: https://doi.org/10.1098/rspa.1954.0180.

An introduction to fractional calculus

Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

May, 2022

Where are we?

We know a general way to obtain FLMM methods of the form

$$y^{(n)} = T_{m-1}(t_n) + \tau^{\beta} \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^{\alpha} \sum_{j=0}^{n} \omega_{n-j} f(t_j, y^{(j)}),$$

- starting from the polynomials (ρ, σ) of an implicit order p method,

Where are we?

We know a general way to obtain FLMM methods of the form

$$y^{(n)} = T_{m-1}(t_n) + \tau^{\beta} \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^{\alpha} \sum_{j=0}^{n} \omega_{n-j} f(t_j, y^{(j)}),$$

- starting from the polynomials (ρ, σ) of an implicit order p method,
- we have seen how to compute the convolution coefficients ω_n,

Where are we?

We know a general way to obtain FLMM methods of the form

$$y^{(n)} = T_{m-1}(t_n) + \tau^{\beta} \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^{\alpha} \sum_{j=0}^{n} \omega_{n-j} f(t_j, y^{(j)}),$$

- starting from the polynomials (ρ, σ) of an implicit order p method,
- we have seen how to compute the convolution coefficients ω_n,
- we have seen how to compute the starting nodes $w_{n,j}$,

Where are we?

We know a general way to obtain FLMM methods of the form

$$y^{(n)} = T_{m-1}(t_n) + \tau^{\beta} \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^{\alpha} \sum_{j=0}^{n} \omega_{n-j} f(t_j, y^{(j)}),$$

- starting from the polynomials (ρ, σ) of an implicit order p method,
- we have seen how to compute the convolution coefficients ω_n,
- we have seen how to compute the starting nodes $w_{n,j}$,
- we need to discuss how we compute the starting values for a multi-step method,

Where are we?

We know a general way to obtain FLMM methods of the form

$$y^{(n)} = T_{m-1}(t_n) + \tau^{\beta} \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^{\alpha} \sum_{j=0}^{n} \omega_{n-j} f(t_j, y^{(j)}),$$

- ✅ starting from the polynomials (ρ, σ) of an implicit order p method,
- ✅ we have seen how to compute the convolution coefficients ω_n,
- ✅ we have seen how to compute the starting nodes $w_{n,j}$,
- 📋 we need to discuss how we compute the starting values for a multi-step method,
- 📋 we still need to discuss how we can efficiently treat the memory term.

Computing the starting values

To **initialize the computation** we need the values $y^{(0)}, \ldots, y^{(s)}$, $s+1$, $s = |\mathcal{A}| = \mathcal{A}_{p-1} \cup \{p-1\}$ with p the order of convergence of the FLMM, and
$\mathcal{A}_{p-1} = \{\nu \in \mathbb{R} \mid \nu = i + j\alpha, \quad i, j \in \mathbb{N}, \nu < p-1\}$.

Computing the starting values

To **initialize the computation** we need the values $y^{(0)}, \dots, y^{(s)}$, $s+1$, $s = |\mathcal{A}| = \mathcal{A}_{p-1} \cup \{p-1\}$ with p the order of convergence of the FLMM, and
$\mathcal{A}_{p-1} = \{\nu \in \mathbb{R} \mid \nu = i + j\alpha, \quad i, j \in \mathbb{N}, \nu < p-1\}$.

- We know $y^{(0)}$ from the initial condition, thus we have to solve for the remaining ones.

Computing the starting values

To **initialize the computation** we need the values $y^{(0)}, \dots, y^{(s)}$, $s+1$, $s = |\mathcal{A}| = \mathcal{A}_{p-1} \cup \{p-1\}$ with p the order of convergence of the FLMM, and
$\mathcal{A}_{p-1} = \{\nu \in \mathbb{R} \,|\, \nu = i + j\alpha, \quad i, j \in \mathbb{N}, \nu < p-1\}$.

- We know $y^{(0)}$ from the initial condition, thus we have to solve for the remaining ones.
- To avoid mixing methods we evaluate all the approximations at the same time by solving

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(s)} \end{bmatrix} = \begin{bmatrix} T_{m-1}(t_1) \\ T_{m-1}(t_2) \\ \vdots \\ T_{m-1}(t_s) \end{bmatrix} + \tau^\alpha \begin{bmatrix} (\omega_1 + w_{1,0}) f_0 \\ (\omega_2 + w_{2,0}) f_0 \\ \vdots \\ (\omega_s + w_{s,0}) f_0 \end{bmatrix} + \tau^\alpha \left(\Omega \otimes I + W \otimes I\right) \begin{bmatrix} f(t_1, y^{(1)}) \\ f(t_2, y^{(2)}) \\ \vdots \\ f(t_s, y^{(s)}) \end{bmatrix}$$

where

$$\Omega = \begin{bmatrix} \omega_0 & & & \\ \omega_1 & \omega_0 & & \\ \vdots & \vdots & \ddots & \\ \omega_{s-1} & \omega_{s-2} & \cdots & \omega_0 \end{bmatrix}, \quad W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,s} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,s} \\ \vdots & \vdots & \ddots & \vdots \\ w_{s,1} & w_{s,2} & \cdots & w_{s,s} \end{bmatrix}$$

Computing the starting values

To **initialize the computation** we need the values $y^{(0)}, \dots, y^{(s)}$, $s+1$, $s = |\mathcal{A}| = \mathcal{A}_{p-1} \cup \{p-1\}$ with p the order of convergence of the FLMM, and
$\mathcal{A}_{p-1} = \{\nu \in \mathbb{R} \,|\, \nu = i + j\alpha, \quad i, j \in \mathbb{N}, \nu < p-1\}$.

- We know $y^{(0)}$ from the initial condition, thus we have to solve for the remaining ones.
- To avoid mixing methods we evaluate all the approximations at the same time by solving

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(s)} \end{bmatrix} = \begin{bmatrix} T_{m-1}(t_1) \\ T_{m-1}(t_2) \\ \vdots \\ T_{m-1}(t_s) \end{bmatrix} + \tau^{\alpha} \begin{bmatrix} (\omega_1 + w_{1,0}) f_0 \\ (\omega_2 + w_{2,0}) f_0 \\ \vdots \\ (\omega_s + w_{s,0}) f_0 \end{bmatrix} + \tau^{\alpha} \left(\Omega \otimes I + W \otimes I\right) \begin{bmatrix} f(t_1, y^{(1)}) \\ f(t_2, y^{(2)}) \\ \vdots \\ f(t_s, y^{(s)}) \end{bmatrix}$$

- This will be in general an $s \times \dim(y^{(j)})$ nonlinear system that we need to solve before starting the iteration.

Computing the starting values

To **initialize the computation** we need the values $y^{(0)}, \dots, y^{(s)}$, $s+1$, $s = |\mathcal{A}| = \mathcal{A}_{p-1} \cup \{p-1\}$ with p the order of convergence of the FLMM, and
$\mathcal{A}_{p-1} = \{\nu \in \mathbb{R} \,|\, \nu = i + j\alpha, \quad i, j \in \mathbb{N}, \nu < p-1\}$.

- We know $y^{(0)}$ from the initial condition, thus we have to solve for the remaining ones.
- To avoid mixing methods we evaluate all the approximations at the same time by solving

$$\begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(s)} \end{bmatrix} = \begin{bmatrix} T_{m-1}(t_1) \\ T_{m-1}(t_2) \\ \vdots \\ T_{m-1}(t_s) \end{bmatrix} + \tau^{\alpha} \begin{bmatrix} (\omega_1 + w_{1,0}) f_0 \\ (\omega_2 + w_{2,0}) f_0 \\ \vdots \\ (\omega_s + w_{s,0}) f_0 \end{bmatrix} + \tau^{\alpha} \left(\Omega \otimes I + W \otimes I\right) \begin{bmatrix} f(t_1, y^{(1)}) \\ f(t_2, y^{(2)}) \\ \vdots \\ f(t_s, y^{(s)}) \end{bmatrix}$$

- This will be in general an $s \times \dim(y^{(j)})$ nonlinear system that we need to solve before starting the iteration.
- If the value of α is not very small, viz s is moderate, and the system of ODEs is moderate this is manageable.

Treating the memory term

If we compute the sum on the coefficients ω_j naively for

$$y^{(n)} = T_{m-1}(t_n) + \tau^\beta \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^\alpha \sum_{j=0}^{n-1} \omega_{n-j} f(t_j, y^{(j)}) + \tau^\alpha \omega_0 f(t_n, y^{(n)}),$$

we end up having a $O(N^2)$ cost! If we do not perform this task efficiently the numerical solution degenerates in an unworkable task as we either refine our grid or enlarge our computational domain.

Treating the memory term

If we compute the sum on the coefficients ω_j naively for

$$y^{(n)} = T_{m-1}(t_n) + \tau^\beta \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^\alpha \sum_{j=0}^{n-1} \omega_{n-j} f(t_j, y^{(j)}) + \tau^\alpha \omega_0 f(t_n, y^{(n)}),$$

we end up having a $O(N^2)$ cost! If we do not perform this task efficiently the numerical solution degenerates in an unworkable task as we either refine our grid or enlarge our computational domain.

- We can try to "forget" part of the lag-term,

Treating the memory term

If we compute the sum on the coefficients ω_j naively for

$$y^{(n)} = T_{m-1}(t_n) + \tau^\beta \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^\alpha \sum_{j=0}^{n-1} \omega_{n-j} f(t_j, y^{(j)}) + \tau^\alpha \omega_0 f(t_n, y^{(n)}),$$

we end up having a $O(N^2)$ cost! If we do not perform this task efficiently the numerical solution degenerates in an unworkable task as we either refine our grid or enlarge our computational domain.

- We can try to "forget" part of the lag-term,
- We can consider using a stretched grid towards t_0 to reduce N,

Treating the memory term

If we compute the sum on the coefficients ω_j naively for

$$y^{(n)} = T_{m-1}(t_n) + \tau^{\beta} \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^{\alpha} \sum_{j=0}^{n-1} \omega_{n-j} f(t_j, y^{(j)}) + \tau^{\alpha} \omega_0 f(t_n, y^{(n)}),$$

we end up having a $O(N^2)$ cost! If we do not perform this task efficiently the numerical solution degenerates in an unworkable task as we either refine our grid or enlarge our computational domain.

- We can try to "forget" part of the lag-term,
- We can consider using a stretched grid towards t_0 to reduce N,
- We can try an approach with nested meshes to reduce the load,

Treating the memory term

If we compute the sum on the coefficients ω_j naively for

$$y^{(n)} = T_{m-1}(t_n) + \tau^\beta \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^\alpha \sum_{j=0}^{n-1} \omega_{n-j} f(t_j, y^{(j)}) + \tau^\alpha \omega_0 f(t_n, y^{(n)}),$$

we end up having a $O(N^2)$ cost! If we do not perform this task efficiently the numerical solution degenerates in an unworkable task as we either refine our grid or enlarge our computational domain.

- We can try to "forget" part of the lag-term,
- We can consider using a stretched grid towards t_0 to reduce N,
- We can try an approach with nested meshes to reduce the load,
- We can exploit the fact that this is a convolution and adopt some FFT tricks.

The FFT trick (Hairer, Lubich, and Schlichte 1985)

The treatment remains the same indifferently for both PI and FLMM method, let us focus here on the generic formulation

$$y^{(n)} = \phi_n + \sum_{j=0}^{n} c_{n-j} f_j.$$

The FFT trick (Hairer, Lubich, and Schlichte 1985)

The treatment remains the same indifferently for both PI and FLMM method, let us focus here on the generic formulation

$$y^{(n)} = \phi_n + \sum_{j=0}^{n} c_{n-j} f_j.$$

- Let r be a moderate number of step, e.g., $r = 2^k$ for a small k, we compute the first steps directly

$$y^{(n)} = \phi_n + \sum_{j=0}^{n} c_{n-j} f_j, \quad n = 0, 1, \ldots, r-1.$$

The FFT trick (Hairer, Lubich, and Schlichte 1985)

The treatment remains the same indifferently for both PI and FLMM method, let us focus here on the generic formulation

$$y^{(n)} = \phi_n + \sum_{j=0}^{n} c_{n-j} f_j.$$

- Let r be a moderate number of step, e.g., $r = 2^k$ for a small k, we compute the first steps directly

$$y^{(n)} = \phi_n + \sum_{j=0}^{n} c_{n-j} f_j, \quad n = 0, 1, \ldots, r-1.$$

- If we now want to compute the next r approximations we can separate the lag term as

$$y^{(n)} = \phi_n + \sum_{j=0}^{r-1} c_{n-j} f_j + \sum_{j=r}^{n} c_{n-j} f_j, \quad n \in \{r, r+1, \ldots, 2r-1\}.$$

The FFT trick (Hairer, Lubich, and Schlichte 1985)

- If we now want to compute the next r approximations we can separate the lag term as

$$y^{(n)} = \phi_n + \sum_{j=0}^{r-1} c_{n-j} f_j + \sum_{j=r}^{n} c_{n-j} f_j, \quad n \in \{r, r+1, \ldots, 2r-1\}.$$

The FFT trick (Hairer, Lubich, and Schlichte 1985)

- If we now want to compute the next r approximations we can separate the lag term as

$$y^{(n)} = \phi_n + \sum_{j=0}^{r-1} c_{n-j} f_j + \sum_{j=r}^{n} c_{n-j} f_j, \quad n \in \{r, r+1, \ldots, 2r-1\}.$$

We can use FFT!

If we call $S_r(n, 0, r-1) = \sum_{j=0}^{r-1} c_{n-j} f_j$, $n \in \{r, r+1, \ldots, 2r-1\}$, the set of partial sums each of length r we can evaluate them with FFT in $O(2r \log_2(2r))$.

The FFT trick (Hairer, Lubich, and Schlichte 1985)

- If we now want to compute the next r approximations we can separate the lag term as

$$y^{(n)} = \phi_n + \sum_{j=0}^{r-1} c_{n-j} f_j + \sum_{j=r}^{n} c_{n-j} f_j, \quad n \in \{r, r+1, \ldots, 2r-1\}.$$

- We can apply the same process recursively if we double every time-interval under consideration

$$y^{(n)} = \phi_n + \sum_{j=0}^{2r-1} c_{n-j} f_j + \sum_{j=2r}^{n} c_{n-j} f_j, \quad n \in \{2r, 2r+1, \ldots, 3r-1\},$$

$$y^{(n)} = \phi_n + \sum_{j=0}^{2r-1} c_{n-j} f_j + \sum_{j=2r}^{3r-1} c_{n-j} f_j + \sum_{j=3r}^{n} c_{n-j} f_j, \quad n \in \{3r, 3r+1, \ldots, 4r-1\},$$

The FFT trick (Hairer, Lubich, and Schlichte 1985)

We can use FFT!

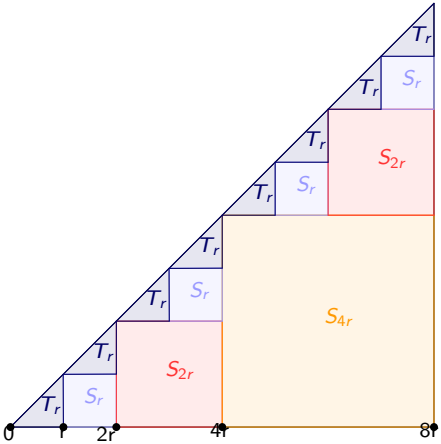
If we call $S_{2r}(n, 0, 2r-1) = \sum_{j=0}^{2r-1} c_{n-j} f_j$, and $S_r(n, 2r, 3r-1) = \sum_{j=2r}^{3r-1} c_{n-j} f_j$ the set of partial sums of lengths $2r$ and r we can evaluate them with FFT in $O(4r \log_2(4r))$ and $O(2r \log_2(2r))$ respectively.

- We can apply the same process recursively if we double every time-interval under consideration

$$y^{(n)} = \phi_n + \sum_{j=0}^{2r-1} c_{n-j} f_j + \sum_{j=2r}^{n} c_{n-j} f_j, \quad n \in \{2r, 2r+1, \ldots, 3r-1\},$$

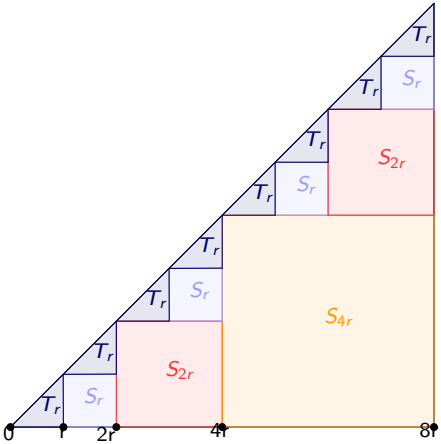
$$y^{(n)} = \phi_n + \sum_{j=0}^{2r-1} c_{n-j} f_j + \sum_{j=2r}^{3r-1} c_{n-j} f_j + \sum_{j=3r}^{n} c_{n-j} f_j, \quad n \in \{3r, 3r+1, \ldots, 4r-1\},$$

The FFT trick (Hairer, Lubich, and Schlichte 1985)



- We can iterate the process for the $4r$ approximations in the interval $n \in \{4r, \ldots, 8r-1\}$, together with the partial sums $S_{4r}(n, 0, 4r-1)$, $S_{2r}(n, 4r, 6r-1)$, $S_r(n, 6r, 7r-1)$ that can be evaluated in $O(8r\log_2(8r))$, $O(4r\log_2(4r))$ and $O(2r\log_2(2r))$ respectively,

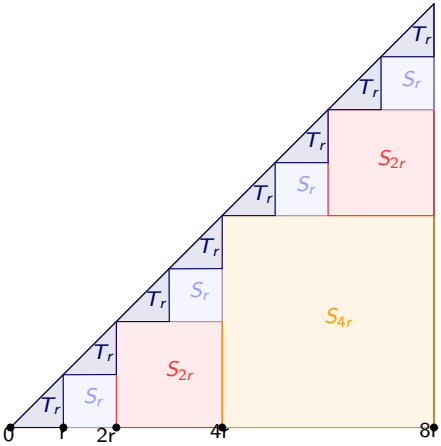
The FFT trick (Hairer, Lubich, and Schlichte 1985)



- We can iterate the process for the $4r$ approximations in the interval $n \in \{4r, \ldots, 8r-1\}$, together with the partial sums $S_{4r}(n, 0, 4r-1)$, $S_{2r}(n, 4r, 6r-1)$, $S_r(n, 6r, 7r-1)$ that can be evaluated in $O(8r\log_2(8r))$, $O(4r\log_2(4r))$ and $O(2r\log_2(2r))$ respectively,
- At each level we have to complete the recursion by computing

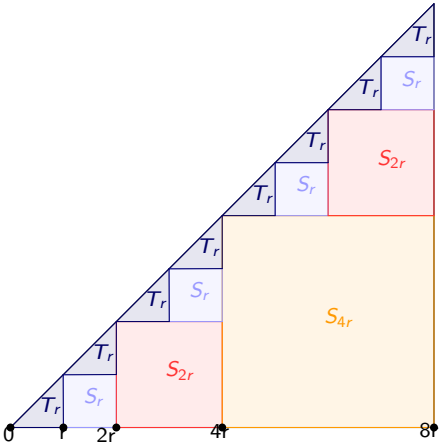
$$T_r(p, n) = \sum_{j=p}^{n} c_{n-j} f_j, \; p = \ell r,$$
$$n \in \{\ell r, \ell r + 1, \ldots, (\ell+1)r - 1\},$$
$$\ell = 0, 1, 2, \ldots.$$

The FFT trick (Hairer, Lubich, and Schlichte 1985)



To determine the whole cost we just have to sum the various components

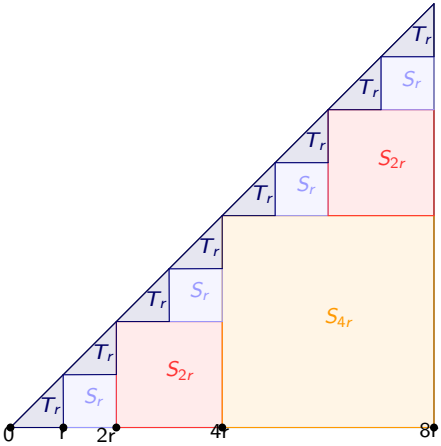
The FFT trick (Hairer, Lubich, and Schlichte 1985)



To determine the whole cost we just have to sum the various components

- Assume that $N = 2^{n_t}$

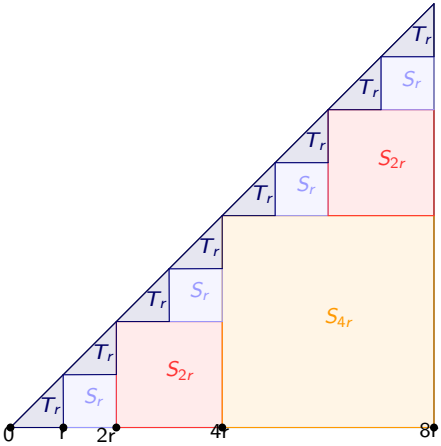
The FFT trick (Hairer, Lubich, and Schlichte 1985)



To determine the whole cost we just have to sum the various components

- Assume that $N = 2^{n_t}$
- $O(N \log_2 N)$ for S_{4r},

The FFT trick (Hairer, Lubich, and Schlichte 1985)

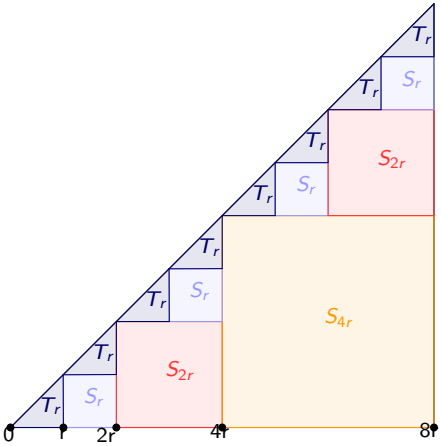


To determine the whole cost we just have to sum the various components

- Assume that $N = 2^{n_t}$
- $O(N \log_2 N)$ for S_{4r},

$+$ $O(N/2 \log_2 N/2)$ for 2 S_{2r}

The FFT trick (Hairer, Lubich, and Schlichte 1985)



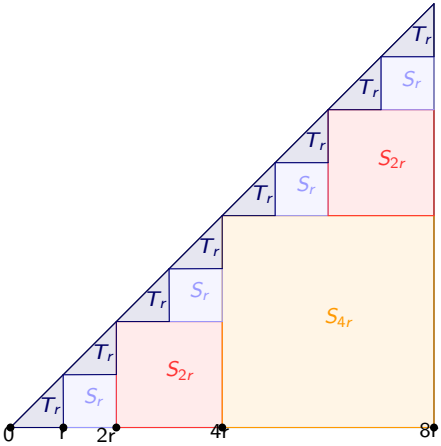
To determine the whole cost we just have to sum the various components

- Assume that $N = 2^{n_t}$
- $O(N \log_2 N)$ for S_{4r},

$+$ $O(N/2 \log_2 N/2)$ for 2 S_{2r}

$+$ $O(N/4 \log_2 N/4)$ for 4 S_r

The FFT trick (Hairer, Lubich, and Schlichte 1985)



To determine the whole cost we just have to sum the various components

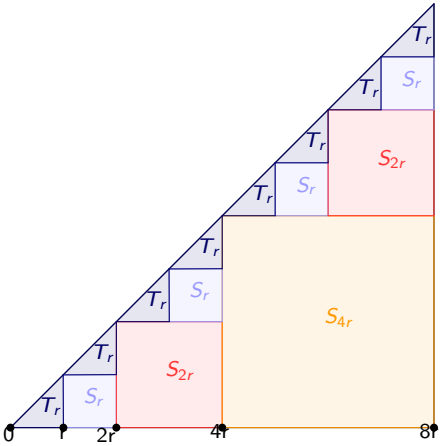
- Assume that $N = 2^{n_t}$
- $O(N \log_2 N)$ for S_{4r},

$+$ $O(N/2 \log_2 N/2)$ for 2 S_{2r}

$+$ $O(N/4 \log_2 N/4)$ for 4 S_r

$+$ $r(r+1)/2$ for the N/r convolutions T_r

The FFT trick (Hairer, Lubich, and Schlichte 1985)



To determine the whole cost we just have to sum the various components

- Assume that $N = 2^{n_t}$
- $O(N \log_2 N)$ for S_{4r},

$+$ $O(N/2 \log_2 N/2)$ for 2 S_{2r}

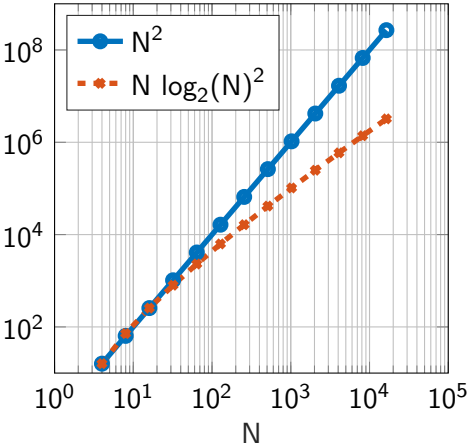
$+$ $O(N/4 \log_2 N/4)$ for 4 S_r

$+$ $r(r+1)/2$ for the N/r convolutions T_r

- In general:

$$N \log_2 N + 2\frac{N}{2} \log_2 \frac{N}{2} + 4\frac{N}{4} \log_2 \frac{N}{4} + \cdots$$
$$+ p\frac{N}{p} \log_2 \frac{N}{p} + \frac{N}{r}\frac{r(r+1)}{2}, \quad p = \frac{N}{2r}$$

The FFT trick (Hairer, Lubich, and Schlichte 1985)



To determine the whole cost we just have to sum the various components

- Assume that $N = 2^{n_t}$
- $O(N \log_2 N)$ for S_{4r},

$+$ $O(N/2 \log_2 N/2)$ for 2 S_{2r}

$+$ $O(N/4 \log_2 N/4)$ for 4 S_r

$+$ $r(r+1)/2$ for the N/r convolutions T_r

- In general:

$$\sum_{j=0}^{\log_2 p} N \log_2 \frac{N}{2^j} + N\frac{r+1}{2} = O(N(\log_2 N)^2).$$

Short-memory principle (Ford and Simpson 2001)

We can try to use a "fixed memory length" to reduce the computational (and memory) load.

$$
\begin{aligned}
y(t_{n+1}) = y(t_n) &+ \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\, \mathrm{d}\tau \\
&+ \frac{1}{\Gamma(\alpha)} \int_{0}^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau, \quad \alpha \in (0, 1).
\end{aligned}
$$

Short-memory principle (Ford and Simpson 2001)

We can try to use a "fixed memory length" to reduce the computational (and memory) load.

$$
\begin{aligned}
y(t_{n+1}) = y(t_n) &+ \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\, \mathrm{d}\tau \\
&+ \frac{1}{\Gamma(\alpha)} \int_{0}^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau, \quad \alpha \in (0, 1).
\end{aligned}
$$

Let us introduce now a fixed window T_M of memory, then

$$
\begin{aligned}
E = &\left| \frac{1}{\Gamma(\alpha)} \int_{0}^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau \right| \\
\leq &\frac{M}{\Gamma(\alpha)} \left| \int_{0}^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1})\, \mathrm{d}\tau \right|
\end{aligned}
$$

Short-memory principle (Ford and Simpson 2001)

We can try to use a "fixed memory length" to reduce the computational (and memory) load.

$$
\begin{aligned}
y(t_{n+1}) = y(t_n) &+ \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\, \mathrm{d}\tau \\
&+ \frac{1}{\Gamma(\alpha)} \int_{0}^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau, \quad \alpha \in (0,1).
\end{aligned}
$$

Let us introduce now a fixed window T_M of memory, then

$$
\begin{aligned}
E &= \left| \frac{1}{\Gamma(\alpha)} \int_{0}^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau \right| \\
&\leq \frac{M}{\Gamma(\alpha)} \left| \int_{0}^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1})\, \mathrm{d}\tau \right| \\
&= \frac{M}{\alpha\Gamma(\alpha)} \left| (\tau + T_M)^{\alpha} - t_{n+1}^{\alpha} - T^{\alpha} + t_n^{\alpha} \right|
\end{aligned}
$$

Short-memory principle (Ford and Simpson 2001)

We can try to use a "fixed memory length" to reduce the computational (and memory) load.

$$y(t_{n+1}) = y(t_n) + \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\, \mathrm{d}\tau$$
$$+ \frac{1}{\Gamma(\alpha)} \int_0^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau, \quad \alpha \in (0, 1).$$

Let us introduce now a fixed window T_M of memory, then

$$\begin{aligned} E = & \left| \frac{1}{\Gamma(\alpha)} \int_0^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau \right| \\ \leq & \frac{M}{\Gamma(\alpha)} \left| \int_0^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1})\, \mathrm{d}\tau \right| \\ = & \frac{M}{\Gamma(\alpha)} \left| \int_{T_M}^{T_M + \tau} z^{\alpha-1}\, \mathrm{d}z - \int_{t_n}^{t_{n+1}} z^{\alpha-1}\, \mathrm{d}z \right| \end{aligned}$$

Short-memory principle (Ford and Simpson 2001)

We can try to use a "fixed memory length" to reduce the computational (and memory) load.

$$y(t_{n+1}) = y(t_n) + \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\, \mathrm{d}\tau$$
$$+ \frac{1}{\Gamma(\alpha)} \int_{0}^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau, \quad \alpha \in (0, 1).$$

Let us introduce now a fixed window T_M of memory, then

$$\begin{aligned} E =& \left| \frac{1}{\Gamma(\alpha)} \int_{0}^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau \right| \\ \leq& \frac{M}{\Gamma(\alpha)} \left| \int_{0}^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1})\, \mathrm{d}\tau \right| \\ =& \frac{M}{\Gamma(\alpha)} \left| \int_{T_M}^{T_M + \tau} z^{\alpha-1}\, \mathrm{d}z - \int_{t_n}^{t_{n+1}} z^{\alpha-1}\, \mathrm{d}z \right| (Mean Value Theorem) \end{aligned}$$

Short-memory principle (Ford and Simpson 2001)

We can try to use a "fixed memory length" to reduce the computational (and memory) load.

$$y(t_{n+1}) = y(t_n) + \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\, \mathrm{d}\tau$$
$$+ \frac{1}{\Gamma(\alpha)} \int_0^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau, \quad \alpha \in (0, 1).$$

Let us introduce now a fixed window T_M of memory, then

$$\begin{aligned} E &= \left| \frac{1}{\Gamma(\alpha)} \int_0^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau \right| \\ &\leq \frac{M}{\Gamma(\alpha)} \left| \int_0^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1})\, \mathrm{d}\tau \right| \\ &= \frac{M}{\Gamma(\alpha)} \left| (z_1^*)^{\alpha-1}\tau - (z_2^*)^{\alpha-1}\tau \right| \quad z_1^* \in [T_M, T_M + \tau],\ z_2^* \in [t_n, t_{n+1}] \end{aligned}$$

Short-memory principle (Ford and Simpson 2001)

We can try to use a "fixed memory length" to reduce the computational (and memory) load.

$$y(t_{n+1}) = y(t_n) + \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\, \mathrm{d}\tau$$
$$+ \frac{1}{\Gamma(\alpha)} \int_0^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau, \quad \alpha \in (0, 1).$$

Let us introduce now a fixed window T_M of memory, then

$$E = \left| \frac{1}{\Gamma(\alpha)} \int_0^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau \right|$$
$$\leq \frac{M}{\Gamma(\alpha)} \left| \int_0^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1})\, \mathrm{d}\tau \right|$$
$$< \frac{M}{\Gamma(\alpha)} T_M^{\alpha-1} \tau, \qquad \alpha \in (0, 1).$$

Short-memory principle (Ford and Simpson 2001)

We can try to use a "fixed memory length" to reduce the computational (and memory) load.

$$y(t_{n+1}) = y(t_n) + \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\, \mathrm{d}\tau$$
$$+ \frac{1}{\Gamma(\alpha)} \int_0^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau, \quad \alpha \in (0, 1).$$

Let us introduce now a fixed window T_M of memory, then

$$E = \left| \frac{1}{\Gamma(\alpha)} \int_0^{t_n - T_M} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau \right| < \frac{M}{\Gamma(\alpha)} T_M^{\alpha-1} \tau, \quad \alpha \in (0, 1).$$

Short-memory principle (Ford and Simpson 2001)

We can try to use a "fixed memory length" to reduce the computational (and memory) load.

$$y(t_{n+1}) = y(t_n) + \frac{1}{\Gamma(\alpha)} \int_{t_n}^{t_{n+1}} (t_{n+1} - \tau)^{\alpha-1} f(\tau, x(\tau))\, \mathrm{d}\tau$$
$$+ \frac{1}{\Gamma(\alpha)} \int_{0}^{t_n} ((t_{n+1} - \tau)^{\alpha-1} - (t_n - \tau)^{\alpha-1}) f(\tau, y(\tau))\, \mathrm{d}\tau, \quad \alpha \in (0,1).$$

Let us introduce now a fixed window T_M of memory, then If we have a global error bound E_{global} with step-length τ we just need to choose

$$T_M > \left(\frac{M}{\Gamma(\alpha) E_{\text{global}}} \right)^{1/1-\alpha}, \quad \alpha \in (0,1),$$

while if we have a local error bound E_{local}

$$T_M > \left(\frac{M\tau}{\Gamma(\alpha) E_{\text{local}}} \right)^{1/1-\alpha}, \quad \alpha \in (0,1).$$

Short-memory principle (Ford and Simpson 2001)

- In case $\alpha \in (0, 1)$ the short memory method with fixed length can be effective and the length T is independent of the full interval of integration.

Short-memory principle (Ford and Simpson 2001)

- In case $\alpha \in (0,1)$ the short memory method with fixed length can be effective and the length T is independent of the full interval of integration.
- Similar bounds can be written for the case $\alpha > 1$, that is

$$E < \frac{M}{\Gamma(\alpha)}(t_{n+1}^{\alpha} - T_M^{\alpha-1})\tau, \quad \alpha > 1.$$

But now to preserve the order of accuracy, we must choose

$$T_M^{\alpha-1} > t_{n+1}^{\alpha-1} - \frac{E_{\text{global}}\Gamma(\alpha)}{M}, \quad \alpha > 1,$$

that we will make us lose all the computational gain.

Short-memory principle (Ford and Simpson 2001)

- ☺ In case $\alpha \in (0,1)$ the short memory method with fixed length can be effective and the length T is independent of the full interval of integration.
- ☹ Similar bounds can be written for the case $\alpha > 1$, that is

$$E < \frac{M}{\Gamma(\alpha)}(t_{n+1}^{\alpha} - T_M^{\alpha-1})\tau, \quad \alpha > 1.$$

But now to preserve the order of accuracy, we must choose

$$T_M^{\alpha-1} > t_{n+1}^{\alpha-1} - \frac{E_{\text{global}}\Gamma(\alpha)}{M}, \quad \alpha > 1,$$

that we will make us lose all the computational gain.

The idea can be refined by using *nested meshes*.

Nested meshes

Zeroing out the memory term is too drastic, we may want to relax this.

Nested meshes

Zeroing out the memory term is too drastic, we may want to relax this.

Scaling properties

$$I^{\alpha}_{[0,t]} f(t) = \int_0^t \frac{f(\tau)}{(t-\tau)^{1-\alpha}} \, \mathrm{d}\tau$$

Nested meshes

Zeroing out the memory term is too drastic, we may want to relax this.

Scaling properties

$$I^{\alpha}_{[0,t]} f(wt) = \int_0^t \frac{f(\tau)}{(wt-\tau)^{1-\alpha}} \, \mathrm{d}\tau$$

Nested meshes

Zeroing out the memory term is too drastic, we may want to relax this.

Scaling properties

$$I^{\alpha}_{[0,t]} f(wt) = w^{\alpha} \int_0^t \frac{f(w\tau)}{(t-\tau)^{1-\alpha}} \, \mathrm{d}\tau$$

Nested meshes

Zeroing out the memory term is too drastic, we may want to relax this.

Scaling properties

Given $p \in \mathbb{N}$ we then have

$$I^{\alpha}_{[0,t]} f(w^p t) = w^{p\alpha} \int_0^t \frac{f(w^p \tau)}{(t-\tau)^{1-\alpha}} \, \mathrm{d}\tau$$

Nested meshes

Zeroing out the memory term is too drastic, we may want to relax this.

Scaling properties

Given $p \in \mathbb{N}$ we then have

$$I^{\alpha}_{[0,t]} f(w^p t) = w^{p\alpha} \int_0^t \frac{f(w^p \tau)}{(t-\tau)^{1-\alpha}} \, \mathrm{d}\tau$$

- We can use the weight on the mesh

$$\Omega^{\alpha}_{\tau} f(n\tau) \approx I^{\alpha}_{[0,t]} f(n\tau), \text{ step length } \tau$$

to compute

$$\Omega^{\alpha}_{w^p\tau} f(nw^p\tau) \approx I^{\alpha}_{[0,t]} f(nw^p\tau), \text{ step length } w^p\tau$$

Nested meshes

Zeroing out the memory term is too drastic, we may want to relax this.

Scaling properties

Given $p \in \mathbb{N}$ we then have

$$I^{\alpha}_{[0,t]} f(w^p t) = w^{p\alpha} \int_0^t \frac{f(w^p \tau)}{(t-\tau)^{1-\alpha}} \, \mathrm{d}\tau$$

- We can use the weight on the mesh

$$\Omega^{\alpha}_{\tau} f(n\tau) \approx I^{\alpha}_{[0,t]} f(n\tau), \text{ step length } \tau$$

  to compute

$$\Omega^{\alpha}_{w^p\tau} f(nw^p\tau) \approx I^{\alpha}_{[0,t]} f(nw^p\tau), \text{ step length } w^p\tau$$

- In summary for any $p \in \mathbb{N}$ we get

$$\Omega^{\alpha}_{\tau} f(n\tau) = \sum_{j=0}^{n} \omega_{n-j} f(j\tau) \;\Leftrightarrow\; \Omega^{\alpha}_{w^p\tau} f(nw^p\tau) = w^{p\alpha} \sum_{j=0}^{n} \omega_{n-j} f(jw^p\tau).$$

Nested meshes (Ford and Simpson 2001)

Nested mesh

Given $\tau \in \mathbb{R}^+$, the mesh $M_\tau = \{\tau n,\ n \in \mathbb{N}\}$. Selected $w, r, p \in \mathbb{N}$, $w > 0$, $r > p$, we have $M_{w^p\tau} \supset M_{w^r\tau}$ and we decompose the interval as

$$[0, t] = [0, t - w^m T] \cup [t - w^m T, t - w^{m-1} T] \cup \cdots \cup [t - wT, t - T] \cup [t - T, t]$$

for $m \in \mathbb{N}$ the smallest integer such that $t < w^{m+1} T$.

Nested meshes (Ford and Simpson 2001)

Nested mesh

Given $\tau \in \mathbb{R}^+$, the mesh $M_\tau = \{\tau n,\ n \in \mathbb{N}\}$. Selected $w, r, p \in \mathbb{N}$, $w > 0$, $r > p$, we have $M_{w^p\tau} \supset M_{w^r\tau}$ and we decompose the interval as

$$[0, t] = [0, t - w^m T] \cup [t - w^m T, t - w^{m-1} T] \cup \cdots \cup [t - wT, t - T] \cup [t - T, t]$$

for $m \in \mathbb{N}$ the smallest integer such that $t < w^{m+1} T$.

- This links the **scaling property** with the singularity of the type $1/(t - \tau)^{1-\alpha}$ suggesting that we should distribute the computational effort logarithmically, and not uniformly.

Nested meshes (Ford and Simpson 2001)

Nested mesh

Given $\tau \in \mathbb{R}^+$, the mesh $M_\tau = \{\tau n,\ n \in \mathbb{N}\}$. Selected $w, r, p \in \mathbb{N}$, $w > 0$, $r > p$, we have $M_{w^p\tau} \supset M_{w^r\tau}$ and we decompose the interval as

$$[0, t] = [0, t - w^m T] \cup [t - w^m T, t - w^{m-1} T] \cup \cdots \cup [t - wT, t - T] \cup [t - T, t]$$

for $m \in \mathbb{N}$ the smallest integer such that $t < w^{m+1} T$.

- This links the **scaling property** with the singularity of the type $1/(t-\tau)^{1-\alpha}$ suggesting that we should distribute the computational effort logarithmically, and not uniformly.
- We rewrite our integral as

$$I^\alpha_{[0,t]} f(t) = I^\alpha_{[t-T,t]} f(t) + \sum_{i=0}^{m-1} I^\alpha_{[t-w^{i+1}T, t-w^i T]} f(t) + I^\alpha_{[0, t-w^m T]} f(t)$$

Nested meshes (Ford and Simpson 2001)

Nested mesh

Given $\tau \in \mathbb{R}^+$, the mesh $M_\tau = \{\tau n,\ n \in \mathbb{N}\}$. Selected $w, r, p \in \mathbb{N}$, $w > 0$, $r > p$, we have $M_{w^p\tau} \supset M_{w^r\tau}$ and we decompose the interval as

$$[0, t] = [0, t - w^m T] \cup [t - w^m T, t - w^{m-1} T] \cup \cdots \cup [t - wT, t - T] \cup [t - T, t]$$

for $m \in \mathbb{N}$ the smallest integer such that $t < w^{m+1} T$.

- This links the **scaling property** with the singularity of the type $1/(t-\tau)^{1-\alpha}$ suggesting that we should distribute the computational effort logarithmically, and not uniformly.
- We rewrite our integral using the scaling property as

$$I^\alpha_{[0,t]} f(t) = I^\alpha_{[t-T,t]} f(t) + \sum_{i=0}^{m-1} w^{i\alpha} I^\alpha_{[t-wT,t-T]} f(w^i t) + w^{m\alpha} I^\alpha_{[0,t-T]} f(w^m t).$$

Nested meshes (Ford and Simpson 2001)

In the discrete approximation of

$$I^{\alpha}_{[0,t]} f(t) = I^{\alpha}_{[t-T,t]} f(t) + \sum_{i=0}^{m-1} w^{i\alpha} I^{\alpha}_{[t-wT,t-T]} f(w^i t) + w^{m\alpha} I^{\alpha}_{[0,t-T]} f(w^m t).$$

we approximate

$$\Omega^{\alpha}_{\tau,[t-w^{i+1}T,t-w^i T]} f(t) \approx \Omega^{\alpha}_{w^i\tau,[t-w^{i+1}T,t-w^i T]} f(t)$$

and substitute

$$w^{i\alpha} \Omega^{\alpha}_{\tau,[t-wT,t-T]} f(t) = \Omega^{\alpha}_{w^i\tau,[t-w^{i+1}T,t-w^i T]} f(t).$$

Nested meshes (Ford and Simpson 2001)

In the discrete approximation of

$$I^{\alpha}_{[0,t]} f(t) = I^{\alpha}_{[t-T,t]} f(t) + \sum_{i=0}^{m-1} w^{i\alpha} I^{\alpha}_{[t-wT,t-T]} f(w^i t) + w^{m\alpha} I^{\alpha}_{[0,t-T]} f(w^m t).$$

we approximate

$$\Omega^{\alpha}_{\tau,[t-w^{i+1}T,t-w^i T]} f(t) \approx \Omega^{\alpha}_{w^i\tau,[t-w^{i+1}T,t-w^i T]} f(t)$$

and substitute

$$w^{i\alpha} \Omega^{\alpha}_{\tau,[t-wT,t-T]} f(t) = \Omega^{\alpha}_{w^i\tau,[t-w^{i+1}T,t-w^i T]} f(t).$$

Theorem (Ford and Simpson 2001, Theorem 1)

The nested mesh scheme preserves the order of the underlying quadrature rule on which it is based.

Nested meshes (Ford and Simpson 2001)

In the discrete approximation of

$$I^{\alpha}_{[0,t]} f(t) = I^{\alpha}_{[t-T,t]} f(t) + \sum_{i=0}^{m-1} w^{i\alpha} I^{\alpha}_{[t-wT,t-T]} f(w^i t) + w^{m\alpha} I^{\alpha}_{[0,t-T]} f(w^m t).$$

we approximate

$$\Omega^{\alpha}_{\tau,[t-w^{i+1}T,t-w^i T]} f(t) \approx \Omega^{\alpha}_{w^i\tau,[t-w^{i+1}T,t-w^i T]} f(t)$$

and substitute

$$w^{i\alpha} \Omega^{\alpha}_{\tau,[t-wT,t-T]} f(t) = \Omega^{\alpha}_{w^i\tau,[t-w^{i+1}T,t-w^i T]} f(t).$$

Theorem (Ford and Simpson 2001, Theorem 1)

The nested mesh scheme preserves the order of the underlying quadrature rule on which it is based.

Proof. For integration over a fixed interval $[0, t]$ the choice of T fixes (independent of h) the number of subranges over which the integral is evaluated, on each of them we have en error $O(h^p)$.

Nested meshes (Ford and Simpson 2001)

- The first benefit is that we evaluate a fixed number of quadrature coefficients and then re-use them on all successive intervals,

Nested meshes (Ford and Simpson 2001)

- The first benefit is that we evaluate a fixed number of quadrature coefficients and then re-use them on all successive intervals,
- This approach cost $O(w^m)$ with respect to $O(w^{2m})$ of the full method,

Nested meshes (Ford and Simpson 2001)

- The first benefit is that we evaluate a fixed number of quadrature coefficients and then re-use them on all successive intervals,
- This approach cost $O(w^m)$ with respect to $O(w^{2m})$ of the full method,
- We could use linear extrapolation techniques to improve the results.

Nested meshes (Ford and Simpson 2001)

- The first benefit is that we evaluate a fixed number of quadrature coefficients and then re-use them on all successive intervals,
- This approach cost $O(w^m)$ with respect to $O(w^{2m})$ of the full method,
- We could use linear extrapolation techniques to improve the results.
- Selecting the various parameter may need a bit of tuning.

Available codes

With respect to the ordinary case for which there exists many reliable and high-performance codes, the choices for computing the solution of fractional differential equation is much more *sparse*.

- From (Garrappa 2018)
  - `FDE_PI1_Ex.m` - Explicit Product-Integration of rectanguar type
  - `FDE_PI1_Im.m` - Implicit Product-Integration of rectanguar type
  - `FDE_PI2_Im.m` - Implicit Product-Integration of trapezoidal type
  - `FDE_PI12_PC.m` - Product-Integration with predictor-corrector
- From (Garrappa 2015)
  - FLMM2 Matlab code - Three implicit second order Fractional Linear Multistep Methods.

A remark

All these methods use direct-solver for the Newton method inside them, there is space to make improvement on the solution strategies. Furthermore, a challenge that yet remains: can we find a strategy that combines the convolution features and savings on the memory?

What do we have now

We know a general way to obtain FLMM methods of the form

$$y^{(n)} = T_{m-1}(t_n) + \tau^{\beta} \sum_{j=0}^{s} w_{n,j} f(t_j, y^{(j)}) + \tau^{\alpha} \sum_{j=0}^{n} \omega_{n-j} f(t_j, y^{(j)}),$$

- starting from the polynomials (ρ, σ) of an implicit order p method,
- we have seen how to compute the convolution coefficients ω_n,
- we have seen how to compute the starting nodes $w_{n,j}$,
- we know how we can compute the starting values for a multi-step method by solving a nonlinear system with Newton,
- we have some hints on how we can efficiently treat the memory term.

A worked out example

Let us write everything for a case, let us start from the 2nd order BDF formula for ODEs

$$y^{(n+2)} - \frac{4}{3}y^{(n+1)} + \frac{1}{3}y^{(n)} = \frac{2}{3}\tau f_{n+2},$$

A worked out example

Let us write everything for a case, let us start from the 2nd order BDF formula for ODEs

$$y^{(n+2)} - \frac{4}{3}y^{(n+1)} + \frac{1}{3}y^{(n)} = \frac{2}{3}\tau f_{n+2},$$

- First of all we write down the (ρ, σ) polynomials defining the scheme:

$$\rho(\zeta) = \zeta^2 - \frac{4}{3}\zeta + \frac{1}{3}, \qquad \sigma(\zeta) = \frac{2}{3}\zeta^2.$$

A worked out example

Let us write everything for a case, let us start from the 2nd order BDF formula for ODEs

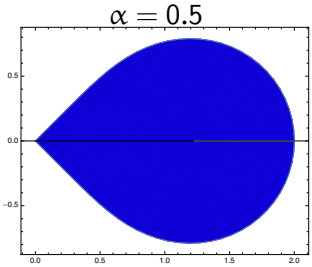
$$y^{(n+2)} - \frac{4}{3}y^{(n+1)} + \frac{1}{3}y^{(n)} = \frac{2}{3}\tau f_{n+2},$$

- First of all we write down the (ρ, σ) polynomials defining the scheme:

$$\rho(\zeta) = \zeta^2 - \frac{4}{3}\zeta + \frac{1}{3}, \qquad \sigma(\zeta) = \frac{2}{3}\zeta^2.$$

- Then we compute the generating function $\omega(\zeta)$

$$\omega(\zeta) = \frac{\rho(1/\zeta)}{\sigma(1/\zeta)} = \frac{2}{3\left(1 - 4\zeta/3 + \zeta^2/3\right)}.$$



$\{1/\omega(\zeta)^\alpha \ : \ |\zeta| \leq 1\}$

A worked out example

Now we need to expand the convolution coefficients of

$$\omega^\alpha(\zeta) = (\omega(\zeta))^\alpha = \frac{2^\alpha}{3^\alpha}(1 - {}^{4\zeta}/_{3} + {}^{\zeta^2}/_{3})^{-\alpha}.$$

A worked out example

Now we need to expand the convolution coefficients of

$$\omega^\alpha(\zeta) = (\omega(\zeta))^\alpha = \frac{2^\alpha}{3^\alpha}(1 - 4\zeta/3 + \zeta^2/3)^{-\alpha}.$$

Theorem (Henrici 1974, Theorem 1.6c, p. 42)

Let $\phi(\zeta) = 1 + \sum_{n=1}^{+\infty} a_n \zeta^n$ be a formal power series. Then for any $\alpha \in \mathbb{C}$, we have

$$(\phi(\zeta))^\alpha = \sum_{n=0}^{+\infty} v_n^{(\alpha)} \zeta^n,$$

where coefficients $v_n^{(\alpha)}$ can be evaluated recursively as

$$v_0^{(\alpha)} = 1, \qquad v_n^{(\alpha)} = \sum_{j=1}^{n} \left(\frac{(\alpha+1)j}{n} - 1 \right) a_j v_{n-j}^{(\alpha)}$$

A worked out example

Now we need to expand the convolution coefficients of

$$\omega^\alpha(\zeta) = (\omega(\zeta))^\alpha = \frac{2^\alpha}{3^\alpha}(1 - {}^{4\zeta}/_{3} + {}^{\zeta^2}/_{3})^{-\alpha}.$$

- $\omega_n = {}^{2^\alpha}/_{3^\alpha}\tilde{\omega}_n,$

A worked out example

Now we need to expand the convolution coefficients of

$$\omega^\alpha(\zeta) = (\omega(\zeta))^\alpha = \frac{2^\alpha}{3^\alpha}(1 - {}^4\zeta/3 + \zeta^2/3)^{-\alpha}.$$

- $\omega_n = {}^{2^\alpha}/3^\alpha \tilde{\omega}_n$,
- $a_1 = -{}^4/3$, $a_2 = {}^1/3$, $a_j = 0$ if $j \geq 3$, thus using

$$\tilde{\omega}_0^{(\alpha)} = 1, \qquad \tilde{\omega}_n^{(\alpha)} = \sum_{j=1}^{n} \left(\frac{(\alpha+1)j}{n} - 1 \right) a_j v_{n-j}^{(\alpha)}$$

A worked out example

Now we need to expand the convolution coefficients of

$$\omega^\alpha(\zeta) = (\omega(\zeta))^\alpha = \frac{2^\alpha}{3^\alpha}(1 - 4\zeta/3 + \zeta^2/3)^{-\alpha}.$$

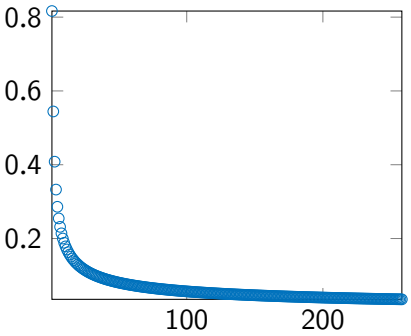
- $\omega_n = 2^\alpha/3^\alpha \tilde{\omega}_n$,
- $a_1 = -4/3$, $a_2 = 1/3$, $a_j = 0$ if $j \geq 3$, thus using

$$\tilde{\omega}_0^{(\alpha)} = 1, \qquad \tilde{\omega}_n^{(\alpha)} = \sum_{j=1}^{n} \left(\frac{(\alpha+1)j}{n} - 1 \right) a_j v_{n-j}^{(\alpha)}$$

- we get $\tilde{\omega}_0 = 1$, $\tilde{\omega}_1 = 4/3\alpha\tilde{\omega}_0 = 4\alpha/3$,

$$\tilde{\omega}_n = \frac{4}{3}\left(1 + \frac{\alpha - 1}{n}\right)\tilde{\omega}_{n-1} + \frac{4}{3}\left(\frac{2(1-\alpha)}{n} - 1\right)\tilde{\omega}_{n-2}.$$

A worked out example

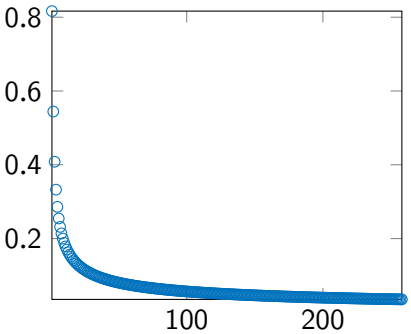


$\alpha = 0.5.$

- Since the a_j are a finite small number, we can compute the coefficients in an $O(N)$ operations,

```
omega = zeros(1,N+1) ;
onethird = 1/3 ; fourthird = 4/3;
twothird_oneminusalpha = 2/3*(1-alpha);
fourthird_oneminusalpha = 4/3*(1-alpha);
omega(1) = 1 ; omega(2) = fourthird*alpha*omega(1);
for n = 2 : N
 omega(n+1) = (fourthird -
 ↪  fourthird_oneminusalpha/n)*omega(n) + ...
 (twothird_oneminusalpha/n - onethird)*omega(n-1);
end
omega = omega*((2/3)^(alpha));
```

A worked out example

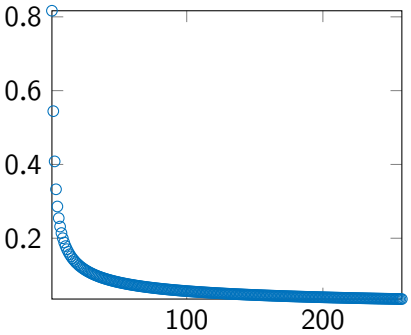


$\alpha = 0.5.$

- Since the a_j are a finite small number, we can compute the coefficients in an $O(N)$ operations,
- We can solve ${}_{CA}D^{0.5}_{[0,2]}y(t) = -2y(t)$, $y(0) = 1$

τ	$\lvert y^{(n)} - y(2)\rvert$	order
2^{-6}	1.44e-04	1.61
2^{-7}	4.42e-05	1.71
2^{-8}	1.28e-05	1.79
2^{-9}	3.57e-06	1.84
2^{-10}	9.68e-07	1.88
2^{-11}	2.85e-07	1.76
2^{-12}	8.17e-08	1.80
2^{-13}	2.29e-08	1.84
2^{-14}	6.27e-09	1.87

A worked out example



$\alpha = 0.5.$

- Since the a_j are a finite small number, we can compute the coefficients in an $O(N)$ operations,
- We can solve ${}_{CA}D^{0.5}_{[0,2]}y(t) = -2y(t)$, $y(0) = 1$
- For the starting weights we have to solve a 3×3 Vandermonde system:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & \sqrt{2} \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} w_{n,0} \\ w_{n,1} \\ w_{n,2} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

number of time-step times.

Reuse

Since we have a fixed time-grid we can reuse the same factorization for the Vandermonde system and compute all the weights in a single sweep.

Fractional Brusselator

The Brusselator is a model of the **autocatalytic chemical reaction**, it is described by

$$\begin{cases} \dot{x}_1 = a - (\mu + 1)x_1 + x_1^2 x_2, \\ \dot{x}_2 = \mu x_1 - x_1^2 x_2, \end{cases} \qquad a, \mu > 0.$$

- If $\mu > a^2 + 1$ then a single Brusselator has a *unique limit cycle*,

Fractional Brusselator

The Brusselator is a model of the **autocatalytic chemical reaction**, it is described by

$$\begin{cases} \dot{x}_1 = a - (\mu + 1)x_1 + x_1^2 x_2, \\ \dot{x}_2 = \mu x_1 - x_1^2 x_2, \end{cases} \qquad a, \mu > 0.$$

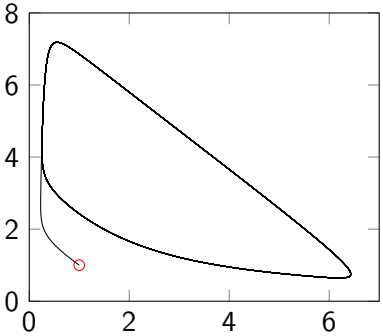
- If $\mu > a^2 + 1$ then a single Brusselator has a *unique limit cycle*,
- If $(a-1)^2 < \mu \leq a^2 + 1$ all the orbits tend to the steady state.

Fractional Brusselator

The Brusselator is a model of the **autocatalytic chemical reaction**, it is described by

$$\begin{cases} \dot{x}_1 = a - (\mu + 1)x_1 + x_1^2 x_2, \\ \dot{x}_2 = \mu x_1 - x_1^2 x_2, \end{cases} \qquad a, \mu > 0.$$

```
a = 1 ; mu = 4 ;
param = [ a , mu ] ;
f_fun = @(t,y,par) [ ...
par(1) - (par(2)+1)*y(1) + y(1)^2*y(2) ; ...
par(2)*y(1) - y(1)^2*y(2) ] ;
t0 = 0 ; T = 100 ;
y0 = [ 1 ; 1] ;
[T,Y] = ode45(@(t,y)
↪  f_fun(t,y,param),[t0,T],y0);
figure(1)
plot(Y(:,1),Y(:,2),'k-',y(1,1),y(2,1),'ro')
```

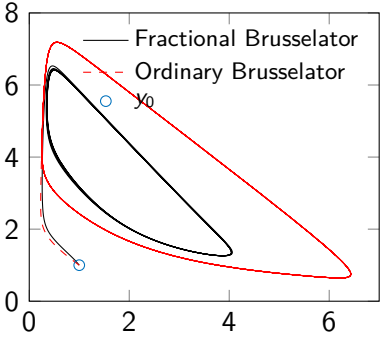


Fractional Brusselator

The Brusselator is a model of the **autocatalytic chemical reaction**, it is described by

$$\begin{cases} {}_{CA}D^{\alpha_1}x(t) = a - (\mu+1)x_1 + x_1^2 x_2, \\ {}_{CA}D^{\alpha_2}x(t) = \mu x_1 - x_1^2 x_2, \end{cases} \qquad a, \mu > 0.$$

```
alpha = [0.8,0.7] ;
h = 1e-2;
[t, y] =
↪  fde_pi1_ex(alpha,f_fun,t0,T,y0,h,param) ;
```



The cycle of the single fractional Brusselator is contained in the region

$$\left\{ (x_1, x_2) : \frac{a}{\mu+1} < x_1 < \frac{2a}{\mu},\ 0 < x_2 < \frac{\mu(1+\mu)}{a} \right\}$$

Fractional Brusselator

The Brusselator is a model of the **autocatalytic chemical reaction**, it is described by

$$\begin{cases} {}_{CA}D^{\alpha_1}x(t) = a - (\mu+1)x_1 + x_1^2 x_2, \\ {}_{CA}D^{\alpha_2}x(t) = \mu x_1 - x_1^2 x_2, \end{cases} \qquad a, \mu > 0.$$

```
alpha = [0.8,0.7] ;
h = 1e-2;
[t, y] =
↪  fde_pi1_ex(alpha,f_fun,t0,T,y0,h,param) ;
```

The cycle of the single fractional Brusselator is contained in the region

$$\left\{ (x_1, x_2) : \frac{a}{\mu+1} < x_1 < \frac{2a}{\mu},\ 0 < x_2 < \frac{\mu(1+\mu)}{a} \right\}$$

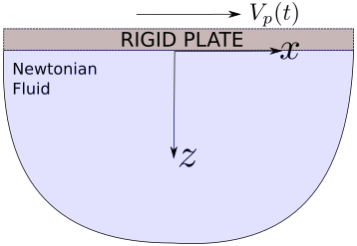
? Of interest (Wang and Li 2007)

Finding the smallest values α_1, α_2 for which a limit cycle exist is of interest.

Bagley-Torvik Model (Bagley and Torvik 1986)

Stoke's Second Problem

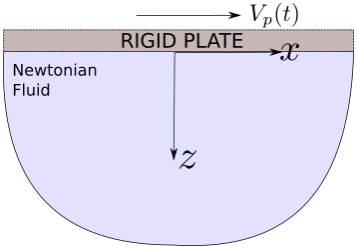
Can we determine the behavior of a half-space of Newtonian, viscous fluid undergoing the motion induced by the prescribed uniform sinusoidal motion of a plate on the surface?



Bagley-Torvik Model (Bagley and Torvik 1986)

Stoke's Second Problem

Can we determine the behavior of a half-space of Newtonian, viscous fluid undergoing the motion induced by the prescribed uniform sinusoidal motion of a plate on the surface?



If we write down the **equation of motion** we find

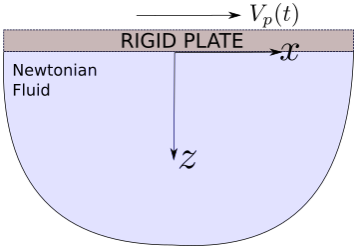
$$\rho \frac{\partial v}{\partial t} = \mu \frac{\partial^2 v}{\partial z^2}$$

- ρ is the *fluid density*, μ is the viscosity, v is the profile of the *transverse fluid velocity*.

Bagley-Torvik Model (Bagley and Torvik 1986)

Stoke's Second Problem

Can we determine the behavior of a half-space of Newtonian, viscous fluid undergoing the motion induced by the prescribed uniform sinusoidal motion of a plate on the surface?



If we write down the **equation of motion** we find

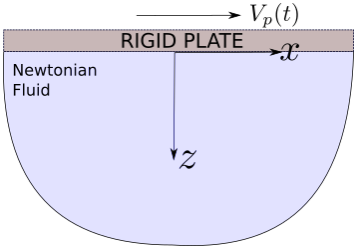
$$\rho[s\tilde{v}(s,z) - v(0,x)] = \mu \frac{d^2\tilde{v}(s,z)}{dz^2},$$

- ρ is the *fluid density*, μ is the viscosity, v is the profile of the *transverse fluid velocity*.
- We apply Laplace transform to the equation $\tilde{v} = \mathcal{L}v(s)$,

Bagley-Torvik Model (Bagley and Torvik 1986)

Stoke's Second Problem

Can we determine the behavior of a half-space of Newtonian, viscous fluid undergoing the motion induced by the prescribed uniform sinusoidal motion of a plate on the surface?



If we write down the **equation of motion** we find

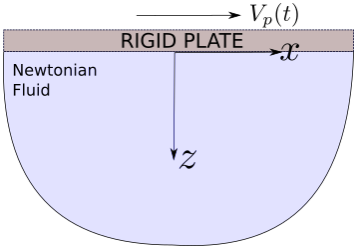
$$\tilde{v}(s,z) = \tilde{v}_p(s)\exp\left(\sqrt{\frac{\rho s}{\mu}}z\right)$$

- ρ is the *fluid density*, μ is the viscosity, v is the profile of the *transverse fluid velocity*.
- We apply Laplace transform to the equation $\tilde{v} = \mathcal{L}v(s)$,
- We solve and impose the boundary condition given by the $\tilde{v}_p = \mathcal{L}V_p(s)$,

Bagley-Torvik Model (Bagley and Torvik 1986)

Stoke's Second Problem

Can we determine the behavior of a half-space of Newtonian, viscous fluid undergoing the motion induced by the prescribed uniform sinusoidal motion of a plate on the surface?



If we write down the **equation of motion** we find

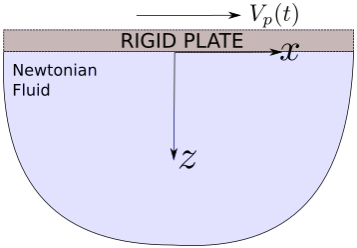
$$\tilde{\sigma}(s,z) = \sqrt{\mu\rho}\sqrt{s}\tilde{v}(s,z) = \sqrt{\mu\rho}\frac{1}{\sqrt{s}}s\tilde{v}(s,z)$$

- ρ is the *fluid density*, μ is the viscosity, v is the profile of the *transverse fluid velocity*.
- We apply Laplace transform to the equation $\tilde{v} = \mathcal{L}v(s)$,
- We solve and impose the boundary condition given by the $\tilde{v}_p = \mathcal{L}V_p(s)$,
- Since the *shear stress* is given by $\sigma(t,z) = \mu v_z(t,z)$ we can write its Laplace transform.

Bagley-Torvik Model (Bagley and Torvik 1986)

Stoke's Second Problem

Can we determine the behavior of a half-space of Newtonian, viscous fluid undergoing the motion induced by the prescribed uniform sinusoidal motion of a plate on the surface?



If we write down the **equation of motion** we find

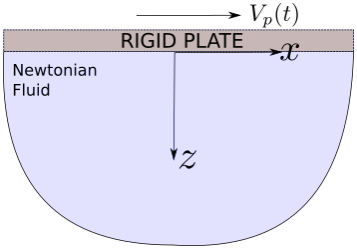
$$\tilde{\sigma}(s,z) = \sqrt{\mu\rho}\mathcal{L}\left\{\frac{1}{\Gamma(1/2)t^{1/2}}\right\} * \mathcal{L}\{v_t\}$$

- ρ is the *fluid density*, μ is the viscosity, v is the profile of the *transverse fluid velocity*.
- We apply Laplace transform to the equation $\tilde{v} = \mathcal{L}v(s)$,
- We solve and impose the boundary condition given by the $\tilde{v}_p = \mathcal{L}V_p(s)$,
- Since the *shear stress* is given by $\sigma(t,z) = \mu v_z(t,z)$ we can write its Laplace transform.
- Finally we invert it.

Bagley-Torvik Model (Bagley and Torvik 1986)

Stoke's Second Problem

Can we determine the behavior of a half-space of Newtonian, viscous fluid undergoing the motion induced by the prescribed uniform sinusoidal motion of a plate on the surface?



If we write down the **equation of motion** we find

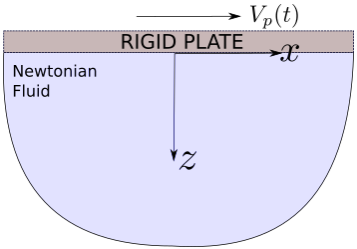
$$\sigma(t,z) = \sqrt{\mu\rho}\frac{1}{\Gamma(1/2)}\int_0^t (t-\tau)^{1/2} v_\tau(\tau,z)\,\mathrm{d}\tau.$$

- ρ is the *fluid density*, μ is the viscosity, v is the profile of the *transverse fluid velocity*.
- We apply Laplace transform to the equation $\tilde{v} = \mathcal{L}v(s)$,
- We solve and impose the boundary condition given by the $\tilde{v}_p = \mathcal{L}V_p(s)$,
- Since the *shear stress* is given by $\sigma(t,z) = \mu v_z(t,z)$ we can write its Laplace transform.
- Finally we invert it.

Bagley-Torvik Model (Bagley and Torvik 1986)

Stoke's Second Problem

Can we determine the behavior of a half-space of Newtonian, viscous fluid undergoing the motion induced by the prescribed uniform sinusoidal motion of a plate on the surface?

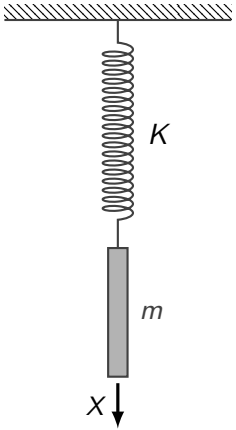


If we write down the **equation of motion** we find

$$\sigma(t,z) = \sqrt{\mu\rho}\, CA\, D^{1/2}_{[0,t]} v(t,z).$$

- ρ is the *fluid density*, μ is the viscosity, v is the profile of the *transverse fluid velocity*.
- We apply Laplace transform to the equation $\tilde{v} = \mathcal{L}v(s)$,
- We solve and impose the boundary condition given by the $\tilde{v}_p = \mathcal{L}V_p(s)$,
- Since the *shear stress* is given by $\sigma(t,z) = \mu v_z(t,z)$ we can write its Laplace transform.
- Finally we invert it.

Bagley-Torvik Model (Bagley and Torvik 1986)

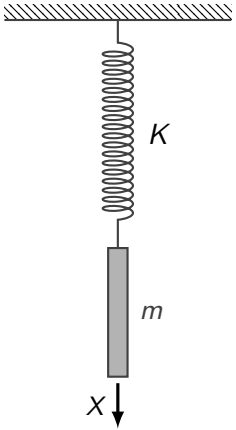


Immersed Plate

Assumptions:

- The spring is massless and its oscillations do not disturb the fluid,
- The area A of the plate is sufficiently large as to produce in the fluid adjacent to the plate the velocity field and stresses we just derived,

Bagley-Torvik Model (Bagley and Torvik 1986)



Immersed Plate

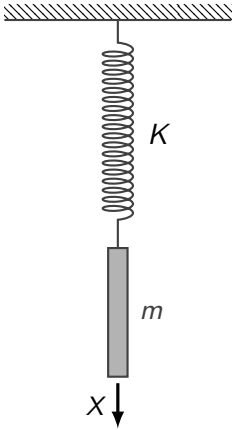
Assumptions:

- The spring is massless and its oscillations do not disturb the fluid,
- The area A of the plate is sufficiently large as to produce in the fluid adjacent to the plate the velocity field and stresses we just derived,

Deriving the equation:

$$m\ddot{X} = F_X = -KX - 2A\sigma(t, 0)$$

Bagley-Torvik Model (Bagley and Torvik 1986)



Immersed Plate

Assumptions:

- The spring is massless and its oscillations do not disturb the fluid,
- The area A of the plate is sufficiently large as to produce in the fluid adjacent to the plate the velocity field and stresses we just derived,

Deriving the equation:

$$m\ddot{X} = F_X = -KX - 2A\sigma(t,0)$$

Using the expression for the *strain* and $V_p(t,0) = \dot{X}(t)$ we find

$$m\ddot{X} + 2A\sqrt{\mu\rho}_{CA} D^{3/2}_{[0,t]} X + KX = 0.$$

Linear Multi-Term FDEs

The Bagley-Torvik model is an example of a Linear Multi-Term FDE, that is, something of the form

$$\lambda_{Q\,CA}D^{\alpha_Q}y(t) + \lambda_{Q-1\,CA}D^{\alpha_{Q-1}}y(t) + \cdots + \lambda_{2\,CA}D^{\alpha_2}y(t) + \lambda_{1\,CA}D^{\alpha_1}y(t) = f(t, y(t)),$$

with

- $\lambda_i \in \mathbb{R}\ \forall\, i = 1, \ldots, Q$,
- $0 < \alpha_1 < \alpha_2 < \ldots < \alpha_{Q-1} < \alpha_Q$ and $\alpha_Q \neq 0$.

For this problem we have $m_Q = \max m_i$, $m_i = \lceil \alpha_i \rceil$, $i = 1, \ldots, Q$ **initial conditions**:

$$y(t_0) = y_0, y'(t_0) = y_0^{(1)}, \ldots, y^{(m_Q-1)}(t_0) = y_0^{(m_Q-1)}.$$

Linear Multi-Term FDEs

The Bagley-Torvik model is an example of a Linear Multi-Term FDE, that is, something of the form

$$\lambda_{Q\,CA}D^{\alpha_Q}y(t) + \lambda_{Q-1\,CA}D^{\alpha_{Q-1}}y(t) + \cdots + \lambda_{2\,CA}D^{\alpha_2}y(t) + \lambda_{1\,CA}D^{\alpha_1}y(t) = f(t, y(t)),$$

with

- $\lambda_i \in \mathbb{R}\ \forall\, i = 1, \ldots, Q$,
- $0 < \alpha_1 < \alpha_2 < \ldots < \alpha_{Q-1} < \alpha_Q$ and $\alpha_Q \neq 0$.

For this problem we have $m_Q = \max m_i$, $m_i = \lceil \alpha_i \rceil$, $i = 1, \ldots, Q$ **initial conditions**:

$$y(t_0) = y_0, y'(t_0) = y_0^{(1)}, \ldots, y^{(m_Q-1)}(t_0) = y_0^{(m_Q-1)}.$$

❓ How can we solve them?

Linear Multi-Term FDEs

We need to **recall one of the properties** we have seen of the Caputo derivatives

(P_1) $I^{\alpha}_{[t_0,T]}\,{}_{CA}D^{\alpha}_{[t_0,T]}y(t) = y(t) - T_{m-1}[y, t_0](t),$

(P_2) $I^{\beta}_{[t_0,T]}\,{}_{CA}D^{\alpha}_{[t_0,T]}y(t) = I^{\beta}_{[t_0,T]}\,{}_{RL}D^{\alpha}_{[t_0,T]}\,[y(t) - T_{m-1}[y; t_0](t)] =$
$I^{\beta-\alpha}_{[t_0,T]}\,[y(t) - T_{m-1}[y; t_0](t)],\ \beta > \alpha.$

Linear Multi-Term FDEs

We need to **recall one of the properties** we have seen of the Caputo derivatives

(P_1) $I^{\alpha}_{[t_0,T]}\,{}_{CA}D^{\alpha}_{[t_0,T]}y(t) = y(t) - T_{m-1}[y,t_0](t),$

(P_2) $I^{\beta}_{[t_0,T]}\,{}_{CA}D^{\alpha}_{[t_0,T]}y(t) = I^{\beta}_{[t_0,T]}\,{}_{RL}D^{\alpha}_{[t_0,T]}\,[y(t) - T_{m-1}[y;t_0](t)] =$
$I^{\beta-\alpha}_{[t_0,T]}\,[y(t) - T_{m-1}[y;t_0](t)],\ \beta > \alpha.$

We start from the multi-term equation

$$\lambda_{Q\,CA}D^{\alpha_Q}y(t) + \lambda_{Q-1\,CA}D^{\alpha_{Q-1}}y(t) + \cdots + \lambda_{2\,CA}D^{\alpha_2}y(t) + \lambda_{1\,CA}D^{\alpha_1}y(t) = f(t,y(t)),$$

Linear Multi-Term FDEs

We need to **recall one of the properties** we have seen of the Caputo derivatives

(P_1) $I^{\alpha}_{[t_0,T]}\,{}_{CA}D^{\alpha}_{[t_0,T]}y(t) = y(t) - T_{m-1}[y,t_0](t),$

(P_2) $I^{\beta}_{[t_0,T]}\,{}_{CA}D^{\alpha}_{[t_0,T]}y(t) = I^{\beta}_{[t_0,T]}\,{}_{RL}D^{\alpha}_{[t_0,T]}\left[y(t) - T_{m-1}[y;t_0](t)\right] =$
$I^{\beta-\alpha}_{[t_0,T]}\left[y(t) - T_{m-1}[y;t_0](t)\right],\ \beta > \alpha.$

We start from the multi-term equation

$$\lambda_Q I^{\alpha_Q}_{[t_0,T]}\left[{}_{CA}D^{\alpha_Q}y(t)\right] = -I^{\alpha_Q}_{[t_0,T]}\left[\sum_{i=1}^{Q-1}\lambda_i\,{}_{CA}D^{\alpha_i}y(t) + f(t,y(t))\right],$$

- we multiply both sides by $I^{\alpha_Q}_{[t_0,T]}$,

Linear Multi-Term FDEs

We need to **recall one of the properties** we have seen of the Caputo derivatives

(P_1) $I^{\alpha}_{[t_0,T]}\,{}_{CA}D^{\alpha}_{[t_0,T]}y(t) = y(t) - T_{m-1}[y,t_0](t),$

(P_2) $I^{\beta}_{[t_0,T]}\,{}_{CA}D^{\alpha}_{[t_0,T]}y(t) = I^{\beta}_{[t_0,T]}\,{}_{RL}D^{\alpha}_{[t_0,T]}\left[y(t) - T_{m-1}[y;t_0](t)\right] =$
$I^{\beta-\alpha}_{[t_0,T]}\left[y(t) - T_{m-1}[y;t_0](t)\right],\ \beta > \alpha.$

We start from the multi-term equation

$$y(t) - T_{m_Q-1}[y,t_0](t) = -\sum_{i=1}^{Q-1}\frac{\lambda_i}{\lambda_Q}I^{\alpha_Q-\alpha_i}_{[t_0,t]}\left[y(t) - T_{m_i-1}[y;t_0](t)\right] + \frac{1}{\lambda_Q}I^{\alpha_Q}_{[t_0,T]}f(t,y(t))$$

- we multiply both sides by $I^{\alpha_Q}_{[t_0,T]}$,
- we use P_1 on the left-hand side, P_2 on the right-hand side,

Linear Multi-Term FDEs

We need to **recall one of the properties** we have seen of the Caputo derivatives

(P_1) $I^{\alpha}_{[t_0,T]}\,{}_{CA}D^{\alpha}_{[t_0,T]}y(t) = y(t) - T_{m-1}[y,t_0](t),$

(P_2) $I^{\beta}_{[t_0,T]}\,{}_{CA}D^{\alpha}_{[t_0,T]}y(t) = I^{\beta}_{[t_0,T]}\,{}_{RL}D^{\alpha}_{[t_0,T]}\left[y(t) - T_{m-1}[y;t_0](t)\right] =$
$I^{\beta-\alpha}_{[t_0,T]}\left[y(t) - T_{m-1}[y;t_0](t)\right],\ \beta > \alpha.$

We start from the multi-term equation

$$y(t) = T_{m_Q-1}[y,t_0](t) - \sum_{i=1}^{Q-1}\frac{\lambda_i}{\lambda_Q} I^{\alpha_Q-\alpha_i}_{[t_0,t]}\left[y(t) - T_{m_i-1}[y;t_0](t)\right] + \frac{1}{\lambda_Q} I^{\alpha_Q}_{[t_0,T]} f(t,y(t))$$

- we multiply both sides by $I^{\alpha_Q}_{[t_0,T]}$,
- we use P_1 on the left-hand side, P_2 on the right-hand side,
- and re-arrange to get an expression for the solution.

Linear Multi-Term FDEs: generalizing PI rules

First we do a bit of rewriting of

$$y(t) = T_{m_Q-1}[y, t_0](t) - \sum_{i=1}^{Q-1} \frac{\lambda_i}{\lambda_Q} I_{[t_0,t]}^{\alpha_Q-\alpha_i} [y(t) - T_{m_i-1}[y; t_0](t)] + \frac{1}{\lambda_Q} I_{[t_0,T]}^{\alpha_Q} f(t, y(t))$$

- we employ the usual fractional integral for polynomials:

$$I_{[t_0,t]}^{\alpha} T_{m-1}[y; t_0](t) = \sum_{k=0}^{m-1} \frac{(t-t_0)^{k+\alpha}}{\Gamma(k+\alpha)} y^{(k)}(t_0), \quad \begin{array}{l} \alpha \in \{\alpha_1, \ldots, \alpha_{Q-1}\}, \\ m \in \{m_1, \ldots, m_{Q-1}\}. \end{array}$$

Linear Multi-Term FDEs: generalizing PI rules

First we do a bit of rewriting of

$$y(t) = T_{m_Q-1}[y, t_0](t) - \sum_{i=1}^{Q-1} \frac{\lambda_i}{\lambda_Q} I_{[t_0,t]}^{\alpha_Q-\alpha_i}\left[y(t) - T_{m_i-1}[y; t_0](t)\right] + \frac{1}{\lambda_Q} I_{[t_0,T]}^{\alpha_Q} f(t, y(t))$$

- we employ the usual fractional integral for polynomials:

$$I_{[t_0,t]}^{\alpha} T_{m-1}[y; t_0](t) = \sum_{k=0}^{m-1} \frac{(t-t_0)^{k+\alpha}}{\Gamma(k+\alpha)} y^{(k)}(t_0), \quad \begin{array}{l} \alpha \in \{\alpha_1, \ldots, \alpha_{Q-1}\}, \\ m \in \{m_1, \ldots, m_{Q-1}\}. \end{array}$$

- We use it to simplify the expression

$$\tilde{T}(t) = T_{m_Q-1}[y; t_0](t) + \sum_{i=1}^{Q-1} \frac{\lambda_i}{\lambda_Q} \sum_{k=0}^{m_i-1} \frac{(t-t_0)^{k+\alpha_Q-\alpha_i}}{\Gamma(k+\alpha_Q-\alpha_i+1)} y^{(k)}(t_0).$$

Linear Multi-Term FDEs: generalizing PI rules

First we do a bit of rewriting of

$$y(t) = \tilde{T}(t) - \sum_{i=1}^{Q-1} \frac{\lambda_i}{\lambda_Q} I_{[t_0,t]}^{\alpha_Q-\alpha_i} y(t) + \frac{1}{\lambda_Q} I_{[t_0,T]}^{\alpha_Q} f(t, y(t)).$$

- we employ the usual fractional integral for polynomials:

$$I_{[t_0,t]}^{\alpha} T_{m-1}[y; t_0](t) = \sum_{k=0}^{m-1} \frac{(t-t_0)^{k+\alpha}}{\Gamma(k+\alpha)} y^{(k)}(t_0), \quad \begin{array}{l} \alpha \in \{\alpha_1, \ldots, \alpha_{Q-1}\}, \\ m \in \{m_1, \ldots, m_{Q-1}\}. \end{array}$$

- We use it to simplify the expression

$$\tilde{T}(t) = T_{m_Q-1}[y; t_0](t) + \sum_{i=1}^{Q-1} \frac{\lambda_i}{\lambda_Q} \sum_{k=0}^{m_i-1} \frac{(t-t_0)^{k+\alpha_Q-\alpha_i}}{\Gamma(k+\alpha_Q-\alpha_i+1)} y^{(k)}(t_0).$$

Linear Multi-Term FDEs: generalizing PI rules

Now we have an expression that we can treat by adapting one of the Product Integral rules

$$y(t) = \tilde{T}(t) - \sum_{i=1}^{Q-1} \frac{\lambda_i}{\lambda_Q} I_{[t_0,t]}^{\alpha_Q-\alpha_i} y(t) + \frac{1}{\lambda_Q} I_{[t_0,T]}^{\alpha_Q} f(t, y(t)).$$

Linear Multi-Term FDEs: generalizing PI rules

Now we have an expression that we can treat by adapting one of the Product Integral rules

$$y(t) = \tilde{T}(t) - \sum_{i=1}^{Q-1} \frac{\lambda_i}{\lambda_Q} I_{[t_0,t]}^{\alpha_Q-\alpha_i} y(t) + \frac{1}{\lambda_Q} I_{[t_0,T]}^{\alpha_Q} f(t, y(t)).$$

We can start from the **explicit rectangular product integral rule** on a uniform grid

$$y^{(n)} = \tilde{T}(t_n) - \sum_{i=1}^{Q-1} \frac{\lambda_i}{\lambda_Q} \tau^{\alpha_Q-\alpha_i} \sum_{j=0}^{n-1} b_{n-j-1}^{(\alpha_Q-\alpha_i)} y^{(j)} + \frac{1}{\lambda_Q} \sum_{j=0}^{n-1} b_{n-j-1}^{(\alpha_Q)} f(t_j, y^{(j)}).$$

with

$$b_n^{(\alpha)} = [(n+1)^\alpha - n^\alpha]/\alpha, \quad n = 1, \ldots, N.$$

Linear Multi-Term FDEs: generalizing PI rules

Now we have an expression that we can treat by adapting one of the Product Integral rules

$$y(t) = \tilde{T}(t) - \sum_{i=1}^{Q-1} \frac{\lambda_i}{\lambda_Q} I_{[t_0,t]}^{\alpha_Q-\alpha_i} y(t) + \frac{1}{\lambda_Q} I_{[t_0,T]}^{\alpha_Q} f(t, y(t)).$$

We can start from the **implicit rectangular product integral rule** on a uniform grid

$$y^{(n)} = \tilde{T}(t_n) - \sum_{i=1}^{Q-1} \frac{\lambda_i}{\lambda_Q} \tau^{\alpha_Q-\alpha_i} \sum_{j=0}^{n} b_{n-j-1}^{(\alpha_Q-\alpha_i)} y^{(j)} + \frac{1}{\lambda_Q} \sum_{j=0}^{n} b_{n-j-1}^{(\alpha_Q)} f(t_j, y^{(j)}).$$

with

$$b_n^{(\alpha)} = [(n+1)^\alpha - n^\alpha]/\alpha, \quad n = 1, \ldots, N.$$

Linear Multi-Term FDEs: generalizing PI rules

Now we have an expression that we can treat by adapting one of the Product Integral rules

$$y(t) = \tilde{T}(t) - \sum_{i=1}^{Q-1} \frac{\lambda_i}{\lambda_Q} I_{[t_0,t]}^{\alpha_Q-\alpha_i} y(t) + \frac{1}{\lambda_Q} I_{[t_0,T]}^{\alpha_Q} f(t, y(t)).$$

We can start from the **implicit rectangular product integral rule** on a uniform grid

$$y^{(n)} = \tilde{T}(t_n) - \sum_{i=1}^{Q-1} \frac{\lambda_i}{\lambda_Q} \tau^{\alpha_Q-\alpha_i} \sum_{j=0}^{n} b_{n-j-1}^{(\alpha_Q-\alpha_i)} y^{(j)} + \frac{1}{\lambda_Q} \sum_{j=0}^{n} b_{n-j-1}^{(\alpha_Q)} f(t_j, y^{(j)}).$$

with

$$b_n^{(\alpha)} = [(n+1)^\alpha - n^\alpha]/\alpha, \quad n = 1, \ldots, N.$$

We can do it similarly for the **Implicit Trapezoidal Rule** and then for the **Predictor-Corrector method** (Diethelm 2003).

Linear Multi-Term FDEs: generalizing PI rules

❓ Can we do something similar for FLMMs?

Linear Multi-Term FDEs: generalizing PI rules

❓ Can we do something similar for FLMMs?

⚙ We don't know how to determine the starting values $w_{n,j}$ for the quadrature. Thus this approach is not viable.

Linear Multi-Term FDEs: generalizing PI rules

❓ Can we do something similar for FLMMs?

⚙ We don't know how to determine the starting values $w_{n,j}$ for the quadrature. Thus this approach is not viable.

Available codes (Garrappa 2018):

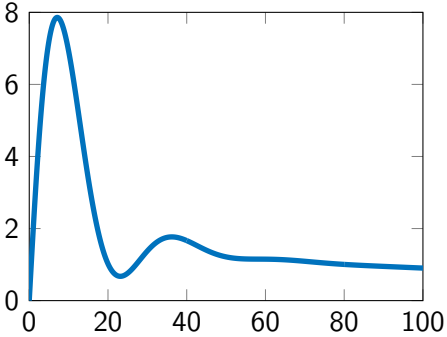
- `MT_FDE_PI1_Ex.m` - Explicit Product-Integration of rectanguar type
- `MT_FDE_PI1_Im.m` - Implicit Product-Integration of rectanguar type
- `MT_FDE_PI2_Im.m` - Implicit Product-Integration of trapezoidal type
- `MT_FDE_PI12_PC.m`- Product-Integration with predictor-corrector

Linear Multi-Term FDEs: back to Bagley-Torvik

We reached the equation

$$m\ddot{X} + 2A\sqrt{\mu\rho}\,{}_{CA}D^{3/2}_{[0,t]}X + KX = 0.$$

```
m = 10; A = 6; K = 3;
mu = 2; rho = 2;
alpha = [2 3/2] ;
lambda = [m 2*A*sqrt(mu*rho)] ;
f_fun = @(t,X) -K*X;
J_fun = @(t,X) -K;
t0 = 0 ; T = 100 ;
X0 = [0 , 2 ];
h = 1e-2;
[t, X] = mt_fde_pi1_ex(alpha, lambda, f_fun,
↪  t0, T, X0, h);
```

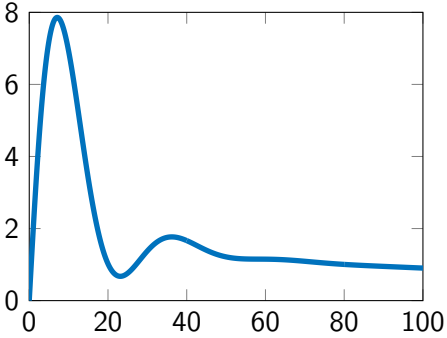


Linear Multi-Term FDEs: back to Bagley-Torvik

We reached the equation

$$m\ddot{X} + 2A\sqrt{\mu\rho}\,{}_{CA}D^{3/2}_{[0,t]}X + KX = 0.$$

```
m = 10; A = 6; K = 3;
mu = 2; rho = 2;
alpha = [2 3/2] ;
lambda = [m 2*A*sqrt(mu*rho)] ;
f_fun = @(t,X) -K*X;
J_fun = @(t,X) -K;
t0 = 0 ; T = 100 ;
X0 = [0 , 2 ];
h = 1e-2;
[t, X] = mt_fde_pi1_ex(alpha, lambda, f_fun,
↪  t0, T, X0, h);
```



But does it fit the reality?

Linear Multi-Term FDEs: back to Bagley-Torvik

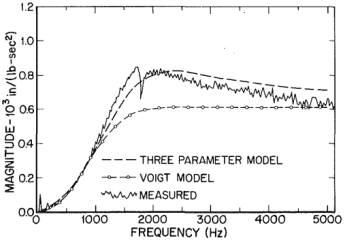


Fig. 5 The magnitude of the transfer function for Case 1

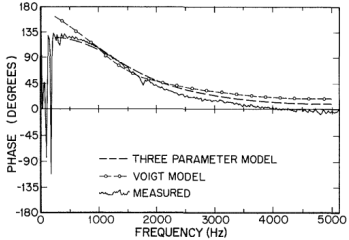


Fig. 6 The phase of the transfer function for Case 1

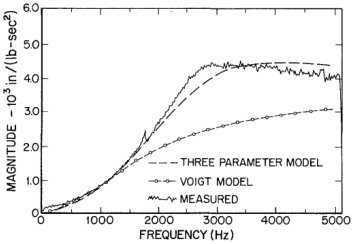


Fig. 7 The magnitude of the transfer function for Case 5

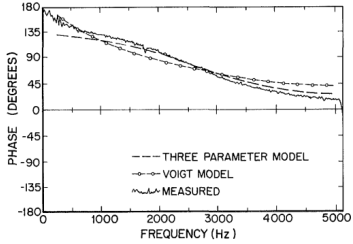


Fig. 8 The phase of the transfer function for Case 5

The model we have derived is a model of the form

$$\sigma(t) = G_0\epsilon(t) + G_1\dot{\epsilon}(t),$$
$$\epsilon(t) = \frac{x(t)}{\delta}$$
$$f(t) = m\ddot{x}(t) + f_p(t),$$
$$f_p(t) = \frac{2A}{\delta}(G_0 + G_1\,{}_{CA}D^{\alpha}x(t)).$$

One can do *parameter tuning* to find the fractional order from experimental data and compare the results with the integer-order model. The results on the left by Bagley and Torvik 1986 show that the fractional model obtain a better fit with the measured data.

Equivalent formulations of the Multi-Term FDEs

In the integer-order case we know how to rewrite the equation

$$y^{(n)}(t) = f(t, y^{(n-1)}(t), \ldots, y^{(1)}(t), y(t)), \quad y^{(j)}(0) = y_0^{(j)}, \quad j = 0, 1, \ldots, n-1,$$

as a **system of first-order equations**.

Equivalent formulations of the Multi-Term FDEs

In the integer-order case we know how to rewrite the equation

$$y^{(n)}(t) = f(t, y^{(n-1)}(t), \ldots, y^{(1)}(t), y(t)), \quad y^{(j)}(0) = y_0^{(j)}, \quad j = 0, 1, \ldots, n-1,$$

as a **system of first-order equations**.

❓ Can we do something similar in the fractional case?

Equivalent formulations of the Multi-Term FDEs

In the integer-order case we know how to rewrite the equation

$$y^{(n)}(t) = f(t, y^{(n-1)}(t), \ldots, y^{(1)}(t), y(t)), \quad y^{(j)}(0) = y_0^{(j)}, \quad j = 0, 1, \ldots, n-1,$$

as a **system of first-order equations**.

(A1) Let us assume that our multi-term equation is of the form

$$ {}_{CA}D^{\alpha_k} y(t) = f(t, {}_{CA}D^{\alpha_{k-1}} y(t), \ldots, {}_{CA}D^{\alpha_1}(t), y(t)), \qquad \begin{array}{l} y^{(j)}(0) = y_0^{(j)}, \\ j = 0, 1, \ldots, n-1, \end{array}$$

for $\alpha_k > \alpha_{k-1} > \cdots > \alpha_1 > 0$, $\alpha_j - \alpha_{j-1} \leq 1 \; \forall j = 1, 2, \ldots, k$, $0 < \alpha_1 \leq 1$.

Equivalent formulations of the Multi-Term FDEs

In the integer-order case we know how to rewrite the equation

$$y^{(n)}(t) = f(t, y^{(n-1)}(t), \ldots, y^{(1)}(t), y(t)), \quad y^{(j)}(0) = y_0^{(j)}, \quad j = 0, 1, \ldots, n-1,$$

as a **system of first-order equations**.

(A1) Let us assume that our multi-term equation is of the form

$$ {}_{CA}D^{\alpha_k} y(t) = f(t, {}_{CA}D^{\alpha_{k-1}} y(t), \ldots, {}_{CA}D^{\alpha_1}(t), y(t)), \qquad \begin{array}{l} y^{(j)}(0) = y_0^{(j)}, \\ j = 0, 1, \ldots, n-1, \end{array}$$

for $\alpha_k > \alpha_{k-1} > \cdots > \alpha_1 > 0$, $\alpha_j - \alpha_{j-1} \leq 1 \; \forall j = 1, 2, \ldots, k$, $0 < \alpha_1 \leq 1$.

(A2) Assume also that $\alpha_j \in \mathbb{Q} \; \forall j = 1, 2, \ldots, k$, and that M is the least common multiple of $\alpha_1, \alpha_2, \ldots, \alpha_k$.

Equivalent formulations of the Multi-Term FDEs

Theorem (Diethelm 2010, Theorem 8.1)

Under the assumptions (A1) and (A2), set $\gamma = 1/M$, and $N = M\alpha_k$, then the IVP is equivalent to

$$\begin{cases} {}_{CA}D^{\gamma} y_0(t) = y_1(t), \\ {}_{CA}D^{\gamma} y_1(t) = y_2(t), \\ \vdots \\ {}_{CA}D^{\gamma} y_{N-2}(t) = y_{N-1}(t), \\ {}_{CA}D^{\gamma} y_{N-1}(t) = f(t, y_0(t), y_{\alpha_{k-1}/M}(t), \ldots, y_{\alpha_1/M}(t), y(t)) \end{cases} \qquad y_i(0) = \begin{cases} y_0^{(j/m)}, & \text{if } \frac{j}{M} \in \mathbb{N}_0, \\ 0, & \text{otherwise.} \end{cases}$$

$\Rightarrow$ whenever $\mathbf{y} = (y_0, \ldots, y_{N-1})^T$ with $y_0 \in \mathcal{C}^{\lceil \alpha_k \rceil}[0, b]$, for some $b > 0$, is a solution of the N-dimensional system, then $y \equiv y_0$ is a solution of the multi-term FDE.

Equivalent formulations of the Multi-Term FDEs

Theorem (Diethelm 2010, Theorem 8.1)

Under the assumptions (A1) and (A2), set $\gamma = 1/M$, and $N = M\alpha_k$, then the IVP is equivalent to

$$\begin{cases} {}_{CA}D^{\gamma} y_0(t) = y_1(t), \\ {}_{CA}D^{\gamma} y_1(t) = y_2(t), \\ \vdots \\ {}_{CA}D^{\gamma} y_{N-2}(t) = y_{N-1}(t), \\ {}_{CA}D^{\gamma} y_{N-1}(t) = f(t, y_0(t), y_{\alpha_{k-1}/M}(t), \ldots, y_{\alpha_1/M}(t), y(t)) \end{cases} \qquad y_i(0) = \begin{cases} y_0^{(j/m)}, & \text{if } \frac{j}{M} \in \mathbb{N}_0, \\ 0, & \text{otherwise.} \end{cases}$$

$\Leftarrow$ whenever $y \in \mathcal{C}^{\lceil \alpha_k \rceil}([0, b])$ is a solution of the multi-term FDE, then the vector function $\mathbf{y} = (y, {}_{CA}D^{\gamma} y, {}_{CA}D^{2\gamma} y, \ldots, {}_{CA}D^{(N-1)\gamma} y)^T$ solves the N-dimensional system.

Equivalent formulations of the Multi-Term FDEs

We can relax (A2) from the *rationality requirement* to a requirement on being commensurable[2].

(A2)' Let $1 \geq \alpha_k > \alpha_{k-1} > \ldots > \alpha_1 > 0$ and assume the equation to be *commensurate*, then we define $\tilde{\alpha}_j = \alpha_j/\alpha_1$ for $j = 1, \ldots, k$, let $\tilde{M}$ be the least common multiple of the denominators of the values $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_k$.

Theorem (Diethelm 2010, Theorem 8.2)

Under the assumption (A1) and (A2)', set $\gamma = \alpha_1/\tilde{M}$ and $N = \tilde{M}\alpha_k/\alpha_1$, then the equivalence relation of the N-dimensional system and of the multi-term FDE holds as in the previous result.

[2] Two non-zero real numbers α and β are said to be commensurable if their ratio $\alpha/\beta \in \mathbb{Q}$.

Equivalent formulations of the Multi-Term FDEs

We can relax (A2) from the *rationality requirement* to a requirement on being commensurable[2].

(A2)' Let $1 \geq \alpha_k > \alpha_{k-1} > \ldots > \alpha_1 > 0$ and assume the equation to be *commensurate*, then we define $\tilde{\alpha}_j = \alpha_j/\alpha_1$ for $j = 1, \ldots, k$, let $\tilde{M}$ be the least common multiple of the denominators of the values $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_k$.

Theorem (Diethelm 2010, Theorem 8.2)

Under the assumption (A1) and (A2)', set $\gamma = \alpha_1/\tilde{M}$ and $N = \tilde{M}\alpha_k/\alpha_1$, then the equivalence relation of the N-dimensional system and of the multi-term FDE holds as in the previous result.

- Existence and uniqueness results can be obtained for the single term reformulation,

[2] Two non-zero real numbers α and β are said to be commensurable if their ratio $\alpha/\beta \in \mathbb{Q}$.

Equivalent formulations of the Multi-Term FDEs

We can relax (A2) from the *rationality requirement* to a requirement on being commensurable[2].

(A2)' Let $1 \geq \alpha_k > \alpha_{k-1} > \ldots > \alpha_1 > 0$ and assume the equation to be *commensurate*, then we define $\tilde{\alpha}_j = \alpha_j/\alpha_1$ for $j = 1, \ldots, k$, let $\tilde{M}$ be the least common multiple of the denominators of the values $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_k$.

Theorem (Diethelm 2010, Theorem 8.2)

Under the assumption (A1) and (A2)', set $\gamma = \alpha_1/\tilde{M}$ and $N = \tilde{M}\alpha_k/\alpha_1$, then the equivalence relation of the N-dimensional system and of the multi-term FDE holds as in the previous result.

- Existence and uniqueness results can be obtained for the single term reformulation,
- See (Ford and Connolly 2009) for other reformulations and comparisons.

[2] Two non-zero real numbers α and β are said to be commensurable if their ratio $\alpha/\beta \in \mathbb{Q}$.

The Method of Lines

Consider a partial differential equations of the form

$$\text{Find } u(\mathbf{x}, t) \text{ s.t. } u_t = \mathcal{L}u, \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^d, t \in I \subseteq \mathbb{R}_+,$$

where $\mathcal{L}$ is a differential operator, either linear or nonlinear, coupled with the opportune boundary conditions, and given suitable initial conditions.

The Method of Lines

Consider a partial differential equations of the form

$$\text{Find } u(\mathbf{x}, t) \text{ s.t. } u_t = \mathcal{L}u, \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^d, t \in I \subseteq \mathbb{R}_+,$$

where $\mathcal{L}$ is a differential operator, either linear or nonlinear, coupled with the opportune boundary conditions, and given suitable initial conditions.
A *classical way* of approaching this task is using a **Method Of Lines** (MOL) approach, that is

1. we discretize w.r.t. the *space variables* with some method (e.g., Finite Elements/Differences/Volumes, meshfree/meshless methods, spectral methods...)

$$M\mathbf{u}_t = F(t, \mathbf{u}), \quad M \in \mathbb{R}^{n_d \times n_d}, \; F : \mathbb{R} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_d}, \; \mathbf{u} : \mathbb{R} \to \mathbb{R}^{n_d}.$$

The Method of Lines

Consider a partial differential equations of the form

$$\text{Find } u(\mathbf{x}, t) \text{ s.t. } u_t = \mathcal{L}u, \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^d, t \in I \subseteq \mathbb{R}_+,$$

where $\mathcal{L}$ is a differential operator, either linear or nonlinear, coupled with the opportune boundary conditions, and given suitable initial conditions.
A *classical way* of approaching this task is using a **Method Of Lines** (MOL) approach, that is

1. we discretize w.r.t. the *space variables* with some method (e.g., Finite Elements/Differences/Volumes, meshfree/meshless methods, spectral methods...)

$$M\mathbf{u}_t = F(t, \mathbf{u}), \quad M \in \mathbb{R}^{n_d \times n_d}, \; F : \mathbb{R} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_d}, \; \mathbf{u} : \mathbb{R} \to \mathbb{R}^{n_d}.$$

2. now we have a (possibly nonlinear, non-autonomous) system of ODEs to which we can apply an integrator.

PDEs with fractional derivatives with respect to time

We can think of using the methods we have seen until now for solving PDEs in which the **derivative with respect to time** has been substituted by the **fractional derivative in the Caputo sense**

$$\text{Find } u(\mathbf{x}, t) \text{ s.t. } {}_{CA}D^{\alpha} u = \mathcal{L} u, \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^d, t \in I \subseteq \mathbb{R}_+.$$

PDEs with fractional derivatives with respect to time

We can think of using the methods we have seen until now for solving PDEs in which the **derivative with respect to time** has been substituted by the **fractional derivative in the Caputo sense**

$$\text{Find } u(\mathbf{x}, t) \text{ s.t. } {}_{CA}D^{\alpha} u = \mathcal{L} u, \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^d, t \in I \subseteq \mathbb{R}_+.$$

Examples:

- Time-fractional diffusion equation

$$ {}_{CA}D_t^{\alpha} u = \operatorname{div}(p(x) \operatorname{grad} u) - q(x) u + F(x, t), \quad 0 < \alpha \leq 1.$$

PDEs with fractional derivatives with respect to time

We can think of using the methods we have seen until now for solving PDEs in which the **derivative with respect to time** has been substituted by the **fractional derivative in the Caputo sense**

$$\text{Find } u(\mathbf{x}, t) \text{ s.t. } {}_{CA}D^{\alpha} u = \mathcal{L} u, \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^d, t \in I \subseteq \mathbb{R}_+.$$

Examples:

- Time-fractional diffusion equation

$$ {}_{CA}D_t^{\alpha} u = \operatorname{div}(p(x) \operatorname{grad} u) - q(x) u + F(x, t), \quad 0 < \alpha \leq 1.$$

- Time-fractional advection-dispersion equation

$$ {}_{CA}D_t^{\alpha} u = \operatorname{div}(p(x) \operatorname{grad} u) - \nu \operatorname{grad}(u), \quad 0 < \alpha \leq 1.$$

PDEs with fractional derivatives with respect to time

We can think of using the methods we have seen until now for solving PDEs in which the **derivative with respect to time** has been substituted by the **fractional derivative in the Caputo sense**

$$\text{Find } u(\mathbf{x}, t) \text{ s.t. } {}_{CA}D^{\alpha}u = \mathcal{L}u, \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^d, t \in I \subseteq \mathbb{R}_+.$$

Examples:

- Time-fractional diffusion equation

$$ {}_{CA}D_t^{\alpha}u = \operatorname{div}(p(x)\operatorname{grad} u) - q(x)u + F(x,t), \quad 0 < \alpha \leq 1.$$

- Time-fractional advection-dispersion equation

$$ {}_{CA}D_t^{\alpha}u = \operatorname{div}(p(x)\operatorname{grad} u) - \nu \operatorname{grad}(u), \quad 0 < \alpha \leq 1.$$

- Time-fractional Schrödinger equation

$$(iT_\rho)^{\alpha}\,{}_{CA}D_t^{\alpha}\psi = -\frac{L_\rho^2}{2N_m}\nabla^2\psi + N_\nu\psi, \quad 0 < \alpha \leq 1.$$

PDEs with fractional derivatives with respect to time

We can think of using the methods we have seen until now for solving PDEs in which the **derivative with respect to time** has been substituted by the **fractional derivative in the Caputo sense**

$$\text{Find } u(\mathbf{x}, t) \text{ s.t. } {}_{CA}D^{\alpha} u = \mathcal{L} u, \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^d, t \in I \subseteq \mathbb{R}_+.$$

Examples:

- Time-fractional Burgers equation equation

$$ {}_{CA}D_t^{\alpha} u = u_{xx} + A u^p u_x, \quad 0 < \alpha \leq 1, \ p > 0.$$

PDEs with fractional derivatives with respect to time

We can think of using the methods we have seen until now for solving PDEs in which the **derivative with respect to time** has been substituted by the **fractional derivative in the Caputo sense**

$$\text{Find } u(\mathbf{x}, t) \text{ s.t. } {}_{CA}D^{\alpha} u = \mathcal{L} u, \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^d, t \in I \subseteq \mathbb{R}_+.$$

Examples:

- Time-fractional Burgers equation equation

$$ {}_{CA}D_t^{\alpha} u = u_{xx} + A u^p u_x, \quad 0 < \alpha \leq 1, \ p > 0.$$

- Time-fractional Korteweg–de Vries equation

$$ {}_{CA}D_t^{\alpha} u = u_{xxx} + A u^p u_x, \quad 0 < \alpha \leq 1, \ p > 0.$$

PDEs with fractional derivatives with respect to time

We can think of using the methods we have seen until now for solving PDEs in which the **derivative with respect to time** has been substituted by the **fractional derivative in the Caputo sense**

$$\text{Find } u(\mathbf{x}, t) \text{ s.t. } {}_{CA}D^{\alpha} u = \mathcal{L} u, \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^d, t \in I \subseteq \mathbb{R}_+.$$

Examples:

- Time-fractional Burgers equation equation

$$ {}_{CA}D_t^{\alpha} u = u_{xx} + A u^p u_x, \quad 0 < \alpha \leq 1, \ p > 0.$$

- Time-fractional Korteweg–de Vries equation

$$ {}_{CA}D_t^{\alpha} u = u_{xxx} + A u^p u_x, \quad 0 < \alpha \leq 1, \ p > 0.$$

- Time-fractional (incompressible) Navier–Stokes equation

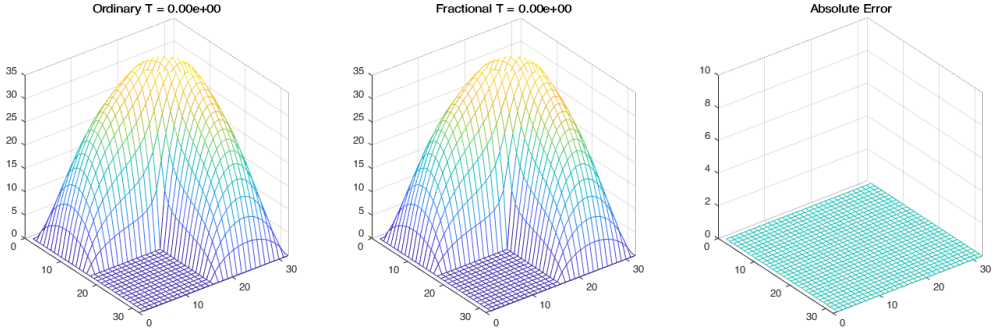
$$\begin{cases} {}_{CA}D_t^{\alpha} (u \cdot \nabla) u = \nu \nabla^2 u - \frac{1}{\rho} \nabla p + f, \\ \nabla \cdot u = 0. \end{cases} \quad 0 < \alpha \leq 1.$$

An example with diffusion

Let us consider the case of

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

and integrate it with the FBDF2 with $\tau = 10^{-2}$.

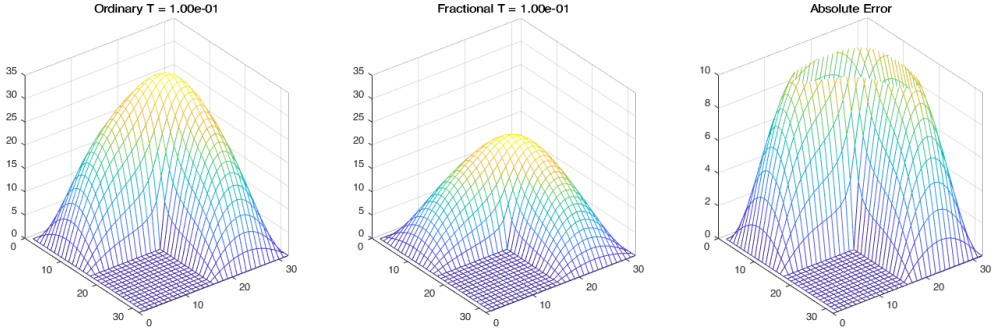


An example with diffusion

Let us consider the case of

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

and integrate it with the FBDF2 with $\tau = 10^{-2}$.

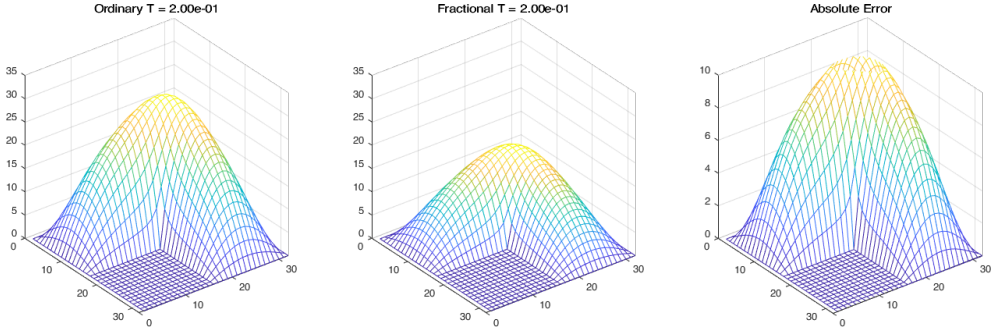


An example with diffusion

Let us consider the case of

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

and integrate it with the FBDF2 with $\tau = 10^{-2}$.

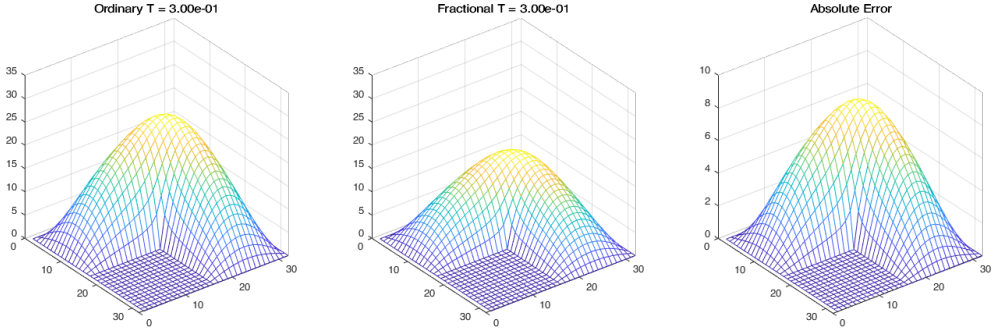


An example with diffusion

Let us consider the case of

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

and integrate it with the FBDF2 with $\tau = 10^{-2}$.

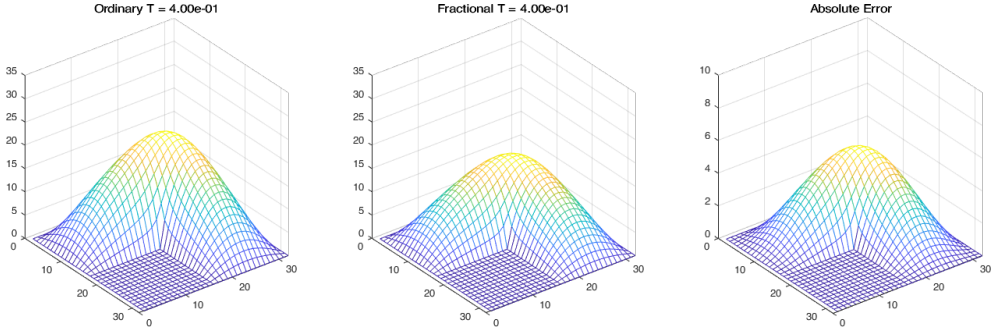


An example with diffusion

Let us consider the case of

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

and integrate it with the FBDF2 with $\tau = 10^{-2}$.

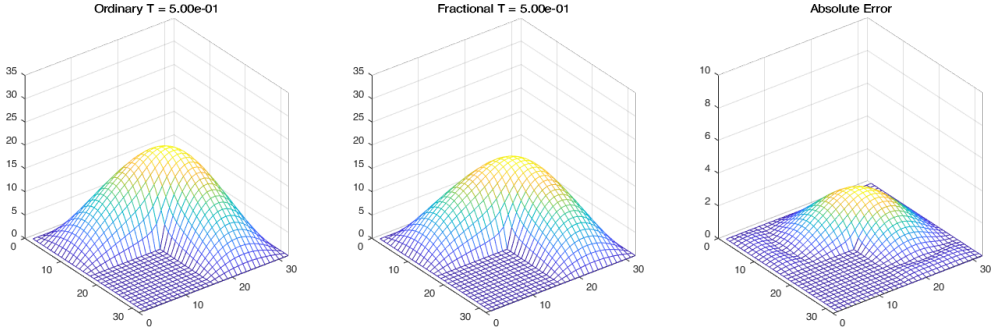


An example with diffusion

Let us consider the case of

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

and integrate it with the FBDF2 with $\tau = 10^{-2}$.

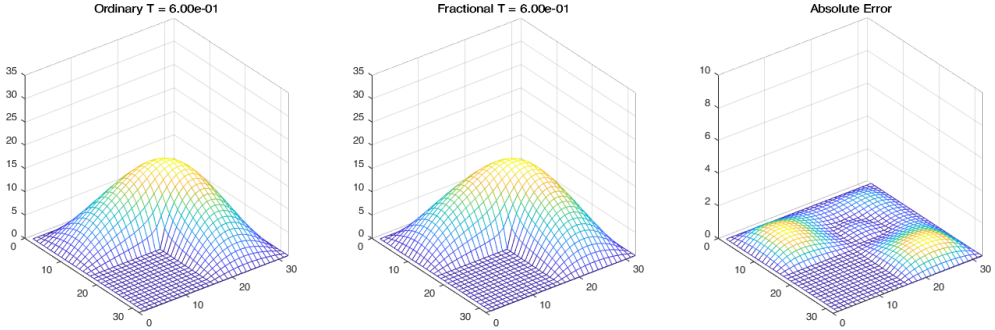


An example with diffusion

Let us consider the case of

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

and integrate it with the FBDF2 with $\tau = 10^{-2}$.

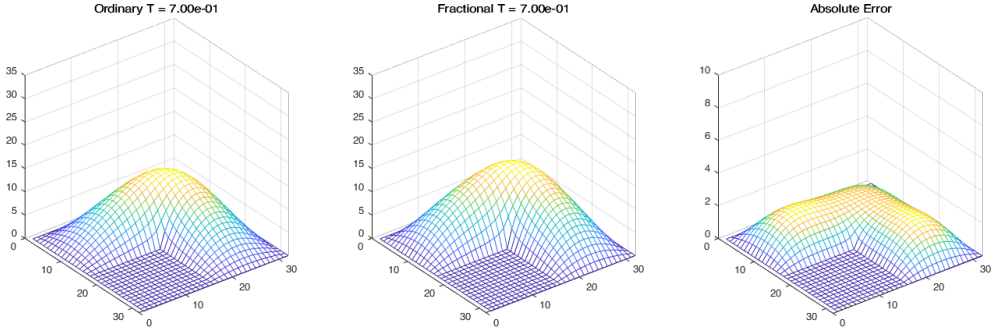


An example with diffusion

Let us consider the case of

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

and integrate it with the FBDF2 with $\tau = 10^{-2}$.

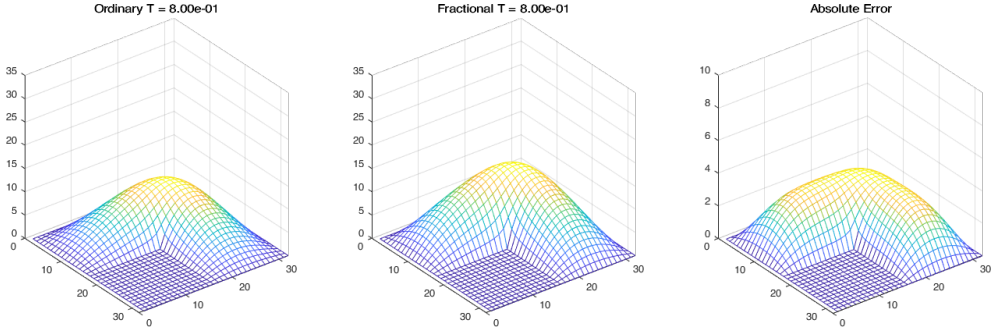


An example with diffusion

Let us consider the case of

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

and integrate it with the FBDF2 with $\tau = 10^{-2}$.

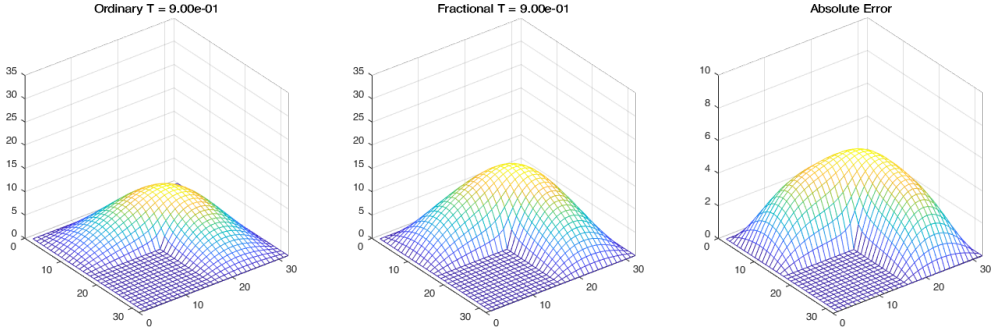


An example with diffusion

Let us consider the case of

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

and integrate it with the FBDF2 with $\tau = 10^{-2}$.

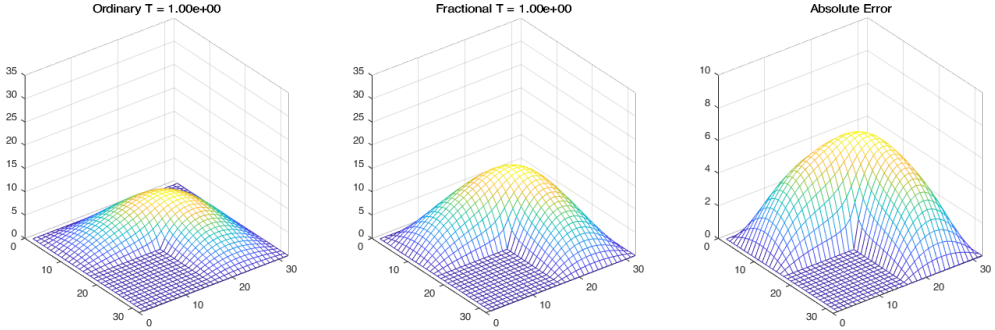


An example with diffusion

Let us consider the case of

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

and integrate it with the FBDF2 with $\tau = 10^{-2}$.

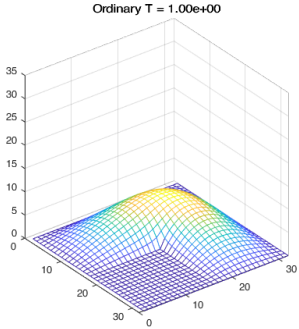


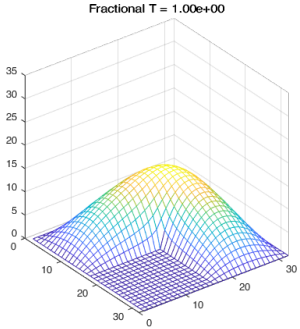
An example with diffusion

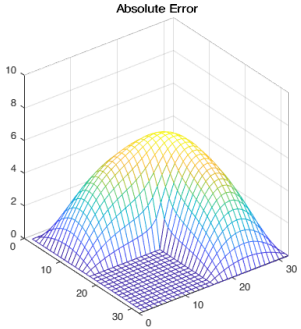
Let us consider the case of

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

and integrate it with the FBDF2 with $\tau = 10^{-2}$.







How can you describe the observed behavior?

Conclusions and next steps

- We have completed the construction of several schemes for the integration of FODEs,

Conclusions and next steps

- We have completed the construction of several schemes for the integration of FODEs,
- We have discussed the case of FODEs with multiple terms and different orders,

Conclusions and next steps

- We have completed the construction of several schemes for the integration of FODEs,
- We have discussed the case of FODEs with multiple terms and different orders,
- We started looking into some time-fractional PDEs using the Method of Lines together with our FODEs algorithms.

Conclusions and next steps

- We have completed the construction of several schemes for the integration of FODEs,
- We have discussed the case of FODEs with multiple terms and different orders,
- We started looking into some time-fractional PDEs using the Method of Lines together with our FODEs algorithms.
- Can we better describe this "subdiffusive" behavior we have observed in time-fractional diffusion equation?

Conclusions and next steps

- ✅ We have completed the construction of several schemes for the integration of FODEs,
- ✅ We have discussed the case of FODEs with multiple terms and different orders,
- ✅ We started looking into some time-fractional PDEs using the Method of Lines together with our FODEs algorithms.
- 📋 Can we better describe this "subdiffusive" behavior we have observed in time-fractional diffusion equation?
- 📋 For linear problems can we investigate the "exponential" fractional integrators?

Bibliography I

- Bagley, R. L. and P. J. Torvik (1986). "On the Fractional Calculus Model of Viscoelastic Behavior". In: *Journal of Rheology* 30.1, pp. 133–155. DOI: 10.1122/1.549887.
- Diethelm, K. (2003). "Efficient solution of multi-term fractional differential equations using $\mathrm{P(EC)}^m\mathrm{E}$ methods". In: *Computing* 71.4, pp. 305–319. ISSN: 0010-485X. DOI: 10.1007/s00607-003-0033-3. URL: https://doi.org/10.1007/s00607-003-0033-3.
- Diethelm, K. (2010). *The analysis of fractional differential equations*. Vol. 2004. Lecture Notes in Mathematics. An application-oriented exposition using differential operators of Caputo type. Springer-Verlag, Berlin, pp. viii+247. ISBN: 978-3-642-14573-5. DOI: 10.1007/978-3-642-14574-2. URL: https://doi.org/10.1007/978-3-642-14574-2.
- Ford, N. J. and J. A. Connolly (2009). "Systems-based decomposition schemes for the approximate solution of multi-term fractional differential equations". In: *J. Comput. Appl. Math.* 229.2, pp. 382–391. ISSN: 0377-0427. DOI: 10.1016/j.cam.2008.04.003. URL: https://doi.org/10.1016/j.cam.2008.04.003.

Bibliography II

- Ford, N. J. and A. C. Simpson (2001). "The numerical solution of fractional differential equations: speed versus accuracy". In: *Numer. Algorithms* 26.4, pp. 333–346. ISSN: 1017-1398. DOI: 10.1023/A:1016601312158. URL: https://doi.org/10.1023/A:1016601312158.
- Garrappa, R. (2015). "Trapezoidal methods for fractional differential equations: theoretical and computational aspects". In: *Math. Comput. Simulation* 110, pp. 96–112. ISSN: 0378-4754. DOI: 10.1016/j.matcom.2013.09.012. URL: https://doi.org/10.1016/j.matcom.2013.09.012.
- — (2018). "Numerical solution of fractional differential equations: A survey and a software tutorial". In: *Mathematics* 6.2, p. 16.
- Hairer, E., C. Lubich, and M. Schlichte (1985). "Fast numerical solution of nonlinear Volterra convolution equations". In: *SIAM J. Sci. Statist. Comput.* 6.3, pp. 532–541. ISSN: 0196-5204. DOI: 10.1137/0906037. URL: https://doi.org/10.1137/0906037.
- Henrici, P. (1974). *Applied and computational complex analysis*. Pure and Applied Mathematics. Volume 1: Power series—integration—conformal mapping—location of zeros. Wiley-Interscience [John Wiley & Sons], New York-London-Sydney, pp. xv+682.

Bibliography III

Wang, Y. and C. Li (2007). "Does the fractional Brusselator with efficient dimension less than 1 have a limit cycle?" In: *Physics Letters A* 363.5, pp. 414–419. ISSN: 0375-9601. DOI: https://doi.org/10.1016/j.physleta.2006.11.038. URL: https://www.sciencedirect.com/science/article/pii/S0375960106018020.

An introduction to fractional calculus

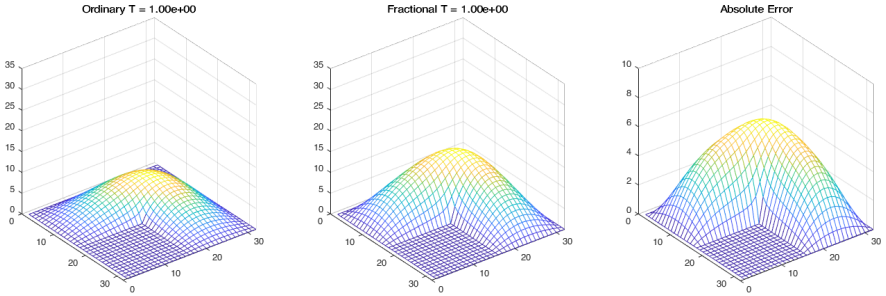
Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

June, 2022

Subdiffusion equations

At the end of the last lecture we had observed the following behavior:



for the solution of:

$$ {}_{CA}D_t^{\alpha} u = 0.05 \nabla^2 u, \quad \alpha = 0.3, 1. $$

The visual effect seemed to be a **slowing down of the diffusion**.

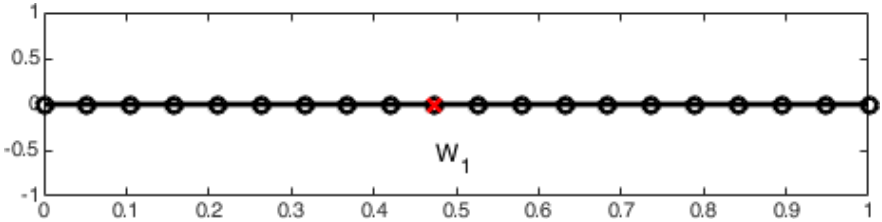
Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,

Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,
- The process can be modelled by the master equation

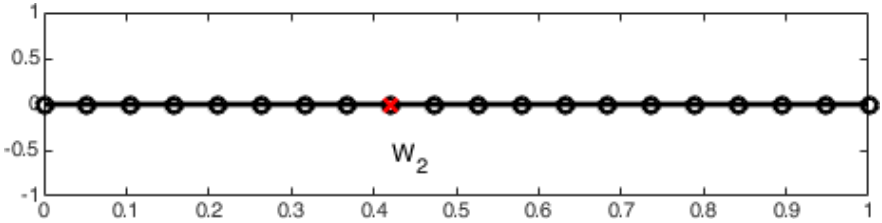
$$W_j(t+\Delta t) = \frac{1}{2}W_{j-1}(t) + \frac{1}{2}W_{j+1}(t)$$



Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,
- The process can be modelled by the master equation

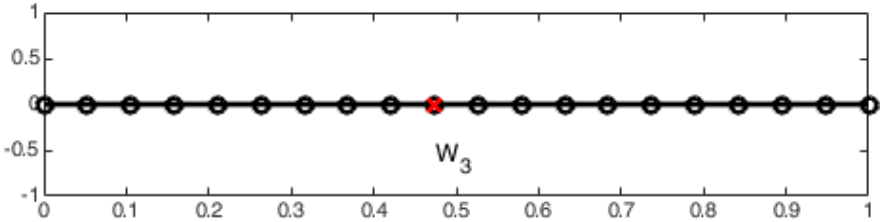
$$W_j(t+\Delta t) = \frac{1}{2}W_{j-1}(t) + \frac{1}{2}W_{j+1}(t)$$



Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,
- The process can be modelled by the master equation

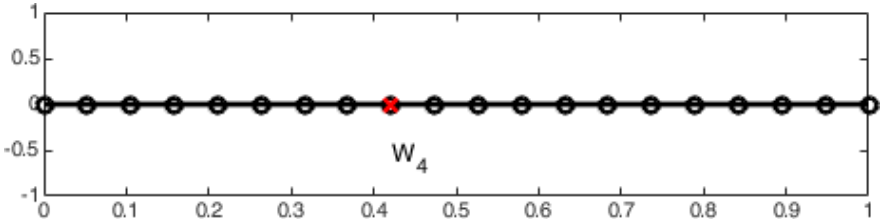
$$W_j(t+\Delta t) = \frac{1}{2}W_{j-1}(t) + \frac{1}{2}W_{j+1}(t)$$



Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,
- The process can be modelled by the master equation

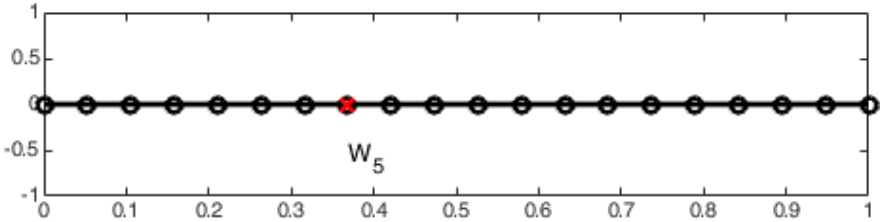
$$W_j(t+\Delta t) = \frac{1}{2}W_{j-1}(t) + \frac{1}{2}W_{j+1}(t)$$



Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,
- The process can be modelled by the master equation

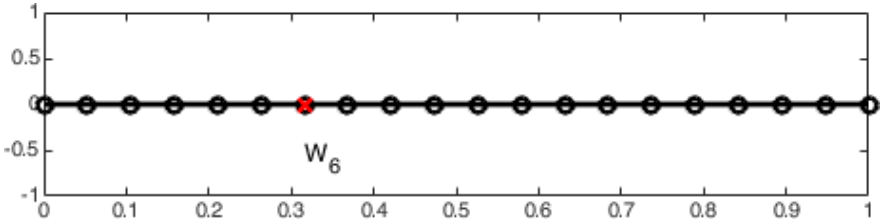
$$W_j(t+\Delta t) = \frac{1}{2}W_{j-1}(t) + \frac{1}{2}W_{j+1}(t)$$



Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,
- The process can be modelled by the master equation

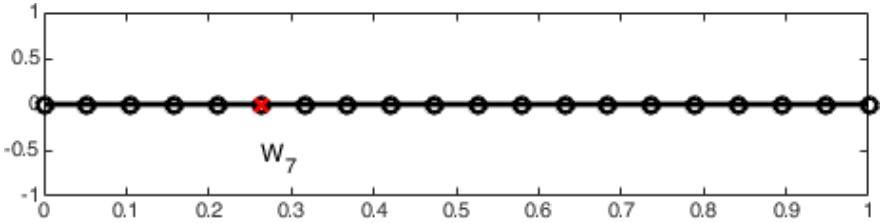
$$W_j(t+\Delta t) = \frac{1}{2}W_{j-1}(t) + \frac{1}{2}W_{j+1}(t)$$



Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,
- The process can be modelled by the master equation

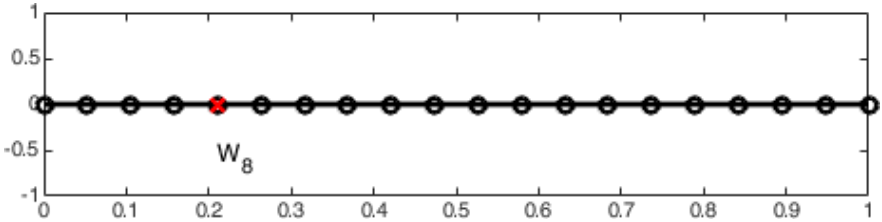
$$W_j(t+\Delta t) = \frac{1}{2}W_{j-1}(t) + \frac{1}{2}W_{j+1}(t)$$



Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,
- The process can be modelled by the master equation

$$W_j(t+\Delta t) = \frac{1}{2}W_{j-1}(t) + \frac{1}{2}W_{j+1}(t)$$



Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,
- The process can be modelled by the master equation

$$W_j(t + \Delta t) = \frac{1}{2} W_{j-1}(t) + \frac{1}{2} W_{j+1}(t)$$

Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,
- The process can be modelled by the master equation

$$W_j(t + \Delta t) = \frac{1}{2} W_{j-1}(t) + \frac{1}{2} W_{j+1}(t)$$

- The master equation defines the *pdf* to be at position j at time $t + \Delta t$ depending on the population of the two adjacent sites $j \pm 1$ at time t.

Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,
- The process can be modelled by the master equation

$$W_j(t + \Delta t) = \frac{1}{2} W_{j-1}(t) + \frac{1}{2} W_{j+1}(t)$$

- The master equation defines the *pdf* to be at position j at time $t + \Delta t$ depending on the population of the two adjacent sites $j \pm 1$ at time t.
- The prefactor $1/2$ tells us that the **process is isotropic** with respect to the left/right direction.

Brownian motion (Metzler and Klafter 2000)

- Consider a 1D lattice with cell size Δx,
- In discrete time steps of span Δt a test particle jumps to one of its neighbour sites,
- The process can be modelled by the master equation

$$W_j(t+\Delta t) = \frac{1}{2}W_{j-1}(t) + \frac{1}{2}W_{j+1}(t)$$

- The master equation defines the *pdf* to be at position j at time $t+\Delta t$ depending on the population of the two adjacent sites $j \pm 1$ at time t.
- The prefactor $1/2$ tells us that the **process is isotropic** with respect to the left/right direction.
- If we let $\Delta t \to 0$, $\Delta x \to 0$ and do a Taylor expansion in both Δ and Δx we get

$$W_j(t+\Delta t) = W_j(t) + \Delta t \frac{\partial W_j}{\partial t} + O([\Delta t]^2), \qquad \text{for } \Delta t \to 0,$$

$$W_{j\pm 1}(t) = W(x,t) \pm \Delta x \frac{\partial W}{\partial x} + \frac{(\Delta x)^2}{2}\frac{\partial^2 W}{\partial x^2} + O([\Delta x]^3), \qquad \text{for } \Delta x \to 0,$$

Brownian motion (Metzler and Klafter 2000)

We now substitute the expansions

$$W_j(t+\Delta t) = W_j(t) + \Delta t \frac{\partial W_j}{\partial t} + O([\Delta t]^2), \qquad \text{for } \Delta t \to 0,$$

$$W_{j\pm 1}(t) = W(x,t) \pm \Delta x \frac{\partial W}{\partial x} + \frac{(\Delta x)^2}{2}\frac{\partial^2 W}{\partial x^2} + O([\Delta x]^3), \qquad \text{for } \Delta x \to 0,$$

in

$$W_j(t+\Delta t) = \frac{1}{2}W_{j-1}(t) + \frac{1}{2}W_{j+1}(t)$$

obtaining

$$W(x,t) + \Delta \mathrm{t}\frac{\partial W}{\partial t} + O\left(\Delta \mathrm{t}^2\right) = W(x,t) + \frac{1}{2}\Delta \mathrm{x}^2 \frac{\partial^2 W}{\partial x^2} + O\left(\Delta \mathrm{x}^3\right)$$

Brownian motion (Metzler and Klafter 2000)

We now substitute the expansions

$$W_j(t+\Delta t) = W_j(t) + \Delta t \frac{\partial W_j}{\partial t} + O([\Delta t]^2), \qquad \text{for } \Delta t \to 0,$$

$$W_{j\pm 1}(t) = W(x,t) \pm \Delta x \frac{\partial W}{\partial x} + \frac{(\Delta x)^2}{2}\frac{\partial^2 W}{\partial x^2} + O([\Delta x]^3), \qquad \text{for } \Delta x \to 0,$$

in

$$W_j(t+\Delta t) = \frac{1}{2}W_{j-1}(t) + \frac{1}{2}W_{j+1}(t)$$

obtaining

$$\frac{\partial W}{\partial t} = \frac{\Delta x^2}{2\Delta t}\frac{\partial^2 W}{\partial x^2} + O\left(\Delta x^3 + \Delta t\right)$$

Brownian motion (Metzler and Klafter 2000)

We now substitute the expansions

$$W_j(t+\Delta t) = W_j(t) + \Delta t \frac{\partial W_j}{\partial t} + O([\Delta t]^2), \qquad \text{for } \Delta t \to 0,$$

$$W_{j\pm 1}(t) = W(x,t) \pm \Delta x \frac{\partial W}{\partial x} + \frac{(\Delta x)^2}{2}\frac{\partial^2 W}{\partial x^2} + O([\Delta x]^3), \qquad \text{for } \Delta x \to 0,$$

in

$$W_j(t+\Delta t) = \frac{1}{2}W_{j-1}(t) + \frac{1}{2}W_{j+1}(t)$$

obtaining

$$\frac{\partial W}{\partial t} = K_1 \frac{\partial^2 W}{\partial x^2}, \qquad K_1 = \lim_{\substack{\Delta x \to 0 \\ \Delta t \to 0}} \frac{\Delta x^2}{2\Delta t} < \infty.$$

Brownian motion

$$\frac{\partial W}{\partial t} = K_1 \frac{\partial^2 W}{\partial x^2}$$

Let us call X the random variable measuring the distance covered in two consecutive jumps

- Assume that the *pdf* of X (appropriately normalised) has existing moments

$$\overline{X} = \sum_i X_i, \qquad \overline{X^2},$$

  and mean time-span Δt between any two individual jump events.

Brownian motion

$$\frac{\partial W}{\partial t} = K_1 \frac{\partial^2 W}{\partial x^2}$$

Let us call X the random variable measuring the distance covered in two consecutive jumps

- Assume that the *pdf* of X (appropriately normalised) has existing moments

$$\overline{X} = \sum_i X_i, \qquad \overline{X^2},$$

  and mean time-span Δt between any two individual jump events.
- Then the **central limit theorem** assures that exists

$$V = \frac{\overline{X}}{\Delta t} \text{ (Mean velocity)} \quad K = \frac{\overline{X^2} - \overline{X}^2}{2\Delta t} \text{ (Diffusion coefficient)}$$

  and that

$$W(x,t) = \frac{1}{2\sqrt{\pi K_1 t}} \exp\left(-x^2/4K_1 t\right).$$

Brownian motion: the Fourier domain

We can rewrite

$$W(x,t) = \frac{1}{2\sqrt{\pi K_1 t}} \exp\left(-x^2/4K_1 t\right).$$

in the **Fourier domain** as

$$W(k,t) = \exp(-K_1 k^2 t), \qquad W_0(x) = \lim_{t\to 0^+} W(x,t) = \delta(x),$$

that solve the Fourier transformed diffusion equation

$$\frac{\partial W}{\partial t} = -K_1 k^2 W(k,t),$$

that is a **relaxation equation**, for a fixed wavenumber k.

From the discrete to the continuous

The **C**ontinuous **T**ime **R**andom **W**alk model (CTRW):

- Both the **length of a given jump**, and the **waiting time** elapsing between two successive jumps are drawn from a pdf $\psi(x, t)$

From the discrete to the continuous

The **C**ontinuous **T**ime **R**andom **W**alk model (CTRW):

- Both the **length of a given jump**, and the **waiting time** elapsing between two successive jumps are drawn from a pdf $\psi(x, t)$
- The jump length pdf

$$\lambda(x) = \int_0^{+\infty} \psi(x, y)\, \mathrm{d}t,$$

Jump length

$\lambda(x)\mathrm{d}x$ produces the probability for a jump length in the interval $(x, x + \mathrm{d}x)$.

From the discrete to the continuous

The **C**ontinuous **T**ime **R**andom **W**alk model (CTRW):

- Both the **length of a given jump**, and the **waiting time** elapsing between two successive jumps are drawn from a pdf $\psi(x,t)$
- The jump length pdf

$$\lambda(x) = \int_0^{+\infty} \psi(x,y)\,\mathrm{d}t,$$

- The waiting time pdf

$$w(t) = \int_{-\infty}^{+\infty} \psi(x,t)\,\mathrm{d}x$$

Waiting time

$w(t)\mathrm{d}t$ produces the probability for a waiting time in the interval $(t, t+\mathrm{d}t)$.

From the discrete to the continuous

The **C**ontinuous **T**ime **R**andom **W**alk model (CTRW):

- Both the **length of a given jump**, and the **waiting time** elapsing between two successive jumps are drawn from a pdf $\psi(x, t)$
- The jump length pdf

$$\lambda(x) = \int_0^{+\infty} \psi(x, y)\, \mathrm{d}t,$$

- The waiting time pdf

$$w(t) = \int_{-\infty}^{+\infty} \psi(x, t)\, \mathrm{d}x$$

- If the jump length and waiting time are **independent random variables** then:

$$\psi(x, t) = w(t)\lambda(x)$$

Characterization of CTRW

To categorise different CTRW one can look at the quantities

$$T = \int_0^{+\infty} t w(t)\,\mathrm{d}t, \text{ (Characteristic waiting time)},$$

and

$$\Sigma^2 = \int_{-\infty}^{+\infty} x^2 \lambda(x)\,\mathrm{d}x \text{ (Jump length variance)},$$

specifically, are they **finite**? Do they **diverge**?

Characterization of CTRW

To categorise different CTRW one can look at the quantities

$$T = \int_0^{+\infty} t w(t)\,\mathrm{d}t, \text{ (Characteristic waiting time)},$$

and

$$\Sigma^2 = \int_{-\infty}^{+\infty} x^2 \lambda(x)\,\mathrm{d}x \text{ (Jump length variance)},$$

specifically, are they **finite**? Do they **diverge**?
The master (Langevin) equation for this process is then given by

$$\eta(x,t) = \int_{-\infty}^{+\infty} \mathrm{d}x' \int_0^{+\infty} \mathrm{d}t'\, \eta(x',t')\psi(x-x',t-t') + \delta(x)\delta(t),$$

Characterization of CTRW

To categorise different CTRW one can look at the quantities

$$T = \int_0^{+\infty} t w(t)\,\mathrm{d}t, \text{ (Characteristic waiting time)},$$

and

$$\Sigma^2 = \int_{-\infty}^{+\infty} x^2 \lambda(x)\,\mathrm{d}x \text{ (Jump length variance)},$$

specifically, are they **finite**? Do they **diverge**?
The master (Langevin) equation for this process is then given by

$$\eta(x,t) = \int_{-\infty}^{+\infty} \mathrm{d}x' \int_0^{+\infty} \mathrm{d}t'\, \eta(x',t')\psi(x-x',t-t') + \delta(x)\delta(t),$$

Pdf of having arrived at position x at time t – $\eta(x,t)$ –

Characterization of CTRW

To categorise different CTRW one can look at the quantities

$$T = \int_0^{+\infty} t w(t)\,\mathrm{d}t, \text{ (Characteristic waiting time)},$$

and

$$\Sigma^2 = \int_{-\infty}^{+\infty} x^2 \lambda(x)\,\mathrm{d}x \text{ (Jump length variance)},$$

specifically, are they **finite**? Do they **diverge**?
The master (Langevin) equation for this process is then given by

$$\eta(x,t) = \int_{-\infty}^{+\infty} \mathrm{d}x' \int_0^{+\infty} \mathrm{d}t'\, \eta(x',t')\psi(x-x',t-t') + \delta(x)\delta(t),$$

Pdf of having arrived at position x at time t – $\eta(x,t)$ – having just arrived at x' at time t' – $\eta(x',t')$ –

Characterization of CTRW

To categorise different CTRW one can look at the quantities

$$T = \int_0^{+\infty} t w(t)\,\mathrm{d}t, \text{ (Characteristic waiting time)},$$

and

$$\Sigma^2 = \int_{-\infty}^{+\infty} x^2 \lambda(x)\,\mathrm{d}x \text{ (Jump length variance)},$$

specifically, are they **finite**? Do they **diverge**?
The master (Langevin) equation for this process is then given by

$$\eta(x,t) = \int_{-\infty}^{+\infty} \mathrm{d}x' \int_0^{+\infty} \mathrm{d}t'\, \eta(x',t')\psi(x-x',t-t') + \delta(x)\delta(t),$$

Pdf of having arrived at position x at time t – $\eta(x,t)$ – having just arrived at x' at time t' – $\eta(x',t')$ – with initial condition $\delta(x)$.

Characterization of CTRW

Then if we use

$$\eta(x,t) = \int_{-\infty}^{+\infty} \mathrm{d}x' \int_0^{+\infty} \mathrm{d}t' \, \eta(x',t')\psi(x-x',t-t') + \delta(x)\delta(t),$$

we can write the pdf of being in x at time t as

$$W(x,t) = \int_0^t \eta(x,t')\Psi(t-t'), \mathrm{d}t, \qquad \Psi(t) = 1 - \int_0^t w(t')\,\mathrm{d}t',$$

where the latter is the cumulative probability assigned to the probability of no jump event during the time interval $t - t'$.

Fact

If both T and Σ^2 are finite the long-time limit corresponds to Brownian motion, e.g., $w(t) = \tau^{-1} exp(-t/\tau)$, $T = \tau$, $\lambda(x) = (4\pi\sigma^2)^{-1/2}\exp(-x^2/4\sigma^2)$, $\Sigma^2 = 2\sigma^2$, we recover the standard diffusion equation.

The CTRW in the Fourier-Laplace domain

We take

$$W(x,t) = \int_0^t \eta(x,t')\Psi(t-t'),\mathrm{d}t, \qquad \Psi(t) = 1 - \int_0^t w(t')\,\mathrm{d}t',$$

and rewrite it again in the **Fourier-Laplace domain** (Fourier for the space variable, Laplace for the time one) as

$$W(k,u) = \frac{1-w(u)}{u}\frac{W_0(k)}{1-\psi(k,u)}, \qquad W_0(k) = \int_{-\infty}^{+\infty} W_0(x)e^{-i2\pi kx}\,\mathrm{d}x.$$

In the **Brownian case**

$$w(u) \sim 1 - u\tau + O(\tau^2), \quad \lambda(k) \sim 1 - \sigma^2 k^2 + O(k^4), \quad W_0(x) = \delta(x)$$

then

$$W(k,u) = \frac{1}{u + K_1 k^2}, \quad K_1 = {}^{\sigma^2}/_{\tau}.$$

The case of long rests

Long rests

The **characteristic waiting time** $T = \int_0^{+\infty} t w(t)\,\mathrm{d}t$ **diverges**, but the jump length variance $\Sigma^2 = \int_{-\infty}^{+\infty} x^2 \lambda(x)\,\mathrm{d}x$ is finite.

The case of long rests

Long rests

The **characteristic waiting time** $T = \int_0^{+\infty} t w(t)\,\mathrm{d}t$ **diverges**, but the jump length variance $\Sigma^2 = \int_{-\infty}^{+\infty} x^2 \lambda(x)\,\mathrm{d}x$ is finite.

- To realize this we can select

$$w(t) \sim A_\alpha \left(\tau/t\right)^{1+\alpha}, \qquad 0 < \alpha < 1,$$

The case of long rests

Long rests

The **characteristic waiting time** $T = \int_0^{+\infty} t w(t)\,\mathrm{d}t$ **diverges**, but the jump length variance $\Sigma^2 = \int_{-\infty}^{+\infty} x^2 \lambda(x)\,\mathrm{d}x$ is finite.

- To realize this we can select

$$w(t) \sim A_\alpha \left(\tau/t\right)^{1+\alpha}, \qquad 0 < \alpha < 1,$$

- For the jump pdf we use again the Gaussian jump length

$$\lambda(x) = (4\pi\sigma^2)^{-1/2} \exp(-x^2/4\sigma^2).$$

The case of long rests

Long rests

The **characteristic waiting time** $T = \int_0^{+\infty} t w(t)\,\mathrm{d}t$ **diverges**, but the jump length variance $\Sigma^2 = \int_{-\infty}^{+\infty} x^2 \lambda(x)\,\mathrm{d}x$ is finite.

- To realize this we can select

$$w(t) \sim A_\alpha \left(\tau/t\right)^{1+\alpha}, \qquad 0 < \alpha < 1,$$

- For the jump pdf we use again the Gaussian jump length

$$\lambda(x) = (4\pi\sigma^2)^{-1/2} \exp(-x^2/4\sigma^2).$$

- To get the form of the equation we first go to the Laplace domain:

$$w(u) \sim 1 - (u\tau)^\alpha,$$

The case of long rests

Long rests

The **characteristic waiting time** $T = \int_0^{+\infty} t w(t)\,\mathrm{d}t$ **diverges**, but the jump length variance $\Sigma^2 = \int_{-\infty}^{+\infty} x^2 \lambda(x)\,\mathrm{d}x$ is finite.

- To realize this we can select

$$w(t) \sim A_\alpha \left(\tau/t\right)^{1+\alpha}, \qquad 0 < \alpha < 1,$$

- For the jump pdf we use again the Gaussian jump length

$$\lambda(x) = (4\pi\sigma^2)^{-1/2} \exp(-x^2/4\sigma^2).$$

- To get the form of the equation we first go to the Laplace domain:

$$w(u) \sim 1 - (u\tau)^\alpha,$$

- and then obtain the expression for $W(k,u)$ in the Fourier-Laplace space

$$W(k,u) = {}^{W_0(k)}/_{u}/(1+K_\alpha u^{-\alpha} k^2).$$

The case of long rests

To get an expression of the equation we use the Laplace transform for fractional integrals:

$$\mathcal{L}\left\{I_{[0,t]}^{-\alpha} W(x,t)\right\} = u^{-\alpha} W(x,u), \qquad \alpha \geq 0,$$

and together with

$$W(k,u) = \frac{W_0(k)/u}{(1 + K_\alpha u^{-\alpha} k^2)}.$$

we infer the fractional integral equation

$$W(x,t) - W_0(x) = I_{[0,t]} K_\alpha \frac{\partial^2}{\partial x^2} W(x,t).$$

The case of long rests

To get an expression of the equation we use the Laplace transform for fractional integrals:

$$\mathcal{L}\left\{I_{[0,t]}^{-\alpha} W(x,t)\right\} = u^{-\alpha} W(x,u), \qquad \alpha \geq 0,$$

and together with

$$W(k,u) = \frac{W_0(k)/u}{(1 + K_\alpha u^{-\alpha} k^2)}.$$

we infer the fractional integral equation, and apply derivative w.r.t. to time

$$\frac{\partial}{\partial t}\left(W(x,t) - W_0(x)\right) = \frac{\partial}{\partial t}\left(I_{[0,t]} K_\alpha \frac{\partial^2}{\partial x^2} W(x,t)\right).$$

The case of long rests

To get an expression of the equation we use the Laplace transform for fractional integrals:

$$\mathcal{L}\left\{I_{[0,t]}^{-\alpha} W(x,t)\right\} = u^{-\alpha} W(x,u), \qquad \alpha \geq 0,$$

and together with

$$W(k,u) = \frac{W_0(k)/u}{(1 + K_\alpha u^{-\alpha} k^2)}.$$

we infer the fractional integral equation

$$\frac{\partial W}{\partial t} = {}_{RL}D_{[0,t]}^{\alpha} K_\alpha \frac{\partial^2}{\partial x^2} W(x,t).$$

The case of long rests

To get an expression of the equation we use the Laplace transform for fractional integrals:

$$\mathcal{L}\left\{I_{[0,t]}^{-\alpha}W(x,t)\right\} = u^{-\alpha}W(x,u), \qquad \alpha \geq 0,$$

and together with

$$W(k,u) = \frac{W_0(k)/u}{(1+K_\alpha u^{-\alpha}k^2)}.$$

we infer the fractional integral equation

$$\frac{\partial W}{\partial t} = {}_{RL}D_{[0,t]}^{\alpha} K_\alpha \frac{\partial^2}{\partial x^2} W(x,t).$$

We can compute also the mean squared displacement

$$\langle x^2(t)\rangle = \mathcal{L}^{-1}\left\{\lim_{k\to 0} -\frac{d^2}{dk^2}W(k,u)\right\} = \frac{2K_\alpha}{\Gamma(1+\alpha)}t^\alpha.$$

The case of long rests

We have obtained a Fractional Differential Equation:

$$\frac{\partial W}{\partial t} = {}_{RL}D^{\alpha}_{[0,t]} K_\alpha \frac{\partial^2}{\partial x^2} W(x,t), \qquad 0 < \alpha < 1$$

but this is not the model we started looking at, that was

$${}_{CA}D^{\alpha}_{[0,t]} W = K_\alpha \frac{\partial^2}{\partial x^2} W(x,t), \qquad 0 < \alpha < 1$$

❓ Are they related?

The case of long rests

We have obtained a Fractional Differential Equation:

$$\frac{\partial W}{\partial t} = {}_{RL}D^{\alpha}_{[0,t]} K_\alpha \frac{\partial^2}{\partial x^2} W(x,t), \qquad 0 < \alpha < 1$$

but this is not the model we started looking at, that was

$${}_{CA}D^{\alpha}_{[0,t]} W = K_\alpha \frac{\partial^2}{\partial x^2} W(x,t), \qquad 0 < \alpha < 1$$

❓ Are they related? It turns out that this is indeed the case (Sokolov and Klafter 2005), the proof involves doing some work in inverting Fourier-Laplace transform.

The case of long rests

We have obtained a Fractional Differential Equation:

$$\frac{\partial W}{\partial t} = {}_{RL}D^{\alpha}_{[0,t]} K_\alpha \frac{\partial^2}{\partial x^2} W(x,t), \qquad 0 < \alpha < 1$$

but this is not the model we started looking at, that was

$${}_{CA}D^{\alpha}_{[0,t]} W = K_\alpha \frac{\partial^2}{\partial x^2} W(x,t), \qquad 0 < \alpha < 1$$

❓ Are they related? It turns out that this is indeed the case (Sokolov and Klafter 2005), the proof involves doing some work in inverting Fourier-Laplace transform.

We now have an *interpretation* of what a Fractional Derivative with respect to time is. We will come back to this when we will speak about fractional derivative with respect to space.

“Exponential” Fractional Integrators

We start from the FDE

$$\begin{cases} {}_{CA}D^{\alpha}_{[t_0,t]}u(t) + \lambda y(t) = f(t), \\ u(0) = u_0, \end{cases} \qquad \alpha \in \mathbb{R}_{>0}, \quad \lambda \in \mathbb{R}, \; u(t) : [t_0, T] \to \mathbb{R}.$$

Then we rewrite the solution as

$$u(t) = e_{\alpha,1}(t - t_0; \lambda) u_0 + \int_{t_0}^{t} e_{\alpha,\alpha}(t - s; \lambda) f(s) \, \mathrm{d}s, \quad e_{\alpha,\beta} = t^{\beta-1} E_{\alpha,\beta}(-\lambda t^{\alpha}),$$

for $E_{\alpha,\beta}(z)$ the Mittag-Leffler (ML) function with two parameters.

“Exponential” Fractional Integrators

We start from the FDE

$$\begin{cases} {}_{CA}D^{\alpha}_{[t_0,t]}u(t) + \lambda y(t) = f(t), \\ u(0) = u_0, \end{cases} \qquad \alpha \in \mathbb{R}_{>0}, \quad \lambda \in \mathbb{R}, \; u(t) : [t_0, T] \to \mathbb{R}.$$

Then we rewrite the solution as

$$u(t) = e_{\alpha,1}(t - t_0; \lambda)u_0 + \int_{t_0}^{t} e_{\alpha,\alpha}(t - s; \lambda)f(s)\,\mathrm{d}s, \quad e_{\alpha,\beta} = t^{\beta-1}E_{\alpha,\beta}(-\lambda t^{\alpha}),$$

for $E_{\alpha,\beta}(z)$ the Mittag-Leffler (ML) function with two parameters.

- We can use this formulation to build different PI rules,

“Exponential” Fractional Integrators

We start from the FDE

$$\begin{cases} {}_{CA}D^{\alpha}_{[t_0,t]}u(t) + \lambda y(t) = f(t), \\ u(0) = u_0, \end{cases} \qquad \alpha \in \mathbb{R}_{>0}, \quad \lambda \in \mathbb{R}, \; u(t) : [t_0, T] \to \mathbb{R}.$$

Then we rewrite the solution as

$$u(t) = e_{\alpha,1}(t - t_0; \lambda)u_0 + \int_{t_0}^{t} e_{\alpha,\alpha}(t - s; \lambda)f(s)\,\mathrm{d}s, \quad e_{\alpha,\beta} = t^{\beta-1}E_{\alpha,\beta}(-\lambda t^{\alpha}),$$

for $E_{\alpha,\beta}(z)$ the Mittag-Leffler (ML) function with two parameters.

- We can use this formulation to build different PI rules,
- We can use it to address the problem

$${}_{CA}D^{\alpha}_{[t_0,t]}U(t) + Ay(t) = F(U(t)), \quad U(0) = U_0.$$

Evaluation of the ML function

For both the approaches we need reliable ways for **computing** the **ML function** on both the **real line** and with **matrix argument**.

Evaluation of the ML function

For both the approaches we need reliable ways for **computing** the **ML function** on both the **real line** and with **matrix argument**.

Scalar case Inversion of the Laplace transform via the Optimal Parabola Contour selection algorithm (Garrappa 2015),

Evaluation of the ML function

For both the approaches we need reliable ways for **computing** the **ML function** on both the **real line** and with **matrix argument**.

Scalar case: Inversion of the Laplace transform via the Optimal Parabola Contour selection algorithm (Garrappa 2015),

Matrix argument: To apply algorithm for matrix-function evaluation we may need also the value of the derivative of the ML function, e.g., Schur-Parlett type algorithm (Garrappa and Popolizio 2018; Higham and Liu 2021).

In general, we expect to mostly need matrix function–times–vector operations:

$$\mathbf{y} = E_{\alpha,\beta}(A)\mathbf{v}, \qquad A \in \mathbb{R}^{n\times n}, \quad \mathbf{y}, \mathbf{v} \in \mathbb{R}^n.$$

Evaluation of the ML function

For both the approaches we need reliable ways for **computing** the **ML function** on both the **real line** and with **matrix argument**.

Scalar case: Inversion of the Laplace transform via the Optimal Parabola Contour selection algorithm (Garrappa 2015),

Matrix argument: To apply algorithm for matrix-function evaluation we may need also the value of the derivative of the ML function, e.g., Schur-Parlett type algorithm (Garrappa and Popolizio 2018; Higham and Liu 2021).

In general, we expect to mostly need matrix function–times–vector operations:

$$\mathbf{y} = E_{\alpha,\beta}(A)\mathbf{v}, \qquad A \in \mathbb{R}^{n\times n}, \quad \mathbf{y}, \mathbf{v} \in \mathbb{R}^n.$$

We postpone it to after we have discussed the actual necessities we have.

PI - “Exponential” Fractional Integrators

We start from the formula

$$u(t) = e_{\alpha,1}(t - t_0; \lambda) u_0 + \int_{t_0}^{t} e_{\alpha,\alpha}(t - s; \lambda) f(s) \, \mathrm{d}s, \quad e_{\alpha,\beta} = t^{\beta-1} E_{\alpha,\beta}(-\lambda t^{\alpha}),$$

and select a grid $\{t_i\}_{i=0}^{N}$, then

$$u(t_n) = e_{\alpha,1}(t_n - t_0; \lambda) u_0 + \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} e_{\alpha,\alpha}(t_n - s; \lambda) f(s) \, \mathrm{d}s.$$

PI - "Exponential" Fractional Integrators

We start from the formula

$$u(t) = e_{\alpha,1}(t - t_0; \lambda) u_0 + \int_{t_0}^{t} e_{\alpha,\alpha}(t - s; \lambda) f(s) \, \mathrm{d}s, \quad e_{\alpha,\beta} = t^{\beta-1} E_{\alpha,\beta}(-\lambda t^{\alpha}),$$

and select a grid $\{t_i\}_{i=0}^{N}$, then

$$u(t_n) = e_{\alpha,1}(t_n - t_0; \lambda) u_0 + \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} e_{\alpha,\alpha}(t_n - s; \lambda) f(s) \, \mathrm{d}s.$$

- In general we have

$$e_{\alpha,\beta}(t; \lambda) = \tau^{\beta-1} e_{\alpha,\beta}(t/\tau; \tau^{\alpha} \lambda)$$

PI - "Exponential" Fractional Integrators

We start from the formula

$$u(t) = e_{\alpha,1}(t - t_0; \lambda) u_0 + \int_{t_0}^{t} e_{\alpha,\alpha}(t - s; \lambda) f(s)\, \mathrm{d}s, \quad e_{\alpha,\beta} = t^{\beta-1} E_{\alpha,\beta}(-\lambda t^{\alpha}),$$

and select a grid $\{t_i\}_{i=0}^{N}$, then

$$u(t_n) = e_{\alpha,1}(t_n - t_0; \lambda) u_0 + \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} e_{\alpha,\alpha}(t_n - s; \lambda) f(s)\, \mathrm{d}s.$$

- In general we have
$$e_{\alpha,\beta}(t; \lambda) = \tau^{\beta-1} e_{\alpha,\beta}(t/\tau; \tau^{\alpha}\lambda)$$
- For $s \in [t_j, t_{j+1}]$ let us consider the *change of variables* $s = t_j + r\tau,\ r \in [0, 1]$

PI - "Exponential" Fractional Integrators

We start from the formula

$$u(t) = e_{\alpha,1}(t - t_0; \lambda) u_0 + \int_{t_0}^{t} e_{\alpha,\alpha}(t - s; \lambda) f(s) \, \mathrm{d}s, \quad e_{\alpha,\beta} = t^{\beta-1} E_{\alpha,\beta}(-\lambda t^{\alpha}),$$

and select a grid $\{t_i\}_{i=0}^{N}$, then

$$u(t_n) = e_{\alpha,1}(t_n - t_0; \lambda) u_0 + \tau^{\alpha} \sum_{j=0}^{n-1} \int_0^1 e_{\alpha,\alpha}(^{(t-t_j)}/_{\tau} - r; \tau^{\alpha}\lambda) f(t_j + r\tau) \, \mathrm{d}r.$$

- In general we have
$$e_{\alpha,\beta}(t; \lambda) = \tau^{\beta-1} e_{\alpha,\beta}(^{t}/_{\tau}; \tau^{\alpha}\lambda)$$
- For $s \in [t_j, t_{j+1}]$ let us consider the *change of variables* $s = t_j + r\tau, \ r \in [0, 1]$

PI - "Exponential" Fractional Integrators

Then a PI rule for

$$u(t_n) = e_{\alpha,1}(t_n - t_0;\lambda)u_0 + \tau^\alpha \sum_{j=0}^{n-1} \int_0^1 e_{\alpha,\alpha}(^{(t-t_j)}/_\tau - r;\tau^\alpha\lambda) f(t_j + r\tau)\,\mathrm{d}r.$$

is obtained by selecting $q+1$ *distinct* nodes $0 \le c_0 < c_1 < \cdots < c_q \le 1$ and replacing $f(t_j + r\tau)$ with

$$p_j^{[q]}(t_j + r\tau) = \sum_{\ell=0}^{q} L_\ell^{[q]}(r) f(t_j + c_\ell\tau), \quad r \in [0,1], \quad L_\ell^{[q]} \text{ Lagrange basis element of degree } q.$$

PI - "Exponential" Fractional Integrators

Then the PI rule is

$$u^{(n)} = e_{\alpha,1}(t_n - t_0;\lambda)y_0 + \tau^\alpha \sum_{j=0}^{n-1}\sum_{\ell=0}^{q} \omega_\ell^{[q;\alpha]}(n-j;\tau^\alpha\lambda) f(t_j + c_\ell\tau).$$

is obtained by selecting $q+1$ *distinct* nodes $0 \le c_0 < c_1 < \cdots < c_q \le 1$ and replacing $f(t_j + r\tau)$ with

$$p_j^{[q]}(t_j + r\tau) = \sum_{\ell=0}^{q} L_\ell^{[q]}(r) f(t_j + c_\ell\tau), \quad r \in [0,1], \quad L_\ell^{[q]} \text{ Lagrange basis element of degree } q.$$

And selecting the weights

$$\omega_\ell^{[q;\alpha]}(n,z) = \int_0^1 e_{\alpha,\alpha}(n-j-r;z) L_\ell^{[q]}(r)\,\mathrm{d}r.$$

PI - "Exponential" Fractional Integrators

Theorem (Garrappa and Popolizio 2011, Theorem 4.2)

Let $\alpha > 0$ and $f(t) \in \mathcal{C}^{q+2}([t_0, T])$. The error of a q-step exponential PI rule is given by

$$u(t_n) - u^{(n)} = \tau^{q+1} \frac{C_0^{[q]}}{(q+1)!} \int_{t_0}^{t_n} e_{\alpha,\alpha}(t_n - s; \lambda) f^{(q+1)}(s)\, \mathrm{d}s + O(\tau^{q+1+\alpha}),$$

where the constant $C_0^{[q]}$ depends only on the nodes c_ℓ.

- For $q = 2$, $c_0 = 0$, $c_1 = 1/2$ $c_2 = 1$, one finds $C_0^{[2]} = 0$, thus an interpolatory formula of order $O(\tau^{q+1+\alpha})$.
- The **general idea** is to select nodes c_ℓ in such way that

$$C_\nu^{[q]} = \int_0^1 \omega_q(r) \xi(1-\nu, 1-r)\, \mathrm{d}r, \quad \nu \in \mathbb{R},$$

for ξ, the *Hurwitz zeta function*, are zeroed out in the error expansion for the method.

The MOL/Matrix case

Let us go back to the case that sparked our interest in going "exponential", that was the MOL problem

$$\begin{cases} {}_{CA}D^{\alpha}_{[0,t]}\mathbf{u}(t) + A\mathbf{u}(t) = \mathbf{g}(t), \quad t > 0, \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

By the variation of constant formula, we have seen that we can express the solution as

$$\mathbf{u}(t) = E_{\alpha,1}(-t^{\alpha}A)\mathbf{u}_0 + \int_0^t (t-s)^{\alpha-1} E_{\alpha,\alpha}(-A(t-s)^{\alpha})\mathbf{g}(s)\,\mathrm{d}s.$$

The MOL/Matrix case

Let us go back to the case that sparked our interest in going "exponential", that was the MOL problem

$$\begin{cases} {}_{CA}D^{\alpha}_{[0,t]}\mathbf{u}(t) + A\mathbf{u}(t) = \mathbf{g}(t), \quad t > 0, \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

By the variation of constant formula, we have seen that we can express the solution as

$$\mathbf{u}(t) = E_{\alpha,1}(-t^{\alpha}A)\mathbf{u}_0 + \int_0^t (t-s)^{\alpha-1} E_{\alpha,\alpha}(-A(t-s)^{\alpha})\mathbf{g}(s)\,\mathrm{d}s.$$

- In the general case we then have to apply one of the PI rules to compute the integral term,

The MOL/Matrix case

Let us go back to the case that sparked our interest in going "exponential", that was the MOL problem

$$\begin{cases} {}_{CA}D^{\alpha}_{[0,t]}\mathbf{u}(t) + A\mathbf{u}(t) = \mathbf{g}(t), & t > 0, \\ \mathbf{u}(0) = \mathbf{u}_0. \end{cases}$$

By the variation of constant formula, we have seen that we can express the solution as

$$\mathbf{u}(t) = E_{\alpha,1}(-t^{\alpha}A)\mathbf{u}_0 + \int_0^t (t-s)^{\alpha-1} E_{\alpha,\alpha}(-A(t-s)^{\alpha})\mathbf{g}(s)\,\mathrm{d}s.$$

- In the general case we then have to apply one of the PI rules to compute the integral term,
- If $g(s) = \sum_{k=0}^{q} s^k \mathbf{v}_k$ for some vectors, we can compute the integral on the right-hand side in *closed form* and obtain

$$\mathbf{u}(t) = E_{\alpha,1}(-t^{\alpha}A)\mathbf{y}_0 + \sum_{k=0}^{q} \Gamma(k+1) t^{\alpha+k} E_{\alpha,\alpha+k+1}(-t^{\alpha}A)\mathbf{v}_k, \qquad t > 0.$$

Matrix functions: the normal case

If A is a normal matrix, and f is a function existing on the spectrum of A, then

$$f(A) = Uf(\Lambda)U^H, \quad U^H U = I, \quad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \; A\mathbf{u}_i = \lambda_i \mathbf{u}_i, \quad U = [\mathbf{u}_1, \ldots, \mathbf{u}_n].$$

This is, e.g., sufficient for the cases in which

- A is the discretization of a self-adjoint operator,
- A is symmetric.

$E_{\alpha,\beta}(z)$ is an **analytic function**, and therefore we can compute it for every possible eigenvalue λ in the spectrum of A.

Matrix functions: the normal case

If A is a normal matrix, and f is a function existing on the spectrum of A, then

$$f(A) = Uf(\Lambda)U^H, \quad U^H U = I, \quad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \; A\mathbf{u}_i = \lambda_i \mathbf{u}_i, \quad U = [\mathbf{u}_1, \ldots, \mathbf{u}_n].$$

This is, e.g., sufficient for the cases in which

- A is the discretization of a self-adjoint operator,
- A is symmetric.

$E_{\alpha,\beta}(z)$ is an **analytic function**, and therefore we can compute it for every possible eigenvalue λ in the spectrum of A.

What about the *non-normal* and *nond-diagonalizable* case? For diagonalizable matrices, we can use the eigendecomposition at the same way.

Matrix functions: the Jordan Canonical Form

Jordan Canonical Form

We recall that any matrix $A \in \mathbb{C}^{n\times n}$ can be expressed in Jordan canonical form as

$$Z^{-1}AZ = J = \mathrm{diag}(J_1, \ldots, J_p), \qquad \text{for } J_k = J_k(\lambda_k) = \begin{bmatrix} \lambda_k & 1 & & \\ & \lambda_k & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{m_k\times m_k},$$

where Z is nonsingular and $m_1 + m_2 + \ldots + m_p = n$. If each block in which the eigenvalue λ_k appears is of size 1 then λ_k is said to be a *semisimple* eigenvalue.

- This is a *theoretical object*, it is useful to prove and define *things*, not to implement *things*.

Matrix functions: the Jordan Canonical Form

Jordan Canonical Form

We recall that any matrix $A \in \mathbb{C}^{n\times n}$ can be expressed in Jordan canonical form as

$$Z^{-1}AZ = J = \mathrm{diag}(J_1, \ldots, J_p), \qquad \text{for } J_k = J_k(\lambda_k) = \begin{bmatrix} \lambda_k & 1 & & \\ & \lambda_k & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{m_k\times m_k},$$

where Z is nonsingular and $m_1 + m_2 + \ldots + m_p = n$. If each block in which the eigenvalue λ_k appears is of size 1 then λ_k is said to be a *semisimple* eigenvalue.

- This is a *theoretical object*, it is useful to prove and define *things*, not to implement *things*.
- Now that we have a decomposition of the matrix, we need to introduce a suitable definition of **being defined on the spectrum**.

Matrix functions: the general case

Let us denote by $\lambda_1, \ldots, \lambda_s$ the distinct eigenvalues of A, and by n_i the order of the largest Jordan block in which the λ_i appears, i.e., the *index* of the eigenvalue λ_i.

Defined on the spectrum

The function f is *defined on the spectrum of* A if the values

$$f^{(j)}(\lambda_i), \qquad j = 0, 1, \ldots, n_i - 1, \quad i = 1, \ldots s,$$

exist, where $f^{(j)}$ denotes the jth derivative of f, with $f^{(0)} = f$.

Matrix functions: the general case

Let us denote by $\lambda_1, \ldots, \lambda_s$ the distinct eigenvalues of A, and by n_i the order of the largest Jordan block in which the λ_i appears, i.e., the *index* of the eigenvalue λ_i.

Defined on the spectrum

The function f is *defined on the spectrum of* A if the values

$$f^{(j)}(\lambda_i), \qquad j = 0, 1, \ldots, n_i - 1, \quad i = 1, \ldots s,$$

exist, where $f^{(j)}$ denotes the jth derivative of f, with $f^{(0)} = f$.

⚠ Again for the ML function and $\alpha > 0$ we have no problem with this.

Matrix functions: the general case

Matrix function

Lef f be defined on the spectrum of $A \in \mathbb{C}^{n\times n}$, which is represented in Jordan canonical form as $Z^{-1}AZ = J$,

$$f(A) = Zf(J)Z^{-1} = Z\,\mathrm{diag}(f(J_1), \ldots, f(J_p))Z^{-1},$$

where

$$f(J_k) = \begin{bmatrix} f(\lambda_k) & f'(\lambda_k) & \cdots & \frac{f^{(m_k-1)}(\lambda_k)}{(m_k-1)!} \\ & f(\lambda_k) & \ddots & \vdots \\ & & \ddots & f'(\lambda_k) \\ & & & f(\lambda_k) \end{bmatrix}.$$

Moreover, let f be a multivalued function and suppose some eigenvalues occur in more than one Jordan block. If the same choice of branch of f is made in each block, then we say that $f(A)$ is a *primary matrix function*.

Matrix functions: computing $f(A)$ and $f(A)\mathbf{v}$

To march our scheme for

$$\mathbf{u}(t) = E_{\alpha,1}(-t^{\alpha}A)\mathbf{u}_0 + \int_0^t (t-s)^{\alpha-1} E_{\alpha,\alpha}(-A(t-s)^{\alpha})\mathbf{g}(s)\,\mathrm{d}s.$$

we need to compute operations of the form $f(A)\mathbf{v}$, *nevertheless*, we will have to compute $f(\cdot)$ at least on some small matrix.

Matrix functions: computing $f(A)$ and $f(A)\mathbf{v}$

To march our scheme for

$$\mathbf{u}(t) = E_{\alpha,1}(-t^\alpha A)\mathbf{u}_0 + \int_0^t (t-s)^{\alpha-1} E_{\alpha,\alpha}(-A(t-s)^\alpha)\mathbf{g}(s)\,\mathrm{d}s.$$

we need to compute operations of the form $f(A)\mathbf{v}$, *nevertheless*, we will have to compute $f(\cdot)$ at least on some small matrix.

Schur decomposition and matrix functions

Given a matrix A there exist always a matrix Q such that $Q^*Q = I$, and a upper triangular matrix T such that $A = QTQ^*$. Then, if f is defined on the spectrum of A we can compute $f(A)$ as $f(A) = Qf(T)Q^*$.

Matrix functions: computing $f(A)$ and $f(A)\mathbf{v}$

To march our scheme for

$$\mathbf{u}(t) = E_{\alpha,1}(-t^\alpha A)\mathbf{u}_0 + \int_0^t (t-s)^{\alpha-1} E_{\alpha,\alpha}(-A(t-s)^\alpha)\mathbf{g}(s)\,\mathrm{d}s.$$

we need to compute operations of the form $f(A)\mathbf{v}$, *nevertheless*, we will have to compute $f(\cdot)$ at least on some small matrix.

Schur decomposition and matrix functions

Given a matrix A there exist always a matrix Q such that $Q^*Q = I$, and a upper triangular matrix T such that $A = QTQ^*$. Then, if f is defined on the spectrum of A we can compute $f(A)$ as $f(A) = Qf(T)Q^*$.

But how do we compute the matrix function of an upper triangular matrix?

Matrix functions: the upper triangular case

Assumption we assume that T is such that each block $T_{i,j}$ has clustered eigenvalues, and distinct diagonal blocks have *far enough* eigenvalues.

If the **assumption** doesn't hold we look for a block permutation.

$$\left[\begin{array}{cc|cc} (T_{1,1})_{1,1} & (T_{1,1})_{1,2} & \multicolumn{2}{c}{\multirow{2}{*}{$T_{1,2}$}} \\ 0 & (T_{1,1})_{2,2} & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}$}} & (T_{2,2})_{1,1} & (T_{2,2})_{1,2} \\ & & 0 & (T_{2,2})_{2,2} \end{array}\right]$$

Close eigenvalues may lead to severe *accuracy loss*, even far apert eigenvalues can produce more inaccurate answers than expected, see (Davies and Higham 2003).

Matrix functions: the upper triangular case

Assumption we assume that T is such that each block $T_{i,j}$ has clustered eigenvalues, and distinct diagonal blocks have *far enough* eigenvalues.

❗ If the **assumption** doesn't hold we look for a block permutation.

$$\left[\begin{array}{cc|cc} (T_{1,1})_{1,1} & (T_{1,1})_{1,2} & \multicolumn{2}{c}{\multirow{2}{*}{$T_{1,2}$}} \\ 0 & (T_{1,1})_{2,2} & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}$}} & (T_{2,2})_{1,1} & (T_{2,2})_{1,2} \\ & & 0 & (T_{2,2})_{2,2} \end{array}\right]$$

- To evaluate $f(T_{ii})$ we use the Taylor series in σ

$$f(T_{i,i}) = \sum_{k=0}^{+\infty} \frac{f^{(k)}}{k!} M^k,$$

for $\sigma = \mathrm{trace}(T_{i,i})/m$, $m = \dim(T_{i,i})$, and $M = T_{i,i} - \sigma I$.

Matrix functions: the upper triangular case

Assumption we assume that T is such that each block $T_{i,j}$ has clustered eigenvalues, and distinct diagonal blocks have *far enough* eigenvalues.

! If the **assumption** doesn't hold we look for a block permutation.

$$\left[\begin{array}{cc|cc} (T_{1,1})_{1,1} & (T_{1,1})_{1,2} & \multicolumn{2}{c}{\multirow{2}{*}{$T_{1,2}$}} \\ 0 & (T_{1,1})_{2,2} & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}$}} & (T_{2,2})_{1,1} & (T_{2,2})_{1,2} \\ & & 0 & (T_{2,2})_{2,2} \end{array}\right]$$

For the off-diagonal blocks we apply the block-Parlett recurrence

$F_{i,i} = f(T_{i,i}),\ i = 1, \dots, n;$

for $j = 2, \dots, n$ **do**

  for $i = j-1, j-2, \dots, 1$ **do**

    Solve Sylvester equation for $F_{i,j}$:

    $T_{i,i}F_{j,j} - F_{i,j}T_{j,j} = F_{i,i}T_{i,j} - T_{i,j}F_{j,j}$

    $+ \sum_{k=0}^{j-1}(F_{i,k} - T_{k,j} - T_{i,k}F_{k,j}).$

  end

end

Matrix functions: the upper triangular case

Assumption we assume that T is such that each block $T_{i,j}$ has clustered eigenvalues, and distinct diagonal blocks have *far enough* eigenvalues.

! If the **assumption** doesn't hold we look for a block permutation.

$$\left[\begin{array}{cc|cc} (T_{1,1})_{1,1} & (T_{1,1})_{1,2} & \multicolumn{2}{c}{\multirow{2}{*}{$T_{1,2}$}} \\ 0 & (T_{1,1})_{2,2} & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{$\mathbf{0}$}} & (T_{2,2})_{1,1} & (T_{2,2})_{1,2} \\ & & 0 & (T_{2,2})_{2,2} \end{array}\right]$$

What we need

To use the algorithm we have sketched out, we need to be able to compute the derivatives of the ML function sufficiently accurately.

Derivatives of the ML function

The key observation for this task is

$$\frac{d^k}{dz^k} E_{\alpha,\beta}(z) = \sum_{j=0}^{+\infty} \frac{(j+k)_k z^j}{\Gamma(\alpha j + \alpha k + \beta)} = \frac{k!}{\Gamma(k+1)} \sum_{j=0}^{+\infty} \frac{\Gamma(j+k+1) z^j}{j!\Gamma(\alpha j + \alpha k + \beta)} = k! E^{k+1}_{\alpha,\alpha k+\beta}(z),$$

where

$$E^{\gamma}_{\alpha,\beta}(z) = \frac{1}{\Gamma(\gamma)} \sum_{j=0}^{+\infty} \frac{\Gamma(1+\gamma) z^j}{j!\Gamma(\alpha j + \beta)},$$

is called the Prabhakar function.

Derivatives of the ML function

The key observation for this task is

$$\frac{d^k}{dz^k}E_{\alpha,\beta}(z) = \sum_{j=0}^{+\infty} \frac{(j+k)_k z^j}{\Gamma(\alpha j + \alpha k + \beta)} = \frac{k!}{\Gamma(k+1)} \sum_{j=0}^{+\infty} \frac{\Gamma(j+k+1) z^j}{j!\Gamma(\alpha j + \alpha k + \beta)} = k! E^{k+1}_{\alpha,\alpha k+\beta}(z),$$

where

$$E^{\gamma}_{\alpha,\beta}(z) = \frac{1}{\Gamma(\gamma)} \sum_{j=0}^{+\infty} \frac{\Gamma(1+\gamma) z^j}{j!\Gamma(\alpha j + \beta)},$$

is called the Prabhakar function.

Its **efficient computation** can be obtained, similarly to the ML function, by means of a *Laplace transform inversion*

$$\mathcal{L}\left\{ t^{\beta-1} E^{\gamma}_{\alpha,\beta}(t^{\alpha} z) \right\}(s) = \frac{s^{\alpha\gamma-\beta}}{(s^{\alpha} - t^{\alpha} z)^{\gamma}}, \quad \Re(s) > 0, \quad |t^{\alpha} z s^{-\alpha}| < 1.$$

Computing the Prabhakar function (Garrappa 2015)

We select $t = 1$ in

$$\mathcal{L}\left\{t^{\beta-1}E^{\gamma}_{\alpha,\beta}(t^{\alpha}z)\right\}(s) = \frac{s^{\alpha\gamma-\beta}}{(s^{\alpha}-t^{\alpha}z)^{\gamma}}, \quad \Re(s) > 0, \quad |t^{\alpha}zs^{-\alpha}| < 1.$$

Computing the Prabhakar function (Garrappa 2015)

Having selected $t = 1$ we have

$$\mathcal{L}\left\{E^{\gamma}_{\alpha,\beta}(z)\right\}(s) = \frac{s^{\alpha\gamma-\beta}}{(s^{\alpha}-z)^{\gamma}}, \quad \Re(s) > 0, \quad |zs^{-\alpha}| < 1.$$

Computing the Prabhakar function (Garrappa 2015)

Having selected $t = 1$ we have

$$\mathcal{L}\left\{E^{\gamma}_{\alpha,\beta}(z)\right\}(s) = \frac{s^{\alpha\gamma-\beta}}{(s^{\alpha}-z)^{\gamma}}, \quad \Re(s) > 0, \quad |zs^{-\alpha}| < 1, \; H_k(z;z) = \frac{s^{\alpha-\beta}}{(s^{\alpha}-z)^{k+1}}.$$

- Since

$$\frac{d^k}{dz^k}E_{\alpha,\beta}(z) = k!E^{k+1}_{\alpha,\alpha k+\beta}(z) = \frac{k!}{2\pi i}\int_{\mathcal{C}} e^s H_k(s;z)\mathrm{d}s \equiv I_k(z),$$

Computing the Prabhakar function (Garrappa 2015)

Having selected $t = 1$ we have

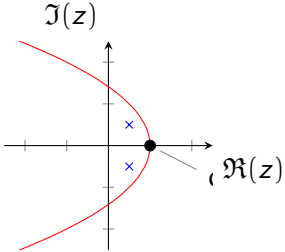
$$\mathcal{L}\left\{E^{\gamma}_{\alpha,\beta}(z)\right\}(s) = \frac{s^{\alpha\gamma-\beta}}{(s^{\alpha}-z)^{\gamma}}, \quad \Re(s) > 0, \quad |zs^{-\alpha}| < 1, \; H_k(z;z) = \frac{s^{\alpha-\beta}}{(s^{\alpha}-z)^{k+1}}.$$

- Since

$$\frac{d^k}{dz^k}E_{\alpha,\beta}(z) = k!E^{k+1}_{\alpha,\alpha k+\beta}(z) = \frac{k!}{2\pi i}\int_{\mathcal{C}} e^s H_k(s;z)\mathrm{d}s \equiv I_k(z),$$

- we use the *Optimal Parabolic Contour* we have already discussed in **Lecture 2** to determine the deformation of the Bromwich line to evaluate

$$I_k^{[N]} = \frac{k!h}{2\pi i}\sum_{j=-N}^{N} e^{\sigma(u_j)}H_k(\sigma(u_j);z)\sigma'(u_j).$$



$s = \mu(iu+1)^2, \; -\infty < u < \infty$

An alternative option (Higham and Liu 2021)

We needed the ML derivatives to apply Schur-Parlett to non-diagonalizable matrices.

An alternative option (Higham and Liu 2021)

We needed the ML derivatives to apply Schur-Parlett to non-diagonalizable matrices.

Diagonalization by perturbation

Let A be nonnormal

$$\tilde{A} = A + E$$

for E a suitable perturbation is *likely to be diagonalizable*. **Diagonalizable matrices are dense in** $\mathbb{C}^{n\times n}$, for a given A and machine precision ϵ then the best approximate diagonalization can be measured in terms of

$$\sigma(A, \epsilon) = \inf_{E,V} \sigma(A, V, E, \epsilon) = \inf_{E,V} \{\kappa_2(V)\epsilon + \|E\|_2\}.$$

An alternative option (Higham and Liu 2021)

We needed the ML derivatives to apply Schur-Parlett to non-diagonalizable matrices.

Diagonalization by perturbation

Let A be nonnormal

$$\tilde{A} = A + E$$

for E a suitable perturbation is *likely to be diagonalizable*. **Diagonalizable matrices are dense in** $\mathbb{C}^{n\times n}$, for a given A and machine precision ϵ then the best approximate diagonalization can be measured in terms of

$$\sigma(A, \epsilon) = \inf_{E,V} \sigma(A, V, E, \epsilon) = \inf_{E,V} \{\kappa_2(V)\epsilon + \|E\|_2\}.$$

We can expect to measure on $f(A)$ by estimating

$$\|f(A+E) - f(A)\| \lesssim \|L_f(A, E)\| \leq \|L_f(A)\|\|E\|,$$

for $L_f(A, E)$ the Fréchet derivative of f at A in direction E, $\|L_f(A)\| = \max_{\|E\|=1} \{\|L_f(A, E)\|\}$.

An alternative option (Higham and Liu 2021)

Fréchet derivative

The **Fréchet derivative** of a matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ at a point $X \in \mathbb{C}^{n\times n}$ is a linear mapping $L : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ $E \mapsto L_f(X, E)$ such that for all $E \in \mathbb{C}^{n\times n}$ we find

$$f(X + E) - f(X) - L(X, E) = o(\|E\|).$$

An alternative option (Higham and Liu 2021)

Fréchet derivative

The **Fréchet derivative** of a matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ at a point $X \in \mathbb{C}^{n\times n}$ is a linear mapping $L : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ $E \mapsto L_f(X, E)$ such that for all $E \in \mathbb{C}^{n\times n}$ we find

$$f(X + E) - f(X) - L(X, E) = o(\|E\|).$$

Thus, in our estimate we have

$$\|f(A + E) - f(A)\| \lesssim \|L_f(A, E)\| \leq \|L_f(A)\|\|E\|,$$

and therefore the change in f induced by E grows as $\|L_f(A)\|_2\|E\|_2$ and there are many cases in which $\|L_f(A)\|_2 \gg 1$.

An alternative option (Higham and Liu 2021)

Fréchet derivative

The **Fréchet derivative** of a matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ at a point $X \in \mathbb{C}^{n\times n}$ is a linear mapping $L : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ $E \mapsto L_f(X, E)$ such that for all $E \in \mathbb{C}^{n\times n}$ we find

$$f(X + E) - f(X) - L(X, E) = o(\|E\|).$$

Thus, in our estimate we have

$$\|f(A + E) - f(A)\| \lesssim \|L_f(A, E)\| \leq \|L_f(A)\|\|E\|,$$

and therefore the change in f induced by E grows as $\|L_f(A)\|_2\|E\|_2$ and there are many cases in which $\|L_f(A)\|_2 \gg 1$.

- The idea from (Higham and Liu 2021) is to use a *structured perturbation*:
  "take E to be upper triangular standard Gaussian matrix."

An alternative option (Higham and Liu 2021)

The idea in few steps

1. Compute the Schur decomposition $A = QTQ^*$,

An alternative option (Higham and Liu 2021)

The idea in few steps

1. Compute the Schur decomposition $A = QTQ^*$,
2. Consider the perturbed matrices $\tilde{T} = T + E$

An alternative option (Higham and Liu 2021)

The idea in few steps

1. Compute the Schur decomposition $A = QTQ^*$,
2. Consider the perturbed matrices $\tilde{T} = T + E$
   - $\tilde{T}$ is still upper triangular,
   - Eigenvectors can be compute by back-substitution: $(\tilde{T} - \tilde{t}_{i,i} I)\mathbf{v}_i = 0,\ i = 1, \ldots, m$,

An alternative option (Higham and Liu 2021)

The idea in few steps

1. Compute the Schur decomposition $A = QTQ^*$,
2. Consider the perturbed matrices $\tilde{T} = T + E$
   - $\tilde{T}$ is still upper triangular,
   - Eigenvectors can be compute by back-substitution: $(\tilde{T} - \tilde{t}_{i,i} I)\mathbf{v}_i = 0,\ i = 1, \ldots, m$,
3. Compute in precision u_h the diagonalization

$$\tilde{T} = VDV^{-1}, \quad D = \mathrm{diag}(\lambda_i),$$

   with **distinct** λ_i,

An alternative option (Higham and Liu 2021)

The idea in few steps

1. Compute the Schur decomposition $A = QTQ^*$,
2. Consider the perturbed matrices $\tilde{T} = T + E$
   - $\tilde{T}$ is still upper triangular,
   - Eigenvectors can be compute by back-substitution: $(\tilde{T} - \tilde{t}_{i,i} I)\mathbf{v}_i = 0,\ i = 1, \ldots, m$,
3. Compute in precision u_h the diagonalization

$$\tilde{T} = VDV^{-1}, \quad D = \mathrm{diag}(\lambda_i),$$

   with **distinct** λ_i,
4. Form $f(\tilde{T}) = Vf(D)V^{-1}$ in precision u_h

An alternative option (Higham and Liu 2021)

The idea in few steps

1. Compute the Schur decomposition $A = QTQ^*$,
2. Consider the perturbed matrices $\tilde{T} = T + E$
   - $\tilde{T}$ is still upper triangular,
   - Eigenvectors can be compute by back-substitution: $(\tilde{T} - \tilde{t}_{i,i} I)\mathbf{v}_i = 0,\ i = 1, \ldots, m$,
3. Compute in precision u_h the diagonalization

$$\tilde{T} = VDV^{-1}, \quad D = \mathrm{diag}(\lambda_i),$$

   with **distinct** λ_i,
4. Form $f(\tilde{T}) = Vf(D)V^{-1}$ in precision u_h

What precision do we need?

To have $\kappa_1(V)u_h \lesssim u$ we select for $c_m u \approx \min_i |\mathrm{diag}(\tilde{t}_{1,1} I - \tilde{T}_{2,2})|$

$$u_h \lesssim \frac{c_m u^2}{\max_{i<j} |\tilde{t}_{i,j}| \left(\max_{i<j} |\tilde{t}_{i,j}| / c_m u + 1\right)^{k-2}}, \quad k = \text{“size of the Jordan block”} \geq 2.$$

From small to large matrices

We now know how to compute $E_{\alpha,\beta}(A)$ for a *small matrix* A, either with

- Classical Schur-Parlett algorithm with Laplace inversion technique for the needed derivative of the ML function (Garrappa and Popolizio 2018),
  - https://it.mathworks.com/matlabcentral/fileexchange/66272-mittag-leffler-function-with-matrix-arguments
- Multiprecision derivative-free Schur-Parlett algorithm (Higham and Liu 2021),
  - https://github.com/Xiaobo-Liu/mp-spalg

From small to large matrices

We now know how to compute $E_{\alpha,\beta}(A)$ for a *small matrix* A, either with

- Classical Schur-Parlett algorithm with Laplace inversion technique for the needed derivative of the ML function (Garrappa and Popolizio 2018),
  - https://it.mathworks.com/matlabcentral/fileexchange/66272-mittag-leffler-function-with-matrix-arguments
- Multiprecision derivative-free Schur-Parlett algorithm (Higham and Liu 2021),
  - https://github.com/Xiaobo-Liu/mp-spalg

What about *large matrices*?

From small to large matrices

We now know how to compute $E_{\alpha,\beta}(A)$ for a *small matrix* A, either with

- Classical Schur-Parlett algorithm with Laplace inversion technique for the needed derivative of the ML function (Garrappa and Popolizio 2018),
  - https://it.mathworks.com/matlabcentral/fileexchange/66272-mittag-leffler-function-with-matrix-arguments
- Multiprecision derivative-free Schur-Parlett algorithm (Higham and Liu 2021),
  - https://github.com/Xiaobo-Liu/mp-spalg

What about *large matrices*?

Projection methods for matrix functions

We can exploit the *subspace projection* idea, take $V \in \mathbb{R}^{n\times k}$ spanning a given subspace $\mathcal{W}_k$

$$f(A)\mathbf{v} \approx Vf(V^TAV)V^T\mathbf{v} \qquad V^TAV \in \mathbb{R}^{k\times k}, \quad k \ll n.$$

Krylov Projection Methods

Different methods are obtained for **different** choices of the **projection spaces** $\mathcal{W}_k(A, \mathbf{v})$.

Krylov Projection Methods

Different methods are obtained for **different** choices of the **projection spaces** $\mathcal{W}_k(A, \mathbf{v})$.

A general framework

Given a set of scalars $\{\sigma_1, \ldots, \sigma_{k-1}\} \subset \overline{\mathbb{C}}$ (the extended complex plane), that are not eigenvalues of A, let

$$q_{k-1}(z) = \prod_{j=1}^{k-1} (\sigma_j - z).$$

The rational Krylov subspace of order k associated with A, $\mathbf{v}$ and q_{k-1} is defined by

$$\mathcal{Q}_k(A, \mathbf{v}) = [q_{k-1}(A)]^{-1} \mathcal{K}_k(A, \mathbf{v}), \qquad \mathcal{K}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A\mathbf{v}, \ldots, A^{k-1}\mathbf{v}\}.$$

Krylov Projection Methods

Different methods are obtained for **different** choices of the **projection spaces** $\mathcal{W}_k(A, \mathbf{v})$.

A general framework

Given a set of scalars $\{\sigma_1, \ldots, \sigma_{k-1}\} \subset \overline{\mathbb{C}}$ (the extended complex plane), that are not eigenvalues of A, let

$$q_{k-1}(z) = \prod_{j=1}^{k-1} (\sigma_j - z).$$

The rational Krylov subspace of order k associated with A, $\mathbf{v}$ and q_{k-1} is defined by

$$\mathcal{Q}_k(A, \mathbf{v}) = [q_{k-1}(A)]^{-1} \mathcal{K}_k(A, \mathbf{v}), \qquad \mathcal{K}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A\mathbf{v}, \ldots, A^{k-1}\mathbf{v}\}.$$

A matrix expression

Given $\{\mu_1, \ldots, \mu_{k-1}\} \subset \overline{\mathbb{C}}$ such that $\sigma_j \neq \mu_j^{-2}$, we define the matrices

$$C_j = (\mu_j \sigma_j A - I)\,(\sigma_j I - A)^{-1}, \text{ and } \mathcal{Q}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, C_1\mathbf{v}, \ldots, C_{k-1} \cdots C_2 C_1 \mathbf{v}\}.$$

Krylov Projection Methods: special cases

A matrix expression

Given $\{\mu_1, \ldots, \mu_{k-1}\} \subset \overline{\mathbb{C}}$ such that $\sigma_j \neq \mu_j^{-2}$, we define the matrices

$$C_j = (\mu_j \sigma_j A - I)\,(\sigma_j I - A)^{-1}, \text{ and } \mathcal{Q}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, C_1\mathbf{v}, \ldots, C_{k-1} \cdots C_2 C_1 \mathbf{v}\}.$$

Polynomial Krylov $\mathcal{W}_k(A, \mathbf{v}) = \mathcal{K}_k(A, \mathbf{v})$ set $\mu_j = 1$ and $\sigma_j = \infty$ for each j,

Krylov Projection Methods: special cases

A matrix expression

Given $\{\mu_1, \ldots, \mu_{k-1}\} \subset \overline{\mathbb{C}}$ such that $\sigma_j \neq \mu_j^{-2}$, we define the matrices

$$C_j = (\mu_j \sigma_j A - I)\,(\sigma_j I - A)^{-1}, \text{ and } \mathcal{Q}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, C_1\mathbf{v}, \ldots, C_{k-1} \cdots C_2 C_1 \mathbf{v}\}.$$

Polynomial Krylov $\mathcal{W}_k(A, \mathbf{v}) = \mathcal{K}_k(A, \mathbf{v})$ set $\mu_j = 1$ and $\sigma_j = \infty$ for each j,

Extended Krylov $\mathcal{W}_{2k-1}(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A^{-1}\mathbf{v}, A\mathbf{v}, \ldots, A^{-(k-1)}\mathbf{v}, A^{k-1}\mathbf{v}\}$, set

$$(\mu_j, \sigma_j) = \begin{cases} (1, \infty), & \text{for } j \text{ even}, \\ (0, 0), & \text{for } j \text{ odd}. \end{cases}$$

Krylov Projection Methods: special cases

A matrix expression

Given $\{\mu_1, \ldots, \mu_{k-1}\} \subset \overline{\mathbb{C}}$ such that $\sigma_j \neq \mu_j^{-2}$, we define the matrices

$$C_j = (\mu_j \sigma_j A - I)\,(\sigma_j I - A)^{-1}, \text{ and } \mathcal{Q}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, C_1\mathbf{v}, \ldots, C_{k-1} \cdots C_2 C_1 \mathbf{v}\}.$$

Polynomial Krylov $\mathcal{W}_k(A, \mathbf{v}) = \mathcal{K}_k(A, \mathbf{v})$ set $\mu_j = 1$ and $\sigma_j = \infty$ for each j,

Extended Krylov $\mathcal{W}_{2k-1}(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A^{-1}\mathbf{v}, A\mathbf{v}, \ldots, A^{-(k-1)}\mathbf{v}, A^{k-1}\mathbf{v}\}$, set

$$(\mu_j, \sigma_j) = \begin{cases} (1, \infty), & \text{for } j \text{ even}, \\ (0, 0), & \text{for } j \text{ odd}. \end{cases}$$

Shift-And-Invert $\mathcal{W}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, (\sigma I - A)^{-1}\mathbf{v}, \ldots, (\sigma I - A)^{-(k-1)}\mathbf{v}\}$, take $\mu_j = 0$ and $\sigma_j = \sigma$ for each j,

The ML function (Moret and Novati 2011)

To estimate the convergence behavior of general projection methods in the non-normal we need the concept of **field of values** (or *numerical range*.)

The ML function (Moret and Novati 2011)

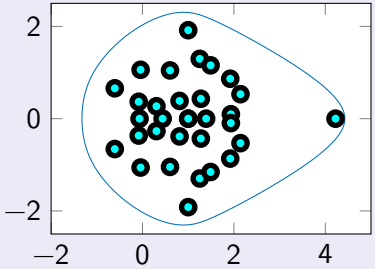
To estimate the convergence behavior of general projection methods in the non-normal we need the concept of **field of values** (or *numerical range*.)

Field of Values/Numerical Range

Given $A \in \mathbb{C}^{N\times N}$ we denote its **field of values** as

$$W(A) = \left\{ \frac{\langle \mathbf{x}, A\mathbf{x}\rangle}{\langle \mathbf{x}, \mathbf{x}\rangle}, \quad \mathbf{0} \neq \mathbf{x} \in \mathbb{C}^N \right\},$$

where $\langle \cdot, \cdot \rangle$ represents the Euclidean inner product.



The ML function (Moret and Novati 2011)

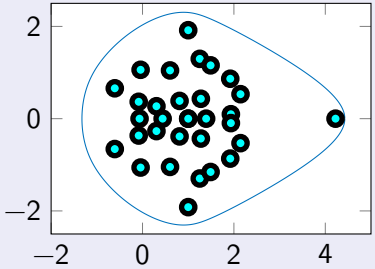
To estimate the convergence behavior of general projection methods in the non-normal we need the concept of **field of values** (or *numerical range*.)

Field of Values/Numerical Range

Given $A \in \mathbb{C}^{N\times N}$ we denote its **field of values** as

$$W(A) = \left\{ \frac{\langle \mathbf{x}, A\mathbf{x}\rangle}{\langle \mathbf{x}, \mathbf{x}\rangle}, \quad \mathbf{0} \neq \mathbf{x} \in \mathbb{C}^N \right\},$$

where $\langle \cdot, \cdot \rangle$ represents the Euclidean inner product.



It has many **properties**, *e.g.*, $W(A) \subseteq D(0, \|A\|)$ (disk centered on 0 with radius $\|A\|$), is *compact*, sub-additive $W(A+B) \subseteq W(A) + W(B)$, unitarily invariant $W(UAU^H) = UW(A)U^H$, *etc.* see (Benzi 2021).

The ML function (Moret and Novati 2011)

Assumptions:

(A1) We assume that $\exists a > 0$, $\theta \in [0, \pi/2)$ such that

$$W(A) \subset \Sigma_{\theta,a} = \{\lambda \in \mathbb{C} \,:\, |\arg(\lambda) - a| \leq \theta\}.$$

(A2) $\beta > 0$, $\alpha \in (0, 2)$ be such that $\alpha\pi/2 < \pi - \theta$, $\varepsilon > 0$ and

$$\frac{\alpha\pi}{2} < \mu \leq \min\{\pi, \alpha\pi\}, \quad \mu < \pi - \theta.$$

Method of choice: we use polynomial Krylov method $\mathcal{K}_m(A, \mathbf{v})$:

$$AV_m = V_m H_m + h_{m+1,m} v_{m+1} \mathbf{e}_m^T, \quad \mathrm{Span}\, V_m = \mathrm{Span}\{\mathbf{v}_i\}_{i=1}^m = \mathcal{K}_m(A, \mathbf{v}), \quad H_m = V_m^H A V_m.$$

We want to bound:

$$R_m = E_{\alpha,\beta}(-A)\mathbf{v} - V_m E_{\alpha,\beta}(-H_m)\mathbf{e}_1, \quad m \geq 1.$$

The ML function (Moret and Novati 2011)

We first express the error in *integral form*, starting from (Podlubny 1999, Theorem 1.1)

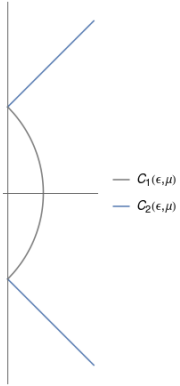
$$E_{\alpha,\beta}(z) = \frac{1}{2\alpha\pi i}\int_{C(\varepsilon,\mu)} \frac{\exp(\lambda^{1/\alpha})\lambda^{1-\beta/\alpha}}{\lambda - z}\,\mathrm{d}\lambda, \quad z \in G^-(\varepsilon,\mu),$$

where

- $\forall\,\varepsilon > 0,\ 0 < \mu < \pi$

$$C(\varepsilon,\mu) = \bigcup \begin{cases} C_1(\varepsilon,\mu) = \{\lambda\,:\,\lambda = \varepsilon\exp(i\varphi), \quad -\mu \le \varphi \le \mu\}, \\ C_2(\varepsilon,\mu) = \{\lambda\,:\,\lambda = r\exp(\pm i\mu), \quad r \ge \varepsilon\}. \end{cases}$$

- The contour $C(\varepsilon,\mu)$ divides the complex plane into two domains, $G^-(\varepsilon,\mu)$ and $G^+(\varepsilon,\mu)$ lying respectively on the left and on the right of $C(\varepsilon,\mu)$.



An Expression for the Error

From the previous we find

$$E_{\alpha,\beta}(-A) = \frac{1}{2\alpha\pi i}\int_{C(\varepsilon,\mu)} \exp(\lambda^{1/\alpha})\lambda^{1-\beta/\alpha}(\lambda I + A)^{-1}\,\mathrm{d}\lambda, \quad \sigma(-A) \in G^{-}(\varepsilon,\mu),$$

and together with

$$R_m = E_{\alpha,\beta}(-A)\mathbf{v} - V_m E_{\alpha,\beta}(-H_m)\mathbf{e}_1, \quad m \geq 1,$$

we write

$$R_m = \frac{1}{2\alpha\pi i}\int_{C(\varepsilon,\mu)} \exp(\lambda^{1/\alpha})\lambda^{1-\beta/\alpha}\delta_m(\lambda), \mathrm{d}\lambda,$$

for

$$\begin{aligned}\delta_m(\lambda) &= (\lambda I + A)^{-1}\mathbf{v} - V_m(\lambda I + H_m)^{-1}\mathbf{e}_1 \\ &= (\lambda I + A)^{-1}\mathbf{v} - V_m(\lambda I + H_m)^{-1}V_m^H\mathbf{v}.\end{aligned}$$

An Expression for the Error

Observe now that

$$\delta_m(\lambda) = (\lambda I + A)^{-1}\mathbf{v} - V_m(\lambda I + H_m)^{-1} V_m^H \mathbf{v} = \Delta_m \mathbf{v},$$

An Expression for the Error

Observe now that

$$\delta_m(\lambda) = (\lambda I + A)^{-1}\mathbf{v} - V_m(\lambda I + H_m)^{-1} V_m^H \mathbf{v} = \Delta_m \mathbf{v},$$

By using the Arnoldi relation, since $\mathbf{v}_{m+1} \perp V_m$:

$$V_m^H(\lambda I + A) V_m = \lambda I + H_m,$$

An Expression for the Error

Observe now that

$$\delta_m(\lambda) = (\lambda I + A)^{-1}\mathbf{v} - V_m(\lambda I + H_m)^{-1} V_m^H \mathbf{v} = \Delta_m \mathbf{v},$$

By using the Arnoldi relation, since $\mathbf{v}_{m+1} \perp V_m$:

$$V_m^H(\lambda I + A) V_m = \lambda I + H_m,$$

Therefore we have

$$\Delta_m(\lambda I + A) V_m = 0.$$

An Expression for the Error

Observe now that

$$\delta_m(\lambda) = (\lambda I + A)^{-1}\mathbf{v} - V_m(\lambda I + H_m)^{-1} V_m^H \mathbf{v} = \Delta_m \mathbf{v},$$

By using the Arnoldi relation, since $\mathbf{v}_{m+1} \perp V_m$:

$$V_m^H(\lambda I + A) V_m = \lambda I + H_m,$$

Therefore we have

$$\Delta_m(\lambda I + A) V_m = 0.$$

For an arbitrary $\mathbf{y} \in \mathbb{C}^m$ we have then

$$(\lambda I + A)^{-1}\mathbf{v} - V_m(\lambda I + H_m)^{-1} V_m^H \mathbf{v} = \Delta_m(\mathbf{v} - (\lambda I + A) V_m \mathbf{y}) = \Delta_m p_m(A)\mathbf{v},$$

where $p_m(z)$ is a polynomial of degree $\leq m$ with $p_m(-\lambda) = 1$.

An Expression for the Error

We have therefore proved that

$$\|\delta_m(A)\| \leq \|(\lambda I + A)^{-1} - V_m(\lambda I + H_m)^{-1} V_m^H\| \|p_m(A)\mathbf{v}\|, \forall p_m \in \mathbb{P}_{\leq m}[z] \text{ with } p_m(-\lambda) = 1.$$

By using (Diele, Moret, and Ragni 2008/09, Lemma 2) we also have the following expression

$$\|\delta_m(\lambda)\| = \frac{\prod_{j=1}^{m} h_{j+1,j}}{|\det(\lambda I + H_m)|} \|(\lambda I + A)^{-1}\mathbf{v}_{m+1}\|.$$

An Expression for the Error

We have therefore proved that

$$\|\delta_m(A)\| \leq \|(\lambda I + A)^{-1} - V_m(\lambda I + H_m)^{-1} V_m^H\| \|p_m(A)\mathbf{v}\|, \forall p_m \in \mathbb{P}_{\leq m}[z] \text{ with } p_m(-\lambda) = 1.$$

By using (Diele, Moret, and Ragni 2008/09, Lemma 2) we also have the following expression

$$\|\delta_m(\lambda)\| = \frac{\prod_{j=1}^{m} h_{j+1,j}}{|\det(\lambda I + H_m)|} \|(\lambda I + A)^{-1}\mathbf{v}_{m+1}\|.$$

To obtain the first bound we call then

$$D(\lambda) = \operatorname{dist}(\lambda, W(-A)) \quad \forall \lambda \in C(\varepsilon, \mu).$$

An Expression for the Error

We have therefore proved that

$$\|\delta_m(A)\| \leq \|(\lambda I + A)^{-1} - V_m(\lambda I + H_m)^{-1} V_m^H\| \|p_m(A)\mathbf{v}\|, \forall p_m \in \mathbb{P}_{\leq m}[z] \text{ with } p_m(-\lambda) = 1.$$

By using (Diele, Moret, and Ragni 2008/09, Lemma 2) we also have the following expression

$$\|\delta_m(\lambda)\| = \frac{\prod_{j=1}^{m} h_{j+1,j}}{|\det(\lambda I + H_m)|} \|(\lambda I + A)^{-1}\mathbf{v}_{m+1}\|.$$

To obtain the first bound we call then

$$D(\lambda) = \operatorname{dist}(\lambda, W(-A)) \quad \forall \lambda \in C(\varepsilon, \mu).$$

Representation function

Using (A1) and (A2) we can find a function $\nu(\varphi)$ such that

$$\forall \lambda = |\lambda| \exp(\pm i\varphi) \in C(\varepsilon, \mu) \quad D(\lambda) \geq \nu(\varphi)|\lambda|, \quad \nu(\varphi) \geq \nu > 0.$$

A First Error Bound

Theorem (Moret and Novati 2011, Theorem 3.2)

Let assumptions (A1) and (A2) hold, then for $m \geq 1$ and for every $M > 0$ we have

$$\|R_m\| \leq \frac{\exp(M) \prod_{j=1}^{m} h_{j+1,j}}{\pi \nu^{m+1} M^{m\alpha+\beta-1}} \left(\frac{\mu}{\alpha} + \frac{\exp(-M(|\cos(\mu/\alpha)| + 1))}{m\alpha - 1 + \beta} \right).$$

A First Error Bound

Theorem (Moret and Novati 2011, Theorem 3.2)

Let assumptions (A1) and (A2) hold, then for $m \geq 1$ and for every $M > 0$ we have

$$\|R_m\| \leq \frac{\exp(M)\prod_{j=1}^{m} h_{j+1,j}}{\pi\nu^{m+1}M^{m\alpha+\beta-1}}\left(\frac{\mu}{\alpha} + \frac{\exp(-M(|\cos(\mu/\alpha)|+1))}{m\alpha-1+\beta}\right).$$

Proof. We use $\|(\lambda I + A)^{-1}\| \leq D(\lambda)^{-1}$ and $W(H_m) \subseteq W(A)$ in the error expression R_m

$$\begin{aligned}\|R_m\| &= \left\|\frac{1}{2\alpha\pi i}\int_{C(\varepsilon,\mu)} \exp(\lambda^{1/\alpha})\lambda^{1-\beta/\alpha}\delta_m(\lambda), \mathrm{d}\lambda\right\| \\ &\leq \frac{\prod_{j=1}^{m} h_{j+1,j}}{2\pi\alpha}\int_{C(\varepsilon,\mu)} \frac{\left|\exp(\lambda^{1/\alpha})\lambda^{1-\beta/\alpha}\right|}{D(\lambda)^{m+1}}\,|\mathrm{d}\lambda|.\end{aligned}$$

A First Error Bound

Theorem (Moret and Novati 2011, Theorem 3.2)

Let assumptions (A1) and (A2) hold, then for $m \geq 1$ and for every $M > 0$ we have

$$\|R_m\| \leq \frac{\exp(M)\prod_{j=1}^{m} h_{j+1,j}}{\pi\nu^{m+1}M^{m\alpha+\beta-1}}\left(\frac{\mu}{\alpha} + \frac{\exp(-M(|\cos(\mu/\alpha)|+1))}{m\alpha-1+\beta}\right).$$

Proof. We use $\|(\lambda I + A)^{-1}\| \leq D(\lambda)^{-1}$ and $W(H_m) \subseteq W(A)$ in the error expression R_m

$$\|R_m\| \leq \frac{\prod_{j=1}^{m} h_{j+1,j}}{2\pi\alpha}(I_1 + I_2),$$

with

$$I_1 = \int_{C_1(\varepsilon,\mu)} \frac{\left|\exp(\lambda^{1/\alpha})\lambda^{1-\beta/\alpha}\right|}{D(\lambda)^{m+1}}|d\lambda| \leq 2\varepsilon^{\frac{1-\beta}{\alpha}-m}\int_0^{\mu}\frac{\exp(\varepsilon^{1/\alpha}\cos(\varphi/\alpha))}{\nu(\varphi)^{m+1}}\,d\varphi,$$

A First Error Bound

Theorem (Moret and Novati 2011, Theorem 3.2)

Let assumptions (A1) and (A2) hold, then for $m \geq 1$ and for every $M > 0$ we have

$$\|R_m\| \leq \frac{\exp(M)\prod_{j=1}^{m} h_{j+1,j}}{\pi\nu^{m+1}M^{m\alpha+\beta-1}}\left(\frac{\mu}{\alpha} + \frac{\exp(-M(|\cos(\mu/\alpha)|+1))}{m\alpha-1+\beta}\right).$$

Proof. We use $\|(\lambda I + A)^{-1}\| \leq D(\lambda)^{-1}$ and $W(H_m) \subseteq W(A)$ in the error expression R_m

$$\|R_m\| \leq \frac{\prod_{j=1}^{m} h_{j+1,j}}{2\pi\alpha}\left(2\varepsilon^{\frac{1-\beta}{\alpha}-m}\int_0^{\mu}\frac{\exp(\varepsilon^{1/\alpha}\cos(\varphi/\alpha))}{\nu(\varphi)^{m+1}}\,\mathrm{d}\varphi + I_2\right),$$

with

$$I_2 = \int_{C_2(\varepsilon,\mu)}\frac{\left|\exp(\lambda^{1/\alpha})\lambda^{1-\beta/\alpha}\right|}{D(\lambda)^{m+1}}|\mathrm{d}\lambda| \leq \frac{2}{\nu^{m+1}}\int_{\varepsilon}^{+\infty}\frac{r^{\frac{1-\beta}{\alpha}}\exp(-r^{\frac{1}{\alpha}}|\cos(\mu/\alpha)|)}{r^{m+1}}\,\mathrm{d}r$$

$$= \frac{2}{\nu^{m+1}}\int_{\varepsilon^{1/\alpha}}^{+\infty}\frac{\exp(-s|\cos(\mu/\alpha)|)}{s^{m\alpha+\beta}}\,\mathrm{d}s \leq \frac{2\alpha\exp(-\varepsilon^{1/\alpha}|\cos(\mu/\alpha)|)}{(m\alpha+\beta-1)\nu^{m+1}\varepsilon^{\frac{m\alpha+\beta-1}{\alpha}}}.$$

A First Error Bound

Theorem (Moret and Novati 2011, Theorem 3.2)

Let assumptions (A1) and (A2) hold, then for $m \geq 1$ and for every $M > 0$ we have

$$\|R_m\| \leq \frac{\exp(M)\prod_{j=1}^{m} h_{j+1,j}}{\pi\nu^{m+1}M^{m\alpha+\beta-1}}\left(\frac{\mu}{\alpha} + \frac{\exp(-M(|\cos(\mu/\alpha)|+1))}{m\alpha-1+\beta}\right).$$

Proof. We use $\|(\lambda I + A)^{-1}\| \leq D(\lambda)^{-1}$ and $W(H_m) \subseteq W(A)$ in the error expression R_m

$$\|R_m\| \leq \frac{\prod_{j=1}^{m} h_{j+1,j}}{2\pi\alpha}\left(2\varepsilon^{\frac{1-\beta}{\alpha}-m}\int_0^{\mu}\frac{\exp(\varepsilon^{1/\alpha}\cos(\varphi/\alpha))}{\nu(\varphi)^{m+1}}\,\mathrm{d}\varphi + \frac{2\alpha\exp(-\varepsilon^{1/\alpha}|\cos(\mu/\alpha)|)}{(m\alpha+\beta-1)\nu^{m+1}\varepsilon^{\frac{m\alpha+\beta-1}{\alpha}}}\right)$$

The result follows then by setting $\varepsilon = M^{\alpha}$ and simplifying the expression. $\square$

A First Error Bound

Theorem (Moret and Novati 2011, Theorem 3.2)

Let assumptions (A1) and (A2) hold, then for $m \geq 1$ and for every $M > 0$ we have

$$\|R_m\| \leq \frac{\exp(M)\prod_{j=1}^{m} h_{j+1,j}}{\pi\nu^{m+1}M^{m\alpha+\beta-1}}\left(\frac{\mu}{\alpha} + \frac{\exp(-M(|\cos(\mu/\alpha)|+1))}{m\alpha-1+\beta}\right).$$

Proof. We use $\|(\lambda I + A)^{-1}\| \leq D(\lambda)^{-1}$ and $W(H_m) \subseteq W(A)$ in the error expression R_m

$$\|R_m\| \leq \frac{\prod_{j=1}^{m} h_{j+1,j}}{2\pi\alpha}\left(2\varepsilon^{\frac{1-\beta}{\alpha}-m}\int_0^{\mu}\frac{\exp(\varepsilon^{1/\alpha}\cos(\varphi/\alpha))}{\nu(\varphi)^{m+1}}\,\mathrm{d}\varphi + \frac{2\alpha\exp(-\varepsilon^{1/\alpha}|\cos(\mu/\alpha)|)}{(m\alpha+\beta-1)\nu^{m+1}\varepsilon^{\frac{m\alpha+\beta-1}{\alpha}}}\right)$$

The result follows then by setting $\varepsilon = M^{\alpha}$ and simplifying the expression. □

⚠ With the same proof another bound for the case of small α can be obtained.

A First Error Bound: small αs

Theorem (Moret and Novati 2011, Theorem 3.2)

Let assumptions (A1) and (A2) hold, then for $m \geq 1$ and for every $M > 0$ we have

$$\|R_m\| \leq \frac{\exp(M) \prod_{j=1}^{m} h_{j+1,j}}{\pi \nu^{m+1} M^{m\alpha+\beta-1}} \left(\frac{\mu}{\alpha} + \frac{\exp(-M(|\cos(\mu/\alpha)| + 1))}{m\alpha - 1 + \beta} \right).$$

Corollary (Moret and Novati 2011, Corollary 3.3)

Let assumptions (A1) and (A2) hold. Let $m \geq 1$ be such that $m\alpha + \beta > 0$, then for every $M > 0$, we have

$$\|R_m\| \leq \frac{\exp(M) \prod_{j=1}^{m} h_{j+1,j}}{4\nu^{m+1} M^{m\alpha}} \frac{4M^{1-\beta}}{\pi} \left(\frac{\mu}{\alpha} + \frac{\exp(-M(1 + |\cos(\mu/\alpha)|))}{M|\cos(\mu/\alpha)|} \right).$$

A First Error Bound: some observations

- The ML function is entire for $\alpha > 0 \Rightarrow$ superlinear convergence for large enough m:

$$M = m\alpha + \beta - 1 \;\Rightarrow\; \|R_m\| \propto \left(\frac{\exp(1)}{M}\right)^M \nu^{-(m+1)} \prod_{j=1}^{m} h_{j+1,j}.$$

A First Error Bound: some observations

- The ML function is entire for $\alpha > 0 \Rightarrow$ superlinear convergence for large enough m:

$$M = m\alpha + \beta - 1 \;\Rightarrow\; \|R_m\| \propto \left(\frac{\exp(1)}{M}\right)^M \nu^{-(m+1)} \prod_{j=1}^{m} h_{j+1,j}.$$

- To better understand this, we use that for every monic polynomial of degree m we find

$$\prod_{j=1}^{m} h_{j+1,j} \leq \|q_m(A)v\|,$$

Therefore, if we take q_m as the **monic Faber polynomial** associated to a closed convex subset $\Omega \supset W(-A)$ we get the bound in terms of the **logarithmic capacity** γ of Ω.

A First Error Bound: some observations

- The ML function is entire for $\alpha > 0 \Rightarrow$ superlinear convergence for large enough m:

$$M = m\alpha + \beta - 1 \;\Rightarrow\; \|R_m\| \propto \left(\frac{\exp(1)}{M}\right)^M \nu^{-(m+1)} \prod_{j=1}^{m} h_{j+1,j}.$$

- To better understand this, we use that for every monic polynomial of degree m we find

$$\prod_{j=1}^{m} h_{j+1,j} \leq 2\gamma^m,$$

  Therefore, if we take q_m as the **monic Faber polynomial** associated to a closed convex subset $\Omega \supset W(-A)$ we get the bound in terms of the **logarithmic capacity** γ of Ω.

$\Rightarrow$ we have discovered:

$$\|R_m\| \propto \left(\frac{\exp(1)}{m\alpha}\right)^{m\alpha} \left(\frac{\gamma}{\nu}\right)^m.$$

Specialized bounds

The bound can be refined under stricter hypotheses.

Specialized bounds

The bound can be refined under stricter hypotheses.

Theorem (Moret and Novati 2011, Theorem 3.5)

Assume that A is Hermitian with $\sigma(A) \subseteq [a, b] \subset [0, +\infty)$. Assume that $0 < \alpha < 1$, $\beta \geq \alpha$. Let $\mu \leq \pi/2$, $\alpha\pi/2 < \mu < \alpha\pi$. Then for every index $m \geq 1$ and for every $M > 0$ we have

$$\|R_m\| \leq \frac{4M^{1-\beta}}{\pi}\left(\frac{\mu}{\alpha} + \frac{\exp(-M(1+|\cos(\mu/\alpha)|))}{M|\cos(\mu/\alpha)|}\right)\exp(M)\Phi(u(M^\alpha \exp(i\mu)))^{-m}.$$

for $\Phi(u) = u + \sqrt{u^2 - 1}$, $u(z) = {}^{(|b+z|+|a+z|)}/_{b-a}$.

Specialized bounds

The bound can be refined under stricter hypotheses.

Theorem (Moret and Novati 2011, Theorem 3.5)

Assume that A is Hermitian with $\sigma(A) \subseteq [a, b] \subset [0, +\infty)$. Assume that $0 < \alpha < 1$, $\beta \geq \alpha$. Let $\mu \leq \pi/2$, $\alpha\pi/2 < \mu < \alpha\pi$. Then for every index $m \geq 1$ and for every $M > 0$ we have

$$\|R_m\| \leq \frac{4M^{1-\beta}}{\pi}\left(\frac{\mu}{\alpha} + \frac{\exp(-M(1+|\cos(\mu/\alpha)|))}{M|\cos(\mu/\alpha)|}\right)\exp(M)\Phi(u(M^\alpha \exp(i\mu)))^{-m}.$$

for $\Phi(u) = u + \sqrt{u^2 - 1}$, $u(z) = {}^{(|b+z|+|a+z|)}/_{b-a}$.

Limiting relation

If $\alpha \to 0$, $\beta = 1$, we have $E_{0,1}(-z) = (1+z)^{-1}$, $|z| < 1$. Then setting $\mu = \alpha\pi$ and letting $M = 1$, we find

$$\|R_m\| \leq \frac{4(\pi\exp(1) - \exp(-1))}{\pi\Phi(u(1))^m}$$

The Shift-and-Invert Method (Moret and Novati 2011)

We remain under the assumptions (A1) and (A2) and consider the matrix

$$Z = (I + hA)^{-1}, \qquad h > 0,$$

together with the space $\mathcal{K}_m(Z, \mathbf{v})$.

The Shift-and-Invert Method (Moret and Novati 2011)

We remain under the assumptions (A1) and (A2) and consider the matrix

$$Z = (I + hA)^{-1}, \qquad h > 0,$$

together with the space $\mathcal{K}_m(Z, \mathbf{v})$.

We can write the **analogous Arnoldi relation** for $U_m = [\mathbf{u}_1, \ldots, \mathbf{u}_m]$ spanning $\mathcal{K}_m(Z, \mathbf{v})$:

$$ZU_m = U_m S_m + s_{m+1,m} u_{m+1} \mathbf{e}_m^T, \qquad S_m = U_m^H Z U_m.$$

The Shift-and-Invert Method (Moret and Novati 2011)

We remain under the assumptions (A1) and (A2) and consider the matrix

$$Z = (I + hA)^{-1}, \qquad h > 0,$$

together with the space $\mathcal{K}_m(Z, \mathbf{v})$.
We can write the **analogous Arnoldi relation** for $U_m = [\mathbf{u}_1, \ldots, \mathbf{u}_m]$ spanning $\mathcal{K}_m(Z, \mathbf{v})$:

$$ZU_m = U_m S_m + s_{m+1,m} u_{m+1} \mathbf{e}_m^T, \qquad S_m = U_m^H Z U_m.$$

The **approximation** is then given by

$$\mathbf{y} = f(A)\mathbf{v} \approx \mathbf{y}_m = V_m f(B_m)\mathbf{e}_1 \text{ where } (I + hB_m)S_m = I.$$

The Shift-and-Invert Method (Moret and Novati 2011)

We remain under the assumptions (A1) and (A2) and consider the matrix

$$Z = (I + hA)^{-1}, \qquad h > 0,$$

together with the space $\mathcal{K}_m(Z, \mathbf{v})$.
We can write the **analogous Arnoldi relation** for $U_m = [\mathbf{u}_1, \ldots, \mathbf{u}_m]$ spanning $\mathcal{K}_m(Z, \mathbf{v})$:

$$ZU_m = U_m S_m + s_{m+1,m} u_{m+1} \mathbf{e}_m^T, \qquad S_m = U_m^H Z U_m.$$

The **approximation** is then given by

$$\mathbf{y} = f(A)\mathbf{v} \approx \mathbf{y}_m = V_m f(B_m)\mathbf{e}_1 \text{ where } (I + hB_m)S_m = I.$$

We can repeat the general error analysis using

$$R_m = E_{\alpha,\beta}(-A)\mathbf{v} - U_m E_{\alpha,\beta}(-B_m)\mathbf{e}_1 = \frac{1}{2\pi\alpha i}\int_{C(\varepsilon,\mu)} \exp(\lambda^{1/\alpha})\lambda^{(1-\beta)/\alpha} b_m(\lambda)\,\mathrm{d}\lambda,$$

for $b_m(\lambda) = (\lambda I + A)^{-1}\mathbf{v} - U_m(\lambda I + B_m)^{-1}\mathbf{e}_1$.

Error bound (Moret and Novati 2011)

Theorem (Moret and Novati 2011, Theorem 4.3)

For every matrix A satisfying (A1) and (A2), assume $0 < \alpha < 1$ and $\beta \geq \alpha$. Then, there exists a function $g(h)$, continuous in any bounded interval $0 < h_1 \leq h \leq h_2$, such that for $m \geq 2$,

$$\|R_m\| \leq \frac{g(h)}{m-1}.$$

Error bound (Moret and Novati 2011)

Theorem (Moret and Novati 2011, Theorem 4.3)

For every matrix A satisfying (A1) and (A2), assume $0 < \alpha < 1$ and $\beta \geq \alpha$. Then, there exists a function $g(h)$, continuous in any bounded interval $0 < h_1 \leq h \leq h_2$, such that for $m \geq 2$,

$$\|R_m\| \leq \frac{g(h)}{m-1}.$$

Theorem (Moret and Novati 2011, Theorem 4.5)

Assume that A is Hermitian with $\sigma(A) \subseteq [a, +\infty)$, $a \geq 0$. Assume $0 < \alpha \leq 2/3$ and $\beta \geq \alpha$. Then, for every $m \geq 1$ we have

$$\|R_m\| \leq \frac{K_1 Q_m h^{\frac{\beta-1}{\alpha}}}{(1+\sqrt{2})^{m-1}} + \frac{K_2 h^{\beta/\alpha}}{(m-1)^2} \exp\left(-\frac{h^{-1/\alpha}}{\sqrt{2}}\right),$$

where $Q_m = \max_{0 \leq |\varphi| \leq 3\alpha\pi/4} \exp\left(h^{-1/\alpha} \cos \varphi/\alpha\right) (1 - \cos \varphi)^{\frac{m-1}{2}}$, with K_1, K_2 constants.

ML function, what have we found?

- The *polynomial method* suffers both for **small** α **values** and for **large field of values**.

ML function, what have we found?

- The *polynomial method* suffers both for **small** α **values** and for **large field of values**.
- For the *shift-and-invert* method the convergence doesn't deteriorate with the size of $W(A)$, its uniform with respect to the h parameter.

ML function, what have we found?

- The *polynomial method* suffers both for **small α values** and for **large field of values**.
- For the *shift-and-invert* method the convergence doesn't deteriorate with the size of $W(A)$, its uniform with respect to the h parameter.
- To obtain a complete method one still has to find a way to repeatedly compute the matrix functions in

$$\mathbf{u}(t) = E_{\alpha,1}(-t^{\alpha}A)\mathbf{u}_0 + \int_0^t (t-s)^{\alpha-1} E_{\alpha,\alpha}(-A(t-s)^{\alpha})\mathbf{g}(s)\,\mathrm{d}s.$$

ML function, what have we found?

- The *polynomial method* suffers both for **small α values** and for **large field of values**.
- For the *shift-and-invert* method the convergence doesn't deteriorate with the size of $W(A)$, its uniform with respect to the h parameter.
- To obtain a complete method one still has to find a way to repeatedly compute the matrix functions in

$$\mathbf{u}(t) = E_{\alpha,1}(-t^{\alpha}A)\mathbf{u}_0 + \int_0^t (t-s)^{\alpha-1} E_{\alpha,\alpha}(-A(t-s)^{\alpha})\mathbf{g}(s)\,\mathrm{d}s.$$

- **Research ideas:** finding better rational approximations/poles/expansions together with error analysis for the ML function.

ML function, what have we found?

- The *polynomial method* suffers both for **small α values** and for **large field of values**.
- For the *shift-and-invert* method the convergence doesn't deteriorate with the size of $W(A)$, its uniform with respect to the h parameter.
- To obtain a complete method one still has to find a way to repeatedly compute the matrix functions in

$$\mathbf{u}(t) = E_{\alpha,1}(-t^\alpha A)\mathbf{u}_0 + \int_0^t (t-s)^{\alpha-1} E_{\alpha,\alpha}(-A(t-s)^\alpha)\mathbf{g}(s)\,\mathrm{d}s.$$

- **Research ideas:** finding better rational approximations/poles/expansions together with error analysis for the ML function.

Other extensions

A variant with *restart* is discussed in (Moret and Popolizio 2014), the combination with other matrix-functions in (Moret and Novati 2019).

Bibliography I

- Benzi, M. (2021). "Some uses of the field of values in numerical analysis". In: *Boll. Unione Mat. Ital.* 14.1, pp. 159–177. ISSN: 1972-6724. DOI: 10.1007/s40574-020-00249-2. URL: https://doi.org/10.1007/s40574-020-00249-2.
- Davies, P. I. and N. J. Higham (2003). "A Schur-Parlett algorithm for computing matrix functions". In: *SIAM J. Matrix Anal. Appl.* 25.2, pp. 464–485. ISSN: 0895-4798. DOI: 10.1137/S0895479802410815. URL: https://doi.org/10.1137/S0895479802410815.
- Diele, F., I. Moret, and S. Ragni (2008/09). "Error estimates for polynomial Krylov approximations to matrix functions". In: *SIAM J. Matrix Anal. Appl.* 30.4, pp. 1546–1565. ISSN: 0895-4798. DOI: 10.1137/070688924. URL: https://doi.org/10.1137/070688924.
- Garrappa, R. (2015). "Numerical evaluation of two and three parameter Mittag-Leffler functions". In: *SIAM J. Numer. Anal.* 53.3, pp. 1350–1369. ISSN: 0036-1429. DOI: 10.1137/140971191. URL: https://doi.org/10.1137/140971191.

Bibliography II

- Garrappa, R. and M. Popolizio (2011). "On accurate product integration rules for linear fractional differential equations". In: *J. Comput. Appl. Math.* 235.5, pp. 1085–1097. ISSN: 0377-0427. DOI: 10.1016/j.cam.2010.07.008. URL: https://doi.org/10.1016/j.cam.2010.07.008.
- — (2018). "Computing the matrix Mittag-Leffler function with applications to fractional calculus". In: *J. Sci. Comput.* 77.1, pp. 129–153. ISSN: 0885-7474. DOI: 10.1007/s10915-018-0699-5. URL: https://doi.org/10.1007/s10915-018-0699-5.
- Higham, N. J. and X. Liu (2021). "A multiprecision derivative-free Schur-Parlett algorithm for computing matrix functions". In: *SIAM J. Matrix Anal. Appl.* 42.3, pp. 1401–1422. ISSN: 0895-4798. DOI: 10.1137/20M1365326. URL: https://doi.org/10.1137/20M1365326.
- Metzler, R. and J. Klafter (2000). "The random walk's guide to anomalous diffusion: a fractional dynamics approach". In: *Phys. Rep.* 339.1, p. 77. ISSN: 0370-1573. DOI: 10.1016/S0370-1573(00)00070-3. URL: https://doi.org/10.1016/S0370-1573(00)00070-3.

Bibliography III

- Moret, I. and P. Novati (2011). "On the convergence of Krylov subspace methods for matrix Mittag-Leffler functions". In: *SIAM J. Numer. Anal.* 49.5, pp. 2144–2164. ISSN: 0036-1429. DOI: 10.1137/080738374. URL: https://doi.org/10.1137/080738374.
- — (2019). "Krylov subspace methods for functions of fractional differential operators". In: *Math. Comp.* 88.315, pp. 293–312. ISSN: 0025-5718. DOI: 10.1090/mcom/3332. URL: https://doi.org/10.1090/mcom/3332.
- Moret, I. and M. Popolizio (2014). "The restarted shift-and-invert Krylov method for matrix functions". In: *Numer. Linear Algebra Appl.* 21.1, pp. 68–80. ISSN: 1070-5325. DOI: 10.1002/nla.1862. URL: https://doi.org/10.1002/nla.1862.
- Podlubny, I. (1999). *Fractional differential equations*. Vol. 198. Mathematics in Science and Engineering. An introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications. Academic Press, Inc., San Diego, CA, pp. xxiv+340. ISBN: 0-12-558840-2.

Bibliography IV

Sokolov, I. M. and J. Klafter (2005). "From diffusion to anomalous diffusion: a century after Einstein's Brownian motion". In: *Chaos* 15.2, pp. 026103, 7. ISSN: 1054-1500. DOI: 10.1063/1.1860472. URL: https://doi.org/10.1063/1.1860472.

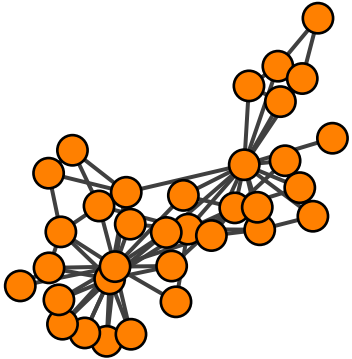
An introduction to fractional calculus

Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

June, 2022

Questions in Complex Networks

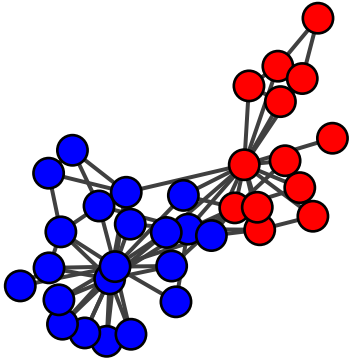


A complex network is a graph with **non-trivial** topological features, neither a structured graph (lattices, Cayley graphs, *etc.*) nor a *completely* random graph.

We are interested in tasks in **exploratory data analysis**, that is analyzing the data to summarize their main characteristics:

- Divide the nodes into groups that are in the same community (clustering),
- Find the "most relevant" nodes in the network (centrality),
- Find the "most relevant" edge in the network (edge centrality)
- Individuation of motifs, computation of fluxes, maximum cuts, *etc.*

Questions in Complex Networks

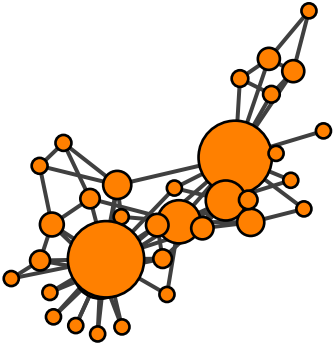


A complex network is a graph with **non-trivial** topological features, neither a structured graph (lattices, Cayley graphs, *etc.*) nor a *completely* random graph.

We are interested in tasks in **exploratory data analysis**, that is analyzing the data to summarize their main characteristics:

- Divide the nodes into groups that are in the same community (clustering),
- Find the "most relevant" nodes in the network (centrality),
- Find the "most relevant" edge in the network (edge centrality)
- Individuation of motifs, computation of fluxes, maximum cuts, *etc.*

Questions in Complex Networks

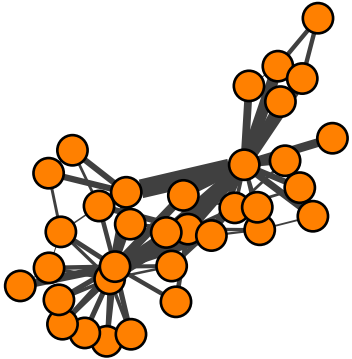


A complex network is a graph with **non-trivial** topological features, neither a structured graph (lattices, Cayley graphs, *etc.*) nor a *completely* random graph.

We are interested in tasks in **exploratory data analysis**, that is analyzing the data to summarize their main characteristics:

- Divide the nodes into groups that are in the same community (clustering),
- Find the "most relevant" nodes in the network (centrality),
- Find the "most relevant" edge in the network (edge centrality)
- Individuation of motifs, computation of fluxes, maximum cuts, *etc.*

Questions in Complex Networks



A complex network is a graph with **non-trivial** topological features, neither a structured graph (lattices, Cayley graphs, *etc.*) nor a *completely* random graph.

We are interested in tasks in **exploratory data analysis**, that is analyzing the data to summarize their main characteristics:

- Divide the nodes into groups that are in the same community (clustering),
- Find the "most relevant" nodes in the network (centrality),
- Find the "most relevant" edge in the network (edge centrality)
- Individuation of motifs, computation of fluxes, maximum cuts, *etc.*

Notation

Network

A *network* $G = (V, E)$ is defined as a pair of sets: a set $V = \{1, 2, \ldots, n\}$ of *nodes* and a set $E \subset V \times V$ of *edges* between them.

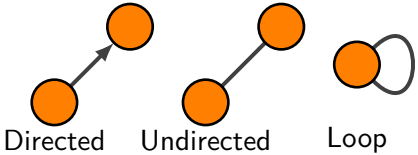
Notation

Network

A *network* $G = (V, E)$ is defined as a pair of sets: a set $V = \{1, 2, \ldots, n\}$ of *nodes* and a set $E \subset V \times V$ of *edges* between them.

Directed/Undirected

If $\forall (i, j) \in E$ then $(j, i) \in E$ the network is said to be *undirected* is *directed* otherwise.

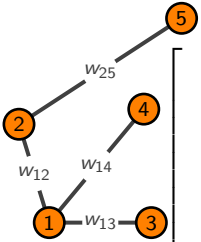


An edge from a node to itself is called a *loop*.

Notation

Network

A *network* $G = (V, E)$ is defined as a pair of sets: a set $V = \{1, 2, \ldots, n\}$ of *nodes* and a set $E \subset V \times V$ of *edges* between them.



$$\begin{bmatrix} 0 & w_{12} & w_{13} & w_{14} & 0 \\ w_{12} & 0 & 0 & 0 & w_{25} \\ w_{13} & 0 & 0 & 0 & 0 \\ w_{14} & 0 & 0 & 0 & 0 \\ 0 & w_{25} & 0 & 0 & 0 \end{bmatrix}$$

Adjacency Matrix

We represent a Network via its *adjacency matrix* $A = (a_{ij}) \in \mathbb{R}^{n \times n}$, entrywise defined as

$$a_{ij} = \begin{cases} w_{ij} & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

where $w_{ij} > 0$ is the weight of edge (i, j).

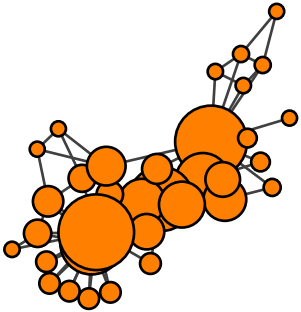
Centrality Measures: the limiting cases

- **Degree centrality:**

$$d_i = \sum_{j=1}^{n} a_{ij} = (A\mathbf{1})_i$$

- **Eigenvector centrality:** $\rho(A) > 0$ the spectral radius of the irreducible $A \geq 0$

$$x_i = \frac{1}{\rho(A)} \sum_{j=1}^{n} a_{ij} x_j$$



Degree centrality is oblivious to the whole topology of the network.

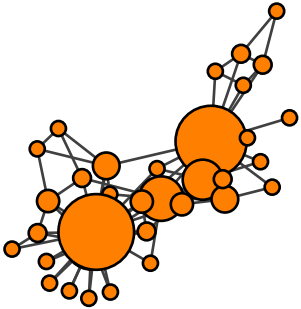
Centrality Measures: the limiting cases

- **Degree centrality:**

$$d_i = \sum_{j=1}^{n} a_{ij} = (A\mathbf{1})_i$$

- **Eigenvector centrality:** $\rho(A) > 0$ the spectral radius of the irreducible $A \geq 0$

$$x_i = \frac{1}{\rho(A)} \sum_{j=1}^{n} a_{ij} x_j$$



Eigenvector centrality considers both the number of neighbors and their importance when assigning scores to nodes.

Walk based centralities and Matrix Functions

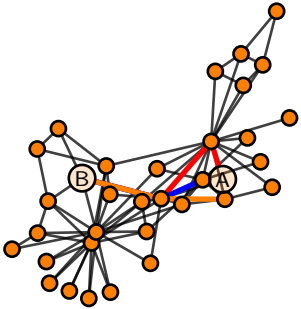
Consider the **analytic function** f in $\{z \in \mathbb{C} : |z| < R_f\}$:

$$f(z) = \sum_{r=0}^{\infty} c_r z^r, \qquad c_r \geq 0$$

then under suitable hypothesis on the spectrum of A we can write:

$$f(A) = \sum_{r=0}^{\infty} c_r A^r.$$

$(A^r)_{i_1,i_{r+1}}$ is the number of walks from i_1 to i_{r+1}.



A *walk* of length r is a sequence of $r+1$ nodes $i_1, i_2, \ldots, i_{r+1}$ such that $(i_\ell, i_{\ell+1}) \in E$ for all $\ell = 1, \ldots, r$.

Walk based centralities and Matrix Functions

Consider the **analytic function** f in $\{z \in \mathbb{C} : |z| < R_f\}$:

$$f(z) = \sum_{r=0}^{\infty} c_r z^r, \qquad c_r \geq 0$$

then under suitable hypothesis on the spectrum of A we can write:

$$f(A) = \sum_{r=0}^{\infty} c_r A^r.$$

$(A^r)_{i_1,i_{r+1}}$ is the number of walks from i_1 to i_{r+1}.

- $(f(A))_{ij}$ is a **weighted sum** of the number of **all walks** of any length that start from node i and end at node j,
- $c_r \to 0$ as r increases thus **walks of longer lengths** are considered to be **less important**,
- The most popular functions used in networks science are $f(z) = e^z$ and $f(z) = (1+z)^{-1}$.

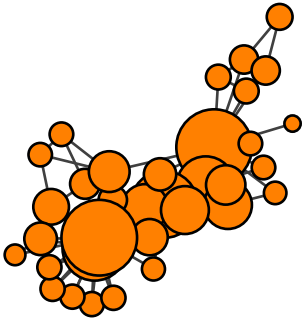
Walk based centralities

- **Subgraph centrality:**

$$s_i(f) = \mathbf{e}_i^T f(A)\mathbf{e}_i = \sum_{r=0}^{\infty} c_r (A^r)_{ii}.$$

- **Total** (node) **communicability:**

$$t_i(f) = \sum_{j=1}^{n} (f(A))_{ij} = \sum_{j=1}^{n} \sum_{r=0}^{\infty} c_r (A^r)_{ij}$$



Subgraph centrality accounts for the returnability of information from a node to itself: it is a weighted count of all the subgraphs node i is involved in.

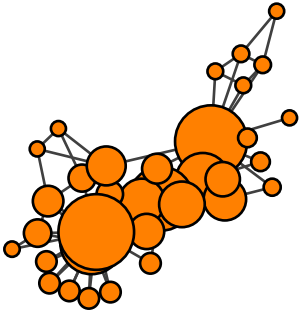
Walk based centralities

- **Subgraph centrality:**

$$s_i(f) = \mathbf{e}_i^T f(A)\mathbf{e}_i = \sum_{r=0}^{\infty} c_r (A^r)_{ii}.$$

- **Total** (node) **communicability:**

$$t_i(f) = \sum_{j=1}^{n} (f(A))_{ij} = \sum_{j=1}^{n} \sum_{r=0}^{\infty} c_r (A^r)_{ij}$$



For the **total comunicability** the importance of a node depends on how well it communicates with the whole network, itself included

The Mittag-Leffler Function

The *Mittag–Leffler (ML) function* is an analytic functions given, $\forall \alpha, \beta > 0$, by

$$E_{\alpha,\beta}(z) = \sum_{r=0}^{\infty} c_r(\alpha,\beta) z^r = \sum_{r=0}^{\infty} \frac{z^r}{\Gamma(\alpha r + \beta)},$$

where

- $c_r(\alpha,\beta) = \Gamma(\alpha r + \beta)^{-1}$,
- $\Gamma(z)$ is the *Euler Gamma function*:

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt.$$

For particular choices of $\alpha, \beta > 0$, the ML function $E_{\alpha,\beta}(z)$ has a nice closed form descriptions.

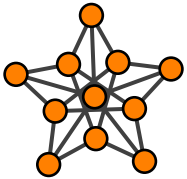
α	β	Function
0	1	$(1-z)^{-1}$ Resolvent
1	1	$\exp(z)$ Exponential
$\frac{1}{2}$	1	$\exp(z^2)\,\mathrm{erfc}(-z)$ Error Function[1]
2	1	$\cosh(\sqrt{z})$ Hyperbolic Cosine
2	2	$\sinh(\sqrt{z})/\sqrt{z}$ Hyperbolic Sine
4	1	$\frac{1}{2}[\cos(z^{1/4}) + \cosh(z^{1/4})]$
1	$k \geq 2$	$z^{1-k}(e^z - \sum_{r=0}^{k-2} \frac{z^r}{r!})$ $\varphi_{k-1}(z) = \sum_{r=0}^{\infty} \frac{z^r}{(r+k-1)!}$

The Mittag-Leffler Function: other occurrences

Another use of it is in the case $E_{1,2}(z) = \psi_1(z)$ for computing the **non-backtracking exponential generating function** for simple graphs (Arrigo et al. 2018) is:

$$\sum_{r=0}^{\infty} \frac{p_r(A)}{r!} = \begin{bmatrix} I & 0 \end{bmatrix} \psi_1(Y) \begin{bmatrix} A \\ A^2 - D \end{bmatrix} + I,$$

where $p_r(A)$ is a matrix whose entries represent the number of non-backtracking walks of length r between any two given nodes



Backtracking walk

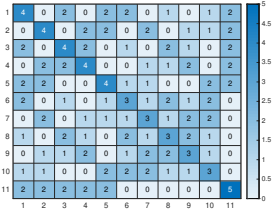
A walk is *backtracking* if it contains at least one pair of successive edeges of the form $i \mapsto j$, $j \mapsto i$. We say that is *non-backtracking* otherwise.

The Mittag-Leffler Function: other occurrences

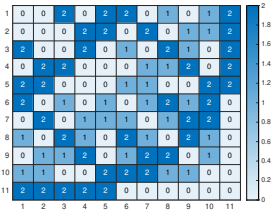
Another use of it is in the case $E_{1,2}(z) = \psi_1(z)$ for computing the **non-backtracking exponential generating function** for simple graphs (Arrigo et al. 2018) is:

$$\sum_{r=0}^{\infty} \frac{p_r(A)}{r!} = \begin{bmatrix} I & 0 \end{bmatrix} \psi_1(Y) \begin{bmatrix} A \\ A^2 - D \end{bmatrix} + I,$$

where $p_r(A)$ is a matrix whose entries represent the number of non-backtracking walks of length r between any two given nodes



A^2



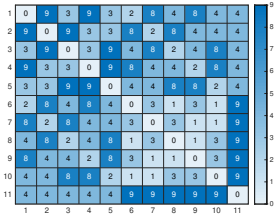
$p_2(A)$

The Mittag-Leffler Function: other occurrences

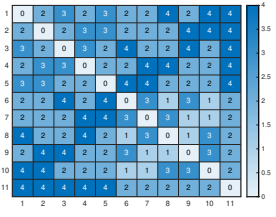
Another use of it is in the case $E_{1,2}(z) = \psi_1(z)$ for computing the **non-backtracking exponential generating function** for simple graphs (Arrigo et al. 2018) is:

$$\sum_{r=0}^{\infty} \frac{p_r(A)}{r!} = \begin{bmatrix} I & 0 \end{bmatrix} \psi_1(Y) \begin{bmatrix} A \\ A^2 - D \end{bmatrix} + I,$$

where $p_r(A)$ is a matrix whose entries represent the number of non-backtracking walks of length r between any two given nodes



A^3



$p_3(A)$

The Mittag-Leffler Function: other occurrences

Another use of it is in the case $E_{1,2}(z) = \psi_1(z)$ for computing the **non-backtracking exponential generating function** for simple graphs (Arrigo et al. 2018) is:

$$\sum_{r=0}^{\infty} \frac{p_r(A)}{r!} = \begin{bmatrix} I & 0 \end{bmatrix} \psi_1(Y) \begin{bmatrix} A \\ A^2 - D \end{bmatrix} + I,$$

where $p_r(A)$ is a matrix whose entries represent the number of non-backtracking walks of length r between any two given nodes $D = \mathrm{diag}(A)$, and Y is the first companion linearization of the matrix polynomial $(D - I) - A\lambda + I\lambda^2$:

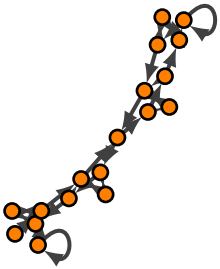
$$Y = \begin{bmatrix} 0 & I \\ I - D & A \end{bmatrix}.$$

The Mittag-Leffler Function: other occurrences

To compute **centrality** and **communicability** indices for **directed networks**, if A is the adjacency matrix of a directed graph, then

$$\mathcal{A} = \begin{bmatrix} O & A \\ A^T & O \end{bmatrix} \Rightarrow \exp(\mathcal{A}) = \begin{bmatrix} \cosh(\sqrt{AA^T}) & A(\sqrt{A^TA})^\dagger \sinh(\sqrt{A^TA}) \\ \sinh(\sqrt{A^TA})(\sqrt{A^TA})^\dagger A^T & \cosh(\sqrt{A^TA})) \end{bmatrix}$$

Centrality and communicability indices for directed networks defined by exploiting the representation of such networks as bipartite graphs; details in (Benzi, Estrada, and Klymko 2013).

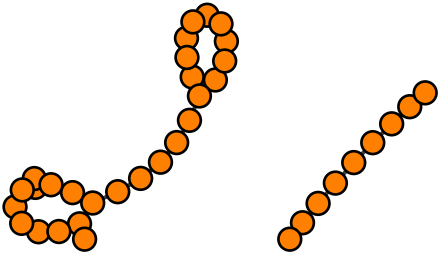


The Mittag-Leffler Function: other occurrences

To compute **centrality** and **communicability** indices for **directed networks**, if A is the adjacency matrix of a directed graph, then

$$\mathcal{A} = \begin{bmatrix} O & A \\ A^T & O \end{bmatrix} \Rightarrow \exp(\mathcal{A}) = \begin{bmatrix} E_2(AA^T) & AE_{2,2}(A^TA) \\ E_{2,2}(A^TA)A & E_2(A^TA) \end{bmatrix}$$

Centrality and communicability indices for directed networks defined by exploiting the representation of such networks as bipartite graphs; details in (Benzi, Estrada, and Klymko 2013).

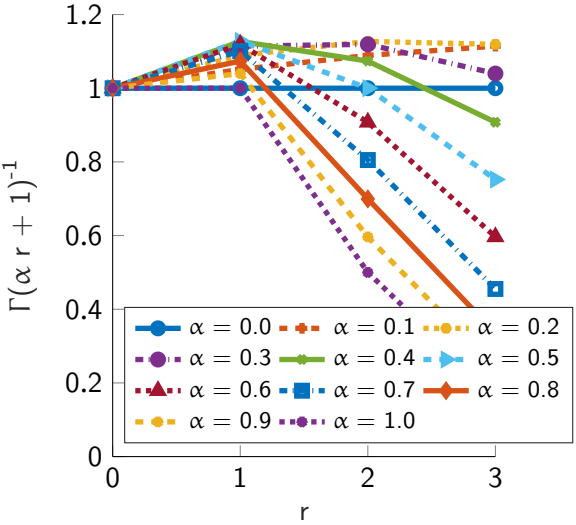


Defining Mittag-Leffler based centralities

For each choice of $\alpha, \beta > 0$ we want to define 🚶 centralities based on

$$E_{\alpha,\beta}(z) = \sum_{r=0}^{\infty} c_r(\alpha, \beta) z^r$$
$$= \sum_{r=0}^{\infty} \frac{z^r}{\Gamma(\alpha r + \beta)},$$

The idea of a 🚶 centrality relies on the fact that walks of longer lengths are less important, **but** $c(r) := \Gamma(\alpha r + 1)$ is not monotonic for certain values of $\alpha \in (0, 1)$!



Enforcing monotonicity

Lemma (Arrigo, D.)

Suppose that $\alpha \in (0,1)$. The coefficients $\widetilde{c}_r(\alpha,\gamma) = \gamma^r c_r(\alpha)$ defining the power series for the entire function $\widetilde{E}_\alpha(z) = E_\alpha(\gamma z)$ are monotonically decreasing as a function of $r = 0, 1, 2, \ldots$ for all $0 < \gamma < \Gamma(\alpha+1)$.

Proof. For each $\alpha \in (0,1)$ we want to determine conditions on $\gamma = \gamma(\alpha)$ that imply that

$$\tilde{c}_r(\alpha,\gamma) \geq \tilde{c}_{r+1}(\alpha,\gamma) \quad \text{for all } r \in \mathbb{N}$$

From the definition of $\tilde{c}_r(\alpha,\gamma)$ we have that the above inequality is equivalent to verifying

$$\gamma \leq \frac{\Gamma(\alpha r + \alpha + 1)}{\Gamma(\alpha r + 1)}, \quad \text{for all } r \geq 0$$

since $\gamma > 0$ and $\Gamma(x) > 0$ for all $x \geq 0$.

Enforcing monotonicity

Lemma (Arrigo, D.)

Suppose that $\alpha \in (0,1)$. The coefficients $\widetilde{c}_r(\alpha,\gamma) = \gamma^r c_r(\alpha)$ defining the power series for the entire function $\widetilde{E}_\alpha(z) = E_\alpha(\gamma z)$ are monotonically decreasing as a function of $r = 0, 1, 2, \ldots$ for all $0 < \gamma < \Gamma(\alpha+1)$.

Proof. Since H_x, the *Harmonic number* for $x \in \mathbb{R}$, is an increasing function of x, $\alpha > 0$ by hypothesis, and $\Gamma(x) > 0$ for all $x \geq 0$, it follows that

$$\frac{d}{dx}\left(\frac{\Gamma(\alpha x+\alpha+1)}{\Gamma(\alpha x+1)}\right) = \frac{\alpha\left(H_{\alpha(x+1)} - H_{\alpha x}\right)\Gamma(\alpha x+\alpha+1)}{\Gamma(\alpha x+1)} \geq 0,$$

and thus the minimum of $\frac{\Gamma(\alpha x+\alpha+1)}{\Gamma(\alpha x+1)}$ is achieved at $x = 0$.

Enforcing monotonicity

Lemma (Arrigo, D.)

Suppose that $\alpha \in (0,1)$. The coefficients $\widetilde{c}_r(\alpha,\gamma) = \gamma^r c_r(\alpha)$ defining the power series for the entire function $\widetilde{E}_\alpha(z) = E_\alpha(\gamma z)$ are monotonically decreasing as a function of $r = 0, 1, 2, \ldots$ for all $0 < \gamma < \Gamma(\alpha+1)$.

Proof. Since H_x, the *Harmonic number* for $x \in \mathbb{R}$, is an increasing function of x, $\alpha > 0$ by hypothesis, and $\Gamma(x) > 0$ for all $x \geq 0$, it follows that

$$\frac{d}{dx}\left(\frac{\Gamma(\alpha x+\alpha+1)}{\Gamma(\alpha x+1)}\right) = \frac{\alpha\left(H_{\alpha(x+1)} - H_{\alpha x}\right)\Gamma(\alpha x+\alpha+1)}{\Gamma(\alpha x+1)} \geq 0,$$

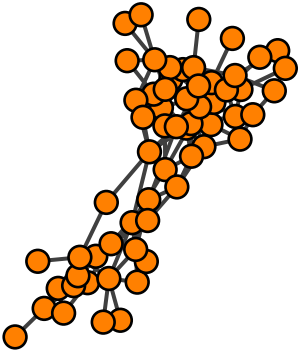
and thus the minimum of $\frac{\Gamma(\alpha x+\alpha+1)}{\Gamma(\alpha x+1)}$ is achieved at $x = 0$.

Take-home message

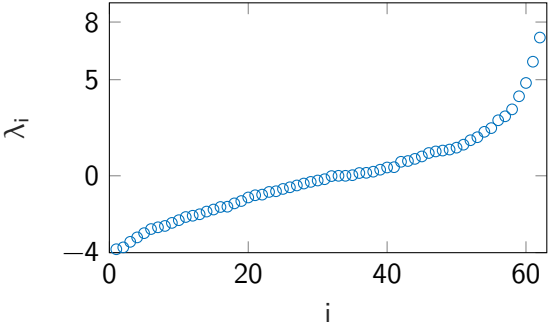
Mittag–Leffler functions with $\alpha \in (0,1)$ can be employed since they have a power series expansion that can be interpreted in terms of walks; however, care should be taken since to enforce monotonic behavior of the coefficients.

A matter of magnitude

Adjacency matrices of simple graphs have **positive** and **negative** eigenvalues ($\mathrm{tr}(A) = 0$)!



Newmann/Dolphins



$E_{0.4,1}(\rho(A)) = E_{0.4,1}(7.1936\ldots) \approx 10^{60}$

A matter of magnitude

We know asymptotic expansions for the ML function for $\theta \in (\frac{\pi\alpha}{2}, \min(\pi, \alpha\pi))$ and any $p \in \mathbb{N}$:

Proposition (Gorenflo et al. 2014, Proposition 3.6)

Let $0 < \alpha < 2$ and $\theta \in (\frac{\pi\alpha}{2}, \min(\pi, \alpha\pi))$. Then we have the following asymptotics for the Mittag–Leffler function for any $p \in \mathbb{N}$

$$E_\alpha(z) = \frac{1}{\alpha} e^{z^{\frac{1}{\alpha}}} - \sum_{k=1}^{p} \frac{z^{-k}}{\Gamma(1-\alpha k)} + O(|z|^{-1-p}),\ |z| \to +\infty,\ |\arg(z)| \leq \theta,$$

$$E_\alpha(z) = -\sum_{k=1}^{p} \frac{z^{-k}}{\Gamma(1-\alpha k)} + O(|z|^{-1-p}),\ |z| \to +\infty,\ \theta \leq |\arg(z)| \leq \pi.$$

We need to set the γ to **scale the largest modulus eigenvalue** in the computable range!

A matter of magnitude

Lemma (Arrigo, D.)

Suppose that $\alpha \in (0, 1]$, and $A \in \mathbb{R}^{n \times n}$ is symmetric. Then for all

$$\gamma \leq \frac{1}{\lambda_{\max}(A)} \left(\bar{K} \log(10) + \log(\alpha)\right)^{\alpha}$$

it holds that $\max_{i,j}(|E_\alpha(\gamma A)|)_{i,j} \leq \bar{N}$ where $\bar{N} \approx 10^{\bar{K}}$ for a given $\bar{K} \in \mathbb{N}$ is the largest representable number on a given machine.

Proof. We have $\lambda_{\max}(\gamma A) = \gamma \lambda_{\max}(A) \in \mathbb{R}$, since A is symmetric; then empolying the asymptotic expansion, and using the fact that $\arg(z) = 0$ for $z \in \mathbb{R}$, for $p = 0$ we find

$$\frac{1}{\alpha} e^{(\gamma \lambda_{\max}(A))^{\frac{1}{\alpha}}} \leq \bar{N} \approx 10^{\bar{K}},$$

which immediately yields the conclusion.

Well-posedness and machine representability

Subgraph and total communicability centralities

Let A be the adjacency matrix of a simple graph $G = (V, E)$. Let $\alpha \in [0, 1]$ and let $0 < \gamma \leq \mu(\alpha)$. Then, for all nodes $i \in V = \{1, 2, \ldots, n\}$ we define:

- ML-subgraph centrality:

$$s_i(\widetilde{E}_\alpha) = E_\alpha(\gamma A)_{ii}$$

- ML-total communicability:

$$t_i(\widetilde{E}_\alpha) = (E_\alpha(\gamma A)\mathbf{1})_i$$

Proposition (Arrigo, D.)

Let A be the adjacency matrix of an undirected network with at least one edge and let $\rho(A) > 0$ be its spectral radius. Moreover, let $\bar{N} \approx 10^{\bar{K}}$ be the largest representable number on a given machine. Then the Mittag–Leffler function $\widetilde{E}_\alpha(z) = E_\alpha(\gamma z)$ is representable in the machine, and admits a series expansion with decreasing coefficients when $\alpha \in (0, 1)$ and $0 < \gamma \leq \mu(\alpha)$

$$\mu(\alpha) := \min \begin{cases} \Gamma(\alpha + 1), \\ \dfrac{\left(\bar{K}\log(10) + \log(\alpha)\right)^\alpha}{\rho(A)} \end{cases}$$

The main idea behind ML centralities

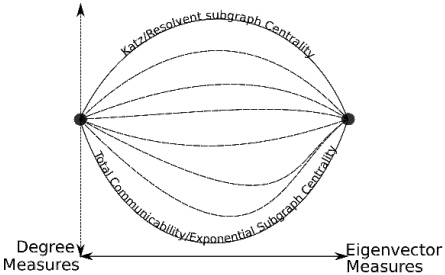
Theorem (Benzi and Klymko 2015)

Let $G = (V, E)$ be a connected, undirected, unweighted network with primitive A, and f an analytic function with strictly positive series expansion defined on the spectrum of A.

- For $\gamma \to 0^+$, the rankings produced by both $\mathbf{s}(\gamma)$ and $\mathbf{t}(\gamma)$ converge to those produced by the vector of degree centralities,
- If in addition f is analytic on the whole real axis or is such that,

$$\sum_{r=0}^{\infty} c_r R_f^r = \lim_{\gamma \to 1^-} \sum_{r=0}^{\infty} c_r t^r R_f^t = +\infty,$$

then, for $t \to R_f/\rho(A)$, the rankings produced by both $\mathbf{s}(\gamma)$ and $\mathbf{t}(\gamma)$ converge to those produced by the eigenvector centrality.



We build measures that "interpolate asymptotically" between four other "central" centralities measures: Degree, Eigenvector, Exponential and Resolvent walk centralities.

ML matrix-function vector products

The tasks of computing ML-subgraph centrality and ML-total communicability relies on the task of computing the ML function "with matrix argument", which is a delicate task

- We can use, e.g., the techniques and the code developed in (Garrappa and Popolizio 2018),
- then for "large networks" we adopt a polynomial Krylov subspace projection technique (Moret and Novati 2011) to handle the computations
  - For V a basis of $\mathcal{K}_m(A, \mathbf{1}) = \mathrm{span}\{\mathbf{v}, A\mathbf{v}, \ldots, A^{m-1}\mathbf{v}\}$

$$\mathbf{t}(\gamma) \approx V E_\alpha(\gamma V^T A V) V^T \mathbf{1},$$

  - For V a basis of $\mathcal{K}_m(A, \mathbf{e}_i) = \mathrm{span}\{\mathbf{e}_i, A\mathbf{e}_i, \ldots, A^{m-1}\mathbf{e}_i\}$,

$$s_i(\gamma) \approx \mathbf{e}_i^T V E_\alpha(\gamma V^T A V) V^T \mathbf{e}_i.$$

ML matrix-function vector products

The tasks of computing ML-subgraph centrality and ML-total communicability relies on the task of computing the ML function "with matrix argument", which is a delicate task

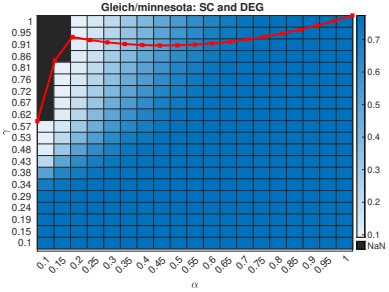
- We can use, e.g., the techniques and the code developed in (Garrappa and Popolizio 2018),
- then for "large networks" we adopt a polynomial Krylov subspace projection technique (Moret and Novati 2011) to handle the computations
  - For V a basis of $\mathcal{K}_m(A, \mathbf{e}_i) = \mathrm{span}\{\mathbf{e}_i, A\mathbf{e}_i, \ldots, A^{m-1}\mathbf{e}_i\}$,

$$s_i(\gamma) \approx \mathbf{e}_i^T V E_\alpha(\gamma V^T A V) V^T \mathbf{e}_i.$$

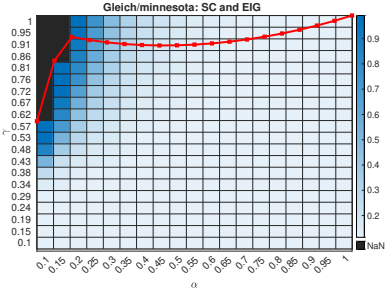
Subgraph centrality is computationally **quite expensive** to derive for all nodes but approximation techniques for few top ranked nodes are available (Fenu et al. 2013).

Numerical Examples

We compare subgraph centrality with **eigenvector centrality** and **degree centrality** as we let α and γ vary on a real-world network



(a)

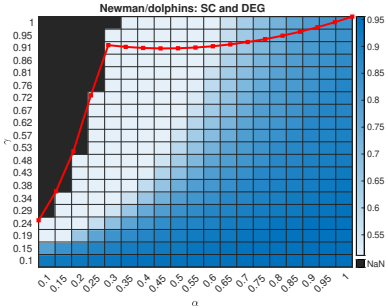


(b)

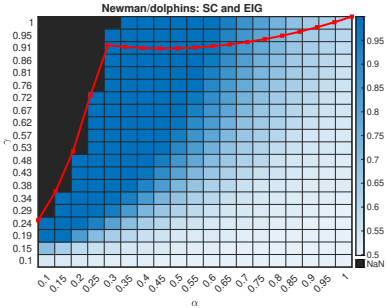
Kendall correlation coefficient between the ranking induced by total communicability vectors $s(\widetilde{E}_\alpha)$ and by (a) degree centrality or (b) eigenvector centrality, the red line displays the value of μ.

Numerical Examples

We compare total comunicability with **eigenvector centrality** and **degree centrality** as we let α and γ vary on a real-world network



(a)

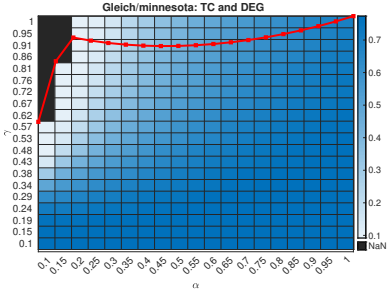


(b)

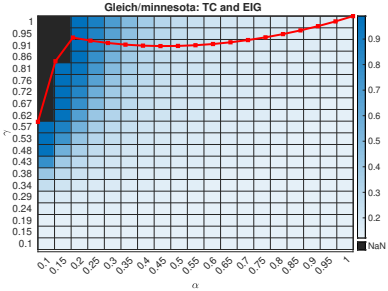
Kendall correlation coefficient between the ranking induced by total communicability vectors $s(\widetilde{E}_\alpha)$ and by (a) degree centrality or (b) eigenvector centrality, the red line displays the value of μ.

Numerical Examples

We compare with **eigenvector centrality** and **degree centrality** as we let α and γ vary on a real-world network



(a)

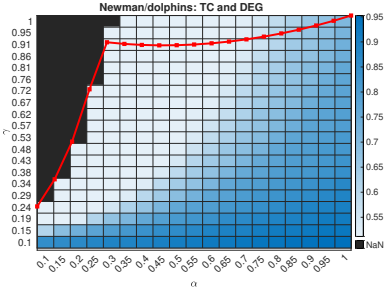


(b)

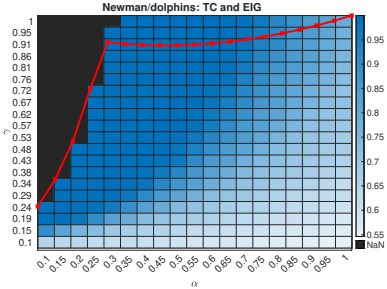
Kendall correlation coefficient between the ranking induced by total communicability vectors $\mathbf{t}(\widetilde{E}_\alpha)$ and by (a) degree centrality or (b) eigenvector centrality, the red line displays the value of μ.

Numerical Examples

We compare with **eigenvector centrality** and **degree centrality** as we let α and γ vary on a real-world network



(a)



(b)

Kendall correlation coefficient between the ranking induced by total communicability vectors $\mathbf{t}(\widetilde{E}_\alpha)$ and by (a) degree centrality or (b) eigenvector centrality, the red line displays the value of μ.

Time-fractional dynamical models on networks

There are **several generalizations** of ODE-based models on networks:

- Time (and space) generalized diffusion equation on networks (Diaz-Diaz and Estrada 2022)

$$ {}_{CA}D^{\alpha}_{[0,t]}\mathbf{f}(t) = -L\mathbf{f}(t), \quad f(0) = \mathbf{f}_0, $$

  for L the graph Laplacian, i.e., $L = \mathrm{diag}(A\mathbf{1}) - A$, A adjacency matrix of an *undirected graph*,
- Decision-making models (West, Turalska, and Grigolini 2015),
- Epidemics modeling with fractional derivative in time on newtorks, e.g., (Huo and Zhao 2016).

Time-fractional dynamical models on networks

There are **several generalizations** of ODE-based models on networks:

- Time (and space) generalized diffusion equation on networks (Diaz-Diaz and Estrada 2022)

$$ {}_{CA}D^{\alpha}_{[0,t]}\mathbf{f}(t) = -L\mathbf{f}(t), \quad f(0) = \mathbf{f}_0, $$

  for L the graph Laplacian, i.e., $L = \mathrm{diag}(A\mathbf{1}) - A$, A adjacency matrix of an *undirected graph*,
- Decision-making models (West, Turalska, and Grigolini 2015),
- Epidemics modeling with fractional derivative in time on newtorks, e.g., (Huo and Zhao 2016).

There are many more models that involve using **fractional derivatives with respect to the "space variables"**, we postpone that discussion after having treated the issue in general for the continuous case.

Other types of fractional derivatives w.r.t. time

Another type of FDE w.r.t. that is gaining traction and interest, they are called **fractional derivatives of distributed order**, i.e.,

$$\int_0^m a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = f(t), \quad m > 0,$$

and more generally

$$\int_0^m a(r)F\left({}_{CA}D^r_{[0,t]}u(t)\right)\,\mathrm{d}r = f(t, u(t)), \quad m > 0.$$

Applications are, e.g.,

- Dielectric induction and diffusion (Caputo 2001),
- Kinetic models (Sokolov, Chechkin, and Klafter 2004),
- Distributed-order oscillators (Atanackovic, Budincevic, and Pilipovic 2005).

Distributed order FDEs

We can connect them with something we have already seen, consider the **multi-term** differential equation:

$$\begin{cases} \displaystyle\sum_{i=1}^{k} \gamma_i {}_{CA}D_{[0,t]}^{r_i} u(t) = f(t, u(t)), & 0 < r_1 < r_2 < \ldots < r_k \\ u^{(\ell)}(0) = \varphi_\ell, & \ell = 0, \ldots, m-1, \ m = \left\lceil \displaystyle\max_{i=1,\ldots,k} r_i \right\rceil . \end{cases}$$

- One way of thinking about the distributed-order equation is therefore as the **limiting case** of with a very large number of terms and where the coefficients γ_i take the values from the function a.

Distributed order FDEs

We can connect them with something we have already seen, consider the **multi-term** differential equation:

$$\begin{cases} \displaystyle\sum_{i=1}^{k} \gamma_i {}_{CA}D^{r_i}_{[0,t]} u(t) = f(t, u(t)), & 0 < r_1 < r_2 < \ldots < r_k \\ u^{(\ell)}(0) = \varphi_\ell, & \ell = 0, \ldots, m-1, \ m = \left\lceil \displaystyle\max_{i=1,\ldots,k} r_i \right\rceil . \end{cases}$$

- One way of thinking about the distributed-order equation is therefore as the **limiting case** of with a very large number of terms and where the coefficients γ_i take the values from the function a.
- What can we say about the solutions?

Distributed order FDEs

For the **linear case**

$$\int_0^m a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = f(t), \quad m > 0, \tag{LDFODE}$$

we can prove existence under some *assumptions*:

(A1) $m \in \mathbb{N}$,

(A2) a is *absolutely integrable* on $[0, m]$ with $\int_0^m a(r)s^r\,\mathrm{d}r \neq 0$ for $\Re(s) > 0$,

(A3) $f \in \mathbb{L}^1([0, \infty)]$,

(A4) u is such that ${}_{CA}D^r_{[0,\infty)}u(t)$ for $t \in [0, +\infty)$ for $r \in [0, m]$.

Distributed order FDEs

For the **linear case**

$$\int_0^m a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = f(t), \quad m > 0, \tag{LDFODE}$$

we can prove existence under some *assumptions*:

(A1) $m \in \mathbb{N}$,

(A2) a is *absolutely integrable* on $[0, m]$ with $\int_0^m a(r)s^r\,\mathrm{d}r \neq 0$ for $\Re(s) > 0$,

(A3) $f \in \mathbb{L}^1([0, \infty)]$,

(A4) u is such that ${}_{CA}D^r_{[0,\infty)}u(t)$ for $t \in [0, +\infty)$ for $r \in [0, m]$.

We apply Laplace transform

$$\mathcal{L}\left\{\int_0^m a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r\right\}(s) = \mathcal{L}\{f\}(s)$$

Distributed order FDEs

For the **linear case**

$$\int_0^m a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = f(t), \quad m > 0, \tag{LDFODE}$$

we can prove existence under some *assumptions*:

(A1) $m \in \mathbb{N}$,

(A2) a is *absolutely integrable* on $[0, m]$ with $\int_0^m a(r)s^r\,\mathrm{d}r \neq 0$ for $\Re(s) > 0$,

(A3) $f \in \mathbb{L}^1([0, \infty)]$,

(A4) u is such that ${}_{CA}D^r_{[0,\infty)}u(t)$ for $t \in [0, +\infty)$ for $r \in [0, m]$.

We apply Laplace transform, then use (A4) and exchange the transform and the integral

$$\int_0^m a(r)\mathcal{L}\left\{{}_{CA}D^r_{[0,t]}u\right\}(s)\,\mathrm{d}r = \mathcal{L}\{f\}(s)$$

Distributed order FDEs

For the **linear case**

$$\int_0^m a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = f(t), \quad m > 0, \tag{LDFODE}$$

we can prove existence under some *assumptions*:

(A1) $m \in \mathbb{N}$,

(A2) a is *absolutely integrable* on $[0, m]$ with $\int_0^m a(r)s^r\,\mathrm{d}r \neq 0$ for $\Re(s) > 0$,

(A3) $f \in \mathbb{L}^1([0, \infty)]$,

(A4) u is such that ${}_{CA}D^r_{[0,\infty)}u(t)$ for $t \in [0, +\infty)$ for $r \in [0, m]$.

We apply Laplace transform, then use (A4) and exchange the transform and the integral

$$\int_0^m a(r)\left(s^r\mathcal{L}\{u\}(s) - u(0)s^{r-1}\right)\,\mathrm{d}r - \sum_{j=1}^{m-1}\int_j^m a(r)u^{(j)}(0)s^{r-j-1}\,\mathrm{d}r = \mathcal{L}\{f\}(s)$$

Distributed order FDEs

For the **linear case**

$$\int_0^m a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = f(t), \quad m > 0, \tag{LDFODE}$$

we can prove existence under some *assumptions*:

(A1) $m \in \mathbb{N}$,

(A2) a is *absolutely integrable* on $[0, m]$ with $\int_0^m a(r)s^r\,\mathrm{d}r \neq 0$ for $\Re(s) > 0$,

(A3) $f \in \mathbb{L}^1([0, \infty)]$,

(A4) u is such that ${}_{CA}D^r_{[0,\infty)}u(t)$ for $t \in [0, +\infty)$ for $r \in [0, m]$.

We apply Laplace transform, then use (A4) and exchange the transform and the integral. After rearranging and inverting using (A1)–(A3)

$$u(t) = u(0) + \left(f * \mathcal{L}^{-1}\left\{\frac{1}{\int_0^m a(z)(s)^z\,\mathrm{d}z}\right\}\right)(t) + \sum_{j=1}^{m-1} u^{(j)}(0)\mathcal{L}^{-1}\left\{\frac{\int_j^m a(r)s^{r-j-1}\,\mathrm{d}r}{\int_0^m s^r a(r)\,\mathrm{d}r}\right\}(t).$$

Distributed order FDEs

For the **linear case**

$$\int_0^m a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = f(t), \quad m > 0, \tag{LDFODE}$$

we can prove existence under some *assumptions*:

(A1) $m \in \mathbb{N}$,

(A2) a is *absolutely integrable* on $[0, m]$ with $\int_0^m a(r)s^r\,\mathrm{d}r \neq 0$ for $\Re(s) > 0$,

(A3) $f \in \mathbb{L}^1([0, \infty)]$,

(A4) u is such that ${}_{CA}D^r_{[0,\infty)}u(t)$ for $t \in [0, +\infty)$ for $r \in [0, m]$.

Theorem (Diethelm and Ford 2009, Theorem 3.1)

Under assumptions (A1)–(A4) on a, f and u, (LDFODE) has a unique solution.

Properties of the (LDFODE) solution

Proposition (Diethelm and Ford 2009)

1. Under assumptions (A1)–(A4) and for fixed $T > 0$ the solution to (LDFODE) satisfies $u^{(m)}(t)$ is bounded and measurable in $[0, T]$.
2. Let $u \in \mathcal{C}^p([O, T])$ with some $p \in \mathbb{N}$ and $T > 0$. For every fixed $t \in [0, T]$, consider ${}_{CA}D^r_{[0,t]}u(t) = z(r)$ as a function of r. Then,
   - At the integer argument $j = 1, 2, \ldots, p-1$ the function z has a *jump discontinuity* that can be described as

$$\lim_{r \to j^+} z(r) - \lim_{r \to j^-} z(r) = -u^{(j)}(0).$$

   - There exist a *continuous transition* iff $u^{(j)}(0) = 0$.

Properties of the (LDFODE) solution

Proposition (Diethelm and Ford 2009)

1. Under assumptions (A1)–(A4) and for fixed $T > 0$ the solution to (LDFODE) satisfies $u^{(m)}(t)$ is bounded and measurable in $[0, T]$.
2. Let $u \in \mathcal{C}^p([O, T])$ with some $p \in \mathbb{N}$ and $T > 0$. For every fixed $t \in [0, T]$, consider ${}_{CA}D^r_{[0,t]}u(t) = z(r)$ as a function of r. Then,
   - At the integer argument $j = 1, 2, \ldots, p-1$ the function z has a *jump discontinuity* that can be described as

$$\lim_{r \to j^+} z(r) - \lim_{r \to j^-} z(r) = -u^{(j)}(0).$$

   - There exist a *continuous transition* iff $u^{(j)}(0) = 0$.

❓ How can we discretize and solve this type of equations?

Discretization strategies

1. We discretize the **integral term** in the **distributed-order** equation

2. We solve the multi-term equation

Discretization strategies

1. We discretize the **integral term** in the **distributed-order** equation
   - Fix $\phi(z) = a(z)\,{}_{CA}D^{z}_{[0,t]}u(t)$ and use a quadrature formula

$$\int_0^m \phi(z)\,\mathrm{d}z \approx \sum_{j=0}^{n} w_j \phi(z_j)$$

   - Every integer value in the interval $[0, m]$ is a z_j, in general every $z_j \in \mathbb{Q}$.
2. We solve the multi-term equation

Discretization strategies

1. We discretize the **integral term** in the **distributed-order** equation
   - Fix $\phi(z) = a(z)\,{}_{CA}D^{z}_{[0,t]}u(t)$ and use a quadrature formula

$$\int_0^m \phi(z)\,\mathrm{d}z \approx \sum_{j=0}^{n} w_j \phi(z_j)$$

   - Every integer value in the interval $[0, m]$ is a z_j, in general every $z_j \in \mathbb{Q}$.
2. We solve the multi-term equation
   - With the choice we have made we now have a multiterm equation of the form

$$\sum_{j=0}^{n} w_j a(z_j)\,{}_{CA}D^{z_j}_{[0,t]}u(t) = f(t), \quad z_1 < z_2 < \ldots < z_n,$$

Discretization strategies

1. We discretize the **integral term** in the **distributed-order** equation
   - Fix $\phi(z) = a(z)\,{}_{CA}D^{z}_{[0,t]}u(t)$ and use a quadrature formula

$$\int_0^m \phi(z)\,\mathrm{d}z \approx \sum_{j=0}^{n} w_j \phi(z_j)$$

   - ⚠ Every integer value in the interval $[0, m]$ is a z_j, in general every $z_j \in \mathbb{Q}$.
2. We solve the multi-term equation
   - With the choice we have made we now have a multiterm equation of the form

$$\sum_{j=0}^{n} w_j a(z_j)\,{}_{CA}D^{z_j}_{[0,t]}u(t) = f(t), \quad z_1 < z_2 < \ldots < z_n,$$

   - We apply the reformulation as a system of equations of order q being the **g**reatest **c**ommon **d**ivisor of the derivative orders.

Error analysis

To select the **quadrature formula** we have to take into account the **jumps in the integrand**

$$\int_0^m a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = \sum_{i=0}^{m-1}\int_i^{i+1} a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = \sum_{i=0}^{m-1}\sum_{j=0}^{n_i} w_{ij}\,a(z_{ij})\,{}_{CA}D^{z_{ij}}_{[0,t]}u(t)$$

with

- $z_{i0} = i,\ z_{i,n_i} = i+1,\ \forall i,$

Error analysis

To select the **quadrature formula** we have to take into account the **jumps in the integrand**

$$\int_0^m a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = \sum_{i=0}^{m-1}\int_i^{i+1} a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = \sum_{i=0}^{m-1}\sum_{j=0}^{n_i} w_{ij}\,a(z_{ij})\,{}_{CA}D^{z_{ij}}_{[0,t]}u(t)$$

with

- $z_{i0} = i,\ z_{i,n_i} = i+1,\ \forall i,$
- $j = 0,\ j = n_i$ the expressions ${}_{CA}D^{z_{ij}}_{[0,t]}u(t)$ must be interpreted as

$$\lim_{s\to z_{i0}^+} {}_{CA}D^s_{[0,t]}u(t) = \lim_{s\to i^+} {}_{CA}D^s_{[0,t]}u(t),$$

$$\lim_{s\to z_{in_i}^-} {}_{CA}D^s_{[0,t]}u(t) = \lim_{s\to (i+1)^-} {}_{CA}D^s_{[0,t]}u(t).$$

Error analysis

To select the **quadrature formula** we have to take into account the **jumps in the integrand**

$$\int_0^m a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = \sum_{i=0}^{m-1}\int_i^{i+1} a(r)\,{}_{CA}D^r_{[0,t]}u(t)\,\mathrm{d}r = \sum_{i=0}^{m-1}\sum_{j=0}^{n_i} w_{ij}\,a(z_{ij})\,{}_{CA}D^{z_{ij}}_{[0,t]}u(t)$$

with

- $z_{i0} = i,\ z_{i,n_i} = i+1,\ \forall i,$
- $j = 0,\ j = n_i$ the expressions ${}_{CA}D^{z_{ij}}_{[0,t]}u(t)$ must be interpreted as

$$\lim_{s\to z_{i0}^+} {}_{CA}D^s_{[0,t]}u(t) = \lim_{s\to i^+} {}_{CA}D^s_{[0,t]}u(t),$$

$$\lim_{s\to z_{in_i}^-} {}_{CA}D^s_{[0,t]}u(t) = \lim_{s\to (i+1)^-} {}_{CA}D^s_{[0,t]}u(t).$$

- The sequence $\{z_j\} = \{z_0 = z_{00}, z_1 = z_{01}, \ldots, z_{n_0} = z_{0n_0} = z_{10} = 1, \ldots\}.$

Error analysis

To proceed further we also need to require further regularity on the a function.
We assume

(Q1) We use a convergent quadrature rule of order $p > 0$,

(Q2) For all i, the weights of the quadrature rule are bounded by

$$C_1 n_i^{-1} \leq \min_{j=0,1,\ldots,n_i} |w_{ij}| \leq \max_{j=0,1,\ldots,n_i} |w_{ij}| \leq C_2 n_i^{-1},$$

with some constants C_1 and C_2.

(Q3) The function a is p-times continuously differentiable on $[0, m]$.

Proposition (Diethelm and Ford 2009)

If $\tilde{u}$ is the solution of (LDFODE) obtained using a quadrature formula satistying (Q1)–(Q4), then

$$u(t) = \tilde{u}(t) + O(\max_i \{n_i^{-p}\}), \qquad \text{for } n_i \to +\infty \ \forall\, i.$$

Error analysis

Thus, if we assume that we apply a numerical method for the multi-term equation which has order of convergence $O(\tau^q)$ we have then

Theorem (Diethelm and Ford 2009, Theorem 4.1)

Under the conditions (A1)–(A4), (Q1)–(Q3), the overall error of the proposed algorithm for (LDFODE) satisfies for $j\tau \in [0, T]$:

$$\max\{|u_j - u(j\tau)| \ : \ j \geq 0, \ j\tau \leq T\} = O(\tau^q) + O(\max_i\{n_i^{-p}\}) \qquad \text{for } n_i \to +\infty \ \forall\, i, \ \tau \to 0.$$

- To reduce the number of terms and the regularity requirements on a one could use a Gauss-type quadrature built explicitely for the given function $a(z)$ (that now needs to be only continuous) (Durastante 2019).

Variable order FDEs

Consider a function $\alpha : [0, T] \subset \mathbb{R}^+ \to (0, 1)$ we can think of generalizing the Riemann-Liouville integral as

$$I_{[0,t]}^{\alpha(t)} = \frac{1}{\Gamma(\alpha(t))} \int_0^t (t-\tau)^{\alpha(t)-1} f(\tau)\, \mathrm{d}\tau,$$

possibly coupled with the Riemann-Liouville variable-order derivative

$${}_{RL}D_{[0,t]}^{\alpha(t)} = \frac{1}{\Gamma(1-\alpha(t))} \frac{d}{dt} \int_0^t (t-\tau)^{-\alpha(t)} f(\tau)\, \mathrm{d}\tau,$$

Variable order FDEs

Consider a function $\alpha : [0, T] \subset \mathbb{R}^+ \to (0, 1)$ we can think of generalizing the Riemann-Liouville integral as

$$I_{[0,t]}^{\alpha(t)} = \frac{1}{\Gamma(\alpha(t))} \int_0^t (t-\tau)^{\alpha(t)-1} f(\tau)\, \mathrm{d}\tau,$$

possibly coupled with the Riemann-Liouville variable-order derivative

$${}_{RL}D_{[0,t]}^{\alpha(t)} = \frac{1}{\Gamma(1-\alpha(t))} \frac{d}{dt} \int_0^t (t-\tau)^{-\alpha(t)} f(\tau)\, \mathrm{d}\tau,$$

⚠ The **characterization** of fractional calculus **based** on **these operators** is rather **problematic** since ${}_{RL}D_{[0,t]}^{\alpha(t)}$ is **not a left-inverse** of $I_{[0,t]}^{\alpha(t)}$; see (Samko 1995).

Variable order FDEs

Consider a function $\alpha : [0, T] \subset \mathbb{R}^+ \to (0, 1)$ we can think of generalizing the Riemann-Liouville integral as

$$I_{[0,t]}^{\alpha(t)} = \frac{1}{\Gamma(\alpha(t))} \int_0^t (t-\tau)^{\alpha(t)-1} f(\tau)\, \mathrm{d}\tau,$$

possibly coupled with the Riemann-Liouville variable-order derivative

$${}_{RL}D_{[0,t]}^{\alpha(t)} = \frac{1}{\Gamma(1-\alpha(t))} \frac{d}{dt} \int_0^t (t-\tau)^{-\alpha(t)} f(\tau)\, \mathrm{d}\tau,$$

⚠ The **characterization** of fractional calculus **based** on **these operators** is rather **problematic** since ${}_{RL}D_{[0,t]}^{\alpha(t)}$ is **not a left-inverse** of $I_{[0,t]}^{\alpha(t)}$; see (Samko 1995).
Some of this generalizations have found use in physical modeling, but they are *problematic from a rigorous point of view*.

Variable order FDEs a Laplace domain version

Among the first ideas in developing a time-variable time-fractional calculus there are three seminal works by **Giambattista Scarpi**

- G. Scarpi, Sopra il moto laminare di liquidi a viscosist variabile nel tempo. Atti Accademia delle Scienze, Isitituto di Bologna, Rendiconti (Ser XII), 9 (1972), pp. 54-68,
- G. Scarpi, Sulla possibilità di un modello reologico intermedio di tipo evolutivo. Atti Accad Naz Lincei Rend Cl Sci Fis Mat Nat (8), 52 (1972), pp. 912-917;
- G. Scarpi, Sui modelli reologici intermedi per liquidi viscoelastici. Atti Accad Sci Torino: Cl Sci Fis Mat Natur, 107 (1973), pp. 239-243.

Recently, this approach has been taken again into account to overcome the limitation given by the *naive* replacement of the $\alpha(t)$ function in the kernel of Fractional Integrals and Derivatives; (Garrappa, Giusti, and Mainardi 2021).

Scarpi's Derivative (Garrappa, Giusti, and Mainardi 2021)

To introduce this new version we need to use again the **Laplace transform** of the Caputo derivative and Riemann-Liouville integrals

$$\mathcal{L}\{{}_{CA}D^{\alpha}_{[0,t]}f(t)\}(s) = s^{\alpha}F(s) - s^{\alpha-1}f(0), \quad \mathcal{L}\{I^{\alpha}_{[0,t]}f(t)\}(s) = \frac{1}{s^{\alpha}}F(s),$$

and consider a **locally integrable** function $\alpha(t) : [0, T] \to (0, 1)$.

Scarpi's Derivative (Garrappa, Giusti, and Mainardi 2021)

To introduce this new version we need to use again the **Laplace transform** of the Caputo derivative and Riemann-Liouville integrals

$$\mathcal{L}\{{}_{CA}D^{\alpha}_{[0,t]}f(t)\}(s) = s^{\alpha}F(s) - s^{\alpha-1}f(0), \quad \mathcal{L}\{I^{\alpha}_{[0,t]}f(t)\}(s) = \frac{1}{s^{\alpha}}F(s),$$

and consider a **locally integrable** function $\alpha(t) : [0, T] \to (0, 1)$.

Scarpi's idea

If $\alpha(t) \equiv \alpha$, $t > 0$, $\mathcal{L}\alpha(s) = A(s) = \alpha/s$, then

$$\mathcal{L}\left\{\frac{t^{-\alpha}}{\Gamma(1-\alpha)}\right\}(s) = s^{sA(s)-1} = s^{\alpha-1} \qquad \mathcal{L}\left\{\frac{t^{\alpha-1}}{\Gamma(\alpha)}\right\}(s) = s^{-sA(s)} = \frac{1}{s^{\alpha}}.$$

Apply the same relation to any $\alpha(t)$ with $A(s) = \mathcal{L}\{\alpha(t), s\} = \int_0^{+\infty} e^{-st}\alpha(t)\,\mathrm{d}t$.

Scarpi's Derivative (Garrappa, Giusti, and Mainardi 2021)

Scarpi Fractional Derivative

Let $\alpha(t) : [0, T] \to (0, 1)$ be a locally integrable function with Laplace transform $A(s)$, and let $f \in \mathbb{L}^1([0, T])$. We define the Scarpi fractional derivative ${}_SD^{\alpha(t)}_{[0,t]}$ of variable order $\alpha(t)$ as

$$ {}_SD^{\alpha(t)}_{[0,t]} f(t) = \frac{d}{dt} \int_0^t \phi_\alpha(t-\tau) f(\tau)\, \mathrm{d}\tau - \phi_\alpha(t) f(0), \qquad t \in (0, T], $$

where the kernel function $\phi_a(t)$ is the inverse Laplace transform

$$ \phi_a(t) = \mathcal{L}^{-1}\{\Phi_\alpha(s)\}(t), \qquad \Phi_\alpha(s) = s^{sA(s)-1}. $$

Scarpi's Derivative (Garrappa, Giusti, and Mainardi 2021)

Scarpi Fractional Derivative

Let $\alpha(t) : [0, T] \to (0, 1)$ be a locally integrable function with Laplace transform $A(s)$, and let $f \in \mathbb{L}^1([0, T])$. We define the Scarpi fractional derivative ${}_SD^{\alpha(t)}_{[0,t]}$ of variable order $\alpha(t)$ as

$$ {}_SD^{\alpha(t)}_{[0,t]} f(t) = \frac{d}{dt} \int_0^t \phi_\alpha(t-\tau) f(\tau)\, d\tau - \phi_\alpha(t) f(0), \qquad t \in (0, T], $$

where the kernel function $\phi_a(t)$ is the inverse Laplace transform

$$ \phi_a(t) = \mathcal{L}^{-1}\{\Phi_\alpha(s)\}(t), \qquad \Phi_\alpha(s) = s^{sA(s)-1}. $$

Proposition (Garrappa, Giusti, and Mainardi 2021, Proposition 2.1)

Let $\alpha(t) : [0, T] \to (0, 1)$ be a locally integrable function with Laplace transform $A(s)$, let $\phi_\alpha(t)$ be the inverse Laplace transform of $\Phi_\alpha(s) = s^{sA(s)-1}$, if $f \in \mathbb{A}([0, T])$ then

$$ {}_SD^{\alpha(t)}_{[0,t]} f(t) = \int_0^t \phi_\alpha(t-\tau) f'(\tau)\, d\tau, \qquad t \in [0, T]. $$

Scarpi's Integral (Garrappa, Giusti, and Mainardi 2021)

To "fix" the behavior of the naive definition we need also the related formulation of the fractional integral, that is having an operator for which

$$ {}_{S}D^{\alpha(t)}_{[0,t]}\, {}_{S}I^{\alpha(t)}_{[0,t]} f(t) = f(t) \qquad I^{\alpha(t)}_{[0,t]}\, {}_{S}D^{\alpha(t)}_{[0,t]} f(t) = f(t) - f(0), $$

Going there-and-back the Laplace domain can be rewritten as the **Sonine condition**

$$ \int_0^t \phi_\alpha(t-\tau)\psi_\alpha(\tau) = 1, \qquad t > 0. $$

Scarpi's Integral (Garrappa, Giusti, and Mainardi 2021)

To "fix" the behavior of the naive definition we need also the related formulation of the fractional integral, that is having an operator for which

$$ {}_sD^{\alpha(t)}_{[0,t]}\, {}_sI^{\alpha(t)}_{[0,t]} f(t) = f(t) \qquad I^{\alpha(t)}_{[0,t]}\, {}_sD^{\alpha(t)}_{[0,t]} f(t) = f(t) - f(0), $$

Going there-and-back the Laplace domain can be rewritten as the **Sonine condition**

$$ \int_0^t \phi_\alpha(t-\tau)\psi_\alpha(\tau) = 1, \qquad t > 0. $$

Scarpi Fractional Integral

Let $\alpha : [0, T] \to (0, 1)$ be a locally integrable function with Laplace transform $A(s)$, let $f \in \mathbb{L}^1([0, T])$ we define the Scarpi fractional integral as

$$ {}_sI^{\alpha(t)}_{[0,t]} f(t) = \int_0^t \psi_\alpha(t-\tau) f(\tau)\, \mathrm{d}\tau, $$

with $\psi_\alpha(t) = \mathcal{L}^{-1}\{\Psi_\alpha(s)\}(t)$ for $\Psi_\alpha(s) = s^{-sA(s)}$.

Finding good $\alpha(t)$ (Garrappa, Giusti, and Mainardi 2021)

In principle not all transition functions $\alpha(t)$ will allow for a suitable definition of a pair of Scarpi's operators.

Finding good $\alpha(t)$ (Garrappa, Giusti, and Mainardi 2021)

In principle not all transition functions $\alpha(t)$ will allow for a suitable definition of a pair of Scarpi's operators.

Necessary condition A necessary requirement to ensure that $\phi(t)$ and $\psi(t)$ form a **Sonine pair** is for them to have an integrable singularity at the origin. two functions

Finding good $\alpha(t)$ (Garrappa, Giusti, and Mainardi 2021)

In principle not all transition functions $\alpha(t)$ will allow for a suitable definition of a pair of Scarpi's operators.

Necessary condition A necessary requirement to ensure that $\phi(t)$ and $\psi(t)$ form a **Sonine pair** is for them to have an integrable singularity at the origin. two functions

Reality we also want our Kernels to be real, but this follows from having a real $\alpha(t)$ and hence $\overline{A(\overline{s})} = A(s)$,

Finding good $\alpha(t)$ (Garrappa, Giusti, and Mainardi 2021)

In principle not all transition functions $\alpha(t)$ will allow for a suitable definition of a pair of Scarpi's operators.

Necessary condition A necessary requirement to ensure that $\phi(t)$ and $\psi(t)$ form a **Sonine pair** is for them to have an integrable singularity at the origin. two functions

Reality we also want our Kernels to be real, but this follows from having a real $\alpha(t)$ and hence $\overline{A(\overline{s})} = A(s)$,

Kernels are LT a necessary conditions to have $\Phi_\alpha(s)$ and $\Psi_\alpha(s)$ Laplace transform of two functions $\phi_\alpha(t)$ and $\psi_\alpha(t)$ is to require

$$\lim_{t \to 0^+} \alpha(t) = \overline{\alpha} \in (0, 1)$$

and then the **initial value Theorem for the Laplace transform** ensures that $\{\Phi_\alpha, \Psi_\alpha\} \to 0$ for $s \to +\infty$, and thus they are the LT transform of two functions.

Finding good $\alpha(t)$ (Garrappa, Giusti, and Mainardi 2021)

In principle not all transition functions $\alpha(t)$ will allow for a suitable definition of a pair of Scarpi's operators.

Necessary condition A necessary requirement to ensure that $\phi(t)$ and $\psi(t)$ form a **Sonine pair** is for them to have an integrable singularity at the origin. two functions

Reality we also want our Kernels to be real, but this follows from having a real $\alpha(t)$ and hence $\overline{A(\overline{s})} = A(s)$,

Kernels are LT a necessary conditions to have $\Phi_\alpha(s)$ and $\Psi_\alpha(s)$ Laplace transform of two functions $\phi_\alpha(t)$ and $\psi_\alpha(t)$ is to require

$$\lim_{t \to 0^+} \alpha(t) = \overline{\alpha} \in (0, 1)$$

and then the **initial value Theorem for the Laplace transform** ensures that $\{\Phi_\alpha, \Psi_\alpha\} \to 0$ for $s \to +\infty$, and thus they are the LT transform of two functions.

$\Rightarrow$ Any function $\alpha(t)$ with LT $A(s)$ is suitable provided tha $\Phi_\alpha(s)$ and $\Psi_\alpha(s)$ are LTs of some functions.

Solving FDEs with Scarpi's Derivative

Consider the case

$$\begin{cases} {}_{S}D^{\alpha(t)}_{[0,t]}y(t) = -\lambda y(t), \\ y(0) = y_0 \end{cases} \qquad \mathbb{R} \ni \lambda > 0$$

Solving FDEs with Scarpi's Derivative

Consider the case

$$\begin{cases} {}_{S}D^{\alpha(t)}_{[0,t]}y(t) = -\lambda y(t), \\ y(0) = y_0 \end{cases} \qquad \mathbb{R} \ni \lambda > 0$$

1. We apply Laplace transform on both sides

$$s^{sA(s)}Y(s) - s^{sA(s)-1}y_0 = -\lambda Y(s)$$

where $Y(s) = \mathcal{L}\{y(t)\}(s)$

Solving FDEs with Scarpi's Derivative

Consider the case

$$\begin{cases} {}_{S}D^{\alpha(t)}_{[0,t]}y(t) = -\lambda y(t), \\ y(0) = y_0 \end{cases} \qquad \mathbb{R} \ni \lambda > 0$$

1. We apply Laplace transform on both sides

$$s^{sA(s)}Y(s) - s^{sA(s)-1}y_0 = -\lambda Y(s)$$

   where $Y(s) = \mathcal{L}\{y(t)\}(s)$

2. Solve for $Y(s)$

$$Y(s) = \frac{y_0}{s(1+\lambda\Psi_\alpha(s))},$$

Solving FDEs with Scarpi's Derivative

Consider the case

$$\begin{cases} {}_{S}D^{\alpha(t)}_{[0,t]}y(t) = -\lambda y(t), \\ y(0) = y_0 \end{cases} \qquad \mathbb{R} \ni \lambda > 0$$

1. We apply Laplace transform on both sides

$$s^{sA(s)}Y(s) - s^{sA(s)-1}y_0 = -\lambda Y(s)$$

   where $Y(s) = \mathcal{L}\{y(t)\}(s)$
2. Solve for $Y(s)$

$$Y(s) = \frac{y_0}{s(1+\lambda\Psi_\alpha(s))},$$

3. **Numerically invert the Laplace transform** with one of the algorithms we have seen when discussing the computation of the Mittag-Leffler function, e.g., parabolic contour and Trapezoidal quadrature

$$y(t) = \mathcal{L}^{-1}\{Y(s)\}(t).$$

An example

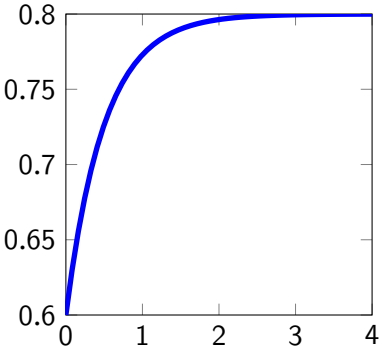
Consider the function

$$\alpha(t) = \alpha_2 + (\alpha_1 - \alpha_2)e^{-ct}$$

together with its Laplace transform

$$A(s) = \int_0^\infty e^{-st}\alpha(t)\,\mathrm{d}t = \frac{\alpha_2 c + \alpha_1 s}{s(c+s)}$$

```
alpha1 = 0.6;
alpha2 = 0.8;
c = 2.0;
a = @(t) alpha2 + (alpha1-alpha2).*exp(-c*t);
A = @(s) (alpha2*c + alpha1*s)./(s.*(c+s));
```



An example

Consider the function

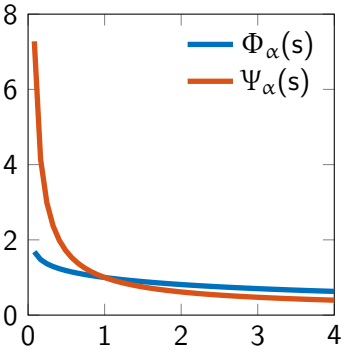
$$\alpha(t) = \alpha_2 + (\alpha_1 - \alpha_2)e^{-ct}$$

together with its Laplace transform

$$A(s) = \int_0^\infty e^{-st}\alpha(t)\,\mathrm{d}t = \frac{\alpha_2 c + \alpha_1 s}{s(c+s)}$$

We can easily visualize also the $\Psi_\alpha(s)$ and $\Phi_\alpha(s)$ kernels.

```
plot(t,t.^(t.*A(t)-1),'-',t,t.^(-t.*A(t)),'-
↪  ','LineWidth',2)
legend('\Phi_\alpha(s)','\Psi_\alpha(s)')
```



An example: inverting the Laplace transform

We can then solve

$$\begin{cases} {}_{S}D_{[0,t]}^{\alpha(t)} y(t) = -0.5 y(t), \\ y(0) = 1 \end{cases}$$

by first setting the various quantities:

```
y0 = 1;
lambda = 0.5;
Psi = @(s) s.^(-s.*A(s));
F = @(s) y0./(s.*(1 + lambda*Psi(s)));
```

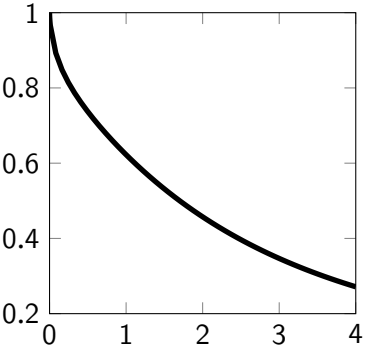
An example: inverting the Laplace transform

We can then solve

$$\begin{cases} {}_{s}D^{\alpha(t)}_{[0,t]}y(t) = -0.5y(t), \\ y(0) = 1 \end{cases}$$

Then inverting the Laplace transform on a **parabolic contour**

```
L = -log(eps); N = ceil(4*L/3/pi);
h = 2*pi/L + L/2/pi/N^2; p = L^3/4/pi^2/N^2;
u = (0:N)*h; f = zeros(size(t));
for n = 1:length(t)
 mu = p/t(n);
 z = mu*(u*1i + 1).^2; z1 = 2*mu*(1i-u);
 G = exp(z.*t(n)).*F(z).*z1;
 f(n) = (imag(G(1))/2+sum(imag(G(2:N+1))))*h/pi;
end
```



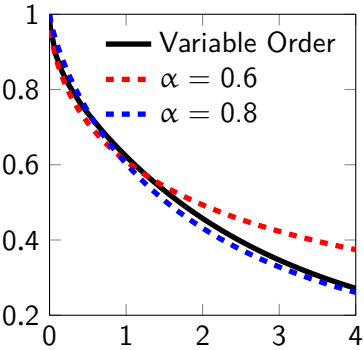
An example: inverting the Laplace transform

We can then solve

$$\begin{cases} {}_{S}D^{\alpha(t)}_{[0,t]}y(t) = -0.5y(t), \\ y(0) = 1 \end{cases}$$

And we can comapre the solution with the one obtained for the two fixed orders, observing that indeed we transition from one behavior to the other:

```
f_fun = @(t,y) -lambda*y;
J_fun = @(t,y) -lambda;
t0 = 0; T = 4; h = 1e-2;
alpha = alpha1;
[t1, y1] = fde_pi2_im(alpha,f_fun,J_fun,t0,T,y0,h);
alpha = alpha2;
[t2, y2] = fde_pi2_im(alpha,f_fun,J_fun,t0,T,y0,h);
```



Possible research directions

- Scarpi FDEs with more difficult dynamics, e.g., the vector case with a non-diagonalizable matrix, non-linear FDEs, *etc.*

Possible research directions

- Scarpi FDEs with more difficult dynamics, e.g., the vector case with a non-diagonalizable matrix, non-linear FDEs, *etc.*
- Algorithms for the automatic selection of contours and parameters given the FDE.

Possible research directions

- Scarpi FDEs with more difficult dynamics, e.g., the vector case with a non-diagonalizable matrix, non-linear FDEs, *etc.*
- Algorithms for the automatic selection of contours and parameters given the FDE.
- In the *complex-network case* Diaz-Diaz and Estrada 2022 explored the case of standard time-fractional evolutions, what about *distributed* or *variable* order? Are they reasonable from a modeling point of view? Can we **efficiently** use them?

Possible research directions

- Scarpi FDEs with more difficult dynamics, e.g., the vector case with a non-diagonalizable matrix, non-linear FDEs, *etc.*
- Algorithms for the automatic selection of contours and parameters given the FDE.
- In the *complex-network case* Diaz-Diaz and Estrada 2022 explored the case of standard time-fractional evolutions, what about *distributed* or *variable* order? Are they reasonable from a modeling point of view? Can we **efficiently** use them?
- All-at-once formulations for the *other* FDEs?

Possible research directions

- Scarpi FDEs with more difficult dynamics, e.g., the vector case with a non-diagonalizable matrix, non-linear FDEs, *etc.*
- Algorithms for the automatic selection of contours and parameters given the FDE.
- In the *complex-network case* Diaz-Diaz and Estrada 2022 explored the case of standard time-fractional evolutions, what about *distributed* or *variable* order? Are they reasonable from a modeling point of view? Can we **efficiently** use them?
- All-at-once formulations for the *other* FDEs?
- General poles for Rational Krylov methods for the computation of Mittag-Leffler matrix-function times vector algorithms?

Conclusions

In this **first part of the course** we have dealt with

- Defining and analyzing properties of Riemann-Liouville integral and derivatives,
- Defining and analyzing properties of Caputo integral and derivatives,
- Existence, uniqueness and regularity of FDEs with Caputo derivatives,
- Explored the connection between time-fractional derivatives and CTRW,
- FDEs with mulitple, distributed and variable orders.

For what concerns **numerical methods** we have seen

- Product Integral Rules and Fractional Linear Multistep Methods for integrating FDEs,
- An overview of some inversion techniques for the Laplace Transform,
- Computation of the Mittag-Leffler function and its derivative on scalar and matrix arguments,
- Krylov methods for the computation of matrix functions.

Programs for the (*near*) future

Bibliography I

- Arrigo, F. et al. (2018). "On the exponential generating function for non-backtracking walks". In: *Linear Algebra Appl.* 556, pp. 381–399. ISSN: 0024-3795. DOI: 10.1016/j.laa.2018.07.010. URL: https://doi.org/10.1016/j.laa.2018.07.010.
- Atanackovic, T. M., M. Budincevic, and S. Pilipovic (2005). "On a fractional distributed-order oscillator". In: *J. Phys. A* 38.30, pp. 6703–6713. ISSN: 0305-4470. DOI: 10.1088/0305-4470/38/30/006. URL: https://doi.org/10.1088/0305-4470/38/30/006.
- Benzi, M., E. Estrada, and C. Klymko (2013). "Ranking hubs and authorities using matrix functions". In: *Linear Algebra Appl.* 438.5, pp. 2447–2474. ISSN: 0024-3795. DOI: 10.1016/j.laa.2012.10.022. URL: https://doi.org/10.1016/j.laa.2012.10.022.
- Benzi, M. and C. Klymko (2015). "On the limiting behavior of parameter-dependent network centrality measures". In: *SIAM J. Matrix Anal. Appl.* 36.2, pp. 686–706. ISSN: 0895-4798. DOI: 10.1137/130950550. URL: https://doi.org/10.1137/130950550.

Bibliography II

- Caputo, M. (2001). "Distributed order differential equations modelling dielectric induction and diffusion". In: *Fract. Calc. Appl. Anal.* 4.4, pp. 421–442. ISSN: 1311-0454.
- Diaz-Diaz, F. and E. Estrada (2022). "Time and space generalized diffusion equation on graph/networks". In: *Chaos, Solitons & Fractals* 156, p. 111791. ISSN: 0960-0779. DOI: https://doi.org/10.1016/j.chaos.2022.111791. URL: https://www.sciencedirect.com/science/article/pii/S0960077922000029.
- Diethelm, K. and N. J. Ford (2009). "Numerical analysis for distributed-order differential equations". In: *Journal of Computational and Applied Mathematics* 225.1, pp. 96–104. ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2008.07.018. URL: https://www.sciencedirect.com/science/article/pii/S0377042708003464.
- Durastante, F. (2019). "Efficient solution of time-fractional differential equations with a new adaptive multi-term discretization of the generalized Caputo-Dzherbashyan derivative". In: *Calcolo* 56.4, Paper No. 36, 24. ISSN: 0008-0624. DOI: 10.1007/s10092-019-0329-0. URL: https://doi.org/10.1007/s10092-019-0329-0.

Bibliography III

- Fenu, C. et al. (2013). "Block Gauss and anti-Gauss quadrature with application to networks". In: *SIAM J. Matrix Anal. Appl.* 34.4, pp. 1655–1684. ISSN: 0895-4798. DOI: 10.1137/120886261. URL: https://doi.org/10.1137/120886261.
- Garrappa, R., A. Giusti, and F. Mainardi (2021). "Variable-order fractional calculus: a change of perspective". In: *Commun. Nonlinear Sci. Numer. Simul.* 102, Paper No. 105904, 16. ISSN: 1007-5704. DOI: 10.1016/j.cnsns.2021.105904. URL: https://doi.org/10.1016/j.cnsns.2021.105904.
- Garrappa, R. and M. Popolizio (2018). "Computing the matrix Mittag-Leffler function with applications to fractional calculus". In: *J. Sci. Comput.* 77.1, pp. 129–153. ISSN: 0885-7474. DOI: 10.1007/s10915-018-0699-5. URL: https://doi.org/10.1007/s10915-018-0699-5.
- Gorenflo, R. et al. (2014). *Mittag-Leffler Functions, Related Topics and Applications*. Springer Monographs in Mathematics. Springer, Heidelberg, pp. xiv+443. ISBN: 978-3-662-43929-6. DOI: 10.1007/978-3-662-43930-2.

Bibliography IV

- Huo, J. and H. Zhao (2016). "Dynamical analysis of a fractional SIR model with birth and death on heterogeneous complex networks". In: *Physica A: Statistical Mechanics and its Applications* 448, pp. 41–56. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2015.12.078. URL: https://www.sciencedirect.com/science/article/pii/S0378437115011061.
- Moret, I. and P. Novati (2011). "On the convergence of Krylov subspace methods for matrix Mittag-Leffler functions". In: *SIAM J. Numer. Anal.* 49.5, pp. 2144–2164. ISSN: 0036-1429. DOI: 10.1137/080738374. URL: https://doi.org/10.1137/080738374.
- Samko, S. G. (1995). "Fractional integration and differentiation of variable order". In: *Anal. Math.* 21.3, pp. 213–236. ISSN: 0133-3852. DOI: 10.1007/BF01911126. URL: https://doi.org/10.1007/BF01911126.
- Sokolov, I., A. Chechkin, and J. Klafter (2004). "Distributed-Order Fractional Kinetics". In: *Acta Physica Polonica. Series B* 35.4, pp. 1323–1341.

Bibliography V

West, B. J., M. Turalska, and P. Grigolini (Apr. 2015). "Fractional calculus ties the microscopic and macroscopic scales of complex network dynamics". In: *New Journal of Physics* 17.4, p. 045009. DOI: 10.1088/1367-2630/17/4/045009. URL: https://doi.org/10.1088/1367-2630/17/4/045009.

An introduction to fractional calculus

Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

September, 2022

Fractional Diffusion Equation

Starting from the past...

Fractional Diffusion Equation

Starting from the past...

- We have seen in **Lecture 5** that there is a connection between **diffusion equations** and **random walks**,

Fractional Diffusion Equation

Starting from the past...

- We have seen in **Lecture 5** that there is a connection between **diffusion equations** and **random walks**,
- Given a **particle** we can act either on the jump length or on the waiting time.

Fractional Diffusion Equation

Starting from the past...

- We have seen in **Lecture 5** that there is a connection between **diffusion equations** and **random walks**,
  - Given a **particle** we can act either on the jump length or on the waiting time.

The **C**ontinuous **T**ime **R**andom **W**alk model (CTRW):

- Both the **length of a given jump**, and the **waiting time** elapsing between two successive jumps are drawn from a pdf $\psi(x, t)$

Fractional Diffusion Equation

Starting from the past...

- We have seen in **Lecture 5** that there is a connection between **diffusion equations** and **random walks**,
- Given a **particle** we can act either on the jump length or on the waiting time.

The **C**ontinuous **T**ime **R**andom **W**alk model (CTRW):

- Both the **length of a given jump**, and the **waiting time** elapsing between two successive jumps are drawn from a pdf $\psi(x, t)$
- $\lambda(x) = \int_0^{+\infty} \psi(x, y)\, \mathrm{d}t$, *jump length*,

Jump length

$\lambda(x)\mathrm{d}x$ produces the probability for a jump length in the interval $(x, x + \mathrm{d}x)$.

Fractional Diffusion Equation

Starting from the past...

- We have seen in **Lecture 5** that there is a connection between **diffusion equations** and **random walks**,
- Given a **particle** we can act either on the jump length or on the waiting time.

The **C**ontinuous **T**ime **R**andom **W**alk model (CTRW):

- Both the **length of a given jump**, and the **waiting time** elapsing between two successive jumps are drawn from a pdf $\psi(x,t)$
- $\lambda(x) = \int_0^{+\infty} \psi(x,y)\,\mathrm{d}t$, *jump length*,
- $w(t) = \int_{-\infty}^{+\infty} \psi(x,t)\,\mathrm{d}x$, *waiting time*,

Waiting time

$w(t)\mathrm{d}t$ produces the probability for a waiting time in the interval $(t, t+\mathrm{d}t)$.

Fractional Diffusion Equation

Starting from the past...

- We have seen in **Lecture 5** that there is a connection between **diffusion equations** and **random walks**,
- Given a **particle** we can act either on the jump length or on the waiting time.

The **C**ontinuous **T**ime **R**andom **W**alk model (CTRW):

- Both the **length of a given jump**, and the **waiting time** elapsing between two successive jumps are drawn from a pdf $\psi(x,t)$
- $\lambda(x) = \int_0^{+\infty} \psi(x,y)\,\mathrm{d}t$, *jump length*,
- $w(t) = \int_{-\infty}^{+\infty} \psi(x,t)\,\mathrm{d}x$, *waiting time*,
- If the jump length and waiting time are **independent random variables** then:

$$\psi(x,t) = w(t)\lambda(x).$$

Characterization of CTRW

To categorise different CTRW one can look at the quantities

$$T = \int_0^{+\infty} t w(t)\,\mathrm{d}t, \text{ (Characteristic waiting time)},$$

and

$$\Sigma^2 = \int_{-\infty}^{+\infty} x^2 \lambda(x)\,\mathrm{d}x \text{ (Jump length variance)},$$

specifically, are they **finite**? Do they **diverge**?

Characterization of CTRW

To categorise different CTRW one can look at the quantities

$$T = \int_0^{+\infty} t w(t)\,\mathrm{d}t, \text{ (Characteristic waiting time)},$$

and

$$\Sigma^2 = \int_{-\infty}^{+\infty} x^2 \lambda(x)\,\mathrm{d}x \text{ (Jump length variance)},$$

specifically, are they **finite**? Do they **diverge**?
The master (Langevin) equation for this process is then given by

$$\eta(x,t) = \int_{-\infty}^{+\infty} \mathrm{d}x' \int_0^{+\infty} \mathrm{d}t'\, \eta(x',t')\psi(x-x',t-t') + \delta(x)\delta(t),$$

Characterization of CTRW

To categorise different CTRW one can look at the quantities

$$T = \int_0^{+\infty} tw(t)\,\mathrm{d}t, \text{ (Characteristic waiting time)},$$

and

$$\Sigma^2 = \int_{-\infty}^{+\infty} x^2\lambda(x)\,\mathrm{d}x \text{ (Jump length variance)},$$

specifically, are they **finite**? Do they **diverge**?
The master (Langevin) equation for this process is then given by

$$\eta(x,t) = \int_{-\infty}^{+\infty} \mathrm{d}x' \int_0^{+\infty} \mathrm{d}t'\,\eta(x',t')\psi(x-x',t-t') + \delta(x)\delta(t),$$

Pdf of having arrived at position x at time t – $\eta(x,t)$ –

Characterization of CTRW

To categorise different CTRW one can look at the quantities

$$T = \int_0^{+\infty} tw(t)\,\mathrm{d}t, \text{ (Characteristic waiting time)},$$

and

$$\Sigma^2 = \int_{-\infty}^{+\infty} x^2\lambda(x)\,\mathrm{d}x \text{ (Jump length variance)},$$

specifically, are they **finite**? Do they **diverge**?
The master (Langevin) equation for this process is then given by

$$\eta(x,t) = \int_{-\infty}^{+\infty} \mathrm{d}x' \int_0^{+\infty} \mathrm{d}t'\, \eta(x',t')\psi(x-x',t-t') + \delta(x)\delta(t),$$

Pdf of having arrived at position x at time t – $\eta(x,t)$ – having just arrived at x' at time t' – $\eta(x',t')$ –

Characterization of CTRW

To categorise different CTRW one can look at the quantities

$$T = \int_0^{+\infty} tw(t)\,\mathrm{d}t, \text{ (Characteristic waiting time)},$$

and

$$\Sigma^2 = \int_{-\infty}^{+\infty} x^2\lambda(x)\,\mathrm{d}x \text{ (Jump length variance)},$$

specifically, are they **finite**? Do they **diverge**?
The master (Langevin) equation for this process is then given by

$$\eta(x,t) = \int_{-\infty}^{+\infty} \mathrm{d}x' \int_0^{+\infty} \mathrm{d}t'\,\eta(x',t')\psi(x-x',t-t') + \delta(x)\delta(t),$$

Pdf of having arrived at position x at time t – $\eta(x,t)$ – having just arrived at x' at time t' – $\eta(x',t')$ – with initial condition $\delta(x)$.

Characterization of CTRW

Then if we use

$$\eta(x,t) = \int_{-\infty}^{+\infty} \mathrm{d}x' \int_0^{+\infty} \mathrm{d}t' \, \eta(x',t')\psi(x-x',t-t') + \delta(x)\delta(t),$$

we can write the pdf of being in x at time t as

$$W(x,t) = \int_0^t \eta(x,t')\Psi(t-t'), \mathrm{d}t, \qquad \Psi(t) = 1 - \int_0^t w(t')\,\mathrm{d}t',$$

where the latter is the cumulative probability assigned to the probability of no jump event during the time interval $t - t'$.

Fact I - Ordinary Diffusion

If both T and Σ^2 are finite the long-time limit corresponds to Brownian motion, e.g., $w(t) = \tau^{-1} exp(-t/\tau)$, $T = \tau$, $\lambda(x) = (4\pi\sigma^2)^{-1/2}\exp(-x^2/4\sigma^2)$, $\Sigma^2 = 2\sigma^2$, we recover the standard diffusion equation.

Characterization of CTRW

Then if we use

$$\eta(x,t) = \int_{-\infty}^{+\infty} \mathrm{d}x' \int_0^{+\infty} \mathrm{d}t' \, \eta(x',t')\psi(x-x',t-t') + \delta(x)\delta(t),$$

we can write the pdf of being in x at time t as

$$W(x,t) = \int_0^t \eta(x,t')\Psi(t-t'), \mathrm{d}t, \qquad \Psi(t) = 1 - \int_0^t w(t')\,\mathrm{d}t',$$

where the latter is the cumulative probability assigned to the probability of no jump event during the time interval $t - t'$.

Fact II - Subdifussion

The **characteristic waiting time** $T = \int_0^{+\infty} tw(t)\,\mathrm{d}t$ **diverges**, but the jump length variance $\Sigma^2 = \int_{-\infty}^{+\infty} x^2\lambda(x)\,\mathrm{d}x$ is finite, we obtain a **subdiffusive process**. Particles make long rests.

Long jumps: Lévy Flights

What if we take a **finite waiting time** and a **diverging jump length**?

Long jumps: Lévy Flights

What if we take a **finite waiting time** and a **diverging jump length**?

- Poissonian waiting time,

Long jumps: Lévy Flights

What if we take a **finite waiting time** and a **diverging jump length**?

- Poissonian waiting time,
- Lévy distribution for the jump length

$$\lambda(k) = \exp(-\sigma^{\mu}|k|^{\mu}) \sim 1 - \sigma^{\mu}|k|^{\mu},$$

Long jumps: Lévy Flights

What if we take a **finite waiting time** and a **diverging jump length**?

- Poissonian waiting time,
- Lévy distribution for the jump length

$$\lambda(k) = \exp(-\sigma^\mu |k|^\mu) \sim 1 - \sigma^\mu |k|^\mu,$$

Asymptotic

For $|x| \gg \sigma$, $1 < \mu < 2 \Rightarrow \lambda(x) \sim A_\mu \sigma^{-\mu} |x|^{-1-\mu}$.

Long jumps: Lévy Flights

What if we take a **finite waiting time** and a **diverging jump length**?

- Poissonian waiting time, **Rmk:** T is **finite** and so the process is **Markovian**!
- Lévy distribution for the jump length

$$\lambda(k) = \exp(-\sigma^\mu |k|^\mu) \sim 1 - \sigma^\mu |k|^\mu,$$

Asymptotic

For $|x| \gg \sigma$, $1 < \mu < 2 \Rightarrow \lambda(x) \sim A_\mu \sigma^{-\mu} |x|^{-1-\mu}$.

Long jumps: Lévy Flights

What if we take a **finite waiting time** and a **diverging jump length**?

- Poissonian waiting time, **Rmk:** T is **finite** and so the process is **Markovian**!
- Lévy distribution for the jump length

$$\lambda(k) = \exp(-\sigma^\mu |k|^\mu) \sim 1 - \sigma^\mu |k|^\mu,$$

Asymptotic

For $|x| \gg \sigma$, $1 < \mu < 2 \Rightarrow \lambda(x) \sim A_\mu \sigma^{-\mu} |x|^{-1-\mu}$.

- In the Fourier-Laplace space we get

$$W(k, u) = \frac{1}{u + K^\mu |k|^\mu},$$

Long jumps: Lévy Flights

What if we take a **finite waiting time** and a **diverging jump length**?

- Poissonian waiting time, **Rmk:** T is **finite** and so the process is **Markovian**!
- Lévy distribution for the jump length

$$\lambda(k) = \exp(-\sigma^\mu |k|^\mu) \sim 1 - \sigma^\mu |k|^\mu,$$

Asymptotic

For $|x| \gg \sigma$, $1 < \mu < 2 \Rightarrow \lambda(x) \sim A_\mu \sigma^{-\mu} |x|^{-1-\mu}$.

- In the Fourier-Laplace space we get

$$W(k, u) = \frac{1}{u + K^\mu |k|^\mu},$$

- then after a (double) inversion

$$\frac{\partial W}{\partial t} = K^\mu \cdot \frac{1}{\Gamma(1-\mu)} \frac{\mathrm{d}}{\mathrm{d}x} \int_{-\infty}^{x} W(\xi, t)(x-\xi)^\alpha \, \mathrm{d}\xi, \quad K = \frac{\sigma^\mu}{\tau}$$

Long jumps: Lévy Flights

What if we take a **finite waiting time** and a **diverging jump length**?

- Poissonian waiting time, **Rmk:** T is **finite** and so the process is **Markovian**!
- Lévy distribution for the jump length

$$\lambda(k) = \exp(-\sigma^\mu |k|^\mu) \sim 1 - \sigma^\mu |k|^\mu,$$

Asymptotic

For $|x| \gg \sigma$, $1 < \mu < 2 \Rightarrow \lambda(x) \sim A_\mu \sigma^{-\mu} |x|^{-1-\mu}$.

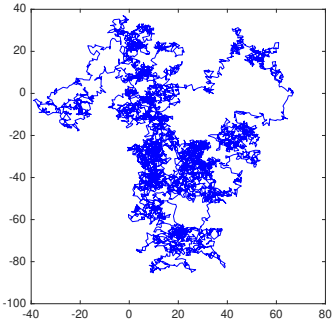
- In the Fourier-Laplace space we get

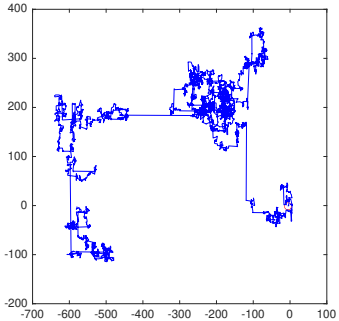
$$W(k, u) = \frac{1}{u + K^\mu |k|^\mu},$$

- then after a (double) inversion

$$\frac{\partial W}{\partial t} = K^\mu \, {}^{RL}D^\mu_{(-\infty, x)} W(x, t), \quad K = \frac{\sigma^\mu}{\tau}$$

Brownian jumps vs Lévy Flights






```
%% Brownian motion
N = 7000;
x = cumsum(randn(N,1));
y = cumsum(randn(N,1));
```

```
%% Levy distribution
N = 7000;
pd_levy = makedist('Stable','alpha',1.5,
↪  'beta',0,'gam',1, 'delta',0);
xl = cumsum(random(pd_levy,N,1));
yl = cumsum(random(pd_levy,N,1));
```

The (Space) Fractional Diffusion Equations

We want to solve our problem in a **domain of finite size**, therefore we have to move the lower and upper bounds of the Riemann-Liouville integral to a finite domain size and **select** some **boundary conditions**.

The (Space) Fractional Diffusion Equations

We want to solve our problem in a **domain of finite size**, therefore we have to move the lower and upper bounds of the Riemann-Liouville integral to a finite domain size and **select** some **boundary conditions**.

Absorbing boundary conditions (Dirichlet)

A common choice is given by: $W(x_l, t) = W(x_r, T) \equiv 0$

The (Space) Fractional Diffusion Equations

We want to solve our problem in a **domain of finite size**, therefore we have to move the lower and upper bounds of the Riemann-Liouville integral to a finite domain size and **select** some **boundary conditions**.

Absorbing boundary conditions (Dirichlet)

A common choice is given by: $W(x_l, t) = W(x_r, T) \equiv 0$

They can be justified in various way

The (Space) Fractional Diffusion Equations

We want to solve our problem in a **domain of finite size**, therefore we have to move the lower and upper bounds of the Riemann-Liouville integral to a finite domain size and **select** some **boundary conditions**.

Absorbing boundary conditions (Dirichlet)

A common choice is given by: $W(x_l, t) = W(x_r, T) \equiv 0$

They can be justified in various way

- Variational formulation from a generalized *Fickian* law (Jin et al. 2015),

The (Space) Fractional Diffusion Equations

We want to solve our problem in a **domain of finite size**, therefore we have to move the lower and upper bounds of the Riemann-Liouville integral to a finite domain size and **select** some **boundary conditions**.

Absorbing boundary conditions (Dirichlet)

A common choice is given by: $W(x_l, t) = W(x_r, T) \equiv 0$

They can be justified in various way

- Variational formulation from a generalized *Fickian* law (Jin et al. 2015),
- Lyapunov inequality (Ferreira 2013).

The (Space) Fractional Diffusion Equations

We want to solve our problem in a **domain of finite size**, therefore we have to move the lower and upper bounds of the Riemann-Liouville integral to a finite domain size and **select** some **boundary conditions**.

Absorbing boundary conditions (Dirichlet)

A common choice is given by: $W(x_l, t) = W(x_r, T) \equiv 0$

They can be justified in various way

- Variational formulation from a generalized *Fickian* law (Jin et al. 2015),
- Lyapunov inequality (Ferreira 2013).

$$\begin{cases} \frac{\partial W}{\partial t} = \theta\, {}^{RL}D^{\alpha}_{[0,x]} W(x,t) + (1-\theta)\, {}^{RL}D^{\alpha}_{[x,1]} W(x,t), & \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \\ W(x,t) = W_0(x). \end{cases} \tag{FDE$_1$}$$

Finite Difference Approaches to Riemann–Liouville

The **first approach** we want to discuss is **finite differences**, thus how can we discretize the Riemann-Liouville operators?

Finite Difference Approaches to Riemann–Liouville

The **first approach** we want to discuss is **finite differences**, thus how can we discretize the Riemann-Liouville operators?

Back to the basics

1. First derivative

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{h \to 0} \frac{f(x) - f(x-h)}{h},$$

Finite Difference Approaches to Riemann–Liouville

The **first approach** we want to discuss is **finite differences**, thus how can we discretize the Riemann-Liouville operators?

Back to the basics

1. First derivative

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{h\to 0} \frac{f(x) - f(x-h)}{h},$$

2. nth derivative

$$\frac{\mathrm{d}^n f}{\mathrm{d}x^n} = \lim_{h\to 0} \frac{\Delta^n f(x)}{h}, \quad \Delta^n f(x) = \sum_{j=0}^{n} \binom{n}{j} (-1)^j f(x - jh).$$

Finite Difference Approaches to Riemann–Liouville

The **first approach** we want to discuss is **finite differences**, thus how can we discretize the Riemann-Liouville operators?

Back to the basics

1. First derivative

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \lim_{h\to 0} \frac{f(x) - f(x-h)}{h},$$

2. nth derivative

$$\frac{\mathrm{d}^n f}{\mathrm{d}x^n} = \lim_{h\to 0} \frac{\Delta^n f(x)}{h}, \quad \Delta^n f(x) = \sum_{j=0}^{n} \binom{n}{j} (-1)^j f(x - jh).$$

Let's use again our favourite trick and replace $n \in \mathbb{N}$ with $\alpha \in \mathbb{R}$!

The Grünwald–Letnikov Fractional Derivative

The Grünwald–Letnikov Fractional Derivative (Grünwald 1867; Letnikov 1868)

Given $\mathbb{R} \ni \alpha > 0$ define the Grünwald–Letnikov fractional derivative of a function $f(x)$ as

$$ {}^{GL}D^{\alpha} f = \lim_{h \to 0} \frac{\Delta^{\alpha} f(x)}{h}, \quad \Delta^{\alpha} f(x) = \sum_{j=0}^{+\infty} \binom{\alpha}{j} (-1)^j f(x - jh), \quad \binom{\alpha}{j} = \frac{\Gamma(\alpha+1)}{j!\Gamma(\alpha - j + 1)}. $$

The Grünwald–Letnikov Fractional Derivative

The Grünwald–Letnikov Fractional Derivative (Grünwald 1867; Letnikov 1868)

Given $\mathbb{R} \ni \alpha > 0$ define the Grünwald–Letnikov fractional derivative of a function $f(x)$ as

$$ {}^{GL}D^{\alpha} f = \lim_{h \to 0} \frac{\Delta^{\alpha} f(x)}{h}, \quad \Delta^{\alpha} f(x) = \sum_{j=0}^{+\infty} \binom{\alpha}{j} (-1)^j f(x - jh), \quad \binom{\alpha}{j} = \frac{\Gamma(\alpha+1)}{j!\Gamma(\alpha - j + 1)}. $$

? For what functions f does it make sense?

The Grünwald–Letnikov Fractional Derivative

The Grünwald–Letnikov Fractional Derivative (Grünwald 1867; Letnikov 1868)

Given $\mathbb{R} \ni \alpha > 0$ define the Grünwald–Letnikov fractional derivative of a function $f(x)$ as

$$ {}^{GL}D^{\alpha} f = \lim_{h \to 0} \frac{\Delta^{\alpha} f(x)}{h}, \quad \Delta^{\alpha} f(x) = \sum_{j=0}^{+\infty} \binom{\alpha}{j} (-1)^j f(x - jh), \quad \binom{\alpha}{j} = \frac{\Gamma(\alpha+1)}{j!\Gamma(\alpha - j + 1)}. $$

- For what functions f does it make sense?
- How is it related to the Riemann-Liouville (and henceforth to the Caputo) fractional derivative?

The Grünwald–Letnikov Fractional Derivative

The Grünwald–Letnikov Fractional Derivative (Grünwald 1867; Letnikov 1868)

Given $\mathbb{R} \ni \alpha > 0$ define the Grünwald–Letnikov fractional derivative of a function $f(x)$ as

$$ {}^{GL}D^{\alpha} f = \lim_{h \to 0} \frac{\Delta^{\alpha} f(x)}{h}, \quad \Delta^{\alpha} f(x) = \sum_{j=0}^{+\infty} \binom{\alpha}{j} (-1)^j f(x - jh), \quad \binom{\alpha}{j} = \frac{\Gamma(\alpha+1)}{j!\Gamma(\alpha - j + 1)}. $$

- For what functions f does it make sense?
- How is it related to the Riemann-Liouville (and henceforth to the Caputo) fractional derivative?
- If we can find an easy relation with the Riemann-Liouville derivative we can **use it to discretize** by truncating Δ^{α} to a given N.

The Grünwald–Letnikov Fractional Derivative

Let us collect the ingredients we need.

- The binomial series

$$(1+z)^\alpha = \sum_{j=0}^{+\infty} \binom{\alpha}{j} z^j,$$

  converges for any $z \in \mathbb{C}$ with $|z| \leq 1$ and any $\alpha > 0$,

- The series

$$\sum_{j=0}^{+\infty} \left| \binom{\alpha}{j} (-1)^j \right| < +\infty,$$

  converges, since $(1+(-1))^\alpha = 0$.

$\Rightarrow$ If we take f to be bounded then ${}^{GL}D^\alpha f$ exists.

The Grünwald–Letnikov Fractional Derivative

Let us take the Fourier transform of $\Delta^\alpha f(x)$

$$\begin{aligned}\int e^{-ikx} \sum_{j=0}^{+\infty} \binom{\alpha}{j} (-1)^j f(x-jh)\,\mathrm{d}x &= \sum_{j=0}^{+\infty} \binom{\alpha}{j} (-1)^j \int e^{-ikx} f(x-jh)\,\mathrm{d}x \\ &= \sum_{j=0}^{\infty} \binom{\alpha}{j} (-1)^j e^{-ikjh} \hat{f}(k) \\ &= (1-e^{-ikh})^\alpha \hat{f}(k).\end{aligned}$$

- We are using the **uniform convergence** of the series $\Delta^\alpha f(x)$,
- furthermore we are **requiring** that each term is integrable.

The Grünwald–Letnikov Fractional Derivative

Let us take the Fourier transform of $\Delta^\alpha f(x)$

$$\int e^{-ikx} \sum_{j=0}^{+\infty} \binom{\alpha}{j} (-1)^j f(x - jh)\, \mathrm{d}x = (1 - e^{-ikh})^\alpha \hat{f}(k).$$

- We are using the **uniform convergence** of the series $\Delta^\alpha f(x)$,
- furthermore we are **requiring** that each term is integrable.

If $k \neq 0$ then the Fourier transform of the GL derivative operator is given by

$$h^{-\alpha}(ikh)^\alpha \left(\frac{1 - e^{-ikh}}{ikh} \right) \hat{f}(k) \to (ik)^\alpha \hat{f}(k), \text{ for } h \to 0.$$

The same holds by direct computation for $k = 0$.

The Grünwald–Letnikov Fractional Derivative

Let us take the Fourier transform of $\Delta^\alpha f(x)$

$$\int e^{-ikx} \sum_{j=0}^{+\infty} \binom{\alpha}{j} (-1)^j f(x - jh)\, \mathrm{d}x = (1 - e^{-ikh})^\alpha \hat{f}(k).$$

- We are using the **uniform convergence** of the series $\Delta^\alpha f(x)$,
- furthermore we are **requiring** that each term is integrable.

If $k \neq 0$ then the Fourier transform of the GL derivative operator is given by

$$h^{-\alpha}(ikh)^\alpha \left(\frac{1 - e^{-ikh}}{ikh} \right) \hat{f}(k) \to (ik)^\alpha \hat{f}(k), \text{ for } h \to 0.$$

The same holds by direct computation for $k = 0$.

$\Rightarrow$ The Fourier transform **converges pointwise** to the same **Fourier transform** of the **Riemann-Liouville** derivative (we are also using the continuity Theorem of Fourier transform.)

The Grünwald–Letnikov Fractional Derivative

What is the connection then?

The Grünwald–Letnikov Fractional Derivative

What is the connection then?

1. Let us look better into the *weights*

$$
\begin{aligned}
g_j^{(\alpha)} \triangleq (-1)^j \binom{\alpha}{j} &= \frac{(-1)^j \Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)} = \\
&= \frac{(-1)^j \alpha(\alpha-1)\cdots(\alpha-j+1)}{\Gamma(j+1)} \\
\text{Distribute } (-1)^j \to &= \frac{(-\alpha)(1-\alpha)\cdot(j-1-\alpha)}{\Gamma(j+1)} \\
&= \frac{-\alpha\Gamma(j-\alpha)}{\Gamma(j+1)\Gamma(1-\alpha)}
\end{aligned}
$$

The Grünwald–Letnikov Fractional Derivative

What is the connection then?

1. Let us look better into the *weights*

$$g_j^{(\alpha)} \triangleq (-1)^j \binom{\alpha}{j} = \frac{-\alpha\Gamma(j-\alpha)}{\Gamma(j+1)\Gamma(1-\alpha)}$$

2. Using $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(x+1) \sim \sqrt{2\pi x}x^x e^{-x}$ for $x \to +\infty$

$$\begin{aligned} g_j^{(\alpha)} &\sim \frac{-\alpha}{\Gamma(1-\alpha)} \frac{\sqrt{2\pi(j-\alpha-1)}(j-\alpha-1)^{j-\alpha-1}e^{-(j-\alpha-1)}}{\sqrt{2\pi j}j^j e^{-j}} \\ &= \frac{-\alpha}{\Gamma(1-\alpha)} \underbrace{\sqrt{\frac{j-\alpha-1}{j}}}_{\to 1} \underbrace{\left(\frac{j-\alpha-1}{j}\right)^{j-\alpha-1}}_{\to e^{-(\alpha+1)}} j^{-\alpha-1}e^{\alpha+1}j^{-\alpha-1} \qquad j \to +\infty. \end{aligned}$$

The Grünwald–Letnikov Fractional Derivative

What is the connection then?

1. Let us look better into the *weights*

$$g_j^{(\alpha)} \triangleq (-1)^j \binom{\alpha}{j} = \frac{-\alpha\Gamma(j-\alpha)}{\Gamma(j+1)\Gamma(1-\alpha)}$$

2. Using $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(x+1) \sim \sqrt{2\pi x}x^x e^{-x}$ for $x \to +\infty$

$$g_j^{(\alpha)} \sim \frac{-\alpha}{\Gamma(1-\alpha)} j^{-\alpha-1} \qquad j \to +\infty.$$

The Grünwald–Letnikov Fractional Derivative

What is the connection then?

1. Let us look better into the *weights*

$$g_j^{(\alpha)} \triangleq (-1)^j \binom{\alpha}{j} = \frac{-\alpha\Gamma(j-\alpha)}{\Gamma(j+1)\Gamma(1-\alpha)}$$

2. Using $\Gamma(x+1) = x\Gamma(x)$ and $\Gamma(x+1) \sim \sqrt{2\pi x}x^x e^{-x}$ for $x \to +\infty$

$$g_j^{(\alpha)} \sim \frac{-\alpha}{\Gamma(1-\alpha)} j^{-\alpha-1} \qquad j \to +\infty.$$

3. Since $g_0^{(\alpha)} = 1$ we write the quotient

$$\frac{\Delta^\alpha f(x)}{\Delta x^\alpha} = (\Delta x)^{-\alpha} \left[f(x) + \sum_{j=1}^{+\infty} g_j^{(\alpha)} f(x - j\Delta x) \right]$$

The Grünwald–Letnikov Fractional Derivative

What is the connection then?

3. Since $g_0^{(\alpha)} = 1$ we write the quotient

$$\frac{\Delta^\alpha f(x)}{\Delta x^\alpha} = (\Delta x)^{-\alpha} \left[f(x) + \sum_{j=1}^{+\infty} g_j^{(\alpha)} f(x - j\Delta x) \right]$$

The Grünwald–Letnikov Fractional Derivative

What is the connection then?

3. Since $g_0^{(\alpha)} = 1$ we write the quotient

$$\frac{\Delta^\alpha f(x)}{\Delta x^\alpha} = (\Delta x)^{-\alpha} \left[f(x) + \sum_{j=1}^{+\infty} g_j^{(\alpha)} f(x - j\Delta x) \right]$$

4. $\sum_{j=0}^{+\infty} w_j = 0$. Then $g_j^{(\alpha)} < 0$ for all $j \geq 1$ and thus $\sum_{j=1}^{+\infty} g_j^{(\alpha)} = -1$. We **define** $b_j^{(\alpha)} = -w_j^{(\alpha)}$ for $j \geq 1$, so that

$$b_j \sim \frac{\alpha}{\Gamma(1-\alpha)} j^{-\alpha-1} \text{ for } j \to +\infty, \qquad \sum_{j=1}^{+\infty} b_j = 1.$$

The Grünwald–Letnikov Fractional Derivative

Then we take $0 < \alpha < 1$

$$
\begin{aligned}
\frac{\Delta^\alpha f(x)}{\Delta x^\alpha} &= (\Delta x)^{-\alpha} \sum_{j=1}^{+\infty} [f(x) - f(x - j\Delta x)] b_j \\
&\approx \sum_{j=1}^{+\infty} [f(x) - f(x - j\Delta x)] \frac{\alpha}{\Gamma(1-\alpha)} (j\Delta x)^{-\alpha-1} \Delta x \\
&\approx \int_0^{+\infty} [f(x) - f(x - y)] \frac{\alpha}{\Gamma(1-\alpha)} y^{-\alpha-1} \, \mathrm{d}y
\end{aligned}
$$

The Grünwald–Letnikov Fractional Derivative

Then we take $0 < \alpha < 1$

$$\begin{aligned}\frac{\Delta^\alpha f(x)}{\Delta x^\alpha} &= (\Delta x)^{-\alpha} \sum_{j=1}^{+\infty} [f(x) - f(x - j\Delta x)] b_j \\ &\approx \sum_{j=1}^{+\infty} [f(x) - f(x - j\Delta x)] \frac{\alpha}{\Gamma(1-\alpha)} (j\Delta x)^{-\alpha-1} \Delta x \\ &\approx \int_0^{+\infty} [f(x) - f(x-y)] \frac{\alpha}{\Gamma(1-\alpha)} y^{-\alpha-1} \, \mathrm{d}y\end{aligned}$$

- Integrate by parts with $u = f(x) - f(x-y)$

$$\frac{1}{\Gamma(1-\alpha)} \int_0^{+\infty} f'(x-y) y^{-\alpha} \, \mathrm{d}y = \frac{1}{\Gamma(1-\alpha)} \int_0^{+\infty} \frac{\mathrm{d}}{\mathrm{d}x} f(x-y) y^{-\alpha} \, \mathrm{d}y$$

The Grünwald–Letnikov Fractional Derivative

Then we take $0 < \alpha < 1$

$$\begin{aligned}\frac{\Delta^\alpha f(x)}{\Delta x^\alpha} &= (\Delta x)^{-\alpha} \sum_{j=1}^{+\infty} [f(x) - f(x - j\Delta x)] b_j \\ &\approx \sum_{j=1}^{+\infty} [f(x) - f(x - j\Delta x)] \frac{\alpha}{\Gamma(1-\alpha)} (j\Delta x)^{-\alpha-1} \Delta x \\ &\approx \int_0^{+\infty} [f(x) - f(x-y)] \frac{\alpha}{\Gamma(1-\alpha)} y^{-\alpha-1} \, \mathrm{d}y\end{aligned}$$

- Integrate by parts with $u = f(x) - f(x-y)$

$$^{CA}D^\alpha_{[0,+\infty]} f(x) = \frac{1}{\Gamma(1-\alpha)} \int_0^{+\infty} f'(x-y) y^{-\alpha} \, \mathrm{d}y = \frac{1}{\Gamma(1-\alpha)} \int_0^{+\infty} \frac{\mathrm{d}}{\mathrm{d}x} f(x-y) y^{-\alpha} \, \mathrm{d}y$$

The Grünwald–Letnikov Fractional Derivative

Then we take $0 < \alpha < 1$

$$\begin{aligned}\frac{\Delta^\alpha f(x)}{\Delta x^\alpha} &= (\Delta x)^{-\alpha} \sum_{j=1}^{+\infty} [f(x) - f(x - j\Delta x)] b_j \\ &\approx \sum_{j=1}^{+\infty} [f(x) - f(x - j\Delta x)] \frac{\alpha}{\Gamma(1-\alpha)} (j\Delta x)^{-\alpha-1} \Delta x \\ &\approx \int_0^{+\infty} [f(x) - f(x-y)] \frac{\alpha}{\Gamma(1-\alpha)} y^{-\alpha-1} \, \mathrm{d}y\end{aligned}$$

- Integrate by parts with $u = f(x) - f(x-y)$... and when you swap the integral and the derivative

$$^{RL}D^\alpha_{[0,+\infty]} = \frac{1}{\Gamma(1-\alpha)} \frac{\mathrm{d}}{\mathrm{d}x} \int_0^{+\infty} f(x-y) y^{-\alpha} \, \mathrm{d}y.$$

The Grünwald–Letnikov Fractional Derivative

Let us move everything to a fixed interval $[a, b]$.

Grünwald–Letnikov *revisited*

Let $\alpha > 0$, $f \in \mathcal{C}^{\lceil \alpha \rceil}([a, b])$, $a < x \leq b$, then

$$ {}^{GL}D_{[a,x]} f(x) = \lim_{N \to +\infty} \frac{\Delta_{h_N}^{\alpha} f(x)}{h_N^{\alpha}} = \lim_{N \to +\infty} \frac{1}{h_N^{\alpha}} \sum_{k=0}^{N} (-1)^k \binom{\alpha}{k} f(x - k h_N), $$

with $h_N = (x - a)/N$.

In the definition we have implicitly extended f (with an abuse of notation) in such a way that

$$ f : (-\infty, b] \to \mathbb{R}, \qquad x \mapsto \begin{cases} f(x), & \text{if } x \in [a, b], \\ 0, & \text{if } x \in (-\infty, a). \end{cases} $$

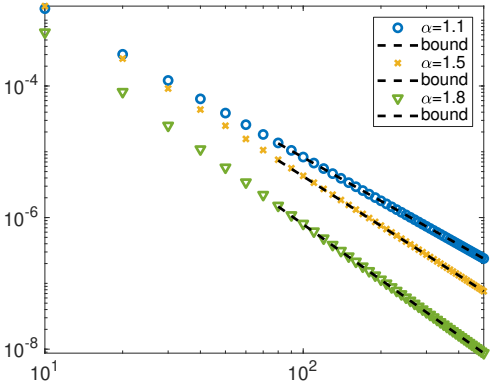
Computing the coefficients

We can compute $N+1$ $g_j^{(\alpha)}$ coefficients in $3N+1$ flops by using the recurrence relation

$$g_j^{(\alpha)} = \left(1 - \frac{\alpha+1}{j}\right) g_{j-1}^{\alpha}, \quad g_0 = 1.$$

In a line of code

```
function [g] = gl(n,alpha)
%GL Produces the N+1 Grunwald-Letnikov
↪  coefficients for a given alpha
g = cumprod([1, 1 - ((alpha+1) ./
↪  (1:n))]);
end
```



A finite difference discretization

Before going to the two-sided case in (FDE_1), let us start with the simpler case

$$\frac{\partial w}{\partial t} = -v(x)\frac{\partial w}{\partial x} + d(x)\,{}^{RL}D^{\alpha}_{[0,x]}w + f(x,t), \qquad 1 < \alpha \leq 2,\ v(x), d(x) \geq 0.$$

A finite difference discretization

Before going to the two-sided case in (FDE_1), let us start with the simpler case

$$\frac{\partial w}{\partial t} = -v(x)\frac{\partial w}{\partial x} + d(x)\,{}^{RL}D^{\alpha}_{[0,x]}w + f(x,t), \qquad 1 < \alpha \leq 2,\ v(x), d(x) \geq 0.$$

1. Substitute the Riemann-Liouville derivative with the Grünwald–Letnikov one,

$$\frac{\partial w}{\partial t} = -v(x)\frac{\partial w}{\partial x} + d(x)\,{}^{GL}D^{\alpha}_{[0,x]}w + f(x,t),$$

A finite difference discretization

Before going to the two-sided case in (FDE_1), let us start with the simpler case

$$\frac{\partial w}{\partial t} = -v(x)\frac{\partial w}{\partial x} + d(x)\,{}^{RL}D^{\alpha}_{[0,x]}w + f(x,t), \qquad 1 < \alpha \leq 2,\ v(x), d(x) \geq 0.$$

1. Substitute the Riemann-Liouville derivative with the Grünwald–Letnikov one,

$$\frac{\partial w}{\partial t} = -v(x)\frac{\partial w}{\partial x} + d(x)\,{}^{GL}D^{\alpha}_{[0,x]}w + f(x,t),$$

2. Choose $N \in \mathbb{N}$ at which to truncate the series expansions

$$\frac{\partial w_i}{\partial t} = -v_i\frac{w_i - w_{i-1}}{h_N} + \frac{d_i}{h_N^{\alpha}}\sum_{k=0}^{i}(-1)^k\binom{\alpha}{k}w_{i-k} + f_i,$$

A finite difference discretization

Before going to the two-sided case in (FDE_1), let us start with the simpler case

$$\frac{\partial w}{\partial t} = -v(x)\frac{\partial w}{\partial x} + d(x)\,{}^{RL}D^{\alpha}_{[0,x]}w + f(x,t), \qquad 1 < \alpha \leq 2,\ v(x), d(x) \geq 0.$$

1. Substitute the Riemann-Liouville derivative with the Grünwald–Letnikov one,

$$\frac{\partial w}{\partial t} = -v(x)\frac{\partial w}{\partial x} + d(x)\,{}^{GL}D^{\alpha}_{[0,x]}w + f(x,t),$$

2. Choose $N \in \mathbb{N}$ at which to truncate the series expansions

$$\frac{\partial w_i}{\partial t} = -v_i\frac{w_i - w_{i-1}}{h_N} + \frac{d_i}{h_N^{\alpha}}\sum_{k=0}^{i}(-1)^k\binom{\alpha}{k}w_{i-k} + f_i,$$

3. Now we need to select a scheme for discretizing it in time: *explicit*? *implicit*?

A finite difference discretization: explicit Euler

Let us select **explicit Euler**

$$\frac{w_i^{n+1} - w_i^n}{\Delta t} = -v_i \frac{w_i^n - w_{i-1}^n}{h_N} + \frac{d_i}{h_N^\alpha} \sum_{k=0}^{i} (-1)^k \binom{\alpha}{k} w_{i-k}^n + f_i^n,$$

A finite difference discretization: explicit Euler

Let us select **explicit Euler**

$$\frac{w_i^{n+1} - w_i^n}{\Delta t} = -v_i \frac{w_i^n - w_{i-1}^n}{h_N} + \frac{d_i}{h_N^\alpha} \sum_{k=0}^{i} g_k w_{i-k}^n + f_i^n,$$

- For convenience we call $g_k = (-1)^k \binom{\alpha}{k}$,

A finite difference discretization: explicit Euler

Let us select **explicit Euler**

$$w_i^{n+1} = w_i^n - \Delta t\, v_i \frac{w_i^n - w_{i-1}^n}{h_N} + \Delta t\, \frac{d_i}{h_N^\alpha} \sum_{k=0}^{i} g_k\, w_{i-k}^n + f_i^n,$$

- For convenience we call $g_k = (-1)^k \binom{\alpha}{k}$,
- Rearrange everything to compute w_i^{n+1}

A finite difference discretization: explicit Euler

Let us select **explicit Euler**

$$w_i^{n+1} = \left(1 - \frac{\Delta t}{h_N} v_i + \frac{\Delta t}{h_N^\alpha} d_i\right) w_i^n + \left(\frac{v_i}{h_N} - \frac{\alpha}{h_N^\alpha} d_i\right) \Delta t w_{i-1}^n + \frac{d_i \Delta t}{h_N^\alpha} \sum_{k=2}^{i} g_k w_{i-k}^n + f_i^n \Delta t,$$

- For convenience we call $g_k = (-1)^k \binom{\alpha}{k}$,
- Rearrange everything to compute w_i^{n+1}, and using that $g_0 = 1$, $g_1 = -\alpha$

A finite difference discretization: explicit Euler

Let us select **explicit Euler**

$$w_i^{n+1} = \left(1 - \frac{\Delta t}{h_N} v_i + \frac{\Delta t}{h_N^\alpha} d_i\right) w_i^n + \left(\frac{v_i}{h_N} - \frac{\alpha}{h_N^\alpha} d_i\right) \Delta t w_{i-1}^n + \frac{d_i \Delta t}{h_N^\alpha} \sum_{k=2}^{i} g_k w_{i-k}^n + f_i^n \Delta t,$$

- For convenience we call $g_k = (-1)^k \binom{\alpha}{k}$,
- Rearrange everything to compute w_i^{n+1}, and using that $g_0 = 1$, $g_1 = -\alpha$
- Is this *stable*? Do we have to put a restriction on the choice of h_N and Δt?

A finite difference discretization: explicit Euler

Let us select **explicit Euler**

$$w_i^{n+1} = \left(1 - \frac{\Delta t}{h_N} v_i + \frac{\Delta t}{h_N^\alpha} d_i\right) w_i^n + \left(\frac{v_i}{h_N} - \frac{\alpha}{h_N^\alpha} d_i\right) \Delta t w_{i-1}^n + \frac{d_i \Delta t}{h_N^\alpha} \sum_{k=2}^{i} g_k w_{i-k}^n + f_i^n \Delta t,$$

- For convenience we call $g_k = (-1)^k \binom{\alpha}{k}$,
- Rearrange everything to compute w_i^{n+1}, and using that $g_0 = 1$, $g_1 = -\alpha$
- Is this *stable*? Do we have to put a restriction on the choice of h_N and Δt?
- Suppose that w_i^0 is affected by an error, i.e., $\widehat{w}_i^0 = w_i^0 + \epsilon_i^0$, we can then look at the **propagation of the error**,

A finite difference discretization: explicit Euler

Let us select **explicit Euler**

$$\widehat{w}_i^1 = \left(1 - \frac{\Delta t}{h_N} v_i + \frac{\Delta t}{h_N^\alpha} d_i\right) \widehat{w}_i^0 + \left(\frac{v_i}{h_N} - \frac{\alpha}{h_N^\alpha} d_i\right) \Delta t w_{i-1}^n + \frac{d_i \Delta t}{h_N^\alpha} \sum_{k=2}^{i} g_k w_{i-k}^n + f_i^n \Delta t,$$

- For convenience we call $g_k = (-1)^k \binom{\alpha}{k}$,
- Rearrange everything to compute w_i^{n+1}, and using that $g_0 = 1$, $g_1 = -\alpha$
- Is this *stable*? Do we have to put a restriction on the choice of h_N and Δt?
- Suppose that w_i^0 is affected by an error, i.e., $\widehat{w}_i^0 = w_i^0 + \epsilon_i^0$, we can then look at the **propagation of the error**,
- We call $\mu_i = 1 - {}^{\Delta t}/_{h_N} v_i + {}^{\Delta t}/_{h_N^\alpha} d_i$

A finite difference discretization: explicit Euler

Let us select **explicit Euler**

$$\widehat{w}_i^1 = \mu_i \widehat{w}_i^0 + \left(\frac{v_i}{h_N} - \frac{\alpha}{h_N^\alpha} d_i \right) \Delta t w_{i-1}^n + \frac{d_i \Delta t}{h_N^\alpha} \sum_{k=2}^{i} g_k w_{i-k}^n + f_i^n \Delta t,$$

- For convenience we call $g_k = (-1)^k \binom{\alpha}{k}$,
- Rearrange everything to compute w_i^{n+1}, and using that $g_0 = 1$, $g_1 = -\alpha$
- Is this *stable*? Do we have to put a restriction on the choice of h_N and Δt?
- Suppose that w_i^0 is affected by an error, i.e., $\widehat{w}_i^0 = w_i^0 + \epsilon_i^0$, we can then look at the **propagation of the error**,
- We call $\mu_i = 1 - {}^{\Delta t}/_{h_N} v_i + {}^{\Delta t}/_{h_N^\alpha} d_i$

A finite difference discretization: explicit Euler

Let us select **explicit Euler**

$$\widehat{w}_i^1 = \mu_i \epsilon_i^0 + c_i^1,$$

- For convenience we call $g_k = (-1)^k \binom{\alpha}{k}$,
- Rearrange everything to compute w_i^{n+1}, and using that $g_0 = 1$, $g_1 = -\alpha$
- Is this *stable*? Do we have to put a restriction on the choice of h_N and Δt?
- Suppose that w_i^0 is affected by an error, i.e., $\widehat{w}_i^0 = w_i^0 + \epsilon_i^0$, we can then look at the **propagation of the error**,
- We call $\mu_i = 1 - {}^{\Delta t}/_{h_N} v_i + {}^{\Delta t}/_{h_N^\alpha} d_i$ and get the expression for the new error.

A finite difference discretization: explicit Euler

Let us select **explicit Euler**

$$\widehat{w}_i^1 = \mu_i \epsilon_i^0 + c_i^1,$$

- For convenience we call $g_k = (-1)^k \binom{\alpha}{k}$,
- Rearrange everything to compute w_i^{n+1}, and using that $g_0 = 1$, $g_1 = -\alpha$
- Is this *stable*? Do we have to put a restriction on the choice of h_N and Δt?
- Suppose that w_i^0 is affected by an error, i.e., $\widehat{w}_i^0 = w_i^0 + \epsilon_i^0$, we can then look at the **propagation of the error**,
- We call $\mu_i = 1 - {}^{\Delta t}/_{h_N} v_i + {}^{\Delta t}/_{h_N^\alpha} d_i$ and get the expression for the new error.
- By iterating the argument we found that the error at step n is amplified by the factor μ_i, that is

$$\epsilon_i^n = \mu_i^n \epsilon_i^0.$$

A finite difference discretization: explicit Euler

Let us select **explicit Euler**

$$\widehat{w}_i^1 = \mu_i \epsilon_i^0 + c_i^1,$$

- For convenience we call $g_k = (-1)^k \binom{\alpha}{k}$,
- Rearrange everything to compute w_i^{n+1}, and using that $g_0 = 1$, $g_1 = -\alpha$
- Is this *stable*? Do we have to put a restriction on the choice of h_N and Δt?
- Suppose that w_i^0 is affected by an error, i.e., $\widehat{w}_i^0 = w_i^0 + \epsilon_i^0$, we can then look at the **propagation of the error**,
- We call $\mu_i = 1 - {}^{\Delta t}/_{h_N} v_i + {}^{\Delta t}/_{h_N^\alpha} d_i$ and get the expression for the new error.
- By iterating the argument we found that the error at step n is amplified by the factor μ_i, that is

$$\epsilon_i^n = \mu_i^n \epsilon_i^0.$$

- To have stability we need to require that exist h_N such that $|\mu_i| < 1$ for all $h < h_N$.

A finite difference discretization: explicit Euler

$$\mu_i \equiv 1 - \frac{\Delta t}{h_N} v_i + \frac{\Delta t}{h_N^\alpha} d_i < 1 \;\Leftrightarrow\; h_N > \left(\frac{d_i}{v_i}\right)^{1/\alpha - 1}$$

⚠ The method is not stable as h is refined!

A finite difference discretization: explicit Euler

$$\mu_i \equiv 1 - \frac{\Delta t}{h_N} v_i + \frac{\Delta t}{h_N^\alpha} d_i < 1 \;\Leftrightarrow\; h_N > \left(\frac{d_i}{v_i}\right)^{1/\alpha - 1}$$

⚠ The method is not stable as h is refined!

Theorem (Meerschaert and Tadjeran 2004)

The **explicit** Euler solution method based on the Grünwald–Letnikov approximation of the Riemann-Liouville fractional derivative is unstable.

A finite difference discretization: implicit Euler

$$\mu_i \equiv 1 - \frac{\Delta t}{h_N} v_i + \frac{\Delta t}{h_N^\alpha} d_i < 1 \;\Leftrightarrow\; h_N > \left(\frac{d_i}{v_i}\right)^{1/\alpha - 1}$$

⚠ The method is not stable as h is refined!

Theorem (Meerschaert and Tadjeran 2004)

The **explicit** Euler solution method based on the Grünwald–Letnikov approximation of the Riemann-Liouville fractional derivative is unstable.

Theorem (Meerschaert and Tadjeran 2004)

The **implicit** Euler solution method based on the Grünwald–Letnikov approximation of the Riemann-Liouville fractional derivative is unstable.

A finite difference discretization: ex/implicit Euler

$$\mu_i \equiv 1 - \frac{\Delta t}{h_N} v_i + \frac{\Delta t}{h_N^\alpha} d_i < 1 \;\Leftrightarrow\; h_N > \left(\frac{d_i}{v_i}\right)^{1/\alpha - 1}$$

⚠ The method is not stable as h is refined!

Theorem (Meerschaert and Tadjeran 2004)

The **explicit** Euler solution method based on the Grünwald–Letnikov approximation of the Riemann-Liouville fractional derivative is unstable.

Theorem (Meerschaert and Tadjeran 2004)

The **implicit** Euler solution method based on the Grünwald–Letnikov approximation of the Riemann-Liouville fractional derivative is unstable.

🧰 And now what? How do we fix it?

The Shifted Grünwald–Letnikov Fractional Derivative

Shifted Grünwald–Letnikov Fractional Derivative

Let $\alpha > 0$, $f \in \mathcal{C}^{\lceil \alpha \rceil}([a,b])$, $a < x \leq b$, $\mathbb{N} \ni p > 0$ then

$$ {}^{GL}D_{[a,x]} f(x) = \lim_{N \to +\infty} \frac{\Delta_{h_N}^{\alpha} f(x)}{h_N^{\alpha}} = \lim_{N \to +\infty} \frac{1}{h_N^{\alpha}} \sum_{k=0}^{N} (-1)^k \binom{\alpha}{k} f(x - (k-p)h_N), $$

with $h_N = (x-a)/N$.

If we **repeat the argument with the Fourier transform**, we discover

$$ \mathfrak{F}\{{}^{GL}D_{[a,x]} f(x)\}(k) = (-ik)^{\alpha} \omega(-ikh) \hat{f}(k), $$

with

$$ \omega(z) = \left(\frac{1 - e^{-z}}{z} \right)^{\alpha} e^{zp} = 1 - \left(p - \frac{\alpha}{2} \right) z + O(|z|^2). $$

The Shifted Grünwald–Letnikov Fractional Derivative

Shifted Grünwald–Letnikov Fractional Derivative

Let $\alpha > 0$, $f \in \mathcal{C}^{\lceil \alpha \rceil}([a,b])$, $a < x \leq b$, $\mathbb{N} \ni p > 0$ then

$$ {}^{GL}D_{[a,x]} f(x) = \lim_{N \to +\infty} \frac{\Delta_{h_N}^{\alpha} f(x)}{h_N^{\alpha}} = \lim_{N \to +\infty} \frac{1}{h_N^{\alpha}} \sum_{k=0}^{N} (-1)^k \binom{\alpha}{k} f(x - (k-p)h_N), $$

with $h_N = (x-a)/N$.

If we **repeat the argument with the Fourier transform**, we discover

$$ \mathfrak{F}\{{}^{GL}D_{[a,x]} f(x)\}(k) = (-ik)^{\alpha} \hat{f}(k) + (ik)^{\alpha} (\omega(-ikh) - 1) \hat{f}(k) $$

with

$$ \omega(z) = \left(\frac{1 - e^{-z}}{z} \right)^{\alpha} e^{zp} = 1 - \left(p - \frac{\alpha}{2} \right) z + O(|z|^2) \Rightarrow |\omega(-ix) - 1| \leq Cx \, \forall \, x \in \mathbb{R}. $$

The Shifted Grünwald–Letnikov Fractional Derivative

Shifted Grünwald–Letnikov Fractional Derivative

Let $\alpha > 0$, $f \in \mathcal{C}^{\lceil \alpha \rceil}([a,b])$, $a < x \leq b$, $\mathbb{N} \ni p > 0$ then

$$ {}^{GL}D_{[a,x]} f(x) = \lim_{N \to +\infty} \frac{\Delta_{h_N}^{\alpha} f(x)}{h_N^{\alpha}} = \lim_{N \to +\infty} \frac{1}{h_N^{\alpha}} \sum_{k=0}^{N} (-1)^k \binom{\alpha}{k} f(x - (k-p)h_N), $$

with $h_N = (x-a)/N$.

If we **repeat the argument with the Fourier transform**, we discover

$$ \mathfrak{F}\{{}^{GL}D_{[a,x]} f(x)\}(k) = \mathcal{F}\{{}^{RL}D_{(-\infty,x]}^{\alpha} f\}(k) + \widehat{\phi}(k,h) $$

with

$$ \omega(z) = \left(\frac{1 - e^{-z}}{z} \right)^{\alpha} e^{zp} = 1 - \left(p - \frac{\alpha}{2} \right) z + O(|z|^2) \Rightarrow |\omega(-ix) - 1| \leq Cx \, \forall \, x \in \mathbb{R}. $$

The Shifted Grünwald–Letnikov Fractional Derivative

Shifted Grünwald–Letnikov Fractional Derivative

Let $\alpha > 0$, $f \in \mathcal{C}^{\lceil \alpha \rceil}([a,b])$, $a < x \leq b$, $\mathbb{N} \ni p > 0$ then

$$ {}^{GL}D_{[a,x]} f(x) = \lim_{N \to +\infty} \frac{\Delta_{h_N}^{\alpha} f(x)}{h_N^{\alpha}} = \lim_{N \to +\infty} \frac{1}{h_N^{\alpha}} \sum_{k=0}^{N} (-1)^k \binom{\alpha}{k} f(x - (k-p)h_N), $$

with $h_N = (x-a)/N$.

If we **repeat the argument with the Fourier transform**, we discover

$$ \mathfrak{F}\{{}^{GL}D_{[a,x]} f(x)\}(k) = \mathcal{F}\{{}^{RL}D_{(-\infty,x]}^{\alpha} f\}(k) + \hat{\phi}(k,h) $$

with

$$ |\phi(k,h)| \leq |k|^{\alpha} C |hk| |\hat{f}(k)| \Rightarrow |\phi(h,x)| < ICh, \quad I = \int_{-\infty}^{+\infty} (1+|k|)^{\alpha+1} |\hat{f}(k)| \, \mathrm{d}k < +\infty. $$

The Shifted Grünwald–Letnikov Fractional Derivative

Shifted Grünwald–Letnikov Fractional Derivative

Let $\alpha > 0$, $f \in \mathcal{C}^{\lceil \alpha \rceil}([a,b])$, $a < x \leq b$, $\mathbb{N} \ni p > 0$ then

$$ {}^{GL}D_{[a,x]} f(x) = \lim_{N \to +\infty} \frac{\Delta_{h_N}^{\alpha} f(x)}{h_N^{\alpha}} = \lim_{N \to +\infty} \frac{1}{h_N^{\alpha}} \sum_{k=0}^{N} (-1)^k \binom{\alpha}{k} f(x - (k-p)h_N), $$

with $h_N = (x-a)/N$.

- They give the same operator uniformly in x as $h \to 0$, therefore we can use the shifted version with *any* shift to approximate the Riemann-Liouville derivative,

The Shifted Grünwald–Letnikov Fractional Derivative

Shifted Grünwald–Letnikov Fractional Derivative

Let $\alpha > 0$, $f \in \mathcal{C}^{\lceil \alpha \rceil}([a,b])$, $a < x \leq b$, $\mathbb{N} \ni p > 0$ then

$$ {}^{GL}D_{[a,x]} f(x) = \lim_{N \to +\infty} \frac{\Delta_{h_N}^{\alpha} f(x)}{h_N^{\alpha}} = \lim_{N \to +\infty} \frac{1}{h_N^{\alpha}} \sum_{k=0}^{N} (-1)^k \binom{\alpha}{k} f(x - (k-p)h_N), $$

with $h_N = (x-a)/N$.

- They give the same operator uniformly in x as $h \to 0$, therefore we can use the shifted version with *any* shift to approximate the Riemann-Liouville derivative,
- To get the *best constant* C we can minimize the $|p - \alpha/2|$ term in $\omega(z)$, that is, we select $p = 1$.

The Shifted Grünwald–Letnikov Fractional Derivative

Shifted Grünwald–Letnikov Fractional Derivative

Let $\alpha > 0$, $f \in \mathcal{C}^{\lceil \alpha \rceil}([a,b])$, $a < x \leq b$, $\mathbb{N} \ni p > 0$ then

$$ {}^{GL}D_{[a,x]} f(x) = \lim_{N \to +\infty} \frac{\Delta_{h_N}^{\alpha} f(x)}{h_N^{\alpha}} = \lim_{N \to +\infty} \frac{1}{h_N^{\alpha}} \sum_{k=0}^{N} (-1)^k \binom{\alpha}{k} f(x - (k-p)h_N), $$

with $h_N = (x-a)/N$.

- They give the same operator uniformly in x as $h \to 0$, therefore we can use the shifted version with *any* shift to approximate the Riemann-Liouville derivative,
- To get the *best constant* C we can minimize the $|p - \alpha/2|$ term in $\omega(z)$, that is, we select $p = 1$.

? Let us see if using the *shifted version* with $p = 1$ solves our **stability problem**.

Back to finite differences: implicit Euler

We use the **shifted Grünwald–Letnikov** and the **implicit Euler** method

$$\frac{w_i^{n+1} - w_i^n}{\Delta t} = -v_i \frac{w_i^{n+1} - w_{i-1}^{n+1}}{h_N} + \frac{d_i}{h_N^\alpha} \sum_{k=0}^{i+1} g_k w_{i-k+1}^{n+1} + f_i^{n+1}.$$

Back to finite differences: implicit Euler

We use the **shifted Grünwald–Letnikov** and the **implicit Euler** method

$$w_i^{n+1} - w_i^n = -E_i(w_i^{n+1} - w_{i-1}^{n+1}) + B_i \sum_{k=0}^{i+1} g_k w_{i-k+1}^{n+1} + \Delta t f_i^{n+1}.$$

- Set $E_i = {}^{v_i \Delta t}/_{h_N}$,
  $B_i = {}^{d_i \Delta t}/_{h_N^\alpha}$,

Back to finite differences: implicit Euler

We use the **shifted Grünwald–Letnikov** and the **implicit Euler** method

$$-g_0 B_i w_{i+1}^{n+1} + (1 + E_i - g_i B_i) w_i^{n+1} - (E_i + g_2 B_i) w_{i-1}^{n+1} - B_i \sum_{k=3}^{i+1} g_k w_{i-k+1}^{n+1} = c_i^n + \Delta t\, f_i^{n+1}.$$

- Set $E_i = {}^{v_i \Delta t}/_{h_N}$, $B_i = {}^{d_i \Delta t}/_{h_N^\alpha}$,
- reorder the system of equations,

Back to finite differences: implicit Euler

We use the **shifted Grünwald–Letnikov** and the **implicit Euler** method

$$-g_0 B_i w_{i+1}^{n+1} + (1 + E_i - g_i B_i) w_i^{n+1} - (E_i + g_2 B_i) w_{i-1}^{n+1} - B_i \sum_{k=3}^{i+1} g_k w_{i-k+1}^{n+1} = c_i^n + \Delta t\, f_i^{n+1}.$$

- Set $E_i = {}^{v_i \Delta t}/_{h_N}$, $B_i = {}^{d_i \Delta t}/_{h_N^\alpha}$,
- reorder the system of equations,
- and obtain

$$A_N \mathbf{w}^{n+1} = \mathbf{w}^n + \Delta t\, \mathbf{f}^{n+1}.$$

Back to finite differences: implicit Euler

We use the **shifted Grünwald–Letnikov** and the **implicit Euler** method

$$-g_0 B_i w_{i+1}^{n+1} + (1 + E_i - g_i B_i) w_i^{n+1} - (E_i + g_2 B_i) w_{i-1}^{n+1} - B_i \sum_{k=3}^{i+1} g_k w_{i-k+1}^{n+1} = c_i^n + \Delta t\, f_i^{n+1}.$$

- Set $E_i = {}^{v_i \Delta t}/_{h_N}$, $B_i = {}^{d_i \Delta t}/_{h_N^\alpha}$,
- reorder the system of equations,
- and obtain

$$A_N \mathbf{w}^{n+1} = \mathbf{w}^n + \Delta t\, \mathbf{f}^{n+1}.$$

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ -E_1 - g_2 B_1 & 1 + E_1 - g_1 B_1 & -g_0 B_1 & \ddots & & \\ -g_3 B_2 & -E_2 - g_2 B_2 & 1 + E_2 - g_1 B_2 & -g_0 B_2 & \ddots & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ -g_N B_{N-1} & \cdots & \cdots & \cdots & \cdots & -g_0 B_{N-1} \\ 0 & \cdots & \cdots & \cdots & \cdots & 1 \end{bmatrix}$$

Back to finite differences: implicit Euler

We use the **shifted Grünwald–Letnikov** and the **implicit Euler** method

$$-g_0 B_i w_{i+1}^{n+1} + (1 + E_i - g_i B_i) w_i^{n+1} - (E_i + g_2 B_i) w_{i-1}^{n+1} - B_i \sum_{k=3}^{i+1} g_k w_{i-k+1}^{n+1} = c_i^n + \Delta t\, f_i^{n+1}.$$

- Set $E_i = {}^{v_i \Delta t}/_{h_N}$, $B_i = {}^{d_i \Delta t}/_{h_N^\alpha}$,
- reorder the system of equations,
- and obtain

$$\begin{aligned} \mathbf{w}^{n+1} &= [w_0^{n+1}, w_1^{n+1}, \ldots, w_N^{n+1}]^T, \\ \mathbf{w}^{n} &= [w_0^{n}, w_1^{n}, \ldots, w_N^{n}]^T, \\ \mathbf{f}^{n+1} &= \Delta t [0, f_1^n, \ldots, f_{N-1}^n, 0]^T. \end{aligned}$$

$$A_N \mathbf{w}^{n+1} = \mathbf{w}^n + \Delta t\, \mathbf{f}^{n+1}.$$

Back to finite differences: implicit Euler

We use the **shifted Grünwald–Letnikov** and the **implicit Euler** method

$$-g_0 B_i w_{i+1}^{n+1} + (1 + E_i - g_i B_i) w_i^{n+1} - (E_i + g_2 B_i) w_{i-1}^{n+1} - B_i \sum_{k=3}^{i+1} g_k w_{i-k+1}^{n+1} = c_i^n + \Delta t\, f_i^{n+1}.$$

- Set $E_i = {}^{v_i \Delta t}/_{h_N}$, $B_i = {}^{d_i \Delta t}/_{h_N^\alpha}$,
- reorder the system of equations,
- and obtain

$$A_N \mathbf{w}^{n+1} = \mathbf{w}^n + \Delta t\, \mathbf{f}^{n+1}.$$

$$\begin{aligned} \mathbf{w}^{n+1} &= [w_0^{n+1}, w_1^{n+1}, \ldots, w_N^{n+1}]^T, \\ \mathbf{w}^n &= [w_0^n, w_1^n, \ldots, w_N^n]^T, \\ \mathbf{f}^{n+1} &= \Delta t [0, f_1^n, \ldots, f_{N-1}^n, 0]^T. \end{aligned}$$

To prove **stability** we need to have $\rho(A_N^{-1}) \leq 1$:

$$\boldsymbol{\epsilon}^1 = A_N^{-1} \boldsymbol{\epsilon}^0.$$

Back to finite differences: implicit Euler

Let $(\lambda, \mathbf{x})$ be an eigencouple of A_N, i.e., $A_N\mathbf{x} = \lambda\mathbf{x}$, $\mathbf{x} \neq \mathbf{0}$.

Back to finite differences: implicit Euler

Let $(\lambda, \mathbf{x})$ be an eigencouple of A_N, i.e., $A_N\mathbf{x} = \lambda\mathbf{x}$, $\mathbf{x} \neq \mathbf{0}$.

1. Choose i such that $|x_i| = \max\{|x_j| \,:\, j = 0, \ldots, N\}$,

Back to finite differences: implicit Euler

Let $(\lambda, \mathbf{x})$ be an eigencouple of A_N, i.e., $A_N\mathbf{x} = \lambda\mathbf{x}$, $\mathbf{x} \neq \mathbf{0}$.

1. Choose i such that $|x_i| = \max\{|x_j| \,:\, j = 0, \ldots, N\}$,
2. Then $\displaystyle\sum_{j=0}^{N} (A_N)_{i,j} x_j = x_i$, and thus

$$\lambda = A_{i,i} + \sum_{\substack{j=0 \\ j\neq i}}^{N} (A_N)_{i,j} \frac{x_j}{x_i},$$

Back to finite differences: implicit Euler

Let $(\lambda, \mathbf{x})$ be an eigencouple of A_N, i.e., $A_N\mathbf{x} = \lambda\mathbf{x}$, $\mathbf{x} \neq \mathbf{0}$.

1. Choose i such that $|x_i| = \max\{|x_j| \,:\, j = 0, \ldots, N\}$,
2. Then $\displaystyle\sum_{j=0}^{N} (A_N)_{i,j} x_j = x_i$, and thus

$$\lambda = A_{i,i} + \sum_{\substack{j=0 \\ j\neq i}}^{N} (A_N)_{i,j} \frac{x_j}{x_i},$$

3. If $i = 0$ or $i = N$ then $\lambda = 1$, otherwise

$$\lambda = 1 + E_i - g_1 B_1 - g_0 B_i \frac{x_{i+1}}{x_i} (E_i + g_2 B_i) \frac{x_{i-1}}{x_i} - B_i \sum_{j=0}^{i-2} h_{i-j+1} \frac{x_j}{x_i}$$

Back to finite differences: implicit Euler

Let $(\lambda, \mathbf{x})$ be an eigencouple of A_N, i.e., $A_N\mathbf{x} = \lambda\mathbf{x}$, $\mathbf{x} \neq \mathbf{0}$.

1. Choose i such that $|x_i| = \max\{|x_j| \,:\, j = 0, \ldots, N\}$,
2. Then $\displaystyle\sum_{j=0}^{N} (A_N)_{i,j} x_j = x_i$, and thus

$$\lambda = A_{i,i} + \sum_{\substack{j=0 \\ j\neq i}}^{N} (A_N)_{i,j} \frac{x_j}{x_i},$$

3. If $i = 0$ or $i = N$ then $\lambda = 1$, otherwise

$$\lambda = 1 + E_i(1 - {}^{x_{i-1}}/_{x_i}) - B_i \left[g_1 + \sum_{\substack{j=0 \\ j\neq i}}^{i+1} g_{i-j+1} \frac{x_j}{x_i} \right]$$

Back to finite differences: implicit Euler

4. We have $\sum_{k\geq 0} g_k = 0$, $\alpha \in (1,2]$ and thus $g_1 = -\alpha$ and $g_k \geq 0$ for $k \neq 1$, thus

$$-g_1 \geq \sum_{\substack{k=0\\k\neq 1}}^{j} g_k \qquad \forall\, j = 0, 1, 2, \ldots$$

furthermore $|x_j/x_i| < 1$, and thus

$$\sum_{\substack{j=0\\j\neq i}}^{i+1} g_{i-j+1} \left|\frac{x_j}{x_i}\right| \leq \sum_{\substack{j=0\\j\neq i}}^{i+1} g_{i-j+1} \leq -g_1.$$

Back to finite differences: implicit Euler

4. We have $\sum_{k\geq 0} g_k = 0$, $\alpha \in (1,2]$ and thus $g_1 = -\alpha$ and $g_k \geq 0$ for $k \neq 1$, thus

$$-g_1 \geq \sum_{\substack{k=0\\k\neq 1}}^{j} g_k \qquad \forall\, j = 0, 1, 2, \ldots$$

   furthermore $|x_j/x_i| < 1$, and thus

$$\sum_{\substack{j=0\\j\neq i}}^{i+1} g_{i-j+1} \left|\frac{x_j}{x_i}\right| \leq \sum_{\substack{j=0\\j\neq i}}^{i+1} g_{i-j+1} \leq -g_1.$$

3. If $i = 0$ or $i = N$ then $\lambda = 1$, otherwise

$$|\lambda| \geq 1 + \underbrace{E_i}_{\geq 0}(1 - \underbrace{x_{i-1}/x_i}_{\leq 1}) + \underbrace{B_i}_{\geq 0}\left[g_1 + \sum_{\substack{j=0\\j\neq i}}^{i+1} g_{i-j+1}\left|\frac{x_j}{x_i}\right|\right] \geq 1.$$

Back to finite differences: implicit Euler

Theorem (Meerschaert and Tadjeran 2004)

The implicit Euler method solution to

$$\frac{\partial w}{\partial t} = -v(x)\frac{\partial w}{\partial x} + d(x)\,{}^{RL}D^{\alpha}_{[0,x]}w + f(x,t), \qquad 1 < \alpha \leq 2,\ v(x), d(x) \geq 0.$$

with boundary conditions $w(0,t) = 0$, $w(1,t) = 0$ for all $t \geq 0$, based on the shifted Grünwald–Letnikov approximation with $h_N = 1/N$, is consistent of order $O(h + \Delta t)$ and *unconditionally stable*.

- We have only a left-sided fractional derivative, we could put a non-homogeneous condition on the right-hand side,
- We can now start **looking into the matrices** to devise solution strategies for the *sequence of linear systems*

$$A_N \mathbf{w}^{n+1} = \mathbf{w}^n + \Delta t\, \mathbf{f}^{n+1}.$$

Grünwald–Letnikov matrices

To look at the matrices we go back to the first form of the diffusion equation (FDE_1)

$$\begin{cases} \frac{\partial W}{\partial t} = \theta\, {}^{RL}D^{\alpha}_{[0,x]} W(x,t) + (1-\theta)\, {}^{RL}D^{\alpha}_{[x,1]} W(x,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \\ W(x,t) = W_0(x). \end{cases}$$

Grünwald–Letnikov matrices

To look at the matrices we go back to the first form of the diffusion equation (FDE_1)

$$\begin{cases} \frac{\partial W}{\partial t} = \theta\, {}^{GL}D^{\alpha}_{[0,x]} W(x,t) + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} W(x,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \\ W(x,t) = W_0(x). \end{cases}$$

1. Substitute the Riemann-Liouville derivative with the Grünwald–Letnikov one,

Grünwald–Letnikov matrices

To look at the matrices we go back to the first form of the diffusion equation (FDE_1)

$$\begin{cases} \frac{\partial W}{\partial t} = \theta\, {}^{GL}D^{\alpha}_{[0,x]} W(x,t) + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} W(x,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \\ W(x,t) = W_0(x). \end{cases}$$

1. Substitute the Riemann-Liouville derivative with the Grünwald–Letnikov one,
2. Choose $N \in \mathbb{N}$ at which to truncate the *shifted* series expansions

$$h_N^{\alpha} \frac{\partial W_i}{\partial t} = \theta \sum_{k=0}^{i+1} (-1)^k \binom{\alpha}{k} W_{i-k+1} + (1-\theta) \sum_{k=0}^{N-i+2} (-1)^k \binom{\alpha}{k} W_{i+k-1}, \; i = 0, \ldots, N.$$

Grünwald–Letnikov matrices

To look at the matrices we go back to the first form of the diffusion equation (FDE_1)

$$\begin{cases} \frac{\partial W}{\partial t} = \theta\, {}^{GL}D^{\alpha}_{[0,x]} W(x,t) + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} W(x,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \\ W(x,t) = W_0(x). \end{cases}$$

1. Substitute the Riemann-Liouville derivative with the Grünwald–Letnikov one,
2. Choose $N \in \mathbb{N}$ at which to truncate the *shifted* series expansions
3. Apply, e.g., *backward Euler* to discretize the derivative w.r.t. time

$$\frac{h_N^{\alpha}}{\Delta t}(W_i^{j+1} - W_i^j) = \theta \sum_{k=0}^{i-k+1} (-1)^k \binom{\alpha}{k} W_{i-k+1}^j + (1-\theta) \sum_{k=0}^{N+i-2} (-1)^k \binom{\alpha}{k} W_{i+k-1}^j, \quad \begin{array}{l} i = 0, \dots, N, \\ j = 0, \dots, M-1. \end{array}$$

The matrix formulation

We call again $\mathbf{w}^j$, $\mathbf{w}^{j+1}$ the vectors containing the solution **on inner grid points**, then we can rewrite the set of linear equations as

$$\left(I_N - \frac{\Delta t}{h_N^\alpha}\left[\theta G_N + (1-\theta) G_N^T\right]\right)\mathbf{w}^{n+1} = \mathbf{w}^n$$

where

$$G_N = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ g_2 & g_1 & g_0 & & \\ \vdots & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & g_0 \\ g_{N-1} & \cdots & g_3 & g_2 & g_1 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

```
function G = glmatrix(N,alpha)
%%GLMATRIX produces the GL discretization of
% the Riemann-Liouville derivative
g = gl(N,alpha);
c = zeros(N,1); r = zeros(1,N);
r(1:2) = g(2:-1:1);
c(1:N) = g(2:end);
G = toeplitz(c,r);
end
```

The matrix formulation

To obtain a simple code for the complete problem

```
%% Discretization data
hN = 1/(N-1); x = 0:hN:1;
dt = hN; t = 0:dt:1;
%% Discretize
G = glmatrix(N,alpha); Gt =
↪  glmatrix(N,alpha).';
I = eye(N,N);
% apply B.C.
G(1,:) = -I(1,:); G(N,:) = -I(N,:);
Gt(1,:) = -I(1,:); Gt(N,:) = -I(N,:);
% Left-hand side
A = I - dt/hN^alpha*(theta*G + (1-theta)*Gt);
% Right-hand side
w = w0(x).';
```

- Select $\theta = 1/2$, $\alpha = 3/2$, and $W_0(x) = 5x(1-x)$,
- Discretize the interval $[0, 1]$ on N points,
- Build the I and G_N matrices,
- Apply the Dirichlet b.c.s,
- Assemble A and $\mathbf{w}^0$.

The matrix formulation

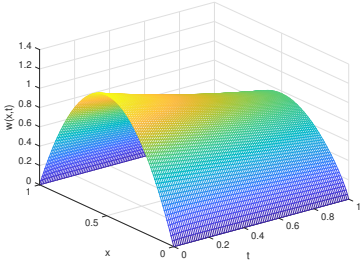
To obtain a simple code for the complete problem

```
%% Discretization data
hN = 1/(N-1); x = 0:hN:1;
dt = hN; t = 0:dt:1;
%% Discretize
G = glmatrix(N,alpha); Gt =
↪  glmatrix(N,alpha).';
I = eye(N,N);
% apply B.C.
G(1,:) = -I(1,:); G(N,:) = -I(N,:);
Gt(1,:) = -I(1,:); Gt(N,:) = -I(N,:);
% Left-hand side
A = I - dt/hN^alpha*(theta*G + (1-theta)*Gt);
% Right-hand side
w = w0(x).';
```

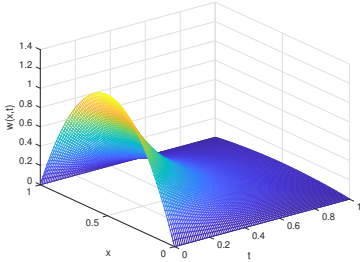
- Select $\theta = 1/2$, $\alpha = 3/2$, and $W_0(x) = 5x(1-x)$,
- Discretize the interval $[0, 1]$ on N points,
- Build the I and G_N matrices,
- Apply the Dirichlet b.c.s,
- Assemble A and $\mathbf{w}^0$.

March the scheme in time:

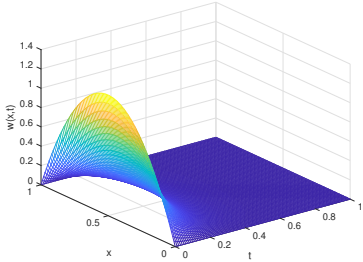
```
for i=2:N
  w = A\w;
end
```



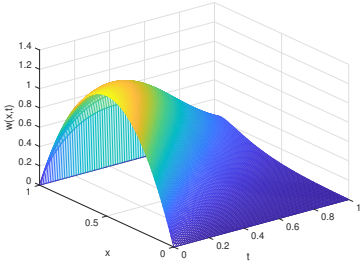
$\alpha = 1.1,\ \theta = 0.5$



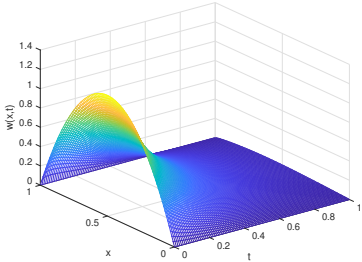
$\alpha = 1.5,\ \theta = 0.5$



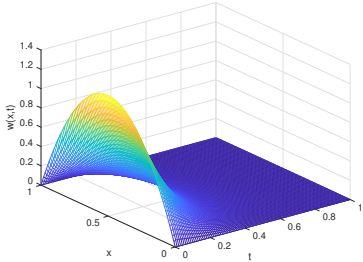
$\alpha = 2.0,\ \theta = 0.5$



$\alpha = 1.1,\ \theta = 0.1$



$\alpha = 1.5,\ \theta = 0.3$



$\alpha = 1.8,\ \theta = 0.9$

The solution step

- How can we **efficiently solve** the linear systems

$$A\mathbf{w}^{n+1} = \mathbf{w}^n,$$

  needed for the time-stepping?
- Can we find a reliable procedure working also for **multi-dimensional cases**?
- Is **dense linear algebra** a compulsory choice?

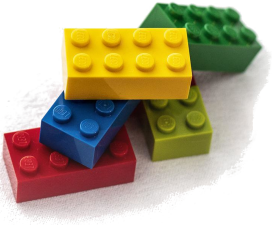
The solution step

- How can we **efficiently solve** the linear systems

$$A\mathbf{w}^{n+1} = \mathbf{w}^n,$$

  needed for the time-stepping?
- Can we find a reliable procedure working also for **multi-dimensional cases**?
- Is **dense linear algebra** a compulsory choice?

These matrices have structures we can **exploit**!



Toeplitz matrices

Toeplitz matrix

A **Toeplitz matrix** is a matrix whose entries are constant along the diagonals

$$T_n(f) = \begin{bmatrix} t_0 & t_{-1} & \dots & t_{2-n} & t_{1-n} \\ t_1 & t_0 & t_{-1} & \dots & t_{2-n} \\ \vdots & t_1 & t_0 & \ddots & \vdots \\ t_{n-2} & \dots & \ddots & \ddots & t_{-1} \\ t_{n-1} & t_{n-2} & \dots & t_1 & t_0 \end{bmatrix}.$$

Toeplitz matrices

Toeplitz matrix

A **Toeplitz matrix** is a matrix whose entries are constant along the diagonals

$$T_n(f) = \begin{bmatrix} t_0 & t_{-1} & \dots & t_{2-n} & t_{1-n} \\ t_1 & t_0 & t_{-1} & \dots & t_{2-n} \\ \vdots & t_1 & t_0 & \ddots & \vdots \\ t_{n-2} & \dots & \ddots & \ddots & t_{-1} \\ t_{n-1} & t_{n-2} & \dots & t_1 & t_0 \end{bmatrix}.$$

Generating function

$$f(x) = \sum_{k=-\infty}^{+\infty} t_k e^{i \cdot kx}, \quad t_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(\theta) e^{-ik\theta} d\theta, \ k = 0, \pm 1, \pm 2, \dots$$

the t_k are the Fourier coefficients is called a *generating function* of the matrix $T_n(f)$.

Circulant matrices

Circulant matrix

A **Circulant matrix** $C_n \in \mathbb{R}^{n \times n}$ is a Toeplitz matrix in which each row is a cyclic shift of the row above it, i.e., $(C_n)_{i,j} = c_{(j-i) \mod n}$:

$$C_n = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & \cdots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \ddots & & \vdots \\ c_{n-2} & c_{n-1} & c_0 & c_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & c_2 \\ \vdots & & \ddots & \ddots & c_0 & c_1 \\ c_1 & \cdots & \cdots & c_{n-2} & c_{n-1} & c_0 \end{bmatrix}.$$

Toeplitz and Circulant matrices: some properties

Properties

1. The operator $T_n : \mathbb{L}^1[-\pi,\pi] \to \mathbb{C}^{n\times n}$ defined by the Toeplitz matrix construction is linear and positive, i.e., if $f \geq 0$ then $T_n(f) = T_n(f)^H \ \forall\, n$ and $\mathbf{x}^H T_n(f)\mathbf{x} \geq 0$ $\forall\, \mathbf{x} \in \mathbb{C}^n$.
2. Given $f \in \mathbb{L}^1[-\pi,\pi]$ such that $m_f = \operatorname{ess\,inf}(f)$ and $M_f = \operatorname{ess\,sup}(f)$.
   If $m_f > -\infty$ then $m_f \leq \lambda_j(T_n(f)) \ \forall j = 1,\dots,n$;
   If $M_f < \infty$ then $M_f \geq \lambda_j(T_n(f)) \ \forall j = 1,\dots,n$.
   If f is not identical to a real constant and both the inequalities hold,
   $$m_f < \lambda_j(T_n(f)) < M_f \quad \forall j = 1,\dots,n.$$
3. Circulant matrices are simultaneously diagonalized by the unitary matrix F_n
   $$(F_n)_{j,k} = \frac{1}{\sqrt{n}} e^{\frac{-2\pi i\, j\, k}{n}}, \mathcal{C} = \left\{ C_n \in \mathbb{C}^{n\times n} \mid C_n = FDF^H \ : \ D = \operatorname{diag}(d_0, d_1, \dots, d_{n-1}) \right\}.$$

Asymptotic distribution - I

Asymptotic eigenvalue distribution

Given a sequence of matrices $\{X_n\}_n \in \mathbb{C}^{d_n \times d_n}$ with $d_n = \{\dim X_n\}_n \overset{n\to+\infty}{\longrightarrow} \infty$ monotonically and a μ-measurable function $f : D \to \mathbb{R}$, with $\mu(D) \in (0, \infty)$, we say that the sequence $\{X\}_n$ is distributed in the sense of the eigenvalues as the function f and write $\{X_n\}_n \sim_\lambda f$ if and only if,

$$\lim_{n\to\infty} \frac{1}{d_n} \sum_{j=0}^{d_n} F(\lambda_j(X_n)) = \frac{1}{\mu(D)} \int_D F(f(t))dt, \quad \forall F \in \mathcal{C}_c(D),$$

where $\lambda_j(\cdot)$ indicates the j-th eigenvalue.

Asymptotic distribution - II

Asymptotic singular value distribution

Given a sequence of matrices $\{X_n\}_n \in \mathbb{C}^{d_n \times d_n}$ with $d_n = \{\dim X_n\}_n \overset{n\to+\infty}{\longrightarrow} \infty$ monotonically and a μ-measurable function $f : D \to \mathbb{R}$, with $\mu(D) \in (0, \infty)$, we say that the sequence $\{X\}_n$ is distributed in the sense of the singular values as the function f and write $\{X_n\}_n \sim_\sigma f$ if and only if

$$\lim_{n\to\infty} \frac{1}{d_n} \sum_{j=0}^{d_n} F(\sigma_j(X_n)) = \frac{1}{\mu(D)} \int_D F(|f(t)|)dt, \quad \forall F \in \mathcal{C}_c(D),$$

where $\sigma_j(\cdot)$ is the j-th singular value.

Asymptotic distribution - III

Theorem (Asymptotic distribution of Toeplitz matrices)

Given the generating function f, $T_n(f)$ is distributed in the sense of the eigenvalues as f, written also as $T_n(f) \sim_\lambda f$, if one of the following conditions hold:

1. (Grenander and Szegö 2001): f is real valued and $f \in \mathbb{L}^\infty$,
2. (Tyrtyshnikov 1996): f is real valued and $f \in \mathbb{L}^2$.

Moreover, $T_n(f)$ is distributed in the sense of the singular values as f, written also as $T_n(f) \sim_\sigma f$, if one of the following conditions hold:

1. (Avram 1988; Parter 1986): $f \in \mathbb{L}^\infty$,
2. (Tyrtyshnikov 1996): $f \in \mathbb{L}^2$.

Singular value distribution of G_N

- The matrix G_N is a **Toeplitz** and **Hessenberg** matrix,

Singular value distribution of G_N

- The matrix G_N is a **Toeplitz** and **Hessenberg** matrix,
- Does it have a **generating function**?

Singular value distribution of G_N

- The matrix G_N is a **Toeplitz** and **Hessenberg** matrix,
- Does it have a **generating function**?
  - **Yes**! And we have already computed it several times! The coefficients $\{g_k^{(\alpha)}\}_k$ where given by the **binomial expansion** of $(1+z)^\alpha$, and thus

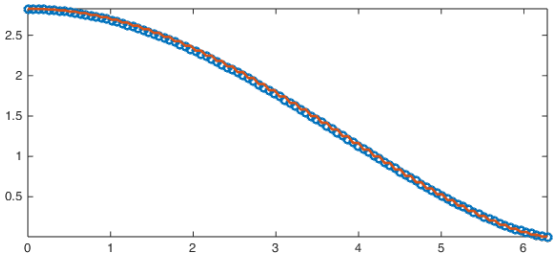
$$f(\theta) = e^{-i\theta}\left(1+\exp(i(\theta+\pi))\right)^\alpha, \qquad \theta \in [0, 2\pi)$$

Singular value distribution of G_N

- The matrix G_N is a **Toeplitz** and **Hessenberg** matrix,
- Does it have a **generating function**?
  - **Yes**! And we have already computed it several times! The coefficients $\{g_k^{(\alpha)}\}_k$ where given by the **binomial expansion** of $(1+z)^\alpha$, and thus

$$f(\theta) = e^{-i\theta}\left(1+\exp(i(\theta+\pi))\right)^\alpha, \qquad \theta \in [0, 2\pi)$$

```
N = 100;
alpha = 1.5;
G = glmatrix(N,alpha);
s = @(t) exp(-1i*t).*(1 + ...
 exp(1i*(t+pi))).^alpha;
sv = svd(G);
th = linspace(0,2*pi,N);
plot(th,sv,'o',th,sort(abs(s(th)),...
 'descend'),'-','LineWidth',2);
```



Conclusion and summary

- We introduced **p**artial **d**ifferential **e**quations with **f**ractional (FPDE) derivative with respect to the space variables,
- we connected *fractional diffusion* and continuous time random walk using *Lévy flights*,
- we introduced the Grünwald-Letnikov fractional derivative, highlighted the connection with the Riemann-Liouville derivative.
- We introduced a *stable discretization* of finite difference type,
- and we started investigating the structure of the underlying matrices.

Next up

- Investigating the structure of the underlying matrices for different FPDEs.
- Looking into some preconditioners and solution strategies based on structured matrices.

Bibliography I

- Avram, F. (1988). "On bilinear forms in Gaussian random variables and Toeplitz matrices". In: *Probab. Theory Related Fields* 79.1, pp. 37–45.
- Ferreira, R. A. C. (2013). "A Lyapunov-type inequality for a fractional boundary value problem". In: *Fract. Calc. Appl. Anal.* 16.4, pp. 978–984. ISSN: 1311-0454. DOI: 10.2478/s13540-013-0060-5. URL: https://doi.org/10.2478/s13540-013-0060-5.
- Grenander, U. and G. Szegö (2001). *Toeplitz forms and their applications*. Vol. 321. University of California Press.
- Grünwald, A. K. (1867). "Über "begrente" Derivationen und deren Anwedung". In: *Zangew Math und Phys* 12, pp. 441–480.
- Jin, B. et al. (2015). "Variational formulation of problems involving fractional order differential operators". In: *Math. Comp.* 84.296, pp. 2665–2700. ISSN: 0025-5718. DOI: 10.1090/mcom/2960. URL: https://doi.org/10.1090/mcom/2960.
- Letnikov, A. V. (1868). *Theory of differentiation with an arbitrary index*.

Bibliography II

- Meerschaert, M. M. and C. Tadjeran (2004). "Finite difference approximations for fractional advection-dispersion flow equations". In: *J. Comput. Appl. Math.* 172.1, pp. 65–77. ISSN: 0377-0427. DOI: 10.1016/j.cam.2004.01.033. URL: https://doi.org/10.1016/j.cam.2004.01.033.
- Parter, S. V. (1986). "On the distribution of the singular values of Toeplitz matrices". In: *Linear Algebra Appl.* 80, pp. 115–130.
- Tyrtyshnikov, E. E. (1996). "A unifying approach to some old and new theorems on distribution and clustering". In: *Linear Algebra Appl.* 232, pp. 1–43.

An introduction to fractional calculus

Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

September, 2022

Solving linear system with Toeplitz-like matrices

In the last lecture we discretized

$$\begin{cases} \frac{\partial W}{\partial t} = \theta\, {}^{RL}D^{\alpha}_{[0,x]} W(x,t) + (1-\theta)\, {}^{RL}D^{\alpha}_{[x,1]} W(x,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

Solving linear system with Toeplitz-like matrices

In the last lecture we discretized

$$\begin{cases} \frac{\partial W}{\partial t} = \theta\, {}^{GL}D^{\alpha}_{[0,x]} W(x,t) + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} W(x,t), & \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

Obtaining

$$\left(I_N - \frac{\Delta t}{h_N^{\alpha}} \left[\theta G_N + (1-\theta) G_N^T \right] \right) \mathbf{w}^{n+1} = \mathbf{w}^n$$

with

$$G_N = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ g_2 & g_1 & g_0 & & \\ \vdots & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & g_0 \\ g_{N-1} & \cdots & g_3 & g_2 & g_1 \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}.$$

Solving linear system with Toeplitz-like matrices

The matrix

$$A_N = I_N - \frac{\Delta t}{h_N^\alpha}\left[\theta G_N + (1-\theta) G_N^T\right],$$

is a **Toepltiz** matrix plus some rank corrections.

- By rearranging the right-hand side or restricting to solve only for the internal nodes we can avoid the rank corrections.

Solving linear system with Toeplitz-like matrices

The matrix

$$A_N = I_N - \frac{\Delta t}{h_N^\alpha}\left[\theta G_N + (1-\theta) G_N^T\right],$$

is a **Toepltiz** matrix plus some rank corrections.

- By rearranging the right-hand side or restricting to solve only for the internal nodes we can avoid the rank corrections.
- How do we solve such systems?

Solving linear system with Toeplitz-like matrices

The matrix

$$A_N = I_N - \frac{\Delta t}{h_N^\alpha}\left[\theta G_N + (1-\theta) G_N^T\right],$$

is a **Toepltiz** matrix plus some rank corrections.

- By rearranging the right-hand side or restricting to solve only for the internal nodes we can avoid the rank corrections.
- How do we solve such systems?
  - Direct methods

Solving linear system with Toeplitz-like matrices

The matrix

$$A_N = I_N - \frac{\Delta t}{h_N^\alpha}\left[\theta G_N + (1-\theta) G_N^T\right],$$

is a **Toepltiz** matrix plus some rank corrections.

- By rearranging the right-hand side or restricting to solve only for the internal nodes we can avoid the rank corrections.
- How do we solve such systems?
  - Direct methods
  - Iterative methods

Solving linear system with Toeplitz-like matrices

The matrix

$$A_N = I_N - \frac{\Delta t}{h_N^\alpha}\left[\theta G_N + (1-\theta) G_N^T\right],$$

is a **Toepltiz** matrix plus some rank corrections.

- By rearranging the right-hand side or restricting to solve only for the internal nodes we can avoid the rank corrections.
- How do we solve such systems?
  - Direct methods $\Rightarrow$ fast and superfast Toeplitz solvers
  - Iterative methods

Solving linear system with Toeplitz-like matrices

The matrix

$$A_N = I_N - \frac{\Delta t}{h_N^\alpha}\left[\theta G_N + (1-\theta) G_N^T\right],$$

is a **Toepltiz** matrix plus some rank corrections.

- By rearranging the right-hand side or restricting to solve only for the internal nodes we can avoid the rank corrections.
- How do we solve such systems?
  - Direct methods $\Rightarrow$ fast and superfast Toeplitz solvers
  - Iterative methods $\Rightarrow$ preconditioned Krylov methods, multigrid solvers/preconditioners

Direct Toeplitz solvers

Direct Toeplitz solver are *mostly* based on the answer to the following question:

❓ is the inverse of a Toeplitz matrix still a Toeplitz matrix?

Direct Toeplitz solvers

Direct Toeplitz solver are *mostly* based on the answer to the following question:

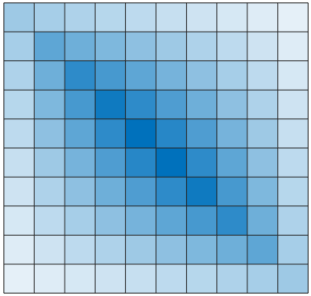
❓ is the inverse of a Toeplitz matrix still a Toeplitz matrix?

$$T_n = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

Direct Toeplitz solvers

Direct Toeplitz solver are *mostly* based on the answer to the following question:

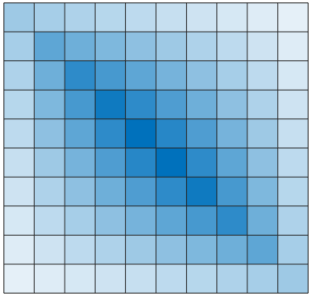
❓ is the inverse of a Toeplitz matrix still a Toeplitz matrix?

$T_n^{-1} =$ 

Direct Toeplitz solvers

Direct Toeplitz solver are *mostly* based on the answer to the following question:

❓ is the inverse of a Toeplitz matrix still a Toeplitz matrix?

$T_n^{-1} =$ 

So the answer is **no**, but... it seems that there is still some structure there, doesn't it?

The Gohberg–Semencul formula

... starting from a **displacement representation** of T_n, i.e.,

$$t_0 T_n = \begin{bmatrix} t_0 & 0 & \cdots & 0 \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ t_{n-1} & t_{n-2} & \cdots & t_0 \end{bmatrix} \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{1-n} \\ 0 & t_0 & \cdots & t_{2-n} \\ 0 & 0 & & \vdots \\ \vdots & \vdots & & t_{-1} \\ 0 & 0 & \cdots & t_0 \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ t_1 & 0 & \cdots & 0 & 0 \\ t_2 & t_1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ t_{n-1} & t_{n-2} & \cdots & t_1 & 0 \end{bmatrix} \begin{bmatrix} 0 & t_{-1} & t_{-2} & \cdots & t_{1-n} \\ 0 & 0 & t_{-1} & \cdots & t_{2-n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & & t_{-1} \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Gohberg and Semencul 1972 obtained a **displacement representation** of the **inverse**

$$z_1 T_n^{-1} = \begin{bmatrix} z_1 & 0 & \cdots & 0 \\ z_2 & z_1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ z_{n-1} & z_{n-2} & \cdots & 0 \\ z_n & z_{n-1} & \cdots & z_1 \end{bmatrix} \begin{bmatrix} v_n & v_{n-1} & \cdots & v_1 \\ 0 & v_n & \cdots & v_2 \\ 0 & 0 & & \vdots \\ \vdots & \vdots & & v_{n-1} \\ 0 & 0 & \cdots & v_n \end{bmatrix} - \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ v_1 & 0 & \cdots & 0 & 0 \\ v_2 & v_1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ v_{n-1} & v_{n-2} & \cdots & v_1 & 0 \end{bmatrix} \begin{bmatrix} 0 & z_n & z_{n-1} & \cdots & z_1 \\ 0 & 0 & z_n & \cdots & z_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & & z_n \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

with $z_1 = v_n$.

Direct Toeplitz solvers

By cleverly computing the vectors $\mathbf{z}$ and $\mathbf{v}$ from the $\{t_n\}_n$ coefficients, one obtains several "fast" and "superfast" algorithms:

Algorithm	Complexity
Levinson 1946	$O(n^2)$
Trench 1964	$O(n^2)$
Zohar 1974	$O(n^2)$
Bitmead and Anderson 1980	$O(n \log^2(n))$
Brent, Gustavson, and Yun 1980	$O(n \log^2(n))$
Hoog 1987	$O(n \log^2(n))$
Ammar and Gragg 1988	$O(n \log^2(n))$
T. F. Chan and Hansen 1992	$O(n^2)$
Bini and Meini 1999	$O(n \log m + m \log^2 m \log n/m)$

n size of the matrix, m size of the bandwidth.

In our case

To treat our case

$$\left(I_N - \frac{\Delta t}{h_N^\alpha} \left[\theta G_N + (1-\theta) G_N^T \right] \right) \mathbf{w}^{n+1} = \mathbf{w}^n$$

we can then apply one of those algorithms (some of them use *symmetry*).

In our case

To treat our case

$$\left(I_N - \frac{\Delta t}{h_N^\alpha}\left[\theta G_N + (1-\theta) G_N^T\right]\right)\mathbf{w}^{n+1} = \mathbf{w}^n$$

we can then apply one of those algorithms (some of them use *symmetry*).

❓ What happens if we need to treat the case

$$\left(I_N - \frac{\Delta t}{h_N^\alpha}\left[D_n^{(1)} G_N + D_n^{(2)} G_N^T\right]\right)\mathbf{w}^{n+1} = \mathbf{w}^n$$

with $D_n^{(\cdot)}$ diagonal matrices coming from the discretization of **anisotropic space-variant diffusion coefficients**?

In our case

To treat our case

$$\left(I_N - \frac{\Delta t}{h_N^\alpha}\left[\theta G_N + (1-\theta) G_N^T\right]\right)\mathbf{w}^{n+1} = \mathbf{w}^n$$

we can then apply one of those algorithms (some of them use *symmetry*).

- What happens if we need to treat the case

$$\left(I_N - \frac{\Delta t}{h_N^\alpha}\left[D_n^{(1)} G_N + D_n^{(2)} G_N^T\right]\right)\mathbf{w}^{n+1} = \mathbf{w}^n$$

  with $D_n^{(\cdot)}$ diagonal matrices coming from the discretization of **anisotropic space-variant diffusion coefficients**?

- What happens if we need to treat **multi-dimensional cases**?

Krylov subspace methods

To overcome these challenges, we use an iterative approach based on **Krylov subspaces**.

Krylov subspace

A *Krylov subspace* $\mathcal{K}$ for the matrix A related to a non null vector $\mathbf{v}$ is defined as

$$\mathcal{K}_m(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \ldots, A^{m-1}\mathbf{v}\}.$$

Krylov subspace methods

To overcome these challenges, we use an iterative approach based on **Krylov subspaces**.

Krylov subspace

A *Krylov subspace* $\mathcal{K}$ for the matrix A related to a non null vector $\mathbf{v}$ is defined as

$$\mathcal{K}_m(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \ldots, A^{m-1}\mathbf{v}\}.$$

The fundamental operation is the **matrix-vector** product.

Krylov subspace methods

To overcome these challenges, we use an iterative approach based on **Krylov subspaces**.

Krylov subspace

A *Krylov subspace* $\mathcal{K}$ for the matrix A related to a non null vector $\mathbf{v}$ is defined as

$$\mathcal{K}_m(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \ldots, A^{m-1}\mathbf{v}\}.$$

- The fundamental operation is the **matrix-vector** product.
- Their use is *effective* when these **products are cheap**.

Krylov subspace methods

To overcome these challenges, we use an iterative approach based on **Krylov subspaces**.

Krylov subspace

A *Krylov subspace* $\mathcal{K}$ for the matrix A related to a non null vector $\mathbf{v}$ is defined as

$$\mathcal{K}_m(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \ldots, A^{m-1}\mathbf{v}\}.$$

- The fundamental operation is the **matrix-vector** product.
- Their use is *effective* when these **products are cheap**.
- We can compute $T_n(f)\mathbf{v}$ in $O(n\log(n))$ operations!

$$C_{2n}\begin{bmatrix}\mathbf{v}\\ \mathbf{0}_n\end{bmatrix} = \underbrace{\begin{bmatrix}T_n(f) & E_n\\ E_n & T_n(f)\end{bmatrix}}_{\text{Circulant}}\begin{bmatrix}\mathbf{v}\\ \mathbf{0}_n\end{bmatrix} = \begin{bmatrix}T_n(f)\mathbf{v}\\ E_n\mathbf{v}\end{bmatrix}, \quad E_n = \begin{bmatrix}0 & t_{n-1} & \cdots & t_2 & t_1\\ t_{1-n} & 0 & t_{n-1} & \cdots & t_2\\ \vdots & t_{1-n} & 0 & \ddots & \vdots\\ t_{-2} & \cdots & \ddots & \ddots & t_{n-1}\\ t_{-1} & t_{-2} & \cdots & t_{1-n} & 0\end{bmatrix}.$$

The Conjugate Gradient Method

When A is **symmetric positive definite** the method of choice is the **C**onjugate **G**radient.

Theorem.

Let A be SPD and $k_2(A) = \lambda_n/\lambda_1$ be the 2–norm condition number of A. We have:

$$\frac{\|\mathbf{r}^{(m)}\|_2}{\|\mathbf{r}^{(0)}\|_2} \leq \sqrt{k_2(A)}\frac{\|\mathbf{x}^* - \mathbf{x}^{(m)}\|_A}{\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_A}.$$

Corollary.

If A is SPD with eigenvalues $0 < \lambda_1 \leq \ldots \leq \lambda_n$, we have

$$\frac{\|\mathbf{x}^* - \mathbf{x}^{(m)}\|_A}{\|\mathbf{x}^* - \mathbf{x}^{(0)}\|_A} \leq 2\left(\frac{\sqrt{k_2(A)} - 1}{\sqrt{k_2(A)} + 1}\right)^m.$$

Input: $A \in \mathbb{R}^{n\times n}$ SPD, N_{max}, $\mathbf{x}^{(0)}$
Output: $\tilde{\mathbf{x}}$, candidate approximation.
$\mathbf{r}^{(0)} \leftarrow \|\mathbf{b} - A\mathbf{x}^{(0)}\|_2$, $\mathbf{r} = \mathbf{r}^{(0)}$, $\mathbf{p} \leftarrow \mathbf{r}$;
$\rho_0 \leftarrow \|\mathbf{r}^{(0)}\|^2$;
for $k = 1, \ldots, N_{max}$ **do**
  if $k = 1$ **then**
    $\mathbf{p} \leftarrow \mathbf{r}$;
  end
  else
    $\beta \leftarrow \rho_1/\rho_0$;
    $\mathbf{p} \leftarrow \mathbf{r} + \beta\,\mathbf{p}$;
  end
  $\mathbf{w} \leftarrow A\,\mathbf{p}$;
  $\alpha \leftarrow \rho_1/\mathbf{p}^T\mathbf{w}$;
  $\mathbf{x} \leftarrow \mathbf{x} + \alpha\mathbf{p}$;
  $\mathbf{r} \leftarrow \mathbf{r} - \alpha\mathbf{w}$;
  $\rho_1 \leftarrow \|\mathbf{r}\|_2^2$;
  if then
    Return: $\tilde{\mathbf{x}} = \mathbf{x}$;
  end
end

The Conjugate Gradient Method

⚠ The bound in the corollary is **descriptive** of the convergence behavior.

The Conjugate Gradient Method

⚠ The bound in the corollary is **descriptive** of the convergence behavior.

Theorem.

Let $A \in \mathbb{R}^{n \times n}$ be SPD. Let m an integer, $1 < m < n$ and $c > 0$ a constant such that for the eigenvalues of A we have

$$0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \ldots \leq \lambda_{n-m+1} \leq c < \ldots \leq \lambda_n.$$

Fixed $\varepsilon > 0$ an upper bound in exact arithmetic for the minimum number of iterations k reducing the relative error in energy norm form the approximation $\mathbf{x}^{(k)}$ generated by CG by ε is given by

$$\min\left\{\left\lceil \frac{1}{2}\sqrt{c/\lambda_1}\log\left(\frac{2}{\varepsilon}\right) + m + 1\right\rceil, n\right\}$$

The Conjugate Gradient Method

⚠ The bound in the corollary is **descriptive** of the convergence behavior.

Theorem.

Let $A \in \mathbb{R}^{n \times n}$ be SPD. Let m an integer, $1 < m < n$ and $c > 0$ a constant such that for the eigenvalues of A we have

$$0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \ldots \leq \lambda_{n-m+1} \leq c < \ldots \leq \lambda_n.$$

Fixed $\varepsilon > 0$ an upper bound in exact arithmetic for the minimum number of iterations k reducing the relative error in energy norm form the approximation $\mathbf{x}^{(k)}$ generated by CG by ε is given by

$$\min\left\{\left\lceil \frac{1}{2}\sqrt{c/\lambda_1}\log\left(\frac{2}{\varepsilon}\right) + m + 1\right\rceil, n\right\}$$

❓ How can we put ourselves in the hypotheses of the Theorem?

Clustered spectra

A proper cluster

A sequence of matrices $\{A_n\}_{n\geq 0}$, $A_n \in \mathbb{C}^{n\times n}$, has a **proper cluster** of eigenvalues in $p \in \mathbb{C}$ if, $\forall \varepsilon > 0$, if the number of eigenvalues of A_n **not in** $D(p, \varepsilon) = \{z \in \mathbb{C} \mid |z - p| < \varepsilon\}$ is bounded by a constant r that does not depend on n. Eigenvalues not in the *proper cluster* are called **outlier** eigenvalues.

Clustered spectra

A proper cluster

A sequence of matrices $\{A_n\}_{n\geq 0}$, $A_n \in \mathbb{C}^{n\times n}$, has a **proper cluster** of eigenvalues in $p \in \mathbb{C}$ if, $\forall \varepsilon > 0$, if the number of eigenvalues of A_n **not in** $D(p,\varepsilon) = \{z \in \mathbb{C} \mid |z-p| < \varepsilon\}$ is bounded by a constant r that does not depend on n. Eigenvalues not in the *proper cluster* are called **outlier** eigenvalues.

❓ Do the matrices

$$A_N = I_N - \frac{\Delta t}{2h_N^\alpha}\left[G_N + G_N^T\right]$$

have a **clustered spectra**?

Clustered spectra

A proper cluster

A sequence of matrices $\{A_n\}_{n\geq 0}$, $A_n \in \mathbb{C}^{n\times n}$, has a **proper cluster** of eigenvalues in $p \in \mathbb{C}$ if, $\forall \varepsilon > 0$, if the number of eigenvalues of A_n **not in** $D(p,\varepsilon) = \{z \in \mathbb{C} \mid |z-p| < \varepsilon\}$ is bounded by a constant r that does not depend on n. Eigenvalues not in the *proper cluster* are called **outlier** eigenvalues.

- Do the matrices

$$A_N = I_N - \frac{\Delta t}{2h_N^\alpha}\left[G_N + G_N^T\right]$$

  have a **clustered spectra**?

- We can investigate this question by looking again at the **spectral distribution** of the sequence $\{A_N\}_N$.

Asymptotic distribution: the symmetric case

$$A_N = I_N - \frac{\Delta t}{2h_N^\alpha}\left[G_N + G_N^T\right],$$

the sequence $\{A_N\}_N$ is **not** yet **ready** for the **analysis**, we have the coefficient $\Delta t/2h_N^\alpha$ that varies with N.

- For *consistency* reason it makes sense to select $\Delta t \equiv h_N \equiv \nu_N$, then, since $\alpha \in (1, 2]$ we have that $\nu^{1-\alpha}$ for $\nu \to 0^+$ goes to $+\infty$.

Asymptotic distribution: the symmetric case

$$A_N = I_N - \frac{\Delta t}{2h_N^\alpha}\left[G_N + G_N^T\right],$$

the sequence $\{A_N\}_N$ is **not** yet **ready** for the **analysis**, we have the coefficient $\Delta t/2h_N^\alpha$ that varies with N.

- ❗ For *consistency* reason it makes sense to select $\Delta t \equiv h_N \equiv \nu_N$, then, since $\alpha \in (1, 2]$ we have that $\nu^{1-\alpha}$ for $\nu \to 0^+$ goes to $+\infty$.

$\Rightarrow$ We look instead at the sequence:

$$\{\nu_N^{\alpha-1} A_N\}_N = \{\nu_N^{\alpha-1} I_N - (G_N + G_N^T)/2\}_N,$$

and is such that $\|\nu^{\alpha-1} I_N\| = \nu^{\alpha-1} < C$ independently of N.

Asymptotic distribution: the symmetric case

$$A_N = I_N - \frac{\Delta t}{2h_N^\alpha}\left[G_N + G_N^T\right],$$

the sequence $\{A_N\}_N$ is **not** yet **ready** for the **analysis**, we have the coefficient $\Delta t/2h_N^\alpha$ that varies with N.

- For *consistency* reason it makes sense to select $\Delta t \equiv h_N \equiv \nu_N$, then, since $\alpha \in (1,2]$ we have that $\nu^{1-\alpha}$ for $\nu \to 0^+$ goes to $+\infty$.

$\Rightarrow$ We look instead at the sequence:

$$\{\nu_N^{\alpha-1} A_N\}_N = \{\nu_N^{\alpha-1} I_N - (G_N + G_N^T)/2\}_N,$$

and is such that $\|\nu^{\alpha-1} I_N\| = \nu^{\alpha-1} < C$ independently of N.

- $\{-(G_N + G_N^T)/2\}_N$ is now a *symmetric* Toeplitz sequence with **known generating function**:

$$p_\alpha(\theta) = f(\theta) + f(-\theta) = -e^{-i\theta}(1-e^{i\theta})^\alpha - e^{i\theta}(1-e^{-i\theta})^\alpha.$$

Asymptotic distribution: the symmetric case

$$A_N = I_N - \frac{\Delta t}{2h_N^\alpha}\left[G_N + G_N^T\right],$$

the sequence $\{A_N\}_N$ is **not** yet **ready** for the **analysis**, we have the coefficient $\Delta t/2h_N^\alpha$ that varies with N.

- For *consistency* reason it makes sense to select $\Delta t \equiv h_N \equiv \nu_N$, then, since $\alpha \in (1,2]$ we have that $\nu^{1-\alpha}$ for $\nu \to 0^+$ goes to $+\infty$.

$\Rightarrow$ We look instead at the sequence:

$$\{\nu_N^{\alpha-1} A_N\}_N = \{\nu_N^{\alpha-1} I_N - (G_N + G_N^T)/2\}_N,$$

and is such that $\|\nu^{\alpha-1} I_N\| = \nu^{\alpha-1} < C$ independently of N.

- $\{-(G_N + G_N^T)/2\}_N$ is now a *symmetric* Toeplitz sequence with **known generating function**:

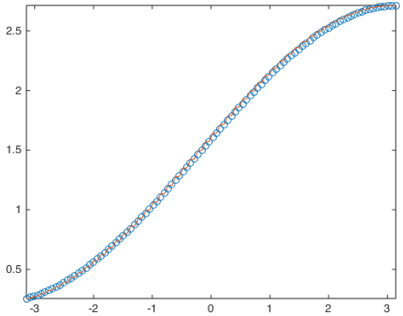
$$p_\alpha(\theta) = f(\theta) + f(-\theta) = -e^{-i\theta}(1-e^{i\theta})^\alpha - e^{i\theta}(1-e^{-i\theta})^\alpha.$$

$\Rightarrow$ We have just discovered that: $\{\nu_N^{\alpha-1} A_N\}_N \sim_\lambda p_\alpha(\theta)$.

Asymptotic distribution: the symmetric case

$$\{\nu_N^{\alpha-1} A_N\} = \left\{ \nu_N^{\alpha-1} I_N - \frac{1}{2}\left[G_N + G_N^T\right] \right\}_N \sim_\lambda p_\alpha(\theta) = -e^{-i\theta}(1-e^{i\theta})^\alpha - e^{i\theta}(1-e^{-i\theta})^\alpha,$$

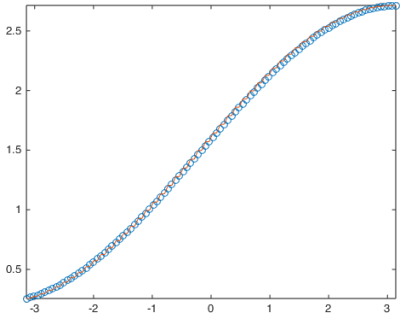
```
N = 100; alpha = 1.3;
hN = 1/(N-1); dt = hN; nu = dt;
G = glmatrix(N,alpha); I = eye(N,N);
A = nu^(alpha-1)*I-0.5*(G+G');
ev = eig(A);
f = @(t)-exp(-1i*t).*(1-exp(1i*t))
↪  .^alpha;
p = @(t)nu^(alpha-1)+0.5*(f(t)
↪  +conj(f(t)));
t = linspace(-pi,pi,N);
plot(t,ev,'o',t,sort(p(t),'ascend'),'-')
```



Asymptotic distribution: the symmetric case

$$\{\nu_N^{\alpha-1} A_N\} = \left\{ \nu_N^{\alpha-1} I_N - \frac{1}{2}\left[G_N + G_N^T\right] \right\}_N \sim_\lambda p_\alpha(\theta) = -e^{-i\theta}(1-e^{i\theta})^\alpha - e^{i\theta}(1-e^{-i\theta})^\alpha,$$

```
N = 100; alpha = 1.3;
hN = 1/(N-1); dt = hN; nu = dt;
G = glmatrix(N,alpha); I = eye(N,N);
A = nu^(alpha-1)*I-0.5*(G+G');
ev = eig(A);
f = @(t)-exp(-1i*t).*(1-exp(1i*t))
↪  .^alpha;
p = @(t)nu^(alpha-1)+0.5*(f(t)
↪  +conj(f(t)));
t = linspace(-pi,pi,N);
plot(t,ev,'o',t,sort(p(t),'ascend'),'-')
```



the spectrum is **not clustered**!

CG with a non clustered spectra

Let us test the CG with different values of α and N.

α	1.8	1.5	1.2
N	Iteration		
100	49	34	16
200	87	42	17
500	155	53	18
1000	209	63	19
5000	398	92	21
10000	523	108	22

- The number if iterations grows with N,
- Smaller values of α seem to be easier.

```
A = nu^(alpha-1)*I-0.5*(G+G'); b = nu^(alpha-1)*ones(N,1);
[x,flag,relres,iter,resvec] = pcg(A,b,1e-6,N)
```

CG with a non clustered spectra

Let us test the CG with different values of α and N.

α	1.8	1.5	1.2
N	Iteration		
100	49	34	16
200	87	42	17
500	155	53	18
1000	209	63	19
5000	398	92	21
10000	523	108	22

- The number if iterations grows with N,
- Smaller values of α seem to be easier.
- We would like **number of iterations** independent on both **size** and value of α. In this case this is called having a method with a **superlinear convergence** and **robust with respect to the parameters**.

```
A = nu^(alpha-1)*I-0.5*(G+G'); b = nu^(alpha-1)*ones(N,1);
[x,flag,relres,iter,resvec] = pcg(A,b,1e-6,N)
```

CG with a non clustered spectra

Let us test the CG with different values of α and N.

α	1.8	1.5	1.2
N	Iteration		
100	49	34	16
200	87	42	17
500	155	53	18
1000	209	63	19
5000	398	92	21
10000	523	108	22

- The number if iterations grows with N,
- Smaller values of α seem to be easier.
- We would like **number of iterations** independent on both **size** and value of α. In this case this is called having a method with a **superlinear convergence** and **robust with respect to the parameters**.
- Can we?

```
A = nu^(alpha-1)*I-0.5*(G+G'); b = nu^(alpha-1)*ones(N,1);
[x,flag,relres,iter,resvec] = pcg(A,b,1e-6,N)
```

Preconditioned CG

- To try and achieve this result we need to *modify the spectrum of the system*, i.e., we need to **precondition**.

Preconditioned CG

- To try and achieve this result we need to *modify the spectrum of the system*, i.e., we need to **precondition**.
- We modify the system

$$A\mathbf{x} = \mathbf{b},$$

into

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b},$$

with M SPD and such that $M^{-1}A$ has a **clustered spectra**.

Input: $A \in \mathbb{R}^{n\times n}$ SPD, N_{max}, $\mathbf{x}^{(0)}$, $M \in \mathbb{R}^{n\times n}$ SPD preconditioner

$\mathbf{r}^{(0)} \leftarrow \mathbf{b} - A\mathbf{x}^{(0)}$, $\mathbf{z}^{(0)} \leftarrow M^{-1}\mathbf{r}^{(0)}$, $\mathbf{p}^{(0)} \leftarrow \mathbf{z}^{(0)}$;

for $j = 0, \ldots, N_{max}$ **do**

  $\alpha_j \leftarrow {<\mathbf{r}^{(j)},\mathbf{z}^{(j)}>}/{A\mathbf{p}^{(j)},\mathbf{p}^{(j)}}$;

  $\mathbf{x}^{(j+1)} \leftarrow \mathbf{x}^{(j)} + \alpha_j\mathbf{p}^{(j)}$;

  $\mathbf{r}^{(j+1)} \leftarrow \mathbf{r}^{(j)} - \alpha_j A\mathbf{p}^{(j)}$;

  if then

    Return: $\tilde{\mathbf{x}} = \mathbf{x}^{(j+1)}$;

  end

  $\mathbf{z}^{(j+1)} \leftarrow M^{-1}\mathbf{r}^{(j+1)}$;

  $\beta_j \leftarrow {<\mathbf{r}^{(j+1)},\mathbf{z}^{(j+1)}>}/{<\mathbf{r}^{(j)},\mathbf{z}^{(j)}>}$;

  $\mathbf{p}^{(j+1)} \leftarrow \mathbf{z}^{(j+1)} + \beta_j\mathbf{p}^{(j)}$;

end

Preconditioned CG

- To try and achieve this result we need to *modify the spectrum of the system*, i.e., we need to **precondition**.
- We modify the system

$$A\mathbf{x} = \mathbf{b},$$

into

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b},$$

with M SPD and such that $M^{-1}A$ has a **clustered spectra**.

Input: $A \in \mathbb{R}^{n\times n}$ SPD, N_{max}, $\mathbf{x}^{(0)}$, $M \in \mathbb{R}^{n\times n}$ SPD preconditioner

$\mathbf{r}^{(0)} \leftarrow \mathbf{b} - A\mathbf{x}^{(0)}$, $\mathbf{z}^{(0)} \leftarrow M^{-1}\mathbf{r}^{(0)}$, $\mathbf{p}^{(0)} \leftarrow \mathbf{z}^{(0)}$;

for $j = 0, \dots, N_{max}$ **do**

  $\alpha_j \leftarrow <\mathbf{r}^{(j)},\mathbf{z}^{(j)}>/_{A\mathbf{p}^{(j)},\mathbf{p}^{(j)}}$;

  $\mathbf{x}^{(j+1)} \leftarrow \mathbf{x}^{(j)} + \alpha_j\mathbf{p}^{(j)}$;

  $\mathbf{r}^{(j+1)} \leftarrow \mathbf{r}^{(j)} - \alpha_j A\mathbf{p}^{(j)}$;

  if then

    Return: $\tilde{\mathbf{x}} = \mathbf{x}^{(j+1)}$;

  end

  $\mathbf{z}^{(j+1)} \leftarrow M^{-1}\mathbf{r}^{(j+1)}$;

  $\beta_j \leftarrow <\mathbf{r}^{(j+1)},\mathbf{z}^{(j+1)}>/_{<\mathbf{r}^{(j)},\mathbf{z}^{(j)}>}$;

  $\mathbf{p}^{(j+1)} \leftarrow \mathbf{z}^{(j+1)} + \beta_j\mathbf{p}^{(j)}$;

end

- M^{-1} has to be easy to apply, possibly it has to have the *same cost of multiplying by A*.

Circulant preconditioners for Toeplitz matrices

- If M is circulant than applying M^{-1} costs $O(n \log n)$ operations, same as applying A.

Circulant preconditioners for Toeplitz matrices

💡 If M is circulant than applying M^{-1} costs $O(n \log n)$ operations, same as applying A.
👁 Observe that, nevertheless, this **doubles the cost per iteration**, can we do better?

Circulant preconditioners for Toeplitz matrices

💡 If M is circulant than applying M^{-1} costs $O(n \log n)$ operations, same as applying A.
👁 Observe that, nevertheless, this **doubles the cost per iteration**, can we do better?

ω-circulant matrices

Let $\omega = \exp(i\theta)$ for $\theta \in [-\pi, \pi]$. A matrix $W_n^{(\omega)}$ of size n is said to be an ω**–circulant matrix** if it has the spectral decomposition

$$W_n^{(\omega)} = \Omega_n^H F_n^H \Lambda_n F_n \Omega_n,$$

where F_n is the Fourier matrix and $\Omega_n = \mathrm{diag}(1, \omega^{-1/n}, \ldots, \omega^{-(n-1)/n})$ and Λ_n is the diagonal matrix of the eigenvalues. In particular 1–circulant matrices are circulant matrices while $\{-1\}$–circulant matrices are the skew–circulant matrices.

Circulant preconditioners for Toeplitz matrices

💡 If M is circulant than applying M^{-1} costs $O(n \log n)$ operations, same as applying A.
👁 Observe that, nevertheless, this **doubles the cost per iteration**, can we do better?

ω-circulant matrices

Let $\omega = \exp(i\theta)$ for $\theta \in [-\pi, \pi]$. A matrix $W_n^{(\omega)}$ of size n is said to be an ω**–circulant matrix** if it has the spectral decomposition

$$W_n^{(\omega)} = \Omega_n^H F_n^H \Lambda_n F_n \Omega_n,$$

where F_n is the Fourier matrix and $\Omega_n = \mathrm{diag}(1, \omega^{-1/n}, \ldots, \omega^{-(n-1)/n})$ and Λ_n is the diagonal matrix of the eigenvalues. In particular 1–circulant matrices are circulant matrices while $\{-1\}$–circulant matrices are the skew–circulant matrices.

💡 We can use them to reduce the overall cost of the preconditioning step!

Circulant preconditioners for Toeplitz matrices

The 🔑 key idea is observing that we can decompose any Toeplitz matrix into the **sum of a circulant and of a skew-circulant matrix**

$$T_n = U_n + V_n,\ U_n = F_n^H \Lambda_n^{(1)} F_n,\ V_n = \Omega_n^H F_n^H \Lambda_n^{(2)} F_n \Omega_n$$

where

$$\begin{aligned} \mathbf{e}_1^T U_n &= 1/2 \left[t_0, t_{-1} + t_{n-1}, \ldots, t_{-(n-1)+t_1} \right], \\ W_n \mathbf{e}_1 &= 1/2 \left[t_0, -(t_{n-1} - t_{-1}), \ldots, -(t_{-1} - t_{n-1}) \right]^T. \end{aligned}$$

Circulant preconditioners for Toeplitz matrices

The 🔑 key idea is observing that we can decompose any Toeplitz matrix into the **sum of a circulant and of a skew-circulant matrix**

$$T_n = U_n + V_n,\ U_n = F_n^H \Lambda_n^{(1)} F_n,\ V_n = \Omega_n^H F_n^H \Lambda_n^{(2)} F_n \Omega_n$$

where

$$\begin{aligned} \mathbf{e}_1^T U_n &= 1/2 \left[t_0, t_{-1} + t_{n-1}, \ldots, t_{-(n-1)+t_1} \right], \\ W_n \mathbf{e}_1 &= 1/2 \left[t_0, -(t_{n-1} - t_{-1}), \ldots, -(t_{-1} - t_{n-1}) \right]^T. \end{aligned}$$

Then we can compute the product

$$\begin{aligned} C_n^{-1} T_n = C_n^{-1} (U_n + V_n) &= C_n^{-1} \left(F_n^H \Lambda_n^{(1)} F_n + \Omega_n^H F_n^H \Lambda_n^{(2)} F_n \Omega_n \right) \\ &= F_n^H \Lambda_n^{-1} F_n \left(F_n^H \Lambda_n^{(1)} F_n + \Omega_n^H F_n^H \Lambda_n^{(2)} F_n \Omega_n \right) \\ &= F_n^H \left[\Lambda_n^{-1} \left(\Lambda_n^{(1)} + F_n \Omega_n^H F_n^H \Lambda_n^{(2)} F_n \Omega_n F_n^H \right) \right] F_n. \end{aligned}$$

Circulant preconditioners for Toeplitz matrices

The 🔑 key idea is observing that we can decompose any Toeplitz matrix into the **sum of a circulant and of a skew-circulant matrix**

$$T_n = U_n + V_n, \;\; U_n = F_n^H \Lambda_n^{(1)} F_n, \;\; V_n = \Omega_n^H F_n^H \Lambda_n^{(2)} F_n \Omega_n$$

where

$$\begin{aligned} \mathbf{e}_1^T U_n &= 1/2 \left[t_0, t_{-1} + t_{n-1}, \ldots, t_{-(n-1)+t_1} \right], \\ W_n \mathbf{e}_1 &= 1/2 \left[t_0, -(t_{n-1} - t_{-1}), \ldots, -(t_{-1} - t_{n-1}) \right]^T. \end{aligned}$$

Then we can compute the product

$$C_n^{-1} T_n = F_n^H \left[\Lambda_n^{-1} \left(\Lambda_n^{(1)} + F_n \Omega_n^H F_n^H \Lambda_n^{(2)} F_n \Omega_n F_n^H \right) \right] F_n.$$

And solve $C_n^{-1} T_n \mathbf{x} = C_n^{-1} \mathbf{b}$ as

$$\Lambda_n^{-1} \left(\Lambda_n^{(1)} + F_n \Omega_n^H F_n^H \Lambda_n^{(2)} F_n \Omega_n F_n^H \right) \underbrace{F_n \mathbf{x}}_{=\tilde{\mathbf{x}}} = \underbrace{\Lambda_n^{-1} F_n \mathbf{b}}_{=\tilde{\mathbf{b}}}$$

Circulant preconditioners for Toeplitz matrices

The 🔑 key idea is observing that we can decompose any Toeplitz matrix into the **sum of a circulant and of a skew-circulant matrix**

$$T_n = U_n + V_n, \;\; U_n = F_n^H \Lambda_n^{(1)} F_n, \;\; V_n = \Omega_n^H F_n^H \Lambda_n^{(2)} F_n \Omega_n$$

where

$$\begin{aligned} \mathbf{e}_1^T U_n &= 1/2 \left[t_0, t_{-1} + t_{n-1}, \ldots, t_{-(n-1)+t_1} \right], \\ W_n \mathbf{e}_1 &= 1/2 \left[t_0, -(t_{n-1} - t_{-1}), \ldots, -(t_{-1} - t_{n-1}) \right]^T. \end{aligned}$$

Then we can compute the product

$$C_n^{-1} T_n = F_n^H \left[\Lambda_n^{-1} \left(\Lambda_n^{(1)} + F_n \Omega_n^H F_n^H \Lambda_n^{(2)} F_n \Omega_n F_n^H \right) \right] F_n.$$

And solve $C_n^{-1} T_n \mathbf{x} = C_n^{-1} \mathbf{b}$ as

$$\Lambda_n^{-1} \left(\Lambda_n^{(1)} + F_n \Omega_n^H F_n^H \Lambda_n^{(2)} F_n \Omega_n F_n^H \right) \underbrace{F_n \mathbf{x}}_{=\tilde{\mathbf{x}}} = \underbrace{\Lambda_n^{-1} F_n \mathbf{b}}_{=\tilde{\mathbf{b}}}$$ 4 FFTs per iteration!

Circulant preconditioners for Toeplitz matrices

❓ This is then *computationally efficient*, can we find the right circulant matrix to have a clustered spectra?

Circulant preconditioners for Toeplitz matrices

❓ This is then *computationally efficient*, can we find the right circulant matrix to have a clustered spectra?
🔧 We need to change the problem into an equivalent one: the aim is **discharging everything on the generating functions**!

Circulant preconditioners for Toeplitz matrices

❓ This is then *computationally efficient*, can we find the right circulant matrix to have a clustered spectra?

🔧 We need to change the problem into an equivalent one: the aim is **discharging everything on the generating functions**!

Continuous convolution

Given two scalar functions f and g in the Schwartz space, i.e., $f, g \in \mathcal{C}^\infty(\mathbb{R})$ such that $\exists\, C^{(f)}_{\alpha,\beta}, C^{(g)}_{\alpha',\beta'} \in \mathbb{R}$ with $\|x^\alpha \partial_\beta f(x)\|_\infty \leq C^{\alpha\beta}$ and $\|x^{\alpha'} \partial_{\beta'} g(x)\|_\infty \leq C^{\alpha'\beta'}$, α, β, α', β' scalar indices, we define the **convolution operation**, "$*$", as

$$[f * g](t) = \int_{-\infty}^{+\infty} f(\tau) g(t-\tau)\, d\tau = \int_{-\infty}^{+\infty} g(\tau) f(t-\tau)\, d\tau.$$

Circulant preconditioners for Toeplitz matrices

❓ This is then *computationally efficient*, can we find the right circulant matrix to have a clustered spectra?

🔧 We need to change the problem into an equivalent one: the aim is **discharging everything on the generating functions**!

Discrete convolution

For two arbitrary 2π–periodic continuous functions,

$$f(\theta) = \sum_{k=-\infty}^{+\infty} t_k e^{ik\theta} \text{ and } g = \sum_{k=-\infty}^{+\infty} s_k e^{ik\theta}$$

their **convolution product** is given by

$$[f * g](\theta) = \sum_{k=-\infty}^{+\infty} s_k t_k e^{ik\theta}.$$

Circulant preconditioners for Toeplitz matrices

❓ This is then *computationally efficient*, can we find the right circulant matrix to have a clustered spectra?

🔧 We need to change the problem into an equivalent one: the aim is **discharging everything on the generating functions**!

Using a Kernel

Given a kernel $\mathcal{K}_n(\theta)$ defined on $[0, 2\pi]$ and a generating function f for a Toeplitz sequence $T_n(f)$, we consider the circulant matrix C_n with eigenvalues given by

$$\lambda_j(C_n) = [\mathcal{K}_n * f]\left(\frac{2\pi j}{n}\right), 0 \leq j < n,$$

Circulant preconditioners for Toeplitz matrices

❓ This is then *computationally efficient*, can we find the right circulant matrix to have a clustered spectra?

🔧 We need to change the problem into an equivalent one: the aim is **discharging everything on the generating functions**!

Using a Kernel

Given a kernel $\mathcal{K}_n(\theta)$ defined on $[0, 2\pi]$ and a generating function f for a Toeplitz sequence $T_n(f)$, we consider the circulant matrix C_n with eigenvalues given by

$$\lambda_j(C_n) = [\mathcal{K}_n * f]\left(\frac{2\pi j}{n}\right), 0 \leq j < n,$$

- We have rewritten the problem of **finding an appropriate preconditioner** to the problem of **approximating the generating function** of the underlying Toeplitz matrix.

Circulant preconditioners for Toeplitz matrices

Theorem (R. H. Chan and Yeung 1992)

Lef f be a 2π–periodic continuous positive function. Let $\mathcal{K}_n(\theta)$ be a kernel such that $\mathcal{K}_n * f \overset{n\to+\infty}{\longrightarrow} f$ uniformly on $[-\pi, \pi]$. If C_n is the sequence of circulant matrices with eigenvalues given by

$$\lambda_j(C_n) = [\mathcal{K}_n * f]\left(\frac{2\pi j}{n}\right), 0 \leq j < n,$$

then the spectra of the sequence $\{C_n^{-1} T_n(f)\}_n$ is clustered around 1.

❓ Is this the result we need?

Circulant preconditioners for Toeplitz matrices

Theorem (R. H. Chan and Yeung 1992)

Lef f be a 2π–periodic continuous positive function. Let $\mathcal{K}_n(\theta)$ be a kernel such that $\mathcal{K}_n * f \overset{n\to+\infty}{\longrightarrow} f$ uniformly on $[-\pi, \pi]$. If C_n is the sequence of circulant matrices with eigenvalues given by

$$\lambda_j(C_n) = [\mathcal{K}_n * f]\left(\frac{2\pi j}{n}\right), 0 \leq j < n,$$

then the spectra of the sequence $\{C_n^{-1} T_n(f)\}_n$ is clustered around 1.

- ? Is this the result we need?
- ! It requires a *continuous positive function* generating function f! Ours is:

$$p_\alpha(\theta) = -e^{-i\theta}(1 - e^{i\theta})^\alpha - e^{i\theta}(1 - e^{-i\theta})^\alpha,$$

and it does seem to have a zero.

Circulant preconditioners: cases with a zero

Order of the zero

Let $f : [a, b] \subset \mathbb{R} \to \mathbb{R}$ be a continuous nonnegative function. We say that f has a zero order $\beta > 0$ at $\theta_0 \in [a, b]$ if there exist two real constants $C_1, C_2 > 0$ such that

$$\liminf_{\theta \to \theta_0} \frac{f(\theta)}{|\theta - \theta_0|^\beta} = C_1, \quad \limsup_{\theta \to \theta_0} \frac{f(\theta)}{|\theta - \theta_0|^\beta} = C_2.$$

Circulant preconditioners: cases with a zero

Order of the zero

Let $f : [a, b] \subset \mathbb{R} \to \mathbb{R}$ be a continuous nonnegative function. We say that f has a zero order $\beta > 0$ at $\theta_0 \in [a, b]$ if there exist two real constants $C_1, C_2 > 0$ such that

$$\liminf_{\theta \to \theta_0} \frac{f(\theta)}{|\theta - \theta_0|^\beta} = C_1, \quad \limsup_{\theta \to \theta_0} \frac{f(\theta)}{|\theta - \theta_0|^\beta} = C_2.$$

Proposition (Donatelli, Mazza, and Serra-Capizzano 2016)

Given $\alpha \in (1, 2)$, then the function $p_\alpha(\theta)$ is nonnegative and has a zero of order α at 0.

Circulant preconditioners: cases with a zero

Order of the zero

Let $f : [a, b] \subset \mathbb{R} \to \mathbb{R}$ be a continuous nonnegative function. We say that f has a zero order $\beta > 0$ at $\theta_0 \in [a, b]$ if there exist two real constants $C_1, C_2 > 0$ such that

$$\liminf_{\theta \to \theta_0} \frac{f(\theta)}{|\theta - \theta_0|^\beta} = C_1, \quad \limsup_{\theta \to \theta_0} \frac{f(\theta)}{|\theta - \theta_0|^\beta} = C_2.$$

Proposition (Donatelli, Mazza, and Serra-Capizzano 2016)

Given $\alpha \in (1, 2)$, then the function $p_\alpha(\theta)$ is nonnegative and has a zero of order α at 0.

Proof. We first prove that is nonnegative by direct computation

$$p_\alpha(\theta) = -\sum_{k=-1}^{+\infty} g_{k+1}^{(\alpha)} (e^{ik\theta} + e^{-ik\theta})$$

Circulant preconditioners: cases with a zero

Order of the zero

Let $f : [a, b] \subset \mathbb{R} \to \mathbb{R}$ be a continuous nonnegative function. We say that f has a zero order $\beta > 0$ at $\theta_0 \in [a, b]$ if there exist two real constants $C_1, C_2 > 0$ such that

$$\liminf_{\theta \to \theta_0} \frac{f(\theta)}{|\theta - \theta_0|^\beta} = C_1, \quad \limsup_{\theta \to \theta_0} \frac{f(\theta)}{|\theta - \theta_0|^\beta} = C_2.$$

Proposition (Donatelli, Mazza, and Serra-Capizzano 2016)

Given $\alpha \in (1, 2)$, then the function $p_\alpha(\theta)$ is nonnegative and has a zero of order α at 0.

Proof. We first prove that is nonnegative by direct computation

$$p_\alpha(\theta) = -\left[2g_1^{(\alpha)} + (g_0^{(\alpha)} + g_2^{(\alpha)})(e^{i\theta} + e^{-i\theta}) + \sum_{k=2}^{+\infty} g_{k+1}^{(\alpha)}(e^{ik\theta} + e^{-ik\theta})\right]$$

Circulant preconditioners: cases with a zero

Order of the zero

Let $f : [a, b] \subset \mathbb{R} \to \mathbb{R}$ be a continuous nonnegative function. We say that f has a zero order $\beta > 0$ at $\theta_0 \in [a, b]$ if there exist two real constants $C_1, C_2 > 0$ such that

$$\liminf_{\theta \to \theta_0} \frac{f(\theta)}{|\theta - \theta_0|^\beta} = C_1, \quad \limsup_{\theta \to \theta_0} \frac{f(\theta)}{|\theta - \theta_0|^\beta} = C_2.$$

Proposition (Donatelli, Mazza, and Serra-Capizzano 2016)

Given $\alpha \in (1, 2)$, then the function $p_\alpha(\theta)$ is nonnegative and has a zero of order α at 0.

Proof. We first prove that is nonnegative by direct computation

$$p_\alpha(\theta) = -\left[2g_1^{(\alpha)} + 2(g_0^{(\alpha)} + g_2^{(\alpha)})\cos\theta + 2\sum_{k=2}^{+\infty} g_{k+1}^{(\alpha)} cos(k\theta)\right]$$

Circulant preconditioners: cases with a zero

Order of the zero

Let $f : [a, b] \subset \mathbb{R} \to \mathbb{R}$ be a continuous nonnegative function. We say that f has a zero order $\beta > 0$ at $\theta_0 \in [a, b]$ if there exist two real constants $C_1, C_2 > 0$ such that

$$\liminf_{\theta \to \theta_0} \frac{f(\theta)}{|\theta - \theta_0|^\beta} = C_1, \quad \limsup_{\theta \to \theta_0} \frac{f(\theta)}{|\theta - \theta_0|^\beta} = C_2.$$

Proposition (Donatelli, Mazza, and Serra-Capizzano 2016)

Given $\alpha \in (1, 2)$, then the function $p_\alpha(\theta)$ is nonnegative and has a zero of order α at 0.

Proof. We first prove that is nonnegative by direct computation

$$p_\alpha(\theta) = -\left[2g_1^{(\alpha)} + 2(g_0^{(\alpha)} + g_2^{(\alpha)})\cos\theta + 2\sum_{k=2}^{+\infty} g_{k+1}^{(\alpha)} cos(k\theta)\right] \geq -2\sum_{k=-1}^{+\infty} g_{k+1}^{(\alpha)} = 0.$$

Circulant preconditioners: cases with a zero

Proposition (Donatelli, Mazza, and Serra-Capizzano 2016)

Given $\alpha \in (1,2)$, then the function $p_\alpha(\theta)$ is nonnegative and has a zero of order α at 0.

Proof. Then we focus on the zero. Let us rewrite

$$1 - e^{i\theta} = \sqrt{2 - 2\cos\theta}\, e^{i\phi}, \quad 1 - e^{-i\theta} = \sqrt{2 - 2\cos\theta}\, e^{i\psi},$$

where

$$\phi = \begin{cases} \arctan\left(\frac{-\sin\theta}{1-\cos\theta}\right), & \theta \neq 0, \\ \lim_{\theta\to 0^+} \arctan\left(\frac{-\sin\theta}{1-\cos\theta}\right) = -\frac{\pi}{2}, & \theta = 0. \end{cases} \quad \psi = -\phi.$$

Circulant preconditioners: cases with a zero

Proposition (Donatelli, Mazza, and Serra-Capizzano 2016)

Given $\alpha \in (1,2)$, then the function $p_\alpha(\theta)$ is nonnegative and has a zero of order α at 0.

Proof. Then we focus on the zero. Let us rewrite

$$1 - e^{i\theta} = \sqrt{2 - 2\cos\theta}\, e^{i\phi}, \quad 1 - e^{-i\theta} = \sqrt{2 - 2\cos\theta}\, e^{i\psi},$$

and write

$$\begin{aligned} p_\alpha(\theta) &= -\, e^{-i\theta}(\sqrt{2 - 2\cos\theta}\, e^{i\phi})^\alpha - e^{i\theta}(\sqrt{2 - 2\cos\theta}\, e^{-i\phi})^\alpha \\ &= -\sqrt{(2 - 2\cos\theta)^\alpha}\, e^{i(\alpha\phi - \theta)} - \sqrt{(2 - 2\cos\theta)^\alpha}\, e^{-i(\alpha\phi - \theta)} \\ &= -2\sqrt{(2 - 2\cos\theta)^\alpha}\, r_\alpha(\theta), \qquad r_\alpha(\theta) = \cos(\alpha\phi - \theta). \end{aligned}$$

Circulant preconditioners: cases with a zero

Proposition (Donatelli, Mazza, and Serra-Capizzano 2016)

Given $\alpha \in (1,2)$, then the function $p_\alpha(\theta)$ is nonnegative and has a zero of order α at 0.

Proof. Then we focus on the zero. Let us rewrite

$$1 - e^{i\theta} = \sqrt{2 - 2\cos\theta}\, e^{i\phi}, \quad 1 - e^{-i\theta} = \sqrt{2 - 2\cos\theta}\, e^{i\psi},$$

and write

$$p_\alpha(\theta) = -2\sqrt{(2 - 2\cos\theta)^\alpha}\, r_\alpha(\theta), \qquad r_\alpha(\theta) = \cos(\alpha\phi - \theta).$$

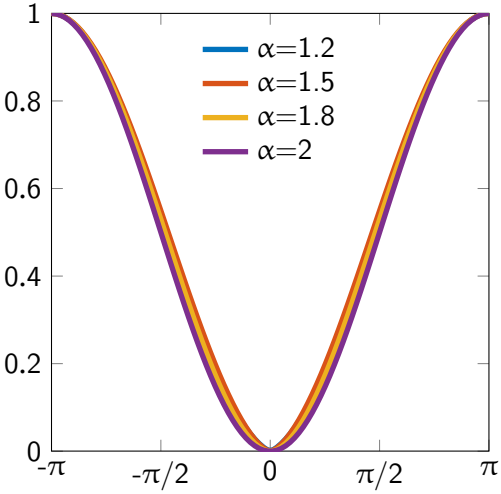
Since $\lim_{\theta\to 0^-} r_\alpha(\theta) = \lim_{\theta\to 0^+} r_\alpha(\theta) = \cos(\alpha\pi/2)$, we find

$$\lim_{\theta\to 0} \frac{p_\alpha(\theta)}{|\theta|^\alpha} = -2 \lim_{\theta\to 0} \frac{(2 - 2\cos\theta)^{\alpha/2}}{|\theta|^\alpha} r_\alpha(\theta) = -2\cos(\alpha\pi/2) \in (0,2),$$

i.e., p_α has a zero of order α at 0 according to the definition. $\square$

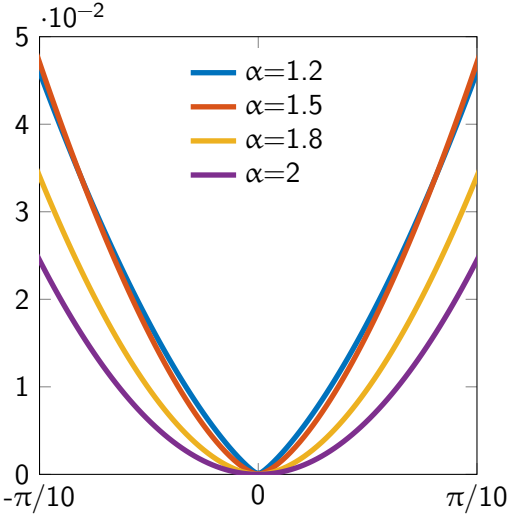
Circulant preconditioners: cases with a zero

```
t = linspace(-pi,pi,100);
f = @(alpha)
 ↪  -exp(-1i*t).*(1-exp(1i*t)).^alpha;
p = @(alpha) f(alpha) +
 ↪  conj(f(alpha));
plot(t,p(1.2)./max(p(1.2)),...
 t,p(1.5)./max(p(1.5)),...
 t,p(1.8)./max(p(1.8)),
 t,p(2)./max(p(2)),...
 'LineWidth',2);
legend({'\alpha=1.2','\alpha=1.5',...
 '\alpha=1.8','\alpha=2'},...
 'Location','north');
```



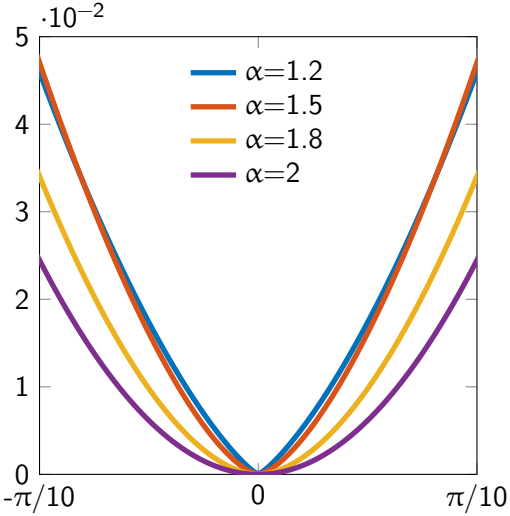
Circulant preconditioners: cases with a zero

- $p_2(\theta) = 2(2 - 2\cos\theta)$, i.e., $2\times$Laplacian generating function,
- $p_\alpha(\theta)/\|p_\alpha\|_\infty$ approaches the order of the zero of the Laplacian in 0, i.e., it increases up to 2 as α tends to 2.



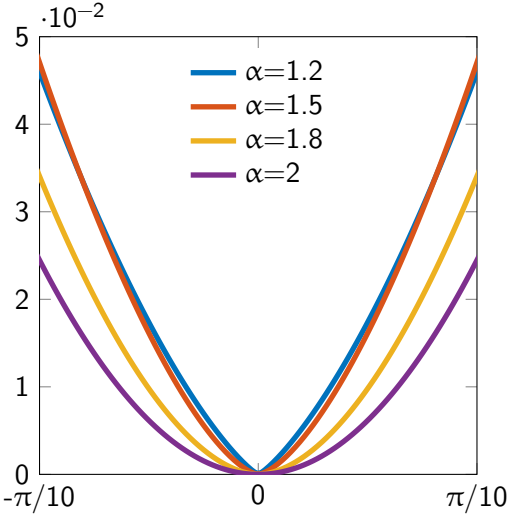
Circulant preconditioners: cases with a zero

- $p_2(\theta) = 2(2 - 2\cos\theta)$, i.e., $2\times$Laplacian generating function,
- $p_\alpha(\theta)/\|p_\alpha\|_\infty$ approaches the order of the zero of the Laplacian in 0, i.e., it increases up to 2 as α tends to 2.
- What can we do for the case in this case?



Circulant preconditioners: cases with a zero

- $p_2(\theta) = 2(2 - 2\cos\theta)$, i.e., $2\times$Laplacian generating function,
- $p_\alpha(\theta)/\|p_\alpha\|_\infty$ approaches the order of the zero of the Laplacian in 0, i.e., it increases up to 2 as α tends to 2.
- What can we do for the case in this case?
- **matching the zeros** of the generating function, *heuristically*, if the preconditioner have a spectrum that behaves as a function g with zeros of the same order, and in the same place of f, then f/g no loner have the problematic behavior...



Generalized Jackson Kernel

Generalized Jackson Kernel

Given $\theta \in [-\pi, \pi]$, $\mathbb{N} \ni r \geq 1$ and $\mathbb{N} \ni m > 0$ such that $r(m-1) < n \leq rm$, i.e., $m = \lceil n/r \rceil$, the generalized Jackson kernel function is defined as,

$$\mathcal{K}_{m,2r}(\theta) = \frac{k_{m,2r}}{m^{2r-1}} \left(\frac{\sin(m\theta/2)}{\sin(\theta/2)} \right)^{2r}, \; k_{m,2r} \text{ s.t. } \frac{1}{2\pi} \int_{-\pi}^{\pi} \mathcal{K}_{m,2r}(\theta) d\theta = 1.$$

We build a **circulant preconditioner** $J_{n,m,r}$ from its eigenvalues using the Jackson kernel

$$\lambda_j(J_{n,m,r}) = [\mathcal{K}_{m,2r} * f] \left(\frac{2j\pi}{n} \right), \; j = 0, \ldots, n-1.$$

Generalized Jackson Kernel

Theorem (R. H. Chan, Ng, and Yip 2002)

Let f be a nonnegative 2π–periodic continuous function with a zero of order 2ν at θ_0. Let $r > \nu$ and $m = \lceil n/r \rceil$. Then there exists numbers a, b independent from n and such that the spectrum of $J_{n,m,r}^{-1} T_n(f)$ is clustered in $[a, b]$ and at most $2\nu + 1$ eigenvalues are not in $[a, b]$ for n sufficiently large.

We build a **circulant preconditioner** $J_{n,m,r}$ from its eigenvalues using the Jackson kernel

$$\lambda_j(J_{n,m,r}) = [\mathcal{K}_{m,2r} * f]\left(\frac{2j\pi}{n}\right), \quad j = 0, \ldots, n-1.$$

Generalized Jackson Kernel

Theorem (R. H. Chan, Ng, and Yip 2002)

Let f be a nonnegative 2π–periodic continuous function with a zero of order 2ν at θ_0. Let $r > \nu$ and $m = \lceil n/r \rceil$. Then there exists numbers a, b independent from n and such that the spectrum of $J_{n,m,r}^{-1} T_n(f)$ is clustered in $[a, b]$ and at most $2\nu + 1$ eigenvalues are not in $[a, b]$ for n sufficiently large.

We build a **circulant preconditioner** $J_{n,m,r}$ from its eigenvalues using the Jackson kernel

$$\lambda_j(J_{n,m,r}) = [\mathcal{K}_{m,2r} * f]\left(\frac{2j\pi}{n}\right), \quad j = 0, \ldots, n-1.$$

With some work can be **generalized** to the case of **multiple zeros of different order**,

Generalized Jackson Kernel

Theorem (R. H. Chan, Ng, and Yip 2002)

Let f be a nonnegative 2π–periodic continuous function with a zero of order 2ν at θ_0. Let $r > \nu$ and $m = \lceil n/r \rceil$. Then there exists numbers a, b independent from n and such that the spectrum of $J_{n,m,r}^{-1} T_n(f)$ is clustered in $[a, b]$ and at most $2\nu + 1$ eigenvalues are not in $[a, b]$ for n sufficiently large.

We build a **circulant preconditioner** $J_{n,m,r}$ from its eigenvalues using the Jackson kernel

$$\lambda_j(J_{n,m,r}) = [\mathcal{K}_{m,2r} * f]\left(\frac{2j\pi}{n}\right), \quad j = 0, \ldots, n-1.$$

- With some work can be **generalized** to the case of **multiple zeros of different order**,
- One can prove also that a and b are **bounded away from zero**.

Time to do some tests

We consider the following **circulant preconditioners**,

Dirichlet kernel, a.k.a. the Strang circulant preconditioner

$$\mathcal{D}_n(\theta) = \frac{\sin\left((n+1/2)\theta\right)}{\sin\left(\theta/2\right)} \qquad \begin{cases} t_k, & 0 < k \leq \lfloor n/2 \rfloor, \\ t_{k-n}, & \lfloor n/2 \rfloor < j < n, \\ c_{n+k}, & 0 < -k < n. \end{cases}$$

Modified Dirichlet kernel, a.k.a. the T. Chan circulant preconditioner

$$1/2\left(\mathcal{D}_{n-1}(\theta) + \mathcal{D}_{n-2}(\theta)\right) \qquad \begin{cases} t_1 + 1/2\bar{t}_{n-1}, & k = 1, \\ t_k + t_{n-k}, & 2 \leq k \leq n-2, \\ 1/2 t_{n-1} + \bar{t}_1, & k = n-1. \end{cases}$$

R.H. Chan $\mathcal{D}_{n-1}(\theta)$ $\quad t_k + \bar{t}_{n-k},\ 0 < k \leq n-1.$

Jackson with $r = 2$.

Time to do some tests

We consider the following **circulant preconditioners**,

Dirichlet kernel, a.k.a. the Strang circulant preconditioner

```
c = fft([t(1:n/2);0;conj(t(n/2:-1:2))].')';
```

Modified Dirichlet kernel, a.k.a. the T. Chan circulant preconditioner

```
coef = (1/n:1/n:1-1/n)';
c = fft([t(1);(1-coef).*t(2:n)+coef.*t1]);
```

R.H. Chan

```
c = fft([t(1);t(2:n)+t1].')';
```

Jackson with $r = 2$.

Time to do some tests

We consider the following **circulant preconditioners**,

Dirichlet kernel, a.k.a. the Strang circulant preconditioner

```
c = fft([t(1:n/2);0;conj(t(n/2:-1:2))].')';
```

Modified Dirichlet kernel, a.k.a. the T. Chan circulant preconditioner

```
coef = (1/n:1/n:1-1/n)';
c = fft([t(1);(1-coef).*t(2:n)+coef.*t1]);
```

R.H. Chan

```
c = fft([t(1);t(2:n)+t1].')';
```

Jackson with $r = 2$.

We test both **clustering properties** and **convergence behavior** inside the **P**reconditioned **C**onjugate **G**radient algorithm.

Jackson Kernel Circulant Preconditioner

For $r = 2, 3, 4$ it can be built as

```
n = length(t);
t1 = conj(t(n:-1:2));
if r == 2 || r == 3 || r == 4
 coef = convol(n,r).';
 c = [t(1)*coef(1)
↪  (coef(2:n).*t(2:n)...
 +coef(n:-1:2).*t1).'];
 c = fft(c)';
else
 error('r needs to be 2, 3 or 4');
end
c = real(c);
```

```
function [ c ] = jacksonprec( t,r )
m = floor(n/r); a = 1:-1/m:1/m; r0 = 1;
coef = [a(m:-1:2) a];
while r0 < r
 M = (2*r0+3)*m; b1 = zeros(M,1);
 c = zeros(M,1); c(1:m) = a;
 c(M:-1:M-m+2) = a(2:m);
 b1(m:m+2*r0*(m-1)) = coef;
 tp = ifft(fft(b1).*fft(c));
 coef = real(tp(1:2*(r0+1)*(m-1)+1));
 r0 = r0+1;
end
M = r*(m-1)+1;
coef = [coef(M:-1:1)' zeros(1,n-M)]';
coef = coef';
end
```

Back to the example

We try to solve again

$$\begin{cases} \frac{\partial W}{\partial t} = \theta\, {}^{RL}D^{\alpha}_{[0,x]} W(x,t) + (1-\theta)\, {}^{RL}D^{\alpha}_{[x,1]} W(x,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \\ W(x,t) = W_0(x). \end{cases}$$

Back to the example

We try to solve again for $\theta = 1/2$

$$T_{N-2}(p_\alpha(\theta))\mathbf{w}^{n+1} \equiv \left(\frac{h_N^\alpha}{\Delta t} I_{N-2} - \frac{1}{2}\left[G_{N-2} + G_{N-2}^T\right]\right)\mathbf{w}^{n+1} = \frac{h_N^\alpha}{\Delta t}\mathbf{w}^n$$

We have removed the *rank corrections* due to the boundary conditions to have a **pure Toeplitz** matrix, i.e., we solve the equation only in the inner nodes.

Back to the example

We try to solve again

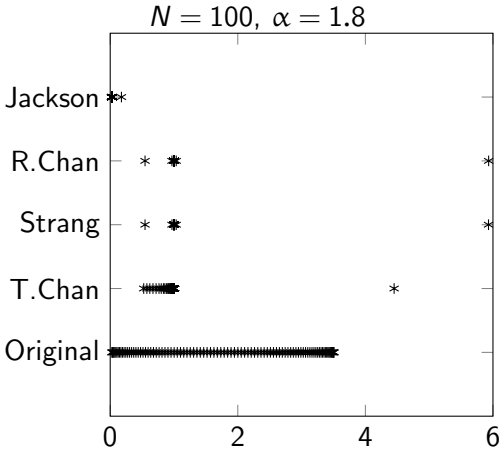
$$T_{N-2}(p_\alpha(\theta))\mathbf{w}^{n+1} \equiv \left(\frac{h_N^\alpha}{\Delta t} I_{N-2} - \frac{1}{2}\left[G_{N-2} + G_{N-2}^T\right]\right)\mathbf{w}^{n+1} = \frac{h_N^\alpha}{\Delta t}\mathbf{w}^n$$

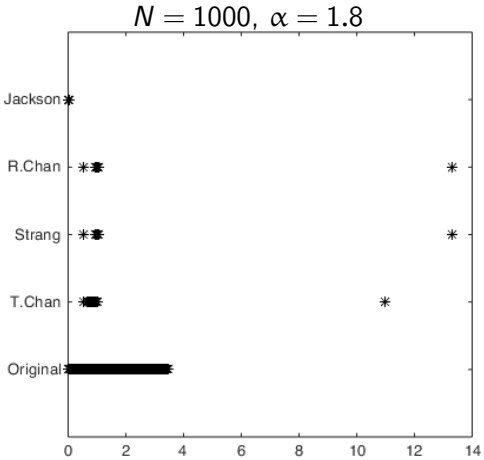
We have removed the *rank corrections* due to the boundary conditions to have a **pure Toeplitz** matrix, i.e., we solve the equation only in the inner nodes.

```
%% Problem data
theta = 0.5;
alpha = 1.8;
w0 = @(x) 5*x.*(1-x);
%% Discretization data
N  = 10;
hN = 1/(N-1); x = 0:hN:1;
dt = hN; t = 0:dt:1;
```

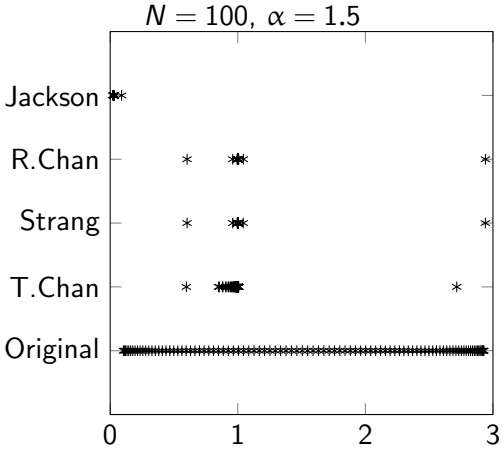
```
%% Discretize
G = glmatrix(N,alpha);
Gr = G(2:N-1,2:N-1); Grt = Gr.';
I = eye(N-2,N-2);
% Left-hand side
nu = hN^alpha/dt;
A = nu*I - (theta*Gr + (1-theta)*Grt);
% Right-hand side
w = w0(x).';
```

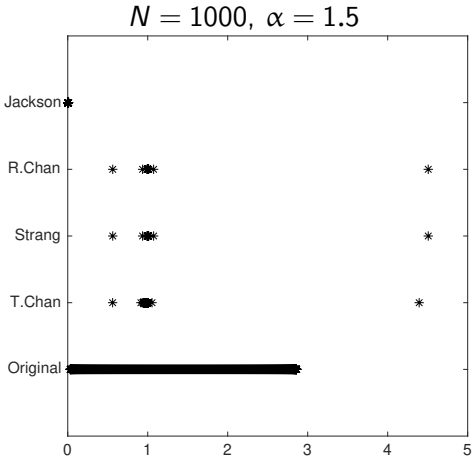
A look at the spectrum



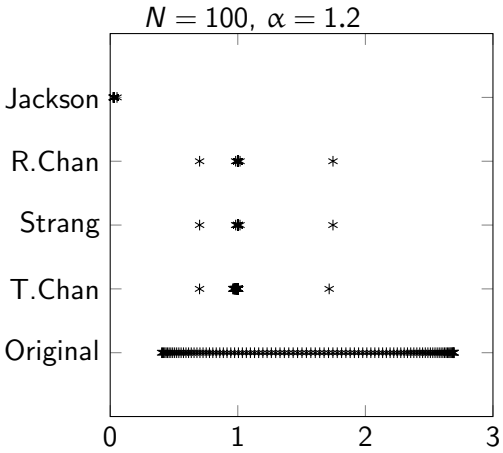


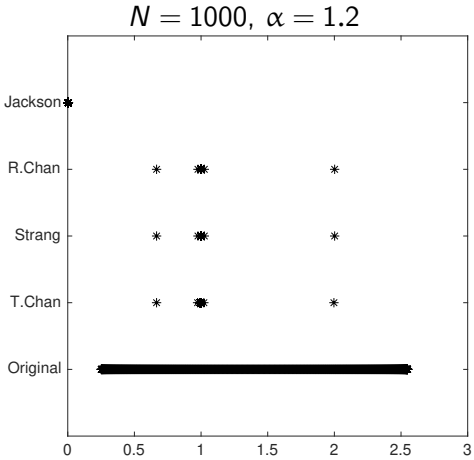
A look at the spectrum



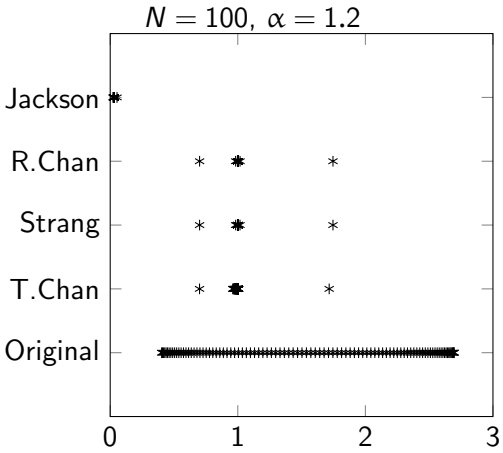


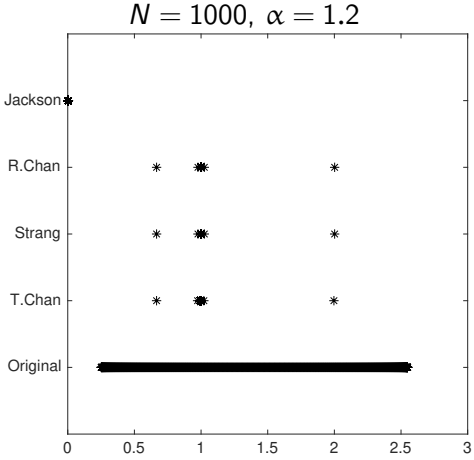
A look at the spectrum





A look at the spectrum

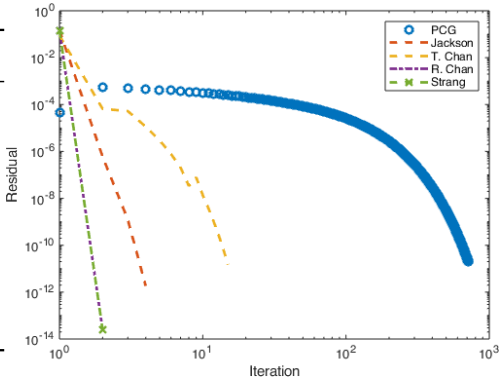




❓ Can you guess what is happening with the Jackson Kernel preconditioner?

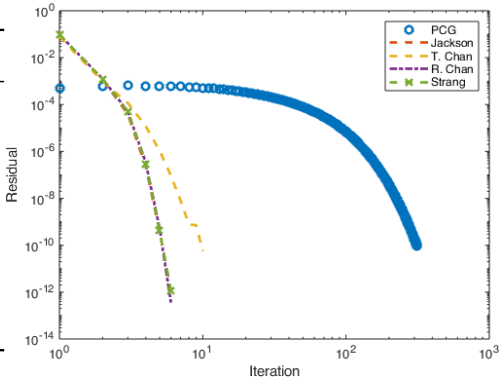
↓ A look at the convergence

α	N	PCG	Jackson	T.Chan	R.Chan	Strang
	2^5	15	6	8	2	2
	2^6	31	6	10	2	2
	2^7	63	6	12	2	2
2.0	2^8	127	5	13	2	2
	2^9	251	5	14	2	2
	2^{10}	464	5	15	2	2
	2^{11}	713	4	15	2	2



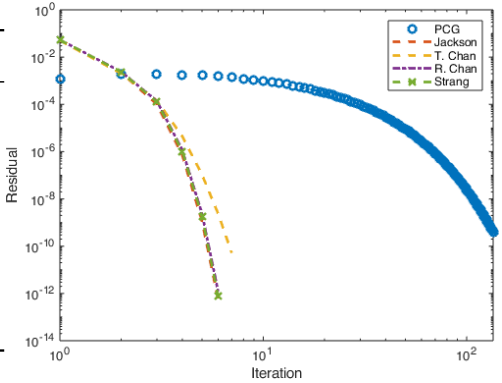
↓ A look at the convergence

α	N	PCG	Jackson	T.Chan	R.Chan	Strang
	2^5	15	6	8	5	5
	2^6	31	6	9	5	5
	2^7	61	6	9	5	5
1.8	2^8	108	6	11	5	5
	2^9	174	6	11	6	5
	2^{10}	234	6	11	6	6
	2^{11}	314	6	10	6	6



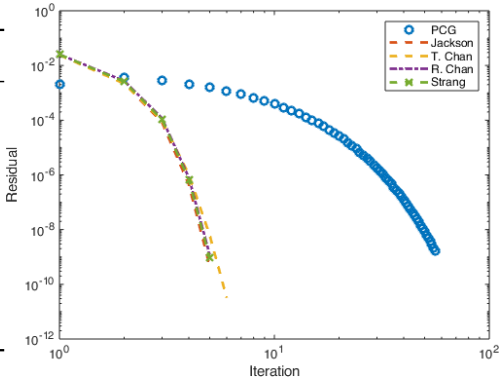
↓ A look at the convergence

α	N	PCG	Jackson	T.Chan	R.Chan	Strang
	2^5	15	6	7	5	5
	2^6	31	6	8	5	5
	2^7	51	6	8	5	5
1.6	2^8	73	5	8	5	5
	2^9	91	5	8	6	5
	2^{10}	111	6	7	6	6
	2^{11}	135	6	7	6	6



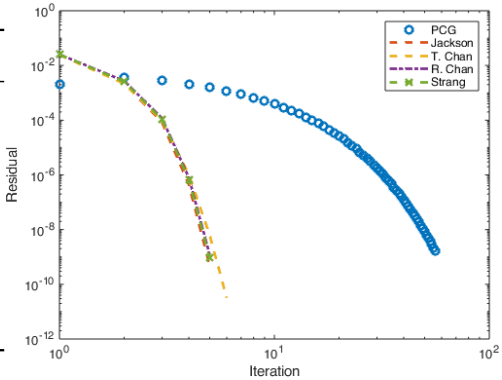
↓ A look at the convergence

α	N	PCG	Jackson	T.Chan	R.Chan	Strang
	2^5	15	5	7	5	5
	2^6	27	5	7	5	5
	2^7	35	5	7	5	5
1.4	2^8	41	5	6	5	5
	2^9	46	5	6	5	5
	2^{10}	51	5	6	5	5
	2^{11}	56	5	6	5	5



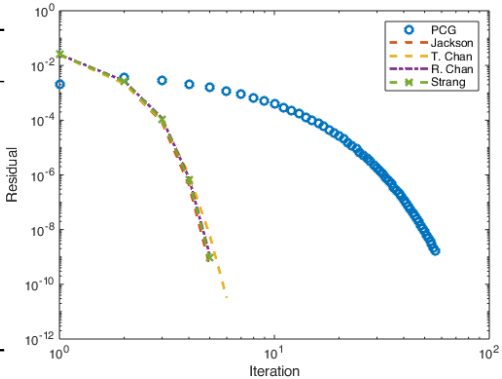
↓ A look at the convergence

α	N	PCG	Jackson	T.Chan	R.Chan	Strang
	2^5	15	5	6	4	4
	2^6	19	5	6	5	5
	2^7	20	5	5	5	5
1.2	2^8	21	5	5	5	5
	2^9	22	5	5	5	5
	2^{10}	22	5	5	5	5
	2^{11}	22	5	5	5	5



↓ A look at the convergence

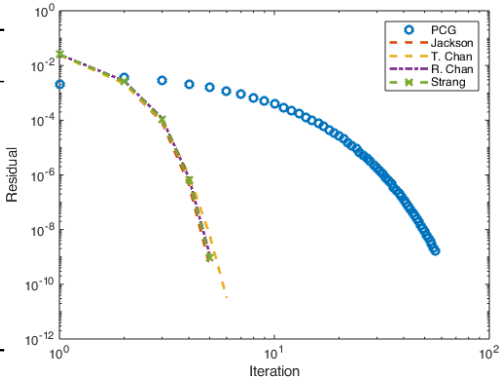
α	N	PCG	Jackson	T.Chan	R.Chan	Strang
	2^5	15	5	6	4	4
	2^6	19	5	6	5	5
	2^7	20	5	5	5	5
1.2	2^8	21	5	5	5	5
	2^9	22	5	5	5	5
	2^{10}	22	5	5	5	5
	2^{11}	22	5	5	5	5



We got **robustness** with respect to both α and N.

↓ A look at the convergence

α	N	PCG	Jackson	T.Chan	R.Chan	Strang
	2^5	15	5	6	4	4
	2^6	19	5	6	5	5
	2^7	20	5	5	5	5
1.2	2^8	21	5	5	5	5
	2^9	22	5	5	5	5
	2^{10}	22	5	5	5	5
	2^{11}	22	5	5	5	5



- We got **robustness** with respect to both α and N.
- What do we do in the non symmetric case, i.e., $\theta \neq 1/2$?

Non symmetric Toeplitz system

If $T_n(f)$ is non symmetric (or more generally, non Hermitian), then f is a complex-valued function then

- we **no longer** have information on the asymptotic **spectral distribution**, but only on the singular values,
- we can **no longer** apply **fast** direct Toeplitz **solvers**,
- we can **no longer** apply the **CG** to $T_n(f)\mathbf{x} = \mathbf{b}$.

? What to do?

Non symmetric Toeplitz system

If $T_n(f)$ is non symmetric (or more generally, non Hermitian), then f is a complex-valued function then

- we **no longer** have information on the asymptotic **spectral distribution**, but only on the singular values,
- we can **no longer** apply **fast** direct Toeplitz **solvers**,
- we can **no longer** apply the **CG** to $T_n(f)\mathbf{x} = \mathbf{b}$.

? What to do?

Apply the PCG to the normal equations (CGNR):

$$T_n(f)^H T_n(f)\mathbf{x} = T_n(f)^H \mathbf{b},$$

Non symmetric Toeplitz system

If $T_n(f)$ is non symmetric (or more generally, non Hermitian), then f is a complex-valued function then

- we **no longer** have information on the asymptotic **spectral distribution**, but only on the singular values,
- we can **no longer** apply **fast** direct Toeplitz **solvers**,
- we can **no longer** apply the **CG** to $T_n(f)\mathbf{x} = \mathbf{b}$.

What to do?

- Apply the PCG to the normal equations (CGNR):

$$T_n(f)^H T_n(f)\mathbf{x} = T_n(f)^H \mathbf{b},$$

- Use another Krylov method: GMRES or TFQMR

Non symmetric Toeplitz system

If $T_n(f)$ is non symmetric (or more generally, non Hermitian), then f is a complex-valued function then

- we **no longer** have information on the asymptotic **spectral distribution**, but only on the singular values,
- we can **no longer** apply **fast** direct Toeplitz **solvers**,
- we can **no longer** apply the **CG** to $T_n(f)\mathbf{x} = \mathbf{b}$.

? What to do?

- Apply the PCG to the normal equations (CGNR):

$$T_n(f)^H T_n(f)\mathbf{x} = T_n(f)^H \mathbf{b},$$

- Use another Krylov method: GMRES or TFQMR
  ? do we know how to precondition these methods?

The GMRES method (Saad and Schultz 1986)

The **G**eneralized **M**inimum **Res**idual (GMRES) is a Krylov projection method approximating the solution of linear system

$$A\mathbf{x} = \mathbf{b}$$

on the **affine subspace**

$$\mathbf{x}^{(0)} + \mathcal{K}_m(A, \mathbf{v}_1), \quad \mathbf{r}^{(0)} = \mathbf{b} - A\,\mathbf{x}^{(0)}, \quad \mathbf{v}_1 = \mathbf{r}^{(0)}/\|\mathbf{r}^{(0)}\|_2$$

, for $\mathbf{x}^{(0)}$ a *starting guess* for the solution.
By this choice, we enforce the **Arnoldi relation**:

$$A\,V_m = V_m H_m + \mathbf{w}_m \mathbf{e}_m^T = V_{m+1}\overline{H}_m, \quad \mathrm{Span}\,V_m = \mathrm{Span}\{\mathbf{v}_1 \ \cdots \ \mathbf{v}_m\} = \mathcal{K}_m(A, \mathbf{v}_1),$$

and H_m $m \times m$ Hessenberg submatrix extracted from $\overline{H}_m$ by deleting the $(m+1)$th line.

The GMRES method (Saad and Schultz 1986)

Input: $A \in \mathbb{R}^{n\times n}$,$\mathbf{b} \in \mathbb{R}^n$, m, $\mathbf{x}^{(0)}$

$\mathbf{r}^{(0)} \leftarrow \mathbf{b} - A\,\mathbf{x}^{(0)}$, $\beta \leftarrow \|\mathbf{r}^{(0)}\|_2$;

$\mathbf{v}_1 \leftarrow {}^{\mathbf{r}^{(0)}}/\beta$;

for $j = 1, \ldots, m$ **do**

  $\mathbf{w}_j \leftarrow A\,\mathbf{v}_j$;

  for $i = 1, \ldots, j$ **do**

    $h_{i,j} \leftarrow < \mathbf{w}_j, \mathbf{v}_i >$;

    $\mathbf{w}_j \leftarrow \mathbf{w}_j - h_{i,j}\,\mathbf{v}_i$;

  end

  $h_{j+1,j} \leftarrow \|\mathbf{w}_j\|_2$;

  if $h_{j+1,j} = 0$ *or convergence*

  then

    $m = j$;

    break;

  end

  $\mathbf{v}_{j+1} = {}^{\mathbf{w}_j}/\|\mathbf{w}_j\|_2$;

end

Compute $\mathbf{y}^{(m)}$ such that $\|\mathbf{r}^{(m)}\|_2 = \|\mathbf{b} - A\,\mathbf{x}^{(m)}\|_2 = \|\beta\mathbf{e}_1 - \underline{H}_m\mathbf{y}\|_2 = \min_{\mathbf{y}\in\mathbb{R}^m}$;

Build candidate approximation $\tilde{\mathbf{x}}$;

The GMRES method (Saad and Schultz 1986)

Input: $A \in \mathbb{R}^{n\times n}$,$\mathbf{b} \in \mathbb{R}^n$, m, $\mathbf{x}^{(0)}$
$\mathbf{r}^{(0)} \leftarrow \mathbf{b} - A\,\mathbf{x}^{(0)}$, $\beta \leftarrow \|\mathbf{r}^{(0)}\|_2$;
$\mathbf{v}_1 \leftarrow {}^{\mathbf{r}^{(0)}}/\beta$;
for $j = 1, \ldots, m$ **do**
  $\mathbf{w}_j \leftarrow A\,\mathbf{v}_j$;
  for $i = 1, \ldots, j$ **do**
    $h_{i,j} \leftarrow < \mathbf{w}_j, \mathbf{v}_i >$;
    $\mathbf{w}_j \leftarrow \mathbf{w}_j - h_{i,j}\,\mathbf{v}_i$;
  end
  $h_{j+1,j} \leftarrow \|\mathbf{w}_j\|_2$;
  if $h_{j+1,j} = 0$ *or convergence*
  then
    $m = j$;
    break;
  end
  $\mathbf{v}_{j+1} = {}^{\mathbf{w}_j}/\|\mathbf{w}_j\|_2$;
end

Compute $\mathbf{y}^{(m)}$ such that $\|\mathbf{r}^{(m)}\|_2 = \|\mathbf{b} - A\,\mathbf{x}^{(m)}\|_2 = \|\beta\mathbf{e}_1 - \underline{H}_m\mathbf{y}\|_2 = \min_{\mathbf{y}\in\mathbb{R}^m}$;
Build candidate approximation $\tilde{\mathbf{x}}$;

Minimizing the residual

At step m, the candidate solution $\mathbf{x}^{(m)}$ is the vector minimizing the 2–norm residual:

$$\|\mathbf{r}^{(m)}\|_2 = \|\mathbf{b} - A\,\mathbf{x}^{(m)}\|_2,$$

with

$$\mathbf{b} - A\,\mathbf{x}^{(m)} = V_{m+1}(\beta\mathbf{e}_1 - \overline{H}_m\mathbf{y}).$$

The GMRES method (Saad and Schultz 1986)

Input: $A \in \mathbb{R}^{n\times n}, \mathbf{b} \in \mathbb{R}^n$, m, $\mathbf{x}^{(0)}$
$\mathbf{r}^{(0)} \leftarrow \mathbf{b} - A\,\mathbf{x}^{(0)}$, $\beta \leftarrow \|\mathbf{r}^{(0)}\|_2$;
$\mathbf{v}_1 \leftarrow \mathbf{r}^{(0)}/\beta$;
for $j = 1, \ldots, m$ **do**
  $\mathbf{w}_j \leftarrow A\,\mathbf{v}_j$;
  for $i = 1, \ldots, j$ **do**
    $h_{i,j} \leftarrow < \mathbf{w}_j, \mathbf{v}_i >$;
    $\mathbf{w}_j \leftarrow \mathbf{w}_j - h_{i,j}\,\mathbf{v}_i$;
  end
  $h_{j+1,j} \leftarrow \|\mathbf{w}_j\|_2$;
  if $h_{j+1,j} = 0$ *or convergence* **then**
    $m = j$;
    break;
  end
  $\mathbf{v}_{j+1} = \mathbf{w}_j/\|\mathbf{w}_j\|_2$;
end

Compute $\mathbf{y}^{(m)}$ such that $\|\mathbf{r}^{(m)}\|_2 = \|\mathbf{b} - A\,\mathbf{x}^{(m)}\|_2 = \|\beta\mathbf{e}_1 - \underline{H}_m\mathbf{y}\|_2 = \min_{\mathbf{y}\in\mathbb{R}^m}$;
Build candidate approximation $\tilde{\mathbf{x}}$;

Minimizing the residual

At step m, the candidate solution $\mathbf{x}^{(m)}$ is the vector minimizing the 2–norm residual:

$$\|\mathbf{r}^{(m)}\|_2 = \|\mathbf{b} - A\,\mathbf{x}^{(m)}\|_2,$$

with

$$\mathbf{b} - A\,\mathbf{x}^{(m)} = V_{m+1}(\beta\mathbf{e}_1 - \overline{H}_m\mathbf{y}).$$

GMRES variants

Variants obtained by different least square problem solutions, and different orthogonalization algorithms.

The GMRES convergence theory (or lack thereof...)

Theorem (Convergence, diagonalizable)

If A can be diagonalized, i.e. if we can find $X \in \mathbb{R}^{n\times n}$ non singular and such that

$$A = X\,\Lambda\,X^{-1},\ \Lambda = \mathrm{diag}(\lambda_1,\dots,\lambda_n),\ K_2(X) = \|X\|_2\,\|X^{-1}\|_2,$$

$K_2(X) = \|X\|_2\,\|X^{-1}\|_2$ condition number of X, then at step m, we have

$$\|r\|_2 \le K_2(X)\|\mathbf{r}^{(0)}\|_2 \min_{\substack{\mathrm{p}(z)\in\mathbb{P}_m\\ \mathrm{p}(0)=1}} \max_{i=1,\dots,n} |\mathrm{p}(\lambda_i)|, \qquad \text{(DiagGMRES)}$$

where $\mathrm{p}(z)$ is the polynomial of degree less or equal to m such that $\mathrm{p}(0) = 1$ and the expression in the right hand side of (DiagGMRES) is minimum.

- ⚠ The eigenvectors can be arbitrarily *ill-conditioned*, i.e., $K_2(X) \ggg 1$,
- ❗ being **diagonalizable** can be a **strong assumption**.

The GMRES convergence theory (or lack thereof...)

Theorem (Almostr everything is possible) (Greenbaum, Pták, and Strakoš 1996)

Given a non-increasing positive sequence $\{f_k\}_{k=0,\dots,n-1}$ with $f_{n-1} > 0$ and a set of non–zero complex numbers $\{\lambda_i\}_{i=1,2,\dots,n} \subset \mathbb{C}$, there exist a matrix A with eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ and a right-hand side $\mathbf{b}$ with $\|\mathbf{b}\| = f_0$ such that the residual vectors $\mathbf{r}^{(k)}$ at each step of the GMRES algorithm applied to solve $A\mathbf{x} = \mathbf{b}$ with $\mathbf{x}^{(0)} = \mathbf{0}$, satisfy $\|\mathbf{r}^{(k)}\| = f_k$, $\forall\, k = 1, 2, \dots, n-1$.

- "Any non-increasing convergence curve is possible for GMRES".

- In the clustered case we can partition $\sigma(A)$ as follows

$$\sigma(A) = \sigma_c(A) \cup \sigma_0(A) \cup \sigma_1(A),$$

where

- $\sigma_c(A)$ denotes the **clustered set** of eigenvalues of A,
- $\sigma_0(A) \cup \sigma_1(A)$ denotes the **set of the outliers**.

The GMRES convergence theory (or lack thereof...)

Theorem (Almostr everything is possible) (Greenbaum, Pták, and Strakoš 1996)

Given a non-increasing positive sequence $\{f_k\}_{k=0,\dots,n-1}$ with $f_{n-1} > 0$ and a set of non–zero complex numbers $\{\lambda_i\}_{i=1,2,\dots,n} \subset \mathbb{C}$, there exist a matrix A with eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ and a right-hand side $\mathbf{b}$ with $\|\mathbf{b}\| = f_0$ such that the residual vectors $\mathbf{r}^{(k)}$ at each step of the GMRES algorithm applied to solve $A\mathbf{x} = \mathbf{b}$ with $\mathbf{x}^{(0)} = \mathbf{0}$, satisfy $\|\mathbf{r}^{(k)}\| = f_k$, $\forall\, k = 1, 2, \dots, n-1$.

- "Any non-increasing convergence curve is possible for GMRES".
- What happens if we have **a clustered spectrum**?
- In the clustered case we can partition $\sigma(A)$ as follows

$$\sigma(A) = \sigma_c(A) \cup \sigma_0(A) \cup \sigma_1(A),$$

where

- $\sigma_c(A)$ denotes the **clustered set** of eigenvalues of A,
- $\sigma_0(A) \cup \sigma_1(A)$ denotes the **set of the outliers**.

GMRES in the clustered and diagonalizable case

$$\sigma(A) = \underbrace{\sigma_c(A)}_{\text{clustered}} \cup \underbrace{\sigma_0(A) \cup \sigma_1(A)}_{\text{outliers}},$$

we assume that

1. the clustered set $\sigma_c(A)$ of eigenvalues is contained in a convex set Ω,
2. and, that denoting two sets of j_0 and j_1 outliers as

$$\sigma_0(A) = \{\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_{j_0}\} \quad \text{and} \quad \sigma_1(A) = \{\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_{j_1}\}$$

where if $\hat{\lambda}_j \in \sigma_0(A)$, we have

$$1 < |1 - z/\hat{\lambda}_j| \leq c_j, \quad \forall z \in \Omega,$$

while, for $\tilde{\lambda}_j \in \sigma_1(A)$,

$$0 < |1 - z/\tilde{\lambda}_j| < 1, \quad \forall z \in \Omega,$$

GMRES in the clustered and diagonalizable case

Theorem

The number of full GMRES iterations j needed to attain a tolerance ε on the relative residual in the 2-norm $\|\mathbf{r}^{(j)}\|_2/\|\mathbf{r}^{(0)}\|_2$ for the linear system $A\mathbf{x} = \mathbf{b}$, where A is diagonalizable, is bounded above by

$$\min\left\{ j_0 + j_1 + \left\lceil \frac{\log(\varepsilon) - \log(\kappa_2(X))}{\log(\rho)} - \sum_{\ell=1}^{j_0} \frac{\log(c_\ell)}{\log(\rho)} \right\rceil, n \right\},$$

where

$$\rho^k = \frac{\left(a/d + \sqrt{(a/d)^2 - 1}\right)^k + \left(a/d + \sqrt{(a/d)^2 - 1}\right)^{-k}}{\left(c/d + \sqrt{(c/d)^2 - 1}\right)^k + \left(c/d + \sqrt{(c/d)^2 - 1}\right)^{-k}},$$

and the set $\Omega \in \mathbb{C}^+$ is the ellipse with center c, focal distance d and major semi axis a.

GMRES the non-diagonalizable case

In this case we have to turn to either the **field of values** or the ε-**pseudospectra** of A. We need to bound the right-hand side of

$$\|\mathbf{r}_m\|_2 \leq \min_{\substack{\mathrm{p}(z)\in\mathbb{P}_m \\ \mathrm{p}(0)=1}} \|\mathrm{p}(A)\mathbf{r}_0\|, \quad m = 1, 2, \ldots$$

or in the worst case scenario

$$\frac{\|\mathbf{r}_m\|_2}{\|\mathbf{r}_0\|} \leq \max_{\substack{\mathbf{v}\in\mathbb{C}^n \\ \|\mathbf{v}\|=1}} \min_{\substack{\mathrm{p}(z)\in\mathbb{P}_m \\ \mathrm{p}(0)=1}} \|\mathrm{p}(A)\mathbf{v}\|, \quad m = 1, 2, \ldots$$

- If A is real, and $M = {}^{(A+A^T)}/_2$ is SPD, then (Eisenstat, Elman, and Schultz 1983)

$$\max_{\substack{\mathbf{v}\in\mathbb{R}^n \\ \|\mathbf{v}\|=1}} \min_{\substack{\mathrm{p}(z)\in\mathbb{P}_m \\ \mathrm{p}(0)=1}} \|\mathrm{p}(A)\mathbf{v}\| \leq \left(1 - \frac{\lambda_{\min}(M)^2}{\lambda_{\max}(A^T A)}\right)^{m/2}.$$

GMRES the non-diagonalizable case

$$\|\mathbf{r}_m\|_2 \leq \min_{\substack{\mathrm{p}(z)\in\mathbb{P}_m \\ \mathrm{p}(0)=1}} \|\mathrm{p}(A)\mathbf{r}_0\|, \quad m = 1, 2, \ldots$$

we recall that the **field of values** of A is given by

$$W(A) = \{< A\mathbf{v}, \mathbf{v} >: \mathbf{v} \in \mathbb{C}^n, \ \|\mathbf{v}\| = 1\}, \qquad \nu(A) = \min_{z\in W(A)} |z|,$$

with $\nu(A)$ the distance of $W(A)$ from the origin.

GMRES the non-diagonalizable case

$$\|\mathbf{r}_m\|_2 \leq \min_{\substack{\mathrm{p}(z)\in\mathbb{P}_m \\ \mathrm{p}(0)=1}} \|\mathrm{p}(A)\mathbf{r}_0\|, \quad m = 1, 2, \ldots$$

we recall that the **field of values** of A is given by

$$W(A) = \{< A\mathbf{v}, \mathbf{v} >: \mathbf{v} \in \mathbb{C}^n, \ \|\mathbf{v}\| = 1\}, \qquad \nu(A) = \min_{z\in W(A)} |z|,$$

with $\nu(A)$ the distance of $W(A)$ from the origin.

- For a general nonsingular A (Eiermann and Ernst 2001)

$$\max_{\substack{\mathbf{v}\in\mathbb{C}^n \\ \|\mathbf{v}\|=1}} \min_{\substack{\mathrm{p}(z)\in\mathbb{P}_m \\ \mathrm{p}(0)=1}} \|\mathrm{p}(A)\mathbf{v}\| \leq (1 - \nu(A)\nu(A^{-1}))^{m/2}.$$

GMRES the non-diagonalizable case

$$\|\mathbf{r}_m\|_2 \leq \min_{\substack{\mathrm{p}(z)\in\mathbb{P}_m \\ \mathrm{p}(0)=1}} \|\mathrm{p}(A)\mathbf{r}_0\|, \quad m = 1, 2, \ldots$$

we recall that the **field of values** of A is given by

$$W(A) = \{< A\mathbf{v}, \mathbf{v} >: \mathbf{v} \in \mathbb{C}^n, \ \|\mathbf{v}\| = 1\}, \qquad \nu(A) = \min_{z\in W(A)} |z|,$$

with $\nu(A)$ the distance of $W(A)$ from the origin.

- For a general nonsingular A (Eiermann and Ernst 2001)

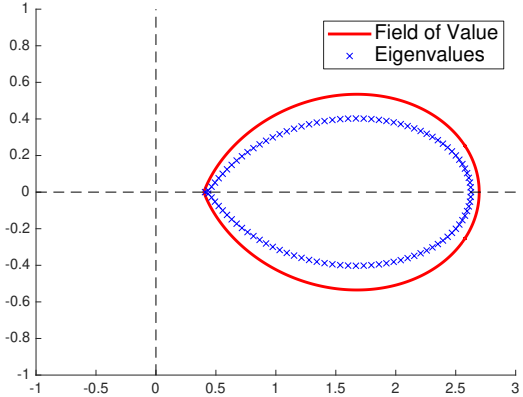
$$\max_{\substack{\mathbf{v}\in\mathbb{C}^n \\ \|\mathbf{v}\|=1}} \min_{\substack{\mathrm{p}(z)\in\mathbb{P}_m \\ \mathrm{p}(0)=1}} \|\mathrm{p}(A)\mathbf{v}\| \leq (1 - \nu(A)\nu(A^{-1}))^{m/2}.$$

- This bound is useful only when $0 \notin W(A)$ and $0 \notin W(A^{-1})$.

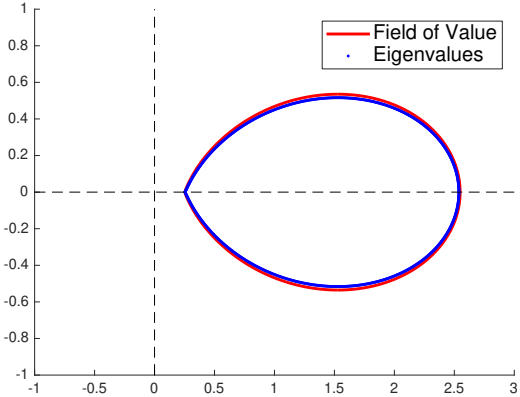
Some experimentation with the FOV in our case

$$\nu_N^{\alpha-1} A_N = \nu_N^{\alpha-1} I_N - \theta G_N + (1-\theta) G_N^T,$$

$\theta = 0.2,\ \alpha = 1.2,\ N = 100$



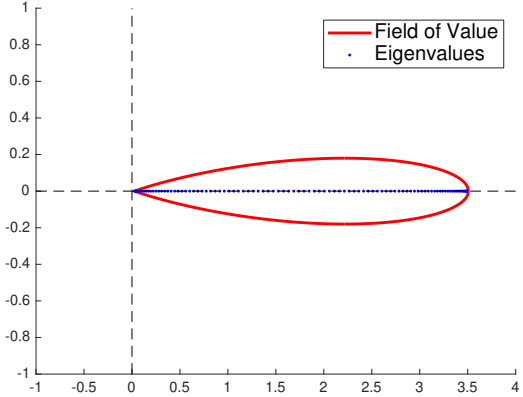
$\theta = 0.2,\ \alpha = 1.2,\ N = 1000$



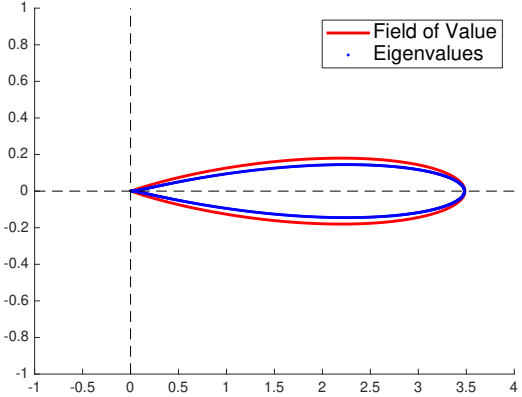
Some experimentation with the FOV in our case

$$\nu_N^{\alpha-1} A_N = \nu_N^{\alpha-1} I_N - \theta G_N + (1-\theta) G_N^T,$$

$\theta = 0.2,\ \alpha = 1.8,\ N = 100$



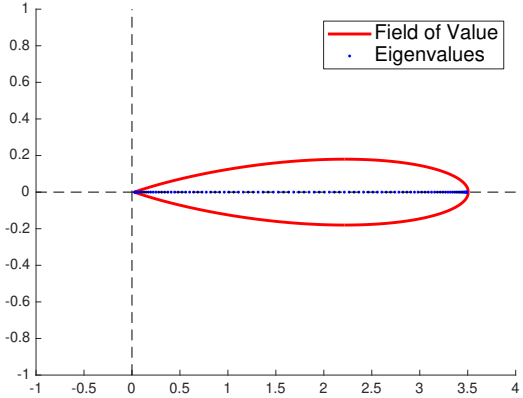
$\theta = 0.2,\ \alpha = 1.8,\ N = 1000$



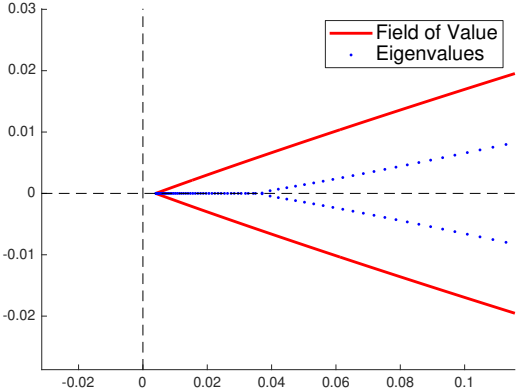
Some experimentation with the FOV in our case

$$\nu_N^{\alpha-1} A_N = \nu_N^{\alpha-1} I_N - \theta G_N + (1-\theta) G_N^T,$$

$\theta = 0.2,\ \alpha = 1.8,\ N = 100$



$\theta = 0.2,\ \alpha = 1.8,\ N = 1000$



Some experimentation with the FOV in our case

Unfortunate truth

In general it is difficult to say something about the Field of Value of preconditioned matrices.

Some experimentation with the FOV in our case

Unfortunate truth

In general it is difficult to say something about the Field of Value of preconditioned matrices.

What do we do in practice?

> *"To speed up the CG-like methods, we can choose a matrix C such that the singular values of the preconditioned matrix $C^{-1}A$ are clustered."* – *(R. H. Chan and Ng 1996, P. 439)*

Some experimentation with the FOV in our case

Unfortunate truth

In general it is difficult to say something about the Field of Value of preconditioned matrices.

- What do we do in practice?

  "To speed up the CG-like methods, we can choose a matrix C such that the singular values of the preconditioned matrix $C^{-1}A$ are clustered." – (R. H. Chan and Ng 1996, P. 439)

- How do we build a **Circulant preconditioner** for a **our non-symmetric Toeplitz** matrix?

Some experimentation with the FOV in our case

Unfortunate truth

In general it is difficult to say something about the Field of Value of preconditioned matrices.

- What do we do in practice?

  "To speed up the CG-like methods, we can choose a matrix C such that the singular values of the preconditioned matrix $C^{-1}A$ are clustered." – (R. H. Chan and Ng 1996, P. 439)

- How do we build a **Circulant preconditioner** for a **our non-symmetric Toeplitz** matrix?
- We can use a suitably modified Strang preconditioner for our case (Lei and Sun 2013)

A Circulant preconditioner (Lei and Sun 2013)

We can build a circulant preconditioner as

$$P = \frac{h_N^\alpha}{\Delta t} I_N + \theta s(G_N) + (1-\theta) s(G_N^T),$$

where

$$(s(G_N))_{:,1} = -\begin{bmatrix} g_1^{(\alpha)} \\ \vdots \\ g^\alpha_{\lfloor (N+1)/2 \rfloor} \\ 0 \\ \vdots \\ 0 \\ g_0^{(\alpha)} \end{bmatrix},$$

```
function [ev,evt] = sunprec(N,alpha)
g = gl(N,alpha);
v = zeros(N,1);
v(1:floor((N+1)/2)) =
↪  g((1:floor((N+1)/2))+1);
v(end) = g(1);
ev = fft(-v);
v = zeros(N,1);
v(1) = g(2);
v(2) = g(1);
v(end:-1:floor((N+1)/2)+2) =
↪  g(3:floor((N+1)/2)+1);
evt = fft(-v);
end
```

A Circulant preconditioner (Lei and Sun 2013)

We can build a circulant preconditioner as

$$P = \frac{h_N^\alpha}{\Delta t} I_N + \theta s(G_N) + (1-\theta) s(G_N^T),$$

where

$$(s(G_N^T))_{:,1} = -\begin{bmatrix} g_1^{(\alpha)} \\ g_0^{(\alpha)} \\ 0 \\ \vdots \\ 0 \\ g^\alpha_{\lfloor (N+1)/2 \rfloor} \\ \vdots \\ g_2^{(\alpha)} \end{bmatrix}.$$

```
function [ev,evt] = sunprec(N,alpha)
g = gl(N,alpha);
v = zeros(N,1);
v(1:floor((N+1)/2)) =
↪  g((1:floor((N+1)/2))+1);
v(end) = g(1);
ev = fft(-v);
v = zeros(N,1);
v(1) = g(2);
v(2) = g(1);
v(end:-1:floor((N+1)/2)+2) =
↪  g(3:floor((N+1)/2)+1);
evt = fft(-v);
end
```

A Circulant preconditioner (Lei and Sun 2013)

We can build a circulant preconditioner as

$$P = \frac{h_N^\alpha}{\Delta t} I_N + \theta s(G_N) + (1-\theta) s(G_N^T),$$

- It uses the **construction of the Strang preconditioner** using only *half o the bandwidth* of the Toeplitz matrices.

```
function [ev,evt] = sunprec(N,alpha)
g = gl(N,alpha);
v = zeros(N,1);
v(1:floor((N+1)/2)) =
↪  g((1:floor((N+1)/2))+1);
v(end) = g(1);
ev = fft(-v);
v = zeros(N,1);
v(1) = g(2);
v(2) = g(1);
v(end:-1:floor((N+1)/2)+2) =
↪  g(3:floor((N+1)/2)+1);
evt = fft(-v);
end
```

A Circulant preconditioner (Lei and Sun 2013)

We can build a circulant preconditioner as

$$P = \frac{h_N^\alpha}{\Delta t} I_N + \theta s(G_N) + (1-\theta) s(G_N^T),$$

- It uses the **construction of the Strang preconditioner** using only *half o the bandwidth* of the Toeplitz matrices.
- All the eigenvalues of $s(G_N)$ and $s(G_N^T)$ fall inside the open disc $\{z \in \mathbb{C} : |z-\alpha| < \alpha\}$ by Gershgorin theorem, indeed:

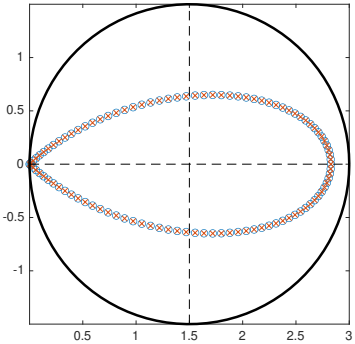
$$r_N = g_0^\alpha + \sum_{k=2}^{\lfloor (N+1)/2 \rfloor} < \sum_{\substack{k=0 \\ k \neq 1}} g_k^{(\alpha)} = -g_1^{(\alpha)} = \alpha.$$

```
function [ev,evt] = sunprec(N,alpha)
g = gl(N,alpha);
v = zeros(N,1);
v(1:floor((N+1)/2)) =
↪  g((1:floor((N+1)/2))+1);
v(end) = g(1);
ev = fft(-v);
v = zeros(N,1);
v(1) = g(2);
v(2) = g(1);
v(end:-1:floor((N+1)/2)+2) =
↪  g(3:floor((N+1)/2)+1);
evt = fft(-v);
end
```

A Circulant preconditioner (Lei and Sun 2013)

We can build a circulant preconditioner as

$$P = \frac{h_N^\alpha}{\Delta t} I_N + \theta s(G_N) + (1-\theta) s(G_N^T),$$




```
function [ev,evt] = sunprec(N,alpha)
g = gl(N,alpha);
v = zeros(N,1);
v(1:floor((N+1)/2)) =
↪  g((1:floor((N+1)/2))+1);
v(end) = g(1);
ev = fft(-v);
v = zeros(N,1);
v(1) = g(2);
v(2) = g(1);
v(end:-1:floor((N+1)/2)+2) =
↪  g(3:floor((N+1)/2)+1);
evt = fft(-v);
end
```

A Circulant preconditioner (Lei and Sun 2013)

❓ Will it work?
We can always write:

$$P^{-1}A_N - I_N = P^{-1}(A_N - P) \qquad ,$$

now for the Strang preconditioner of a Toeplitz matrix with with generating function in the Wiener class, it holds that for any $\varepsilon > 0$ exists N' and M' such that

$$A_N - s(A_N) = U_N + V_N, \quad \mathrm{rank}(U_N) \leq M' \text{ and } \|V_N\|_2 < \varepsilon \ \forall N > N'.$$

A Circulant preconditioner (Lei and Sun 2013)

❓ Will it work?
We can always write:

$$P^{-1}A_N - I_N = P^{-1}(A_N - P) = P_N^{-1}U_N - P_N^{-1}V_N,$$

now for the Strang preconditioner of a Toeplitz matrix with with generating function in the Wiener class, it holds that for any $\varepsilon > 0$ exists N' and M' such that

$$A_N - s(A_N) = U_N + V_N, \quad \mathrm{rank}(U_N) \leq M' \text{ and } \|V_N\|_2 < \varepsilon \ \forall N > N'.$$

A Circulant preconditioner (Lei and Sun 2013)

❓ Will it work?
We can always write:

$$P^{-1}A_N - I_N = P^{-1}(A_N - P) = P_N^{-1}U_N - P_N^{-1}V_N,$$

now for the Strang preconditioner of a Toeplitz matrix with with generating function in the Wiener class, it holds that for any $\varepsilon > 0$ exists N' and M' such that

$$A_N - s(A_N) = U_N + V_N, \quad \mathrm{rank}(U_N) \leq M' \text{ and } \|V_N\|_2 < \varepsilon \ \forall N > N'.$$

🔧 $\mathrm{rank}(P_N^{-1}U_N) \leq \mathrm{rank}(U_N) \leq M'$,

A Circulant preconditioner (Lei and Sun 2013)

❓ Will it work?

We can always write:

$$P^{-1}A_N - I_N = P^{-1}(A_N - P) = P_N^{-1}U_N - P_N^{-1}V_N,$$

now for the Strang preconditioner of a Toeplitz matrix with with generating function in the Wiener class, it holds that for any $\varepsilon > 0$ exists N' and M' such that

$$A_N - s(A_N) = U_N + V_N, \quad \mathrm{rank}(U_N) \leq M' \text{ and } \|V_N\|_2 < \varepsilon \; \forall N > N'.$$

- $\mathrm{rank}(P_N^{-1}U_N) \leq \mathrm{rank}(U_N) \leq M'$,
- $\forall\, k = 1, 2, \ldots, N$, $|\lambda(P_N)| \geq \Re(\Lambda(P_N)_{k,k}) = h_N^{\alpha}/\Delta t + \theta\Re(\Lambda(s(G_N))_{kk}) + (1-\theta)\Re(\Lambda(s(G_N^T))_{kk}) \geq h_N^{\alpha}/\Delta t > 0$ and thus $\|P_N^{-1}\|_2 \leq \Delta t/h_N^{\alpha}$

A Circulant preconditioner (Lei and Sun 2013)

❓ Will it work?

We can always write:

$$P^{-1}A_N - I_N = P^{-1}(A_N - P) = P_N^{-1}U_N - P_N^{-1}V_N,$$

now for the Strang preconditioner of a Toeplitz matrix with with generating function in the Wiener class, it holds that for any $\varepsilon > 0$ exists N' and M' such that

$$A_N - s(A_N) = U_N + V_N, \quad \mathrm{rank}(U_N) \leq M' \text{ and } \|V_N\|_2 < \varepsilon \ \forall N > N'.$$

- $\mathrm{rank}(P_N^{-1}U_N) \leq \mathrm{rank}(U_N) \leq M'$,
- $\|P_N^{-1}V_N\| \leq \|P_N^{-1}\|_2 \|V_N\|_2 < {}^{\varepsilon\Delta t}/_{h_N^{\alpha}}$.

A Circulant preconditioner (Lei and Sun 2013)

❓ Will it work?

We can always write:

$$P^{-1}A_N - I_N = P^{-1}(A_N - P) = P_N^{-1}U_N - P_N^{-1}V_N \Rightarrow \text{“small rank”} \ + \ \text{“small norm”},$$

now for the Strang preconditioner of a Toeplitz matrix with with generating function in the Wiener class, it holds that for any $\varepsilon > 0$ exists N' and M' such that

$$A_N - s(A_N) = U_N + V_N, \quad \mathrm{rank}(U_N) \leq M' \text{ and } \|V_N\|_2 < \varepsilon \ \forall N > N'.$$

- $\mathrm{rank}(P_N^{-1}U_N) \leq \mathrm{rank}(U_N) \leq M'$,
- $\|P_N^{-1}V_N\| \leq \|P_N^{-1}\|_2 \|V_N\|_2 < {}^{\varepsilon\Delta t}/_{h_N^{\alpha}}$.
- If we select Δt and h_N in such a way that ${}^{h_N^{\alpha}}/_{\Delta t}$ is bounded and bounded away from zero we have the result.

Results with GMRES

$$\left(\frac{h_N^\alpha}{\Delta t} I_{N-2} - \left[\theta G_{N-2} + (1-\theta) G_{N-2}^T\right]\right) \mathbf{w}^{n+1} = \frac{h_N^\alpha}{\Delta t}, \quad \theta = 0.2$$

Results with GMRES

```
[ev,evt] = sunprec(N,alpha);
c = nu + theta*ev + (1-theta)*evt;
P = @(x) cprec(c,x);
[X,FLAGsun,RELRESsun,ITERsun,RESVECsun] = gmres(A,(nu*w),[],1e-9,N,P);
```

α	N	GMRES	P
	2^5	28	6
	2^6	31	6
	2^7	33	6
1.2	2^8	34	6
	2^9	35	6
	2^{10}	36	6
	2^{11}	36	6

α	N	GMRES	P
	2^5	31	6
	2^6	46	6
	2^7	54	6
1.4	2^8	62	7
	2^9	69	7
	2^{10}	78	7
	2^{11}	87	7

α	N	GMRES	P
	2^5	32	6
	2^6	59	6
	2^7	82	7
1.6	2^8	105	7
	2^9	128	7
	2^{10}	156	7
	2^{11}	189	7

α	N	GMRES	P
	2^5	32	6
	2^6	64	6
	2^7	109	6
1.8	2^8	162	7
	2^9	222	7
	2^{10}	287	7
	2^{11}	372	7

Conclusion and summary

- We have discussed the solution of Toeplitz linear systems,
- Studied the usage and convergence of PCG and GMRES method,
- Tested the usage of Circulant preconditioners for Toeplitz linear systems.

Next up

- We need to discuss the next problem in difficulty

$$\begin{cases} \frac{\partial W}{\partial t} = d^{+}(x,t)\,{}^{RL}D^{\alpha}_{[0,x]}W(x,t) + d^{-}(x,t)\,{}^{RL}D^{\alpha}_{[x,1]}W(x,t), & \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

- What happens if we go to **more than one spatial dimension**?

Bibliography I

- Ammar, G. S. and W. B. Gragg (1988). "Superfast solution of real positive definite Toeplitz systems". In: *SIAM J. Matrix Anal. Appl.* 9.1, pp. 61–76.
- Bini, D. A. and B. Meini (1999). "Effective methods for solving banded Toeplitz systems". In: *SIAM J. Matrix Anal. Appl.* 20.3, pp. 700–719. ISSN: 0895-4798. DOI: 10.1137/S0895479897324585. URL: https://doi.org/10.1137/S0895479897324585.
- Bitmead, R. R. and B. D. Anderson (1980). "Asymptotically fast solution of Toeplitz and related systems of linear equations". In: *Linear Algebra Appl.* 34, pp. 103–116.
- Brent, R. P., F. G. Gustavson, and D. Y. Yun (1980). "Fast solution of Toeplitz systems of equations and computation of Padé approximants". In: *J. Algorithms* 1.3, pp. 259–295.
- Chan, R. H. and M. K. Ng (1996). "Conjugate gradient methods for Toeplitz systems". In: *SIAM Rev.* 38.3, pp. 427–482. ISSN: 0036-1445. DOI: 10.1137/S0036144594276474. URL: https://doi.org/10.1137/S0036144594276474.

Bibliography II

- Chan, R. H., M. K. Ng, and A. M. Yip (2002). "The best circulant preconditioners for Hermitian Toeplitz systems. II. The multiple-zero case". In: *Numer. Math.* 92.1, pp. 17–40. ISSN: 0029-599X. DOI: 10.1007/s002110100354. URL: https://doi.org/10.1007/s002110100354.
- Chan, R. H. and M.-C. Yeung (1992). "Circulant preconditioners constructed from kernels". In: *SIAM J. Numer. Anal.* 29.4, pp. 1093–1103. ISSN: 0036-1429. DOI: 10.1137/0729066. URL: https://doi.org/10.1137/0729066.
- Chan, T. F. and P. C. Hansen (1992). "A look–ahead Levinson algorithm for general Toeplitz systems". In: *IEEE Transactions on signal processing* 40.5, pp. 1079–1090.
- Donatelli, M., M. Mazza, and S. Serra-Capizzano (2016). "Spectral analysis and structure preserving preconditioners for fractional diffusion equations". In: *J. Comput. Phys.* 307, pp. 262–279. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2015.11.061. URL: https://doi.org/10.1016/j.jcp.2015.11.061.

Bibliography III

- Eiermann, M. and O. G. Ernst (2001). "Geometric aspects of the theory of Krylov subspace methods". In: *Acta Numer.* 10, pp. 251–312. ISSN: 0962-4929. DOI: 10.1017/S0962492901000046. URL: https://doi.org/10.1017/S0962492901000046.
- Eisenstat, S. C., H. C. Elman, and M. H. Schultz (1983). "Variational iterative methods for nonsymmetric systems of linear equations". In: *SIAM J. Numer. Anal.* 20.2, pp. 345–357. ISSN: 0036-1429. DOI: 10.1137/0720023. URL: https://doi.org/10.1137/0720023.
- Gohberg, I. C. and A. A. Semencul (1972). "The inversion of finite Toeplitz matrices and their continual analogues". In: *Mat. Issled.* 7.2(24), pp. 201–223, 290. ISSN: 0542-9994.
- Greenbaum, A., V. Pták, and Z. Strakoš (1996). "Any Nonincreasing Convergence Curve is Possible for GMRES". In: *SIAM J. Matrix Anal. Appl.* 17.3, pp. 465–469. DOI: 10.1137/S0895479894275030. eprint: http://dx.doi.org/10.1137/S0895479894275030. URL: http://dx.doi.org/10.1137/S0895479894275030.
- Hoog, F. de (1987). "A new algorithm for solving Toeplitz systems of equations". In: *Linear Algebra Appl.* 88, pp. 123–138.

Bibliography IV

- Lei, S.-L. and H.-W. Sun (2013). "A circulant preconditioner for fractional diffusion equations". In: *J. Comput. Phys.* 242, pp. 715–725. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2013.02.025. URL: https://doi.org/10.1016/j.jcp.2013.02.025.
- Levinson, N. (1946). "The Wiener (root mean square) error criterion in filter design and prediction". In: *J. Math. Phys.* 25.1, pp. 261–278.
- Saad, Y. and M. H. Schultz (1986). "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems". In: *SIAM J. Sci. Statist. Comput.* 7.3, pp. 856–869. ISSN: 0196-5204. DOI: 10.1137/0907058. URL: https://doi.org/10.1137/0907058.
- Trench, W. F. (1964). "An algorithm for the inversion of finite Toeplitz matrices". In: *SIAM J. Appl. Math.* 12.3, pp. 515–522.
- Zohar, S. (1974). "The solution of a Toeplitz set of linear equations". In: *J. Assoc. Comput. Mach.* 21.2, pp. 272–276.

An introduction to fractional calculus

Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

October, 2022

Variable coefficients cases

We now want to solve the *slightly* more complex case

$$\begin{cases} \frac{\partial W}{\partial t} = d^+(x,t)\,{}^{RL}D^{\alpha}_{[0,x]}W(x,t) + d^-(x,t)\,{}^{RL}D^{\alpha}_{[x,1]}W(x,t), \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

with $d^+(x,t), d^-(x,t) \geq 0$ and **not identically** zero.

Variable coefficients cases

We now want to solve the *slightly* more complex case

$$\begin{cases} \frac{\partial W}{\partial t} = d^+(x,t)\,{}^{RL}D^{\alpha}_{[0,x]}W(x,t) + d^-(x,t)\,{}^{RL}D^{\alpha}_{[x,1]}W(x,t), \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

with $d^+(x,t), d^-(x,t) \geq 0$ and **not identically** zero.

1. We go through all the **same discretization procedure**: from Riemann–Liouville to (shifted) Grünwald–Letnikov, then series truncation, *etc.*

Variable coefficients cases

We now want to solve the *slightly* more complex case

$$\begin{cases} \frac{\partial W}{\partial t} = d^+(x,t)\,{}^{RL}D^{\alpha}_{[0,x]}W(x,t) + d^-(x,t)\,{}^{RL}D^{\alpha}_{[x,1]}W(x,t), \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

with $d^+(x,t), d^-(x,t) \geq 0$ and **not identically** zero.

1. We go through all the **same discretization procedure**: from Riemann–Liouville to (shifted) Grünwald–Letnikov, then series truncation, *etc.*
2. we obtain a matrix sequence of the form

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T\right),$$

where $D_N^{\pm}$ are **diagonal matrices** whose entries **sample the functions** $d_N^{\pm}(x,t)$ on the finite difference grid.

Variable coefficients cases

We now want to solve the *slightly* more complex case

$$\begin{cases} \frac{\partial W}{\partial t} = d^+(x,t)\,{}^{RL}D^{\alpha}_{[0,x]}W(x,t) + d^-(x,t)\,{}^{RL}D^{\alpha}_{[x,1]}W(x,t), \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

with $d^+(x,t), d^-(x,t) \geq 0$ and **not identically** zero.

1. We go through all the **same discretization procedure**: from Riemann–Liouville to (shifted) Grünwald–Letnikov, then series truncation, *etc.*
2. we obtain a matrix sequence of the form

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T\right),$$

where $D_N^{\pm}$ are **diagonal matrices** whose entries **sample the functions** $d_N^{\pm}(x,t)$ on the finite difference grid.

- We **no longer have Toeplitz matrices**!

Not all hope is lost

- We can still perform **fast matrix-vector products**:

$$A_N\mathbf{x} = \nu\mathbf{x} - D_N^+(G_N\mathbf{x}) - D_N^-(G_N^T\mathbf{x})$$

still $O(N \log N)$ cost.

- Maybe we can use some trick to reuse circulant preconditioners
  1. If $d_N^{\pm}(x,t)$ do not vary much maybe we can **average them**, i.e.,

$$P(t) = \nu I - \widehat{d^+}(t)s(G_N) - \widehat{d^-}(t)s(G_N^T),$$

with $\widehat{d^{\pm}}(t) = 1/N \sum_{i=1}^{N} d^{\pm}(x_i,t)$

The averaging trick

Does it work?

$$d^+(x,t) = \Gamma(3-\alpha)x^\alpha, \qquad d^-(x,t) = \Gamma(3-\alpha)(2-x)^\alpha$$

```
w0 = @(x) 5*x.*(1-x);
hN = 1/(N-1); x = 0:hN:1; dt = hN; t = 0:dt:1;
dplus = @(x,t) gamma(3-alpha).*x.^alpha;
dminus = @(x,t) gamma(3-alpha).*(2-x).^alpha;
% Discretize
G = glmatrix(N,alpha); Gr = G; Grt = G.'; I = eye(N,N);
Dplus = diag(dplus(x,0)); Dminus = diag(dminus(x,0));
% Left-hand side
nu = hN^alpha/dt;
A = nu*I - (Dplus*Gr + Dminus*Grt);
```

The averaging trick

Does it work?

$$d^+(x,t) = \Gamma(3-\alpha)x^\alpha, \qquad d^-(x,t) = \Gamma(3-\alpha)(2-x)^\alpha$$

```
% Solve
[ev,evt] = sunprec(N,alpha);
c = nu + mean(dplus(x,0))*ev + mean(dminus(x,0))*evt;
P = @(x) cprec(c,x);
[X,FLAGsun,RELRESsun,ITERsun,RESVECsun] = gmres(A,(nu*w),[],1e-9,N,P);
```

The averaging trick

Does it work?

$$d^+(x,t) = \Gamma(3-\alpha)x^\alpha, \qquad d^-(x,t) = \Gamma(3-\alpha)(2-x)^\alpha$$

α	N	GMRES	P
	2^5	31	13
	2^6	50	14
	2^7	64	14
1.2	2^8	75	15
	2^9	84	15
	2^{10}	91	14
	2^{11}	96	14

α	N	GMRES	P
	2^5	31	13
	2^6	59	14
	2^7	92	15
1.4	2^8	127	15
	2^9	161	15
	2^{10}	196	15
	2^{11}	231	15

α	N	GMRES	P
	2^5	32	13
	2^6	62	13
	2^7	112	14
1.6	2^8	183	14
	2^9	262	14
	2^{10}	353	14
	2^{11}	456	14

α	N	GMRES	P
	2^5	32	12
	2^6	64	12
	2^7	126	13
1.8	2^8	225	13
	2^9	378	13
	2^{10}	559	12
	2^{11}	779	12

The averaging trick

Does it work?

$$d^+(x,t) = \Gamma(3-\alpha)x^\alpha, \qquad d^-(x,t) = \Gamma(3-\alpha)(2-x)^\alpha$$

α	N	GMRES	P
	2^5	31	13
	2^6	50	14
	2^7	64	14
1.2	2^8	75	15
	2^9	84	15
	2^{10}	91	14
	2^{11}	96	14

α	N	GMRES	P
	2^5	31	13
	2^6	59	14
	2^7	92	15
1.4	2^8	127	15
	2^9	161	15
	2^{10}	196	15
	2^{11}	231	15

α	N	GMRES	P
	2^5	32	13
	2^6	62	13
	2^7	112	14
1.6	2^8	183	14
	2^9	262	14
	2^{10}	353	14
	2^{11}	456	14

α	N	GMRES	P
	2^5	32	12
	2^6	64	12
	2^7	126	13
1.8	2^8	225	13
	2^9	378	13
	2^{10}	559	12
	2^{11}	779	12

We have **doubled the number of iterations** but things still seem reasonable...

Can we prove anything?

What did we actually prove for the *constant coefficient case*?

Can we prove anything?

What did we actually prove for the *constant coefficient case*?

- We computed the **asymptotic spectral distribution** of the matrix sequence $\{\nu A_N\}_N$ (*eigenvalues* for the symmetric case, *singular values* for the general case);

Can we prove anything?

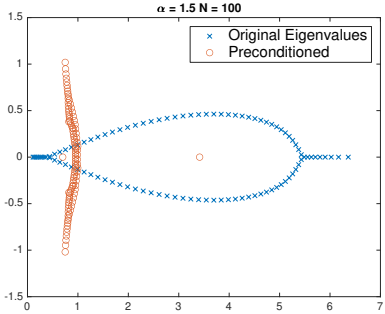
What did we actually prove for the *constant coefficient case*?

- We computed the **asymptotic spectral distribution** of the matrix sequence $\{\nu A_N\}_N$ (*eigenvalues* for the symmetric case, *singular values* for the general case);
- We proved that $P^{-1}A_N - I =$ "small norm" + "small rank", i.e., that the preconditioner delivered a **clustering of the eigenvalues**.

Can we prove anything?

What did we actually prove for the *constant coefficient case*?

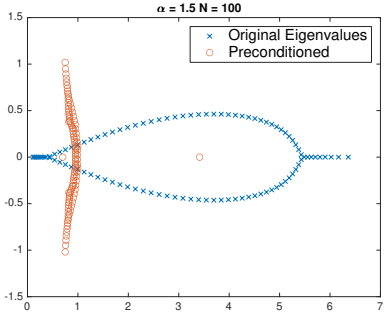
- We proved that $P^{-1}A_N - I =$ "small norm" + "small rank", i.e., that the preconditioner delivered a **clustering of the eigenvalues**.

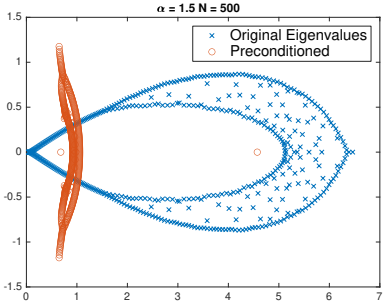


Can we prove anything?

What did we actually prove for the *constant coefficient case*?

- We proved that $P^{-1}A_N - I =$ "small norm" + "small rank", i.e., that the preconditioner delivered a **clustering of the eigenvalues**.

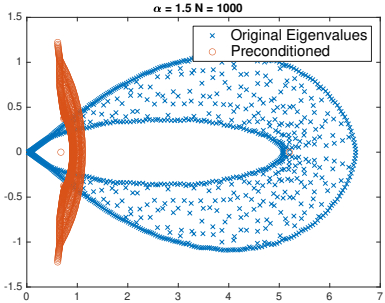


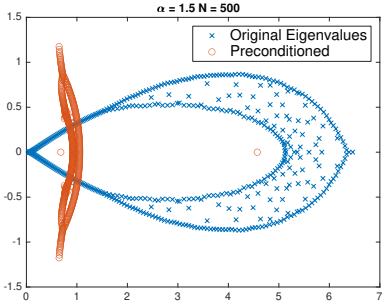


Can we prove anything?

What did we actually prove for the *constant coefficient case*?

- We proved that $P^{-1}A_N - I =$ "small norm" + "small rank", i.e., that the preconditioner delivered a **clustering of the eigenvalues**.

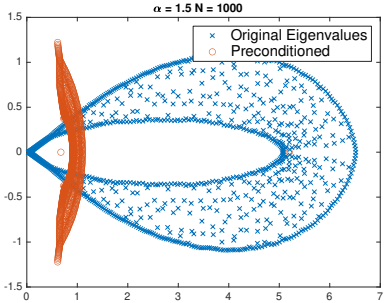


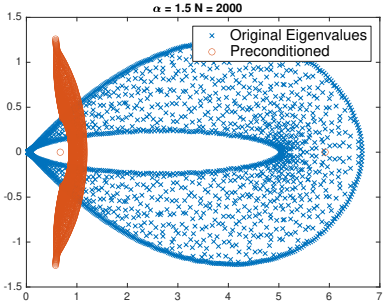


Can we prove anything?

What did we actually prove for the *constant coefficient case*?

- We proved that $P^{-1}A_N - I =$ "small norm" + "small rank", i.e., that the preconditioner delivered a **clustering of the eigenvalues**.

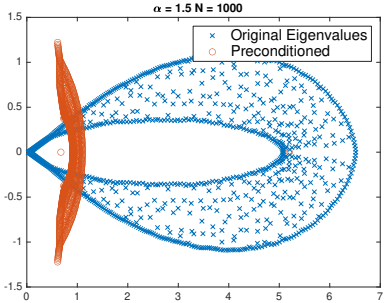


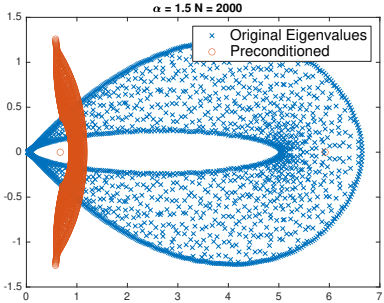


Can we prove anything?

What did we actually prove for the *constant coefficient case*?

- We proved that $P^{-1}A_N - I =$ "small norm" + "small rank", i.e., that the preconditioner delivered a **clustering of the eigenvalues**.



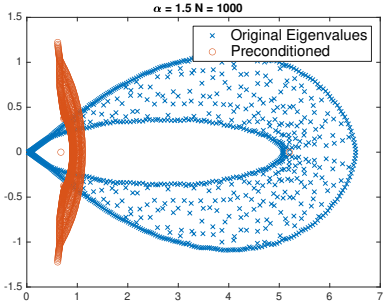


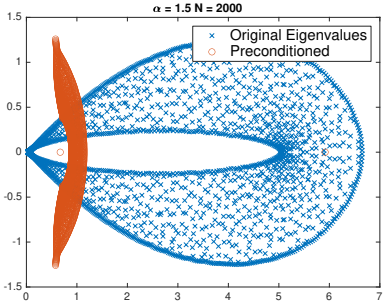
- We **don't have a cluster**, yet eigenvalues are in a fairly small region.

Can we prove anything?

What did we actually prove for the *constant coefficient case*?

- We proved that $P^{-1}A_N - I =$ "small norm" + "small rank", i.e., that the preconditioner delivered a **clustering of the eigenvalues**.





We **don't have a cluster**, yet eigenvalues are in a fairly small region.

let's investigate!

Having a cluster: $C_n - A_n$

For two matrix sequences $\{C_n\}_n$ and $\{A_n\}_n$ (both of order n) we say that they are ε-close by rank if

$$\forall\, \varepsilon > 0 \; A_n - C_n = E_{n,\varepsilon} + R_{n,\varepsilon}, \qquad \begin{array}{l} \|E_{n,\varepsilon}\|_2 \leq \varepsilon, \\ \mathrm{rank}(R_{n,\varepsilon}) \leq r(n,\varepsilon) = o(n) \text{ for } n \to +\infty, \end{array} \qquad (\varepsilon\text{-close})$$

Having a cluster: $C_n - A_n$

For two matrix sequences $\{C_n\}_n$ and $\{A_n\}_n$ (both of order n) we say that they are ε-close by rank if

$$\forall\, \varepsilon > 0 \; A_n - C_n = E_{n,\varepsilon} + R_{n,\varepsilon}, \qquad \begin{array}{l} \|E_{n,\varepsilon}\|_2 \leq \varepsilon, \\ \mathrm{rank}(R_{n,\varepsilon}) \leq r(n,\varepsilon) = o(n) \text{ for } n \to +\infty, \end{array} \qquad (\varepsilon\text{-close})$$

Let $\gamma_n(\varepsilon)$ count how many singular values $\sigma(A_n - C_n)$ are greater than ε, i.e.,

$$\gamma_n(\varepsilon) = \left|\{j \,:\, \sigma_j(A_n - C_n) > \varepsilon, \quad j = 1, \ldots, n\}\right|,$$

Having a cluster: $C_n - A_n$

For two matrix sequences $\{C_n\}_n$ and $\{A_n\}_n$ (both of order n) we say that they are ε-close by rank if

$$\forall\, \varepsilon > 0 \; A_n - C_n = E_{n,\varepsilon} + R_{n,\varepsilon}, \qquad \begin{array}{l} \|E_{n,\varepsilon}\|_2 \leq \varepsilon, \\ \mathrm{rank}(R_{n,\varepsilon}) \leq r(n,\varepsilon) = o(n) \text{ for } n \to +\infty, \end{array} \qquad (\varepsilon\text{-close})$$

Let $\gamma_n(\varepsilon)$ count how many singular values $\sigma(A_n - C_n)$ are greater than ε, i.e.,

$$\gamma_n(\varepsilon) = |\{j \,:\, \sigma_j(A_n - C_n) > \varepsilon, \quad j = 1, \ldots, n\}|,$$

$\Rightarrow$ (ε-close) is telling us that $\gamma_n(\varepsilon) = o(n)$ for $n \to +\infty$.

Having a cluster: $C_n - A_n$

For two matrix sequences $\{C_n\}_n$ and $\{A_n\}_n$ (both of order n) we say that they are ε-close by rank if

$$\forall\, \varepsilon > 0 \; A_n - C_n = E_{n,\varepsilon} + R_{n,\varepsilon}, \qquad \begin{array}{l} \|E_{n,\varepsilon}\|_2 \leq \varepsilon, \\ \mathrm{rank}(R_{n,\varepsilon}) \leq r(n,\varepsilon) = o(n) \text{ for } n \to +\infty, \end{array} \qquad (\varepsilon\text{-close})$$

Let $\gamma_n(\varepsilon)$ count how many singular values $\sigma(A_n - C_n)$ are greater than ε, i.e.,

$$\gamma_n(\varepsilon) = |\{j \,:\, \sigma_j(A_n - C_n) > \varepsilon, \quad j = 1, \ldots, n\}|\,,$$

$\Rightarrow$ (ε-close) is telling us that $\gamma_n(\varepsilon) = o(n)$ for $n \to +\infty$.

Then we know that $\{A_n - C_n\}_n$ has a singular value **cluster** at zero, if $\gamma_n(\varepsilon) = O(1)$ which holds equally with $r(n,\varepsilon) = r(\varepsilon) = O(1)$ for any $\varepsilon > 0$ then we have a **proper cluster** by the definition we have seen during the last lecture.

Having a cluster: $C_n^{-1}A_n - I_n$

To **estimate the convergence rate** we have shown that $C_n^{-1}A_n$ and I_n are (ε-close) matrix sequences, one usually use the **following nomenclature**

- C_n is **superlinear** for A_n if $r(n, \varepsilon) = O(1)$,
- C_n is **sublinear** for A_n if $r(n, \varepsilon) = o(n)$.

Having a cluster: $C_n^{-1}A_n - I_n$

To **estimate the convergence rate** we have shown that $C_n^{-1}A_n$ and I_n are (ε-close) matrix sequences, one usually use the **following nomenclature**

- C_n is **superlinear** for A_n if $r(n, \varepsilon) = O(1)$,
- C_n is **sublinear** for A_n if $r(n, \varepsilon) = o(n)$.

Strategy

It is usually easier to prove that C_n and A_n are (ε-close), rather than $C_n^{-1}A_n$ and I_n.

Having a cluster: $C_n^{-1}A_n - I_n$

To **estimate the convergence rate** we have shown that $C_n^{-1}A_n$ and I_n are (ε-close) matrix sequences, one usually use the **following nomenclature**

- C_n is **superlinear** for A_n if $r(n, \varepsilon) = O(1)$,
- C_n is **sublinear** for A_n if $r(n, \varepsilon) = o(n)$.

Strategy

It is usually easier to prove that C_n and A_n are (ε-close), rather than $C_n^{-1}A_n$ and I_n.

Proposition

If C_n and C_n^{-1} are **bounded uniformly** in n, then A_n and C_n are (ε-close) by $O(1)$ rank if and only if $C_n^{-1}A_n$ and I_n are.

Having a cluster: $C_n^{-1}A_n - I_n$

To **estimate the convergence rate** we have shown that $C_n^{-1}A_n$ and I_n are (ε-close) matrix sequences, one usually use the **following nomenclature**

- C_n is **superlinear** for A_n if $r(n, \varepsilon) = O(1)$,
- C_n is **sublinear** for A_n if $r(n, \varepsilon) = o(n)$.

Strategy

It is usually easier to prove that C_n and A_n are (ε-close), rather than $C_n^{-1}A_n$ and I_n.

Proposition

If C_n and C_n^{-1} are **bounded uniformly** in n, then A_n and C_n are (ε-close) by $O(1)$ rank if and only if $C_n^{-1}A_n$ and I_n are.

Proof.

$$A_n - C_n = C_n(C_n^{-1}A_n - I_n), \text{ and } C_n^{-1}A_n - I_n = C_n^{-1}(A_n - C_n).$$

Having a cluster: $C_n^{-1}A_n - I_n$

To **estimate the convergence rate** we have shown that $C_n^{-1}A_n$ and I_n are (ε-close) matrix sequences, one usually use the **following nomenclature**

- C_n is **superlinear** for A_n if $r(n, \varepsilon) = O(1)$,
- C_n is **sublinear** for A_n if $r(n, \varepsilon) = o(n)$.

Strategy

It is usually easier to prove that C_n and A_n are (ε-close), rather than $C_n^{-1}A_n$ and I_n.

Proposition

If C_n and C_n^{-1} are **bounded uniformly** in n, then A_n and C_n are (ε-close) by $O(1)$ rank if and only if $C_n^{-1}A_n$ and I_n are.

Proof.

$$C_n^{-1}A_n - I_n = C_n^{-1}(E_{n,\varepsilon} + R_{n,\varepsilon}) = C_n^{-1}E_{n,\varepsilon} + C_n^{-1}R_{n,\varepsilon}$$

Having a cluster: $C_n^{-1}A_n - I_n$

To **estimate the convergence rate** we have shown that $C_n^{-1}A_n$ and I_n are (ε-close) matrix sequences, one usually use the **following nomenclature**

- C_n is **superlinear** for A_n if $r(n,\varepsilon) = O(1)$,
- C_n is **sublinear** for A_n if $r(n,\varepsilon) = o(n)$.

Strategy

It is usually easier to prove that C_n and A_n are (ε-close), rather than $C_n^{-1}A_n$ and I_n.

Proposition

If C_n and C_n^{-1} are **bounded uniformly** in n, then A_n and C_n are (ε-close) by $O(1)$ rank if and only if $C_n^{-1}A_n$ and I_n are.

Proof.

$$C_n^{-1}A_n - I_n = C_n^{-1}E_{n,\varepsilon} + C_n^{-1}R_{n,\varepsilon}, \quad \|C_n^{-1}E_{n,\varepsilon}\| \leq \varepsilon/\|C_n\|_2, \quad \mathrm{rank}(C_n^{-1}R_{n,\varepsilon}) \leq r(n,\varepsilon) = O(1). \quad \square$$

Having a cluster: $C_n^{-1}A_n - I_n$

The connection between boundedness and ε-closeness can also be inverted, i.e.,

Proposition

Let C_n be non singular. If C_n is bounded uniformly in n and A_n and C_n are not (ε-close) by $O(1)$ rank, then C_n **is not superlinear** for A_n.

Proof.

- Both propositions makes assumption on C_n, can we say something without having to impose anything on C_n, $\|C_n\|_2$ or $\|C_n^{-1}\|_2$?

Having a cluster: $C_n^{-1}A_n - I_n$

The connection between boundedness and ε-closeness can also be inverted, i.e.,

Proposition

Let C_n be non singular. If C_n is bounded uniformly in n and A_n and C_n are not (ε-close) by $O(1)$ rank, then C_n **is not superlinear** for A_n.

Proof. By *contradiction*, if C_n is *superlinear* for A_n, then $C_n^{-1}A_n - I_n$ is the sum of a term of bounded norm $\varepsilon/\|C_n\|_2$ and a term of rank bounded by $O(1)$.

- Both propositions makes assumption on C_n, can we say something without having to impose anything on C_n, $\|C_n\|_2$ or $\|C_n^{-1}\|_2$?

Having a cluster: $C_n^{-1}A_n - I_n$

The connection between boundedness and ε-closeness can also be inverted, i.e.,

Proposition

Let C_n be non singular. If C_n is bounded uniformly in n and A_n and C_n are not (ε-close) by $O(1)$ rank, then C_n **is not superlinear** for A_n.

Proof. By *contradiction*, if C_n is *superlinear* for A_n, then $C_n^{-1}A_n - I_n$ is the sum of a term of bounded norm $\varepsilon/\|C_n\|_2$ and a term of rank bounded by $O(1)$. Therefore,

$$A_n - C_n = C_n(C_n^{-1}A_n - I_n),$$

is the sum of a term of norm bounded by ε and a term of *constant rank*: this contradicts the assumption that A_n and C_n are not (ε-close) by $O(1)$ rank. □

Both propositions makes assumption on C_n, can we say something without having to impose anything on C_n, $\|C_n\|_2$ or $\|C_n^{-1}\|_2$?

Having a cluster: $C_n^{-1}A_n - I_n$

Proposition

Let A_n and C_n be non singular. If A_n is bounded uniformly in n and if A_n and C_n are not (ε-close) by $O(1)$ rank, then C_n is not superlinear for A_n.

Proof.

Having a cluster: $C_n^{-1}A_n - I_n$

Proposition

Let A_n and C_n be non singular. If A_n is bounded uniformly in n and if A_n and C_n are not (ε-close) by $O(1)$ rank, then C_n is not superlinear for A_n.

Proof. We prove it again by contradiction.

Having a cluster: $C_n^{-1}A_n - I_n$

Proposition

Let A_n and C_n be non singular. If A_n is bounded uniformly in n and if A_n and C_n are not (ε-close) by $O(1)$ rank, then C_n is not superlinear for A_n.

Proof. We prove it again by contradiction. If C_n is superlinear for A_n, then (ε-close) holds for $C_n^{-1}A_n - I_n$ with $r(n, \varepsilon) = O(1)$.

Having a cluster: $C_n^{-1}A_n - I_n$

Proposition

Let A_n and C_n be non singular. If A_n is bounded uniformly in n and if A_n and C_n are not (ε-close) by $O(1)$ rank, then C_n is not superlinear for A_n.

Proof. We prove it again by contradiction. If C_n is superlinear for A_n, then (ε-close) holds for $C_n^{-1}A_n - I_n$ with $r(n, \varepsilon) = O(1)$. We use Sherman-Morrison-Woodbury formula to show that

$$A_n^{-1}C_n - I_n = E_{n,\varepsilon} + R_{n,\varepsilon}, \quad \|E_{n,\varepsilon} < \varepsilon \text{ and } R_{n,\varepsilon} = O(1).$$

Having a cluster: $C_n^{-1}A_n - I_n$

Proposition

Let A_n and C_n be non singular. If A_n is bounded uniformly in n and if A_n and C_n are not (ε-close) by $O(1)$ rank, then C_n is not superlinear for A_n.

Proof. We prove it again by contradiction. If C_n is superlinear for A_n, then (ε-close) holds for $C_n^{-1}A_n - I_n$ with $r(n,\varepsilon) = O(1)$. We use Sherman-Morrison-Woodbury formula to show that

$$A_n^{-1}C_n - I_n = E_{n,\varepsilon} + R_{n,\varepsilon}, \quad \|E_{n,\varepsilon} < \varepsilon \text{ and } R_{n,\varepsilon} = O(1).$$

Therefore,

$$-(A_n - C_n) = A_n(A_n^{-1}C_n - I_n)$$

is the sum of a term of norm bounded by $O(\varepsilon)$ and a term o constant rank 💣 this contradicts A_n and C_n non being (ε-close) by $O(1)$ rank. □

Having a cluster: $C_n^{-1}A_n - I_n$

Proposition

Let A_n and C_n be non singular. If A_n is bounded uniformly in n and if A_n and C_n are not (ε-close) by $O(1)$ rank, then C_n is not superlinear for A_n.

Proof. We prove it again by contradiction. If C_n is superlinear for A_n, then (ε-close) holds for $C_n^{-1}A_n - I_n$ with $r(n,\varepsilon) = O(1)$. We use Sherman-Morrison-Woodbury formula to show that

$$A_n^{-1}C_n - I_n = E_{n,\varepsilon} + R_{n,\varepsilon}, \quad \|E_{n,\varepsilon} < \varepsilon \text{ and } R_{n,\varepsilon} = O(1).$$

Therefore,

$$-(A_n - C_n) = A_n(A_n^{-1}C_n - I_n)$$

is the sum of a term of norm bounded by $O(\varepsilon)$ and a term o constant rank 💣 this contradicts A_n and C_n non being (ε-close) by $O(1)$ rank. □

❓ If we have information on the *spectral distribution* of the involved sequences, can we conclude something?

Asymptotic spectral distribution for non-Toeplitz sequences

For **Toeplitz matrices** we discovered that the following definitions holds for suitably chosen generating functions f.

Asymptotic eigenvalue distribution

Given a sequence of matrices $\{X_n\}_n \in \mathbb{C}^{d_n \times d_n}$ with $d_n = \{\dim X_n\}_n \overset{n\to+\infty}{\longrightarrow} \infty$ monotonically and a μ-measurable function $f : D \to \mathbb{R}$, with $\mu(D) \in (0, \infty)$, we say that the sequence $\{X\}_n$ is distributed in the sense of the eigenvalues as the function f and write $\{X_n\}_n \sim_\lambda f$ if and only if,

$$\lim_{n\to\infty} \frac{1}{d_n} \sum_{j=0}^{d_n} F(\lambda_j(X_n)) = \frac{1}{\mu(D)} \int_D F(f(t))dt, \quad \forall F \in \mathcal{C}_c(D),$$

where $\lambda_j(\cdot)$ indicates the j-th eigenvalue.

Asymptotic spectral distribution for non-Toeplitz sequences

For **Toeplitz matrices** we discovered that the following definitions holds for suitably chosen generating functions f.

Asymptotic singular value distribution

Given a sequence of matrices $\{X_n\}_n \in \mathbb{C}^{d_n \times d_n}$ with $d_n = \{\dim X_n\}_n \overset{n\to+\infty}{\longrightarrow} \infty$ monotonically and a μ-measurable function $f : D \to \mathbb{R}$, with $\mu(D) \in (0, \infty)$, we say that the sequence $\{X\}_n$ is distributed in the sense of the singular values as the function f and write $\{X_n\}_n \sim_\sigma f$ if and only if

$$\lim_{n\to\infty} \frac{1}{d_n} \sum_{j=0}^{d_n} F(\sigma_j(X_n)) = \frac{1}{\mu(D)} \int_D F(|f(t)|)dt, \quad \forall F \in \mathcal{C}_c(D),$$

where $\sigma_j(\cdot)$ is the j-th singular value.

Asymptotic spectral distribution for non-Toeplitz sequences

For **Toeplitz matrices** we discovered that the following definitions holds for suitably chosen generating functions f.

❓ Are there any other matrix sequences for which these definitions hold?

Asymptotic spectral distribution for non-Toeplitz sequences

For **Toeplitz matrices** we discovered that the following definitions holds for suitably chosen generating functions f.

❓ Are there any other matrix sequences for which these definitions hold?

1. Sequence of matrices describing the **energy on fractals**, *e.g.*, a version of the Szegö limit theorems on the Sierpiński gasket (Okoudjou, Rogers, and Strichartz 2010);

Asymptotic spectral distribution for non-Toeplitz sequences

For **Toeplitz matrices** we discovered that the following definitions holds for suitably chosen generating functions f.

❓ Are there any other matrix sequences for which these definitions hold?

1. Sequence of matrices describing the **energy on fractals**, *e.g.*, a version of the Szegö limit theorems on the Sierpiński gasket (Okoudjou, Rogers, and Strichartz 2010);
2. **L**ocally **T**oeplitz Sequences (Tilli 1998);

Asymptotic spectral distribution for non-Toeplitz sequences

For **Toeplitz matrices** we discovered that the following definitions holds for suitably chosen generating functions f.

❓ Are there any other matrix sequences for which these definitions hold?

1. Sequence of matrices describing the **energy on fractals**, *e.g.*, a version of the Szegö limit theorems on the Sierpiński gasket (Okoudjou, Rogers, and Strichartz 2010);
2. **L**ocally **T**oeplitz Sequences (Tilli 1998);
3. **G**eneralized **L**ocally **T**oeplitz Sequences (Garoni and Serra-Capizzano 2017, 2018).

Asymptotic spectral distribution for non-Toeplitz sequences

For **Toeplitz matrices** we discovered that the following definitions holds for suitably chosen generating functions f.

❓ Are there any other matrix sequences for which these definitions hold?

1. Sequence of matrices describing the **energy on fractals**, *e.g.*, a version of the Szegö limit theorems on the Sierpiński gasket (Okoudjou, Rogers, and Strichartz 2010);
2. **L**ocally **T**oeplitz Sequences (Tilli 1998);
3. **G**eneralized **L**ocally **T**oeplitz Sequences (Garoni and Serra-Capizzano 2017, 2018).

GLT Sequences

They are a $*$-algebra of matrix sequences $\{A_N\}_N$ to which we can extend some of the techniques and results we have briefly discussed for Toeplitz sequences. They can be used to describe **asymptotic spectral properties** of matrix sequences coming from the **discretization of differential equations** on **highly regular meshes**.

GLT Sequences (Garoni and Serra-Capizzano 2017, 2018)

- The **machinery** and the **relative notation** is unfortunately **cumbersome**.

GLT Sequences (without the agonizing pain)

☺ We need just **few tools** to get a couple of results for the case at hand.

GLT Sequences (without the agonizing pain)

☺ We need just **few tools** to get a couple of results for the case at hand.

Theorem (Axiomatic description) (Garoni and Serra-Capizzano 2017, 2018)

1. Each GLT sequence has a singular value symbol $f(x,\theta)$ for $(x,\theta) \in [0,1] \times [-\pi,\pi]$. If the sequence is Hermitian, then the distribution also holds in the eigenvalue sense. If $\{A_N\}_N$ has a GLT symbol $f(x,\theta)$ we will write $\{A_N\}_N \sim_{\mathrm{GLT}} f(x,\theta)$.
2. The set of GLT sequences form a $*$-algebra, i.e., it is closed under linear combinations, products, inversion (whenever the symbol is singular, at most, in a set of zero Lebesgue measure), and conjugation.
3. Every Toeplitz sequence generated by an $\mathbb{L}^1$ function $f = f(\theta)$ is a GLT sequence and its symbol is f. Every *diagonal sampling* matrix $(D_n)_{ii} = a(i/n)$ obtained from a continuous $a(x)$ is a GLT sequence and its symbol is a.
4. Every sequence which is distributed as the constant zero in the singular value sense is a GLT sequence with symbol 0.

GLT Sequences (without the agonizing pain)

☺ We need just **few tools** to get a couple of results for the case at hand.

Theorem (Axiomatic description) (Garoni and Serra-Capizzano 2017, 2018)

5. If $\{A_N\}_N \sim_{\mathrm{GLT}} \kappa$ and the matrices A_N are such that $A_N = X_N + Y_n$, where
   - every X_N is Hermitian,
   - the spectral norms of X_N and Y_N are uniformly bounded with respect to N,
   - the trace-norm of Y_N divided by the matrix size N converges to 0,

   then the distribution holds in the eigenvalue sense.

GLT Sequences (without the agonizing pain)

☺ We need just **few tools** to get a couple of results for the case at hand.

Theorem (Axiomatic description) (Garoni and Serra-Capizzano 2017, 2018)

5. If $\{A_N\}_N \sim_{\mathrm{GLT}} \kappa$ and the matrices A_N are such that $A_N = X_N + Y_n$, where
   - every X_N is Hermitian,
   - the spectral norms of X_N and Y_N are uniformly bounded with respect to N,
   - the trace-norm of Y_N divided by the matrix size N converges to 0,

   then the distribution holds in the eigenvalue sense.

? Okay, but what do we do with this stuff?

GLT Sequences (without the agonizing pain)

☺ We need just **few tools** to get a couple of results for the case at hand.

Theorem (Axiomatic description) (Garoni and Serra-Capizzano 2017, 2018)

5. If $\{A_N\}_N \sim_{\mathrm{GLT}} \kappa$ and the matrices A_N are such that $A_N = X_N + Y_n$, where
   - every X_N is Hermitian,
   - the spectral norms of X_N and Y_N are uniformly bounded with respect to N,
   - the trace-norm of Y_N divided by the matrix size N converges to 0,

   then the distribution holds in the eigenvalue sense.

? Okay, but what do we do with this stuff?

We take the sequence we have $\{A_n\}_n$ from our problem, and we try to show that it can be obtained via the $*$-algebra properties as the linear combination/product (with maybe some inversions and some zero distributed sequences) of GLT matrices of which we know the symbol (a.k.a., Toeplitz and diagonal matrices).

GLT Sequences (without the agonizing pain)

☺ We need just **few tools** to get a couple of results for the case at hand.

Theorem (Axiomatic description) (Garoni and Serra-Capizzano 2017, 2018)

5. If $\{A_N\}_N \sim_{\mathrm{GLT}} \kappa$ and the matrices A_N are such that $A_N = X_N + Y_n$, where
   - every X_N is Hermitian,
   - the spectral norms of X_N and Y_N are uniformly bounded with respect to N,
   - the trace-norm of Y_N divided by the matrix size N converges to 0,

   then the distribution holds in the eigenvalue sense.

❓ Okay, but what do we do with this stuff?

🔨 We take the sequence we have $\{A_n\}_n$ from our problem, and we try to show that it can be obtained via the $*$-algebra properties as the linear combination/product (with maybe some inversions and some zero distributed sequences) of GLT matrices of which we know the symbol (a.k.a., Toeplitz and diagonal matrices).

🪄 If we are successful, then we know the spectral distribution of our sequence.

GLT stuff for the case at hand

We want to discover the **GLT symbol**, a.k.a., the **spectral distribution** for the discretization of:

$$\begin{cases} \frac{\partial W}{\partial t} = d^+(x,t)\,{}^{RL}D^{\alpha}_{[0,x]}W(x,t) + d^-(x,t)\,{}^{RL}D^{\alpha}_{[x,1]}W(x,t), \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

GLT stuff for the case at hand

We want to discover the **GLT symbol**, a.k.a., the **spectral distribution** for:

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$

GLT stuff for the case at hand

We want to discover the **GLT symbol**, a.k.a., the **spectral distribution** for:

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$

Theorem (Donatelli, Mazza, and Serra-Capizzano 2016)

We assume $\nu = O(1)$, and that for a fixed instant of time t_m the functions $d^+(x,t) \equiv d^+(x)$ and $d^-(x,t) \equiv d^-(x)$ are both Riemann integrable over $[0,1]$, then

$$\{A_N\}_N \sim_{\mathrm{GLT}} h_\alpha(x,\theta) = d^+(x) f_\alpha(\theta) + d^-(x) f_\alpha(-\theta), \quad (x,\theta) \in [0,1] \times [-\pi,\pi].$$

GLT stuff for the case at hand

We want to discover the **GLT symbol**, a.k.a., the **spectral distribution** for:

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$

Theorem (Donatelli, Mazza, and Serra-Capizzano 2016)

We assume $\nu = O(1)$, and that for a fixed instant of time t_m the functions $d^+(x,t) \equiv d^+(x)$ and $d^-(x,t) \equiv d^-(x)$ are both Riemann integrable over $[0,1]$, then

$$\{A_N\}_N \sim_{\mathrm{GLT}} h_\alpha(x,\theta) = d^+(x) f_\alpha(\theta) + d^-(x) f_\alpha(-\theta), \quad (x,\theta) \in [0,1] \times [-\pi,\pi].$$

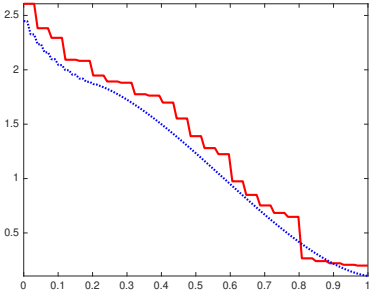
Proof. The diagonal elements of the matrices $D_N^\pm$ are a uniform sampling of the functions $d^\pm(x) \in [0,1]$, thus $D_N^\pm \sim_{\mathrm{GLT}} d^\pm(x)$. Toeplitz matrices G_N and G_N^T are also $\{G_N\}_N \sim_{\mathrm{GLT}} f_\alpha(\theta)$ and $\{G_N^T\}_N \sim_{\mathrm{GLT}} f_\alpha(-\theta)$. Finally $\{\nu I_N\}_N \sim_{\mathrm{GLT}} 0$ since $\nu = o(1)$ by hypothesis. The conclusion than follows from the $*$-algebra property, i.e.,

$$\{A_N\}_N \sim_{\mathrm{GLT}} 0 + d^+(x) p_\alpha(\theta) + d^-(x) p_\alpha(-\theta) = h_\alpha(x,\theta). \quad \square$$

GLT stuff for the case at hand

```
alpha = 1.5; N = 100;
hN = 1/(N-1);  x = 0:hN:1; dt = hN;
dplus=@(x)gamma(3-alpha).*x.^alpha;
dminus=@(x)gamma(3-alpha).*(1-x).^alpha;
G = glmatrix(N,alpha); % Discretize
Gr = G; Grt = G.'; I = eye(N,N);
Dplus = diag(dplus(x));
Dminus = diag(dminus(x));
nu = hN^alpha/dt;
A = nu*I -(Dplus*Gr + Dminus*Grt);
f = @(theta) -exp(-1i*theta).*...
(1-exp(1i*theta)).^alpha;
xsq = linspace(0,1,sqrt(N));
tsq = linspace(-pi,pi,sqrt(N));
[X,THETA] = meshgrid(xsq,tsq);
```

```
h = @(x,theta) nu +
↪  (dplus(x).*f(theta) ...
+ dminus(x).*f(-theta));
sv = svd(full(A));
```

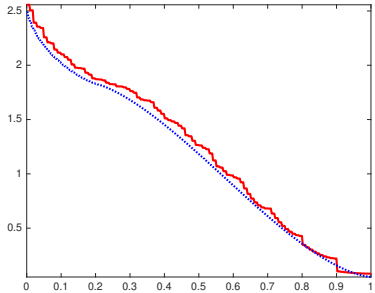


$N = 100$

GLT stuff for the case at hand

```
alpha = 1.5; N = 100;
hN = 1/(N-1);  x = 0:hN:1; dt = hN;
dplus=@(x)gamma(3-alpha).*x.^alpha;
dminus=@(x)gamma(3-alpha).*(1-x).^alpha;
G = glmatrix(N,alpha); % Discretize
Gr = G; Grt = G.'; I = eye(N,N);
Dplus = diag(dplus(x));
Dminus = diag(dminus(x));
nu = hN^alpha/dt;
A = nu*I -(Dplus*Gr + Dminus*Grt);
f = @(theta) -exp(-1i*theta).*...
(1-exp(1i*theta)).^alpha;
xsq = linspace(0,1,sqrt(N));
tsq = linspace(-pi,pi,sqrt(N));
[X,THETA] = meshgrid(xsq,tsq);
```

```
h = @(x,theta) nu +
↪  (dplus(x).*f(theta) ...
+ dminus(x).*f(-theta));
sv = svd(full(A));
```

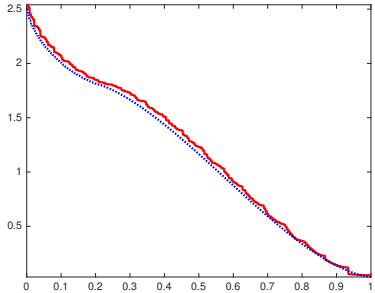


$N = 400$

GLT stuff for the case at hand

```
alpha = 1.5; N = 100;
hN = 1/(N-1);  x = 0:hN:1; dt = hN;
dplus=@(x)gamma(3-alpha).*x.^alpha;
dminus=@(x)gamma(3-alpha).*(1-x).^alpha;
G = glmatrix(N,alpha); % Discretize
Gr = G; Grt = G.'; I = eye(N,N);
Dplus = diag(dplus(x));
Dminus = diag(dminus(x));
nu = hN^alpha/dt;
A = nu*I -(Dplus*Gr + Dminus*Grt);
f = @(theta) -exp(-1i*theta).*...
(1-exp(1i*theta)).^alpha;
xsq = linspace(0,1,sqrt(N));
tsq = linspace(-pi,pi,sqrt(N));
[X,THETA] = meshgrid(xsq,tsq);
```

```
h = @(x,theta) nu +
↪  (dplus(x).*f(theta) ...
+ dminus(x).*f(-theta));
sv = svd(full(A));
```

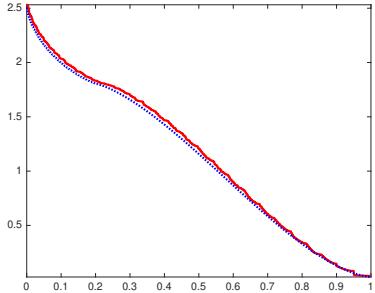


$N = 900$

GLT stuff for the case at hand

```
alpha = 1.5; N = 100;
hN = 1/(N-1);  x = 0:hN:1; dt = hN;
dplus=@(x)gamma(3-alpha).*x.^alpha;
dminus=@(x)gamma(3-alpha).*(1-x).^alpha;
G = glmatrix(N,alpha); % Discretize
Gr = G; Grt = G.'; I = eye(N,N);
Dplus = diag(dplus(x));
Dminus = diag(dminus(x));
nu = hN^alpha/dt;
A = nu*I -(Dplus*Gr + Dminus*Grt);
f = @(theta) -exp(-1i*theta).*...
(1-exp(1i*theta)).^alpha;
xsq = linspace(0,1,sqrt(N));
tsq = linspace(-pi,pi,sqrt(N));
[X,THETA] = meshgrid(xsq,tsq);
```

```
h = @(x,theta) nu +
↪  (dplus(x).*f(theta) ...
+ dminus(x).*f(-theta));
sv = svd(full(A));
```

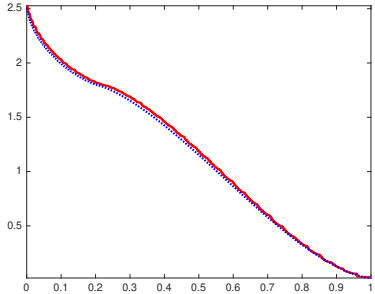


$N = 1600$

GLT stuff for the case at hand

```
alpha = 1.5; N = 100;
hN = 1/(N-1);  x = 0:hN:1; dt = hN;
dplus=@(x)gamma(3-alpha).*x.^alpha;
dminus=@(x)gamma(3-alpha).*(1-x).^alpha;
G = glmatrix(N,alpha); % Discretize
Gr = G; Grt = G.'; I = eye(N,N);
Dplus = diag(dplus(x));
Dminus = diag(dminus(x));
nu = hN^alpha/dt;
A = nu*I -(Dplus*Gr + Dminus*Grt);
f = @(theta) -exp(-1i*theta).*...
(1-exp(1i*theta)).^alpha;
xsq = linspace(0,1,sqrt(N));
tsq = linspace(-pi,pi,sqrt(N));
[X,THETA] = meshgrid(xsq,tsq);
```

```
h = @(x,theta) nu +
↪  (dplus(x).*f(theta) ...
+ dminus(x).*f(-theta));
sv = svd(full(A));
```

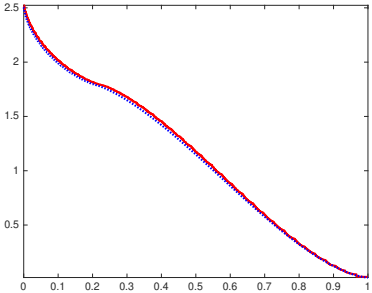


$N = 2500$

GLT stuff for the case at hand

```
alpha = 1.5; N = 100;
hN = 1/(N-1);  x = 0:hN:1; dt = hN;
dplus=@(x)gamma(3-alpha).*x.^alpha;
dminus=@(x)gamma(3-alpha).*(1-x).^alpha;
G = glmatrix(N,alpha); % Discretize
Gr = G; Grt = G.'; I = eye(N,N);
Dplus = diag(dplus(x));
Dminus = diag(dminus(x));
nu = hN^alpha/dt;
A = nu*I -(Dplus*Gr + Dminus*Grt);
f = @(theta) -exp(-1i*theta).*...
(1-exp(1i*theta)).^alpha;
xsq = linspace(0,1,sqrt(N));
tsq = linspace(-pi,pi,sqrt(N));
[X,THETA] = meshgrid(xsq,tsq);
```

```
h = @(x,theta) nu +
↪  (dplus(x).*f(theta) ...
+ dminus(x).*f(-theta));
sv = svd(full(A));
```

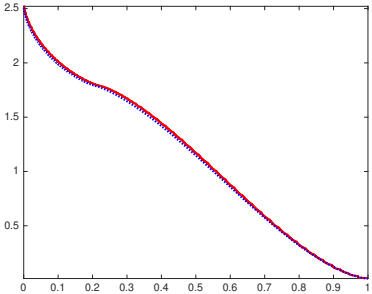


$N = 3600$

GLT stuff for the case at hand

```
alpha = 1.5; N = 100;
hN = 1/(N-1);  x = 0:hN:1; dt = hN;
dplus=@(x)gamma(3-alpha).*x.^alpha;
dminus=@(x)gamma(3-alpha).*(1-x).^alpha;
G = glmatrix(N,alpha); % Discretize
Gr = G; Grt = G.'; I = eye(N,N);
Dplus = diag(dplus(x));
Dminus = diag(dminus(x));
nu = hN^alpha/dt;
A = nu*I -(Dplus*Gr + Dminus*Grt);
f = @(theta) -exp(-1i*theta).*...
(1-exp(1i*theta)).^alpha;
xsq = linspace(0,1,sqrt(N));
tsq = linspace(-pi,pi,sqrt(N));
[X,THETA] = meshgrid(xsq,tsq);
```

```
h = @(x,theta) nu +
↪  (dplus(x).*f(theta) ...
+ dminus(x).*f(-theta));
sv = svd(full(A));
```

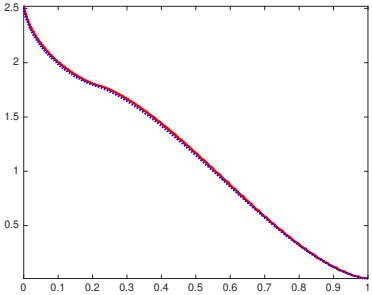


$N = 4900$

GLT stuff for the case at hand

```
alpha = 1.5; N = 100;
hN = 1/(N-1);  x = 0:hN:1; dt = hN;
dplus=@(x)gamma(3-alpha).*x.^alpha;
dminus=@(x)gamma(3-alpha).*(1-x).^alpha;
G = glmatrix(N,alpha); % Discretize
Gr = G; Grt = G.'; I = eye(N,N);
Dplus = diag(dplus(x));
Dminus = diag(dminus(x));
nu = hN^alpha/dt;
A = nu*I -(Dplus*Gr + Dminus*Grt);
f = @(theta) -exp(-1i*theta).*...
(1-exp(1i*theta)).^alpha;
xsq = linspace(0,1,sqrt(N));
tsq = linspace(-pi,pi,sqrt(N));
[X,THETA] = meshgrid(xsq,tsq);
```

```
h = @(x,theta) nu +
↪  (dplus(x).*f(theta) ...
+ dminus(x).*f(-theta));
sv = svd(full(A));
```



$N = 6400$

GLT: a negative result for circulant matrices

❓ And so we have the **asymptotic distribution** of our **singular values**, but what do we do with it?

GLT: a negative result for circulant matrices

❓ And so we have the **asymptotic distribution** of our **singular values**, but what do we do with it?

❗ The function

$$k(x,\theta) = d^+(x) f_\alpha(\theta) + d^-(x) f_\alpha(-\theta) \text{ for } (x,\theta) \in [0,1] \times [-\pi,\pi],$$

depends on **both** x and θ, on the other hand **any circulant preconditioner** will **depend only** on the θ **variable**!

GLT: a negative result for circulant matrices

❓ And so we have the **asymptotic distribution** of our **singular values**, but what do we do with it?

❗ The function

$$k(x,\theta) = d^+(x) f_\alpha(\theta) + d^-(x) f_\alpha(-\theta) \text{ for } (x,\theta) \in [0,1] \times [-\pi,\pi],$$

depends on **both** x and θ, on the other hand **any circulant preconditioner** will **depend only** on the θ **variable**!

$\Rightarrow$ No circulant preconditioner will ever cluster the singular values of a sequence with a "space variant" spectral distribution.

GLT: a negative result for circulant matrices

❓ And so we have the **asymptotic distribution** of our **singular values**, but what do we do with it?

❗ The function

$$k(x,\theta) = d^+(x) f_\alpha(\theta) + d^-(x) f_\alpha(-\theta) \text{ for } (x,\theta) \in [0,1] \times [-\pi,\pi],$$

depends on **both** x and θ, on the other hand **any circulant preconditioner** will **depend only** on the θ **variable**!

$\Rightarrow$ No circulant preconditioner will ever cluster the singular values of a sequence with a "space variant" spectral distribution.

❓ What type of preconditioner can we use to solve this issue?

Structure preserving preconditioners

The GLT class of sequences is a $*$-algebra, thus we can try to **proecondition** the sequence $\{A_N\}_N$ with **something from the same class**. We then look for:

Structure preserving preconditioners

The GLT class of sequences is a $*$-algebra, thus we can try to **proecondition** the sequence $\{A_N\}_N$ with **something from the same class**. We then look for:

- A sequence $\{P_N\}_N$ in the GLT class,

Structure preserving preconditioners

The GLT class of sequences is a $*$-algebra, thus we can try to **proecondition** the sequence $\{A_N\}_N$ with **something from the same class**. We then look for:

- A sequence $\{P_N\}_N$ in the GLT class,
- A sequence $\{P_N\}_N$ such that $\{P_N^{-1}A_N\}_N \sim_{\mathrm{GLT}} 1$,

Structure preserving preconditioners

The GLT class of sequences is a $*$-algebra, thus we can try to **proecondition** the sequence $\{A_N\}_N$ with **something from the same class**. We then look for:

- A sequence $\{P_N\}_N$ in the GLT class,
- A sequence $\{P_N\}_N$ such that $\{P_N^{-1}A_N\}_N \sim_{\mathrm{GLT}} 1$,
- A sequence $\{P_N\}_N$ that is *easy enough* to invert.

Structure preserving preconditioners

The GLT class of sequences is a $*$-algebra, thus we can try to **proecondition** the sequence $\{A_N\}_N$ with **something from the same class**. We then look for:

- A sequence $\{P_N\}_N$ in the GLT class,
- A sequence $\{P_N\}_N$ such that $\{P_N^{-1}A_N\}_N \sim_{\mathrm{GLT}} 1$,
- A sequence $\{P_N\}_N$ that is *easy enough* to invert.

An old idea anew

This a modification of an old idea, if we take a Toeplitz system $T_n(f)$ then we can use $T_n(1/f)$ as a preconditioner!

- $P_n^{-1} = T_n(1/f)$ **is not the inverse** of $T_n(f)$,
- If we have $T_n(1/f)$, its application cost is $O(n \log n)$,

Structure preserving preconditioners

The GLT class of sequences is a $*$-algebra, thus we can try to **proecondition** the sequence $\{A_N\}_N$ with **something from the same class**. We then look for:

- A sequence $\{P_N\}_N$ in the GLT class,
- A sequence $\{P_N\}_N$ such that $\{P_N^{-1}A_N\}_N \sim_{\mathrm{GLT}} 1$,
- A sequence $\{P_N\}_N$ that is *easy enough* to invert.

An old idea anew

This a modification of an old idea, if we take a Toeplitz system $T_n(f)$ then we can use $T_n(1/f)$ as a preconditioner!

- $P_n^{-1} = T_n(1/f)$ **is not the inverse** of $T_n(f)$,
- If we have $T_n(1/f)$, its application cost is $O(n \log n)$,
- Computing the Fourier coefficients of $1/f$ can be expensive.

Preconditioning Toeplitz with Toeplitz

We have expressed the Fourier coefficients of f as

$$t_k = \frac{1}{2\pi}\int_0^{2\pi} f(\theta)e^{-ik\theta}\,\mathrm{d}\theta, \qquad k = 0, \pm 1, \pm 2, \ldots,$$

we say that f is

- of **analytic type** if $t_k = 0$ for $k < 0$, or
- of **coanalytic type** if $t_k = 0$ for $k > 0$.

Lemma

Let f be of analytic type (or respectively coanalytic type) and $a_0 \neq 0$. Then $T_n(f)$ is invertible if and only if $1/f$ is bounded and of analytic type (or respectively coanalytic type). In either case, we have $T_n(1/f)T_n(f) = T_n(f)T_n(1/f) = I_n$, for I_n is the identity matrix.

Preconditioning Toeplitz with Toeplitz

Lemma (Chan and Ng 1993)

Let f be a **positive** trigonometric polynomial of degree K

$$f(\theta) = \sum_{k=-K}^{K} t_k e^{ik\theta}.$$

Then for $n > 2K$, $\mathrm{rank}(T_n(1/f)T_n(f) - I_n) \leq 2K$.

Proof. Let

$$\frac{1}{f(\theta)} = \sum_{k=-\infty}^{+\infty} \rho_k e^{ik\theta}$$

Preconditioning Toeplitz with Toeplitz

Lemma (Chan and Ng 1993)

Let f be a **positive** trigonometric polynomial of degree K

$$f(\theta) = \sum_{k=-K}^{K} t_k e^{ik\theta}.$$

Then for $n > 2K$, $\mathrm{rank}(T_n(1/f)T_n(f) - I_n) \leq 2K$.

Proof. Let

$$\frac{1}{f(\theta)} = \sum_{k=-\infty}^{+\infty} \rho_k e^{ik\theta} \Rightarrow \sum_{k=-K}^{K} t_k \rho_{m-k} = \begin{cases} 1, & \text{if } m = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Preconditioning Toeplitz with Toeplitz

Lemma (Chan and Ng 1993)

Let f be a **positive** trigonometric polynomial of degree K

$$f(\theta) = \sum_{k=-K}^{K} t_k e^{ik\theta}.$$

Then for $n > 2K$, $\mathrm{rank}(T_n(1/f)T_n(f) - I_n) \leq 2K$.

Proof. Let

$$\frac{1}{f(\theta)} = \sum_{k=-\infty}^{+\infty} \rho_k e^{ik\theta} \Rightarrow \sum_{k=-K}^{K} t_k \rho_{m-k} = \begin{cases} 1, & \text{if } m = 0, \\ 0, & \text{otherwise.} \end{cases}$$

Thus for $n > 2K$, the entries of $T_n(1/f)T_n(f) - I_n$ are all zeros except possibly entries in its first and last K columns. $\square$

Preconditioning Toeplitz with Toeplitz

Given $|\alpha| < 1$ consider

$$f(\theta) = \frac{1 + \alpha^2 - \alpha e^{i\theta} - \alpha e^{-i\theta}}{1 - \alpha^2}$$

$T_n(f)$ is tridiagonal and SPD.

```
function T = kacmatrix(n,alpha)
%KACMATRIX Kac-Murdock-Szego matrices
e = ones(n,1);
T = spdiags(([-alpha,1+alpha^2,-alpha]
↪  ./(1-alpha^2)).*e,-1:1,n,n);
end
```

We can express

$$\frac{1}{f(\theta)} = \sum_{k=-\infty}^{+\infty} t^{|k|} e^{ik\theta} = \frac{1 - \alpha^2}{(1 - \alpha e^{i\theta})(1 - \alpha e^{-i\theta})},$$

and $T_n(1/f)$ is then a **dense Toeplitz matrix**.

Preconditioning Toeplitz with Toeplitz

We can compute the coefficients in an **inefficient way** and apply it to the CG/PCG

N	CG	PCG
32	20	2
64	20	2
128	20	2
256	20	2
512	20	2
1024	20	2
2048	20	2

$\alpha = 0.5$

```
function T = invkacmatrix(n,alpha)
%INVKACMATRIX Gives back the 1/Kac-Murdock-Szego
↪  matrices
f = @(th) (1 - alpha^2)./((1-alpha*exp(1i*th))
↪  .*(1-alpha*exp(-1i*th)));
c = zeros(n,1); r = zeros(1,n);
for k=1:n
 r(k) = integral(@(th) f(th).*exp(1i*th*(k-1)),0,2*pi)
↪  /(2*pi);
 c(k) = integral(@(th) f(th).*exp(-1i*th*(k-1)),0,2*pi)
↪  /(2*pi);
end
T = real(toeplitz(r,c));
end
```

Preconditioning Toeplitz with Toeplitz

We can compute the coefficients in an **inefficient way** and apply it to the CG/PCG

N	CG	PCG
32	6	2
64	6	2
128	6	2
256	6	2
512	6	2
1024	6	2
2048	6	2

$\alpha = 0.1$

```
function T = invkacmatrix(n,alpha)
%INVKACMATRIX Gives back the 1/Kac-Murdock-Szego
↪  matrices
f = @(th) (1 - alpha^2)./((1-alpha*exp(1i*th))
↪  .*(1-alpha*exp(-1i*th)));
c = zeros(n,1); r = zeros(1,n);
for k=1:n
 r(k) = integral(@(th) f(th).*exp(1i*th*(k-1)),0,2*pi)
↪  /(2*pi);
 c(k) = integral(@(th) f(th).*exp(-1i*th*(k-1)),0,2*pi)
↪  /(2*pi);
end
T = real(toeplitz(r,c));
end
```

Preconditioning Toeplitz with Toeplitz

We can compute the coefficients in an **inefficient way** and apply it to the CG/PCG

N	CG	PCG
32	20	3
64	20	2
128	20	2
256	20	2
512	20	2
1024	20	2
2048	20	2

$\alpha = 0.8$

```
function T = invkacmatrix(n,alpha)
%INVKACMATRIX Gives back the 1/Kac-Murdock-Szego
↪  matrices
f = @(th) (1 - alpha^2)./((1-alpha*exp(1i*th))
↪  .*(1-alpha*exp(-1i*th)));
c = zeros(n,1); r = zeros(1,n);
for k=1:n
 r(k) = integral(@(th) f(th).*exp(1i*th*(k-1)),0,2*pi)
↪  /(2*pi);
 c(k) = integral(@(th) f(th).*exp(-1i*th*(k-1)),0,2*pi)
↪  /(2*pi);
end
T = real(toeplitz(r,c));
end
```

Preconditioning Toeplitz with Toeplitz

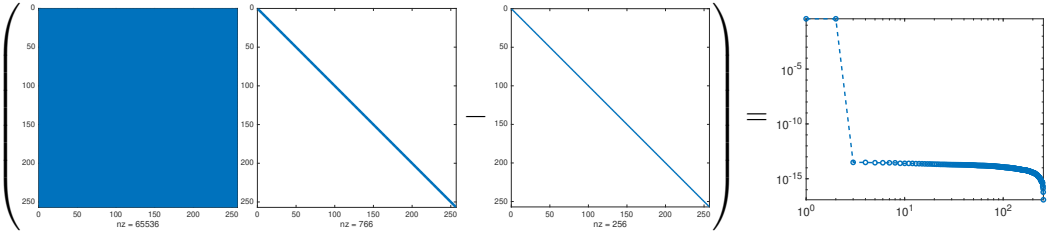
We can compute the coefficients in an **inefficient way** and apply it to the CG/PCG

$$\mathrm{rank}\,(T_n(1/f)\,T_n(f) - I_n) = 2$$

Preconditioning Toeplitz with Toeplitz

We can compute the coefficients in an **inefficient way** and apply it to the CG/PCG

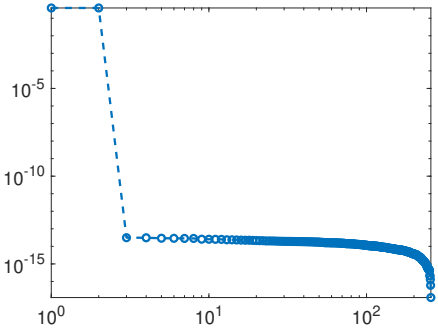
rank



Preconditioning Toeplitz with Toeplitz

We can compute the coefficients in an **inefficient way** and apply it to the CG/PCG

$$\mathrm{rank}\,(T_n(1/f)\,T_n(f) - I_n) =$$



An **exercise** to make the evaluation and construction of the involved quantities would be using the `fft` to compute the Fourier coefficients of $1/f(\theta)$.

Preconditioning Toeplitz with Toeplitz

Lemma (Chan and Ng 1993)

Let f be a positive 2π-periodic continuous function. Then for all $\varepsilon > 0$, there exists positive integers M and N such that for all $n > N$,

$$T_n(1/f)\,T_n(f) = I_n + L_n + U_n, \text{where}\, \mathrm{rank}(L_n) \leq M \text{ and } \|U_n\|_2 < \varepsilon.$$

Proof. By the Weierstrass Theorem, there exists a positive trigonometric polynomial

$$p_K(\theta) = \sum_{k=-K}^{+K} \rho_k e^{ik\theta}, \quad \rho_{-k} = \overline{\rho}_k, \text{ such that } f_{\min}/2 \leq p_K(\theta) \leq 2f_{\max}\ \forall\, \theta \in [0, 2\pi], \text{ and}$$

$$\max_{\theta\in[0,2\pi]} |f(\theta) - p_K(\theta)| \leq \frac{f_{\min}}{2}(-1 + \sqrt{1+\varepsilon}) \min\left\{\frac{f_{\min}}{2f_{\max}}, 1\right\}.$$

Preconditioning Toeplitz with Toeplitz

Lemma (Chan and Ng 1993)

Let f be a positive 2π-periodic continuous function. Then for all $\varepsilon > 0$, there exists positive integers M and N such that for all $n > N$,

$$T_n(1/f)\,T_n(f) = I_n + L_n + U_n, \text{where}\, \mathrm{rank}(L_n) \leq M \text{ and } \|U_n\|_2 < \varepsilon.$$

Proof. We write

$$\begin{aligned} T_n(1/f)\,T_n(f) &= T_n(1/f)\,T_n^{-1}(1/p_K)\,T_n(1/p_K)\,T_n(p_K)\,T_n^{-1}(p_K)\,T_n(f) \\ &= (I_n + V_n)\,(T_n(1/p_K)\,T_n(p_K))\,(I_n + W_n) \end{aligned}$$

where $V_n = (T_n(1/f) - T_n(1/p_K)\,T_n^{-1}(1/p_K))$ and $W_n = T_n^{-1}(p_k)(T_n(f) - T_n(p_K))$

Preconditioning Toeplitz with Toeplitz

Lemma (Chan and Ng 1993)

Let f be a positive 2π-periodic continuous function. Then for all $\varepsilon > 0$, there exists positive integers M and N such that for all $n > N$,

$$T_n(1/f)\,T_n(f) = I_n + L_n + U_n, \text{where}\, \mathrm{rank}(L_n) \leq M \text{ and } \|U_n\|_2 < \varepsilon.$$

Proof. We write

$$T_n(1/f)\,T_n(f) = (I_n + V_n)\,(T_n(1/p_K)\,T_n(p_K))\,(I_n + W_n)$$

and by the property of the generating functions and the Weierstrass Theorem

$$\|T_n^{-1}(p_K)\|_2 \leq \frac{2}{f_{\min}},\ \|T_n^{-1}(1/p_K)\|_2 \leq 2f_{\max},\ \|T_n(f) - T_n(p_K)\|_2 \leq \frac{(-1+\sqrt{1+\varepsilon})f_{\min}}{2},$$

$$\|T_n(1/f) - T_n(1/p_K)\|_2 \leq \max_{\theta in[0,2\pi]} \left|\frac{1}{f(\theta)} - \frac{1}{p_K(\theta)}\right| \leq \frac{2}{f_{\min}^2} \max_{\theta\in[0,2\pi]} |f(\theta) - p_K(\theta)|$$

Preconditioning Toeplitz with Toeplitz

Lemma (Chan and Ng 1993)

Let f be a positive 2π-periodic continuous function. Then for all $\varepsilon > 0$, there exists positive integers M and N such that for all $n > N$,

$$T_n(1/f)\,T_n(f) = I_n + L_n + U_n, \text{where}\, \mathrm{rank}(L_n) \leq M \text{ and } \|U_n\|_2 < \varepsilon.$$

Proof. We write

$$T_n(1/f)\,T_n(f) = (I_n + V_n)\,(T_n(1/p_K)\,T_n(p_K))\,(I_n + W_n)$$

and by the property of the generating functions and the Weierstrass Theorem

$$\|T_n^{-1}(p_K)\|_2 \leq \frac{2}{f_{\min}},\ \|T_n^{-1}(1/p_K)\|_2 \leq 2f_{\max},\ \|T_n(f) - T_n(p_K)\|_2 \leq \frac{(-1+\sqrt{1+\varepsilon})f_{\min}}{2},$$

$$\|T_n(1/f) - T_n(1/p_K)\|_2 \leq \frac{2}{f_{\min}^2} \max_{\theta\in[0,2\pi]} |f(\theta) - p_K(\theta)| \leq \frac{-1+\sqrt{1+\varepsilon}}{2f_{\max}}.$$

Preconditioning Toeplitz with Toeplitz

Lemma (Chan and Ng 1993)

Let f be a positive 2π-periodic continuous function. Then for all $\varepsilon > 0$, there exists positive integers M and N such that for all $n > N$,

$$T_n(1/f)\,T_n(f) = I_n + L_n + U_n, \text{where}\, \mathrm{rank}(L_n) \leq M \text{ and } \|U_n\|_2 < \varepsilon.$$

Proof. We write

$$T_n(1/f)\,T_n(f) = (I_n + V_n)\,(T_n(1/p_K)\,T_n(p_K))\,(I_n + W_n).$$

Using the **lemma on trigonometric polynomials** and using $n > 2K$ we have

$$T_n(1/p_K)\,T_n(p_K) = I_n + \tilde{L}_n \text{ with } \mathrm{rank}(\tilde{L}_n) \leq 2K.$$

Preconditioning Toeplitz with Toeplitz

Lemma (Chan and Ng 1993)

Let f be a positive 2π-periodic continuous function. Then for all $\varepsilon > 0$, there exists positive integers M and N such that for all $n > N$,

$$T_n(1/f)\,T_n(f) = I_n + L_n + U_n, \text{where}\, \mathrm{rank}(L_n) \leq M \text{ and } \|U_n\|_2 < \varepsilon.$$

Proof. We write

$$T_n(1/f)\,T_n(f) = (I_n + V_n)(I_n + \tilde{L}_n)(I_n + W_n).$$

Using the **lemma on trigonometric polynomials** and using $n > 2K$ we have

$$T_n(1/p_K)\,T_n(p_K) = I_n + \tilde{L}_n \text{ with } \mathrm{rank}(\tilde{L}_n) \leq 2K.$$

Preconditioning Toeplitz with Toeplitz

Lemma (Chan and Ng 1993)

Let f be a positive 2π-periodic continuous function. Then for all $\varepsilon > 0$, there exists positive integers M and N such that for all $n > N$,

$$T_n(1/f)\,T_n(f) = I_n + L_n + U_n, \text{where } \mathrm{rank}(L_n) \leq M \text{ and } \|U_n\|_2 < \varepsilon.$$

Proof. We write

$$T_n(1/f)\,T_n(f) = (I_n + V_n)(I_n + \tilde{L}_n)(I_n + W_n) \equiv I_n + L_n + U_n,$$

where

$$U_n = V_n + W_n + V_n W_n, \quad L_n = \tilde{L}_n(I_n + W_n) + V_n \tilde{L}_n(I_n + W_n),$$

and using the previous relations

$$\mathrm{rank}(L_n) \leq 4K, \text{ and } \|U_n\|_2 \leq \varepsilon. \quad \square$$

Preconditioning Toeplitz with Toeplitz

Theorem (Chan and Ng 1993)

Let f be a **positive** 2π-periodic continuous function. Then for all $\varepsilon > 0$, there exist positive integers M and N such that for all $n > N$, at most M eigenvalues of $T_n(1/f)\,T_n(f) - I_n$ have absolute value greater than ε.

Proof (idea). The HPD matrix $X_n = T_n^{1/2}(1/f)\,T_n(f)\,T_n^{1/2}(1/f) \sim T_n(1/f)\,T_n(f)$. Use the decomposition of the previous Theorem and the uniform boundedness of $T_n^{\pm 1/2}(1/f)$. $\square$

Preconditioning Toeplitz with Toeplitz

Theorem (Chan and Ng 1993)

Let f be a **positive** 2π-periodic continuous function. Then for all $\varepsilon > 0$, there exist positive integers M and N such that for all $n > N$, at most M eigenvalues of $T_n(1/f)\,T_n(f) - I_n$ have absolute value greater than ε.

Proof (idea). The HPD matrix $X_n = T_n^{1/2}(1/f)\,T_n(f)\,T_n^{1/2}(1/f) \sim T_n(1/f)\,T_n(f)$. Use the decomposition of the previous Theorem and the uniform boundedness of $T_n^{\pm 1/2}(1/f)$. $\square$

- We still need **positive** generating functions,

Preconditioning Toeplitz with Toeplitz

Theorem (Chan and Ng 1993)

Let f be a **positive** 2π-periodic continuous function. Then for all $\varepsilon > 0$, there exist positive integers M and N such that for all $n > N$, at most M eigenvalues of $T_n(1/f)\,T_n(f) - I_n$ have absolute value greater than ε.

Proof (idea). The HPD matrix $X_n = T_n^{1/2}(1/f)\,T_n(f)\,T_n^{1/2}(1/f) \sim T_n(1/f)\,T_n(f)$. Use the decomposition of the previous Theorem and the uniform boundedness of $T_n^{\pm 1/2}(1/f)$. $\square$

- We still need **positive** generating functions,
- If f is not given explicitly or the evaluation of $1/f(\theta)$ are costly the approach is infeasible.

Preconditioning Toeplitz with Toeplitz

Theorem (Chan and Ng 1993)

Let f be a **positive** 2π-periodic continuous function. Then for all $\varepsilon > 0$, there exist positive integers M and N such that for all $n > N$, at most M eigenvalues of $T_n(1/f)\,T_n(f) - I_n$ have absolute value greater than ε.

Proof (idea). The HPD matrix $X_n = T_n^{1/2}(1/f)\,T_n(f)\,T_n^{1/2}(1/f) \sim T_n(1/f)\,T_n(f)$. Use the decomposition of the previous Theorem and the uniform boundedness of $T_n^{\pm 1/2}(1/f)$. $\square$

- We still need **positive** generating functions,
- If f is not given explicitly or the evaluation of $1/f(\theta)$ are costly the approach is infeasible.
- The **idea** from (Chan and Ng 1993) is to reduce the cost of working with f and $1/f$ by using convolution products with Kernel functions.

Preconditioning GLT with GLT

GLT sequences are a $*$-algebra, some of the analysis is therefore greatly simplified.

Theorem (Garoni and Serra-Capizzano 2017, Section 8.4)

Let $\{A_N\}_N$ be a sequence of Hermitian matrices such that $\{A_N\}_N \sim_{GLT} \kappa$, and let $\{P_N\}_N$ be a sequence of Hermitian positive definite matrices such that $\{P_N\}_N \sim_{GLT} \xi$, and $\xi \neq 0$ a.e. Then

$$\{P_N^{-1} A_N\}_N \sim_{\mathrm{GLT}} \xi^{-1}\kappa, \qquad \{P_N^{-1} A_N\}_N \sim_{\sigma,\lambda} (\xi^{-1}\kappa, \mathcal{I}^d).$$

Preconditioning GLT with GLT

GLT sequences are a $*$-algebra, some of the analysis is therefore greatly simplified.

Theorem (Garoni and Serra-Capizzano 2017, Section 8.4)

Let $\{A_N\}_N$ be a sequence of Hermitian matrices such that $\{A_N\}_N \sim_{GLT} \kappa$, and let $\{P_N\}_N$ be a sequence of Hermitian positive definite matrices such that $\{P_N\}_N \sim_{GLT} \xi$, and $\xi \neq 0$ a.e. Then

$$\{P_N^{-1} A_N\}_N \sim_{\mathrm{GLT}} \xi^{-1}\kappa, \qquad \{P_N^{-1} A_N\}_N \sim_{\sigma,\lambda} (\xi^{-1}\kappa, \mathcal{I}^d).$$

☺ We need less than positive!

Preconditioning GLT with GLT

GLT sequences are a $*$-algebra, some of the analysis is therefore greatly simplified.

Theorem (Garoni and Serra-Capizzano 2017, Section 8.4)

Let $\{A_N\}_N$ be a sequence of Hermitian matrices such that $\{A_N\}_N \sim_{GLT} \kappa$, and let $\{P_N\}_N$ be a sequence of Hermitian positive definite matrices such that $\{P_N\}_N \sim_{GLT} \xi$, and $\xi \neq 0$ a.e. Then

$$\{P_N^{-1} A_N\}_N \sim_{\mathrm{GLT}} \xi^{-1}\kappa, \qquad \{P_N^{-1} A_N\}_N \sim_{\sigma,\lambda} (\xi^{-1}\kappa, \mathcal{I}^d).$$

- We need less than positive!
- If we move to the non-symmetric case, we are left just with a relation with respect to the singular values.

Preconditioning GLT with GLT

GLT sequences are a $*$-algebra, some of the analysis is therefore greatly simplified.

Theorem (Garoni and Serra-Capizzano 2017, Section 8.4)

Let $\{A_N\}_N$ be a sequence of Hermitian matrices such that $\{A_N\}_N \sim_{GLT} \kappa$, and let $\{P_N\}_N$ be a sequence of Hermitian positive definite matrices such that $\{P_N\}_N \sim_{GLT} \xi$, and $\xi \neq 0$ a.e. Then

$$\{P_N^{-1}A_N\}_N \sim_{\mathrm{GLT}} \xi^{-1}\kappa, \qquad \{P_N^{-1}A_N\}_N \sim_{\sigma,\lambda} (\xi^{-1}\kappa, \mathcal{I}^d).$$

- We need less than positive!
- If we move to the non-symmetric case, we are left just with a relation with respect to the singular values.
- The **general idea for a GLT preconditioner** is then to find a GLT sequence $\{P_N\}_N$
  - that is easy to invert,
  - and such that $\xi^1\kappa = 1$ or *at least* a quantity bounded and bounded away from zero.

Preconditioning GLT with GLT

Let us finally go back to our case of interest

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$

we build a preconditioner with the **same structure** such that

⚙ we have a *small bandwidth* $\Rightarrow$ a **small computational cost**,

Preconditioning GLT with GLT

Let us finally go back to our case of interest

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$

we build a preconditioner with the **same structure** such that

- we have a *small bandwidth* $\Rightarrow$ a **small computational cost**,
- the symbol of a bandwidth Toeplitz matrix is a trigonometric polynomial, hence the **zero of the symbol cannot be of fractional order**.

Preconditioning GLT with GLT

Let us finally go back to our case of interest

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$

we build a preconditioner with the **same structure** such that

- we have a *small bandwidth* $\Rightarrow$ a **small computational cost**,
- the symbol of a bandwidth Toeplitz matrix is a trigonometric polynomial, hence the **zero of the symbol cannot be of fractional order**.
- $P_{1,N} = \nu I + D_N^+ B_N + D_N^- B_N^T$, $B_n = T_n(1 - exp(-i\theta))$,

Preconditioning GLT with GLT

Let us finally go back to our case of interest

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$

we build a preconditioner with the **same structure** such that

- we have a *small bandwidth* $\Rightarrow$ a **small computational cost**,
- the symbol of a bandwidth Toeplitz matrix is a trigonometric polynomial, hence the **zero of the symbol cannot be of fractional order**.
- $P_{1,N} = \nu I + D_N^+ B_N + D_N^- B_N^T$, $B_n = T_n(1 - exp(-i\theta))$,
- $P_{2,N} = \nu I + D_N^+ L_N + D_N^- L_N^T$, $B_n = T_n(2 - 2\cos(\theta))$

Preconditioning GLT with GLT

Let us finally go back to our case of interest

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$

we build a preconditioner with the **same structure** such that

- we have a *small bandwidth* $\Rightarrow$ a **small computational cost**,
- the symbol of a bandwidth Toeplitz matrix is a trigonometric polynomial, hence the **zero of the symbol cannot be of fractional order**.
- $P_{1,N} = \nu I + D_N^+ B_N + D_N^- B_N^T$, $B_n = T_n(1 - exp(-i\theta))$,
- $\{P_{1,N}\}_N \sim_{GLT} p_1(x,\theta) = d_+(x)(1 - e^{-i\theta}) + d_-(x)(1 - e^{i\theta})$, holds only in the singular value sense!
- $P_{2,N} = \nu I + D_N^+ L_N + D_N^- L_N^T$, $B_n = T_n(2 - 2\cos(\theta))$

Preconditioning GLT with GLT

Let us finally go back to our case of interest

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$

we build a preconditioner with the **same structure** such that

- we have a *small bandwidth* $\Rightarrow$ a **small computational cost**,
- the symbol of a bandwidth Toeplitz matrix is a trigonometric polynomial, hence the **zero of the symbol cannot be of fractional order**.
- $P_{1,N} = \nu I + D_N^+ B_N + D_N^- B_N^T$, $B_n = T_n(1 - exp(-i\theta))$,
- $\{P_{1,N}\}_N \sim_{GLT} p_1(x,\theta) = d_+(x)(1 - e^{-i\theta}) + d_-(x)(1 - e^{i\theta})$, holds only in the singular value sense!
- $P_{2,N} = \nu I + D_N^+ L_N + D_N^- L_N^T$, $B_n = T_n(2 - 2\cos(\theta))$
- $\{P_{2,N}\}_N \sim_{GLT} p_2(x,\theta) = (d_+(x) + d_-(x))(2 - 2\cos(\theta))$, holds also in the eigenvalue sense!

Preconditioning GLT with GLT

Since the symbol of a bandwidth Toeplitz matrix is a trigonometric polynomial, hence the **zero of the symbol cannot be of fractional order**:

$$d_{\pm}(x,t) = d > 0 \; : \; \lim_{\theta \to 0} \frac{h(x,\theta)}{p_k(x,\theta)} = +\infty, \qquad k \in \{1,2\}.$$

Preconditioning GLT with GLT

Since the symbol of a bandwidth Toeplitz matrix is a trigonometric polynomial, hence the **zero of the symbol cannot be of fractional order**:

$$d_{\pm}(x,t) = d > 0 \; : \; \lim_{\theta \to 0} \frac{h(x,\theta)}{p_k(x,\theta)} = +\infty, \qquad k \in \{1,2\}.$$

Theorem (Serra 1995, Theorem 3.1)

Let f be an integrable function defined on $[-\pi,\pi]$ having in $x = x_0$ the unique zero of order ρ. Then, by choosing $2k$ the even number which minimizes the distance from ρ and setting $g = |x - x_0|^{2k}$, the condition number of $T_n(g)^{-1}T_n(f)$ is asymptotical to $n^{2k-\rho}$.

Preconditioning GLT with GLT

Since the symbol of a bandwidth Toeplitz matrix is a trigonometric polynomial, hence the **zero of the symbol cannot be of fractional order**:

$$d_{\pm}(x,t) = d > 0 \; : \; \lim_{\theta \to 0} \frac{h(x,\theta)}{p_k(x,\theta)} = +\infty, \qquad k \in \{1,2\}.$$

Theorem (Serra 1995, Theorem 3.1)

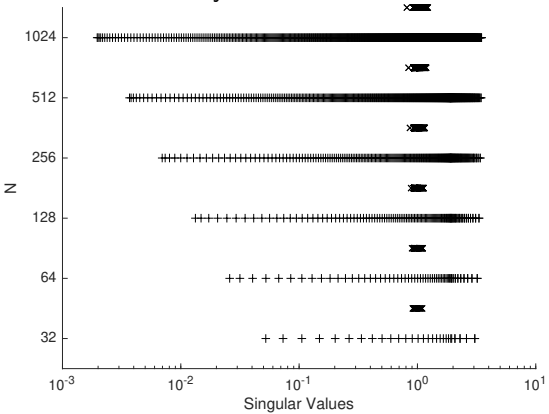
Let f be an integrable function defined on $[-\pi,\pi]$ having in $x = x_0$ the unique zero of order ρ. Then, by choosing $2k$ the even number which minimizes the distance from ρ and setting $g = |x - x_0|^{2k}$, the condition number of $T_n(g)^{-1}T_n(f)$ is asymptotical to $n^{2k-\rho}$.

In our case

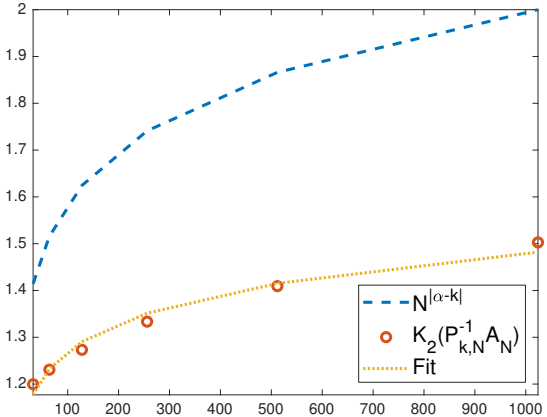
We expect the condition number of the preconditioned matrix to be $O(N^{|\alpha-k|})$, $k \in \{1,2\}$.

Preconditioning GLT with GLT

Let's numerically test our idea.

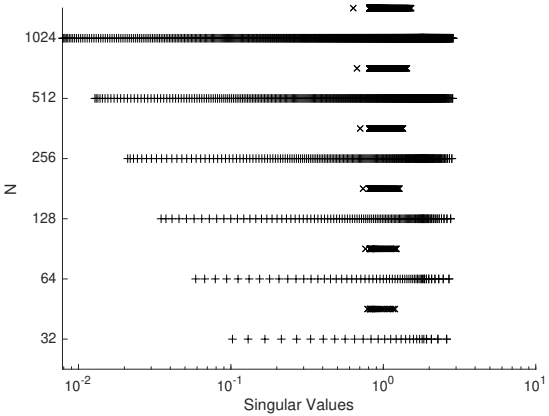


$\alpha = 1.9,\ k = 2$

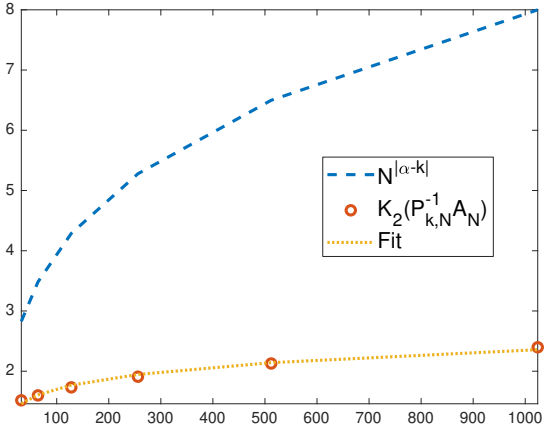


Preconditioning GLT with GLT

Let's numerically test our idea.



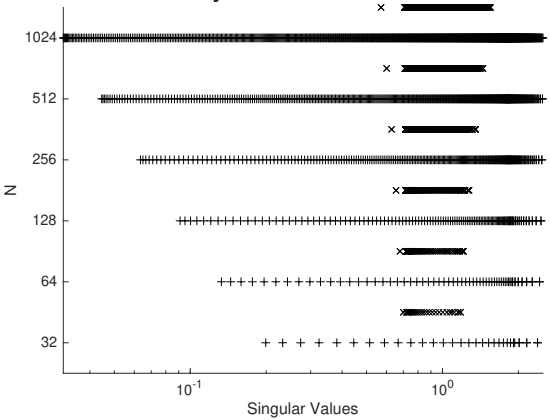
$\alpha = 1.7,\ k = 2$



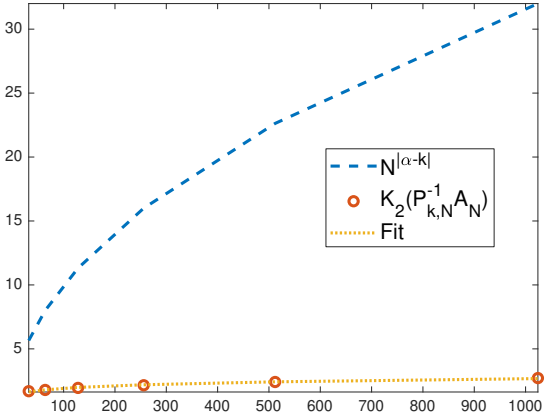
$\alpha = 1.7,\ k = 2$

Preconditioning GLT with GLT

Let's numerically test our idea.



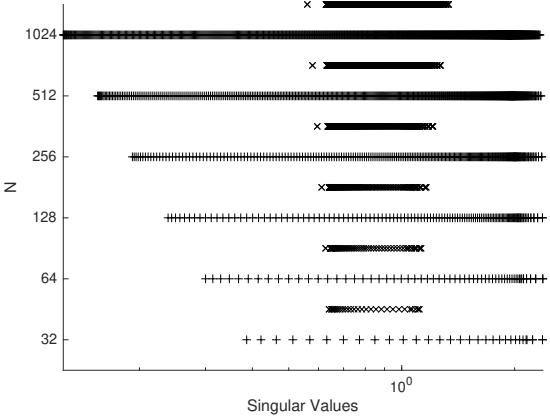
$\alpha = 1.5,\ k = 2$



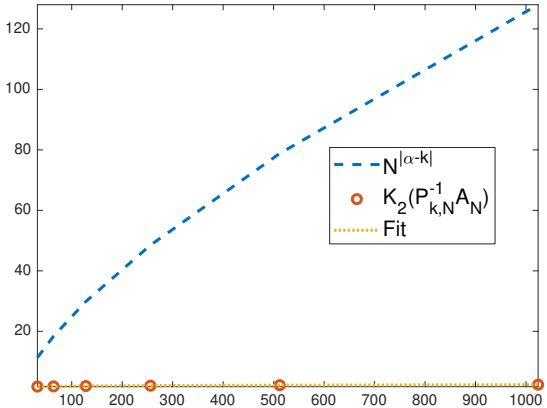
$\alpha = 1.5,\ k = 2$

Preconditioning GLT with GLT

Let's numerically test our idea.



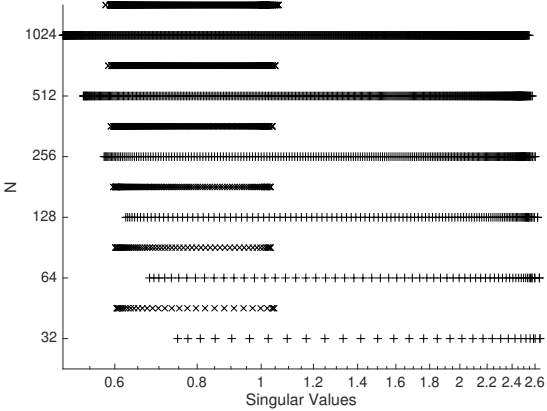
$\alpha = 1.3,\ k = 2$



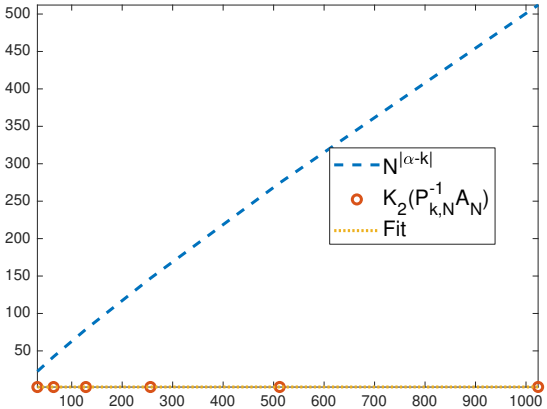
$\alpha = 1.3,\ k = 2$

Preconditioning GLT with GLT

Let's numerically test our idea.



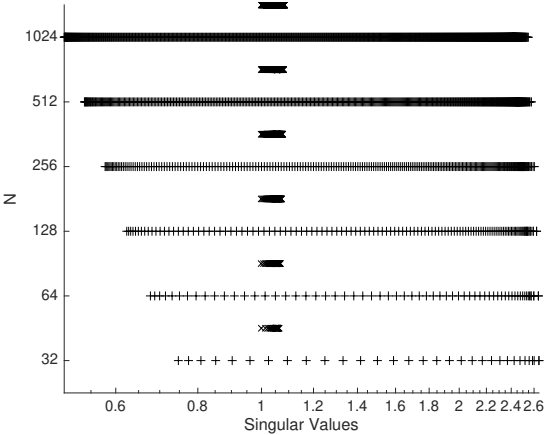
$\alpha = 1.1,\ k = 2$



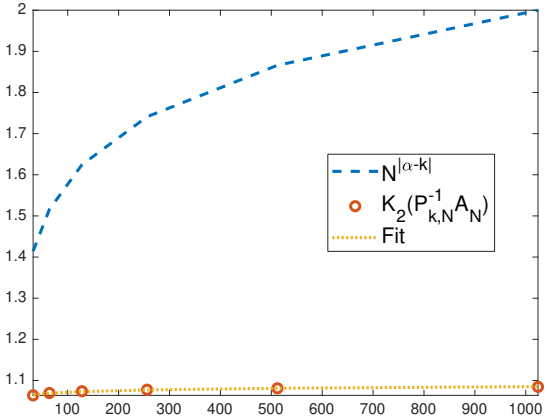
$\alpha = 1.1,\ k = 2$

Preconditioning GLT with GLT

Let's numerically test our idea.



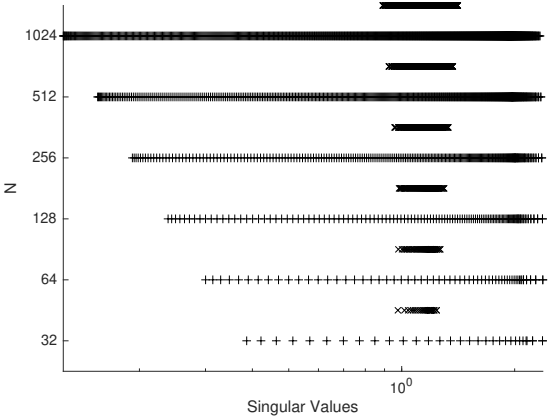
$\alpha = 1.1,\ k = 1$



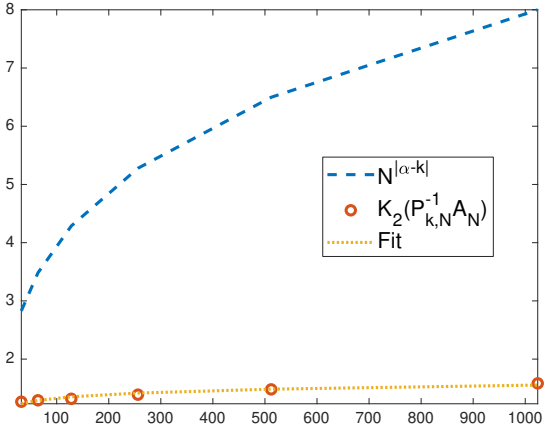
$\alpha = 1.1,\ k = 1$

Preconditioning GLT with GLT

Let's numerically test our idea.



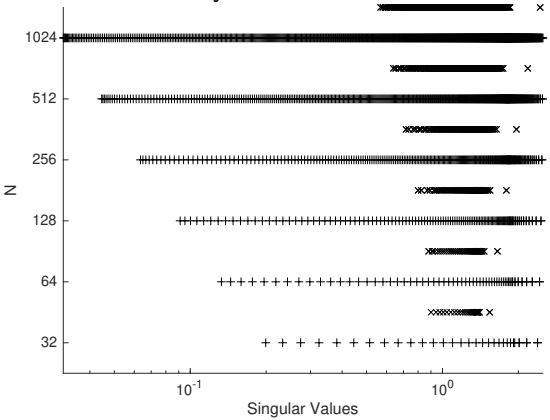
$\alpha = 1.3,\ k = 1$



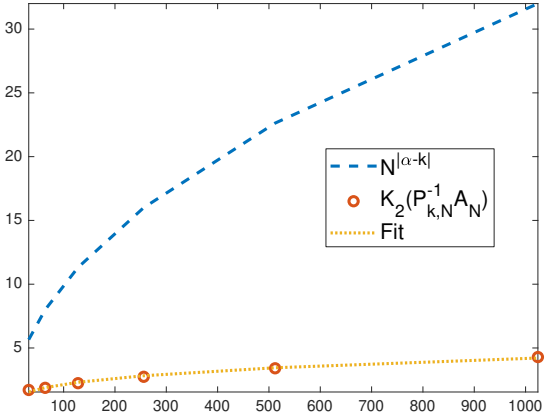
$\alpha = 1.3,\ k = 1$

Preconditioning GLT with GLT

Let's numerically test our idea.



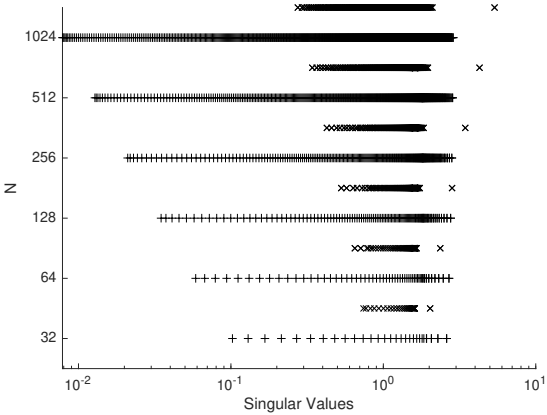
$\alpha = 1.5,\ k = 1$



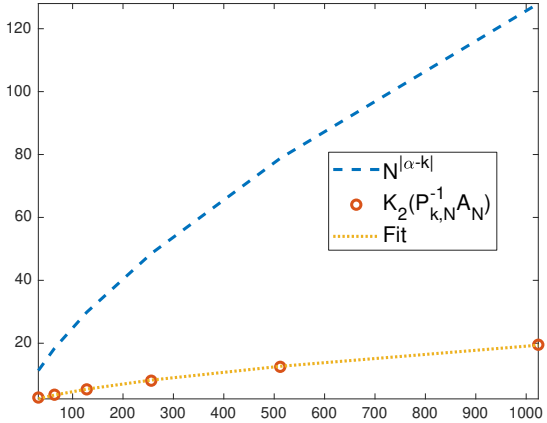
$\alpha = 1.5,\ k = 1$

Preconditioning GLT with GLT

Let's numerically test our idea.



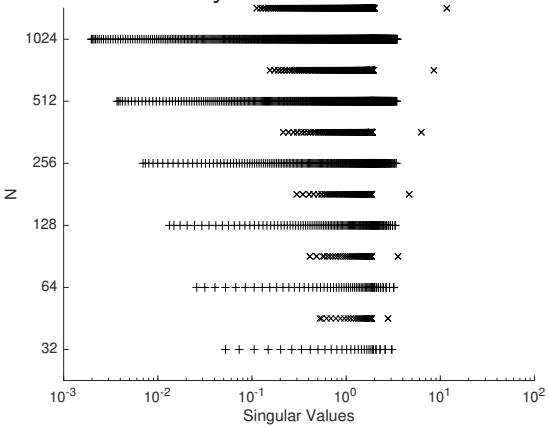
$\alpha = 1.7,\ k = 1$



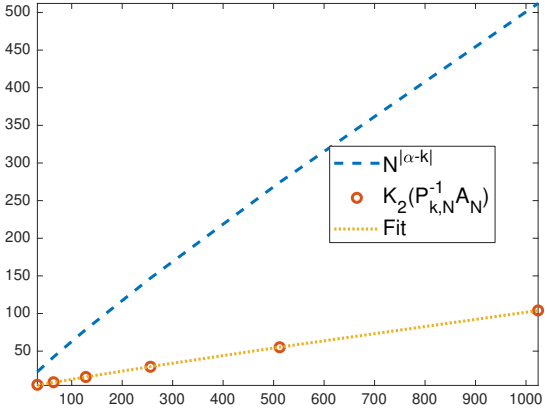
$\alpha = 1.7,\ k = 1$

Preconditioning GLT with GLT

Let's numerically test our idea.



$\alpha = 1.3,\ k = 1$



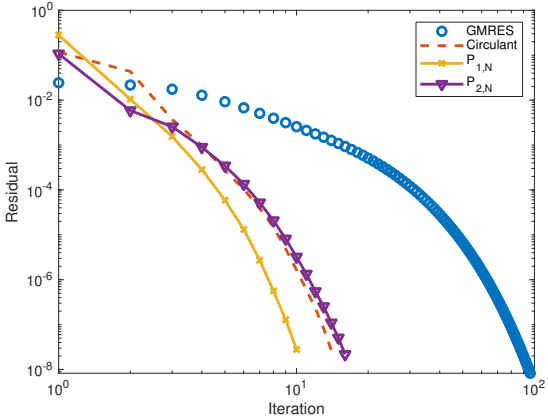
$\alpha = 1.3,\ k = 1$

Preconditioning GLT with GLT

Test case is

$$d^+(x,t) = \Gamma(3-\alpha)x^\alpha, \qquad d^-(x,t) = \Gamma(3-\alpha)(2-x)^\alpha$$

α	N	GMRES	P	$P_{1,N}$	$P_{2,N}$
1.2	2^5	31	13	10	13
	2^6	50	14	11	15
	2^7	64	14	11	16
	2^8	75	15	11	16
	2^9	84	15	11	16
	2^{10}	91	14	10	16
	2^{11}	96	14	10	16

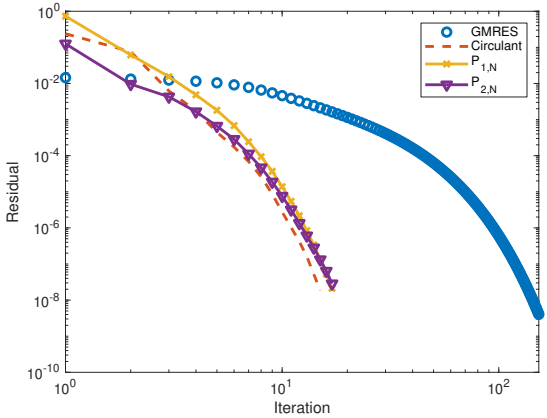


Preconditioning GLT with GLT

Test case is

$$d^+(x,t) = \Gamma(3-\alpha)x^\alpha, \qquad d^-(x,t) = \Gamma(3-\alpha)(2-x)^\alpha$$

α	N	GMRES	P	$P_{1,N}$	$P_{2,N}$
1.3	2^5	31	13	13	14
	2^6	55	14	15	15
	2^7	79	15	16	16
	2^8	100	15	16	17
	2^9	119	15	16	17
	2^{10}	136	15	17	17
	2^{11}	153	15	17	17

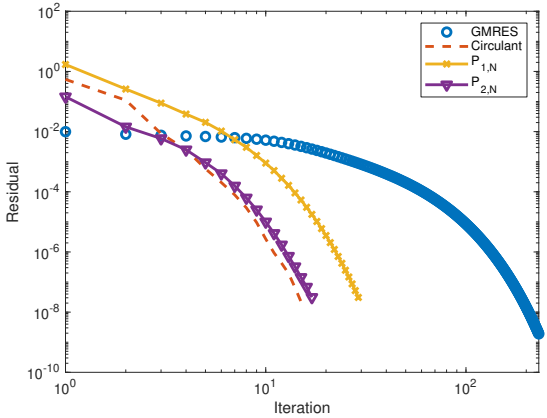


Preconditioning GLT with GLT

Test case is

$$d^+(x,t) = \Gamma(3-\alpha)x^\alpha, \qquad d^-(x,t) = \Gamma(3-\alpha)(2-x)^\alpha$$

α	N	GMRES	P	$P_{1,N}$	$P_{2,N}$
1.4	2^5	31	13	16	13
	2^6	59	14	20	15
	2^7	92	15	23	16
	2^8	127	15	25	16
	2^9	161	15	26	17
	2^{10}	196	15	28	17
	2^{11}	231	15	29	17

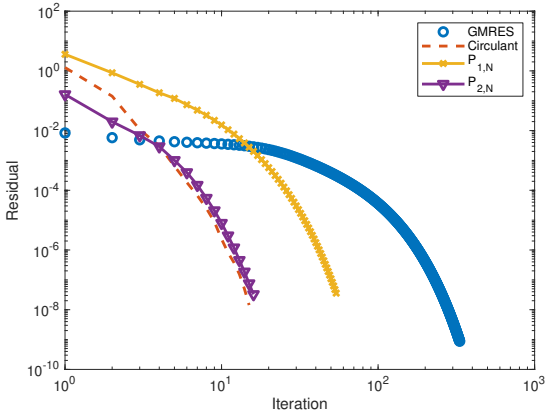


Preconditioning GLT with GLT

Test case is

$$d^+(x,t) = \Gamma(3-\alpha)x^\alpha, \qquad d^-(x,t) = \Gamma(3-\alpha)(2-x)^\alpha$$

α	N	GMRES	P	$P_{1,N}$	$P_{2,N}$
1.5	2^5	32	13	19	12
	2^6	61	14	25	14
	2^7	104	15	32	15
	2^8	155	15	38	15
	2^9	209	15	43	16
	2^{10}	268	15	49	16
	2^{11}	332	15	54	16

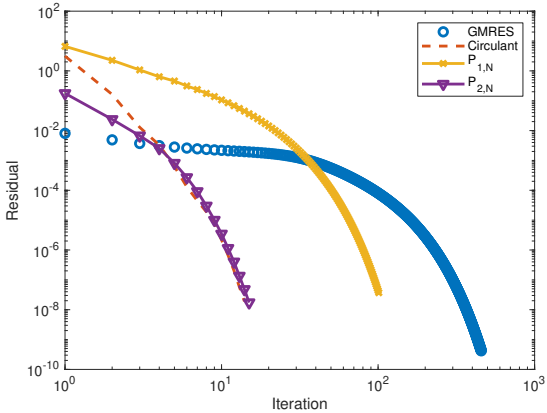


Preconditioning GLT with GLT

Test case is

$$d^+(x,t) = \Gamma(3-\alpha)x^\alpha, \qquad d^-(x,t) = \Gamma(3-\alpha)(2-x)^\alpha$$

α	N	GMRES	P	$P_{1,N}$	$P_{2,N}$
1.6	2^5	32	13	22	11
	2^6	62	13	31	12
	2^7	112	14	42	13
	2^8	183	14	55	14
	2^9	262	14	69	14
	2^{10}	353	14	84	15
	2^{11}	456	14	101	15

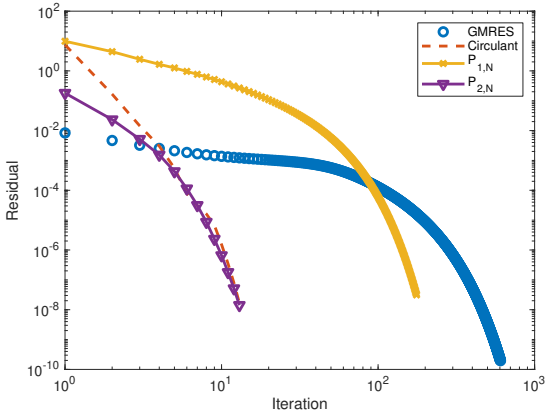


Preconditioning GLT with GLT

Test case is

$$d^+(x,t) = \Gamma(3-\alpha)x^\alpha, \qquad d^-(x,t) = \Gamma(3-\alpha)(2-x)^\alpha$$

α	N	GMRES	P	$P_{1,N}$	$P_{2,N}$
1.7	2^5	32	12	25	10
	2^6	64	13	38	11
	2^7	118	13	55	12
	2^8	207	13	77	12
	2^9	319	13	104	12
	2^{10}	449	13	136	13
	2^{11}	605	13	176	13

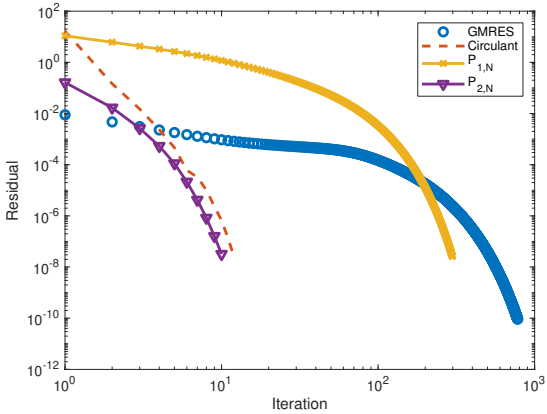


Preconditioning GLT with GLT

Test case is

$$d^+(x,t) = \Gamma(3-\alpha)x^\alpha, \qquad d^-(x,t) = \Gamma(3-\alpha)(2-x)^\alpha$$

α	N	GMRES	P	$P_{1,N}$	$P_{2,N}$
1.8	2^5	32	12	27	9
	2^6	64	12	44	9
	2^7	126	13	71	10
	2^8	225	13	108	10
	2^9	378	13	157	10
	2^{10}	559	12	219	10
	2^{11}	779	12	298	10

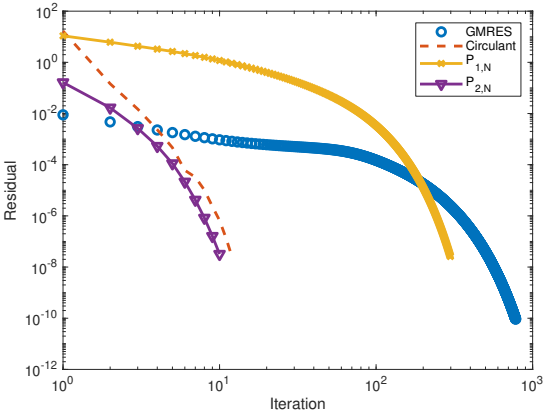


Preconditioning GLT with GLT

Test case is

$$d^+(x,t) = \Gamma(3-\alpha)x^\alpha, \qquad d^-(x,t) = \Gamma(3-\alpha)(2-x)^\alpha$$

α	N	GMRES	P	$P_{1,N}$	$P_{2,N}$
1.8	2^5	32	12	27	9
	2^6	64	12	44	9
	2^7	126	13	71	10
	2^8	225	13	108	10
	2^9	378	13	157	10
	2^{10}	559	12	219	10
	2^{11}	779	12	298	10



To **do better** we need to move towards Multigrid methods.

Circulant matrices at any cost

Despite the **clear negative results** concerning the impossibility of obtaining a cluster using circulant matrices in the space-dependent case, the *literature contains several attempts* in this direction.

Circulant matrices at any cost

Despite the **clear negative results** concerning the impossibility of obtaining a cluster using circulant matrices in the space-dependent case, the *literature contains several attempts* in this direction.

One of the most reused idea originates from (Pan et al. 2014), and goes as follows

1. We want to solve a "diagonal times Toeplitz" linear system, i.e.,

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$

Circulant matrices at any cost

Despite the **clear negative results** concerning the impossibility of obtaining a cluster using circulant matrices in the space-dependent case, the *literature contains several attempts* in this direction.

One of the most reused idea originates from (Pan et al. 2014), and goes as follows

1. We want to solve a "diagonal times Toeplitz" linear system, i.e.,

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$

2. Call $d_i^+ = d^+(x_i)$ and $d_i^- = d(x_i)$, $i = 1, 2, \ldots, N$,

Circulant matrices at any cost

Despite the **clear negative results** concerning the impossibility of obtaining a cluster using circulant matrices in the space-dependent case, the *literature contains several attempts* in this direction.

One of the most reused idea originates from (Pan et al. 2014), and goes as follows

1. We want to solve a "diagonal times Toeplitz" linear system, i.e.,

$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$

2. Call $d_i^+ = d^+(x_i)$ and $d_i^- = d(x_i)$, $i = 1, 2, \ldots, N$,
3. Define the Toeplitz matrices

$$K_i = \nu I_n - \left(d_i^+ G_N + d_i^- G_N^T \right), \qquad i = 1, 2, \ldots, N.$$

Circulant matrices at any cost

Despite the **clear negative results** concerning the impossibility of obtaining a cluster using circulant matrices in the space-dependent case, the *literature contains several attempts* in this direction.

One of the most reused idea originates from (Pan et al. 2014), and goes as follows

1. We want to solve a "diagonal times Toeplitz" linear system, i.e.,
$$A_N = \nu I_N - \left(D_N^+ G_N + D_N^- G_N^T \right),$$
2. Call $d_i^+ = d^+(x_i)$ and $d_i^- = d(x_i)$, $i = 1, 2, \ldots, N$,
3. Define the Toeplitz matrices
$$K_i = \nu I_n - \left(d_i^+ G_N + d_i^- G_N^T \right), \qquad i = 1, 2, \ldots, N.$$
4. Since $\mathbf{e}_i^T A_N = e_i^T K_i$, approximate
$$\mathbf{e}_i^T A^{-1} \approx \mathbf{e}_i^T K_i^{-1}.$$

Circulant matrices at any cost

But how do we approximate the inversion?

Circulant matrices at any cost

But how do we approximate the inversion?

- Build $P_1 = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^T K_i^{-1}$

Circulant matrices at any cost

But how do we approximate the inversion?

- Build $P_1 = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^T K_i^{-1}$
  - it costs too much! N Toeplitz solve per iteration.

Circulant matrices at any cost

But how do we approximate the inversion?

- Build $P_1 = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^T K_i^{-1}$

  it costs too much! N Toeplitz solve per iteration.

- Build $P_2 = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^T C_i^{-1}$ with $C_i = s(K_i)$ (Strang preconditioner)

Circulant matrices at any cost

But how do we approximate the inversion?

- Build $P_1 = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^T K_i^{-1}$

  it costs too much! N Toeplitz solve per iteration.

- Build $P_2 = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^T C_i^{-1}$ with $C_i = s(K_i)$ (Strang preconditioner)

  it still cost too much! $O(N^2 \log(N))$ per iteration.

Circulant matrices at any cost

But how do we approximate the inversion?

- Build $P_1 = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^T K_i^{-1}$

  it costs too much! N Toeplitz solve per iteration.

- Build $P_2 = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^T C_i^{-1}$ with $C_i = s(K_i)$ (Strang preconditioner)

  it still cost too much! $O(N^2 \log(N))$ per iteration.

- Build $P_3 = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^T \sum_{j=1}^{\ell} \phi_j(x_i) C_j^{-1}$

  where for $\ell \ll N$ values $\{x_{i_j}\}_{j=1}^{\ell} \subset \{x_i\}_{i=1}^{N}$ $\phi_j(x)$ are the basis of the piecewise linear interpolation of

$$q_\lambda(x) = \frac{1}{\nu + \lambda d^+(x) + \bar{\lambda} d^-(x)}, \qquad \lambda \in \mathbb{C}.$$

Circulant matrices at any cost

But how do we approximate the inversion?

- Build $P_1 = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^T K_i^{-1}$

  it costs too much! N Toeplitz solve per iteration.

- Build $P_2 = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^T C_i^{-1}$ with $C_i = s(K_i)$ (Strang preconditioner)

  it still cost too much! $O(N^2 \log(N))$ per iteration.

- Build $P_3 = \sum_{i=1}^{N} \mathbf{e}_i \mathbf{e}_i^T \sum_{j=1}^{\ell} \phi_j(x_i) C_j^{-1}$ The cost is now $O(\ell N \log N)$ operations.

  where for $\ell \ll N$ values $\{x_{i_j}\}_{j=1}^{\ell} \subset \{x_i\}_{i=1}^{N}$ $\phi_j(x)$ are the basis of the piecewise linear interpolation of

$$q_\lambda(x) = \frac{1}{\nu + \lambda d^+(x) + \bar{\lambda} d^-(x)}, \qquad \lambda \in \mathbb{C}.$$

Circulant matrices at any cost

The analysis of the 😐 P_3 preconditioner is quite involved, furthermore

- the iteration number dependence on the selection of the interpolation nodes and the value of λ is unclear,
- the **resulting preconditioner** is **always a circulant matrix**, thus the general theory tells us that there is no hope of getting a cluster of any sort.

Circulant matrices at any cost

The analysis of the 😐 P_3 preconditioner is quite involved, furthermore

- ⚙ the iteration number dependence on the selection of the interpolation nodes and the value of λ is unclear,
- ⚙ the **resulting preconditioner** is **always a circulant matrix**, thus the general theory tells us that there is no hope of getting a cluster of any sort.

⚠ The extension of this preconditioners to the multi-dimensional settings is even more challenging: interpolation of surfaces, and higher dimensional objects is a tough problem!

Circulant matrices at any cost

The analysis of the 😐 P_3 preconditioner is quite involved, furthermore

- the iteration number dependence on the selection of the interpolation nodes and the value of λ is unclear,
- the **resulting preconditioner** is **always a circulant matrix**, thus the general theory tells us that there is no hope of getting a cluster of any sort.

⚠ The extension of this preconditioners to the multi-dimensional settings is even more challenging: interpolation of surfaces, and higher dimensional objects is a tough problem!

✘ For these reasons we will not pursue further these results, if you are interested start from (Pan et al. 2014), and look to the next episodes.

Multidimensional cases

What happens if our equation becomes

$$\begin{cases} \frac{\partial W}{\partial t} = \left(\theta\, {}^{RL}D^{\alpha}_{[0,x]} \cdot + (1-\theta)\, {}^{RL}D^{\alpha}_{[x,1]} \cdot\right) W(x,y,t) + \qquad \theta \in [0,1], \\ \qquad \left(\theta\, {}^{RL}D^{\alpha}_{[0,y]} \cdot + (1-\theta)\, {}^{RL}D^{\alpha}_{[y,1]} \cdot\right) W(x,y,t) \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

- If we repeat the discretization procedure we have used in the 1D case we end up with a **block-Toeplitz-with-Toeplitz-blocks** matrix,
- then we could attempt solution by using a block-circulant-with-circulant-blocks preconditioner! In the 1D case (either symmetric or not) the procedure was working, maybe we are lucky...

Multidimensional cases

What happens if our equation becomes

$$\begin{cases} \frac{\partial W}{\partial t} = \left(d_x^+(x,t)\,{}^{RL}D^{\alpha}_{[0,x]}\cdot + 1-\theta)d_x^-(x,t){}^{RL}D^{\alpha}_{[x,1]}\cdot\right) W(x,y,t) +, \\ \qquad\quad \left(d_y^+(x,y,t)\,{}^{RL}D^{\alpha}_{[0,y]}\cdot + 1-\theta)d_y^-(x,y,t){}^{RL}D^{\alpha}_{[y,1]}\cdot\right) W(x,y,t) \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

- It should not be difficult to imagine, but in this case we should end up again with a **matrix sequence of GLT type**,
- we can attempt the solution by doing something similar to what we have done in the 1D case: using a Toeplitz preconditioner...

A negative result

In the constant coefficient case we have a **general negative result**:

> *"Any Circulant-Like Preconditioner for Multilevel Matrices Is Not Superlinear" – Serra Capizzano and Tyrtyshnikov 1999*

Theorem (Serra Capizzano and Tyrtyshnikov 1999, Theorem 4.1)

For $I_n + A_n$, $A_n = A_n(f)$ a p-level Toeplitz matrix, any preconditioner for the form $I_n + C_n$, where p_n is a p-level circulant matrix, is not superlinear.

A negative result

In the constant coefficient case we have a **general negative result**:

> *"Any Circulant-Like Preconditioner for Multilevel Matrices Is Not Superlinear" – Serra Capizzano and Tyrtyshnikov 1999*

Theorem (Serra Capizzano and Tyrtyshnikov 1999, Theorem 4.1)

For $I_n + A_n$, $A_n = A_n(f)$ a p-level Toeplitz matrix, any preconditioner for the form $I_n + C_n$, where p_n is a p-level circulant matrix, is not superlinear.

- The **number of iterations** for the preconditioned system **will always depend on the size of the system**!

A negative result

In the constant coefficient case we have a **general negative result**:

> *"Any Circulant-Like Preconditioner for Multilevel Matrices Is Not Superlinear" – Serra Capizzano and Tyrtyshnikov 1999*

Theorem (Serra Capizzano and Tyrtyshnikov 1999, Theorem 4.1)

For $I_n + A_n$, $A_n = A_n(f)$ a p-level Toeplitz matrix, any preconditioner for the form $I_n + C_n$, where p_n is a p-level circulant matrix, is not superlinear.

- The **number of iterations** for the preconditioned system **will always depend on the size of the system**!
- The dependence can still by milder than the one of the original system, thus there are cases in which this could be worthwhile (at least for a while).

A negative result

In the constant coefficient case we have a **general negative result**:

> *"Any Circulant-Like Preconditioner for Multilevel Matrices Is Not Superlinear" – Serra Capizzano and Tyrtyshnikov 1999*

Theorem (Serra Capizzano and Tyrtyshnikov 1999, Theorem 4.1)

For $I_n + A_n$, $A_n = A_n(f)$ a p-level Toeplitz matrix, any preconditioner for the form $I_n + C_n$, where p_n is a p-level circulant matrix, is not superlinear.

- The **number of iterations** for the preconditioned system **will always depend on the size of the system**!
- The dependence can still by milder than the one of the original system, thus there are cases in which this could be worthwhile (at least for a while).

It is a difficult world

Already the case with constant coefficient is difficult to treat. Maybe we can find a way to *reduce the number of dimensions*.

Another negative result and a proposal

- The result we have obtained by means of GLT theory for the variable coefficient case remains valid also in two dimensions: *no circulant preconditioner can have a strong cluster*!

Another negative result and a proposal

- The result we have obtained by means of GLT theory for the variable coefficient case remains valid also in two dimensions: *no circulant preconditioner can have a strong cluster*!
- We could attempt generalizing the $P_{1,N}$ and $P_{2,N}$ preconditioners to the new setting.

Another negative result and a proposal

- The result we have obtained by means of GLT theory for the variable coefficient case remains valid also in two dimensions: *no circulant preconditioner can have a strong cluster*!
- We could attempt generalizing the $P_{1,N}$ and $P_{2,N}$ preconditioners to the new setting.
- The matrix of the system in 2D has now the form

$$A_{\mathbf{N}} = \nu I_{\mathbf{N}} - \left(D^{+}_{\mathbf{N}}(G_{N_x} \otimes I_{N_y}) + D^{-}_{\mathbf{N}}(I_{N_x} \otimes G_{N_y})\right), \qquad \mathbf{N} = (N_x, N_y).$$

  - If the **diffusion coefficients** are **constants**, this a BTTB matrix,
  - If the **diffusion coefficients** are **space variant**, we can show (following the same road as before) that the resulting matrix sequence is a GLT sequence.

Another negative result and a proposal

- The result we have obtained by means of GLT theory for the variable coefficient case remains valid also in two dimensions: *no circulant preconditioner can have a strong cluster*!
- We could attempt generalizing the $P_{1,N}$ and $P_{2,N}$ preconditioners to the new setting.
- The matrix of the system in 2D has now the form

$$A_{\mathbf{N}} = \nu I_{\mathbf{N}} - \left(D^{+}_{\mathbf{N}}(G_{N_x} \otimes I_{N_y}) + D^{-}_{\mathbf{N}}(I_{N_x} \otimes G_{N_y})\right), \qquad \mathbf{N} = (N_x, N_y).$$

  - If the **diffusion coefficients** are **constants**, this a BTTB matrix,
  - If the **diffusion coefficients** are **space variant**, we can show (following the same road as before) that the resulting matrix sequence is a GLT sequence.
- $P_{1,\mathbf{N}} = \nu I_{\mathbf{N}} - \left(D^{+}_{\mathbf{N}}(T_{N_x}(1 - e^{-i\theta_1}) \otimes I_{N_y}) + D^{-}_{\mathbf{N}}(I_{N_x} \otimes T_{N_y}(1 - e^{-i\theta_2}))\right)$;
- $P_{2,\mathbf{N}} = \nu I_{\mathbf{N}} - \left(D^{+}_{\mathbf{N}}(T_{N_x}(2 - 2\cos(\theta_1)) \otimes I_{N_y}) + D^{-}_{\mathbf{N}}(I_{N_x} \otimes T_{N_y}(2 - 2\cos(\theta_2)))\right)$.

The structure preserving preconditioners

❗ To apply both $P_{1,\mathbf{N}}$ and $P_{2,\mathbf{N}}$ we now need to **solve an auxiliary sparse linear system** related to the **discretization of a 2D problem**.

- Using a sparse direct solver is not going to scale well with $\mathbf{N} = (N_x, N_y)$,

The structure preserving preconditioners

❗ To apply both $P_{1,\mathbf{N}}$ and $P_{2,\mathbf{N}}$ we now need to **solve an auxiliary sparse linear system** related to the **discretization of a 2D problem**.

- Using a sparse direct solver is not going to scale well with $\mathbf{N} = (N_x, N_y)$,
- We need to **employ an iterative technique** to do the **preconditioner application**!

The structure preserving preconditioners

❗ To apply both $P_{1,\mathbf{N}}$ and $P_{2,\mathbf{N}}$ we now need to **solve an auxiliary sparse linear system** related to the **discretization of a 2D problem**.

- Using a sparse direct solver is not going to scale well with $\mathbf{N} = (N_x, N_y)$,
- We need to **employ an iterative technique** to do the **preconditioner application**!
- Methods of this type are usually called **multi-iterative methods**
  - $\Rightarrow$ If we apply $P_{1,\mathbf{N}}$ or $P_{2,\mathbf{N}}$ using a fixed number of iterations of a fixed point technique, then we can still use GMRES,
  - $\Rightarrow$ If we apply $P_{1,\mathbf{N}}$ or $P_{2,\mathbf{N}}$ using a variable number of iterations of a fixed point technique or a *nonstationary solver*, then we have to use the Flexible-GMRES.

The structure preserving preconditioners

To apply both $P_{1,\mathbf{N}}$ and $P_{2,\mathbf{N}}$ we now need to **solve an auxiliary sparse linear system** related to the **discretization of a 2D problem**.

- Using a sparse direct solver is not going to scale well with $\mathbf{N} = (N_x, N_y)$,
- We need to **employ an iterative technique** to do the **preconditioner application**!
- Methods of this type are usually called **multi-iterative methods**
  - $\Rightarrow$ If we apply $P_{1,\mathbf{N}}$ or $P_{2,\mathbf{N}}$ using a fixed number of iterations of a fixed point technique, then we can still use GMRES,
  - $\Rightarrow$ If we apply $P_{1,\mathbf{N}}$ or $P_{2,\mathbf{N}}$ using a variable number of iterations of a fixed point technique or a *nonstationary solver*, then we have to use the Flexible-GMRES.

What is the right combination?

The right combination of iterative schemes to use does really depend on the machine we have under our hands!

Flexible-GMRES (Saad 1993)

The **Flexible variant of GMRES** is built from the *right-preconditioned* GMRES algorithm.

Input: $A \in \mathbb{R}^{n\times n}$, m, $\mathbf{x}^{(0)}$, $M \in \mathbb{R}^{n\times n}$

1 $\mathbf{r}^{(0)} \leftarrow b - A\mathbf{x}^{(0)}$; `/* Arnoldi process */`

2 $\beta \leftarrow \|\mathbf{r}^{(0)}\|_2$, $\mathbf{v}^{(1)} \leftarrow \mathbf{r}^{(0)}/\beta$;

3 **for** $j = 1, \ldots, m$ **do**

4 $\quad \mathbf{z}^{(j)} \leftarrow P^{-1}\mathbf{v}^{(j)}$;

5 $\quad \mathbf{w} \leftarrow A\mathbf{z}^{(j)}$;

6 $\quad$ **for** $i = 1, \ldots, j$ **do**

7 $\quad\quad h_{i,j} \leftarrow < \mathbf{w}, \mathbf{v}^{(i)} >$;

8 $\quad\quad \mathbf{w} \leftarrow \mathbf{w} - h_{i,j}\mathbf{v}^{(i)}$;

9 $\quad$ **end**

10 $\quad h_{j+1,j} \leftarrow \|\mathbf{w}\|_2$;

11 $\quad \mathbf{v}^{(j+1)} \leftarrow \mathbf{w}/h_{j+1,j}$;

12 **end**

13 $V_m \leftarrow [\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(m)}]$;

`// Build the Krylov subspace basis`

14 $\mathbf{y}^{(m)} \leftarrow \arg\min_{\mathbf{y}} \|\beta\mathbf{e}_1 - \overline{H}_m\mathbf{y}\|_2$;

15 $\mathbf{x}^{(m)} \leftarrow \mathbf{x}^{(0)} + P^{-1}V_m\mathbf{y}^{(m)}$;

`// Conv. check, possibly a restart`

16 **if** *Stopping criteria satisfied* **then**

17 $\quad$ **Return:** $\tilde{\mathbf{x}} = \mathbf{x}^{(m)}$;

18 **else**

19 $\quad \mathbf{x}^{(0)} \leftarrow \mathbf{x}^{(m)}$; `/* Restart */`

20 $\quad$ **goto** 1;

21 **end**

Same preconditioner

Line 15 forms the approximate solution of the linear system as $\mathbf{x}^{(0)} + P^{-1}V_m\mathbf{y}^{(m)}$.

Flexible-GMRES (Saad 1993)

The **Flexible variant of GMRES** is built from the *right-preconditioned* GMRES algorithm.

Input: $A \in \mathbb{R}^{n\times n}$, m, $\mathbf{x}^{(0)}$, $M \in \mathbb{R}^{n\times n}$

1 $\mathbf{r}^{(0)} \leftarrow b - A\mathbf{x}^{(0)}$; `/* Arnoldi process */`

2 $\beta \leftarrow \|\mathbf{r}^{(0)}\|_2$, $\mathbf{v}^{(1)} \leftarrow \mathbf{r}^{(0)}/\beta$;

3 **for** $j = 1, \ldots, m$ **do**

4 $\quad \mathbf{z}^{(j)} \leftarrow P^{-1}\mathbf{v}^{(j)}$;

5 $\quad \mathbf{w} \leftarrow A\mathbf{z}^{(j)}$;

6 $\quad$ **for** $i = 1, \ldots, j$ **do**

7 $\quad\quad h_{i,j} \leftarrow < \mathbf{w}, \mathbf{v}^{(i)} >$;

8 $\quad\quad \mathbf{w} \leftarrow \mathbf{w} - h_{i,j}\mathbf{v}^{(i)}$;

9 $\quad$ **end**

10 $\quad h_{j+1,j} \leftarrow \|\mathbf{w}\|_2$;

11 $\quad \mathbf{v}^{(j+1)} \leftarrow \mathbf{w}/h_{j+1,j}$;

12 **end**

13 $Z_m \leftarrow [\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}]$;

`// Build the Krylov subspace basis`

14 $\mathbf{y}^{(m)} \leftarrow \arg\min_{\mathbf{y}} \|\beta\mathbf{e}_1 - \overline{H}_m\mathbf{y}\|_2$;

15 $\mathbf{x}^{(m)} \leftarrow \mathbf{x}^{(0)} + Z_m\mathbf{y}^{(m)}$;

`// Conv. check, possibly a restart`

16 **if** *Stopping criteria satisfied* **then**

17 $\quad$ **Return:** $\tilde{\mathbf{x}} = \mathbf{x}^{(m)}$;

18 **else**

19 $\quad \mathbf{x}^{(0)} \leftarrow \mathbf{x}^{(m)}$; `/* Restart */`

20 $\quad$ **goto** 1;

21 **end**

Changing preconditioner

Line 15 forms the approximate solution of the linear system as $\mathbf{x}^{(0)} + Z_m\mathbf{y}^{(m)}$.

Flexible-GMRES (Saad 1993)

With this variant of the GMRES we are solving

$$AP^{-1}\mathbf{y} = \mathbf{b}, \text{ with } P\mathbf{x} = \mathbf{y},$$

with a preconditioner P whose action depends on the vector to which it is applied,

- in **terms of memory** we have to store two basis instead of one,
- ↓ we use the **true residual** instead of the preconditioned one: *the results are more reliable!*

Flexible-GMRES (Saad 1993)

With this variant of the GMRES we are solving

$$AP^{-1}\mathbf{y} = \mathbf{b}, \text{ with } P\mathbf{x} = \mathbf{y},$$

with a preconditioner P whose action depends on the vector to which it is applied,

- in **terms of memory** we have to store two basis instead of one,
- we use the **true residual** instead of the preconditioned one: *the results are more reliable!*

Some **usual choices** of multi-iterative schemes are

- Inner/Outer GMRES method: we fix a preconditioner P, solve the systems

$$\mathbf{z}^{(j)} \leftarrow P^{-1}\mathbf{v}^{(j)},$$

  by a recursive call to GMRES;
- A Multigrid algorithm in which some *smoother* or *coarse solver* is non stationary;
- Non stationary polynomial preconditioners.

Exploiting the Kronecker structure

The **multidimensional case** has a **new structure** we can exploit: **Kronecker sums**!

$$A_{\mathbf{N}} = \nu I_{\mathbf{N}} - \left(D^{+}_{\mathbf{N}} (G_{N_x} \otimes I_{N_y}) + D^{-}_{\mathbf{N}} (I_{N_x} \otimes G_{N_y}) \right), \qquad \mathbf{N} = (N_x, N_y).$$

Exploiting the Kronecker structure

The **multidimensional case** has a **new structure** we can exploit: **Kronecker sums**!

$$A_{\mathbf{N}} = \nu I_{N_x} \otimes I_{N_y} - \left(D_{\mathbf{N}}^{+}(G_{N_x} \otimes I_{N_y}) + D_{\mathbf{N}}^{-}(I_{N_x} \otimes G_{N_y}) \right), \qquad \mathbf{N} = (N_x, N_y).$$

Exploiting the Kronecker structure

The **multidimensional case** has a **new structure** we can exploit: **Kronecker sums**!

$$A_{\mathbf{N}} = \nu I_{N_x} \otimes I_{N_y} - \left(D^{+}_{\mathbf{N}}(G_{N_x} \otimes I_{N_y}) + D^{-}_{\mathbf{N}}(I_{N_x} \otimes G_{N_y})\right), \qquad \mathbf{N} = (N_x, N_y).$$

If we assume **separable coefficients**, i.e.,

$$d^{+}(x,y) = d^{+}_1(x) d^{+}_2(y), \quad d^{-}(x,y) = d^{-}_1(x) d^{-}_2(y).$$

Exploiting the Kronecker structure

The **multidimensional case** has a **new structure** we can exploit: **Kronecker sums**!

$$A_{\mathbf{N}} = \nu I_{N_x} \otimes I_{N_y} - \Big((D^{+}_{\mathbf{1},\mathbf{N_x}} \otimes D^{+}_{2,N_y})(G_{N_x} \otimes I_{N_y}) + (D^{-}_{\mathbf{1},\mathbf{N_x}} \otimes D^{-}_{2,N_y})(I_{N_x} \otimes G_{N_y}) \Big)$$

- If we assume **separable coefficients**, i.e.,

$$d^{+}(x,y) = d_1^{+}(x) d_2^{+}(y), \quad d^{-}(x,y) = d_1^{-}(x) d_2^{-}(y).$$

- We write the solution vector $\mathbf{x}$ as a matrix X such that $\mathbf{x} = \mathrm{vec}(X)$, where $\mathrm{vec}(\cdot)$ is the operation that stacks the columns of X, and the right-hand side $\mathbf{b}$ as B with $\mathbf{b} = \mathrm{vec}(B)$.

Exploiting the Kronecker structure

The **multidimensional case** has a **new structure** we can exploit: **Kronecker sums**!

$$\text{Find } X \text{ s.t. } \nu X - D^{+}_{2,N_y} X G^{T}_{N_x} D^{+}_{1,N_x} - D^{-}_{2,N_y} G_{N_y} X D^{-}_{1,N_x} = B$$

- If we assume **separable coefficients**, i.e.,

$$d^{+}(x,y) = d^{+}_1(x) d^{+}_2(y), \quad d^{-}(x,y) = d^{-}_1(x) d^{-}_2(y).$$

- We write the solution vector $\mathbf{x}$ as a matrix X such that $\mathbf{x} = \mathrm{vec}(X)$, where $\mathrm{vec}(\cdot)$ is the operation that stacks the columns of X, and the right-hand side $\mathbf{b}$ as B with $\mathbf{b} = \mathrm{vec}(B)$.
- We got ourselves a **matrix equation** involving objects of "smaller size".

Conclusion and summary

- We have characterized the **spectral properties** of the involved matrix sequences,
- We investigated several preconditioning strategies that made use of the **structure** of the **underlying matrices**,
- We started investigating multi-iterative schemes and looking for ways of reducing the dimensionality of the involved problems.

Next up

- How and when do we solve the **matrix equation** formulation,
- What do we do when we have more than two dimensions?
- All-at-once formulations.

Bibliography I

- Chan, R. H. and K.-P. Ng (1993). "Toeplitz preconditioners for Hermitian Toeplitz systems". In: *Linear Algebra Appl.* 190, pp. 181–208. ISSN: 0024-3795. DOI: 10.1016/0024-3795(93)90226-E. URL: https://doi.org/10.1016/0024-3795(93)90226-E.
- Donatelli, M., M. Mazza, and S. Serra-Capizzano (2016). "Spectral analysis and structure preserving preconditioners for fractional diffusion equations". In: *J. Comput. Phys.* 307, pp. 262–279. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2015.11.061. URL: https://doi.org/10.1016/j.jcp.2015.11.061.
- Garoni, C. and S. Serra-Capizzano (2017). *Generalized locally Toeplitz sequences: theory and applications. Vol. I.* Springer, Cham, pp. xi+312. ISBN: 978-3-319-53678-1; 978-3-319-53679-8. DOI: 10.1007/978-3-319-53679-8. URL: https://doi.org/10.1007/978-3-319-53679-8.
- — (2018). *Generalized locally Toeplitz sequences: theory and applications. Vol. II.* Springer, Cham, pp. xi+194. ISBN: 978-3-030-02232-7; 978-3-030-02233-4. DOI: 10.1007/978-3-030-02233-4. URL: https://doi.org/10.1007/978-3-030-02233-4.

Bibliography II

- Okoudjou, K. A., L. G. Rogers, and R. S. Strichartz (2010). "Szegö limit theorems on the Sierpiński gasket". In: *J. Fourier Anal. Appl.* 16.3, pp. 434–447. ISSN: 1069-5869. DOI: 10.1007/s00041-009-9102-0. URL: https://doi.org/10.1007/s00041-009-9102-0.
- Pan, J. et al. (2014). "Preconditioning techniques for diagonal-times-Toeplitz matrices in fractional diffusion equations". In: *SIAM J. Sci. Comput.* 36.6, A2698–A2719. ISSN: 1064-8275. DOI: 10.1137/130931795. URL: https://doi.org/10.1137/130931795.
- Saad, Y. (1993). "A flexible inner-outer preconditioned GMRES algorithm". In: *SIAM J. Sci. Comput.* 14.2, pp. 461–469. ISSN: 1064-8275. DOI: 10.1137/0914028. URL: https://doi.org/10.1137/0914028.
- Serra, S. (1995). "New PCG based algorithms for the solution of Hermitian Toeplitz systems". In: *Calcolo* 32.3-4, 153–176 (1997). ISSN: 0008-0624. DOI: 10.1007/BF02575833. URL: https://doi.org/10.1007/BF02575833.

Bibliography III

Serra Capizzano, S. and E. Tyrtyshnikov (1999). "Any circulant-like preconditioner for multilevel matrices is not superlinear". In: *SIAM J. Matrix Anal. Appl.* 21.2, pp. 431–439. ISSN: 0895-4798. DOI: 10.1137/S0895479897331941. URL: https://doi.org/10.1137/S0895479897331941.

Tilli, P. (1998). "Locally Toeplitz sequences: spectral properties and applications". In: *Linear Algebra Appl.* 278.1-3, pp. 91–120. ISSN: 0024-3795. DOI: 10.1016/S0024-3795(97)10079-9. URL: https://doi.org/10.1016/S0024-3795(97)10079-9.

An introduction to fractional calculus

Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

October, 2022

The curse of dimensionality

In the last lesson we saw that:

- Circulant preconditioners **cannot cluster** cases with variable coefficients,

The curse of dimensionality

In the last lesson we saw that:

- Circulant preconditioners **cannot cluster** cases with variable coefficients,
- Multilevel circulant preconditioners **cannot cluster** multilevel Toeplitz systems,

The curse of dimensionality

In the last lesson we saw that:

- Circulant preconditioners **cannot cluster** cases with variable coefficients,
- Multilevel circulant preconditioners **cannot cluster** multilevel Toeplitz systems,
- Preconditioner based on matrix algebras with fast simultaneous diagonalization **cannot cluster** multilevel Toeplitz systems.

The curse of dimensionality

In the last lesson we saw that:

- Circulant preconditioners **cannot cluster** cases with variable coefficients,
- Multilevel circulant preconditioners **cannot cluster** multilevel Toeplitz systems,
- Preconditioner based on matrix algebras with fast simultaneous diagonalization **cannot cluster** multilevel Toeplitz systems.
- Can we **reduce the dimensionality** of the problem to reuse the information and good results we have in the 1D case?

The curse of dimensionality

In the last lesson we saw that:

- Circulant preconditioners **cannot cluster** cases with variable coefficients,
- Multilevel circulant preconditioners **cannot cluster** multilevel Toeplitz systems,
- Preconditioner based on matrix algebras with fast simultaneous diagonalization **cannot cluster** multilevel Toeplitz systems.
- Can we **reduce the dimensionality** of the problem to reuse the information and good results we have in the 1D case?
  - Alternating-direction implicit method,

The curse of dimensionality

In the last lesson we saw that:

- Circulant preconditioners **cannot cluster** cases with variable coefficients,
- Multilevel circulant preconditioners **cannot cluster** multilevel Toeplitz systems,
- Preconditioner based on matrix algebras with fast simultaneous diagonalization **cannot cluster** multilevel Toeplitz systems.
- Can we **reduce the dimensionality** of the problem to reuse the information and good results we have in the 1D case?
  - Alternating-direction implicit method,
  - reformulation as *matrix equations*,

The curse of dimensionality

In the last lesson we saw that:

- Circulant preconditioners **cannot cluster** cases with variable coefficients,
- Multilevel circulant preconditioners **cannot cluster** multilevel Toeplitz systems,
- Preconditioner based on matrix algebras with fast simultaneous diagonalization **cannot cluster** multilevel Toeplitz systems.
- Can we **reduce the dimensionality** of the problem to reuse the information and good results we have in the 1D case?
  - Alternating-direction implicit method,
  - reformulation as *matrix equations*,
  - reformulation as *tensor problems*.

Matrix equation reformulations

The simplest way of introducing this reformulation is to go back to the 1D problem (now with a *source term*):

$$\begin{cases} \frac{\partial W}{\partial t} = \theta\, {}^{RL}D^{\alpha}_{[0,x]} W(x,t) + (1-\theta)\, {}^{RL}D^{\alpha}_{[x,1]} W(x,t) + f(x,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

Matrix equation reformulations

The simplest way of introducing this reformulation is to go back to the 1D problem (now with a *source term*):

$$\begin{cases} \frac{\partial W}{\partial t} = \theta\, {}^{GL}D^{\alpha}_{[0,x]} W(x,t) + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} W(x,t) + f(x,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

To solve everything we have to solve the **sequence of linear systems**

$$\frac{1}{\Delta t}\left(\mathbf{W}^{(m+1)} - \mathbf{W}^{(m)}\right) = \frac{1}{h^{\alpha}}\left(\theta G_N + (1-\theta) G_N^T\right) \mathbf{W}^{(m+1))} + \mathbf{f}^{(m+1)}, \; m = 0, \ldots, M-1.$$

Matrix equation reformulations

The simplest way of introducing this reformulation is to go back to the 1D problem (now with a *source term*):

$$\begin{cases} \frac{\partial W}{\partial t} = \theta\, {}^{GL}D^{\alpha}_{[0,x]} W(x,t) + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} W(x,t) + f(x,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

To solve everything we have to solve the **sequence of linear systems**

$$\frac{1}{\Delta t}\left(\mathbf{W}^{(m+1)} - \mathbf{W}^{(m)}\right) = \frac{1}{h^{\alpha}}\left(\theta G_N + (1-\theta) G_N^T\right) \mathbf{W}^{(m+1))} + \mathbf{f}^{(m+1)}, \; m = 0, \ldots, M-1.$$

💡 Do we really have to solve this sequentially?

Matrix equation reformulations

Following (Breiten, Simoncini, and Stoll 2016), we can **collect the time steps altogether**:

$$\left(B_M \otimes I_N - \frac{\Delta t}{h^\alpha} I_M \otimes T_N\right) \hat{\mathbf{W}} = \mathbf{F},$$

since

$$\begin{bmatrix} I_N - \frac{\Delta t}{h^\alpha} T_N & & & \\ -I_N & I_N - \frac{\Delta t}{h^\alpha} T_N & & \\ & \ddots & \ddots & \\ & & -I_N & I_N - \frac{\Delta t}{h^\alpha} T_N \end{bmatrix} \begin{bmatrix} \mathbf{W}^{(1)} \\ \mathbf{W}^{(2)} \\ \vdots \\ \mathbf{W}^{(M-1)} \end{bmatrix} = \begin{bmatrix} \mathbf{W}^{(0)} + \Delta t \mathbf{f}^{(1)} \\ \Delta t \mathbf{f}^{(2)} \\ \vdots \\ \Delta t \mathbf{f}^{(M)}, \end{bmatrix}$$

for $T_N = \left(\theta G_N + (1-\theta) G_N^T\right)$, $B_M = T_M(1 - e^{i\theta})$.

Matrix equation reformulations

Following (Breiten, Simoncini, and Stoll 2016), we can **collect the time steps altogether**:

$$\left(B_M \otimes I_N - \frac{\Delta t}{h^\alpha} I_M \otimes T_N\right) \hat{\mathbf{W}} = \mathbf{F},$$

since

$$\begin{bmatrix} I_N - \frac{\Delta t}{h^\alpha} T_N & & & \\ -I_N & I_N - \frac{\Delta t}{h^\alpha} T_N & & \\ & \ddots & \ddots & \\ & & -I_N & I_N - \frac{\Delta t}{h^\alpha} T_N \end{bmatrix} \begin{bmatrix} \mathbf{W}^{(1)} \\ \mathbf{W}^{(2)} \\ \vdots \\ \mathbf{W}^{(M-1)} \end{bmatrix} = \begin{bmatrix} \mathbf{W}^{(0)} + \Delta t \mathbf{f}^{(1)} \\ \Delta t \mathbf{f}^{(2)} \\ \vdots \\ \Delta t \mathbf{f}^{(M)}, \end{bmatrix}$$

for $T_N = \left(\theta G_N + (1-\theta) G_N^T\right)$, $B_M = T_M(1 - e^{i\theta})$.

This is now a **coupled system** of size $MN \times MN$, that is larger and uglier than before...

Matrix equation reformulations

❓ Where is the advantage in dealing with

$$\left(B_M \otimes I_N - \frac{\Delta t}{h^\alpha} I_M \otimes T_N \right) \widehat{\mathbf{W}} = \mathbf{F}?$$

Matrix equation reformulations

❓ Where is the advantage in dealing with

$$(B_M \otimes I_N + I_M \otimes A_N)\,\hat{\mathbf{W}} = \mathbf{F}, \qquad A_N = -\frac{\Delta t}{h^\alpha} T_N?$$

Matrix equation reformulations

❓ Where is the advantage in dealing with

$$(B_M \otimes I_N + I_M \otimes A_N)\,\hat{\mathbf{W}} = \mathbf{F}, \qquad A_N = -\frac{\Delta t}{h^\alpha} T_N?$$

Let's call $W = [\mathbf{W}^{(1)}|\cdots|\mathbf{W}^{(M)}]_{N\times M}$, $F = [\mathbf{W}^{(0)} + \Delta t\mathbf{f}^{(1)}|\cdots|\Delta t\mathbf{f}^{M}]_{N\times M}$, and rewrite our problem as

🔨 Compute $W \in \mathbb{R}^{N\times M}$ s.t. $A_N W + W B_M^T = F$.

Matrix equation reformulations

❓ Where is the advantage in dealing with

$$(B_M \otimes I_N + I_M \otimes A_N)\,\hat{\mathbf{W}} = \mathbf{F}, \qquad A_N = -\frac{\Delta t}{h^\alpha} T_N?$$

Let's call $W = [\mathbf{W}^{(1)}|\cdots|\mathbf{W}^{(M)}]_{N\times M}$, $F = [\mathbf{W}^{(0)} + \Delta t\mathbf{f}^{(1)}|\cdots|\Delta t\mathbf{f}^{M}]_{N\times M}$, and rewrite our problem as

🔨 Compute $W \in \mathbb{R}^{N\times M}$ s.t. $A_N W + W B_M^T = F$.

💡 This is a well-know object called **Sylvester equation**!

Matrix equation reformulations

❓ Where is the advantage in dealing with

$$(B_M \otimes I_N + I_M \otimes A_N)\,\hat{\mathbf{W}} = \mathbf{F}, \qquad A_N = -\frac{\Delta t}{h^\alpha} T_N?$$

Let's call $W = [\mathbf{W}^{(1)}|\cdots|\mathbf{W}^{(M)}]_{N\times M}$, $F = [\mathbf{W}^{(0)} + \Delta t\mathbf{f}^{(1)}|\cdots|\Delta t\mathbf{f}^{M}]_{N\times M}$, and rewrite our problem as

$$\text{Compute } W \in \mathbb{R}^{N\times M} \text{ s.t. } A_N W + W B_M^T = F.$$

💡 This is a well-know object called **Sylvester equation**!

❓ Did we gain anything?

Matrix equation reformulations

❓ Where is the advantage in dealing with

$$(B_M \otimes I_N + I_M \otimes A_N)\,\hat{\mathbf{W}} = \mathbf{F}, \qquad A_N = -\frac{\Delta t}{h^\alpha} T_N?$$

Let's call $W = [\mathbf{W}^{(1)}|\cdots|\mathbf{W}^{(M)}]_{N\times M}$, $F = [\mathbf{W}^{(0)} + \Delta t\mathbf{f}^{(1)}|\cdots|\Delta t\mathbf{f}^{M}]_{N\times M}$, and rewrite our problem as

$$\text{Compute } W \in \mathbb{R}^{N\times M} \text{ s.t. } A_N W + W B_M^T = F.$$

- This is a well-know object called **Sylvester equation**!
- ❓ Did we gain anything? Back to this in a few moment...

Matrix equation reformulations

❓ Where is the advantage in dealing with

$$(B_M \otimes I_N + I_M \otimes A_N)\,\hat{\mathbf{W}} = \mathbf{F}, \qquad A_N = -\frac{\Delta t}{h^\alpha} T_N?$$

Let's call $W = [\mathbf{W}^{(1)}|\cdots|\mathbf{W}^{(M)}]_{N\times M}$, $F = [\mathbf{W}^{(0)} + \Delta t\mathbf{f}^{(1)}|\cdots|\Delta t\mathbf{f}^{M}]_{N\times M}$, and rewrite our problem as

$$\text{Compute } W \in \mathbb{R}^{N\times M} \text{ s.t. } A_N W + W B_M^T = F.$$

- This is a well-know object called **Sylvester equation**!
- ❓ Did we gain anything? Back to this in a few moment...
- ❓ Since we are accumulating all the time steps in one step, is it appropriate to simply use one of the methods we already know (e.g. Euler, BDFs, Adams', etc.) or can we do better? Next lecture!

What about the 2D problem?

What happens if we want then to reformulate:

$$\begin{cases} \frac{\partial W}{\partial t} = \left(\theta\, {}^{RL}D^{\alpha}_{[0,x]} \cdot + (1-\theta)\, {}^{RL}D^{\alpha}_{[x,1]} \cdot\right) W(x,y,t) \\ \qquad\qquad + \left(\theta\, {}^{RL}D^{\alpha}_{[0,y]} \cdot + (1-\theta)\, {}^{RL}D^{\alpha}_{[y,1]} \cdot\right) W(x,y,t) + f(x,y,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

What about the 2D problem?

What happens if we want then to reformulate:

$$\begin{cases} \frac{\partial W}{\partial t} = \left(\theta\, {}^{GL}D^{\alpha}_{[0,x]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} \cdot\right) W(x,y,t) \\ \qquad\qquad + \left(\theta\, {}^{GL}D^{\alpha}_{[0,y]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[y,1]} \cdot\right) W(x,y,t) + f(x,y,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

What about the 2D problem?

What happens if we want then to reformulate:

$$\begin{cases} \frac{\partial W}{\partial t} = \left(\theta\, {}^{GL}D^{\alpha}_{[0,x]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} \cdot\right) W(x,y,t) \\ \qquad\qquad + \left(\theta\, {}^{GL}D^{\alpha}_{[0,y]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[y,1]} \cdot\right) W(x,y,t) + f(x,y,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

By the usual procedure

$$\frac{1}{\Delta t}\left(\mathbf{W}^{(m+1)} - \mathbf{W}^{(m)}\right) = \frac{1}{h^{\alpha}}\left((\theta G_{N_x} + (1-\theta) G_{N_x}^T) \otimes I_{N_y} + I_{N_x} \otimes (\theta G_{N_y} + (1-\theta) G_{N_y}^T)\right) \mathbf{W}^{(m+1)} + \mathbf{f}^{(m+1)}, \quad m = 0, \ldots, M-1.$$

What about the 2D problem?

What happens if we want then to reformulate:

$$\begin{cases} \frac{\partial W}{\partial t} = \left(\theta\, {}^{GL}D^{\alpha}_{[0,x]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} \cdot\right) W(x,y,t) \\ \qquad\qquad + \left(\theta\, {}^{GL}D^{\alpha}_{[0,y]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[y,1]} \cdot\right) W(x,y,t) + f(x,y,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

By the usual procedure

$$\frac{1}{\Delta t}\left(\mathbf{W}^{(m+1)} - \mathbf{W}^{(m)}\right) = \left(\tilde{G}_{N_x} \otimes I_{N_y} + I_{N_x} \otimes \tilde{G}_{N_y}\right) \mathbf{W}^{(m+1)} + \mathbf{f}^{(m+1)}, \quad m = 0, \ldots, M-1.$$

What about the 2D problem?

What happens if we want then to reformulate:

$$\begin{cases} \frac{\partial W}{\partial t} = \left(\theta\, {}^{GL}D^{\alpha}_{[0,x]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} \cdot\right) W(x,y,t) \\ \qquad\qquad + \left(\theta\, {}^{GL}D^{\alpha}_{[0,y]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[y,1]} \cdot\right) W(x,y,t) + f(x,y,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

By the usual procedure

$$\frac{1}{\Delta t}\left(\mathbf{W}^{(m+1)} - \mathbf{W}^{(m)}\right) = \underbrace{\left(\tilde{G}_{N_x} \otimes I_{N_y} + I_{N_x} \otimes \tilde{G}_{N_y}\right)}_{G_{N_x N_y}} \mathbf{W}^{(m+1)} + \mathbf{f}^{(m+1)}, \quad m = 0, \ldots, M-1.$$

What about the 2D problem?

What happens if we want then to reformulate:

$$\begin{cases} \frac{\partial W}{\partial t} = \left(\theta\, {}^{GL}D^{\alpha}_{[0,x]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} \cdot\right) W(x,y,t) \\ \qquad\qquad + \left(\theta\, {}^{GL}D^{\alpha}_{[0,y]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[y,1]} \cdot\right) W(x,y,t) + f(x,y,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

By the usual procedure

$$\left(I_{N_x N_y} - \Delta t G_{N_x N_y}\right) \mathbf{W}^{(m+1)} = \mathbf{W}^{(m)} + \Delta \mathbf{f}^{m+1}, \quad m = 0, \ldots, M-1.$$

What about the 2D problem?

What happens if we want then to reformulate:

$$\begin{cases} \frac{\partial W}{\partial t} = \left(\theta\, {}^{GL}D^{\alpha}_{[0,x]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} \cdot\right) W(x,y,t) \\ \qquad\qquad + \left(\theta\, {}^{GL}D^{\alpha}_{[0,y]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[y,1]} \cdot\right) W(x,y,t) + f(x,y,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

By the usual procedure

$$\left(I_{N_x N_y} - \Delta t G_{N_x N_y}\right) \mathbf{W}^{(m+1)} = \mathbf{W}^{(m)} + \Delta \mathbf{f}^{m+1}, \quad m = 0, \ldots, M-1.$$

- The **clever observation** is now that

$$I_{N_x N_y} - \Delta t G_{N_x N_y} = I_{N_y} \otimes \left(\frac{1}{2} I_{N_x} - \Delta t \tilde{G}_{N_x}\right) + \left(\frac{1}{2} I_{N_y} - \Delta t \tilde{G}_{N_y}\right) \otimes I_{N_x}.$$

What about the 2D problem?

What happens if we want then to reformulate:

$$\begin{cases} \frac{\partial W}{\partial t} = \left(\theta\, {}^{GL}D^{\alpha}_{[0,x]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} \cdot\right) W(x,y,t) \\ \qquad\qquad + \left(\theta\, {}^{GL}D^{\alpha}_{[0,y]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[y,1]} \cdot\right) W(x,y,t) + f(x,y,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

By the usual procedure

$$\left(\frac{1}{2} I_{N_x} - \Delta t \tilde{G}_{N_x}\right) \tilde{W}^{(m+1)} + \tilde{W}^{(m+1)} \left(\frac{1}{2} I^{N_y} - \Delta t \tilde{G}_{N_y}\right)^T = \tilde{W}^{(m)} + \Delta t F^{(m+1)}, \quad m = 0, \ldots, M-1.$$

- The **clever observation** is now that

$$I_{N_x N_y} - \Delta t G_{N_x N_y} = I_{N_y} \otimes \left(\frac{1}{2} I_{N_x} - \Delta t \tilde{G}_{N_x}\right) + \left(\frac{1}{2} I_{N_y} - \Delta t \tilde{G}_{N_y}\right) \otimes I_{N_x}.$$

What about the 2D problem?

What happens if we want then to reformulate:

$$\begin{cases} \frac{\partial W}{\partial t} = \left(\theta\, {}^{GL}D^{\alpha}_{[0,x]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[x,1]} \cdot\right) W(x,y,t) \\ \qquad\qquad + \left(\theta\, {}^{GL}D^{\alpha}_{[0,y]} \cdot + (1-\theta)\, {}^{GL}D^{\alpha}_{[y,1]} \cdot\right) W(x,y,t) + f(x,y,t), \qquad \theta \in [0,1], \\ W(0,t) = W(1,t) = 0, \qquad W(x,t) = W_0(x). \end{cases}$$

By the usual procedure

$$\left(\frac{1}{2} I_{N_x} - \Delta t \tilde{G}_{N_x}\right) \tilde{W}^{(m+1)} + \tilde{W}^{(m+1)} \left(\frac{1}{2} I^{N_y} - \Delta t \tilde{G}_{N_y}\right)^T = \tilde{W}^{(m)} + \Delta t F^{(m+1)}, \quad m = 0, \ldots, M-1.$$

- We now have a **sequence of Sylvester equations** for $m = 0, \ldots, M-1$.
- The matrix coefficients are related to *rescaled* 1D problems.

Solving Sylvester equations (Simoncini 2016)

This rewriting effort will be worth it only if we can **efficiently solve** Sylvester equations:

$$AX + XB = C, \quad A \in \mathbb{R}^{N \times N},\ B \in \mathbb{R}^{M \times M}, \quad C \in \mathbb{R}^{N \times M}.$$

Solving Sylvester equations (Simoncini 2016)

This rewriting effort will be worth it only if we can **efficiently solve** Sylvester equations:

$$AX + XB = C, \quad A \in \mathbb{R}^{N\times N},\ B \in \mathbb{R}^{M\times M}, \quad C \in \mathbb{R}^{N\times M}.$$

The solution can be expressed in **closed form** in a number of ways, e.g., as *integrals of resolvents*

$$X = -\frac{1}{4\pi^2}\int_{\Gamma_1}\int_{\Gamma_2} \frac{(\gamma I_N - A)^{-1} C (\mu I_M - B)^{-1}}{\lambda + \mu}\, \mathrm{d}\mu \mathrm{d}\lambda,$$

for Γ_1, Γ_2 contours containing and sufficiently close to the spectra of A and B, respectively.

Solving Sylvester equations (Simoncini 2016)

This rewriting effort will be worth it only if we can **efficiently solve** Sylvester equations:

$$AX + XB = C, \quad A \in \mathbb{R}^{N\times N},\ B \in \mathbb{R}^{M\times M}, \quad C \in \mathbb{R}^{N\times M}.$$

The solution can be expressed in **closed form** in a number of ways, e.g., as *integrals of exponentials*

$$X = -\int_0^{+\infty} e^{At} C e^{Bt}\,\mathrm{d}t,$$

for A and B with a spectra separated by a vertical line.

Solving Sylvester equations (Simoncini 2016)

This rewriting effort will be worth it only if we can **efficiently solve** Sylvester equations:

$$AX + XB = C, \quad A \in \mathbb{R}^{N\times N},\ B \in \mathbb{R}^{M\times M}, \quad C \in \mathbb{R}^{N\times M}.$$

The solution can be expressed in **closed form** in a number of ways, e.g., in the *diagonalizable case*, by means of *similarity transformations*

$$U^{-1}AU = \mathrm{diag}(\lambda_1, \ldots, \lambda_N), \qquad V^{-1}BV = \mathrm{diag}(\mu_1, \ldots, \mu_M),$$

then

$$X = U\tilde{X}V, \quad \tilde{x}_{i,j} = \frac{1}{\lambda_i + \mu_j}(U^{-1}CV)_{i,j}.$$

Solving Sylvester equations (Simoncini 2016)

This rewriting effort will be worth it only if we can **efficiently solve** Sylvester equations:

$$AX + XB = C, \quad A \in \mathbb{R}^{N\times N},\ B \in \mathbb{R}^{M\times M}, \quad C \in \mathbb{R}^{N\times M}.$$

The solution can be expressed in **closed form** in a number of ways, e.g., in the *diagonalizable case*, by means of *similarity transformations*

$$U^{-1}AU = \mathrm{diag}(\lambda_1, \ldots, \lambda_N), \qquad V^{-1}BV = \mathrm{diag}(\mu_1, \ldots, \mu_M),$$

then

$$X = U\tilde{X}V, \quad \tilde{x}_{i,j} = \frac{1}{\lambda_i + \mu_j}(U^{-1}CV)_{i,j}.$$

Numerical mehtods

These formulations can exploited to devise numerical methods, to avoid a very long detour, we are going to just mention a couple of them; read (Simoncini 2016) for the full story.

The Bartels and Stewart 1972 algorithm

Input: A, B, C
Compute Schur factorizations
$URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

The Bartels and Stewart 1972 algorithm

- The first step costs $O(N^3)$ and $O(M^3)$ operations by **QR algorithm** for general A and B,

Input: A, B, C
Compute Schur factorizations
$URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

The Bartels and Stewart 1972 algorithm

- The first step costs $O(N^3)$ and $O(M^3)$ operations by **QR algorithm** for general A and B,
- The last step is just two matrix-matrix multiplications.

Input: A, B, C
Compute Schur factorizations
$URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

The Bartels and Stewart 1972 algorithm

- The first step costs $O(N^3)$ and $O(M^3)$ operations by **QR algorithm** for general A and B,
- The last step is just two matrix-matrix multiplications.

Input: A, B, C
Compute Schur factorizations
$URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

We can solve the system with triangular coefficients by substitution

$$R^H Y + YS = U^H CV$$

The Bartels and Stewart 1972 algorithm

- The first step costs $O(N^3)$ and $O(M^3)$ operations by **QR algorithm** for general A and B,
- The last step is just two matrix-matrix multiplications.

Input: A, B, C
Compute Schur factorizations
$URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

We can solve the system with triangular coefficients by substitution

$$\begin{bmatrix} \spadesuit & & \\ \spadesuit & \spadesuit & \\ \spadesuit & \spadesuit & \spadesuit \end{bmatrix} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \begin{bmatrix} \clubsuit & \clubsuit & \clubsuit \\ & \clubsuit & \clubsuit \\ & & \clubsuit \end{bmatrix} = \begin{bmatrix} \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \end{bmatrix}$$

The Bartels and Stewart 1972 algorithm

- The first step costs $O(N^3)$ and $O(M^3)$ operations by **QR algorithm** for general A and B,
- The last step is just two matrix-matrix multiplications.

Input: A, B, C
Compute Schur factorizations $URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

We can solve the system with triangular coefficients by substitution

$$\begin{bmatrix} \spadesuit & & \\ \spadesuit & \spadesuit & \\ \spadesuit & \spadesuit & \spadesuit \end{bmatrix} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \begin{bmatrix} \clubsuit & \clubsuit & \clubsuit \\ & \clubsuit & \clubsuit \\ & & \clubsuit \end{bmatrix} = \begin{bmatrix} \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \end{bmatrix}$$

$(Y)_{1,1}$ element is readily obtained by solving: $(\spadesuit + \clubsuit)(Y)_{11} = \bigstar$.

The Bartels and Stewart 1972 algorithm

- The first step costs $O(N^3)$ and $O(M^3)$ operations by **QR algorithm** for general A and B,
- The last step is just two matrix-matrix multiplications.

Input: A, B, C
Compute Schur factorizations
$URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

We can solve the system with triangular coefficients by substitution

$$\begin{bmatrix} \spadesuit & & \\ \spadesuit & \spadesuit & \\ \spadesuit & \spadesuit & \spadesuit \end{bmatrix} \begin{bmatrix} y_{1,1} & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} + \begin{bmatrix} y_{11} & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \begin{bmatrix} \clubsuit & \clubsuit & \clubsuit \\ & \clubsuit & \clubsuit \\ & & \clubsuit \end{bmatrix} = \begin{bmatrix} \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \end{bmatrix}$$

Then we proceed with the first row...

The Bartels and Stewart 1972 algorithm

- The first step costs $O(N^3)$ and $O(M^3)$ operations by **QR algorithm** for general A and B,
- The last step is just two matrix-matrix multiplications.

Input: A, B, C
Compute Schur factorizations $URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

We can solve the system with triangular coefficients by substitution

$$\begin{bmatrix} \spadesuit & & \\ \spadesuit & \spadesuit & \\ \spadesuit & \spadesuit & \spadesuit \end{bmatrix} \begin{bmatrix} y_{1,1} & y_{12} & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} + \begin{bmatrix} y_{11} & y_{12} & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \begin{bmatrix} \clubsuit & \clubsuit & \clubsuit \\ & \clubsuit & \clubsuit \\ & & \clubsuit \end{bmatrix} = \begin{bmatrix} \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \end{bmatrix}$$

Then we proceed with the first row...

The Bartels and Stewart 1972 algorithm

- The first step costs $O(N^3)$ and $O(M^3)$ operations by **QR algorithm** for general A and B,
- The last step is just two matrix-matrix multiplications.

Input: A, B, C
Compute Schur factorizations
$URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

We can solve the system with triangular coefficients by substitution

$$\begin{bmatrix} \spadesuit & & \\ \spadesuit & \spadesuit & \\ \spadesuit & \spadesuit & \spadesuit \end{bmatrix} \begin{bmatrix} y_{1,1} & y_{12} & y_{1,3} \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} + \begin{bmatrix} y_{11} & y_{12} & y_{1,3} \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \begin{bmatrix} \clubsuit & \clubsuit & \clubsuit \\ & \clubsuit & \clubsuit \\ & & \clubsuit \end{bmatrix} = \begin{bmatrix} \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \end{bmatrix}$$

Then we proceed with the first row... and every other row.

The Bartels and Stewart 1972 algorithm

- The first step costs $O(N^3)$ and $O(M^3)$ operations by **QR algorithm** for general A and B,
- The last step is just two matrix-matrix multiplications.

Input: A, B, C
Compute Schur factorizations
$URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

We can solve the system with triangular coefficients by substitution

$$\begin{bmatrix} \spadesuit & & \\ \spadesuit & \spadesuit & \\ \spadesuit & \spadesuit & \spadesuit \end{bmatrix} \begin{bmatrix} y_{1,1} & y_{12} & y_{1,3} \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} + \begin{bmatrix} y_{11} & y_{12} & y_{1,3} \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \begin{bmatrix} \clubsuit & \clubsuit & \clubsuit \\ & \clubsuit & \clubsuit \\ & & \clubsuit \end{bmatrix} = \begin{bmatrix} \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \end{bmatrix}$$

Then we proceed with the first row... and every other row.

The Bartels and Stewart 1972 algorithm

- The first step costs $O(N^3)$ and $O(M^3)$ operations by **QR algorithm** for general A and B,
- The last step is just two matrix-matrix multiplications.

Input: A, B, C
Compute Schur factorizations
$URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

We can solve the system with triangular coefficients by substitution

$$\begin{bmatrix} \spadesuit & & \\ \spadesuit & \spadesuit & \\ \spadesuit & \spadesuit & \spadesuit \end{bmatrix} \begin{bmatrix} y_{1,1} & y_{12} & y_{1,3} \\ y_{2,1} & y_{2,2} & y_{2,3} \\ \times & \times & \times \end{bmatrix} + \begin{bmatrix} y_{11} & y_{12} & y_{1,3} \\ y_{2,1} & y_{2,2} & y_{2,3} \\ \times & \times & \times \end{bmatrix} \begin{bmatrix} \clubsuit & \clubsuit & \clubsuit \\ & \clubsuit & \clubsuit \\ & & \clubsuit \end{bmatrix} = \begin{bmatrix} \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \end{bmatrix}$$

Then we proceed with the first row... and every other row.

The Bartels and Stewart 1972 algorithm

- The first step costs $O(N^3)$ and $O(M^3)$ operations by **QR algorithm** for general A and B,
- The last step is just two matrix-matrix multiplications.

Input: A, B, C
Compute Schur factorizations $URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

We can solve the system with triangular coefficients by substitution

$$\begin{bmatrix} \spadesuit & & \\ \spadesuit & \spadesuit & \\ \spadesuit & \spadesuit & \spadesuit \end{bmatrix} \begin{bmatrix} y_{1,1} & y_{12} & y_{1,3} \\ y_{2,1} & y_{2,2} & y_{2,3} \\ y_{3,1} & y_{3,2} & y_{3,3} \end{bmatrix} + \begin{bmatrix} y_{11} & y_{12} & y_{1,3} \\ y_{2,1} & y_{2,2} & y_{2,3} \\ y_{3,1} & y_{3,2} & y_{3,3} \end{bmatrix} \begin{bmatrix} \clubsuit & \clubsuit & \clubsuit \\ & \clubsuit & \clubsuit \\ & & \clubsuit \end{bmatrix} = \begin{bmatrix} \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \end{bmatrix}$$

Then we proceed with the first row... and every other row.
$\Rightarrow$ The **leading cost** is the Schur factorization $O(N^3 + M^3)$! only **small matrices**.

The Bartels and Stewart 1972 algorithm

- The first step costs $O(N^3)$ and $O(M^3)$ operations by **QR algorithm** for general A and B,
- The last step is just two matrix-matrix multiplications.

Input: A, B, C
Compute Schur factorizations $URU^H = A^H$ and $B = VSV^H$;
Solve $R^H Y + YS = U^H CV$ for Y;
Compute $X = UYV^H$;

We can solve the system with triangular coefficients by substitution

$$\begin{bmatrix} \spadesuit & & \\ \spadesuit & \spadesuit & \\ \spadesuit & \spadesuit & \spadesuit \end{bmatrix} \begin{bmatrix} y_{1,1} & y_{12} & y_{1,3} \\ y_{2,1} & y_{2,2} & y_{2,3} \\ y_{3,1} & y_{3,2} & y_{3,3} \end{bmatrix} + \begin{bmatrix} y_{11} & y_{12} & y_{1,3} \\ y_{2,1} & y_{2,2} & y_{2,3} \\ y_{3,1} & y_{3,2} & y_{3,3} \end{bmatrix} \begin{bmatrix} \clubsuit & \clubsuit & \clubsuit \\ & \clubsuit & \clubsuit \\ & & \clubsuit \end{bmatrix} = \begin{bmatrix} \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \\ \bigstar & \bigstar & \bigstar \end{bmatrix}$$

Then we proceed with the first row... and every other row.

$\Rightarrow$ The **leading cost** is the Schur factorization $O(N^3 + M^3)$! only **small matrices**.

We may gain something if A and B are in **upper Hessenberg form**...

The small case scenario

There are a number of variations that we can apply for the case of small matrices

The small case scenario

There are a number of variations that we can apply for the case of small matrices

- We can use **real Schur form** instead of the complex one, avoids complex arithmetic, but now for in the second step we have to solve some Sylvester equation with 2×2 coefficients. We do it by going back to a small linear system.

The small case scenario

There are a number of variations that we can apply for the case of small matrices

- We can use **real Schur form** instead of the complex one, avoids complex arithmetic, but now for in the second step we have to solve some Sylvester equation with 2×2 coefficients. We do it by going back to a small linear system.
- We can go directly for the Hessenberg form instead of Schur (Golub, Nash, and Van Loan 1979).

The small case scenario

There are a number of variations that we can apply for the case of small matrices

- We can use **real Schur form** instead of the complex one, avoids complex arithmetic, but now for in the second step we have to solve some Sylvester equation with 2×2 coefficients. We do it by going back to a small linear system.
- We can go directly for the Hessenberg form instead of Schur (Golub, Nash, and Van Loan 1979).
- If A is much larger than B then we can work on the block case

$$\begin{bmatrix} A_{11} & A_{12} \\ & A_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} B = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}.$$

The small case scenario

There are a number of variations that we can apply for the case of small matrices

- We can use **real Schur form** instead of the complex one, avoids complex arithmetic, but now for in the second step we have to solve some Sylvester equation with 2×2 coefficients. We do it by going back to a small linear system.
- We can go directly for the Hessenberg form instead of Schur (Golub, Nash, and Van Loan 1979).
- If A is much larger than B then we can work on the block case

$$\begin{bmatrix} A_{11} & A_{12} \\ & A_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} B = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}.$$

The small case scenario

There are a number of variations that we can apply for the case of small matrices

- We can use **real Schur form** instead of the complex one, avoids complex arithmetic, but now for in the second step we have to solve some Sylvester equation with 2×2 coefficients. We do it by going back to a small linear system.
- We can go directly for the Hessenberg form instead of Schur (Golub, Nash, and Van Loan 1979).
- If A is much larger than B then we can work on the block case.
- If $B = -A$ and C is *small rank*, then we are falling back to our "fast small-displacement-rank solver" scenario, e.g., (Gohberg, Kailath, and Olshevsky 1995).

The small case scenario

There are a number of variations that we can apply for the case of small matrices

- We can use **real Schur form** instead of the complex one, avoids complex arithmetic, but now for in the second step we have to solve some Sylvester equation with 2×2 coefficients. We do it by going back to a small linear system.
- We can go directly for the Hessenberg form instead of Schur (Golub, Nash, and Van Loan 1979).
- If A is much larger than B then we can work on the block case.
- If $B = -A$ and C is *small rank*, then we are falling back to our "fast small-displacement-rank solver" scenario, e.g., (Gohberg, Kailath, and Olshevsky 1995).

But our cases are not small...

If only we knew a way to from a large matrix setting, to a small one made of Hessenberg matrices...

The small case scenario

There are a number of variations that we can apply for the case of small matrices

- We can use **real Schur form** instead of the complex one, avoids complex arithmetic, but now for in the second step we have to solve some Sylvester equation with 2×2 coefficients. We do it by going back to a small linear system.
- We can go directly for the Hessenberg form instead of Schur (Golub, Nash, and Van Loan 1979).
- If A is much larger than B then we can work on the block case.
- If $B = -A$ and C is *small rank*, then we are falling back to our "fast small-displacement-rank solver" scenario, e.g., (Gohberg, Kailath, and Olshevsky 1995).

☺ But our cases are not small...

If only we knew a way to from a large matrix setting, to a small one made of Hessenberg matrices... wait a second, we may know a trick or two for this! ☺

When in doubt: project!

When we have to solve **linear systems** with a **large matrix**, we have seen that a good solution is represented by the Krylov projection methods.

❓ Can we do something similar for this problem too?

Theorem (Simoncini 2016, Theorem 4)

Let A and B be stable[1] and real symmetric, with spectra contained in $[a, b]$ and $[c, d]$, respectively. Define $\eta = 2(b-a)(d-c)/((a+c)(b+d))$. Assume C is of **rank** p. Then the singular values $\sigma_1 \geq \cdots \geq \sigma_{\min\{M,N\}}$ of the solution X to the Sylvester equation satisfy

$$\frac{\sigma_{pr+1}}{\sigma_1} \leq \left(\frac{1-\sqrt{k_r'}}{1+\sqrt{k_r'}}\right)^2, \; 1 \leq pr < n, \; k_r' = \frac{1}{1+\eta+\sqrt{\eta(\eta+2)}}.$$

[1] A matrix is called stable (or sometimes *Hurwitz*) if every eigenvalue has strictly negative real part.

When in doubt: project!

When we have to solve **linear systems** with a **large matrix**, we have seen that a good solution is represented by the Krylov projection methods.

❓ Can we do something similar for this problem too? *sometimes*, it is a **matter of rank**.

Theorem (Simoncini 2016, Theorem 4)

Let A and B be stable[1] and real symmetric, with spectra contained in $[a, b]$ and $[c, d]$, respectively. Define $\eta = 2(b-a)(d-c)/((a+c)(b+d))$. Assume C is of **rank** p. Then the singular values $\sigma_1 \geq \cdots \geq \sigma_{\min\{M,N\}}$ of the solution X to the Sylvester equation satisfy

$$\frac{\sigma_{pr+1}}{\sigma_1} \leq \left(\frac{1-\sqrt{k_r'}}{1+\sqrt{k_r'}}\right)^2, \; 1 \leq pr < n, \; k_r' = \frac{1}{1+\eta+\sqrt{\eta(\eta+2)}}.$$

[1] A matrix is called stable (or sometimes *Hurwitz*) if every eigenvalue has strictly negative real part.

Projection methods for low-rank right-hand sides

Let us **assume** that $\mathrm{rank}\,(C) = p \ll \min\{n, m\}$.

Projection methods for low-rank right-hand sides

Let us **assume** that $\mathrm{rank}\,(C) = p \ll \min\{n, m\}$.

- Decompose $C = C_1 C_2^H$;

Projection methods for low-rank right-hand sides

Let us **assume** that $\mathrm{rank}\,(C) = p \ll \min\{n, m\}$.

- Decompose $C = C_1 C_2^H$;
- Select $\mathcal{V} = \mathrm{span}(V_k)$ and $\mathcal{W} = \mathrm{span}(W_j)$ subspaces of $\mathbb{C}^N$ and $\mathbb{C}^M$;

Projection methods for low-rank right-hand sides

Let us **assume** that $\mathrm{rank}\,(C) = p \ll \min\{n, m\}$.

- Decompose $C = C_1 C_2^H$;
- Select $\mathcal{V} = \mathrm{span}(V_k)$ and $\mathcal{W} = \mathrm{span}(W_j)$ subspaces of $\mathbb{C}^N$ and $\mathbb{C}^M$;
- Basis V_k, $k \ll N$, and W_j, $j \ll M$, are *orthonormal* and such that $\mathcal{V}$ and $\mathcal{W}$ are not orthogonal to $\mathrm{range}(C_1)$ and $\mathrm{range}(C_2)$ respectively;

Projection methods for low-rank right-hand sides

Let us **assume** that $\mathrm{rank}\,(C) = p \ll \min\{n, m\}$.

- Decompose $C = C_1 C_2^H$;
- Select $\mathcal{V} = \mathrm{span}(V_k)$ and $\mathcal{W} = \mathrm{span}(W_j)$ subspaces of $\mathbb{C}^N$ and $\mathbb{C}^M$;
- Basis V_k, $k \ll N$, and W_j, $j \ll M$, are *orthonormal* and such that $\mathcal{V}$ and $\mathcal{W}$ are not orthogonal to $\mathrm{range}(C_1)$ and $\mathrm{range}(C_2)$ respectively;
- Build an approximation $\tilde{X} = V_k Y W_j^H \approx X$ with residual $R = C_1 C_2^H - A\tilde{X} - \tilde{X}B$.

Projection methods for low-rank right-hand sides

Let us **assume** that $\mathrm{rank}\,(C) = p \ll \min\{n, m\}$.

- Decompose $C = C_1 C_2^H$;
- Select $\mathcal{V} = \mathrm{span}(V_k)$ and $\mathcal{W} = \mathrm{span}(W_j)$ subspaces of $\mathbb{C}^N$ and $\mathbb{C}^M$;
- Basis V_k, $k \ll N$, and W_j, $j \ll M$, are *orthonormal* and such that $\mathcal{V}$ and $\mathcal{W}$ are not orthogonal to $\mathrm{range}(C_1)$ and $\mathrm{range}(C_2)$ respectively;
- Build an approximation $\tilde{X} = V_k Y W_j^H \approx X$ with residual $R = C_1 C_2^H - A\tilde{X} - \tilde{X}B$.

Galerkin (orthogonality) condition

Call $\tilde{\mathbf{x}} = \mathrm{vec}(\tilde{X}) = (W_j \otimes V_k)\,\mathrm{vec}(Y)$, then we want V_k and W_k to be selected as

$$(W_j \otimes V_k)^H(\mathbf{c} - \mathcal{A}\mathbf{x}) = 0 \;\Leftrightarrow\; V_k^H R W_j = 0 \text{ with } \mathcal{A} = B^T \otimes I + I \otimes A,\ \mathbf{c} = \mathrm{vec}(C).$$

Projection methods for low-rank right-hand sides

Let us **assume** that $\mathrm{rank}\,(C) = p \ll \min\{n, m\}$.

- Decompose $C = C_1 C_2^H$;
- Select $\mathcal{V} = \mathrm{span}(V_k)$ and $\mathcal{W} = \mathrm{span}(W_j)$ subspaces of $\mathbb{C}^N$ and $\mathbb{C}^M$;
- Basis V_k, $k \ll N$, and W_j, $j \ll M$, are *orthonormal* and such that $\mathcal{V}$ and $\mathcal{W}$ are not orthogonal to $\mathrm{range}(C_1)$ and $\mathrm{range}(C_2)$ respectively;
- Build an approximation $\tilde{X} = V_k Y W_j^H \approx X$ with residual $R = C_1 C_2^H - A\tilde{X} - \tilde{X}B$.

Galerkin (orthogonality) condition

Call $\tilde{\mathbf{x}} = \mathrm{vec}(\tilde{X}) = (W_j \otimes V_k)\,\mathrm{vec}(Y)$, then we want V_k and W_k to be selected as

$$(W_j \otimes V_k)^H(\mathbf{c} - \mathcal{A}\mathbf{x}) = 0 \;\Leftrightarrow\; V_k^H R W_j = 0 \text{ with } \mathcal{A} = B^T \otimes I + I \otimes A,\ \mathbf{c} = \mathrm{vec}(C).$$

- To compute Y, solve the **small Sylvester equation**:

$$V_k^H A V_k Y + Y W_j^H B W_j = V_k^H C_1 (W_j^H C_2)^H.$$

Projection methods for low-rank right-hand sides

Let us **assume** that $\mathrm{rank}\,(C) = p \ll \min\{n, m\}$.

- Decompose $C = C_1 C_2^H$;
- Select $\mathcal{V} = \mathrm{span}(V_k)$ and $\mathcal{W} = \mathrm{span}(W_j)$ subspaces of $\mathbb{C}^N$ and $\mathbb{C}^M$;
- Basis V_k, $k \ll N$, and W_j, $j \ll M$, are *orthonormal* and such that $\mathcal{V}$ and $\mathcal{W}$ are not orthogonal to $\mathrm{range}(C_1)$ and $\mathrm{range}(C_2)$ respectively;
- Build an approximation $\tilde{X} = V_k Y W_j^H \approx X$ with residual $R = C_1 C_2^H - A\tilde{X} - \tilde{X}B$.

Galerkin (orthogonality) condition

Call $\tilde{\mathbf{x}} = \mathrm{vec}(\tilde{X}) = (W_j \otimes V_k)\,\mathrm{vec}(Y)$, then we want V_k and W_k to be selected as

$$(W_j \otimes V_k)^H(\mathbf{c} - \mathcal{A}\mathbf{x}) = 0 \;\Leftrightarrow\; V_k^H R W_j = 0 \text{ with } \mathcal{A} = B^T \otimes I + I \otimes A,\ \mathbf{c} = \mathrm{vec}(C).$$

- To compute Y, solve the **small Sylvester equation**:

$$V_k^H A V_k Y + Y W_j^H B W_j = V_k^H C_1 (W_j^H C_2)^H \Rightarrow \text{Bartels and Stewart.}$$

Projection methods for low-rank right-hand sides

Existence of the solution

If $V_k^H A V_k$ and $-W_j^H B W_j$ have **disjoint spectra** we can solve

$$V_k^H A V_k Y + Y W_j^H B W_j = V_k^H C_1 (W_j^H C_2)^H \qquad \forall\, C = C_1 C_2^H.$$

To enforce it, is sufficient to have A and $-B$ with disjoint field of values.

Projection methods for low-rank right-hand sides

The cost of **one iteration** for $m > n$ and $p = \mathrm{rank}(C)$ is then given by

Input: A, B, C_1 and C_2
Orthogonalize columns of C_1 to get $\mathbf{v}_1 = V_1$;
Orthogonalize columns of C_2 to get $\mathbf{v}_2 = W_1$;
for $k = 1, 2, \ldots,$ **do**
  Compute Y_k solution to
  $V_k^H A V_k Y + Y W_k^H B W_k - V_k^H C_1 (W_k^H C_2)^H = 0$;
  if *converged* **then**
    Return V_k, Y_k and W_k such that
    $X_k = V_k Y_k W_k^*$ and **stop**.
  end
  `/* Compute next bases blocks */`
  Compute $\tilde{v}$ and $\hat{w}$ from the **approximate space**;
  Make $\hat{v}$ orthogonal w.r.t. $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$;
  Make $\hat{w}$ orthogonal w.r.t. $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$;
  Orthogonalize col.s of $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$ for $\mathbf{v}_{k+1}$ and $\mathbf{w}_{k+1}$;
  Update: $V_{k+1} = [V_k, \mathbf{v}_{k+1}]$, $W_{k+1} = [W_k, \mathbf{w}_{k+1}]$;
end

Projection methods for low-rank right-hand sides

The cost of **one iteration** for $m > n$ and $p = \mathrm{rank}(C)$ is then given by

- $O((kp)^3)$ flops for the solution of the projected problem,

Input: A, B, C_1 and C_2
Orthogonalize columns of C_1 to get $\mathbf{v}_1 = V_1$;
Orthogonalize columns of C_2 to get $\mathbf{v}_2 = W_1$;
for $k = 1, 2, \ldots,$ **do**
  Compute Y_k solution to
  $V_k^H A V_k Y + Y W_k^H B W_k - V_k^H C_1 (W_k^H C_2)^H = 0$;
  if *converged* **then**
    Return V_k, Y_k and W_k such that
    $X_k = V_k Y_k W_k^*$ and **stop**.
  end
  `/* Compute next bases blocks */`
  Compute $\tilde{v}$ and $\hat{w}$ from the **approximate space**;
  Make $\hat{v}$ orthogonal w.r.t. $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$;
  Make $\hat{w}$ orthogonal w.r.t. $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$;
  Orthogonalize col.s of $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$ for $\mathbf{v}_{k+1}$ and $\mathbf{w}_{k+1}$;
  Update: $V_{k+1} = [V_k, \mathbf{v}_{k+1}]$, $W_{k+1} = [W_k, \mathbf{w}_{k+1}]$;
end

Projection methods for low-rank right-hand sides

The cost of **one iteration** for $m > n$ and $p = \mathrm{rank}(C)$ is then given by

- $O((kp)^3)$ flops for the solution of the projected problem,
- Orthogonalization of the new basis vectors with respect to the older vectors: $O(mkp^2)$,

Input: A, B, C_1 and C_2
Orthogonalize columns of C_1 to get $\mathbf{v}_1 = V_1$;
Orthogonalize columns of C_2 to get $\mathbf{v}_2 = W_1$;
for $k = 1, 2, \ldots,$ **do**
  Compute Y_k solution to
  $V_k^H A V_k Y + Y W_k^H B W_k - V_k^H C_1 (W_k^H C_2)^H = 0$;
  if *converged* **then**
    Return V_k, Y_k and W_k such that
    $X_k = V_k Y_k W_k^*$ and **stop**.
  end
  `/* Compute next bases blocks */`
  Compute $\tilde{v}$ and $\hat{w}$ from the **approximate space**;
  Make $\hat{v}$ orthogonal w.r.t. $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$;
  Make $\hat{w}$ orthogonal w.r.t. $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$;
  Orthogonalize col.s of $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$ for $\mathbf{v}_{k+1}$ and $\mathbf{w}_{k+1}$;
  Update: $V_{k+1} = [V_k, \mathbf{v}_{k+1}]$, $W_{k+1} = [W_k, \mathbf{w}_{k+1}]$;
end

Projection methods for low-rank right-hand sides

The cost of **one iteration** for $m > n$ and $p = \mathrm{rank}(C)$ is then given by

- $O((kp)^3)$ flops for the solution of the projected problem,
- Orthogonalization of the new basis vectors with respect to the older vectors: $O(mkp^2)$,
- Orthogonalization of the new block: $O(mp^2)$.

Input: A, B, C_1 and C_2
Orthogonalize columns of C_1 to get $\mathbf{v}_1 = V_1$;
Orthogonalize columns of C_2 to get $\mathbf{v}_2 = W_1$;
for $k = 1, 2, \ldots,$ **do**
  Compute Y_k solution to
  $V_k^H A V_k Y + Y W_k^H B W_k - V_k^H C_1 (W_k^H C_2)^H = 0$;
  if *converged* **then**
    Return V_k, Y_k and W_k such that
    $X_k = V_k Y_k W_k^*$ and **stop**.
  end
  `/* Compute next bases blocks */`
  Compute $\tilde{v}$ and $\hat{w}$ from the **approximate space**;
  Make $\hat{v}$ orthogonal w.r.t. $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$;
  Make $\hat{w}$ orthogonal w.r.t. $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$;
  Orthogonalize col.s of $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$ for $\mathbf{v}_{k+1}$ and $\mathbf{w}_{k+1}$;
  Update: $V_{k+1} = [V_k, \mathbf{v}_{k+1}]$, $W_{k+1} = [W_k, \mathbf{w}_{k+1}]$;
end

Projection methods for low-rank right-hand sides

The cost of **one iteration** for $m > n$ and $p = \mathrm{rank}(C)$ is then given by

- $O((kp)^3)$ flops for the solution of the projected problem,
- Orthogonalization of the new basis vectors with respect to the older vectors: $O(mkp^2)$,
- Orthogonalization of the new block: $O(mp^2)$.

Loss of rank

If the generated basis experiences loss of rank, deflation procedures can be applied to remove redundant columns.

Input: A, B, C_1 and C_2
Orthogonalize columns of C_1 to get $\mathbf{v}_1 = V_1$;
Orthogonalize columns of C_2 to get $\mathbf{v}_2 = W_1$;
for $k = 1, 2, \ldots,$ **do**
  Compute Y_k solution to
  $V_k^H A V_k Y + Y W_k^H B W_k - V_k^H C_1 (W_k^H C_2)^H = 0$;
  if *converged* **then**
    Return V_k, Y_k and W_k such that
    $X_k = V_k Y_k W_k^*$ and **stop**.
  end
  `/* Compute next bases blocks */`
  Compute $\tilde{v}$ and $\hat{w}$ from the **approximate space**;
  Make $\hat{v}$ orthogonal w.r.t. $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$;
  Make $\hat{w}$ orthogonal w.r.t. $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$;
  Orthogonalize col.s of $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$ for $\mathbf{v}_{k+1}$ and $\mathbf{w}_{k+1}$;
  Update: $V_{k+1} = [V_k, \mathbf{v}_{k+1}]$, $W_{k+1} = [W_k, \mathbf{w}_{k+1}]$;
end

Projection methods for low-rank right-hand sides

The cost of **one iteration** for $m > n$ and $p = \mathrm{rank}(C)$ is then given by

- $O((kp)^3)$ flops for the solution of the projected problem,
- Orthogonalization of the new basis vectors with respect to the older vectors: $O(mkp^2)$,
- Orthogonalization of the new block: $O(mp^2)$.

Loss of rank

If the generated basis experiences loss of rank, deflation procedures can be applied to remove redundant columns.

Input: A, B, C_1 and C_2
Orthogonalize columns of C_1 to get $\mathbf{v}_1 = V_1$;
Orthogonalize columns of C_2 to get $\mathbf{v}_2 = W_1$;
for $k = 1, 2, \ldots,$ **do**
  Compute Y_k solution to
  $V_k^H A V_k Y + Y W_k^H B W_k - V_k^H C_1 (W_k^H C_2)^H = 0$;
  if *converged* **then**
    Return V_k, Y_k and W_k such that
    $X_k = V_k Y_k W_k^*$ and **stop**.
  end
  `/* Compute next bases blocks */`
  Compute $\tilde{v}$ and $\hat{w}$ from the **approximate space**;
  Make $\hat{v}$ orthogonal w.r.t. $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$;
  Make $\hat{w}$ orthogonal w.r.t. $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$;
  Orthogonalize col.s of $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$ for $\mathbf{v}_{k+1}$ and $\mathbf{w}_{k+1}$;
  Update: $V_{k+1} = [V_k, \mathbf{v}_{k+1}]$, $W_{k+1} = [W_k, \mathbf{w}_{k+1}]$;
end

Selection of $\mathcal{V}$ and $\mathcal{W}$

❓ How do we select the approximation spaces $\mathcal{V}$ and $\mathcal{W}$?

- Standard **block** Krylov subspace

$$\mathcal{V} = \mathrm{range}\{[C_1, AC_1, A^2C_2, \ldots]\}, \quad \mathcal{W} = \mathrm{range}\{[C_2, B^HC_1, (B^H)^2C_2, \ldots]\},$$

- Rational **block** Krylov subspace

$$\begin{aligned}\mathcal{V} &= \mathrm{range}\{[(A+\sigma_1 I)^{-1}C_1, (A+\sigma_2 I)^{-1}(A+\sigma_1 I)^{-1}C_1, \ldots]\},\\ \mathcal{W} &= \mathrm{range}\{[(B^H+\eta_1 I)^{-1}C_2, (B^H+\eta_2 I)^{-1}(B^H+\eta_1 I)^{-1}C_2, \ldots]\},\end{aligned}$$

- Global Krylov subspace:

$$\mathcal{V} = \left\{ \sum_{i\geq 0} A^i C_i \gamma_i, \quad \gamma_i \in \mathbb{R} \right\} = \mathrm{span}\{C_1, AC_1, A^2C_2, \ldots\}$$

where the linear combination is performed blockwise, and analogously for $\mathcal{W}$.

Stopping criterions

To change the "**if** *converged*" in the algorithm we have to monitor the residual, e.g.,

$$\|R\|_2 = \|A\tilde{X} + \tilde{X}B - C_1C_2^*\|_2 \text{ or } \|R\|_F = \|A\tilde{X} + \tilde{X}B - C_1C_2^*\|_F.$$

Stopping criterions

To change the "**if** *converged*" in the algorithm we have to monitor the residual, e.g.,

$$\|R\|_2 = \|A\tilde{X} + \tilde{X}B - C_1C_2^*\|_2 \text{ or } \|R\|_F = \|A\tilde{X} + \tilde{X}B - C_1C_2^*\|_F.$$

⚠ R is **dense** and **large**: we should avoid assembling it!

Stopping criterions

To change the "**if** *converged*" in the algorithm we have to monitor the residual, e.g.,

$$\|R\|_2 = \|A\tilde{X} + \tilde{X}B - C_1 C_2^*\|_2 \text{ or } \|R\|_F = \|A\tilde{X} + \tilde{X}B - C_1 C_2^*\|_F.$$

- R is **dense** and **large**: we should avoid assembling it!
- If we are using Krylov subspaces, we can employ Arnoldi-like relations to this end:

$$AV_k = [V_k, \hat{v}_k]\underline{H}_k \text{ and } B^H W_j = [W_j, \hat{w}_j]\underline{K}_j,$$

  with $[V_k, \hat{v}_k]$ and $[W_j, \hat{w}_j]$ having orthonormal columns.
- If $\exists\, C_1^{(k)}$ and $C_2^{(j)}$ s.t. $C_1 = [V_k, \hat{v}_k]C_1^{(k)}$ and $C_2 = [W_j, \hat{w}_j]C_2^{(j)}$

$$\begin{aligned}
\|R\|_F &= \|AV_k Y W_j^H + V_k Y W_j^H B - \hat{V}_k C_1^{(k)} (\hat{W}_j C_2^{(j)})^H\|_F \\
&= \left\| [V_k \hat{v}_k] \left(\underline{H}_k Y [I, 0] + [I; 0] Y \underline{K}_j^H - C_1^{(k)} (C_2^{(j)})^H \right) [W_j, \hat{w}_j]^H \right\|_F \\
&= \|\underline{H}_k Y [I, 0] + [I; 0] Y \underline{K}_j^H - C_1^{(k)} (C_2^{(j)})^H\|_F.
\end{aligned}$$

Stopping criterions

To change the "**if** *converged*" in the algorithm we have to monitor the residual, e.g.,

$$\|R\|_2 = \|A\tilde{X} + \tilde{X}B - C_1C_2^*\|_2 \text{ or } \|R\|_F = \|A\tilde{X} + \tilde{X}B - C_1C_2^*\|_F.$$

- R is **dense** and **large**: we should avoid assembling it!
- If we are using Krylov subspaces, we can employ Arnoldi-like relations to this end:

$$AV_k = [V_k, \hat{v}_k]\underline{H}_k \text{ and } B^H W_j = [W_j, \hat{w}_j]\underline{K}_j,$$

  with $[V_k, \hat{v}_k]$ and $[W_j, \hat{w}_j]$ having orthonormal columns.
- If $\exists\, C_1^{(k)}$ and $C_2^{(j)}$ s.t. $C_1 = [V_k, \hat{v}_k]C_1^{(k)}$ and $C_2 = [W_j, \hat{w}_j]C_2^{(j)}$

$$\|R\|_F = \|\underline{H}_k Y[I, 0] + [I; 0]Y\underline{K}_j^H - C_1^{(k)}(C_2^{(j)})^H\|_F.$$

- The matrix in the last norm is small **if** k and j are small, if we are under the conditions on the spaces we can **monitor the residual along the way**.

Variants: as many as for standard Krylov methods

- We can use different approximation space dimensions for A and B.

Variants: as many as for standard Krylov methods

- We can use different approximation space dimensions for A and B.
- We can use different approximation spaces of the same dimension.

Variants: as many as for standard Krylov methods

- We can use different approximation space dimensions for A and B.
- We can use different approximation spaces of the same dimension.
- We could use nonsymmetric Lanczos (*oblique* subspaces) if $B = A^H$ and $C_1 C_2^*$ is nonsymmetric to build simultaneously $K_j(A, C_1)$ and $K_j(A, C_2)$.

Variants: as many as for standard Krylov methods

- We can use different approximation space dimensions for A and B.
- We can use different approximation spaces of the same dimension.
- We could use nonsymmetric Lanczos (*oblique* subspaces) if $B = A^H$ and $C_1 C_2^*$ is nonsymmetric to build simultaneously $K_j(A, C_1)$ and $K_j(A, C_2)$.
- We could use Krylov methods with restart to save memory.

Variants: as many as for standard Krylov methods

- We can use different approximation space dimensions for A and B.
- We can use different approximation spaces of the same dimension.
- We could use nonsymmetric Lanczos (*oblique* subspaces) if $B = A^H$ and $C_1 C_2^*$ is nonsymmetric to build simultaneously $K_j(A, C_1)$ and $K_j(A, C_2)$.
- We could use Krylov methods with restart to save memory.
- If A and B are symmetric (and not necessarily equal), we could use short-term block recurrences.

Variants: as many as for standard Krylov methods

- We can use different approximation space dimensions for A and B.
- We can use different approximation spaces of the same dimension.
- We could use nonsymmetric Lanczos (*oblique* subspaces) if $B = A^H$ and $C_1 C_2^*$ is nonsymmetric to build simultaneously $K_j(A, C_1)$ and $K_j(A, C_2)$.
- We could use Krylov methods with restart to save memory.
- If A and B are symmetric (and not necessarily equal), we could use short-term block recurrences.

The review by Simoncini 2016 has pointers to all the different strategies available.

Variants: as many as for standard Krylov methods

- We can use different approximation space dimensions for A and B.
- We can use different approximation spaces of the same dimension.
- We could use nonsymmetric Lanczos (*oblique* subspaces) if $B = A^H$ and $C_1 C_2^*$ is nonsymmetric to build simultaneously $K_j(A, C_1)$ and $K_j(A, C_2)$.
- We could use Krylov methods with restart to save memory.
- If A and B are symmetric (and not necessarily equal), we could use short-term block recurrences.

The review by Simoncini 2016 has pointers to all the different strategies available.

Where were we?

For the two equations we wanted to solve we have then the following questions:

- Is our C low-rank?
- What type of Krylov subspace should we select?
- Does any of this stuff converge at all?

Low-rank, regularity and separability

For the 1D+1D case we have to solve

$$A_N W + W B_M^T = F, \text{ with } F = [\mathbf{W}^{(0)} + \Delta t \mathbf{f}^{(1)} | \cdots | \Delta t \mathbf{f}^{M}]_{N \times M},$$

with $(\mathbf{f}^{(m)})_i = f(x_i, t_m)$.

Low-rank, regularity and separability

- For the 1D+1D case we have to solve

$$A_N W + W B_M^T = F, \text{ with } F = [\mathbf{W}^{(0)} + \Delta t \mathbf{f}^{(1)} | \cdots | \Delta t \mathbf{f}^{M}]_{N \times M},$$

with $(\mathbf{f}^{(m)})_i = f(x_i, t_m)$.

- For the 1D+2D case we have to solve for $m = 0, \ldots, M-1$

$$\left(\frac{1}{2} I_{N_x} - \Delta t \tilde{G}_{N_x}\right) \tilde{W}^{(m+1)} + \tilde{W}^{(m+1)} \left(\frac{1}{2} I^{N_y} - \Delta t \tilde{G}_{N_y}\right)^T = \tilde{W}^{(m)} + \Delta t F^{(m+1)},$$

with $(F^{(m+1)})_{i,j} = f(x_i, y_j, t_{m+1})$.

Low-rank, regularity and separability

For the 1D+1D case we have to solve

$$A_N W + W B_M^T = F, \text{ with } F = [\mathbf{W}^{(0)} + \Delta t \mathbf{f}^{(1)} | \cdots | \Delta t \mathbf{f}^{M}]_{N \times M},$$

with $(\mathbf{f}^{(m)})_i = f(x_i, t_m)$.

For the 1D+2D case we have to solve for $m = 0, \ldots, M-1$

$$\left(\frac{1}{2} I_{N_x} - \Delta t \tilde{G}_{N_x}\right) \tilde{W}^{(m+1)} + \tilde{W}^{(m+1)} \left(\frac{1}{2} I^{N_y} - \Delta t \tilde{G}_{N_y}\right)^T = \tilde{W}^{(m)} + \Delta t F^{(m+1)},$$

with $(F^{(m+1)})_{i,j} = f(x_i, y_j, t_{m+1})$.

Low-Rank

When is it that these matrices have a fixed, size-independent "small" rank?

Low-rank, regularity and separability

- If a function $f(x, y) = f_1(x) f_2(y)$ then

$$\begin{bmatrix} f(x_1, y_1) & f(x_1, y_2) & \cdots & f(x_1, y_n) \\ f(x_2, y_1) & f(x_2, y_2) & \cdots & f(x_2, y_n) \\ \vdots & \vdots & \ddots & \vdots \\ f(x_n, y_1) & f(x_n, y_2) & \cdots & f(x_n, y_n) \end{bmatrix}$$

Low-rank, regularity and separability

If a function $f(x, y) = f_1(x)f_2(y)$ then

$$\begin{bmatrix} f_1(x_1)f_2(y_1) & f_1(x_1)f_2(y_2) & \cdots & f_1(x_1)f_2(y_n) \\ f_1(x_2)f_2(y_1) & f_1(x_2)f_2(y_2) & \cdots & f_1(x_2)f_2(y_n) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_n)f_2(y_1) & f_1(x_n)f_2(y_2) & \cdots & f_1(x_n)f_2(y_n) \end{bmatrix}$$

Low-rank, regularity and separability

If a function $f(x, y) = f_1(x)f_2(y)$ then

$$\begin{bmatrix} f_1(x_1)f_2(y_1) & f_1(x_1)f_2(y_2) & \cdots & f_1(x_1)f_2(y_n) \\ f_1(x_2)f_2(y_1) & f_1(x_2)f_2(y_2) & \cdots & f_1(x_2)f_2(y_n) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_n)f_2(y_1) & f_1(x_n)f_2(y_2) & \cdots & f_1(x_n)f_2(y_n) \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_n) \end{bmatrix} \begin{bmatrix} f_2(y_1) & f_2(y_2) & \cdots & f_n(y_n) \end{bmatrix}$$

Low-rank, regularity and separability

If a function $f(x,y) = f_1(x)f_2(y)$ then

$$\begin{bmatrix} f_1(x_1)f_2(y_1) & f_1(x_1)f_2(y_2) & \cdots & f_1(x_1)f_2(y_n) \\ f_1(x_2)f_2(y_1) & f_1(x_2)f_2(y_2) & \cdots & f_1(x_2)f_2(y_n) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_n)f_2(y_1) & f_1(x_n)f_2(y_2) & \cdots & f_1(x_n)f_2(y_n) \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_n) \end{bmatrix} \begin{bmatrix} f_2(y_1) & f_2(y_2) & \cdots & f_n(y_n) \end{bmatrix}$$

To have a simple example:

```
n = 10;
f1 = @(x) exp(-2*x); f2 = @(y) sin(2*pi*y); f = @(x,y) f1(x).*f2(y);
x = linspace(0,1,n); y = linspace(0,1,n);
[X,Y] = meshgrid(x,y);
A = f(X.',Y.'); a1 = f1(x); a2 = f2(y);
norm(A-a1.'*a2)
```

that answers us `>> ans = 0`.

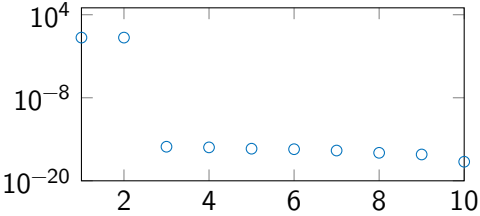
Low-rank, regularity and separability

If a function $f(x,y) = f_1(x)f_2(y)$ then

$$\begin{bmatrix} f_1(x_1)f_2(y_1) & f_1(x_1)f_2(y_2) & \cdots & f_1(x_1)f_2(y_n) \\ f_1(x_2)f_2(y_1) & f_1(x_2)f_2(y_2) & \cdots & f_1(x_2)f_2(y_n) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_n)f_2(y_1) & f_1(x_n)f_2(y_2) & \cdots & f_1(x_n)f_2(y_n) \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_n) \end{bmatrix} \begin{bmatrix} f_2(y_1) & f_2(y_2) & \cdots & f_n(y_n) \end{bmatrix}$$

What happens if $f(x,y)$ is not separable? E.g., if $f(x,y) = \sin(\pi(x+y))$?

```
n = 10;
f = @(x,y) sin(pi*(x+y));
x = linspace(0,1,n);
y = linspace(0,1,n);
[X,Y] = meshgrid(x,y); A = f(X.',Y.');
sv = svd(A);
```



Low-rank, regularity and separability

- If a function $f(x, y) = f_1(x) f_2(y)$ then

$$\begin{bmatrix} f_1(x_1)f_2(y_1) & f_1(x_1)f_2(y_2) & \cdots & f_1(x_1)f_2(y_n) \\ f_1(x_2)f_2(y_1) & f_1(x_2)f_2(y_2) & \cdots & f_1(x_2)f_2(y_n) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_n)f_2(y_1) & f_1(x_n)f_2(y_2) & \cdots & f_1(x_n)f_2(y_n) \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_n) \end{bmatrix} \begin{bmatrix} f_2(y_1) & f_2(y_2) & \cdots & f_n(y_n) \end{bmatrix}$$

What happens if $f(x, y)$ is not separable? E.g., if $f(x, y) = \sin(\pi(x + y))$?

$$\sin(\pi(x + y)) = \sin(\pi x)\cos(\pi y) + \cos(\pi x)\sin(\pi y)$$

is the **sum of** two **separable functions**, i.e., we get a matrix that has rank equal to 2.

Low-rank, regularity and separability

If a function $f(x, y) = f_1(x)f_2(y)$ then

$$\begin{bmatrix} f_1(x_1)f_2(y_1) & f_1(x_1)f_2(y_2) & \cdots & f_1(x_1)f_2(y_n) \\ f_1(x_2)f_2(y_1) & f_1(x_2)f_2(y_2) & \cdots & f_1(x_2)f_2(y_n) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_n)f_2(y_1) & f_1(x_n)f_2(y_2) & \cdots & f_1(x_n)f_2(y_n) \end{bmatrix} = \begin{bmatrix} f_1(x_1) \\ f_1(x_2) \\ \vdots \\ f_1(x_n) \end{bmatrix} \begin{bmatrix} f_2(y_1) & f_2(y_2) & \cdots & f_n(y_n) \end{bmatrix}$$

What happens if $f(x, y)$ is not separable? E.g., if $f(x, y) = \sin(\pi(x + y))$?

$$\sin(\pi(x + y)) = \sin(\pi x)\cos(\pi y) + \cos(\pi x)\sin(\pi y)$$

is the **sum of** two **separable functions**, i.e., we get a matrix that has rank equal to 2.

We can try to **generalize** this **decomposition idea** to more general functions!

Low-rank, regularity and separability

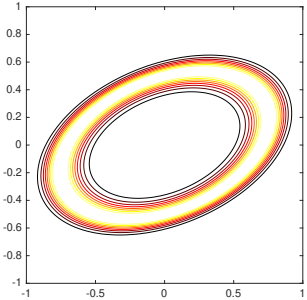
We can approximate a function of two variables as the sum of separable functions

$$f(x,y) = \sum_{k=1}^{K} f_k T_k(x) T_k(y), \quad \{T_k(\cdot)\}_k \text{ Čebyšëv polynomials.}$$

Example (Using `Chebfun` (Driscoll, Hale, and Trefethen 2014))

Consider $f(x,y) = \exp(-40(x^2 - xy + 2y^2 - 1/2)^2)$.

```
cheb.xy
ff=@(x,y)exp(-40*(x.^2-x.*y+2*y.^2-1/2).^2);
f=chebfun2(ff);
levels = 0.1:0.1:0.9;
contour(f,levels);
axis([-1 1 -1 1]);
axis square
```



Low-rank, regularity and separability

We can approximate a function of two variables as the sum of separable functions

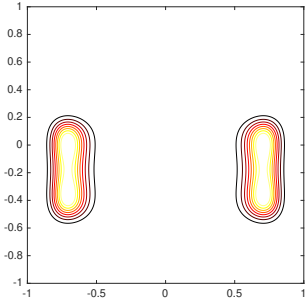
$$f(x,y) = \sum_{k=1}^{K} f_k T_k(x) T_k(y), \quad \{T_k(\cdot)\}_k \text{ Čebyšëv polynomials.}$$

Example (Using `Chebfun` (Driscoll, Hale, and Trefethen 2014))

Consider $f(x,y) = \exp(-40(x^2 - xy + 2y^2 - 1/2)^2)$.

Forcing a rank K approximation

```
levels = 0.1:0.1:0.9;
for K = 1:9
 contour(chebfun2(ff,K),levels)
 xlim([-1 1]), axis equal
end
```



Low-rank, regularity and separability

We can approximate a function of two variables as the sum of separable functions

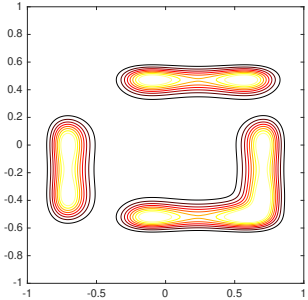
$$f(x,y) = \sum_{k=1}^{K} f_k T_k(x) T_k(y), \quad \{T_k(\cdot)\}_k \text{ Čebyšëv polynomials.}$$

Example (Using `Chebfun` (Driscoll, Hale, and Trefethen 2014))

Consider $f(x,y) = \exp(-40(x^2 - xy + 2y^2 - 1/2)^2)$.

Forcing a rank K approximation

```
levels = 0.1:0.1:0.9;
for K = 1:9
 contour(chebfun2(ff,K),levels)
 xlim([-1 1]), axis equal
end
```



Low-rank, regularity and separability

We can approximate a function of two variables as the sum of separable functions

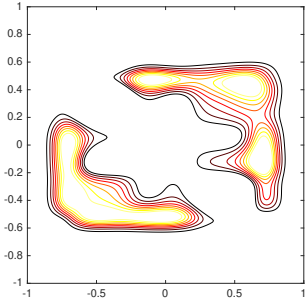
$$f(x,y) = \sum_{k=1}^{K} f_k T_k(x) T_k(y), \quad \{T_k(\cdot)\}_k \text{ Čebyšëv polynomials.}$$

Example (Using `Chebfun` (Driscoll, Hale, and Trefethen 2014))

Consider $f(x,y) = \exp(-40(x^2 - xy + 2y^2 - 1/2)^2)$.

Forcing a rank K approximation

```
levels = 0.1:0.1:0.9;
for K = 1:9
 contour(chebfun2(ff,K),levels)
 xlim([-1 1]), axis equal
end
```



Low-rank, regularity and separability

We can approximate a function of two variables as the sum of separable functions

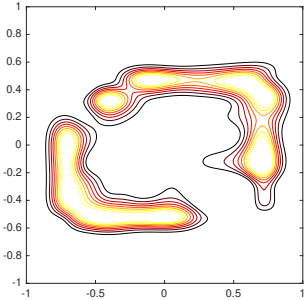
$$f(x,y) = \sum_{k=1}^{K} f_k T_k(x) T_k(y), \quad \{T_k(\cdot)\}_k \text{ Čebyšëv polynomials.}$$

Example (Using `Chebfun` (Driscoll, Hale, and Trefethen 2014))

Consider $f(x,y) = \exp(-40(x^2 - xy + 2y^2 - 1/2)^2)$.

Forcing a rank K approximation

```
levels = 0.1:0.1:0.9;
for K = 1:9
 contour(chebfun2(ff,K),levels)
 xlim([-1 1]), axis equal
end
```



Low-rank, regularity and separability

We can approximate a function of two variables as the sum of separable functions

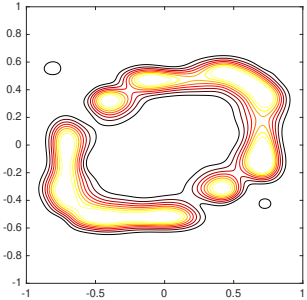
$$f(x,y) = \sum_{k=1}^{K} f_k T_k(x) T_k(y), \quad \{T_k(\cdot)\}_k \text{ Čebyšëv polynomials.}$$

Example (Using `Chebfun` (Driscoll, Hale, and Trefethen 2014))

Consider $f(x,y) = \exp(-40(x^2 - xy + 2y^2 - 1/2)^2)$.

Forcing a rank K approximation

```
levels = 0.1:0.1:0.9;
for K = 1:9
 contour(chebfun2(ff,K),levels)
 xlim([-1 1]), axis equal
end
```



Low-rank, regularity and separability

We can approximate a function of two variables as the sum of separable functions

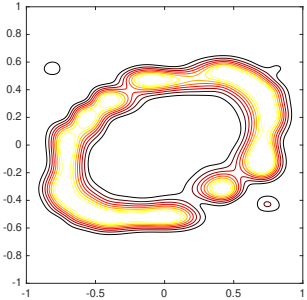
$$f(x,y) = \sum_{k=1}^{K} f_k T_k(x) T_k(y), \quad \{T_k(\cdot)\}_k \text{ Čebyšëv polynomials.}$$

Example (Using `Chebfun` (Driscoll, Hale, and Trefethen 2014))

Consider $f(x,y) = \exp(-40(x^2 - xy + 2y^2 - 1/2)^2)$.

Forcing a rank K approximation

```
levels = 0.1:0.1:0.9;
for K = 1:9
 contour(chebfun2(ff,K),levels)
 xlim([-1 1]), axis equal
end
```



Low-rank, regularity and separability

We can approximate a function of two variables as the sum of separable functions

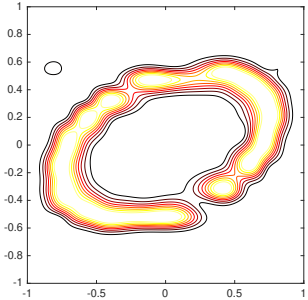
$$f(x,y) = \sum_{k=1}^{K} f_k T_k(x) T_k(y), \quad \{T_k(\cdot)\}_k \text{ Čebyšëv polynomials.}$$

Example (Using `Chebfun` (Driscoll, Hale, and Trefethen 2014))

Consider $f(x,y) = \exp(-40(x^2 - xy + 2y^2 - 1/2)^2)$.

Forcing a rank K approximation

```
levels = 0.1:0.1:0.9;
for K = 1:9
 contour(chebfun2(ff,K),levels)
 xlim([-1 1]), axis equal
end
```



Low-rank, regularity and separability

We can approximate a function of two variables as the sum of separable functions

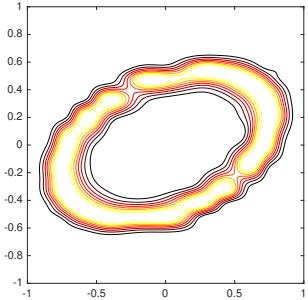
$$f(x,y) = \sum_{k=1}^{K} f_k T_k(x) T_k(y), \quad \{T_k(\cdot)\}_k \text{ Čebyšëv polynomials.}$$

Example (Using `Chebfun` (Driscoll, Hale, and Trefethen 2014))

Consider $f(x,y) = \exp(-40(x^2 - xy + 2y^2 - 1/2)^2)$.

Forcing a rank K approximation

```
levels = 0.1:0.1:0.9;
for K = 1:9
 contour(chebfun2(ff,K),levels)
 xlim([-1 1]), axis equal
end
```



Low-rank, regularity and separability

We can approximate a function of two variables as the sum of separable functions

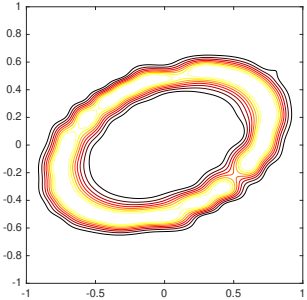
$$f(x,y) = \sum_{k=1}^{K} f_k T_k(x) T_k(y), \quad \{T_k(\cdot)\}_k \text{ Čebyšëv polynomials.}$$

Example (Using `Chebfun` (Driscoll, Hale, and Trefethen 2014))

Consider $f(x,y) = \exp(-40(x^2 - xy + 2y^2 - 1/2)^2)$.

Forcing a rank K approximation

```
levels = 0.1:0.1:0.9;
for K = 1:9
 contour(chebfun2(ff,K),levels)
 xlim([-1 1]), axis equal
end
```



Low-rank, regularity and separability

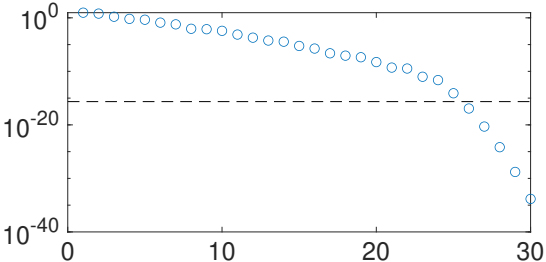
We can approximate a function of two variables as the sum of separable functions

$$f(x,y) = \sum_{k=1}^{K} f_k T_k(x) T_k(y), \quad \{T_k(\cdot)\}_k \text{ Čebyšëv polynomials.}$$

Example (Using `Chebfun` (Driscoll, Hale, and Trefethen 2014))

Consider $f(x,y) = \exp(-40(x^2 - xy + 2y^2 - 1/2)^2)$.

$$F = (f(x_i, x_j))_{i,j} \qquad \mathrm{rank}(F) =$$



$\approx 25.$

Low-rank, regularity and separability

- Not every right-hand side will have a small-enough rank!

Low-rank, regularity and separability

- Not every right-hand side will have a small-enough rank!
- Whenever we have **closed form expression** of the **involved functions** we can work with polynomial basis expansion to discover the rank.

Low-rank, regularity and separability

- Not every right-hand side will have a small-enough rank!
- Whenever we have **closed form expression** of the **involved functions** we can work with polynomial basis expansion to discover the rank.
- To actually compute the decomposition $C = C_1 C_2^H$ we need we can either

Low-rank, regularity and separability

- Not every right-hand side will have a small-enough rank!
- Whenever we have **closed form expression** of the **involved functions** we can work with polynomial basis expansion to discover the rank.
- To actually compute the decomposition $C = C_1 C_2^H$ we need we can either
  - assemble everything and use the **SVD**,

Low-rank, regularity and separability

- Not every right-hand side will have a small-enough rank!
- Whenever we have **closed form expression** of the **involved functions** we can work with polynomial basis expansion to discover the rank.
- To actually compute the decomposition $C = C_1 C_2^H$ we need we can either
  - assemble everything and use the **SVD**,
  - work with **polynomial expansion** and truncate it for small enough coefficients, e.g., (Beckermann and Townsend 2019; Carvajal, Chapman, and Geddes 2005; Townsend and Trefethen 2013),

Low-rank, regularity and separability

- Not every right-hand side will have a small-enough rank!
- Whenever we have **closed form expression** of the **involved functions** we can work with polynomial basis expansion to discover the rank.
- To actually compute the decomposition $C = C_1 C_2^H$ we need we can either
  - assemble everything and use the **SVD**,
  - work with **polynomial expansion** and truncate it for small enough coefficients, e.g., (Beckermann and Townsend 2019; Carvajal, Chapman, and Geddes 2005; Townsend and Trefethen 2013),
  - using algorithm that only need to compute *few* entries of A, such as Adaptive-Cross-Approximation (Tyrtyshnikov 2000), or RandSVD-type algorithms (Halko, Martinsson, and Tropp 2011).

Low-rank, regularity and separability

- Not every right-hand side will have a small-enough rank!
- Whenever we have **closed form expression** of the **involved functions** we can work with polynomial basis expansion to discover the rank.
- To actually compute the decomposition $C = C_1 C_2^H$ we need we can either
  - assemble everything and use the **SVD**,
  - work with **polynomial expansion** and truncate it for small enough coefficients, e.g., (Beckermann and Townsend 2019; Carvajal, Chapman, and Geddes 2005; Townsend and Trefethen 2013),
  - using algorithm that only need to compute *few* entries of A, such as Adaptive-Cross-Approximation (Tyrtyshnikov 2000), or RandSVD-type algorithms (Halko, Martinsson, and Tropp 2011).

Approximating approximating we could get where we wanted...

Let us remember that the approximation of the low-rank term must be done together with the approximation induced by the FDE solution method. We may not need to go as far as machine precision.

Selecting the Krylov subspace

If we are now in the case of a **low rank** right-hand side, we have to select Krylov subspaces for the spaces $\mathcal{V}$ and $\mathcal{W}$.

- From the work we have done in the last couple of lectures, we know how to solve linear systems involving discretization of 1D problems,

Selecting the Krylov subspace

If we are now in the case of a **low rank** right-hand side, we have to select Krylov subspaces for the spaces $\mathcal{V}$ and $\mathcal{W}$.

- From the work we have done in the last couple of lectures, we know how to solve linear systems involving discretization of 1D problems,
- Rational (block) Krylov subspace can therefore be a good choice!

$$\mathcal{V} = \mathrm{range}\{[(A+\sigma_1 I)^{-1}C_1, (A+\sigma_2 I)^{-1}(A+\sigma_1 I)^{-1}C_1, \ldots]\},$$
$$\mathcal{W} = \mathrm{range}\{[(B^H+\eta_1 I)^{-1}C_2, (B^H+\eta_2 I)^{-1}(B^H+\eta_1 I)^{-1}C_2, \ldots]\},$$

Selecting the Krylov subspace

If we are now in the case of a **low rank** right-hand side, we have to select Krylov subspaces for the spaces $\mathcal{V}$ and $\mathcal{W}$.

- From the work we have done in the last couple of lectures, we know how to solve linear systems involving discretization of 1D problems,
- Rational (block) Krylov subspace can therefore be a good choice!

$$\begin{aligned}\mathcal{V} &= \mathrm{range}\{[(A+\sigma_1 I)^{-1}C_1, (A+\sigma_2 I)^{-1}(A+\sigma_1 I)^{-1}C_1, \ldots]\},\\ \mathcal{W} &= \mathrm{range}\{[(B^H+\eta_1 I)^{-1}C_2, (B^H+\eta_2 I)^{-1}(B^H+\eta_1 I)^{-1}C_2, \ldots]\},\end{aligned}$$

- ... but how do we **select the poles**?

Selecting the Krylov subspace

If we are now in the case of a **low rank** right-hand side, we have to select Krylov subspaces for the spaces $\mathcal{V}$ and $\mathcal{W}$.

- From the work we have done in the last couple of lectures, we know how to solve linear systems involving discretization of 1D problems,
- Rational (block) Krylov subspace can therefore be a good choice!

$$\mathcal{V} = \mathrm{range}\{[(A+\sigma_1 I)^{-1}C_1, (A+\sigma_2 I)^{-1}(A+\sigma_1 I)^{-1}C_1, \ldots]\},$$
$$\mathcal{W} = \mathrm{range}\{[(B^H+\eta_1 I)^{-1}C_2, (B^H+\eta_2 I)^{-1}(B^H+\eta_1 I)^{-1}C_2, \ldots]\},$$

- ... but how do we **select the poles**?
- This is not an easy problem in general! A maybe lazy (but surprisingly well behaving) choice is to set $\{\sigma_i, \eta_i\} \in \{0, \infty\} \Rightarrow$ if we choose the two values alternately, then we get the Extended Krylov Subspace.

The Extended Krylov Subspace approach

If $B = A^T$ and $C = C_1 C_2^T$ with $C_1 = C_2$, we can generate the space:

$$\mathbb{EK}(A, C_1) = \mathrm{range}([C_1, A^{-1}C_1, AC_1, A^{-2}C_1, A^2C_1, \ldots]) = \mathcal{V} = \mathcal{W}.$$

The resulting algorithm is the KPIK method by (Simoncini 2007), and can be easily extended to solve the general case, by building both

$$\mathcal{V} = \mathbb{EK}(A, C_1) = \mathrm{range}([C_1, A^{-1}C_1, AC_1, A^{-2}C_1, A^2C_1, \ldots]),$$
$$\mathcal{W} = \mathbb{EK}(B^T, C_2) = \mathrm{range}([C_2, B^{-T}C_2, B^TC_2, A^{-2T}C_2, A^{2T}C_2, \ldots]).$$

The Extended Krylov Subspace approach

If $B = A^T$ and $C = C_1 C_2^T$ with $C_1 = C_2$, we can generate the space:

$$\mathbb{EK}(A, C_1) = \mathrm{range}([C_1, A^{-1}C_1, AC_1, A^{-2}C_1, A^2C_1, \ldots]) = \mathcal{V} = \mathcal{W}.$$

The resulting algorithm is the KPIK method by (Simoncini 2007), and can be easily extended to solve the general case, by building both

$$\mathcal{V} = \mathbb{EK}(A, C_1) = \mathrm{range}([C_1, A^{-1}C_1, AC_1, A^{-2}C_1, A^2C_1, \ldots]),$$
$$\mathcal{W} = \mathbb{EK}(B^T, C_2) = \mathrm{range}([C_2, B^{-T}C_2, B^TC_2, A^{-2T}C_2, A^{2T}C_2, \ldots]).$$

For our two problems, we have to solve systems and do mat-vec with matrices

$$\text{1D: } A = \frac{-\Delta t}{h_N^\alpha}(\theta G_N + (1-\theta)G_N^T) \qquad B = T_M(1 - e^{i\theta})$$

$$\text{2D: } A = \frac{1}{2}I_{N_x} - \frac{\Delta t}{h_{N_x}^\alpha}(\theta G_{N_x} + (1-\theta)G_{N_x}^T) \qquad B = \frac{1}{2}I_{N_y} - \frac{\Delta t}{h_{N_y}^\alpha}(\theta G_{N_y} + (1-\theta)G_{N_y}^T)$$

A couple of examples - I

Let us start from the 1D+1D case

$$\begin{cases} \frac{\partial W}{\partial t} = \Gamma(3-\alpha)x^{\alpha}\,{}^{RL}D^{\alpha}_{[0,x]}W + \Gamma(3-\alpha)(2-x)^{\alpha}{}^{RL}D^{\alpha}_{[x,2]}W - x(x-2)e^{-t}, \\ W(0,t) = W(1,t) = 0, \quad W(x,0) = 5x(2-x); \end{cases}$$

We can **discretize it** in the usual way:

```
w0 = @(x) 5*x.*(2-x);
hN = 2/(N-1); x = 0:hN:2;
dt = hN; t = 0:dt:1; M = length(t);
dplus=@(x,t)gamma(3-alpha).*x.^alpha;
dmin=@(x,t)gamma(3-alpha).*(2-x).^alpha;
f= @(x,t) -x.*(x-2).*exp(-t);
G = glmatrix(N,alpha);
Gr = G; Grt = G.';
Dplus = diag(dplus(x,0));
Dminus = diag(dmin(x,0));
I = eye(N,N); e = ones(N,1);
```

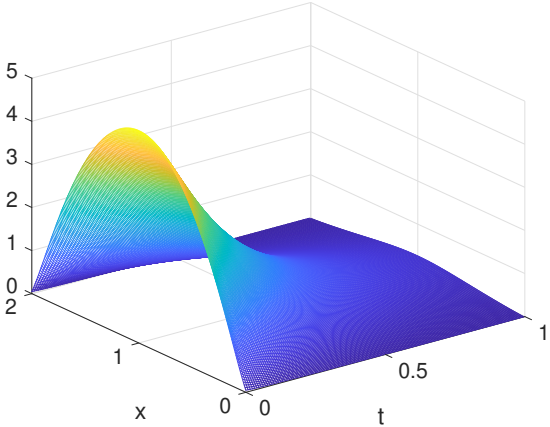
```
A = -dt*(Dplus*Gr +
↪  Dminus*Grt)/hN^alpha;
B = spdiags([-e,e],-1:0,M,M);
[X,T] = meshgrid(x,t);
C = dt*f(X,T);
C(1,:) = w0(x) + C(1,:);
C = -C';
[U,S,V] = svd(C);
C1 = U(:,1:2)*sqrt(S(1:2,1:2));
C2 = (sqrt(S(1:2,1:2))*
↪  V(:,1:2).').';
```

A couple of examples - I

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!



$N = 2^8$, $M = 2^7$, $\alpha = 1.5$

A couple of examples - I

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

α	$N = 2M$	IT	Rel. Residual
1.2	2^5	7	4.982093e-10
	2^6	11	7.629176e-11
	2^7	15	3.721767e-10
	2^8	21	2.406077e-10
	2^9	28	4.726518e-10
	2^{10}	37	8.250742e-10
	2^{11}	50	5.928325e-10

A couple of examples - I

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

α	$N = 2M$	IT	Rel. Residual
1.3	2^5	8	7.473189e-41
	2^6	10	3.324155e-10
	2^7	14	1.876221e-10
	2^8	18	6.104754e-10
	2^9	24	4.098504e-10
	2^{10}	31	5.142375e-10
	2^{11}	40	6.702602e-10

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

A couple of examples - I

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

α	$N = 2M$	IT	Rel. Residual
1.4	2^5	7	4.900654e-10
	2^6	10	4.402728e-11
	2^7	13	1.970841e-10
	2^8	17	2.024635e-10
	2^9	22	5.120085e-10
	2^{10}	28	8.263324e-10
	2^{11}	36	8.596848e-10

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

A couple of examples - I

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

α	$N = 2M$	IT	Rel. Residual
1.5	2^5	7	1.235969e-10
	2^6	9	2.799035e-10
	2^7	13	1.007848e-10
	2^8	16	6.145733e-10
	2^9	21	7.639171e-10
	2^{10}	27	5.857467e-10
	2^{11}	34	8.065585e-10

A couple of examples - I

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

α	$N = 2M$	IT	Rel. Residual
1.6	2^5	7	2.480357e-11
	2^6	9	8.683894e-11
	2^7	13	7.692141e-11
	2^8	16	3.792143e-10
	2^9	21	3.991222e-10
	2^{10}	26	6.017048e-10
	2^{11}	33	6.133773e-10

A couple of examples - I

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

α	$N = 2M$	IT	Rel. Residual
1.7	2^5	7	5.588528e-12
	2^6	8	6.692127e-10
	2^7	12	8.189936e-10
	2^8	16	3.403250e-10
	2^9	20	9.093120e-10
	2^{10}	26	3.550244e-10
	2^{11}	32	7.478792e-10

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

A couple of examples - I

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

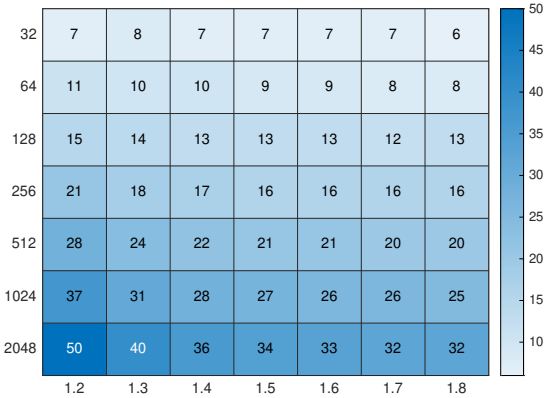
- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

α	$N = 2M$	IT	Rel. Residual
1.8	2^5	6	6.097527e-10
	2^6	8	9.737670e-11
	2^7	13	6.202872e-11
	2^8	16	2.193864e-10
	2^9	20	7.469866e-10
	2^{10}	25	8.191797e-10
	2^{11}	32	5.086938e-10

A couple of examples - I

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```



- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

A couple of examples - II

We can then try the 1D+2D case

$$\begin{cases} \frac{\partial W}{\partial t} = \Gamma(3-\alpha)x^{\alpha}\,{}^{RL}D^{\alpha}_{[0,x]}W + \Gamma(3-\alpha)(2-x)^{\alpha}{}^{RL}D^{\alpha}_{[x,2]}W \\ \qquad + \Gamma(3-\alpha)y^{\alpha}\,{}^{RL}D^{\alpha}_{[0,y]}W + \Gamma(3-\alpha)(2-y)^{\alpha}{}^{RL}D^{\alpha}_{[y,2]}W \\ \qquad + \sin(\pi x)\sin(\pi y)e^{-t}, \\ W(x,y,t) = 0, & (x,y) \in \partial[0,2]^2, \\ W(x,y,0) = 5x(2-x)y(2-y), \end{cases}$$

for which the discretization proceeds along the usual lines, i.e,

```
hN = 2/(N-1); x = 0:hN:2; y = 0:hN:2; [X,Y] = meshgrid(x,y);
dt = hN; t = 0:dt:1; M = length(t);
w0 = @(x,y) 5*x.*(2-x).*y.*(2-y);
dplus = @(x,t) gamma(3-alpha).*x.^alpha;
dminus = @(x,t) gamma(3-alpha).*(2-x).^alpha;
f = @(x,y,t) sin(pi*x).*sin(pi*y).*exp(-t);
```

A couple of examples - II

We can then try the 1D+2D case

$$\begin{cases} \frac{\partial W}{\partial t} = \Gamma(3-\alpha)x^{\alpha}\,{}^{RL}D^{\alpha}_{[0,x]}W + \Gamma(3-\alpha)(2-x)^{\alpha}{}^{RL}D^{\alpha}_{[x,2]}W \\ \qquad + \Gamma(3-\alpha)y^{\alpha}\,{}^{RL}D^{\alpha}_{[0,y]}W + \Gamma(3-\alpha)(2-y)^{\alpha}{}^{RL}D^{\alpha}_{[y,2]}W \\ \qquad + \sin(\pi x)\sin(\pi y)e^{-t}, \\ W(x,y,t) = 0, & (x,y) \in \partial[0,2]^2, \\ W(x,y,0) = 5x(2-x)y(2-y), \end{cases}$$

for which the discretization proceeds along the usual lines, i.e,

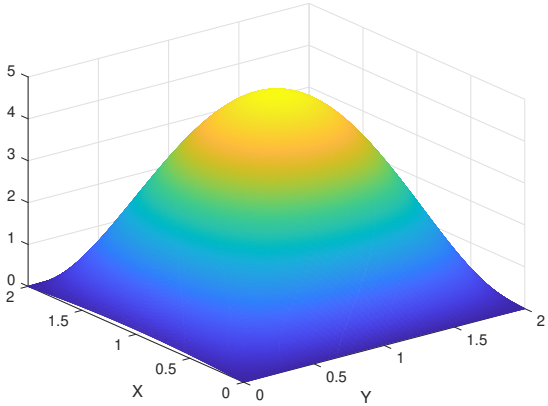
```
G = glmatrix(N,alpha); Gr = G; Grt = G.';
Dplus = diag(dplus(x,0)); Dminus = diag(dminus(x,0));
I = eye(N,N); e = ones(N,1);
A = 0.5*I -dt*(Dplus*Gr + Dminus*Grt)/hN^alpha; % Left-hand side
B = (0.5*I -dt*(Dplus*Gr + Dminus*Grt)/hN^alpha).';
C = w0(X,Y) + dt*f(X,Y,t(1)); C = -C'; [U,S,V] = svd(C);  % Right-hand side
C1 = U(:,1:2)*sqrt(S(1:2,1:2)); C2 = (sqrt(S(1:2,1:2))*V(:,1:2).').';
```

A couple of examples - II

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!



$N = 2^8,\ M = 2^8,\ \alpha = 1.5$

A couple of examples - II

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

α	$N = M$	IT	Rel. Residual
1.2	2^5	7	8.572314e-12
	2^6	9	1.035235e-10
	2^7	10	6.376925e-10
	2^8	11	4.294848e-10
	2^9	11	4.831316e-10
	2^{10}	11	3.340377e-10
	2^{11}	10	8.493637e-10

A couple of examples - II

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

α	$N = M$	IT	Rel. Residual
1.3	2^5	7	7.117681e-11
	2^6	9	7.410001e-11
	2^7	10	6.311608e-10
	2^8	11	6.629092e-10
	2^9	11	7.935697e-10
	2^{10}	11	5.256769e-10
	2^{11}	11	3.021361e-10

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

A couple of examples - II

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

α	$N = M$	IT	Rel. Residual
1.4	2^5	7	6.199844e-11
	2^6	9	5.440959e-11
	2^7	10	6.223106e-10
	2^8	12	2.743756e-10
	2^9	12	6.270319e-10
	2^{10}	12	4.310692e-10
	2^{11}	11	4.849822e-10

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

A couple of examples - II

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

α	$N = M$	IT	Rel. Residual
1.5	2^5	7	5.108938e-11
	2^6	8	7.696608e-10
	2^7	10	5.554438e-10
	2^8	12	3.501633e-10
	2^9	13	4.696907e-10
	2^{10}	13	5.839644e-10
	2^{11}	12	6.172378e-10

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

A couple of examples - II

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

α	$N = M$	IT	Rel. Residual
1.6	2^5	7	4.147318e-11
	2^6	9	1.120891e-10
	2^7	10	4.652358e-10
	2^8	12	3.624143e-10
	2^9	13	6.835564e-10
	2^{10}	14	5.920602e-10
	2^{11}	13	8.882506e-10

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

A couple of examples - II

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

α	$N = M$	IT	Rel. Residual
1.7	2^5	7	3.321348e-11
	2^6	9	9.437180e-11
	2^7	10	7.551800e-10
	2^8	12	3.268160e-10
	2^9	13	7.715645e-10
	2^{10}	14	8.954668e-10
	2^{11}	15	5.806398e-10

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

A couple of examples - II

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```

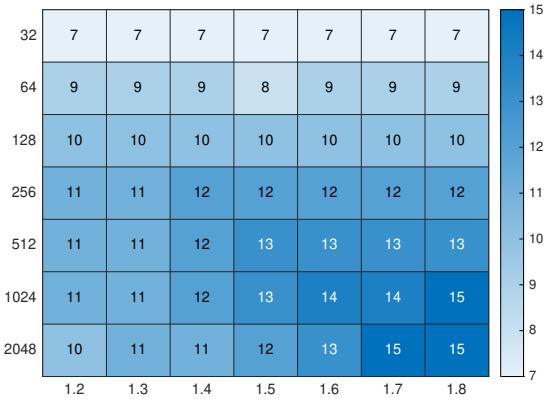
α	$N = M$	IT	Rel. Residual
1.8	2^5	7	2.639521e-11
	2^6	9	7.654578e-11
	2^7	10	6.909946e-10
	2^8	12	4.424195e-10
	2^9	13	7.255110e-10
	2^{10}	15	4.728355e-10
	2^{11}	15	8.400505e-10

- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

A couple of examples - II

And then use the `kpik_sylv` solver from V. Simoncini's software:

```
m = 100;
tol = 1e-9;
[LA,UA] = lu(A); % Direct solutions!
[LB,UB] = lu(B);
[X1,X2,res]=kpik_sylv(A,LA,UA,
↪  B,LB,UB,C1,C2,m,tol);
SOL = X1*X2'; % Not clever al all!
```



- We are using LU-factorization and direct solutions;
- We are reassembling the solution!

Convergence

❓ What can we say about the convergence?

Convergence

❓ What can we say about the convergence?

⚙ If A is symmetric and positive definite, and $B = A^T$, i.e., we are solving a Lyapunov equation, and using **polynomial Krylov subspace**:

Theorem (Simoncini and Druskin 2009, Proposition 3.1)

Let A be symmetric and positive definite, and let $\lambda_{\min}$ be the smallest eigenvalue of A. Let $\hat{\lambda}_{\min}$, $\hat{\lambda}_{\max}$ be the extreme eigenvalue of $A + \lambda_{\min} I$ and $\hat{\kappa} = \hat{\lambda}_{\max}/\hat{\lambda}_{\min}$. Then

$$\|X - X_m\| \leq 4 \frac{\sqrt{\hat{\kappa}} + 1}{\hat{\lambda}_{\min}\sqrt{\hat{\kappa}}} \left(\frac{\sqrt{\hat{\kappa}} - 1}{\sqrt{\hat{\kappa}} + 1} \right)^m .$$

Convergence

❓ What can we say about the convergence?

⚙ If A is symmetric and positive definite, and $B = A^T$, i.e., we are solving a Lyapunov equation, and using **polynomial Krylov subspace**:

Theorem (Simoncini and Druskin 2009, Proposition 3.1)

Let A be symmetric and positive definite, and let $\lambda_{\min}$ be the smallest eigenvalue of A. Let $\hat{\lambda}_{\min}$, $\hat{\lambda}_{\max}$ be the extreme eigenvalue of $A + \lambda_{\min} I$ and $\hat{\kappa} = \hat{\lambda}_{\max}/\hat{\lambda}_{\min}$. Then

$$\|X - X_m\| \leq 4 \frac{\sqrt{\hat{\kappa}} + 1}{\hat{\lambda}_{\min}\sqrt{\hat{\kappa}}} \left(\frac{\sqrt{\hat{\kappa}} - 1}{\sqrt{\hat{\kappa}} + 1} \right)^m .$$

⚠ If $B = A^T$ but A is **no longer symmetric**, one then needs again bounds related to the Field-of-Values of A, see (Simoncini and Druskin 2009).

Convergence

If we have $B \neq A^T$ things are more involved and due to (Beckermann 2011), and we need preliminary work.

- First of all, we need a more manageable expression of the rational Krylov subspace, let us re-brand the poles in the extended complex plane $\overline{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ as

$$z_{A,1}, \ldots, z_{A,m} \in \overline{\mathbb{C}} \setminus \Lambda(A), \qquad z_{B,1}, \ldots, z_{B,n} \in \overline{\mathbb{C}} \setminus \Lambda(B),$$

  and introduce the polynomials

$$Q_A(z) = \prod_{\substack{j=1 \\ z_{A,j} \neq \infty}}^{m} (z - z_{A,j}) \text{ and } Q_B(z) = \prod_{\substack{j=1 \\ z_{B,j} \neq \infty}}^{n} (z - z_{A,j}).$$

Convergence

If we have $B \neq A^T$ things are more involved and due to (Beckermann 2011), and we need preliminary work.

- First of all, we need a more manageable expression of the rational Krylov subspace, let us re-brand the poles in the extended complex plane $\overline{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ as

$$z_{A,1}, \ldots, z_{A,m} \in \overline{\mathbb{C}} \setminus \Lambda(A), \qquad z_{B,1}, \ldots, z_{B,n} \in \overline{\mathbb{C}} \setminus \Lambda(B),$$

  and introduce the polynomials

$$Q_A(z) = \prod_{\substack{j=1 \\ z_{A,j} \neq \infty}}^{m} (z - z_{A,j}) \text{ and } Q_B(z) = \prod_{\substack{j=1 \\ z_{B,j} \neq \infty}}^{n} (z - z_{A,j}).$$

- The two rational spaces can then be written as

$$\mathcal{V} = \{R_A(A)C_1 \,:\, R_A \in \mathbb{P}_{m-1}/Q_A\}, \qquad \mathcal{W} = \{R_B(B)^H C_2 \,:\, R_B \in \mathbb{P}_{n-1}/Q_B\}.$$

Convergence

- Consider the **rational functions** for the projected matrices A_m and B_n on $\mathcal{V}$ and $\mathcal{W}$

$$R_A^G(z) = \frac{\det(zI - A)}{Q_A(z)} \in \mathbb{P}_m/Q_A, \qquad R_B^G(z) = \frac{\det(zI - B_n)}{Q_B(z)} \in \mathbb{P}_n/Q_B$$

Convergence

- Consider the **rational functions** for the projected matrices A_m and B_n on $\mathcal{V}$ and $\mathcal{W}$

$$R_A^G(z) = \frac{\det(zI - A)}{Q_A(z)} \in \mathbb{P}_m/Q_A, \qquad R_B^G(z) = \frac{\det(zI - B_n)}{Q_B(z)} \in \mathbb{P}_n/Q_B$$

Theorem (Beckermann 2011, Theorem 2.1)

Let $\operatorname{rank}(C) = 1$. The rational Galerkin residual ρ can be decomposed into the sum

$$\rho = \rho_{1,2} + \rho_{2,1} + \rho_{2,2}, \qquad \|\rho\|_F^2 = \|\rho_{1,2}\|_F^2 + \|\rho_{2,1}\|_F^2 + \|\rho_{2,2}\|_F^2,$$

with, $C_{1,m} = U^H C_1$, $C_{2,n} = V^H C_2$, and

$$\rho_{1,2} U \frac{1}{R_B^G}(A_m) C_{1,m} C_2^H R_B^G(B), \quad \rho_{2,1} = R_A^G(A) C_1 C_{2,n}^H \frac{1}{R_A^G}(B_n) V^H,$$

$$\rho_{2,2} = \frac{R_A^G(A) C_1 C_2^H R_B^G(B)}{R_A^G(\infty) R_B^G(\infty)}.$$

Convergence

Consider the **rational functions** for the projected matrices A_m and B_n on $\mathcal{V}$ and $\mathcal{W}$

$$R_A^G(z) = \frac{\det(zI - A)}{Q_A(z)} \in \mathbb{P}_m/Q_A, \qquad R_B^G(z) = \frac{\det(zI - B_n)}{Q_B(z)} \in \mathbb{P}_n/Q_B$$

Theorem (Beckermann 2011, Theorem 2.1)

Let $\operatorname{rank}(C) = 1$. The rational Galerkin residual ρ can be decomposed into the sum

$$\rho = \rho_{1,2} + \rho_{2,1} + \rho_{2,2}, \qquad \|\rho\|_F^2 = \|\rho_{1,2}\|_F^2 + \|\rho_{2,1}\|_F^2 + \|\rho_{2,2}\|_F^2,$$

with, $C_{1,m} = U^H C_1$, $C_{2,n} = V^H C_2$, and

$$\|\rho_{2,2}\|_F = \inf_{\substack{R_A \in \mathbb{P}_m/Q_A \\ R_B \in \mathbb{P}_n/Q_B}} \left\| \frac{R_A(A) C_1 C_2^H R_B(B)}{R_A(\infty) R_B(\infty)} \right\|_F = \|(I - UU^H) C_1 C_2^H (I - VV^H)\|_F,$$

Convergence

- Consider the **rational functions** for the projected matrices A_m and B_n on $\mathcal{V}$ and $\mathcal{W}$

$$R_A^G(z) = \frac{\det(zI - A)}{Q_A(z)} \in \mathbb{P}_m/Q_A, \qquad R_B^G(z) = \frac{\det(zI - B_n)}{Q_B(z)} \in \mathbb{P}_n/Q_B$$

Theorem (Beckermann 2011, Theorem 2.1)

Let $\operatorname{rank}(C) = 1$. The rational Galerkin residual ρ can be decomposed into the sum

$$\rho = \rho_{1,2} + \rho_{2,1} + \rho_{2,2}, \qquad \|\rho\|_F^2 = \|\rho_{1,2}\|_F^2 + \|\rho_{2,1}\|_F^2 + \|\rho_{2,2}\|_F^2,$$

with, $C_{1,m} = U^H C_1$, $C_{2,n} = V^H C_2$, and

$$\|\rho_{1,2}\|_F = \min_{R_B \in \mathbb{P}_m/Q_B} \left[\|R_B(A_m) C_{1,m} C_2^H R_B(B)\|_F + c_0 \|\frac{1}{R_B}(A_m) C_{1,m} C_{2,n}^H R_B(B_n)\|_F \right],$$

for $c_0 = 2\operatorname{diam}(W(A), W(B))/\operatorname{dist}(W(A), W(B))$.

Convergence

- Consider the **rational functions** for the projected matrices A_m and B_n on $\mathcal{V}$ and $\mathcal{W}$

$$R_A^G(z) = \frac{\det(zI - A)}{Q_A(z)} \in \mathbb{P}_m/Q_A, \qquad R_B^G(z) = \frac{\det(zI - B_n)}{Q_B(z)} \in \mathbb{P}_n/Q_B$$

Theorem (Beckermann 2011, Theorem 2.1)

Let $\operatorname{rank}(C) = 1$. The rational Galerkin residual ρ can be decomposed into the sum

$$\rho = \rho_{1,2} + \rho_{2,1} + \rho_{2,2}, \qquad \|\rho\|_F^2 = \|\rho_{1,2}\|_F^2 + \|\rho_{2,1}\|_F^2 + \|\rho_{2,2}\|_F^2,$$

with, $C_{1,m} = U^H C_1$, $C_{2,n} = V^H C_2$, and

$$\|\rho_{2,1}\|_F = \min_{R_A \in \mathbb{P}_m/Q_A} \left[\|R_A(A) C_1 C_{2,n}^H \frac{1}{R_A}(B_n)\|_F + c_0 \|R_A(A_m) C_{1,m} C_{2,n}^H \frac{1}{R_A}(B_n)\|_F \right],$$

for $c_0 = 2\operatorname{diam}(W(A), W(B))/\operatorname{dist}(W(A), W(B))$.

Convergence

- Consider the **rational functions** for the projected matrices A_m and B_n on $\mathcal{V}$ and $\mathcal{W}$

$$R_A^G(z) = \frac{\det(zI - A)}{Q_A(z)} \in \mathbb{P}_m/Q_A, \qquad R_B^G(z) = \frac{\det(zI - B_n)}{Q_B(z)} \in \mathbb{P}_n/Q_B$$

- Now we have a **representation of the residual** in the **orthogonal bases** associated to the given Krylov subspaces, and furthermore we know that $\rho_{2,2} = 0$ if at least one of the $z_{A,j}$ or $z_{B,j}$ is ∞, i.e., if either of the initial vectors are in the subspace.

Convergence

- Consider the **rational functions** for the projected matrices A_m and B_n on $\mathcal{V}$ and $\mathcal{W}$

$$R_A^G(z) = \frac{\det(zI - A)}{Q_A(z)} \in \mathbb{P}_m/Q_A, \qquad R_B^G(z) = \frac{\det(zI - B_n)}{Q_B(z)} \in \mathbb{P}_n/Q_B$$

- Now we have a **representation of the residual** in the **orthogonal bases** associated to the given Krylov subspaces, and furthermore we know that $\rho_{2,2} = 0$ if at least one of the $z_{A,j}$ or $z_{B,j}$ is ∞, i.e., if either of the initial vectors are in the subspace.
- The bounds are then obtained by having upper-bounds of the quantities

$$E_m(\spadesuit, Q_\spadesuit, z) = \min_{p \in \mathbb{P}_\heartsuit} \frac{\left\| \frac{P}{Q_\spadesuit}(\spadesuit) \right\|}{\left| \frac{P}{Q_\spadesuit}(z) \right|}, \text{ for } \spadesuit = \{A, B\},\ \heartsuit = \{m, n\}.$$

Convergence

- Consider the **rational functions** for the projected matrices A_m and B_n on $\mathcal{V}$ and $\mathcal{W}$

$$R_A^G(z) = \frac{\det(zI - A)}{Q_A(z)} \in \mathbb{P}_m/Q_A, \qquad R_B^G(z) = \frac{\det(zI - B_n)}{Q_B(z)} \in \mathbb{P}_n/Q_B$$

- Now we have a **representation of the residual** in the **orthogonal bases** associated to the given Krylov subspaces, and furthermore we know that $\rho_{2,2} = 0$ if at least one of the $z_{A,j}$ or $z_{B,j}$ is ∞, i.e., if either of the initial vectors are in the subspace.
- The bounds are then obtained by having upper-bounds of the quantities

$$E_m(\spadesuit, Q_\spadesuit, z) = \min_{p \in \mathbb{P}_\heartsuit} \frac{\left\| \frac{P}{Q_\spadesuit}(\spadesuit) \right\|}{\left| \frac{P}{Q_\spadesuit}(z) \right|}, \text{ for } \spadesuit = \{A, B\},\ \heartsuit = \{m, n\}.$$

$\Rightarrow$ This can be faced by using the upper bound given by Crouziex upper-bound for matrix-functions.

Convergence: potential theory

In order to obtain the bounds and the rate of convergence, we need to work with the **Green functions** of $\overline{\mathbb{C}} \setminus W(A)$ and $\overline{\mathbb{C}} \setminus W(B)$ with poles at $\zeta \in \mathbb{C}$ called $g_A(\cdot, \zeta)$ and $g_B(\cdot, \zeta)$ respectively; (Saff and Totik 1997).

Convergence: potential theory

In order to obtain the bounds and the rate of convergence, we need to work with the **Green functions** of $\overline{\mathbb{C}} \setminus W(A)$ and $\overline{\mathbb{C}} \setminus W(B)$ with poles at $\zeta \in \mathbb{C}$ called $g_A(\cdot, \zeta)$ and $g_B(\cdot, \zeta)$ respectively; (Saff and Totik 1997).

With this **potential functions** the bound can then be expressed in terms of the functions

$$u_{A,m}(z) = \exp\left(-\sum_{j=1}^{m} g_A(z, z_{A,j})\right), \text{ and } u_{B,n}(z) = \exp\left(-\sum_{j=1}^{n} g_B(z, z_{B,j})\right).$$

Convergence: potential theory

In order to obtain the bounds and the rate of convergence, we need to work with the **Green functions** of $\overline{\mathbb{C}} \setminus W(A)$ and $\overline{\mathbb{C}} \setminus W(B)$ with poles at $\zeta \in \mathbb{C}$ called $g_A(\cdot, \zeta)$ and $g_B(\cdot, \zeta)$ respectively; (Saff and Totik 1997).

With this **potential functions** the bound can then be expressed in terms of the functions

$$u_{A,m}(z) = \exp\left(-\sum_{j=1}^{m} g_A(z, z_{A,j})\right), \text{ and } u_{B,n}(z) = \exp\left(-\sum_{j=1}^{n} g_B(z, z_{B,j})\right).$$

A mad research idea

Given the case we are interested in, can we find **optimal poles**, i.e., the one minimizing the bounds and have both α robustness, and M and N independence?

Let's blow up the bridges

❓ What do we do if the **space coefficients** are **not separable**?

Let's blow up the bridges

❓ What do we do if the **space coefficients** are **not separable**?

⚙ We decompose

$$d^{\pm}(x,y) = \sum_{k=1}^{K} t_k^{\pm} T_k(x) T_k(y)$$

and substitute in our equation obtaining

$$\sum_{k=1}^{K} \left(\tilde{A}_k X + X \tilde{B}_k^T \right) = C_1 C_2^T .$$

Let's blow up the bridges

What do we do if the **space coefficients** are **not separable**?

We decompose

$$d^{\pm}(x,y) = \sum_{k=1}^{K} t_k^{\pm} T_k(x) T_k(y)$$

and substitute in our equation obtaining

$$\sum_{k=1}^{K} \left(\tilde{A}_k X + X \tilde{B}_k^T \right) = C_1 C_2^T.$$

We can try generalize the Galerkin projection

$$\sum_{k=1}^{2K} \hat{A}_k X \hat{B}_k = C_1 C_2^T \;\Rightarrow\; \sum_{k=1}^{2K} (V_m^T \hat{A}_k V_m) X (W_m^T \hat{B}_k W_m) = V_m C_1 (W_m^T C_2)^T,$$

Let's blow up the bridges

What do we do if the **space coefficients** are **not separable**?

We decompose

$$d^{\pm}(x,y) = \sum_{k=1}^{K} t_k^{\pm} T_k(x) T_k(y)$$

and substitute in our equation obtaining

$$\sum_{k=1}^{K} \left(\tilde{A}_k X + X \tilde{B}_k^T \right) = C_1 C_2^T.$$

We can try generalize the Galerkin projection

$$\sum_{k=1}^{2K} \hat{A}_k X \hat{B}_k = C_1 C_2^T \;\Rightarrow\; \sum_{k=1}^{2K} (V_m^T \hat{A}_k V_m) X (W_m^T \hat{B}_k W_m) = V_m C_1 (W_m^T C_2)^T,$$

How do we select $\mathcal{V}$ and $\mathcal{W}$? How do we generate nested subspace? How do we solve the reduced multiterm equation?

Let's blow up the bridges

❓ What do we do if the **space coefficients** are **not separable**?

⚙ We decompose

$$d^{\pm}(x,y) = \sum_{k=1}^{K} t_k^{\pm} T_k(x) T_k(y)$$

and substitute in our equation obtaining

$$\sum_{k=1}^{K} \left(\tilde{A}_k X + X \tilde{B}_k^T \right) = C_1 C_2^T.$$

🔨 We can try generalize the Galerkin projection

$$\sum_{k=1}^{2K} \hat{A}_k X \hat{B}_k = C_1 C_2^T \;\Rightarrow\; \sum_{k=1}^{2K} (V_m^T \hat{A}_k V_m) X (W_m^T \hat{B}_k W_m) = V_m C_1 (W_m^T C_2)^T,$$

⚙ How do we select $\mathcal{V}$ and $\mathcal{W}$? How do we generate nested subspace? How do we solve the reduced multiterm equation? $\Rightarrow$ many more questions than answers... ☹.

Let's blow up the bridges

❓ What if the **convergence rate** is **poor**?

Let's blow up the bridges

❓ What if the **convergence rate** is **poor**?

Let's blow up the bridges

❓ What if the **convergence rate** is **poor**?

Let's blow up the bridges

❓ What if the **convergence rate** is **poor**?

Since convergence depends on the spectrum, we may be tempted to precondition the equation with a matrix P, i.e.,

$$(P^{-1}AP)P^{-1}XP^{-H} + P^{-1}XP^{-H}(P^{H}BP^{-H}) = P^{-1}CP^{-H},$$

that is of no use since $P^{-1}AP \sim A$ and $P^{-1}BP \sim B$.

❓ Can we use the **Kronecker structure** to put together **1D+2D** case as a **single matrix equation** or, more generally, **1D+dD equations** as a **single matrix equation**?

Let's blow up the bridges

What if the **convergence rate** is **poor**?

Since convergence depends on the spectrum, we may be tempted to precondition the equation with a matrix P, i.e.,

$$(P^{-1}AP)P^{-1}XP^{-H} + P^{-1}XP^{-H}(P^{H}BP^{-H}) = P^{-1}CP^{-H},$$

that is of no use since $P^{-1}AP \sim A$ and $P^{-1}BP \sim B$.

Can we use the **Kronecker structure** to put together **1D+2D** case as a **single matrix equation** or, more generally, **1D+dD equations** as a **single matrix equation**?

This is a whole *different can of worms* and it's called *tensor equations*.

Let's blow up the bridges

❓ What if the **convergence rate** is **poor**?

Since convergence depends on the spectrum, we may be tempted to precondition the equation with a matrix P, i.e.,

$$(P^{-1}AP)P^{-1}XP^{-H} + P^{-1}XP^{-H}(P^{H}BP^{-H}) = P^{-1}CP^{-H},$$

that is of no use since $P^{-1}AP \sim A$ and $P^{-1}BP \sim B$.

$\Rightarrow$ the only possibility is **playing around with the poles**.

❓ Can we use the **Kronecker structure** to put together **1D+2D** case as a **single matrix equation** or, more generally, **1D+dD equations** as a **single matrix equation**?

🔧 This is a whole *different can of worms* and it's called *tensor equations*.

❓ What if the **right-hand side** is **not low rank**?

Let's blow up the bridges

❓ What if the **convergence rate** is **poor**?

Since convergence depends on the spectrum, we may be tempted to precondition the equation with a matrix P, i.e.,

$$(P^{-1}AP)P^{-1}XP^{-H} + P^{-1}XP^{-H}(P^{H}BP^{-H}) = P^{-1}CP^{-H},$$

that is of no use since $P^{-1}AP \sim A$ and $P^{-1}BP \sim B$.

$\Rightarrow$ the only possibility is **playing around with the poles**.

❓ Can we use the **Kronecker structure** to put together **1D+2D** case as a **single matrix equation** or, more generally, **1D+dD equations** as a **single matrix equation**?

🔧 This is a whole *different can of worms* and it's called *tensor equations*.

❓ What if the **right-hand side** is **not low rank**?

🔨 We can use some **approximation strategy**, solve the matrix-equation **incompletely** and use it as a **preconditioner** inside a FGMRES method, or *turn to other structures*...

Rank-structured matrices

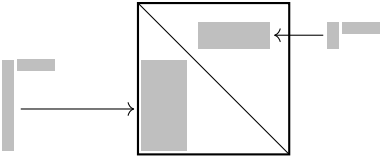
Global low-rank matrices is **not the only setting** in which computations can be spared!

Rank-structured matrices

Global low-rank matrices is **not the only setting** in which computations can be spared!

Quasiseparable matrix

A matrix A is *quasiseparable* of order k if the maximum of the ranks of all its submatrices contained in the strictly upper or lower part is less or equal than k.

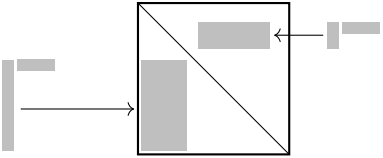


Rank-structured matrices

Global low-rank matrices is **not the only setting** in which computations can be spared!

Quasiseparable matrix

A matrix A is *quasiseparable* of order k if the maximum of the ranks of all its submatrices contained in the strictly upper or lower part is less or equal than k.



Example: k-banded matrices

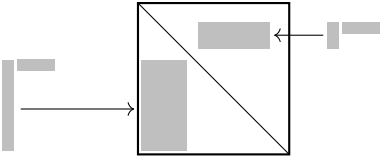
A banded matrix with bandwidth k is quasiseparable of order (at most) k. In particular, diagonal matrices are quasiseparable of order 0, tridiagonal matrices are quasiseparable of order 1, *etc.*

Rank-structured matrices

Global low-rank matrices is **not the only setting** in which computations can be spared!

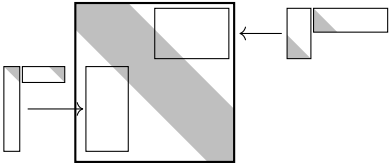
Quasiseparable matrix

A matrix A is *quasiseparable* of order k if the maximum of the ranks of all its submatrices contained in the strictly upper or lower part is less or equal than k.



Example: k-banded matrices

A banded matrix with bandwidth k is quasiseparable of order (at most) k. In particular, diagonal matrices are quasiseparable of order 0, tridiagonal matrices are quasiseparable of order 1, *etc.*



Rank-structured matrices

Theorem (Massei, Palitta, and Robol 2018, Theorem 2.7)

Let A and B be **symmetric positive definite** matrices of **quasiseparable** rank k_A and k_B, respectively, and suppose that the spectra of A and B are both contained in the interval $[a, b]$. Then, if X solves the Sylvester equation $AX + XB = C$, with C of **quasiseparable** rank k_C, a generic off-diagonal block Y of X satisfies

$$\frac{\sigma_{1+k\ell}(Y)}{\sigma_1(Y)} \leq 4\rho^{-2\ell},$$

where $k \triangleq k_A + k_B + k_C$, $\rho = \exp\left(\frac{\pi^2}{2\mu(\frac{b}{a})}\right)$ and $\mu(\cdot)$ the Grötzsch ring function

$$\mu(\lambda) \triangleq \frac{\pi}{2}\frac{K(\sqrt{1-\lambda^2})}{K(\lambda)}, \qquad K(\lambda) \triangleq \int_0^1 \frac{1}{(1-t^2)(1-\lambda^2 t^2)}\,\mathrm{d}t.$$

Rank-structured matrices

Theorem (Massei, Palitta, and Robol 2018, Theorem 2.7)

Let A and B be **symmetric positive definite** matrices of **quasiseparable** rank k_A and k_B, respectively, and suppose that the spectra of A and B are both contained in the interval $[a, b]$. Then, if X solves the Sylvester equation $AX + XB = C$, with C of **quasiseparable** rank k_C, a generic off-diagonal block Y of X satisfies

$$\frac{\sigma_{1+k\ell}(Y)}{\sigma_1(Y)} \leq 4\rho^{-2\ell},$$

where $k \triangleq k_A + k_B + k_C$, $\rho = \exp\left(\frac{\pi^2}{2\mu(\frac{b}{a})}\right)$ and $\mu(\cdot)$ the Grötzsch ring function

$$\mu(\lambda) \triangleq \frac{\pi}{2}\frac{K(\sqrt{1-\lambda^2})}{K(\lambda)}, \qquad K(\lambda) \triangleq \int_0^1 \frac{1}{(1-t^2)(1-\lambda^2 t^2)}\,\mathrm{d}t.$$

⚠ As usual, the **non-symmetric case** requires using the field-of-values!

Rank-structured matrices

Theorem (Massei, Palitta, and Robol 2018, Theorem 2.12)

Let A, B be matrices of quasiseparable rank k_A and k_B respectively and such that $W(A) \subseteq E$ and $W(-B) \subseteq F$. Consider the Sylvester equation $AX + XB = C$, with C of quasiseparable rank k_C. Then a generic off-diagonal block Y of the solution X satisfies

$$\frac{\sigma_{1+k\ell}(Y)}{\sigma_1(Y)} \leq \mathcal{C}^2 \cdot Z_\ell(E, F), \qquad k := k_A + k_B + k_C.$$

Where $Z_\ell(E, F)$ is the solution of the **Zolotarev problem**

$$Z_\ell(E, F) \triangleq \inf_{r(x) \in \mathcal{R}_{\ell,\ell}} \frac{\max_{x \in E} |r(x)|}{\min_{y \in F} |r(y)|}, \qquad \ell \geq 1,$$

for $\mathcal{R}_{\ell,\ell}$ is the set of rational functions of degree at most (ℓ, ℓ), and $\mathcal{C}$ is the Crouzeix universal constant.

The Zolotarev 3rd Problem

Zolotarev's **third problem** is exactly the computation of

$$Z_\ell(E,F) \triangleq \inf_{r(x)\in\mathcal{R}_{\ell,\ell}} \frac{\max_{x\in E}|r(x)|}{\min_{y\in F}|r(y)|}, \qquad \ell \geq 1,$$

for two given sets E, F and a degree ℓ, *informally*:

> *"Find a rational function that is as small as possible on a set E while being ≥ 1 in absolute value on another set F"*

The Zolotarev 3rd Problem

Zolotarev's **third problem** is exactly the computation of

$$Z_\ell(E,F) \triangleq \inf_{r(x)\in\mathcal{R}_{\ell,\ell}} \frac{\max_{x\in E}|r(x)|}{\min_{y\in F}|r(y)|}, \qquad \ell \geq 1,$$

for two given sets E, F and a degree ℓ, *informally*:

> *"Find a rational function that is as small as possible on a set E while being ≥ 1 in absolute value on another set F"*

- For general sets E and F the solution is not explicitly known.

The Zolotarev 3rd Problem

Zolotarev's **third problem** is exactly the computation of

$$Z_\ell(E,F) \triangleq \inf_{r(x)\in\mathcal{R}_{\ell,\ell}} \frac{\max_{x\in E}|r(x)|}{\min_{y\in F}|r(y)|}, \qquad \ell \geq 1,$$

for two given sets E, F and a degree ℓ, *informally*:

> *"Find a rational function that is as small as possible on a set E while being ≥ 1 in absolute value on another set F"*

- For general sets E and F the solution is not explicitly known.
- However, there are cases where a solution is known.

The Zolotarev 3rd Problem

Zolotarev's **third problem** is exactly the computation of

$$Z_\ell(E,F) \triangleq \inf_{r(x)\in\mathcal{R}_{\ell,\ell}} \frac{\max_{x\in E}|r(x)|}{\min_{y\in F}|r(y)|}, \qquad \ell \geq 1,$$

for two given sets E, F and a degree ℓ, *informally*:

> *"Find a rational function that is as small as possible on a set E while being ≥ 1 in absolute value on another set F"*

- For general sets E and F the solution is not explicitly known.
- However, there are cases where a solution is known.

Example: two equal intervals

One can prove that for $E = [-b,-1]$ and $F = [1,b]$ the solution is

$$\sup_{x\in[-b,1]\cup[1,b]} |R(x) - \mathrm{sgn}(x)| = \frac{\sqrt{Z_\ell(E,F)}}{1+Z_\ell(E,F)}$$

The Zolotarev 3rd Problem

Zolotarev's **third problem** is exactly the computation of

$$Z_\ell(E,F) \triangleq \inf_{r(x)\in\mathcal{R}_{\ell,\ell}} \frac{\max_{x\in E}|r(x)|}{\min_{y\in F}|r(y)|}, \qquad \ell \geq 1,$$

for two given sets E, F and a degree ℓ, *informally*:

> *"Find a rational function that is as small as possible on a set E while being ≥ 1 in absolute value on another set F"*

- For general sets E and F the solution is not explicitly known.
- However, there are cases where a solution is known.

Example: two equal intervals

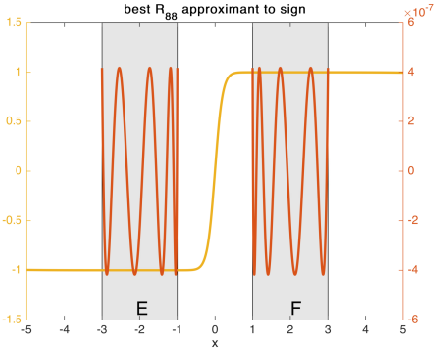
One can prove that for $E = [-b,-1]$ and $F = [1,b]$ the solution is

$$\sup_{x\in[-b,1]\cup[1,b]} |R(x) - \mathrm{sgn}(x)| = \frac{\sqrt{Z_\ell(E,F)}}{1+Z_\ell(E,F)} \Rightarrow \text{This is Zolotarev } 4^{\text{th}} \text{ problem!}$$

The Zolotarev 4th Problem

A closed form solution, involving Jacobi elliptic functions, is available in the RKToolbox

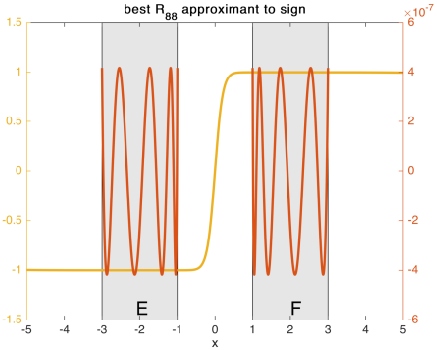
```
b = 3; % E = [-b,-1] and F = [1,b]
k = 8; % Degree of rational approximant to sign.
% Solution to Z's fourth problem:
r = rkfun.gallery('sign', k/2, b);
% Plot the computed rational function:
x = linspace(-5, 5, 1000);
y1 = linspace(-3, -1, 1000);
y2 = linspace(1, 3, 1000);
fill([-b -1 -1 -b -b], 1.5*[-1 -1 1 1 -1], .9*[1 1
↪  1] ),
hold on
fill([b 1 1 b b],1.5*[-1 -1 1 1 -1],.9*[1 1 1] )
[~,l1,l2] = plotyy(x,r(x),[y1 0 y2],[(1-abs(r(y1)))
↪  NaN (1-abs(r(y2)))]);
l1.LineWidth = 2; l2.LineWidth = 2;
hold off
```



The Zolotarev 4th Problem

A closed form solution, involving Jacobi elliptic functions, is available in the RKToolbox

```
b = 3; % E = [-b,-1] and F = [1,b]
k = 8; % Degree of rational approximant to sign.
% Solution to Z's fourth problem:
r = rkfun.gallery('sign', k/2, b);
% Plot the computed rational function:
x = linspace(-5, 5, 1000);
y1 = linspace(-3, -1, 1000);
y2 = linspace(1, 3, 1000);
fill([-b -1 -1 -b -b], 1.5*[-1 -1 1 1 -1], .9*[1 1
↪  1] ),
hold on
fill([b 1 1 b b],1.5*[-1 -1 1 1 -1],.9*[1 1 1] )
[~,l1,l2] = plotyy(x,r(x),[y1 0 y2],[(1-abs(r(y1)))
↪  NaN (1-abs(r(y2)))]);
l1.LineWidth = 2; l2.LineWidth = 2;
hold off
```



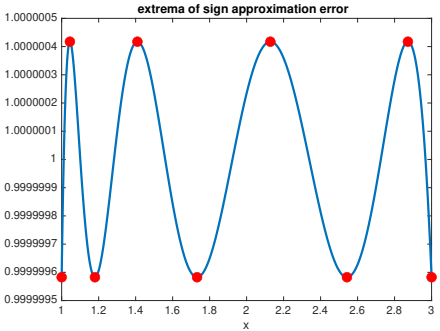
$\Rightarrow$ Zolotarev problem is then solved by solving for $Z_\ell(E, F)$.

The Zolotarev 3rd Problem

Solve for $Z_\ell(E,F)$ s.t. $$\sup_{x\in[-b,1]\cup[1,b]} |R(x)-\mathrm{sgn}(x)| = \frac{\sqrt{Z_\ell(E,F)}}{1+Z_\ell(E,F)}$$

```
% Extrema for [-1,-1/b]\cup [1/b,1]:
K = ellipke(1-1/b^2);
[sn, cn, dn] = ellipj((0:k)*K/k, 1-1/b^2);
% Transplant to [-b,-1]\cup [1,b]:
extrema = b*dn;
vals = 1-r(extrema);
c = mean( vals(1:2:end) );
e = eig( [ 2-4/c^2 1 ; 1 0 ] );
Zk = min(abs(e))
```

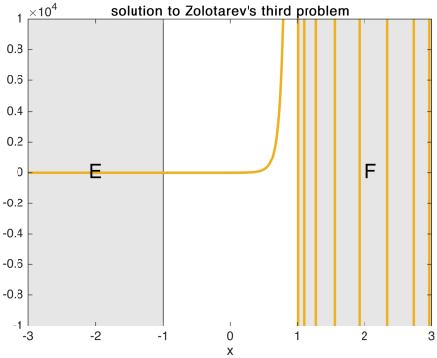
From which we obtain `Zk = 4.3542e-14`.



The Zolotarev 3rd Problem

To visualize the function realizing the extrema, one can use a Mobius transform to convert the best rational approximation to the sgn function that solves the 4th problem $r(x)$ to the extremal rational function $R_{\ell,\ell}(x)$ solving the 3rd:

$$R_{\ell,\ell}(x) = \frac{\frac{1+Z_\ell(E,F)}{(1-Z_\ell(E,F))r(x)}}{\left(1 - \frac{1+Z_\ell(E,F)}{1-Z_\ell(E,F)} r(x)\right)}$$

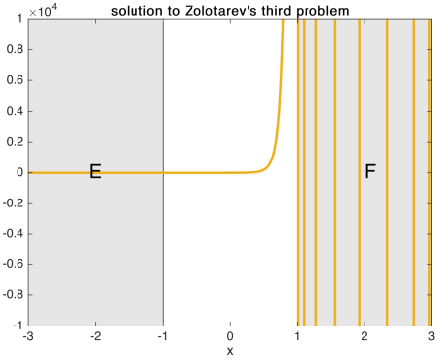


The Zolotarev 3rd Problem

To visualize the function realizing the extrema, one can use a Mobius transform to convert the best rational approximation to the sgn function that solves the 4th problem $r(x)$ to the extremal rational function $R_{\ell,\ell}(x)$ solving the 3rd:

$$R_{\ell,\ell}(x) = \frac{\frac{1+Z_\ell(E,F)}{(1-Z_\ell(E,F))r(x)}}{\left(1 - \frac{1+Z_\ell(E,F)}{1-Z_\ell(E,F)} r(x)\right)}$$

- There are **other cases** for which one can solve the 3rd problem, e.g., *unsymmetrical intervals*, or *rectangles* (Istace and Thiran 1995).

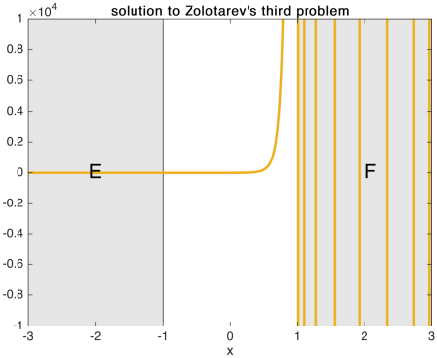


The Zolotarev 3rd Problem

To visualize the function realizing the extrema, one can use a Mobius transform to convert the best rational approximation to the sgn function that solves the 4th problem $r(x)$ to the extremal rational function $R_{\ell,\ell}(x)$ solving the 3rd:

$$R_{\ell,\ell}(x) = \frac{\frac{1+Z_\ell(E,F)}{(1-Z_\ell(E,F))r(x)}}{\left(1 - \frac{1+Z_\ell(E,F)}{1-Z_\ell(E,F)} r(x)\right)}$$

- There are **other cases** for which one can solve the 3rd problem, e.g., *unsymmetrical intervals*, or *rectangles* (Istace and Thiran 1995).
- If we are satisfied by the quasi-separability rank of the solution we can then attempt it!



Conclusion and summary

- We have reformulated several of our problems in terms of matrix equations,
- We have discussed projection methods for the solution of Sylvester equations,
- We have seen some limitations of the approach and shown a possible extension.

Next up

- More on rank-structured matrices and related solution strategies,
- All-at-once in time: using different methods to march in time than the standard ones,
- Still some other approaches with structured preconditioners.

Bibliography I

- Bartels, R. H. and G. W. Stewart (Sept. 1972). "Solution of the Matrix Equation $AX + XB = C$ [F4]". In: *Commun. ACM* 15.9, pp. 820–826. ISSN: 0001-0782. DOI: 10.1145/361573.361582. URL: https://doi.org/10.1145/361573.361582.
- Beckermann, B. (2011). "An error analysis for rational Galerkin projection applied to the Sylvester equation". In: *SIAM J. Numer. Anal.* 49.6, pp. 2430–2450. ISSN: 0036-1429. DOI: 10.1137/110824590. URL: https://doi.org/10.1137/110824590.
- Beckermann, B. and A. Townsend (2019). "Bounds on the singular values of matrices with displacement structure". In: *SIAM Rev.* 61.2. Revised reprint of "On the singular values of matrices with displacement structure" [MR3717820], pp. 319–344. ISSN: 0036-1445. DOI: 10.1137/19M1244433. URL: https://doi.org/10.1137/19M1244433.
- Breiten, T., V. Simoncini, and M. Stoll (2016). "Low-rank solvers for fractional differential equations". In: *Electron. Trans. Numer. Anal.* 45, pp. 107–132.
- Carvajal, O. A., F. W. Chapman, and K. O. Geddes (2005). "Hybrid symbolic-numeric integration in multiple dimensions via tensor-product series". In: *Proceedings of the 2005 international symposium on Symbolic and algebraic computation*, pp. 84–91.

Bibliography II

- Driscoll, T. A., N. Hale, and L. N. Trefethen (2014). *Chebfun guide*.
- Gohberg, I., T. Kailath, and V. Olshevsky (1995). "Fast Gaussian elimination with partial pivoting for matrices with displacement structure". In: *Math. Comp.* 64.212, pp. 1557–1576. ISSN: 0025-5718. DOI: 10.2307/2153371. URL: https://doi.org/10.2307/2153371.
- Golub, G., S. Nash, and C. Van Loan (1979). "A Hessenberg-Schur method for the problem AX + XB= C". In: *IEEE Transactions on Automatic Control* 24.6, pp. 909–913. DOI: 10.1109/TAC.1979.1102170.
- Halko, N., P. G. Martinsson, and J. A. Tropp (2011). "Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions". In: *SIAM Rev.* 53.2, pp. 217–288. ISSN: 0036-1445. DOI: 10.1137/090771806. URL: https://doi.org/10.1137/090771806.
- Istace, M.-P. and J.-P. Thiran (1995). "On the third and fourth Zolotarev problems in the complex plane". In: *SIAM J. Numer. Anal.* 32.1, pp. 249–259. ISSN: 0036-1429. DOI: 10.1137/0732009. URL: https://doi.org/10.1137/0732009.

Bibliography III

- Massei, S., D. Palitta, and L. Robol (2018). "Solving rank-structured Sylvester and Lyapunov equations". In: *SIAM J. Matrix Anal. Appl.* 39.4, pp. 1564–1590. ISSN: 0895-4798. DOI: 10.1137/17M1157155. URL: https://doi.org/10.1137/17M1157155.
- Saff, E. B. and V. Totik (1997). *Logarithmic potentials with external fields*. Vol. 316. Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Appendix B by Thomas Bloom. Springer-Verlag, Berlin, pp. xvi+505. ISBN: 3-540-57078-0. DOI: 10.1007/978-3-662-03329-6. URL: https://doi.org/10.1007/978-3-662-03329-6.
- Simoncini, V. (2007). "A new iterative method for solving large-scale Lyapunov matrix equations". In: *SIAM J. Sci. Comput.* 29.3, pp. 1268–1288. ISSN: 1064-8275. DOI: 10.1137/06066120X. URL: https://doi.org/10.1137/06066120X.
- — (2016). "Computational methods for linear matrix equations". In: *SIAM Rev.* 58.3, pp. 377–441. ISSN: 0036-1445. DOI: 10.1137/130912839. URL: https://doi.org/10.1137/130912839.

Bibliography IV

Simoncini, V. and V. Druskin (2009). "Convergence analysis of projection methods for the numerical solution of large Lyapunov equations". In: *SIAM J. Numer. Anal.* 47.2, pp. 828–843. ISSN: 0036-1429. DOI: 10.1137/070699378. URL: https://doi.org/10.1137/070699378.

Townsend, A. and L. N. Trefethen (2013). "An extension of Chebfun to two dimensions". In: *SIAM J. Sci. Comput.* 35.6, pp. C495–C518. ISSN: 1064-8275. DOI: 10.1137/130908002. URL: https://doi.org/10.1137/130908002.

Tyrtyshnikov, E. E. (2000). "Incomplete cross approximation in the mosaic-skeleton method". In: vol. 64. 4. International GAMM-Workshop on Multigrid Methods (Bonn, 1998), pp. 367–380. DOI: 10.1007/s006070070031. URL: https://doi.org/10.1007/s006070070031.

An introduction to fractional calculus

Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

October, 2022

Sylvester with quasiseparable matrices

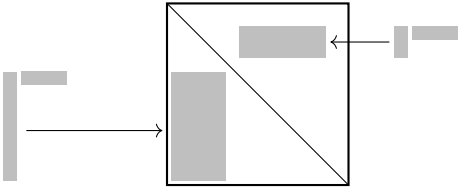
Let's start again from the problem we wanted to solve

$$AX + XB^T = C, \qquad A \in \mathbb{R}^{n\times n}, \;\; B \in \mathbb{R}^{m\times m}, \;\; X, C \in \mathbb{R}^{n\times m},$$

with A, B, and C **quasiseparable**

Quasiseparable matrix

A matrix A is *quasiseparable* of order k if the maximum of the ranks of all its submatrices contained in the strictly upper or lower part is less or equal than k.



💡 We have seen that A, B, and C quasiseparable $\Rightarrow$ X with decay of the singular values of off-diagonal blocks of C.

Sylvester with quasiseparable matrices

Theorem (Massei, Palitta, and Robol 2018, Theorem 2.12)

Let A, B be matrices of quasiseparable rank k_A and k_B respectively and such that $W(A) \subseteq E$ and $W(-B) \subseteq F$. Consider the Sylvester equation $AX + XB = C$, with C of quasiseparable rank k_C. Then a generic off-diagonal block Y of the solution X satisfies

$$\frac{\sigma_{1+k\ell}(Y)}{\sigma_1(Y)} \leq \mathcal{C}^2 \cdot Z_\ell(E, F), \qquad k := k_A + k_B + k_C.$$

Where $Z_\ell(E, F)$ is the solution of the **Zolotarev problem**

$$Z_\ell(E, F) \triangleq \inf_{r(x) \in \mathcal{R}_{\ell,\ell}} \frac{\max_{x \in E} |r(x)|}{\min_{y \in F} |r(y)|}, \qquad \ell \geq 1,$$

for $\mathcal{R}_{\ell,\ell}$ is the set of rational functions of degree at most (ℓ, ℓ), and $\mathcal{C}$ is the Crouzeix universal constant.

Sylvester with quasiseparable matrices

❓ Do the *decaying singular values* in the blocks implies the existence of a **quasiseparable approximant**?

Sylvester with quasiseparable matrices

❓ Do the *decaying singular values* in the blocks implies the existence of a **quasiseparable approximant**?

ϵ-quasiseparable matrices of rank k (ε-qsrank k)

We say that A has *ϵ-quasiseparable rank k* if, for every off-diagonal block Y, $\sigma_{k+1}(Y) \leq \epsilon$. If the property holds for the lower (respectively upper) offdiagonal blocks, we say that A has lower (respectively upper) ϵ-quasiseparable rank k.

Sylvester with quasiseparable matrices

❓ Do the *decaying singular values* in the blocks implies the existence of a **quasiseparable approximant**?

ϵ-quasiseparable matrices of rank k (ε-qsrank k)

We say that A has *ϵ-quasiseparable rank k* if, for every off-diagonal block Y, $\sigma_{k+1}(Y) \leq \epsilon$. If the property holds for the lower (respectively upper) offdiagonal blocks, we say that A has lower (respectively upper) ϵ-quasiseparable rank k.

Submatrices and off-diagonal blocks

If a matrix A has ϵ-quasiseparable rank k, then any of its principal submatrix A' has ϵ-quasiseparable rank k.

Any off-diagonal block Y of A' is also an off-diagonal block of $A \Rightarrow \sigma_{k+1}(Y) \leq \epsilon$.

Sylvester with quasiseparable matrices

❓ Do the *decaying singular values* in the blocks implies the existence of a **quasiseparable approximant**?

ϵ-quasiseparable matrices of rank k (ε-qsrank k)

We say that A has *ϵ-quasiseparable rank k* if, for every off-diagonal block Y, $\sigma_{k+1}(Y) \leq \epsilon$. If the property holds for the lower (respectively upper) offdiagonal blocks, we say that A has lower (respectively upper) ϵ-quasiseparable rank k.

Submatrices and off-diagonal blocks

If a matrix A has ϵ-quasiseparable rank k, then any of its principal submatrix A' has ϵ-quasiseparable rank k.

Any off-diagonal block Y of A' is also an off-diagonal block of $A \Rightarrow \sigma_{k+1}(Y) \leq \epsilon$.

For $\oplus$ the direct sum

Technical lemma

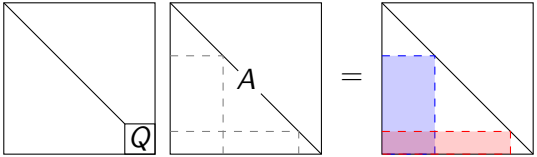
Let A be a matrix with ϵ-quasiseparable rank k, Q any $(k+1) \times (k+1)$ unitary matrix. Then, $(I_{n-k-1} \oplus Q)A$ also has ϵ-quasiseparable rank k.

Sylvester with quasiseparable matrices

❓ Do the *decaying singular values* in the blocks implies the existence of a **quasiseparable approximant**?

ϵ-quasiseparable matrices of rank k (ε-qsrank k)

We say that A has *ϵ-quasiseparable rank k* if, for every off-diagonal block Y, $\sigma_{k+1}(Y) \leq \epsilon$. If the property holds for the lower (respectively upper) offdiagonal blocks, we say that A has lower (respectively upper) ϵ-quasiseparable rank k.



Q acts on the tall block of A without changing its singular values, while the small one has small rank thanks to the small number of rows.

Sylvester with quasiseparable matrices

Theorem (Massei, Palitta, and Robol 2018, Theorem 2.16)

Let A be of ϵ-quasiseparable rank k, for $\epsilon > 0$. Then, there exists a matrix δA of norm bounded by $\|\delta A\|_2 \leq 2\sqrt{n} \cdot \epsilon$ so that $A + \delta A$ is k-quasiseparable.

Sylvester with quasiseparable matrices

Theorem (Massei, Palitta, and Robol 2018, Theorem 2.16)

Let A be of ϵ-quasiseparable rank k, for $\epsilon > 0$. Then, there exists a matrix δA of norm bounded by $\|\delta A\|_2 \leq 2\sqrt{n} \cdot \epsilon$ so that $A + \delta A$ is k-quasiseparable.

A matrix with ϵ-quasiseparable rank of k can be well-approximated by a matrix with exact quasiseparable rank k!

Sylvester with quasiseparable matrices

Theorem (Massei, Palitta, and Robol 2018, Theorem 2.16)

Let A be of ϵ-quasiseparable rank k, for $\epsilon > 0$. Then, there exists a matrix δA of norm bounded by $\|\delta A\|_2 \leq 2\sqrt{n} \cdot \epsilon$ so that $A + \delta A$ is k-quasiseparable.

A matrix with ϵ-quasiseparable rank of k can be well-approximated by a matrix with exact quasiseparable rank k!

If the **spectra** *of A and $-B$ are* **well-separated** *in the Zolotarev sense, we can* **preserve structure***!*

Sylvester with quasiseparable matrices

Theorem (Massei, Palitta, and Robol 2018, Theorem 2.16)

Let A be of ϵ-quasiseparable rank k, for $\epsilon > 0$. Then, there exists a matrix δA of norm bounded by $\|\delta A\|_2 \leq 2\sqrt{n} \cdot \epsilon$ so that $A + \delta A$ is k-quasiseparable.

A matrix with ϵ-quasiseparable rank of k can be well-approximated by a matrix with exact quasiseparable rank k!

If the **spectra** *of A and $-B$ are* **well-separated** *in the Zolotarev sense, we can* **preserve structure***!*

How can we operate efficiently with these matrix structures?

Sylvester with quasiseparable matrices

Theorem (Massei, Palitta, and Robol 2018, Theorem 2.16)

Let A be of ϵ-quasiseparable rank k, for $\epsilon > 0$. Then, there exists a matrix δA of norm bounded by $\|\delta A\|_2 \leq 2\sqrt{n} \cdot \epsilon$ so that $A + \delta A$ is k-quasiseparable.

A matrix with ϵ-quasiseparable rank of k can be well-approximated by a matrix with exact quasiseparable rank k!

If the **spectra** *of A and $-B$ are* **well-separated** *in the Zolotarev sense, we can* **preserve structure***!*

How can we operate efficiently with these matrix structures?

Reduced cost for BLAS-like operations,

Sylvester with quasiseparable matrices

Theorem (Massei, Palitta, and Robol 2018, Theorem 2.16)

Let A be of ϵ-quasiseparable rank k, for $\epsilon > 0$. Then, there exists a matrix δA of norm bounded by $\|\delta A\|_2 \leq 2\sqrt{n} \cdot \epsilon$ so that $A + \delta A$ is k-quasiseparable.

A matrix with ϵ-quasiseparable rank of k can be well-approximated by a matrix with exact quasiseparable rank k!

If the **spectra** *of A and $-B$ are* **well-separated** *in the Zolotarev sense, we can* **preserve structure***!*

How can we operate efficiently with these matrix structures?

Reduced cost for BLAS-like operations,

Contained storage cost.

Sylvester with quasiseparable matrices

Theorem (Massei, Palitta, and Robol 2018, Theorem 2.16)

Let A be of ϵ-quasiseparable rank k, for $\epsilon > 0$. Then, there exists a matrix δA of norm bounded by $\|\delta A\|_2 \leq 2\sqrt{n} \cdot \epsilon$ so that $A + \delta A$ is k-quasiseparable.

A matrix with ϵ-quasiseparable rank of k can be well-approximated by a matrix with exact quasiseparable rank k!

If the **spectra** *of A and $-B$ are* **well-separated** *in the Zolotarev sense, we can* **preserve structure***!*

How can we operate efficiently with these matrix structures?

Reduced cost for BLAS-like operations,

Contained storage cost.

Hierarchical matrix formats!

Hierarchical matrix formats

There exist **many hierarchical matrix formats**:

- $\mathcal{H}$-Matrices,
- $\mathcal{H}^2$-Matrices,
- **H**ierarchical **O**ff-**D**iagonal **L**ow-**R**ank (HODLR),
- **H**ierarchically **S**emi**S**eparable (HSS),
- **B**lock **L**ow-**R**ank (BLR).

Hierarchical matrix formats

There exist **many hierarchical matrix formats**:

- $\mathcal{H}$-Matrices,
- $\mathcal{H}^2$-Matrices,
- **H**ierarchical **O**ff-**D**iagonal **L**ow-**R**ank (HODLR),
- **H**ierarchically **S**emi**S**eparable (HSS),
- **B**lock **L**ow-**R**ank (BLR).

The topic would deserve a Ph.D. course on its own...

Hierarchical matrix formats

There exist **many hierarchical matrix formats**:

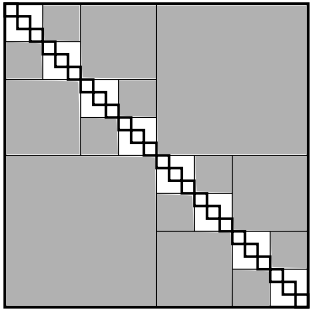
- $\mathcal{H}$-Matrices,
- $\mathcal{H}^2$-Matrices,
- **H**ierarchical **O**ff-**D**iagonal **L**ow-**R**ank (HODLR),
- **H**ierarchically **S**emi**S**eparable (HSS),
- **B**lock **L**ow-**R**ank (BLR).

The topic would deserve a Ph.D. course on its own... We are gonna focus only on the case of HODLR matrices (Hackbusch 2015, Chapter 3).

Hierarchical matrix formats

There exist **many hierarchical matrix formats**:

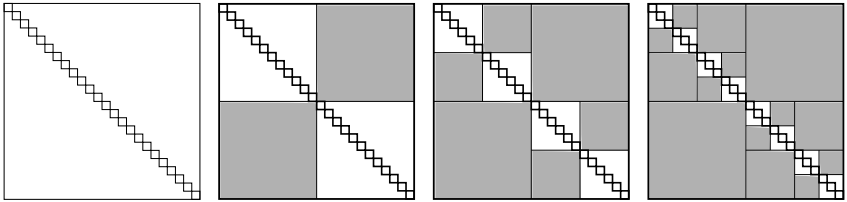
- $\mathcal{H}$-Matrices,
- $\mathcal{H}^2$-Matrices,
- **H**ierarchical **O**ff-**D**iagonal **L**ow-**R**ank (HODLR),
- **H**ierarchically **S**emi**S**eparable (HSS),
- **B**lock **L**ow-**R**ank (BLR).



The topic would deserve a Ph.D. course on its own... We are gonna focus only on the case of HODLR matrices (Hackbusch 2015, Chapter 3).

HODLR-matrices

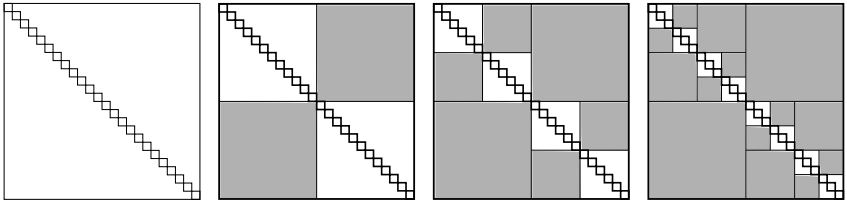
The **general idea**:



- The **grey blocks** are **low rank matrices** represented **in** a **compressed form**,
- the *diagonal blocks* in the last step are *stored as dense matrices*.

HODLR-matrices

The **general idea**:



- The **grey blocks** are **low rank matrices** represented **in** a **compressed form**,
- the *diagonal blocks* in the last step are *stored as dense matrices*.
- We need now a **formal definition** and a way to **define operations**.

HODLR-matrices: trees

Cluster tree

Given $n \in \mathbb{N}$, let $\mathcal{T}_p$ be a completely balanced binary tree of depth p whose nodes are subsets of $\{1, \ldots, n\}$. We say that $\mathcal{T}_p$ is a *cluster tree* if it satisfies:

HODLR-matrices: trees

Cluster tree

Given $n \in \mathbb{N}$, let $\mathcal{T}_p$ be a completely balanced binary tree of depth p whose nodes are subsets of $\{1, \ldots, n\}$. We say that $\mathcal{T}_p$ is a *cluster tree* if it satisfies:

- The root is $I_1^0 := I = \{1, \ldots, n\}$.

HODLR-matrices: trees

Cluster tree

Given $n \in \mathbb{N}$, let $\mathcal{T}_p$ be a completely balanced binary tree of depth p whose nodes are subsets of $\{1, \dots, n\}$. We say that $\mathcal{T}_p$ is a *cluster tree* if it satisfies:

- The root is $I_1^0 := I = \{1, \dots, n\}$.
- The nodes at level ℓ, denoted by $I_1^\ell, \dots, I_{2^\ell}^\ell$, form a partitioning of $\{1, \dots, n\}$ into consecutive indices:

$$I_i^\ell = \{n_{i-1}^{(\ell)} + 1 \dots, n_i^{(\ell)} - 1, n_i^{(\ell)}\}$$

  for some integers $0 = n_0^{(\ell)} \leq n_1^{(\ell)} \leq \dots \leq n_{2^\ell}^{(\ell)} = n$, $\ell = 0, \dots p$. In particular, if $n_{i-1}^{(\ell)} = n_i^{(\ell)}$ then $I_i^\ell = \emptyset$.

HODLR-matrices: trees

Cluster tree

Given $n \in \mathbb{N}$, let $\mathcal{T}_p$ be a completely balanced binary tree of depth p whose nodes are subsets of $\{1, \dots, n\}$. We say that $\mathcal{T}_p$ is a *cluster tree* if it satisfies:

- The root is $I_1^0 := I = \{1, \dots, n\}$.
- The nodes at level ℓ, denoted by $I_1^\ell, \dots, I_{2^\ell}^\ell$, form a partitioning of $\{1, \dots, n\}$ into consecutive indices:
$$I_i^\ell = \{n_{i-1}^{(\ell)} + 1 \dots, n_i^{(\ell)} - 1, n_i^{(\ell)}\}$$
for some integers $0 = n_0^{(\ell)} \leq n_1^{(\ell)} \leq \dots \leq n_{2^\ell}^{(\ell)} = n$, $\ell = 0, \dots p$. In particular, if $n_{i-1}^{(\ell)} = n_i^{(\ell)}$ then $I_i^\ell = \emptyset$.
- The node I_i^ℓ has children $I_{2i-1}^{\ell+1}$ and $I_{2i}^{\ell+1}$, for any $1 \leq \ell \leq p-1$. The children form a partitioning of their parent.

HODLR-matrices: trees

Cluster tree

Given $n \in \mathbb{N}$, let $\mathcal{T}_p$ be a completely balanced binary tree of depth p whose nodes are subsets of $\{1, \dots, n\}$. We say that $\mathcal{T}_p$ is a *cluster tree* if it satisfies:

- The root is $I_1^0 := I = \{1, \dots, n\}$.
- The nodes at level ℓ, denoted by $I_1^\ell, \dots, I_{2^\ell}^\ell$, form a partitioning of $\{1, \dots, n\}$ into consecutive indices:
$$I_i^\ell = \{n_{i-1}^{(\ell)} + 1 \dots, n_i^{(\ell)} - 1, n_i^{(\ell)}\}$$
for some integers $0 = n_0^{(\ell)} \leq n_1^{(\ell)} \leq \dots \leq n_{2^\ell}^{(\ell)} = n$, $\ell = 0, \dots p$. In particular, if $n_{i-1}^{(\ell)} = n_i^{(\ell)}$ then $I_i^\ell = \emptyset$.
- The node I_i^ℓ has children $I_{2i-1}^{\ell+1}$ and $I_{2i}^{\ell+1}$, for any $1 \leq \ell \leq p-1$. The children form a partitioning of their parent.

Nodes at a level ℓ partition A into a $2^\ell \times 2^\ell$ block matrix with blocks $\{A(I_i^\ell, I_j^\ell)\}_{i,j=1}^{2^\ell}$.

HODLR-matrices: trees

$I = \{1, 2, 3, 4, 5, 6, 7, 8\}$

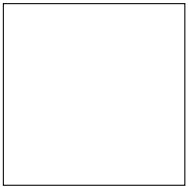
$I_1^1 = \{1, 2, 3, 4\}$ $\quad$ $I_2^1 = \{5, 6, 7, 8\}$

$I_1^2 = \{1, 2\}$ $\quad$ $I_2^2 = \{3, 4\}$ $\quad$ $I_3^2 = \{5, 6\}$ $\quad$ $I_4^2 = \{7, 8\}$

$I_1^3 = \{1\}$ $\quad$ $I_2^3 = \{2\}$ $\quad$ $I_3^3 = \{3\}$ $\quad$ $I_4^3 = \{4\}$ $\quad$ $I_5^3 = \{5\}$ $\quad$ $I_6^3 = \{6\}$ $\quad$ $I_7^3 = \{7\}$ $\quad$ $I_8^3 = \{8\}$

- The root $I = \{1, \ldots, 8\}$,

$\ell = 0$



HODLR-matrices: trees

$$I = \{1,2,3,4,5,6,7,8\}$$

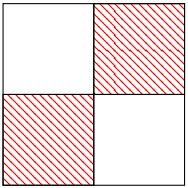
$$I_1^1 = \{1,2,3,4\} \qquad I_2^1 = \{5,6,7,8\}$$

$$I_1^2 = \{1,2\} \qquad I_2^2 = \{3,4\} \qquad I_3^2 = \{5,6\} \qquad I_4^2 = \{7,8\}$$

$$I_1^3 = \{1\} \quad I_2^3 = \{2\} \quad I_3^3 = \{3\} \quad I_4^3 = \{4\} \quad I_5^3 = \{5\} \quad I_6^3 = \{6\} \quad I_7^3 = \{7\} \quad I_8^3 = \{8\}$$

- The root $I = \{1, \ldots, 8\}$,
- Nodes at level 1: I_1^1 and I_1^2,

$\ell = 1$



HODLR-matrices: trees

$$I = \{1,2,3,4,5,6,7,8\}$$

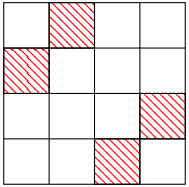
$$I_1^1 = \{1,2,3,4\} \qquad I_2^1 = \{5,6,7,8\}$$

$$I_1^2 = \{1,2\} \qquad I_2^2 = \{3,4\} \qquad I_3^2 = \{5,6\} \qquad I_4^2 = \{7,8\}$$

$$I_1^3 = \{1\} \quad I_2^3 = \{2\} \quad I_3^3 = \{3\} \quad I_4^3 = \{4\} \quad I_5^3 = \{5\} \quad I_6^3 = \{6\} \quad I_7^3 = \{7\} \quad I_8^3 = \{8\}$$

- The root $I = \{1, \ldots, 8\}$,
- Nodes at level 1: I_1^1 and I_1^2,
- Nodes at level 2: $(I_1^1) = \{I_1^2, I_2^2\}$, $(I_2^1) = \{I_3^2, I_4^2\}$,

$\ell = 2$



HODLR-matrices: trees

$$I = \{1,2,3,4,5,6,7,8\}$$

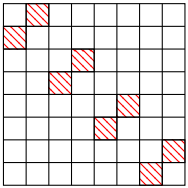
$$I_1^1 = \{1,2,3,4\} \qquad I_2^1 = \{5,6,7,8\}$$

$$I_1^2 = \{1,2\} \qquad I_2^2 = \{3,4\} \qquad I_3^2 = \{5,6\} \qquad I_4^2 = \{7,8\}$$

$$I_1^3 = \{1\} \quad I_2^3 = \{2\} \quad I_3^3 = \{3\} \quad I_4^3 = \{4\} \quad I_5^3 = \{5\} \quad I_6^3 = \{6\} \quad I_7^3 = \{7\} \quad I_8^3 = \{8\}$$

- The root $I = \{1, \ldots, 8\}$,
- Nodes at level 1: I_1^1 and I_1^2,
- Nodes at level 2: $(I_1^1) = \{I_1^2, I_2^2\}$, $(I_2^1) = \{I_3^2, I_4^2\}$,
- Nodes at level 3: $(I_1^2) = \{I_1^3, I_2^3\}$, $\ldots$, $(I_4^2) = \{I_7^3, I_8^3\}$.

$\ell = 3$



HODLR-matrices: definition

HODLR matrix

Let $A \in \mathbb{R}^{n \times n}$ and consider a cluster tree $\mathcal{T}_p$.

1. Given $k \in \mathbb{N}$, A is said to be a $(\mathcal{T}_p, k)$-HODLR matrix if every off-diagonal block

$$A(I_i^\ell, I_j^\ell) \quad \text{such that } I_i^\ell \text{ and } I_j^\ell \text{ are siblings in } \mathcal{T}_p, \quad \ell = 1, \ldots, p,$$

   has rank at most k.

2. The HODLR rank of A (with respect to $\mathcal{T}_p$) is the smallest integer k such that A is a $(\mathcal{T}_p, k)$-HODLR matrix.

HODLR-matrices: definition

HODLR matrix

Let $A \in \mathbb{R}^{n \times n}$ and consider a cluster tree $\mathcal{T}_p$.

1. Given $k \in \mathbb{N}$, A is said to be a $(\mathcal{T}_p, k)$-HODLR matrix if every off-diagonal block

$$A(I_i^\ell, I_j^\ell) \quad \text{such that } I_i^\ell \text{ and } I_j^\ell \text{ are siblings in } \mathcal{T}_p, \quad \ell = 1, \ldots, p,$$

   has rank at most k.

2. The HODLR rank of A (with respect to $\mathcal{T}_p$) is the smallest integer k such that A is a $(\mathcal{T}_p, k)$-HODLR matrix.

- $\mathcal{T}_p$ is often chosen to be as balanced as possible, i.e., cardinalities of I_i^ℓ are nearly equal for a given ℓ, with a dept determined by a minimal diagonal block size $n_{\min}$.

HODLR-matrices: definition

HODLR matrix

Let $A \in \mathbb{R}^{n \times n}$ and consider a cluster tree $\mathcal{T}_p$.

1. Given $k \in \mathbb{N}$, A is said to be a $(\mathcal{T}_p, k)$-HODLR matrix if every off-diagonal block

$$A(I_i^\ell, I_j^\ell) \quad \text{such that } I_i^\ell \text{ and } I_j^\ell \text{ are siblings in } \mathcal{T}_p, \quad \ell = 1, \ldots, p,$$

   has rank at most k.

2. The HODLR rank of A (with respect to $\mathcal{T}_p$) is the smallest integer k such that A is a $(\mathcal{T}_p, k)$-HODLR matrix.

- $\mathcal{T}_p$ is often chosen to be as balanced as possible, i.e., cardinalities of I_i^ℓ are nearly equal for a given ℓ, with a dept determined by a minimal diagonal block size $n_{\min}$.
- The classical choice is to have a **binary tree**, i.e., $n = 2^p n_{\min}$.

HODLR-matrices: occupied space

If we assume **identical ranks** k and a **balanced partitioning** then

- Storage for off-diagonal blocks $A(I_i^\ell, I_j^\ell) = U_i^{(\ell)}(V_j^{(\ell)})^T$, $U_i^{(\ell)}, V_j^{(\ell)} \in \mathbb{R}^{m_\ell \times k}$:
  On level $\ell > 0$ there are 2^ℓ off-diagonal blocks

$$2k \sum_{\ell=1}^{p} 2^\ell m_\ell = 2kn_0 \sum_{\ell=1}^{p} 2^\ell 2^{p-\ell} 2kn_0 p 2^p = 2knp = 2kn \log_2(n/n_0),$$

HODLR-matrices: occupied space

If we assume **identical ranks** k and a **balanced partitioning** then

- Storage for off-diagonal blocks $A(I_i^\ell, I_j^\ell) = U_i^{(\ell)}(V_j^{(\ell)})^T$, $U_i^{(\ell)}, V_j^{(\ell)} \in \mathbb{R}^{m_\ell \times k}$:
  On level $\ell > 0$ there are 2^ℓ off-diagonal blocks

$$2k \sum_{\ell=1}^{p} 2^\ell m_\ell = 2kn_0 \sum_{\ell=1}^{p} 2^\ell 2^{p-\ell} 2kn_0 p 2^p = 2knp = 2kn \log_2(n/n_0),$$

- Storage requirements for diagonal blocks

$$2^p n_0^2 = nn_0,$$

HODLR-matrices: occupied space

If we assume **identical ranks** k and a **balanced partitioning** then

- Storage for off-diagonal blocks $A(I_i^\ell, I_j^\ell) = U_i^{(\ell)}(V_j^{(\ell)})^T$, $U_i^{(\ell)}, V_j^{(\ell)} \in \mathbb{R}^{m_\ell \times k}$:
  On level $\ell > 0$ there are 2^ℓ off-diagonal blocks

$$2k \sum_{\ell=1}^{p} 2^\ell m_\ell = 2kn_0 \sum_{\ell=1}^{p} 2^\ell 2^{p-\ell} 2kn_0 p 2^p = 2knp = 2kn \log_2(n/n_0),$$

- Storage requirements for diagonal blocks

$$2^p n_0^2 = nn_0,$$

- Total, assuming $n_0 = O(1)$, is then

$$O(kn \log n).$$

HODLR-matrices: occupied space

If we assume **identical ranks** k and a **balanced partitioning** then

- Storage for off-diagonal blocks $A(I_i^\ell, I_j^\ell) = U_i^{(\ell)}(V_j^{(\ell)})^T$, $U_i^{(\ell)}, V_j^{(\ell)} \in \mathbb{R}^{m_\ell \times k}$:
  On level $\ell > 0$ there are 2^ℓ off-diagonal blocks

$$2k \sum_{\ell=1}^{p} 2^\ell m_\ell = 2kn_0 \sum_{\ell=1}^{p} 2^\ell 2^{p-\ell} 2kn_0 p 2^p = 2knp = 2kn \log_2(n/n_0),$$

- Storage requirements for diagonal blocks

$$2^p n_0^2 = nn_0,$$

- Total, assuming $n_0 = O(1)$, is then

$$O(kn \log n).$$

- Both requirements on ranks and partitioning can be relaxed to obtain similar results.

HODLR-matrices: building the representation

⚠ Is **non trivial** to construct structured representations efficiently, especially if you want to avoid computing the whole n^2 coefficients!

HODLR-matrices: building the representation

⚠ Is **non trivial** to construct structured representations efficiently, especially if you want to avoid computing the whole n^2 coefficients!

- Build a **cluster tree** $\mathcal{T}_p$ for the given index set,

HODLR-matrices: building the representation

Is **non trivial** to construct structured representations efficiently, especially if you want to avoid computing the whole n^2 coefficients!

- Build a **cluster tree** $\mathcal{T}_p$ for the given index set,

If A is **dense**:

- Use Householder **QR decomposition** with column pivoting or **SVD** on off-diagonal blocks,

HODLR-matrices: building the representation

Is **non trivial** to construct structured representations efficiently, especially if you want to avoid computing the whole n^2 coefficients!

- Build a **cluster tree** $\mathcal{T}_p$ for the given index set,

If A is **dense**:

- Use Householder **QR decomposition** with column pivoting or **SVD** on off-diagonal blocks,
- The **rank** of each off-diagonal block $A(I_i^p, I_j^p)$ is chosen such that the spectral **norm of the approximation error** is bounded by ϵ times $\|A(I_i^p, I_j^p)\|_2$.

HODLR-matrices: building the representation

Is **non trivial** to construct structured representations efficiently, especially if you want to avoid computing the whole n^2 coefficients!

- Build a **cluster tree** $\mathcal{T}_p$ for the given index set,

If A is **dense**:

- Use Householder **QR decomposition** with column pivoting or **SVD** on off-diagonal blocks,
- The **rank** of each off-diagonal block $A(I_i^p, I_j^p)$ is chosen such that the spectral **norm of the approximation error** is bounded by ϵ times $\|A(I_i^p, I_j^p)\|_2$.

If A is **sparse**:

- Use a **two sided Lanczos method** only requiring matrix-vector multiplications with an off-diagonal block and its transpose, combined with recompression to each off-diagonal block.

HODLR-matrices: building the representation

Is **non trivial** to construct structured representations efficiently, especially if you want to avoid computing the whole n^2 coefficients!

- Build a **cluster tree** $\mathcal{T}_p$ for the given index set,

If A is **dense**:

- Use Householder **QR decomposition** with column pivoting or **SVD** on off-diagonal blocks,
- The **rank** of each off-diagonal block $A(I_i^p, I_j^p)$ is chosen such that the spectral **norm of the approximation error** is bounded by ϵ times $\|A(I_i^p, I_j^p)\|_2$.

If A is **sparse**:

- Use a **two sided Lanczos method** only requiring matrix-vector multiplications with an off-diagonal block and its transpose, combined with recompression to each off-diagonal block.

If A is **structured** use an *ad-hoc* constructor!

HODLR of Grünwald–Letnikov

Theorem (Fiedler 2010, Theorem A)

Let $\mathbf{x}, \mathbf{y}$ two real vectors of length N, with ascending and descending ordered entries, respectively. Moreover, we denote with $C(\mathbf{x}, \mathbf{y})$ the Cauchy matrix defined by

$$C_{ij} = \frac{1}{x_i - y_j}, \qquad i, j = 1, \ldots, N.$$

If $C(\mathbf{x}, \mathbf{y}) = C(\mathbf{x}, \mathbf{y})^T$, $x_i \in [a, b]$, $y_j \in [c, d]$ with $a > d$, then $C(\mathbf{x}, \mathbf{y})$ is positive definite.

HODLR of Grünwald–Letnikov

Theorem (Fiedler 2010, Theorem A)

Let $\mathbf{x}, \mathbf{y}$ two real vectors of length N, with ascending and descending ordered entries, respectively. Moreover, we denote with $C(\mathbf{x}, \mathbf{y})$ the Cauchy matrix defined by

$$C_{ij} = \frac{1}{x_i - y_j}, \qquad i, j = 1, \ldots, N.$$

If $C(\mathbf{x}, \mathbf{y}) = C(\mathbf{x}, \mathbf{y})^T$, $x_i \in [a, b]$, $y_j \in [c, d]$ with $a > d$, then $C(\mathbf{x}, \mathbf{y})$ is positive definite.

Theorem (Beckermann and Townsend 2019, Theorem 5.5)

Let H be a positive semidefinite Hankel matrix of size N. Then, the ϵ-rank of H is bounded by

$$\mathrm{rank}_\epsilon(H) \leq 2 + 2 \left\lceil \frac{2}{\pi^2} \log\left(\frac{4}{\pi} N\right) \log\left(\frac{16}{\epsilon}\right) \right\rceil \triangleq \mathfrak{B}(N, \epsilon).$$

HODLR of Grünwald–Letnikov

We need to work with $G_N \in \mathbb{R}^{N\times N}$

$$G_N = -\begin{bmatrix} g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0 & 0 \\ g_2^{(\alpha)} & g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ g_{N-1}^{(\alpha)} & \ddots & \ddots & \ddots & g_1^{(\alpha)} & g_0^{(\alpha)} \\ g_N^{(\alpha)} & g_{N-1}^{(\alpha)} & \cdots & \cdots & g_2^{(\alpha)} & g_1^{(\alpha)} \end{bmatrix}$$

Lemma (Massei, Mazza, and Robol 2019)

Consider the Hankel matrix H defined as

$$H = (h_{ij}), \qquad h_{ij} = g_{i+j}^{(\alpha)},$$

for $1 \le \alpha \le 2$. Then, H is positive semidefinite.

- Show that H is obtained as the sum of a positive definite Cauchy matrix and a positive semidefinite matrix.
- Use the result by Beckermann and Townsend 2019.

HODLR of Grünwald–Letnikov

Proof. For $k \geq 2$ we rewrite $g_k^{(\alpha)}$ as

$$\begin{aligned} g_k^{(\alpha)} &= \frac{(-1)^k}{k!}\alpha(\alpha-1)\dots(\alpha-k+1) \\ &= \frac{\alpha(\alpha-1)}{k!}(k-\alpha-1)(k-\alpha-2)\dots(2-\alpha) \\ &= \alpha(\alpha-1)\frac{\Gamma(k-\alpha)}{\Gamma(k+1)\Gamma(2-\alpha)}. \end{aligned}$$

HODLR of Grünwald–Letnikov

Proof. For $k \geq 2$ we rewrite $g_k^{(\alpha)}$ as

$$g_k^{(\alpha)} = \alpha(\alpha-1)\frac{\Gamma(k-\alpha)}{\Gamma(k+1)\Gamma(2-\alpha)}.$$

Use the Gauss representation of the Euler Γ

$$\Gamma(z) = \lim_{m\to\infty} \frac{m!m^z}{z(z+1)(z+2)\dots(z+m)}, \quad z \neq \{0,-1,-2,\dots\},$$

we rewrite

$$g_k^{(\alpha)} = \alpha(\alpha-1)\lim_{m\to\infty}\frac{1}{m!m^3}\prod_{p=0}^{m}\frac{k+1+p}{k-\alpha+p}(2-\alpha+p).$$

HODLR of Grünwald–Letnikov

Proof. For $k \geq 2$ we rewrite $g_k^{(\alpha)}$ as

$$g_k^{(\alpha)} = \alpha(\alpha-1)\frac{\Gamma(k-\alpha)}{\Gamma(k+1)\Gamma(2-\alpha)}.$$

We rewrite

$$H = \lim_{m \to +\infty} H_0 \circ \ldots \circ H_m, \qquad (H_p)_{ij} = \frac{i+j+1+p}{i+j-\alpha+p}$$

for $\circ$ the Hadamard product, $\{H_j\}_{j=0}^m$ Hankel matrices.

HODLR of Grünwald–Letnikov

Proof. For $k \geq 2$ we rewrite $g_k^{(\alpha)}$ as

$$g_k^{(\alpha)} = \alpha(\alpha-1)\frac{\Gamma(k-\alpha)}{\Gamma(k+1)\Gamma(2-\alpha)}.$$

We rewrite

$$H = \lim_{m\to+\infty} H_0 \circ \ldots \circ H_m, \qquad (H_p)_{ij} = \frac{i+j+1+p}{i+j-\alpha+p}$$

for $\circ$ the Hadamard product, $\{H_j\}_{j=0}^m$ Hankel matrices. Schur Product Theorem tells us that "*the Hadamard product of two positive definite matrices is also a positive definite matrix*"
$\Rightarrow$ If $H_0 \circ \ldots \circ H_m$ is positive semidefinite for every m then H is also positive semidefinite.

HODLR of Grünwald–Letnikov

Proof. For $k \geq 2$ we rewrite $g_k^{(\alpha)}$ as

$$g_k^{(\alpha)} = \alpha(\alpha-1)\frac{\Gamma(k-\alpha)}{\Gamma(k+1)\Gamma(2-\alpha)}.$$

We rewrite

$$H = \lim_{m\to+\infty} H_0 \circ \ldots \circ H_m, \qquad (H_p)_{ij} = \frac{i+j+1+p}{i+j-\alpha+p}$$

for $\circ$ the Hadamard product, $\{H_j\}_{j=0}^m$ Hankel matrices. Rewrite

$$(H_p)_{ij} = \frac{i+j+1+p}{i+j-\alpha+p} = 1+\frac{\alpha+1}{i+j-\alpha+p}$$

HODLR of Grünwald–Letnikov

Proof. For $k \geq 2$ we rewrite $g_k^{(\alpha)}$ as

$$g_k^{(\alpha)} = \alpha(\alpha-1)\frac{\Gamma(k-\alpha)}{\Gamma(k+1)\Gamma(2-\alpha)}.$$

We rewrite

$$H = \lim_{m\to+\infty} H_0 \circ \ldots \circ H_m, \qquad (H_p)_{ij} = \frac{i+j+1+p}{i+j-\alpha+p}$$

for $\circ$ the Hadamard product, $\{H_j\}_{j=0}^m$ Hankel matrices. Rewrite

$$(H_p)_{ij} = 1 + \frac{\alpha+1}{i+j-\alpha+p}, \quad H_p = \mathbf{1}\mathbf{1}^T + (\alpha+1)\cdot C(\mathbf{x}, -\mathbf{x}), \quad \mathbf{x} = \begin{bmatrix} 1 \\ \vdots \\ N \end{bmatrix} + \frac{p-\alpha}{2}\mathbf{1},$$

$\mathbf{x} \geq 0$ for $\alpha < 2$, thus $C(\mathbf{x}, -\mathbf{x})$ is PD. Then H_p is positive semidefinite as the sum of a PD and positive semidefinite matrix. $\square$

HODLR of Grünwald–Letnikov

Proposition (Massei, Mazza, and Robol 2019, Lemma 3.15)

For every $\epsilon > 0$, the ϵ-qsrank of G_N is bounded by

$$\mathrm{qsrank}_\epsilon(G_N) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right) = 2 + 2\left\lceil \frac{2}{\pi^2} \log\left(\frac{4}{\pi} N\right) \log\left(\frac{32}{\epsilon}\right)\right\rceil.$$

Proof.

HODLR of Grünwald–Letnikov

Proposition (Massei, Mazza, and Robol 2019, Lemma 3.15)

For every $\epsilon > 0$, the ϵ-qsrank of G_N is bounded by

$$\mathrm{qsrank}_\epsilon(G_N) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right) = 2 + 2\left\lceil \frac{2}{\pi^2} \log\left(\frac{4}{\pi} N\right) \log\left(\frac{32}{\epsilon}\right)\right\rceil.$$

Proof. We just need to work on the lower triangle, for the upper the rank is at most 1 (Hessenberg).

HODLR of Grünwald–Letnikov

Proposition (Massei, Mazza, and Robol 2019, Lemma 3.15)

For every $\epsilon > 0$, the ϵ-qsrank of G_N is bounded by

$$\mathrm{qsrank}_\epsilon(G_N) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right) = 2 + 2\left\lceil \frac{2}{\pi^2} \log\left(\frac{4}{\pi} N\right) \log\left(\frac{32}{\epsilon}\right)\right\rceil.$$

Proof. Let $Y \in \mathbb{R}^{s \times t}$ be any lower off-diagonal block of G_N. *Without loss of generality* we assume that Y is maximal, i.e. $s + t = N$.

HODLR of Grünwald–Letnikov

Proposition (Massei, Mazza, and Robol 2019, Lemma 3.15)

For every $\epsilon > 0$, the ϵ-qsrank of G_N is bounded by

$$\mathrm{qsrank}_\epsilon(G_N) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right) = 2 + 2\left\lceil \frac{2}{\pi^2} \log\left(\frac{4}{\pi} N\right) \log\left(\frac{32}{\epsilon}\right)\right\rceil.$$

Proof. Let $Y \in \mathbb{R}^{s \times t}$ be any lower off-diagonal block of G_N. *Without loss of generality* we assume that Y is maximal, i.e. $s + t = N$. (If $\mathrm{rank}(Y + \delta Y) = k$ and $\|\delta Y\|_2 \leq \epsilon \|G_N\|_2$ then the submatrices of δY verify the analogous claim for the corresponding ones of Y.)

HODLR of Grünwald–Letnikov

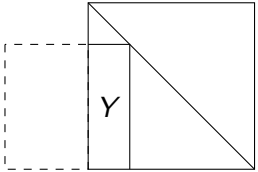
Proposition (Massei, Mazza, and Robol 2019, Lemma 3.15)

For every $\epsilon > 0$, the ϵ-qsrank of G_N is bounded by

$$\mathrm{qsrank}_\epsilon(G_N) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right) = 2 + 2\left\lceil \frac{2}{\pi^2} \log\left(\frac{4}{\pi} N\right) \log\left(\frac{32}{\epsilon}\right)\right\rceil.$$

Proof. Let $Y \in \mathbb{R}^{s \times t}$ be any lower off-diagonal block of G_N. *Without loss of generality* we assume that Y is maximal, i.e. $s + t = N$.
Entries Y are given by $Y_{ij} = -g^{(\alpha)}_{1+i-j+t}$. Call $h = \max\{s, t\}$, and A the $h \times h$ matrix defined by $A_{ij} = -g^{(\alpha)}_{1+i-j+h}$.

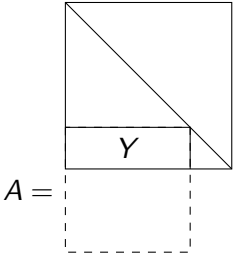
HODLR of Grünwald–Letnikov

$A =$ 

For every $1 \leq i \leq s$ and $1 \leq j \leq t$ one have
$Y_{ij} = -g^{(\alpha)}_{1+i-j+t} = -g^{(\alpha)}_{1+i-(j-t+h)+h} = A_{i,j-t+h}$.

Proof. Let $Y \in \mathbb{R}^{s \times t}$ be any lower off-diagonal block of G_N. *Without loss of generality* we assume that Y is maximal, i.e. $s + t = N$.
Entries Y are given by $Y_{ij} = -g^{(\alpha)}_{1+i-j+t}$. Call $h = \max\{s, t\}$, and A the $h \times h$ matrix defined by $A_{ij} = -g^{(\alpha)}_{1+i-j+h}$. Y coincides with either the last t columns or the first s rows of A.

HODLR of Grünwald–Letnikov



For every $1 \leq i \leq s$ and $1 \leq j \leq t$ one have
$Y_{ij} = -g^{(\alpha)}_{1+i-j+t} = -g^{(\alpha)}_{1+i-(j-t+h)+h} = A_{i,j-t+h}$.

Proof. Let $Y \in \mathbb{R}^{s \times t}$ be any lower off-diagonal block of G_N. *Without loss of generality* we assume that Y is maximal, i.e. $s + t = N$.
Entries Y are given by $Y_{ij} = -g^{(\alpha)}_{1+i-j+t}$. Call $h = \max\{s, t\}$, and A the $h \times h$ matrix defined by $A_{ij} = -g^{(\alpha)}_{1+i-j+h}$. Y coincides with either the last t columns or the first s rows of A.

HODLR of Grünwald–Letnikov

Proposition (Massei, Mazza, and Robol 2019, Lemma 3.15)

For every $\epsilon > 0$, the ϵ-qsrank of G_N is bounded by

$$\mathrm{qsrank}_\epsilon(G_N) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right) = 2 + 2\left\lceil \frac{2}{\pi^2} \log\left(\frac{4}{\pi}N\right) \log\left(\frac{32}{\epsilon}\right)\right\rceil.$$

Proof. Let $Y \in \mathbb{R}^{s\times t}$ be any lower off-diagonal block of G_N. *Without loss of generality* we assume that Y is maximal, i.e. $s + t = N$.
Entries Y are given by $Y_{ij} = -g^{(\alpha)}_{1+i-j+t}$. Call $h = \max\{s, t\}$, and A the $h \times h$ matrix defined by $A_{ij} = -g^{(\alpha)}_{1+i-j+h}$. Y coincides with either the last t columns or the first s rows of A. In particular, Y is a submatrix of A and therefore $\|Y\|_2 \leq \|A\|_2$.

HODLR of Grünwald–Letnikov

Proposition (Massei, Mazza, and Robol 2019, Lemma 3.15)

For every $\epsilon > 0$, the ϵ-qsrank of G_N is bounded by

$$\mathrm{qsrank}_\epsilon(G_N) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right) = 2 + 2\left\lceil \frac{2}{\pi^2} \log\left(\frac{4}{\pi}N\right) \log\left(\frac{32}{\epsilon}\right)\right\rceil.$$

Proof. Let $Y \in \mathbb{R}^{s\times t}$ be any lower off-diagonal block of G_N. *Without loss of generality* we assume that Y is maximal, i.e. $s + t = N$.
Entries Y are given by $Y_{ij} = -g^{(\alpha)}_{1+i-j+t}$. Call $h = \max\{s, t\}$, and A the $h \times h$ matrix defined by $A_{ij} = -g^{(\alpha)}_{1+i-j+h}$. Y coincides with either the last t columns or the first s rows of A. In particular, Y is a submatrix of A and therefore $\|Y\|_2 \leq \|A\|_2$. ⚙ We need now to estimate $\|A\|_2$ in terms of $\|G_N\|_2$, thus we partition

$$A = \begin{bmatrix} A^{(11)} & A^{(12)} \\ A^{(21)} & A^{(22)} \end{bmatrix}, \qquad A^{(ij)} \in \mathbb{C}^{m_{ij}\times n_{ij}}, \qquad \begin{cases} m_{1j} = n_{i1} = \lceil \frac{h}{2} \rceil \\ m_{2j} = n_{i2} = \lfloor \frac{h}{2} \rfloor \end{cases}, \qquad \begin{cases} h \leq N-1, \\ m_{i,j} + n_{i,j} \leq N, \end{cases}$$

HODLR of Grünwald–Letnikov

Proof. and consider the subdiagonal block $T^{(ij)}$ of G_N defined by

$$T^{(ij)} = G_N(N-m_{ij}+1:N, N-m_{ij}-n_{ij}+1:N-m_{ij}), \qquad i,j=1,2, \qquad \begin{array}{l} T^{(ij)} \in \mathbb{R}^{m_{ij}\times n_{ij}}, \\ m_{ij}+n_{ij} \leq N. \end{array}$$

HODLR of Grünwald–Letnikov

Proof. and consider the subdiagonal block $T^{(ij)}$ of G_N defined by

$$T^{(ij)} = G_N(N-m_{ij}+1:N, N-m_{ij}-n_{ij}+1:N-m_{ij}), \qquad i,j=1,2, \qquad \begin{array}{l} T^{(ij)} \in \mathbb{R}^{m_{ij}\times n_{ij}}, \\ m_{ij}+n_{ij} \leq N. \end{array}$$

- Since $g_j^{(\alpha)} > g_{j+1}^{(\alpha)} > 0$, $|T^{(ij)}| \geq |A^{(ij)}|$ for every $i,j=1,2$,

HODLR of Grünwald–Letnikov

Proof. and consider the subdiagonal block $T^{(ij)}$ of G_N defined by

$$T^{(ij)} = G_N(N-m_{ij}+1:N, N-m_{ij}-n_{ij}+1:N-m_{ij}), \qquad i,j=1,2, \qquad \begin{array}{l} T^{(ij)} \in \mathbb{R}^{m_{ij}\times n_{ij}}, \\ m_{ij}+n_{ij} \leq N. \end{array}$$

- Since $g_j^{(\alpha)} > g_{j+1}^{(\alpha)} > 0$, $|T^{(ij)}| \geq |A^{(ij)}|$ for every $i,j=1,2$,
- Being $T^{(ij)}$ and $A^{(ij)}$ nonpositive and the 2 norm monotonous, $\|A^{(ij)}\|_2 \leq \|T^{(ij)}\|_2$.

HODLR of Grünwald–Letnikov

Proof. and consider the subdiagonal block $T^{(ij)}$ of G_N defined by

$$T^{(ij)} = G_N(N-m_{ij}+1:N, N-m_{ij}-n_{ij}+1:N-m_{ij}), \qquad i,j=1,2, \qquad \begin{array}{l} T^{(ij)} \in \mathbb{R}^{m_{ij}\times n_{ij}}, \\ m_{ij}+n_{ij} \le N. \end{array}$$

- Since $g_j^{(\alpha)} > g_{j+1}^{(\alpha)} > 0$, $|T^{(ij)}| \ge |A^{(ij)}|$ for every $i,j=1,2$,
- Being $T^{(ij)}$ and $A^{(ij)}$ nonpositive and the 2 norm monotonous, $\|A^{(ij)}\|_2 \le \|T^{(ij)}\|_2$.
- By exploiting

$$\begin{aligned} \|A\|_2 &\le \left\| \begin{bmatrix} A^{(11)} & \\ & A^{(22)} \end{bmatrix} \right\|_2 + \left\| \begin{bmatrix} & A^{(12)} \\ A^{(21)} & \end{bmatrix} \right\|_2 \\ &= \max\{\|A^{(11)}\|_2, \|A^{(22)}\|_2\} + \max\{\|A^{(12)}\|_2, \|A^{(21)}\|_2\} \end{aligned} \quad \Rightarrow \quad \|A\|_2 \le 2\|G_N\|_2.$$

HODLR of Grünwald–Letnikov

Proof. and consider the subdiagonal block $T^{(ij)}$ of G_N defined by

$$T^{(ij)} = G_N(N-m_{ij}+1:N, N-m_{ij}-n_{ij}+1:N-m_{ij}), \qquad i,j=1,2, \qquad \begin{array}{l} T^{(ij)} \in \mathbb{R}^{m_{ij}\times n_{ij}}, \\ m_{ij}+n_{ij} \le N. \end{array}$$

- Since $g_j^{(\alpha)} > g_{j+1}^{(\alpha)} > 0$, $|T^{(ij)}| \ge |A^{(ij)}|$ for every $i,j=1,2$,
- Being $T^{(ij)}$ and $A^{(ij)}$ nonpositive and the 2 norm monotonous, $\|A^{(ij)}\|_2 \le \|T^{(ij)}\|_2$.
- By exploiting

$$\begin{aligned} \|A\|_2 &\le \left\| \begin{bmatrix} A^{(11)} & \\ & A^{(22)} \end{bmatrix} \right\|_2 + \left\| \begin{bmatrix} & A^{(12)} \\ A^{(21)} & \end{bmatrix} \right\|_2 \\ &= \max\{\|A^{(11)}\|_2, \|A^{(22)}\|_2\} + \max\{\|A^{(12)}\|_2, \|A^{(21)}\|_2\} \end{aligned} \quad \Rightarrow \quad \|A\|_2 \le 2\|G_N\|_2.$$

- Conclude by the result on Hankel matrices!

HODLR of Grünwald–Letnikov

Proposition (Massei, Mazza, and Robol 2019, Lemma 3.15)

For every $\epsilon > 0$, the ϵ-qsrank of G_N is bounded by

$$\mathrm{qsrank}_\epsilon(G_N) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right) = 2 + 2\left\lceil \frac{2}{\pi^2} \log\left(\frac{4}{\pi} N\right) \log\left(\frac{32}{\epsilon}\right)\right\rceil.$$

Proof. We call J the $h \times h$ flip matrix, so that $-AJ$ is Hankel and positive semidefinite:

$$\mathrm{rank}_{\frac{\epsilon}{2}}(A) = \mathrm{rank}_{\frac{\epsilon}{2}}(AJ) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right).$$

HODLR of Grünwald–Letnikov

Proposition (Massei, Mazza, and Robol 2019, Lemma 3.15)

For every $\epsilon > 0$, the ϵ-qsrank of G_N is bounded by

$$\mathrm{qsrank}_\epsilon(G_N) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right) = 2 + 2\left\lceil \frac{2}{\pi^2} \log\left(\frac{4}{\pi}N\right) \log\left(\frac{32}{\epsilon}\right)\right\rceil.$$

Proof. We call J the $h \times h$ flip matrix, so that $-AJ$ is Hankel and positive semidefinite:

$$\mathrm{rank}_{\frac{\epsilon}{2}}(A) = \mathrm{rank}_{\frac{\epsilon}{2}}(AJ) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right).$$

Y is a submatrix of A, thus there exists δY such that

$$\|\delta Y\|_2 \leq \varepsilon \|G_N\|_2 \text{ and } \mathrm{rank}(Y + \delta Y) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right).$$

HODLR of Grünwald–Letnikov

Proposition (Massei, Mazza, and Robol 2019, Lemma 3.15)

For every $\epsilon > 0$, the ϵ-qsrank of G_N is bounded by

$$\mathrm{qsrank}_\epsilon(G_N) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right) = 2 + 2\left\lceil \frac{2}{\pi^2} \log\left(\frac{4}{\pi}N\right) \log\left(\frac{32}{\epsilon}\right)\right\rceil.$$

Proof. We call J the $h \times h$ flip matrix, so that $-AJ$ is Hankel and positive semidefinite:

$$\mathrm{rank}_{\frac{\epsilon}{2}}(A) = \mathrm{rank}_{\frac{\epsilon}{2}}(AJ) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right).$$

Y is a submatrix of A, thus there exists δY such that

$$\|\delta Y\|_2 \leq \varepsilon \|G_N\|_2 \text{ and } \mathrm{rank}(Y + \delta Y) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right).$$

$\Rightarrow \mathrm{qsrank}_\epsilon(G_N) \leq \mathfrak{B}\left(N, \frac{\epsilon}{2}\right)$. □

HODLR of Grünwald–Letnikov

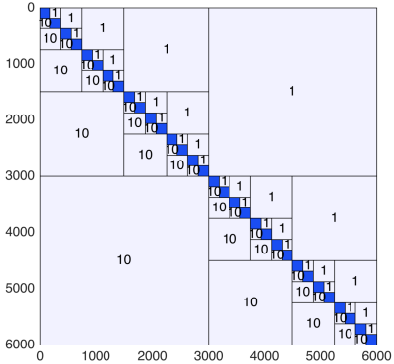
Let's do some experiments with the `hm-toolbox` (Massei, Robol, and Kressner 2020).

```
function G = glhodlrmatrix(N,alpha,tol)
%%GLMATRIX produces the GL discretization of
% the Riemann-Liouville derivative in HODLR
% format
g = gl(N,alpha);
c = zeros(N,1);
r = zeros(1,N);
r(1:2) = g(2:-1:1);
c(1:N) = g(2:end);
hodlroption( 'threshold', tol);
G = hodlr('toeplitz',c,r);
end
```

HODLR of Grünwald–Letnikov

Let's do some experiments with the hm-toolbox (Massei, Robol, and Kressner 2020).

```
function G = glhodlrmatrix(N,alpha,tol)
%%GLMATRIX produces the GL discretization of
% the Riemann-Liouville derivative in HODLR
% format
g = gl(N,alpha);
c = zeros(N,1);
r = zeros(1,N);
r(1:2) = g(2:-1:1);
c(1:N) = g(2:end);
hodlroption( 'threshold', tol);
G = hodlr('toeplitz',c,r);
end
```

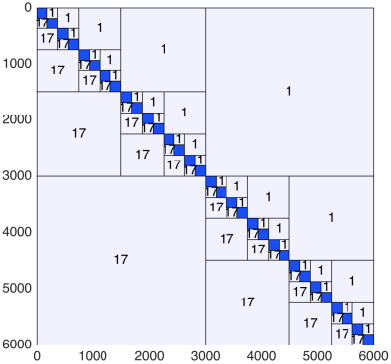



```
G = glhodlrmatrix(6000,1.5,1e-6);
```

HODLR of Grünwald–Letnikov

Let's do some experiments with the hm-toolbox (Massei, Robol, and Kressner 2020).

```
function G = glhodlrmatrix(N,alpha,tol)
%%GLMATRIX produces the GL discretization of
% the Riemann-Liouville derivative in HODLR
% format
g = gl(N,alpha);
c = zeros(N,1);
r = zeros(1,N);
r(1:2) = g(2:-1:1);
c(1:N) = g(2:end);
hodlroption( 'threshold', tol);
G = hodlr('toeplitz',c,r);
end
```

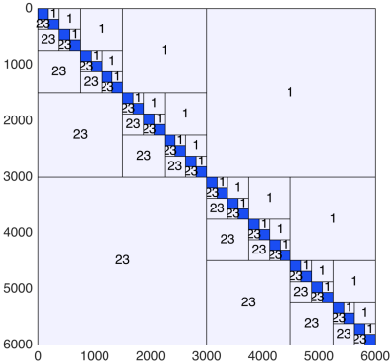



```
G = glhodlrmatrix(6000,1.5,1e-9);
```

HODLR of Grünwald–Letnikov

Let's do some experiments with the `hm-toolbox` (Massei, Robol, and Kressner 2020).

```
function G = glhodlrmatrix(N,alpha,tol)
%%GLMATRIX produces the GL discretization of
% the Riemann-Liouville derivative in HODLR
% format
g = gl(N,alpha);
c = zeros(N,1);
r = zeros(1,N);
r(1:2) = g(2:-1:1);
c(1:N) = g(2:end);
hodlroption( 'threshold', tol);
G = hodlr('toeplitz',c,r);
end
```




```
G = glhodlrmatrix(6000,1.5,1e-12);
```

HODLR Matrix: the whole discretization

Matrix G_N was only a piece of the whole discretization matrix

$$A_N = I_N + \frac{\Delta t}{h^\alpha}\left(D^+_{(m)} G_N + D^-_{(m)} G_N^T\right),$$

does it share the same structure?

HODLR Matrix: the whole discretization

Matrix G_N was only a piece of the whole discretization matrix

$$A_N = I_N + \frac{\Delta t}{h^\alpha}\left(D^+_{(m)} G_N + D^-_{(m)} G_N^T\right),$$

does it share the same structure?

Corollary (Massei, Mazza, and Robol 2019, Corollary 3.16)

$$\mathrm{qsrank}_\epsilon(A_N) \leq 3 + 2\left\lceil \frac{2}{\pi^2}\log\left(\frac{4}{\pi}N\right)\log\left(\frac{32}{\hat{\epsilon}}\right)\right\rceil, \quad \hat{\epsilon} \triangleq \frac{\|A_N\|}{\|G_N\| \cdot \max\{\|D^+_{(m)}\|, \|D^-_{(m)}\|\}}\epsilon.$$

HODLR Matrix: the whole discretization

Matrix G_N was only a piece of the whole discretization matrix

$$A_N = I_N + \frac{\Delta t}{h^\alpha}\left(D^+_{(m)} G_N + D^-_{(m)} G_N^T\right),$$

does it share the same structure?

Corollary (Massei, Mazza, and Robol 2019, Corollary 3.16)

$$\mathrm{qsrank}_\epsilon(A_N) \leq 3 + 2\left\lceil \frac{2}{\pi^2}\log\left(\frac{4}{\pi}N\right)\log\left(\frac{32}{\hat{\epsilon}}\right)\right\rceil, \quad \hat{\epsilon} \triangleq \frac{\|A_N\|}{\|G_N\| \cdot \max\{\|D^+_{(m)}\|, \|D^-_{(m)}\|\}}\epsilon.$$

Proof. Result is invariant under scaling, so assume wlog that $\frac{\Delta t}{h^\alpha} = 1$.

HODLR Matrix: the whole discretization

Matrix G_N was only a piece of the whole discretization matrix

$$A_N = I_N + \frac{\Delta t}{h^\alpha}\left(D^+_{(m)} G_N + D^-_{(m)} G_N^T\right),$$

does it share the same structure?

Corollary (Massei, Mazza, and Robol 2019, Corollary 3.16)

$$\mathrm{qsrank}_\epsilon(A_N) \leq 3 + 2\left\lceil \frac{2}{\pi^2}\log\left(\frac{4}{\pi}N\right)\log\left(\frac{32}{\hat{\epsilon}}\right)\right\rceil, \quad \hat{\epsilon} \triangleq \frac{\|A_N\|}{\|G_N\| \cdot \max\{\|D^+_{(m)}\|, \|D^-_{(m)}\|\}}\epsilon.$$

Proof. Result is invariant under scaling, so assume wlog that $\frac{\Delta t}{h^\alpha} = 1$. A generic off-diagonal block Y, wlog in the lower triangular part, If Y does not intersect the first subdiagonal, is a subblock of $D^+_{(m)} G_N$, so there exists a perturbation δY with norm bounded by $\|\delta Y\| \leq \|D^+_{(m)}\|\|G_N\| \cdot \hat{\epsilon}$ such that $Y + \delta Y$ has rank at most $\mathfrak{B}(N, \hat{\epsilon}/2)$. In particular, δY satisfies $\|\delta Y\| \leq \|A_N\| \cdot \epsilon$.

HODLR Matrix: the whole discretization

Matrix G_N was only a piece of the whole discretization matrix

$$A_N = I_N + \frac{\Delta t}{h^\alpha}\left(D^+_{(m)} G_N + D^-_{(m)} G_N^T\right),$$

does it share the same structure?

Corollary (Massei, Mazza, and Robol 2019, Corollary 3.16)

$$\mathrm{qsrank}_\epsilon(A_N) \leq 3 + 2\left\lceil \frac{2}{\pi^2}\log\left(\frac{4}{\pi}N\right)\log\left(\frac{32}{\hat{\epsilon}}\right)\right\rceil, \quad \hat{\epsilon} \triangleq \frac{\|A_N\|}{\|G_N\| \cdot \max\{\|D^+_{(m)}\|, \|D^-_{(m)}\|\}}\epsilon.$$

Proof. Result is invariant under scaling, so assume wlog that $\frac{\Delta t}{h^\alpha} = 1$. Since we have excluded one subdiagonal, a generic off-diagonal block Y we can find a perturbation with norm bounded by $\|A_N\| \cdot \epsilon$ such that $Y + \delta Y$ has rank $1 + \mathfrak{B}(N, \hat{\epsilon}/2)$. □

A HODLR right-hand side

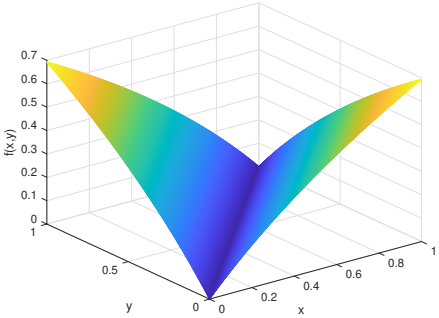
❓ What are right-hand sides functions $f(x, y, t)$ so that the matrix C has a HODLR structure?

A HODLR right-hand side

❓ What are right-hand sides functions $f(x, y, t)$ so that the matrix C has a HODLR structure?

Consider the function

$$f(x, y) = \log(\tau + |x - y|), \quad \tau > 0.$$



A HODLR right-hand side

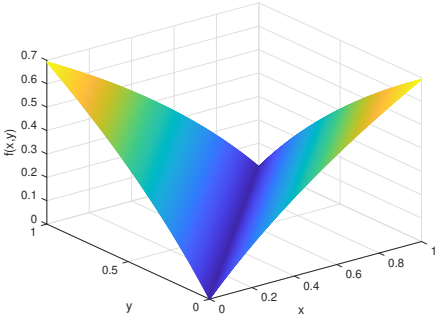
❓ What are right-hand sides functions $f(x, y, t)$ so that the matrix C has a HODLR structure?

Consider the function

$$f(x, y) = \log(\tau + |x - y|), \quad \tau > 0.$$

⚙ If we discretize it by *finite differences* on a rectangular domain we find

$$C_{i,j} = \log(\tau + |x_i - y_j|)$$



A HODLR right-hand side

❓ What are right-hand sides functions $f(x, y, t)$ so that the matrix C has a HODLR structure?

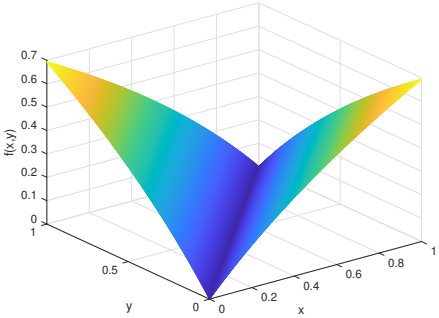
Consider the function

$$f(x, y) = \log(\tau + |x - y|), \quad \tau > 0.$$

⚙ If we discretize it by *finite differences* on a rectangular domain we find

$$C_{i,j} = \log(\tau + |x_i - y_j|)$$

💡 The modulus function it is **not regular in the whole domain** but it is **analytic** when the **sign** of $x - y$ is **constant**.



A HODLR right-hand side

❓ What are right-hand sides functions $f(x, y, t)$ so that the matrix C has a HODLR structure?

Consider the function

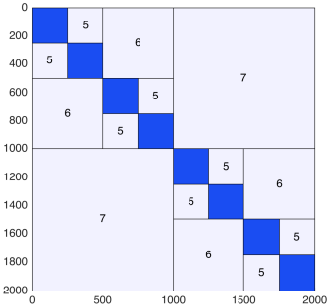
$$f(x, y) = \log(\tau + |x - y|), \quad \tau > 0.$$

⚙ If we discretize it by *finite differences* on a rectangular domain we find

$$C_{i,j} = \log(\tau + |x_i - y_j|)$$

💡 The modulus function it is **not regular in the whole domain** but it is **analytic** when the **sign** of $x - y$ is **constant**.

🔨 We can use again Chebyshev basis to approximate it in a separable fashion.




```
x = linspace(0,1,N); y = linspace(0,1,N);
[X,Y] = meshgrid(x,y); tau = 1;
C = log(tau + abs(X-Y)); hC = hodlr(C);
```

Separability (a bit more formally)

Separable expansion (Hackbusch 2015, Definition 4.4)

Take a function $\chi(x, y) : X \times Y \to \mathbb{R}$, we call

$$\chi(x, y) = \sum_{\nu=1}^{r} \phi_\nu^{(r)}(x)\psi_\nu^{(r)}(y) + R_r(x, y), \qquad \text{for } x \in X,\ y \in Y,$$

a *separable expansion* of χ with r terms in $X \times Y$ with remainder R_r.

- To have an idea of the **goodness** of the *separable expansion*, we would like to have $\{\|R_r\|_\infty, \|R_r\|_{\mathbb{L}^p}\} \xrightarrow{r\to 0} 0$ as fast as possible, e.g., **exponentially**.

Separability (a bit more formally)

Separable expansion (Hackbusch 2015, Definition 4.4)

Take a function $\chi(x, y) : X \times Y \to \mathbb{R}$, we call

$$\chi(x, y) = \sum_{\nu=1}^{r} \phi_\nu^{(r)}(x)\psi_\nu^{(r)}(y) + R_r(x, y), \qquad \text{for } x \in X,\ y \in Y,$$

a *separable expansion* of χ with r terms in $X \times Y$ with remainder R_r.

- To have an idea of the **goodness** of the *separable expansion*, we would like to have $\{\|R_r\|_\infty, \|R_r\|_{\mathbb{L}^p}\} \xrightarrow{r\to 0} 0$ as fast as possible, e.g., **exponentially**.
- If $\|R_r\| \leq c_1 \exp(-c_2 r^\alpha) \Rightarrow \|R_r\| \leq \varepsilon$ if $r \geq \left\lceil \left(\frac{1}{c_2} \log^{1/\alpha} \frac{c_1}{\varepsilon} \right) \right\rceil = O(\log^{1/\alpha} 1/\varepsilon)$ $\varepsilon \to 0$.

Separability (a bit more formally)

Separable expansion (Hackbusch 2015, Definition 4.4)

Take a function $\chi(x, y) : X \times Y \to \mathbb{R}$, we call

$$\chi(x, y) = \sum_{\nu=1}^{r} \phi_\nu^{(r)}(x)\psi_\nu^{(r)}(y) + R_r(x, y), \qquad \text{for } x \in X,\ y \in Y,$$

a *separable expansion* of χ with r terms in $X \times Y$ with remainder R_r.

- To have an idea of the **goodness** of the *separable expansion*, we would like to have $\{\|R_r\|_\infty, \|R_r\|_{\mathbb{L}^p}\} \xrightarrow{r\to 0} 0$ as fast as possible, e.g., **exponentially**.
- If $\|R_r\| \leq c_1 \exp(-c_2 r^\alpha) \Rightarrow \|R_r\| \leq \varepsilon$ if $r \geq \left\lceil \left(\frac{1}{c_2} \log^{1/\alpha} \frac{c_1}{\varepsilon} \right) \right\rceil = O(\log^{1/\alpha} 1/\varepsilon)$ $\varepsilon \to 0$.
- We can use Taylor expansions, Chebyshev expansion, Hermite/Lagrange interpolation, cross approximation... In all the cases, the behavior of R_r is tied to the regularity of $\chi(x, y)$; see (Hackbusch 2015, Chapter 4).

BLAS with HODLR format

❓ We now have **everything represented in the right format**, but can we operate with it?

BLAS with HODLR format

❓ We now have **everything represented in the right format**, but can we operate with it?

$\mathbf{y} = A\mathbf{x}$: Matrix-vector products, *recursively*:

$$\mathbf{y}(I_1^1) = A(I_1^1, I_1^1)\mathbf{x}(I_1^1) + A(I_1^1, I_2^1)\mathbf{x}(I_2^1),$$
$$\mathbf{y}(I_2^1) = A(I_2^1, I_1^1)\mathbf{x}(I_1^1) + A(I_2^1, I_2^1)\mathbf{x}(I_2^1).$$

BLAS with HODLR format

❓ We now have **everything represented in the right format**, but can we operate with it?

$\mathbf{y} = A\mathbf{x}$: Matrix-vector products, *recursively*:

$$\mathbf{y}(I_1^1) = A(I_1^1, I_1^1)\mathbf{x}(I_1^1) + A(I_1^1, I_2^1)\mathbf{x}(I_2^1),$$
$$\mathbf{y}(I_2^1) = A(I_2^1, I_1^1)\mathbf{x}(I_1^1) + A(I_2^1, I_2^1)\mathbf{x}(I_2^1).$$

⚙ Off-diagonal blocks $A(I_1^1, I_2^1)$ and $A(I_2^1, I_1^1)$ are obtained by multiplying $n/2 \times n/2$ low-rank matrix with vector. This **cost** $c_{\mathrm{LR}\cdot\mathbf{x}}(n/2) = 2nk$.

BLAS with HODLR format

❓ We now have **everything represented in the right format**, but can we operate with it?

$\mathbf{y} = A\mathbf{x}$: Matrix-vector products, *recursively*:

$$\mathbf{y}(I_1^1) = A(I_1^1, I_1^1)\mathbf{x}(I_1^1) + A(I_1^1, I_2^1)\mathbf{x}(I_2^1),$$
$$\mathbf{y}(I_2^1) = A(I_2^1, I_1^1)\mathbf{x}(I_1^1) + A(I_2^1, I_2^1)\mathbf{x}(I_2^1).$$

- ⚙ Off-diagonal blocks $A(I_1^1, I_2^1)$ and $A(I_2^1, I_1^1)$ are obtained by multiplying $n/2 \times n/2$ low-rank matrix with vector. This **cost** $c_{\mathrm{LR}\cdot\mathbf{x}}(n/2) = 2nk$.
- ⚙ Diagonal blocks are processed recursively at a **cost**

$$c_{A\cdot\mathbf{x}}(n) = 2c_{A\cdot\mathbf{x}}(n/2) + 4kn + n.$$

BLAS with HODLR format

❓ We now have **everything represented in the right format**, but can we operate with it?

$\mathbf{y} = A\mathbf{x}$: Matrix-vector products, *recursively*:

$$\mathbf{y}(I_1^1) = A(I_1^1, I_1^1)\mathbf{x}(I_1^1) + A(I_1^1, I_2^1)\mathbf{x}(I_2^1),$$
$$\mathbf{y}(I_2^1) = A(I_2^1, I_1^1)\mathbf{x}(I_1^1) + A(I_2^1, I_2^1)\mathbf{x}(I_2^1).$$

- ⚙ Off-diagonal blocks $A(I_1^1, I_2^1)$ and $A(I_2^1, I_1^1)$ are obtained by multiplying $n/2 \times n/2$ low-rank matrix with vector. This **cost** $c_{\mathrm{LR}\cdot\mathbf{x}}(n/2) = 2nk$.
- ⚙ Diagonal blocks are processed recursively at a **cost**

$$c_{A\cdot\mathbf{x}}(n) = 2c_{A\cdot\mathbf{x}}(n/2) + 4kn + n.$$

Master theorem (*divide and conquer*): $c_{A\cdot\mathbf{x}}(n) = (4k+1)\log_2(n)n$.

BLAS with HODLR format

$C = A + B$: Adding two equally partitioned HODLR matrices **increases the ranks** of off-diagonal blocks by a factor 2.

BLAS with HODLR format

$C = A + B$: Adding two equally partitioned HODLR matrices **increases the ranks** of off-diagonal blocks by a factor 2.

- We need truncation $\mathfrak{T}_k(A(I_1^\ell, I_j^\ell) + B(I_1^\ell, I_j^\ell))$, costs

$$c_{\text{LR+LR}} = c_{\text{SVD}} \times (nk^2 + k^3),$$

where c_{SVD} is the cost of the given low-rank truncation algorithm (SVD, rand-SVD, QR, ...)

BLAS with HODLR format

$C = A + B$: Adding two equally partitioned HODLR matrices **increases the ranks** of off-diagonal blocks by a factor 2.

- We need truncation $\mathfrak{T}_k(A(I_1^\ell, I_j^\ell) + B(I_1^\ell, I_j^\ell))$, costs

$$c_{\text{LR+LR}} = c_{\text{SVD}} \times (nk^2 + k^3),$$

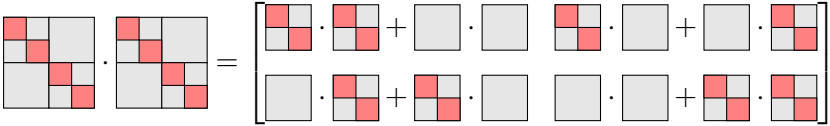
where c_{SVD} is the cost of the given low-rank truncation algorithm (SVD, rand-SVD, QR, ...)

Total cost is then:

$$\begin{aligned}\sum_{\ell=1}^{p} 2^\ell c_{\text{LR+LR}}(m_\ell) =& c_{\text{SVD}} \sum_{\ell=1}^{p} 2^\ell (k^3 + m_\ell k^2) \\ \leq& c_{\text{SVD}} \left(2^{p+1} k^3 + \sum_{\ell=1}^{p} 2^\ell 2^{p-\ell} n_0 k^2 \right) \\ \leq& c_{\text{SVD}} \left(2nk^3 + n \log_2(n) k^2 \right).\end{aligned}$$

BLAS with HODLR format

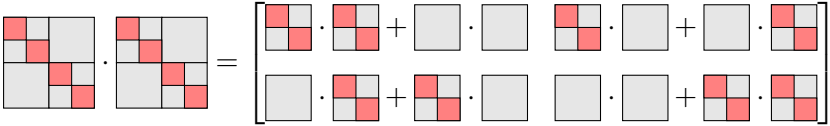
$C = AB$: Matrix-matrix multiplication can also be done recursively



where ⊞ is a $n/2 \times n/2$ HODLR matrix and □ is a low-rank block.

BLAS with HODLR format

$C = AB$: Matrix-matrix multiplication can also be done recursively

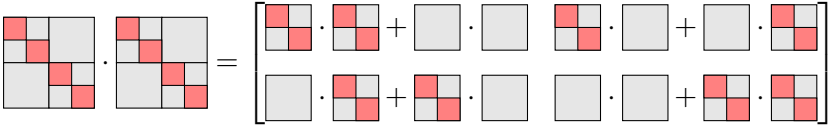


where ⊞ is a $n/2 \times n/2$ HODLR matrix and □ is a low-rank block.

1. ⊞ · ⊞ · of 2 HODLR $n/2$ matrices,

BLAS with HODLR format

$C = AB$: Matrix-matrix multiplication can also be done recursively

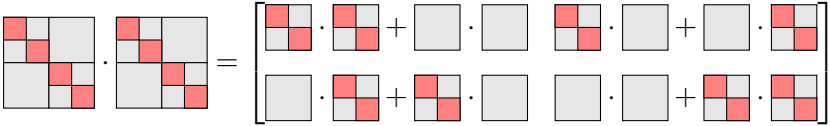


where ⊞ is a $n/2 \times n/2$ HODLR matrix and □ is a low-rank block.

1. ⊞ · ⊞ · of 2 HODLR $n/2$ matrices,
2. □ · □ · of 2 low-rank blocks,

BLAS with HODLR format

$C = AB$: Matrix-matrix multiplication can also be done recursively

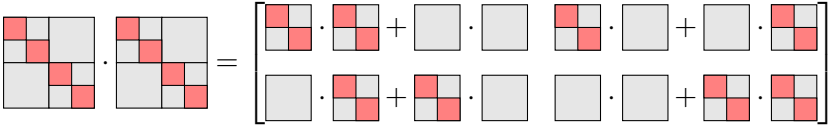


where ⊞ is a $n/2 \times n/2$ HODLR matrix and □ is a low-rank block.

1. ⊞ · ⊞ · of 2 HODLR $n/2$ matrices,
2. □ · □ · of 2 low-rank blocks,
3. ⊞ · □ · of HODLR times low-rank,

BLAS with HODLR format

$C = AB$: Matrix-matrix multiplication can also be done recursively

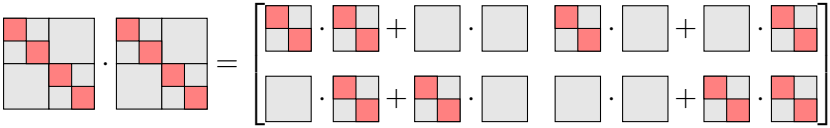


where ⊞ is a $n/2 \times n/2$ HODLR matrix and □ is a low-rank block.

1. ⊞ · ⊞ · of 2 HODLR $n/2$ matrices,
2. □ · □ · of 2 low-rank blocks,
3. ⊞ · □ · of HODLR times low-rank,
4. □ · ⊞ · of low-rank times HODLR.

BLAS with HODLR format

$C = AB$: Matrix-matrix multiplication can also be done recursively



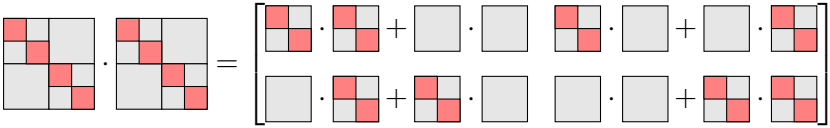
where ⊞ is a $n/2 \times n/2$ HODLR matrix and □ is a low-rank block.

1. ⊞ · ⊞ · of 2 HODLR $n/2$ matrices,
2. □ · □ · of 2 low-rank blocks,
3. ⊞ · □ · of HODLR times low-rank,
4. □ · ⊞ · of low-rank times HODLR.

$$c_{H\cdot H}(n) = 2\left(c_{H\cdot H}(n/2) + c_{\mathrm{LR}\cdot\mathrm{LR}}(n/2) + c_{H\cdot\mathrm{LR}}(n/2) + c_{\mathrm{LR}\cdot H}(n/2) + c_{H+\mathrm{LR}}(n/2) + c_{\mathrm{LR}+\mathrm{LR}}(n/2)\right)$$

BLAS with HODLR format

$C = AB$: Matrix-matrix multiplication can also be done recursively



where ⊞ is a $n/2 \times n/2$ HODLR matrix and □ is a low-rank block.

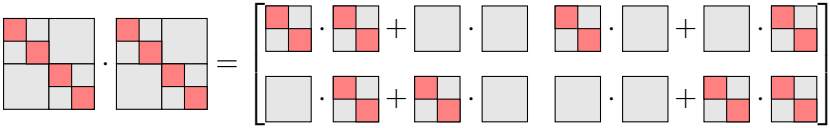
1. ⊞ · ⊞ · of 2 HODLR $n/2$ matrices,
2. □ · □ · of 2 low-rank blocks,
3. ⊞ · □ · of HODLR times low-rank,
4. □ · ⊞ · of low-rank times HODLR.

$$c_{H\cdot H}(n) = 2\left(c_{H\cdot H}(n/2) + c_{\mathrm{LR}\cdot\mathrm{LR}}(n/2) + c_{H\cdot\mathrm{LR}}(n/2) + c_{\mathrm{LR}\cdot H}(n/2) + c_{H+\mathrm{LR}}(n/2) + c_{\mathrm{LR}+\mathrm{LR}}(n/2)\right)$$

$c_{\mathrm{LR}\cdot\mathrm{LR}}(n) = 4nk^2$

BLAS with HODLR format

$C = AB$: Matrix-matrix multiplication can also be done recursively



where ⊞ is a $n/2 \times n/2$ HODLR matrix and □ is a low-rank block.

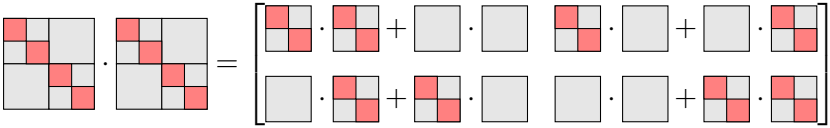
1. ⊞ · ⊞ · of 2 HODLR $n/2$ matrices,
2. □ · □ · of 2 low-rank blocks,
3. ⊞ · □ · of HODLR times low-rank,
4. □ · ⊞ · of low-rank times HODLR.

$$c_{H\cdot H}(n) = 2\left(c_{H\cdot H}(n/2) + c_{\mathrm{LR}\cdot\mathrm{LR}}(n/2) + c_{H\cdot\mathrm{LR}}(n/2) + c_{\mathrm{LR}\cdot H}(n/2) + c_{H+\mathrm{LR}}(n/2) + c_{\mathrm{LR}+\mathrm{LR}}(n/2)\right)$$

$$c_{H\cdot\mathrm{LR}}(n) = c_{\mathrm{LR}\cdot H} = k c_{H\mathrm{v}}(n) = k(4k+1)\log_2(n)n$$

BLAS with HODLR format

$C = AB$: Matrix-matrix multiplication can also be done recursively



where ⊞ is a $n/2 \times n/2$ HODLR matrix and □ is a low-rank block.

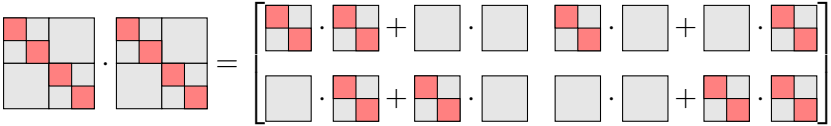
1. ⊞ · ⊞ · of 2 HODLR $n/2$ matrices,
2. □ · □ · of 2 low-rank blocks,
3. ⊞ · □ · of HODLR times low-rank,
4. □ · ⊞ · of low-rank times HODLR.

$$c_{H\cdot H}(n) = 2\left(c_{H\cdot H}(n/2) + c_{\mathrm{LR}\cdot\mathrm{LR}}(n/2) + c_{H\cdot\mathrm{LR}}(n/2) + c_{\mathrm{LR}\cdot H}(n/2) + c_{H+\mathrm{LR}}(n/2) + c_{\mathrm{LR}+\mathrm{LR}}(n/2)\right)$$

$$c_{H+\mathrm{LR}}(n) = c_{H+H}(n) = c_{\mathrm{SVD}}(nk^3 + n\log(n)k^2)$$

BLAS with HODLR format

$C = AB$: Matrix-matrix multiplication can also be done recursively



where ⊞ is a $n/2 \times n/2$ HODLR matrix and □ is a low-rank block.

1. ⊞ · ⊞ · of 2 HODLR $n/2$ matrices,
2. □ · □ · of 2 low-rank blocks,
3. ⊞ · □ · of HODLR times low-rank,
4. □ · ⊞ · of low-rank times HODLR.

Total cost $c_{H \cdot H}(n) \in O(k^3 n \log n + k^2 n \log^2 n)$.

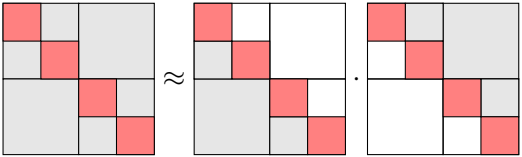
BLAS with HODLR format

Approximate solution of a linear system $A\mathbf{x} = \mathbf{b}$ with HODLR matrix A:

BLAS with HODLR format

Approximate solution of a linear system $A\mathbf{x} = \mathbf{b}$ with HODLR matrix A:

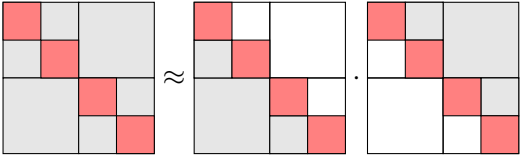
$A \approx LU$ Approximate LU-factorization $A \approx LU$ in HODLR format:



BLAS with HODLR format

Approximate solution of a linear system $A\mathbf{x} = \mathbf{b}$ with HODLR matrix A:

$A \approx LU$ Approximate LU-factorization $A \approx LU$ in HODLR format:



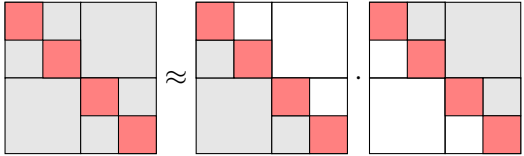
Forward substitution to solve $L\mathbf{y} = \mathbf{b}$,

Backward substitution to solve $U\mathbf{x} = \mathbf{y}$.

BLAS with HODLR format

Approximate solution of a linear system $A\mathbf{x} = \mathbf{b}$ with HODLR matrix A:

$A \approx LU$ Approximate LU-factorization $A \approx LU$ in HODLR format:



Forward substitution to solve $L\mathbf{y} = \mathbf{b}$,

Backward substitution to solve $U\mathbf{x} = \mathbf{y}$.

We need to analyze the two steps separately.

BLAS with HODLR format

Approximate LU factorization, on level $\ell = 1$:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad U = \begin{bmatrix} U_{11} & U_{12} \\ O & U_{22} \end{bmatrix}$$

It is done in four steps

BLAS with HODLR format

Approximate LU factorization, on level $\ell = 1$:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad U = \begin{bmatrix} U_{11} & U_{12} \\ O & U_{22} \end{bmatrix}$$

It is done in four steps

1. Compute LU factors L_{11}, U_{11} of A_{11},

BLAS with HODLR format

Approximate LU factorization, on level $\ell = 1$:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad U = \begin{bmatrix} U_{11} & U_{12} \\ O & U_{22} \end{bmatrix}$$

It is done in four steps

1. Compute LU factors L_{11}, U_{11} of A_{11},
2. Compute $U_{12} = L_{11}^{-1} A_{12}$ by **forward substitution**,

BLAS with HODLR format

Approximate LU factorization, on level $\ell = 1$:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad U = \begin{bmatrix} U_{11} & U_{12} \\ O & U_{22} \end{bmatrix}$$

It is done in four steps

1. Compute LU factors L_{11}, U_{11} of A_{11},
2. Compute $U_{12} = L_{11}^{-1} A_{12}$ by **forward substitution**,
3. Compute $L_{21} = A_{21} U_{11}^{-1}$ by **backward substitution**,

BLAS with HODLR format

Approximate LU factorization, on level $\ell = 1$:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad U = \begin{bmatrix} U_{11} & U_{12} \\ O & U_{22} \end{bmatrix}$$

It is done in four steps

1. Compute LU factors L_{11}, U_{11} of A_{11},
2. Compute $U_{12} = L_{11}^{-1} A_{12}$ by **forward substitution**,
3. Compute $L_{21} = A_{21} U_{11}^{-1}$ by **backward substitution**,
4. Compute LU factors L_{22}, U_{22} of $A_{22} - L_{21} U_{12}$.

BLAS with HODLR format

Approximate LU factorization, on level $\ell = 1$:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad U = \begin{bmatrix} U_{11} & U_{12} \\ O & U_{22} \end{bmatrix}$$

It is done in four steps

1. Compute LU factors L_{11}, U_{11} of A_{11},
2. Compute $U_{12} = L_{11}^{-1} A_{12}$ by **forward substitution**,
3. Compute $L_{21} = A_{21} U_{11}^{-1}$ by **backward substitution**,
4. Compute LU factors L_{22}, U_{22} of $A_{22} - L_{21} U_{12}$.

The analysis of the cost is *analogous to the matrix-matrix multiplication case*, **but** we need to know how to do and how-much does forward/backward substitution costs.

BLAS with HODLR format

Forward substitution with lower triangular L in HODLR format: $\mathbf{y} = L^{-1}\mathbf{b}$

$$L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

with L_{21} low-rank, and L_{11}, L_{22} HODLR.

BLAS with HODLR format

Forward substitution with lower triangular L in HODLR format: $\mathbf{y} = L^{-1}\mathbf{b}$

$$L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

with L_{21} low-rank, and L_{11}, L_{22} HODLR.

1. Solve $L_{11}\mathbf{y}_1 = \mathbf{b}_1$,

BLAS with HODLR format

Forward substitution with lower triangular L in HODLR format: $\mathbf{y} = L^{-1}\mathbf{b}$

$$L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

with L_{21} low-rank, and L_{11}, L_{22} HODLR.

1. Solve $L_{11}\mathbf{y}_1 = \mathbf{b}_1$,
2. Compute $\tilde{\mathbf{b}}_2 = \mathbf{b}_2 - L_{21}\mathbf{y}_1$,

BLAS with HODLR format

Forward substitution with lower triangular L in HODLR format: $\mathbf{y} = L^{-1}\mathbf{b}$

$$L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

with L_{21} low-rank, and L_{11}, L_{22} HODLR.

1. Solve $L_{11}\mathbf{y}_1 = \mathbf{b}_1$,
2. Compute $\tilde{\mathbf{b}}_2 = \mathbf{b}_2 - L_{21}\mathbf{y}_1$,
3. Solve $L_{22}\mathbf{y}_2 = \tilde{\mathbf{b}}_2$.

BLAS with HODLR format

Forward substitution with lower triangular L in HODLR format: $\mathbf{y} = L^{-1}\mathbf{b}$

$$L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

with L_{21} low-rank, and L_{11}, L_{22} HODLR.

1. Solve $L_{11}\mathbf{y}_1 = \mathbf{b}_1$,
2. Compute $\tilde{\mathbf{b}}_2 = \mathbf{b}_2 - L_{21}\mathbf{y}_1$,
3. Solve $L_{22}\mathbf{y}_2 = \tilde{\mathbf{b}}_2$.

Cost recursively:

$$c_{\text{forw}} = 2c_{\text{forw}}(n/2) + (2k+1)n.$$

BLAS with HODLR format

Forward substitution with lower triangular L in HODLR format: $\mathbf{y} = L^{-1}\mathbf{b}$

$$L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

with L_{21} low-rank, and L_{11}, L_{22} HODLR.

1. Solve $L_{11}\mathbf{y}_1 = \mathbf{b}_1$,
2. Compute $\tilde{\mathbf{b}}_2 = \mathbf{b}_2 - L_{21}\mathbf{y}_1$,
3. Solve $L_{22}\mathbf{y}_2 = \tilde{\mathbf{b}}_2$.

Cost recursively:

$$c_{\text{forw}} = 2c_{\text{forw}}(n/2) + (2k+1)n.$$

On level $\ell = p$, we have the direct solution of $2^p = n/n_0$ linear systems of size $n_0 \times n_0$.

BLAS with HODLR format

Forward substitution with lower triangular L in HODLR format: $\mathbf{y} = L^{-1}\mathbf{b}$

$$L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

with L_{21} low-rank, and L_{11}, L_{22} HODLR.

1. Solve $L_{11}\mathbf{y}_1 = \mathbf{b}_1$,
2. Compute $\tilde{\mathbf{b}}_2 = \mathbf{b}_2 - L_{21}\mathbf{y}_1$,
3. Solve $L_{22}\mathbf{y}_2 = \tilde{\mathbf{b}}_2$.

Cost recursively:

$$c_{\text{forw}} = 2c_{\text{forw}}(n/2) + (2k+1)n.$$

On level $\ell = p$, we have the direct solution of $2^p = n/n_0$ linear systems of size $n_0 \times n_0$.

Total cost $c_{\text{forw}} \in O(kn\log(n))$, and analogously for backward substitution.

BLAS with HODLR format

Forward substitution with lower triangular L in HODLR format: $\mathbf{y} = L^{-1}\mathbf{b}$

$$L = \begin{bmatrix} L_{11} & O \\ L_{21} & L_{22} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$$

with L_{21} low-rank, and L_{11}, L_{22} HODLR.

1. Solve $L_{11}\mathbf{y}_1 = \mathbf{b}_1$,
2. Compute $\tilde{\mathbf{b}}_2 = \mathbf{b}_2 - L_{21}\mathbf{y}_1$,
3. Solve $L_{22}\mathbf{y}_2 = \tilde{\mathbf{b}}_2$.

Cost recursively:

$$c_{\text{forw}} = 2c_{\text{forw}}(n/2) + (2k+1)n.$$

On level $\ell = p$, we have the direct solution of $2^p = n/n_0$ linear systems of size $n_0 \times n_0$.

- Total cost $c_{\text{forw}} \in O(kn\log(n))$, and analogously for backward substitution.
- Total cost $c_{\text{LU}}(n) \lesssim c_{H\cdot H}(n) \in O(k^3 n \log n + k^2 n \log^2 n)$.

BLAS with HODLR format

The `hm-toolbox` (Massei, Robol, and Kressner 2020) contains all the routines.

- They *overload* the standard MATLAB operation by the same name, i.e., if you have variables in the right class you operate directly in this format.
- One can use different **cluster tree** $\mathcal{T}_p$ to get smaller ranks. They are determined by the partitioning of the index set on the leaf level and represented as the vector $\mathbf{c} = [n_1^{(p)}, \ldots, n_{2^p}^{(p)}]$, change it to change the HODLR matrix.

Operation	HODLR complexity
`A*v`	$\mathcal{O}(kn \log n)$
`A\v`	$\mathcal{O}(k^2 n \log^2 n)$
`A+B`	$\mathcal{O}(k^2 n \log n)$
`A*B`	$\mathcal{O}(k^2 n \log^2 n)$
`A\B`	$\mathcal{O}(k^2 n \log^2 n)$
`inv(A)`	$\mathcal{O}(k^2 n \log^2 n)$
`A.*B`[2]	$\mathcal{O}(k^4 n \log n)$
`lu(A), chol(A)`	$\mathcal{O}(k^2 n \log^2 n)$
`qr(A)`	$\mathcal{O}(k^2 n \log^2 n)$
compression	$\mathcal{O}(k^2 n \log(n))$

[2] The complexity of the Hadamard product is dominated by the recompression stage due to the k^2 HODLR rank of $A \circ B$. Without recompression the cost is $\mathcal{O}(k^2 n \log n)$.

HODLR solver for the 1D case

We can modify our first example to get a solution for the 1D problem in the new format.

```
%% Discretization
N = 2^7; hN = 1/(N-1);  x = 0:hN:1; dt = hN;
alpha = 1.5; % Coefficients
dplus=@(x)gamma(3-alpha).*x.^alpha;
dminus=@(x)gamma(3-alpha).*(1-x).^alpha;
w = @(x) 5*x.*(1-x);
tol = 1e-9; % HODLR building
tic;
G = glhodlrmatrix(N,alpha,tol);
Dplus = hodlr('diagonal',dplus(x));
Dminus = hodlr('diagonal',dminus(x));
I = hodlr('eye', N);
nu = hN^alpha/dt;
A = nu*I -(Dplus*G + Dminus*G');
buildtime = toc;
```

```
%% Solving
[L,U] = lu(A);
flu = @() lu(A);
timelu = timeit(flu,2);
w = w(x).';
solvetime = 0;
for i=1:N
 tic;
 w = U\(L\(nu*w));
 solvetime = solvetime + toc;
end
solvetime = solvetime/N;
```

HODLR solver for the 1D case

We can modify our first example to get a solution for the 1D problem in the new format.

```
%% Discretization
N = 2^7; hN = 1/(N-1);  x = 0:hN:1; dt = hN;
alpha = 1.5; % Coefficients
dplus=@(x)gamma(3-alpha).*x.^alpha;
dminus=@(x)gamma(3-alpha).*(1-x).^alpha;
w = @(x) 5*x.*(1-x);
tol = 1e-9; % HODLR building
tic;
G = glhodlrmatrix(N,alpha,tol);
Dplus = hodlr('diagonal',dplus(x));
Dminus = hodlr('diagonal',dminus(x));
I = hodlr('eye', N);
nu = hN^alpha/dt;
A = nu*I -(Dplus*G + Dminus*G');
buildtime = toc;
```

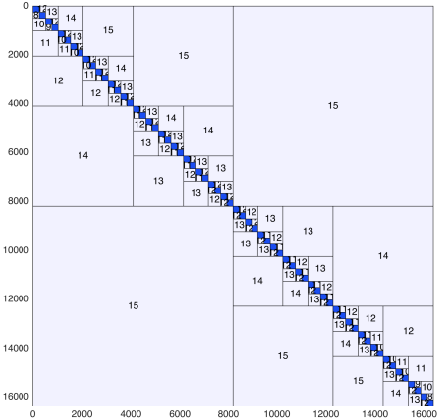
```
%% Solving
[L,U] = lu(A);
flu = @() lu(A);
timelu = timeit(flu,2);
w = w(x).';
solvetime = 0;
for i=1:N
 tic;
 w = U\(L\(nu*w));
 solvetime = solvetime + toc;
end
solvetime = solvetime/N;
```

Let us try looking at the timings.

HODLR solver for the 1D case

We take $\alpha = 1.5$, and $\varepsilon = 10^{-9}$

N	Build (s)	LU (s)	Avg. Solve (s)
2^7	8.96e-03	1.44e-04	2.93e-04
2^8	1.35e-02	4.63e-04	3.33e-04
2^9	3.14e-02	2.05e-03	5.41e-04
2^{10}	7.28e-02	6.21e-03	9.35e-04
2^{11}	1.59e-01	1.63e-02	1.75e-03
2^{12}	3.85e-01	4.33e-02	3.68e-03
2^{13}	8.81e-01	1.27e-01	7.99e-03
2^{14}	2.19e+00	3.73e-01	1.55e-02

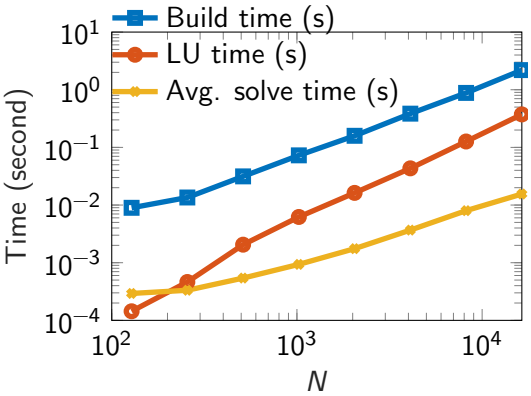


Largest matrix occupies 46.25 Mb, against the 2 Gb of the dense storage and the 0.87 Mb of storing three diagonals and $2 \times (2N - 1)$ for the Toeplitz storage.

HODLR solver for the 1D case

We take $\alpha = 1.5$, and $\varepsilon = 10^{-9}$

N	Build (s)	LU (s)	Avg. Solve (s)
2^7	8.96e-03	1.44e-04	2.93e-04
2^8	1.35e-02	4.63e-04	3.33e-04
2^9	3.14e-02	2.05e-03	5.41e-04
2^{10}	7.28e-02	6.21e-03	9.35e-04
2^{11}	1.59e-01	1.63e-02	1.75e-03
2^{12}	3.85e-01	4.33e-02	3.68e-03
2^{13}	8.81e-01	1.27e-01	7.99e-03
2^{14}	2.19e+00	3.73e-01	1.55e-02

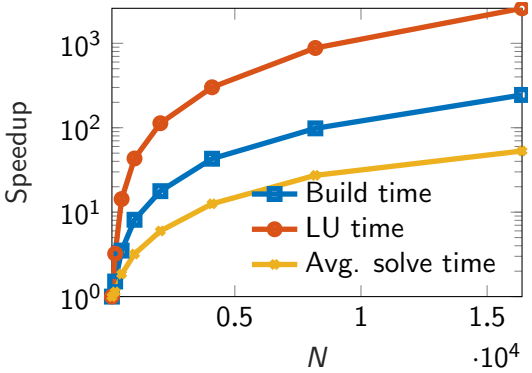


Largest matrix occupies 46.25 Mb, against the 2 Gb of the dense storage and the 0.87 Mb of storing three diagonals and $2 \times (2N - 1)$ for the Toeplitz storage.

HODLR solver for the 1D case

We take $\alpha = 1.5$, and $\varepsilon = 10^{-9}$

N	Build (s)	LU (s)	Avg. Solve (s)
2^7	8.96e-03	1.44e-04	2.93e-04
2^8	1.35e-02	4.63e-04	3.33e-04
2^9	3.14e-02	2.05e-03	5.41e-04
2^{10}	7.28e-02	6.21e-03	9.35e-04
2^{11}	1.59e-01	1.63e-02	1.75e-03
2^{12}	3.85e-01	4.33e-02	3.68e-03
2^{13}	8.81e-01	1.27e-01	7.99e-03
2^{14}	2.19e+00	3.73e-01	1.55e-02



Largest matrix occupies 46.25 Mb, against the 2 Gb of the dense storage and the 0.87 Mb of storing three diagonals and $2 \times (2N - 1)$ for the Toeplitz storage.

Back to Sylvester (Massei, Palitta, and Robol 2018)

To solve the Sylvester equation with HODLR coefficients

$$AX + XB^T = C, \qquad A \in \mathbb{R}^{n\times n}, \;\; B \in \mathbb{R}^{m\times m}, \;\; X, C \in \mathbb{R}^{n\times m},$$

we can use the integral formulation

$$X = \int_0^{+\infty} e^{-At} C e^{-B^T t} \, \mathrm{d}t.$$

Back to Sylvester (Massei, Palitta, and Robol 2018)

To solve the Sylvester equation with HODLR coefficients

$$AX + XB^T = C, \qquad A \in \mathbb{R}^{n\times n}, \;\; B \in \mathbb{R}^{m\times m}, \;\; X, C \in \mathbb{R}^{n\times m},$$

we can use the integral formulation

$$X = \int_0^{+\infty} e^{-At} C e^{-B^T t} \, \mathrm{d}t.$$

We perform the *change of variables*: $t = f(\theta) \triangleq L \cdot \cot\left(\frac{\theta}{2}\right)^2$, rewriting the integral as

$$X = 2L \int_0^{\pi} \frac{\sin(\theta)}{(1-\cos(\theta))^2} \; e^{-Af(\theta)} C e^{-B^T f(\theta)} \, \mathrm{d}\theta,$$

with L a parameter to be optimized for convergence.

Back to Sylvester (Massei, Palitta, and Robol 2018)

We now have an integral on a finite domain $\Rightarrow$ **Gauss-Legendre quadrature**

$$X \approx \sum_{j=1}^{m} \omega_j \cdot e^{-Af(\theta_j)} C e^{-B^T f(\theta_j)},$$

for $\{\theta_j, w_j\}_{j=1}^m$ are the Legendre points and weights, and $\omega_j = 2Lw_j \cdot \frac{\sin(\theta_j)}{(1-\cos(\theta_j))^2}$.

Back to Sylvester (Massei, Palitta, and Robol 2018)

We now have an integral on a finite domain $\Rightarrow$ **Gauss-Legendre quadrature**

$$X \approx \sum_{j=1}^{m} \omega_j \cdot e^{-Af(\theta_j)} C e^{-B^T f(\theta_j)},$$

for $\{\theta_j, w_j\}_{j=1}^m$ are the Legendre points and weights, and $\omega_j = 2Lw_j \cdot \frac{\sin(\theta_j)}{(1-\cos(\theta_j))^2}$.

❓ The **dominant cost** is now computing $e^{-Af(\theta_j)}$ and $e^{-B^T f(\theta_j)}$, how do we do it?

Back to Sylvester (Massei, Palitta, and Robol 2018)

We now have an integral on a finite domain $\Rightarrow$ **Gauss-Legendre quadrature**

$$X \approx \sum_{j=1}^{m} \omega_j \cdot e^{-Af(\theta_j)} C e^{-B^T f(\theta_j)},$$

for $\{\theta_j, w_j\}_{j=1}^m$ are the Legendre points and weights, and $\omega_j = 2Lw_j \cdot \frac{\sin(\theta_j)}{(1-\cos(\theta_j))^2}$.

❓ The **dominant cost** is now computing $e^{-Af(\theta_j)}$ and $e^{-B^T f(\theta_j)}$, how do we do it?

</> A **good idea** could be using *rational approximation* to $\exp(t)$

Back to Sylvester (Massei, Palitta, and Robol 2018)

We now have an integral on a finite domain $\Rightarrow$ **Gauss-Legendre quadrature**

$$X \approx \sum_{j=1}^{m} \omega_j \cdot e^{-Af(\theta_j)} C e^{-B^T f(\theta_j)},$$

for $\{\theta_j, w_j\}_{j=1}^m$ are the Legendre points and weights, and $\omega_j = 2Lw_j \cdot \frac{\sin(\theta_j)}{(1-\cos(\theta_j))^2}$.

❓ The **dominant cost** is now computing $e^{-Af(\theta_j)}$ and $e^{-B^T f(\theta_j)}$, how do we do it?

</> A **good idea** could be using *rational approximation* to $\exp(t)$

(d, d)-Padé with *scaling and squaring* $e^A = (e^{2^{-k}A})^{2^k}$ and $k = \lceil \log_2 \|A\|_2 \rceil$.

Back to Sylvester (Massei, Palitta, and Robol 2018)

We now have an integral on a finite domain $\Rightarrow$ **Gauss-Legendre quadrature**

$$X \approx \sum_{j=1}^{m} \omega_j \cdot e^{-Af(\theta_j)} C e^{-B^T f(\theta_j)},$$

for $\{\theta_j, w_j\}_{j=1}^m$ are the Legendre points and weights, and $\omega_j = 2Lw_j \cdot \frac{\sin(\theta_j)}{(1-\cos(\theta_j))^2}$.

❓ The **dominant cost** is now computing $e^{-Af(\theta_j)}$ and $e^{-B^T f(\theta_j)}$, how do we do it?

</> A **good idea** could be using *rational approximation* to $\exp(t)$

- (d, d)-Padé with *scaling and squaring* $e^A = (e^{2^{-k}A})^{2^k}$ and $k = \lceil \log_2 \|A\|_2 \rceil$.
- Rational Chebyshev function (Popolizio and Simoncini 2008):

$$e^x \approx \frac{r_1}{x - s_1} + \ldots + \frac{r_d}{x - s_d}.$$

requiring d inversions and additions that is uniformly accurate for every positive value of t, and thus is better in the case in which $\|A\|_2$ is large.

Back to Sylvester (Massei, Palitta, and Robol 2018)

Input: lyap_integral

A, B, C, m;

/* Solves $AX + XB^T = C$ with m integration points */

$L \leftarrow 100$; /* Should be tuned for accuracy! */

$[w, \theta] \leftarrow \mathrm{GaussLegendrePts}\, m$;

/* Integration points and weights on $[0, \pi]$ */

$X \leftarrow 0_{n \times n}$;

for $i = 1, \ldots, m$ **do**

$f \leftarrow L \cdot \cot(\frac{\theta_i}{2})^2$;

$X \leftarrow X + w_i \frac{\sin(\theta_i)}{(1-\cos\theta_i)^2} \cdot \mathrm{expm}\,(-f \cdot A) \cdot C \cdot \mathrm{expm}\,(-f \cdot B^T)$;

end

$X \leftarrow 2L \cdot X$;

Back to Sylvester (Massei, Palitta, and Robol 2018)

Input: lyap_integral
A, B, C, m;
/* Solves $AX + XB^T = C$ with m integration points */
$L \leftarrow 100$; /* Should be tuned for accuracy! */
$[w, \theta] \leftarrow \mathrm{GaussLegendrePts}\, m$;
/* Integration points and weights on $[0, \pi]$ */
$X \leftarrow 0_{n \times n}$;
for $i = 1, \ldots, m$ **do**
  $f \leftarrow L \cdot \cot(\frac{\theta_i}{2})^2$;
  $X \leftarrow X + w_i \frac{\sin(\theta_i)}{(1-\cos\theta_i)^2} \cdot \mathrm{expm}\,(-f \cdot A) \cdot C \cdot \mathrm{expm}\,(-f \cdot B^T)$;
end
$X \leftarrow 2L \cdot X$;

Mixed structures

If the right-hand side C is low-rank, and the structure in the matrices A and B is HODLR, thus permitting to perform fast matrix vector multiplications and system solutions; then we can apply the *extended Krylov subspace method* we had already seen.

Back to Sylvester (Massei, Palitta, and Robol 2018)

Input: lyap_integral
A, B, C, m;
/* Solves $AX + XB^T = C$ with m integration points */
$L \leftarrow 100$; /* Should be tuned for accuracy! */
$[w, \theta] \leftarrow \mathrm{GaussLegendrePts}\, m$;
/* Integration points and weights on $[0, \pi]$ */
$X \leftarrow 0_{n \times n}$;
for $i = 1, \ldots, m$ **do**
  $f \leftarrow L \cdot \cot(\frac{\theta_i}{2})^2$;
  $X \leftarrow X + w_i \frac{\sin(\theta_i)}{(1-\cos\theta_i)^2} \cdot \mathrm{expm}(-f \cdot A) \cdot C \cdot \mathrm{expm}(-f \cdot B^T)$;
end
$X \leftarrow 2L \cdot X$;

Mixed structures

If the right-hand side C is low-rank, and the structure in the matrices A and B is HODLR, thus permitting to perform fast matrix vector multiplications and system solutions; then we can apply the *extended Krylov subspace method* we had already seen.

Build

$$\mathbb{EK}_s(A, U) = \mathrm{span}\{U, A^{-1}U, AU, \ldots\}$$
$$\mathbb{EK}_s(B^T, V) = \mathrm{span}\{V, B^{-T}V, B^TV, \ldots\},$$

Back to Sylvester (Massei, Palitta, and Robol 2018)

Input: lyap_integral
A, B, C, m;
/* Solves $AX + XB^T = C$ with m integration points */
$L \leftarrow 100$; /* Should be tuned for accuracy! */
$[w, \theta] \leftarrow \mathrm{GaussLegendrePts}\, m$;
/* Integration points and weights on $[0, \pi]$ */
$X \leftarrow 0_{n \times n}$;
for $i = 1, \ldots, m$ **do**
  $f \leftarrow L \cdot \cot(\frac{\theta_i}{2})^2$;
  $X \leftarrow X + w_i \frac{\sin(\theta_i)}{(1-\cos\theta_i)^2} \cdot \mathrm{expm}(-f \cdot A) \cdot C \cdot \mathrm{expm}(-f \cdot B^T)$;
end
$X \leftarrow 2L \cdot X$;

Mixed structures

If the right-hand side C is low-rank, and the structure in the matrices A and B is HODLR, thus permitting to perform fast matrix vector multiplications and system solutions; then we can apply the *extended Krylov subspace method* we had already seen.

Build $\mathbb{EK}_s(A, U)$, $\mathbb{EK}_s(B^T, V)$, project on $\tilde{A}_s = U_s^* A U_s$, $\tilde{B}_s = V_s^* B V_s$, $\tilde{U} = U_s^* U$, and $\tilde{V} = V_s^* V$.

Back to Sylvester (Massei, Palitta, and Robol 2018)

Input: lyap_integral
A, B, C, m;
/* Solves $AX + XB^T = C$ with m integration points */
$L \leftarrow 100$; /* Should be tuned for accuracy! */
$[w, \theta] \leftarrow \mathrm{GaussLegendrePts}\, m$;
/* Integration points and weights on $[0, \pi]$ */
$X \leftarrow 0_{n \times n}$;
for $i = 1, \ldots, m$ **do**
  $f \leftarrow L \cdot \cot(\frac{\theta_i}{2})^2$;
  $X \leftarrow X + w_i \frac{\sin(\theta_i)}{(1-\cos\theta_i)^2} \cdot \mathrm{expm}(-f \cdot A) \cdot C \cdot \mathrm{expm}(-f \cdot B^T)$;
end
$X \leftarrow 2L \cdot X$;

Mixed structures

If the right-hand side C is low-rank, and the structure in the matrices A and B is HODLR, thus permitting to perform fast matrix vector multiplications and system solutions; then we can apply the *extended Krylov subspace method* we had already seen.

Build $\mathbb{EK}_s(A, U)$, $\mathbb{EK}_s(B^T, V)$, project on $\tilde{A}_s = U_s^* A U_s$, $\tilde{B}_s = V_s^* B V_s$, $\tilde{U} = U_s^* U$, and $\tilde{V} = V_s^* V$. Solve $\tilde{A}_s X_s + X_s \tilde{B}_s = \tilde{U}\tilde{V}^T$ with **dense arithmetic**.

Back to Sylvester (Massei, Palitta, and Robol 2018)

Input: lyap_integral
A, B, C, m;
/* Solves $AX + XB^T = C$ with m integration points */
$L \leftarrow 100$; /* Should be tuned for accuracy! */
$[w, \theta] \leftarrow \mathrm{GaussLegendrePts}\, m$;
/* Integration points and weights on $[0, \pi]$ */
$X \leftarrow 0_{n \times n}$;
for $i = 1, \ldots, m$ **do**
  $f \leftarrow L \cdot \cot(\frac{\theta_i}{2})^2$;
  $X \leftarrow X + w_i \frac{\sin(\theta_i)}{(1-\cos\theta_i)^2} \cdot \mathrm{expm}(-f \cdot A) \cdot C \cdot \mathrm{expm}(-f \cdot B^T)$;
end
$X \leftarrow 2L \cdot X$;

Mixed structures

If the right-hand side C is low-rank, and the structure in the matrices A and B is HODLR, thus permitting to perform fast matrix vector multiplications and system solutions; then we can apply the *extended Krylov subspace method* we had already seen.

Build $\mathbb{EK}_s(A, U)$, $\mathbb{EK}_s(B^T, V)$, project on $\tilde{A}_s = U_s^* A U_s$, $\tilde{B}_s = V_s^* B V_s$, $\tilde{U} = U_s^* U$, and $\tilde{V} = V_s^* V$. Solve $\tilde{A}_s X_s + X_s \tilde{B}_s = \tilde{U}\tilde{V}^T$ with **dense arithmetic**. An approximation is $U_s X_s V_s^*$.

Back to Sylvester (Massei, Palitta, and Robol 2018)

Input: lyap_integral
A, B, C, m;
/* Solves $AX + XB^T = C$ with m integration points */
$L \leftarrow 100$; /* Should be tuned for accuracy! */
$[w, \theta] \leftarrow \mathrm{GaussLegendrePts}\, m$;
/* Integration points and weights on $[0, \pi]$ */
$X \leftarrow 0_{n \times n}$;
for $i = 1, \ldots, m$ **do**
  $f \leftarrow L \cdot \cot(\frac{\theta_i}{2})^2$;
  $X \leftarrow X + w_i \frac{\sin(\theta_i)}{(1-\cos\theta_i)^2} \cdot \mathrm{expm}\,(-f \cdot A) \cdot C \cdot \mathrm{expm}\,(-f \cdot B^T)$;
end
$X \leftarrow 2L \cdot X$;

Mixed structures

If the right-hand side C is low-rank, and the structure in the matrices A and B is HODLR, thus permitting to perform fast matrix vector multiplications and system solutions; then we can apply the *extended Krylov subspace method* we had already seen.

Build $\mathbb{EK}_s(A, U)$, $\mathbb{EK}_s(B^T, V)$, project on $\tilde{A}_s = U_s^* A U_s$, $\tilde{B}_s = V_s^* B V_s$, $\tilde{U} = U_s^* U$, and $\tilde{V} = V_s^* V$. Solve $\tilde{A}_s X_s + X_s \tilde{B}_s = \tilde{U}\tilde{V}^T$ with **dense arithmetic**. An approximation is $U_s X_s V_s^*$. Another viable approach in the literature is (Kressner, Massei, and Robol 2019).

A numerical test (Massei, Mazza, and Robol 2019)

We use the usual square $[0,1]^2$, and the source f

$$f(x,y,t) = 100 \cdot (\sin(10\pi x)\cos(\pi y) + \sin(10t)\sin(\pi x) \cdot y(1-y)) \,.$$

for both **constant coefficient** $d^+ = d^- = 1$, and variable coefficients

$$d_1^+(x) = \Gamma(1.2)(1+x)^{\alpha_1}, \qquad d_1^-(x) = \Gamma(1.2)(2-x)^{\alpha_1},$$
$$d_2^+(y) = \Gamma(1.2)(1+y)^{\alpha_2}, \qquad d_2^-(y) = \Gamma(1.2)(2-y)^{\alpha_2}.$$

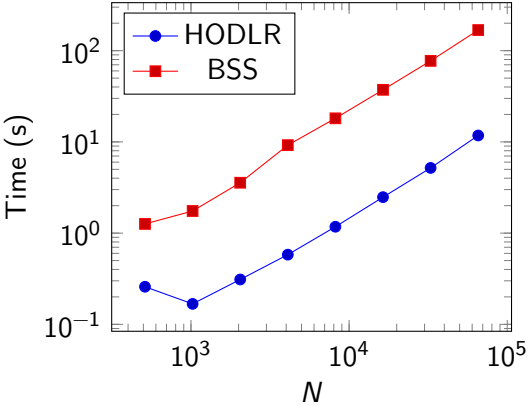
The fractional orders are $\alpha_1 = 1.3, \alpha_2 = 1.7$, and $\alpha_1 = 1.7, \alpha_2 = 1.9$. Methods are

- Sylvester by Extended-Krylov with stopping $\epsilon = 10^{-6}$ (HODLR),
  - HODLR arithmetic is set to work with a truncation of 10^{-8}.
- Sylvester by Extended-Krylov with stopping $\epsilon = 10^{-6}$ (Breiten, Simoncini, and Stoll 2016),
  - Inner solve with: GMRES with tolerance 10^{-7} and *structured preconditioners*,

A numerical test (Massei, Mazza, and Robol 2019)

Constant coefficient with $\alpha_1 = 1.3$ and $\alpha_2 = 1.7$.

N	t_{HODLR}	t_{BSS}	rank_ϵ	qsrank_ϵ
512	0.26	1.26	14	11
1,024	0.17	1.75	15	11
2,048	0.31	3.57	15	12
4,096	0.58	9.21	16	12
8,192	1.17	18.14	16	13
16,384	2.48	37.24	16	13
32,768	5.18	77.28	16	14
65,536	11.76	168.29	15	14

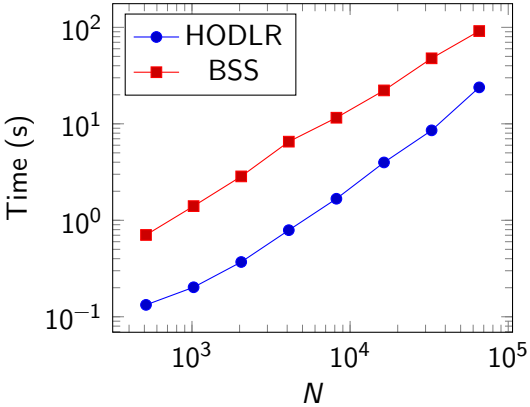


`FD_Example.m` from github.com/numpi/fme

A numerical test (Massei, Mazza, and Robol 2019)

Constant coefficient with $\alpha_1 = 1.7$ and $\alpha_2 = 1.9$.

N	t_{HODLR}	t_{BSS}	rank_ϵ	qsrank_ϵ
512	0.13	0.7	17	10
1,024	0.2	1.4	18	10
2,048	0.37	2.85	19	11
4,096	0.79	6.53	20	11
8,192	1.67	11.57	20	11
16,384	3.98	22.2	21	11
32,768	8.56	47.75	22	11
65,536	23.86	91.53	23	11

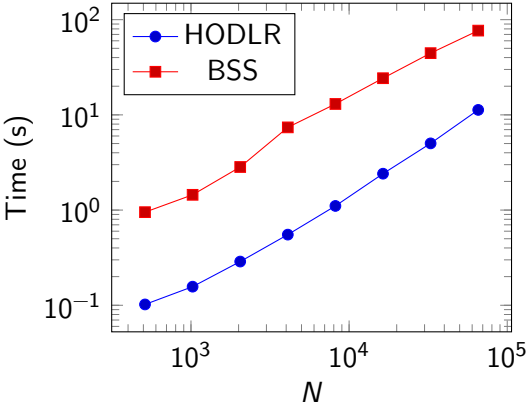


`FD_Example.m` from github.com/numpi/fme

A numerical test (Massei, Mazza, and Robol 2019)

Non-constant coefficient case with $\alpha_1 = 1.3$ and $\alpha_2 = 1.7$.

N	t_{HODLR}	t_{BSS}	rank_ϵ	qsrank_ϵ
512	0.1	0.95	14	10
1,024	0.16	1.45	14	11
2,048	0.29	2.83	15	12
4,096	0.55	7.39	16	12
8,192	1.11	13.02	16	13
16,384	2.41	24.27	16	13
32,768	5.02	44.5	16	14
65,536	11.28	76.78	16	14

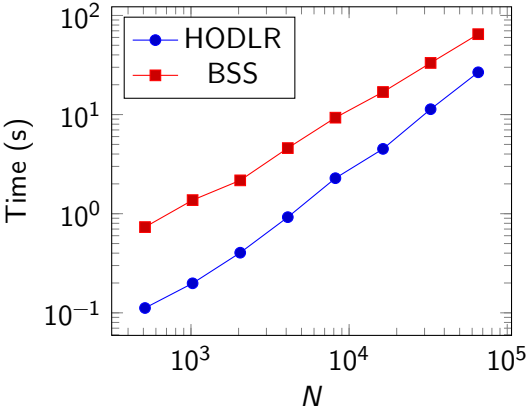


`FD_Example_vc.m` from github.com/numpi/fme

A numerical test (Massei, Mazza, and Robol 2019)

Non-constant coefficient case with $\alpha_1 = 1.7$ and $\alpha_2 = 1.9$.

N	t_{HODLR}	t_{BSS}	rank_ϵ	qsrank_ϵ
512	0.11	0.73	18	10
1,024	0.2	1.37	19	10
2,048	0.4	2.17	20	11
4,096	0.92	4.59	21	11
8,192	2.28	9.31	22	11
16,384	4.51	16.89	22	11
32,768	11.33	33.19	23	12
65,536	26.71	64.73	24	12



`FD_Example_vc.m` from github.com/numpi/fme

The tale of the matrix equation: the moral of the story.

- There is an advantage with respect to using Toeplitz-based BLAS like operations,

The tale of the matrix equation: the moral of the story.

- There is an advantage with respect to using Toeplitz-based BLAS like operations,
- In (Massei, Mazza, and Robol 2019) they are solving the case

$$\left(\frac{1}{2}I_{N_x} - \Delta t\tilde{G}_{N_x}\right)\tilde{W}^{(m+1)} + \tilde{W}^{(m+1)}\left(\frac{1}{2}I_{N_y} - \Delta t\tilde{G}_{N_y}\right)^T = \tilde{W}^{(m)} + \Delta t F^{(m+1)}, \quad m = 0, \ldots, M-1.$$

here the spectrum is *fictitiously independent from the discretization*, i.e., all matrix-equation solvers perform a number of iteration independent from the system size: the cost is reduced to the extended Krylov subspace cost! **But** we still have time-stepping to do.

The tale of the matrix equation: the moral of the story.

- There is an advantage with respect to using Toeplitz-based BLAS like operations,
- In (Massei, Mazza, and Robol 2019) they are solving the case

$$\left(\frac{1}{2}I_{N_x} - \Delta t\tilde{G}_{N_x}\right)\tilde{W}^{(m+1)} + \tilde{W}^{(m+1)}\left(\frac{1}{2}I_{N_y} - \Delta t\tilde{G}_{N_y}\right)^T = \tilde{W}^{(m)} + \Delta t F^{(m+1)}, \quad m = 0, \ldots, M-1.$$

  here the spectrum is *fictitiously independent from the discretization*, i.e., all matrix-equation solvers perform a number of iteration independent from the system size: the cost is reduced to the extended Krylov subspace cost! **But** we still have time-stepping to do.

- The case in which the matrix equation solver has a number of iterations dependent on the problem size is not yet resolved:

  ☻ Low-rank *but* 😠 **no preconditioner** – VS – 😠 **Full memory** *but* ☻ preconditioners

The tale of the matrix equation: the moral of the story.

- There is an advantage with respect to using Toeplitz-based BLAS like operations,
- In (Massei, Mazza, and Robol 2019) they are solving the case

$$\left(\frac{1}{2}I_{N_x} - \Delta t \tilde{G}_{N_x}\right) \tilde{W}^{(m+1)} + \tilde{W}^{(m+1)} \left(\frac{1}{2}I_{N_y} - \Delta t \tilde{G}_{N_y}\right)^T = \tilde{W}^{(m)} + \Delta t F^{(m+1)}, \quad m = 0, \dots, M-1.$$

  here the spectrum is *fictitiously independent from the discretization*, i.e., all matrix-equation solvers perform a number of iteration independent from the system size: the cost is reduced to the extended Krylov subspace cost! **But** we still have time-stepping to do.
- The case in which the matrix equation solver has a number of iterations dependent on the problem size is not yet resolved:

  ☻ Low-rank *but* 😠 **no preconditioner** – VS – 😠 **Full memory** *but* ☻ preconditioners
- Still looking for a way to solve **everything** *all-at-once* compactly.

Conclusion and summary

- We have seen how to work with matrices in HODLR format,
- We have discussed a couple of strategy to solve Sylvester equations with HODLR coefficients,
- We have applied all the machinery to solve a time step of a 2D equation FDE.

Next up

- Back to *all-at-once* solution with respect to both space and time,
- Linear multistep formulas in boundary value form,
- Structured preconditioner for LMFs,
- Tensor-Train reformulation of the problem.

Bibliography I

- Beckermann, B. and A. Townsend (2019). "Bounds on the singular values of matrices with displacement structure". In: *SIAM Rev.* 61.2. Revised reprint of "On the singular values of matrices with displacement structure" [MR3717820], pp. 319–344. ISSN: 0036-1445. DOI: 10.1137/19M1244433. URL: https://doi.org/10.1137/19M1244433.
- Breiten, T., V. Simoncini, and M. Stoll (2016). "Low-rank solvers for fractional differential equations". In: *Electron. Trans. Numer. Anal.* 45, pp. 107–132.
- Fiedler, M. (2010). "Notes on Hilbert and Cauchy matrices". In: *Linear Algebra Appl.* 432.1, pp. 351–356. ISSN: 0024-3795. DOI: 10.1016/j.laa.2009.08.014. URL: https://doi.org/10.1016/j.laa.2009.08.014.
- Hackbusch, W. (2015). *Hierarchical matrices: algorithms and analysis*. Vol. 49. Springer Series in Computational Mathematics. Springer, Heidelberg, pp. xxv+511. ISBN: 978-3-662-47323-8; 978-3-662-47324-5. DOI: 10.1007/978-3-662-47324-5. URL: https://doi.org/10.1007/978-3-662-47324-5.

Bibliography II

- Kressner, D., S. Massei, and L. Robol (2019). "Low-rank updates and a divide-and-conquer method for linear matrix equations". In: *SIAM J. Sci. Comput.* 41.2, A848–A876. ISSN: 1064-8275. DOI: 10.1137/17M1161038. URL: https://doi.org/10.1137/17M1161038.
- Massei, S., M. Mazza, and L. Robol (2019). "Fast solvers for two-dimensional fractional diffusion equations using rank structured matrices". In: *SIAM J. Sci. Comput.* 41.4, A2627–A2656. ISSN: 1064-8275. DOI: 10.1137/18M1180803. URL: https://doi.org/10.1137/18M1180803.
- Massei, S., D. Palitta, and L. Robol (2018). "Solving rank-structured Sylvester and Lyapunov equations". In: *SIAM J. Matrix Anal. Appl.* 39.4, pp. 1564–1590. ISSN: 0895-4798. DOI: 10.1137/17M1157155. URL: https://doi.org/10.1137/17M1157155.
- Massei, S., L. Robol, and D. Kressner (2020). "hm-toolbox: MATLAB software for HODLR and HSS matrices". In: *SIAM J. Sci. Comput.* 42.2, pp. C43–C68. ISSN: 1064-8275. DOI: 10.1137/19M1288048. URL: https://doi.org/10.1137/19M1288048.

Bibliography III

Popolizio, M. and V. Simoncini (2008). "Acceleration techniques for approximating the matrix exponential operator". In: *SIAM J. Matrix Anal. Appl.* 30.2, pp. 657–683. ISSN: 0895-4798. DOI: 10.1137/060672856. URL: https://doi.org/10.1137/060672856.

An introduction to fractional calculus

Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

October, 2022

All-at-once

We have seen that for a problem of the form

$$\begin{cases} u_t = \mathcal{L}(u), & u : \Omega \times [0, T] \to \mathbb{R}^d, \ \Omega \subseteq \mathbb{R}^d \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}), & \\ \mathcal{B}(u) = 0, & \mathbf{x} \in \partial\Omega. \end{cases}$$

with

- $\mathcal{L}(\cdot)$ a *linear* and *autonomous* differential operator (possibly involving fractional derivatives),
- or changing u_t with ${}^{CA}D^{\alpha}_{[0,t]}u$,

we can **rewrite it** as a single linear system/matrix equation.

All-at-once

We have seen that for a problem of the form

$$\begin{cases} \mathbf{u}_t = \mathcal{L}_h(\mathbf{u}), \quad \mathbf{u} : \mathbb{R}^n \times [0, T] \to \mathbb{R}^n \\ \mathbf{u}(0) = \mathbf{u}_0, \\ \mathcal{B}_h(\mathbf{u}) = 0. \end{cases}$$

with

- $\mathcal{L}(\cdot)$ a *linear* and *autonomous* differential operator (possibly involving fractional derivatives),
- or changing u_t with ${}^{CA}D^{\alpha}_{[0,t]}u$,

we can **rewrite it** as a single linear system/matrix equation.

To *abstract the procedure* let's think about working the **M**ethod **O**f **L**ine way!

All-at-once: system of autonomous ODE

Following the MOL trail, we now have to solve a system of autonomous ODEs:

$$M\mathbf{u}_t(t) = L\mathbf{u}(t), \qquad M, L \in \mathbb{R}^{n\times n},$$

that could be a **d**ifferential-**a**lgebraic system of **e**quations (DAE) if $\det(M) = 0$.

All-at-once: system of autonomous ODE

Following the MOL trail, we now have to solve a system of autonomous ODEs:

$$M\mathbf{u}_t(t) = L\mathbf{u}(t), \qquad M, L \in \mathbb{R}^{n\times n},$$

- that could be a **d**ifferential-**a**lgebraic system of **e**quations (DAE) if $\det(M) = 0$.
- To formulate the *all-at-once* procedure, one has to select a method to *march in time* the solution:

All-at-once: system of autonomous ODE

Following the MOL trail, we now have to solve a system of autonomous ODEs:

$$M\mathbf{u}_t(t) = L\mathbf{u}(t), \qquad M, L \in \mathbb{R}^{n\times n},$$

- that could be a **d**ifferential-**a**lgebraic system of **e**quations (DAE) if $\det(M) = 0$.
- To formulate the *all-at-once* procedure, one has to select a method to *march in time* the solution:
  - Linear multistep methods,
  - Runge-Kutta methods,
  - General linear methods (a mix of the two above strategies).

Linear Multistep Methods

Given a general ODE of the form

$$u'(t) = f(t, u(t)), \quad u(t_0) = u_0,$$

a k-step LMM is a recursion of the form with step-size $h = t_{n+k} - t_{n+k-1} > 0$

$$\sum_{j=0}^{k} \alpha_j u_{n+j} = \sum_{j=0}^{k} h\beta_j f_{n+j}, \qquad f_m \triangleq f(t_m, y_m),$$

with coefficients $\alpha_j \in \mathbb{R}$ and $\beta_j \in \mathbb{R}$ $(j = 0, \ldots, k)$, and we are **interested only** in **implicit methods**, i.e., $\beta_k \neq 0$.

Linear Multistep Methods

Given a general ODE of the form

$$u'(t) = f(t, u(t)), \quad u(t_0) = u_0,$$

a k-step LMM is a recursion of the form with step-size $h = t_{n+k} - t_{n+k-1} > 0$

$$\sum_{j=0}^{k} \alpha_j u_{n+j} = \sum_{j=0}^{k} h\beta_j f_{n+j}, \qquad f_m \triangleq f(t_m, y_m),$$

with coefficients $\alpha_j \in \mathbb{R}$ and $\beta_j \in \mathbb{R}$ $(j = 0, \ldots, k)$, and we are **interested only** in **implicit methods**, i.e., $\beta_k \neq 0$.
They can be analyzed by looking at the **polynomials**

$$\rho(\zeta) = \sum_{j=0}^{k} \alpha_j \zeta^j = (\zeta - 1) \sum_{j=0}^{k-1} \gamma_j \zeta^j = (\zeta - 1) \cdot \rho_R(\zeta), \qquad \sigma(\zeta) = \sum_{j=0}^{k} \beta_j \zeta^j.$$

Linear Multistep Methods

0-stable method

A method is 0-stable if all roots of $\rho(\zeta) = (\zeta - 1) \cdot \rho_R(\zeta) = 0$ lie inside or on the unit circle, with no multiple unimodular roots.

- *Zero stability* is necessary for convergence,
- It is a condition on the *extraneous operator* $\rho_R(\zeta)$, i.e., a condition on the k coefficients $\{\gamma_j\}_{j=0}^{k-1}$.

Linear Multistep Methods

0-stable method

A method is 0-stable if all roots of $\rho(\zeta) = (\zeta - 1) \cdot \rho_R(\zeta) = 0$ lie inside or on the unit circle, with no multiple unimodular roots.

- *Zero stability* is necessary for convergence,
- It is a condition on the *extraneous operator* $\rho_R(\zeta)$, i.e., a condition on the k coefficients $\{\gamma_j\}_{j=0}^{k-1}$.

A-stable method

The behavior of these methods can be analyzed by applying them on the test problem $y' = ky$ subject to the initial condition $y(0) = 1$ with $k \in \mathbb{C}$. The solution of this equation is $y(t) = e^{kt}$. If the numerical method exhibits the same behavior of the solution for a fixed step size, then the method is said to be A-stable.

Linear Multistep Methods

0-stable method

A method is 0-stable if all roots of $\rho(\zeta) = (\zeta - 1) \cdot \rho_R(\zeta) = 0$ lie inside or on the unit circle, with no multiple unimodular roots.

- *Zero stability* is necessary for convergence,
- It is a condition on the *extraneous operator* $\rho_R(\zeta)$, i.e., a condition on the k coefficients $\{\gamma_j\}_{j=0}^{k-1}$.

A-stable method

The behavior of these methods can be analyzed by applying them on the test problem $y' = ky$ subject to the initial condition $y(0) = 1$ with $k \in \mathbb{C}$. The solution of this equation is $y(t) = e^{kt}$. If the numerical method exhibits the same behavior of the solution for a fixed step size, then the method is said to be A-stable.

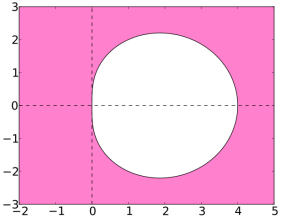
- Usually one ends up with limitations involving the admissible h.

Linear Multistep Methods: initial values

If we use a LMM with $k > 1$ we need more starting values than the one we have!

We are interested in **diffusion dominated problems**, thus **B**ackward-**D**ifferentiation **F**ormula**s** are a common choice.

	$\{\alpha_k\}_k,\ \beta_k = 1,\ \beta_j = 0,\ j \leq k$					
BDF2				$1/2$	-2	$3/2$
BDF3			$-1/3$	$3/2$	-3	$11/6$
BDF4		$1/4$	$-4/3$	3	-4	$25/12$
BDF5	$-1/5$	$5/4$	$-10/3$	5	-5	$137/60$
BDF6	$1/6$	$-6/5$	$15/4$	$-20/3$	$15/2$	-6	$147/60$



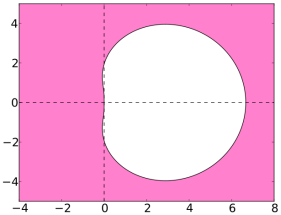
Methods with $k > 6$ are not zero-stable so they cannot be used.

Linear Multistep Methods: initial values

If we use a LMM with $k > 1$ we need more starting values than the one we have!

We are interested in **diffusion dominated problems**, thus **B**ackward-**D**ifferentiation **F**ormula**s** are a common choice.

	$\{\alpha_k\}_k,\ \beta_k = 1,\ \beta_j = 0,\ j \leq k$					
BDF2				$1/2$	-2	$3/2$
BDF3			$-1/3$	$3/2$	-3	$11/6$
BDF4		$1/4$	$-4/3$	3	-4	$25/12$
BDF5	$-1/5$	$5/4$	$-10/3$	5	-5	$137/60$
BDF6	$1/6$	$-6/5$	$15/4$	$-20/3$	$15/2$	-6	$147/60$



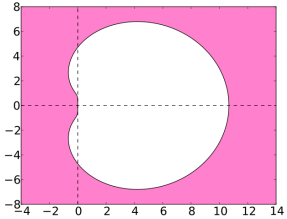
- Methods with $k > 6$ are not zero-stable so they cannot be used.
- If we want to use BDF6 we need 5 initial conditions, and have only one.
- We can use lower order BDFs to generate the step we need.

Linear Multistep Methods: initial values

If we use a LMM with $k > 1$ we need more starting values than the one we have!

We are interested in **diffusion dominated problems**, thus **B**ackward-**D**ifferentiation **F**ormula**s** are a common choice.

	$\{\alpha_k\}_k,\ \beta_k = 1,\ \beta_j = 0,\ j \leq k$					
BDF2				$1/2$	-2	$3/2$
BDF3			$-1/3$	$3/2$	-3	$11/6$
BDF4		$1/4$	$-4/3$	3	-4	$25/12$
BDF5	$-1/5$	$5/4$	$-10/3$	5	-5	$137/60$
BDF6	$1/6$	$-6/5$	$15/4$	$-20/3$	$15/2$	-6	$147/60$



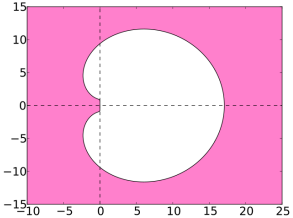
- Methods with $k > 6$ are not zero-stable so they cannot be used.
- If we want to use BDF6 we need 5 initial conditions, and have only one.
- We can use lower order BDFs to generate the step we need.

Linear Multistep Methods: initial values

If we use a LMM with $k > 1$ we need more starting values than the one we have!

We are interested in **diffusion dominated problems**, thus **B**ackward-**D**ifferentiation **F**ormula**s** are a common choice.

	$\{\alpha_k\}_k,\ \beta_k = 1,\ \beta_j = 0,\ j \leq k$					
BDF2				$1/2$	-2	$3/2$
BDF3			$-1/3$	$3/2$	-3	$11/6$
BDF4		$1/4$	$-4/3$	3	-4	$25/12$
BDF5	$-1/5$	$5/4$	$-10/3$	5	-5	$137/60$
BDF6	$1/6$ $\quad -6/5$	$15/4$	$-20/3$	$15/2$	-6	$147/60$



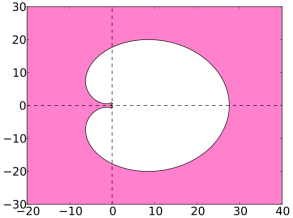
- Methods with $k > 6$ are not zero-stable so they cannot be used.
- If we want to use BDF6 we need 5 initial conditions, and have only one.
- We can use lower order BDFs to generate the step we need.

Linear Multistep Methods: initial values

If we use a LMM with $k > 1$ we need more starting values than the one we have!

We are interested in **diffusion dominated problems**, thus **B**ackward-**D**ifferentiation **F**ormula**s** are a common choice.

	$\{\alpha_k\}_k,\ \beta_k = 1,\ \beta_j = 0,\ j \leq k$					
BDF2				$1/2$	-2	$3/2$
BDF3			$-1/3$	$3/2$	-3	$11/6$
BDF4		$1/4$	$-4/3$	3	-4	$25/12$
BDF5	$-1/5$	$5/4$	$-10/3$	5	-5	$137/60$
BDF6	$1/6$	$-6/5$	$15/4$	$-20/3$	$15/2$	-6	$147/60$



- Methods with $k > 6$ are not zero-stable so they cannot be used.
- If we want to use BDF6 we need 5 initial conditions, and have only one.
- We can use lower order BDFs to generate the step we need.

Linear Multistep Methods

From what we have seen in the last lectures we can write down the problem as

$$(A_m \otimes M_n - hB_m \otimes L_n)\mathbf{u} = \mathbf{f},$$

Linear Multistep Methods

From what we have seen in the last lectures we can write down the problem as

$$(A_m \otimes M_n - hB_m \otimes L_n)\mathbf{u} = \mathbf{f},$$

$$A_m = \begin{bmatrix} 1 & & & & & & & \\ -2 & 3/2 & & & & & & \\ 3/2 & -3 & 11/6 & & & & & \\ -4/3 & 3 & -4 & 25/12 & & & & \\ 5/4 & -10/3 & 5 & -5 & 137/60 & & & \\ -6/5 & 15/4 & -20/3 & 15/2 & -6 & 147/60 & & \\ 1/6 & -6/5 & 15/4 & -20/3 & 15/2 & -6 & 147/60 & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}_{(m-1)\times(m-1)}$$

Linear Multistep Methods

From what we have seen in the last lectures we can write down the problem as

$$(A_m \otimes M_n - hB_m \otimes L_n)\mathbf{u} = \mathbf{f},$$

$$B_m = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}_{(m-1)\times(m-1)}$$

Linear Multistep Methods

From what we have seen in the last lectures we can write down the problem as

$$(A_m \otimes M_n - hB_m \otimes L_n)\mathbf{u} = \mathbf{f},$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{u}_0 + f(t_1) \\ -1/2\mathbf{u}_0 + f(t_2) \\ 1/3\mathbf{u}_0 + f(t_3) \\ -1/4\mathbf{u}_0 + f(t_4) \\ 1/5\mathbf{u}_0 + f(t_5) \\ -1/6\mathbf{u}_0 + f(t_6) \\ f(t_7) \\ \vdots \end{bmatrix}$$

A simple example

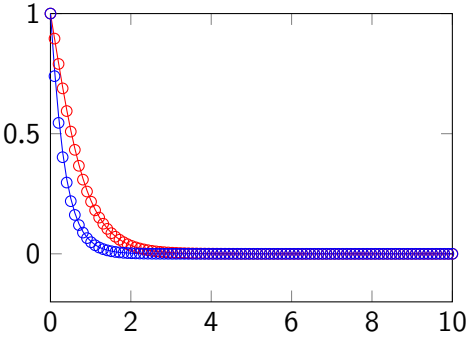
```
L = [-2, 1; 0, -3]; % Problem
y0 = [1;1];
n = length(L);
% Discretize
m = 100;
T = linspace(0,10,m); h = T(2)-T(1);
r = zeros(m-1,1); c = zeros(m-1,1);
r(1:7)=[147/60,-6,15/2,-20/3,15/4,-6/5,1/6];
c(1) = 147/60;
A = toeplitz(r,c);
A(1,1) = 1; % Fix BCs
A(2,1) = -2; A(2,2) = 3/2;
A(3,1) = 3/2; A(3,2) = -3; A(3,3) = 11/6;
A(4,1) = -4/3; A(4,2) = 3; A(4,3) = -4;
↪  A(4,4) = 25/12;
A(5,1) = 5/4; A(5,2) = -10/3; A(5,3) = 5;
```

```
A(5,4) = -5; A(5,5) = 137/60;
In = speye(n,n);
Im = speye(m-1,m-1);
%% Build rhs:
b = zeros((m-1)*n,1);
b(1:2) = y0;
b(3:4) = -1/2*y0;
b(5:6) = 1/3*y0;
b(7:8) = -1/4*y0;
b(9:10) = 1/5*y0;
b(11:12) = -1/6*y0;
% SOLVE (Linear system)
M = kron(A,In)-h*kron(Im,L);
x = M\b;
```

A simple example

We can compare the solution with `ode15s`, and visualize it

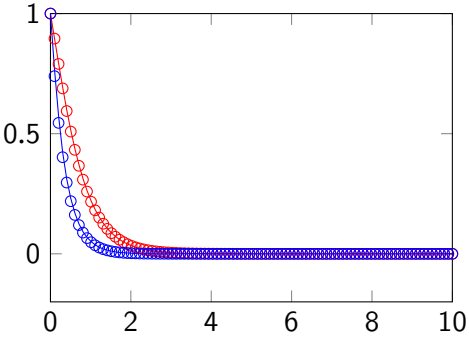
```
[tt,yy] = ode15s(@(t,y) L*y,T,y0);
X = reshape(x,n,m-1);
X = [y0,X];
% Plot
plot(T,X(1,:),'r-',T,X(2,:),'b-',...
T,yy(:,1),'ro',...
T,yy(:,2),'bo');
```



A simple example

We can compare the solution with `ode15s`, and visualize it

```
[tt,yy] = ode15s(@(t,y) L*y,T,y0);
X = reshape(x,n,m-1);
X = [y0,X];
% Plot
plot(T,X(1,:),'r-',T,X(2,:),'b-',...
T,yy(:,1),'ro',...
T,yy(:,2),'bo');
```

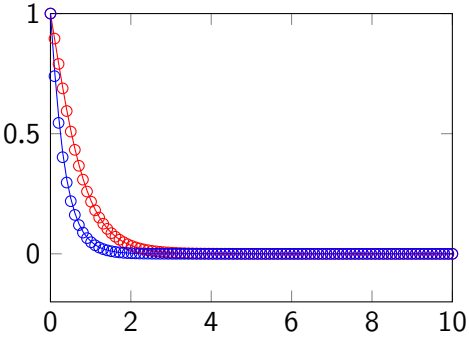


We could solve everything using a matrix-equation based solver,

A simple example

We can compare the solution with `ode15s`, and visualize it

```
[tt,yy] = ode15s(@(t,y) L*y,T,y0);
X = reshape(x,n,m-1);
X = [y0,X];
% Plot
plot(T,X(1,:),'r-',T,X(2,:),'b-',...
T,yy(:,1),'ro',...
T,yy(:,2),'bo');
```

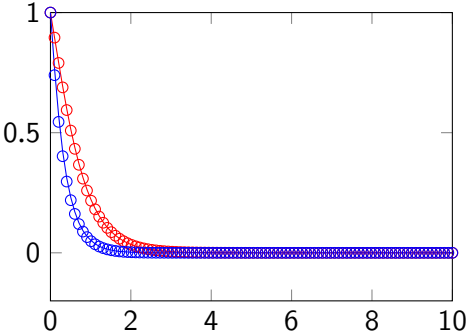


- We could solve everything using a matrix-equation based solver,
- but we are looking at a case in which $m = 2$ with a "non refinable" space operator.

A simple example

We can compare the solution with `ode15s`, and visualize it

```
[tt,yy] = ode15s(@(t,y) L*y,T,y0);
X = reshape(x,n,m-1);
X = [y0,X];
% Plot
plot(T,X(1,:),'r-',T,X(2,:),'b-',...
T,yy(:,1),'ro',...
T,yy(:,2),'bo');
```



- We could solve everything using a matrix-equation based solver,
- but we are looking at a case in which $m = 2$ with a "non refinable" space operator.

- What can we say about the A_m matrix?

Matrix properties

A_m is a banded Toeplitz matrix plus a rank correction.

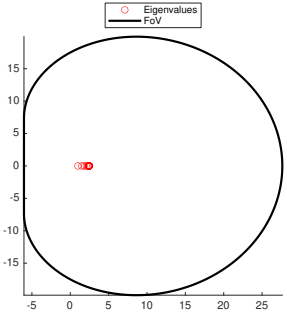
$$A_m = \begin{bmatrix} 1 & & & & & & & \\ -2 & 3/2 & & & & & & \\ 3/2 & -3 & 11/6 & & & & & \\ -4/3 & 3 & -4 & 25/12 & & & & \\ 5/4 & -10/3 & 5 & -5 & 137/60 & & & \\ -6/5 & 15/4 & -20/3 & 15/2 & -6 & 147/60 & & \\ 1/6 & -6/5 & 15/4 & -20/3 & 15/2 & -6 & 147/60 & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

We know the eigenvalues in closed form: it's lower triangular!

Matrix properties

A_m is a banded Toeplitz matrix plus a rank correction.

$$A_m = \begin{bmatrix} 1 & & & & & & & \\ -2 & 3/2 & & & & & & \\ 3/2 & -3 & 11/6 & & & & & \\ -4/3 & 3 & -4 & 25/12 & & & & \\ 5/4 & -10/3 & 5 & -5 & 137/60 & & & \\ -6/5 & 15/4 & -20/3 & 15/2 & -6 & 147/60 & & \\ 1/6 & -6/5 & 15/4 & -20/3 & 15/2 & -6 & 147/60 & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

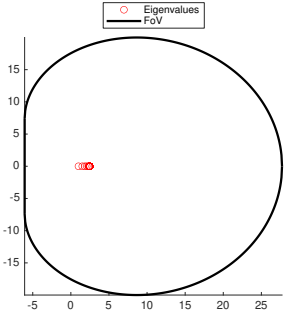


- We know the eigenvalues in closed form: it's lower triangular!
- The Field-Of-Values contains the origin... bad for bounds!

Matrix properties

A_m is a banded Toeplitz matrix plus a rank correction.

$$A_m = \begin{bmatrix} 1 & & & & & & & \\ -2 & 3/2 & & & & & & \\ 3/2 & -3 & 11/6 & & & & & \\ -4/3 & 3 & -4 & 25/12 & & & & \\ 5/4 & -10/3 & 5 & -5 & 137/60 & & & \\ -6/5 & 15/4 & -20/3 & 15/2 & -6 & 147/60 & & \\ 1/6 & -6/5 & 15/4 & -20/3 & 15/2 & -6 & 147/60 & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

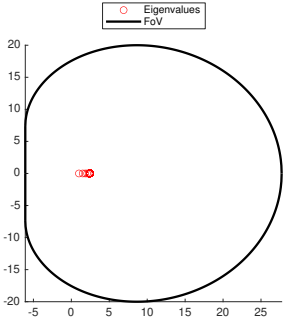


- We know the eigenvalues in closed form: it's lower triangular!
- The Field-Of-Values contains the origin... bad for bounds!

Matrix properties

A_m is a banded Toeplitz matrix plus a rank correction.

$$A_m = \begin{bmatrix} 1 & & & & & & & \\ -2 & 3/2 & & & & & & \\ 3/2 & -3 & 11/6 & & & & & \\ -4/3 & 3 & -4 & 25/12 & & & & \\ 5/4 & -10/3 & 5 & -5 & 137/60 & & & \\ -6/5 & 15/4 & -20/3 & 15/2 & -6 & 147/60 & & \\ 1/6 & -6/5 & 15/4 & -20/3 & 15/2 & -6 & 147/60 & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$



- We know the eigenvalues in closed form: it's lower triangular!
- The Field-Of-Values contains the origin... bad for bounds!
- Its clearly non diagonalizable, if we try and look at the condition number of the eigenvector matrix $\kappa_2(X_{100}) = 7.30 \times 10^{111}$.

Matrix properties

Indeed, already for the BDF1 (a.k.a. the implicit Euler method) we have

$$A_m = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}_{(m-1)\times(m-1)}$$

- It is a Jordan block, so *no diagonalization*,

Matrix properties

Indeed, already for the BDF1 (a.k.a. the implicit Euler method) we have

$$A_m = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}_{(m-1)\times(m-1)}$$

- It is a Jordan block, so *no diagonalization*,
- What do we expect for the matrix equation solver?

Matrix properties

Indeed, already for the BDF1 (a.k.a. the implicit Euler method) we have

$$A_m = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}_{(m-1)\times(m-1)}$$

BDF1, $\alpha = 1.5$

m	n	IT	Residual
64	128	13	1.007848e-10
128	256	16	6.145733e-10
256	512	21	7.639171e-10
512	1024	27	5.857467e-10
1024	2048	34	8.065585e-10
2048	4096	42	9.819085e-10

- It is a Jordan block, so *no diagonalization*,
- What do we expect for the matrix equation solver?

Matrix properties

Indeed, already for the BDF1 (a.k.a. the implicit Euler method) we have

$$A_m = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}_{(m-1)\times(m-1)}$$

- It is a Jordan block, so *no diagonalization*,
- What do we expect for the matrix equation solver?

BDF1, $\alpha = 1.5$

m	n	IT	Residual
64	128	13	1.007848e-10
128	256	16	6.145733e-10
256	512	21	7.639171e-10
512	1024	27	5.857467e-10
1024	2048	34	8.065585e-10
2048	4096	42	9.819085e-10

BDF6, $\alpha = 1.5$

m	n	IT	Residual
64	128	21	3.651570e-10
128	256	33	1.746513e-10
256	512	71	2.530720e-15
512	1024	128	1.975160e-22
1024	2048	251	4.157259e-10
2048	4096	495	6.310887e-10

Matrix properties

Indeed, already for the BDF1 (a.k.a. the implicit Euler method) we have

$$A_m = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}_{(m-1)\times(m-1)}$$

- It is a Jordan block, so *no diagonalization*,
- What do we expect for the matrix equation solver? Nothing good!

BDF1, $\alpha = 1.5$

m	n	IT	Residual
64	128	13	1.007848e-10
128	256	16	6.145733e-10
256	512	21	7.639171e-10
512	1024	27	5.857467e-10
1024	2048	34	8.065585e-10
2048	4096	42	9.819085e-10

BDF6, $\alpha = 1.5$

m	n	IT	Residual
64	128	21	3.651570e-10
128	256	33	1.746513e-10
256	512	71	2.530720e-15
512	1024	128	1.975160e-22
1024	2048	251	4.157259e-10
2048	4096	495	6.310887e-10

Linear Multistep Methods in Boundary Value Form

❓ Part of the problem, are those triangular matrices. Can we do something?

Linear Multistep Methods in Boundary Value Form

❓ Part of the problem, are those triangular matrices. Can we do something?

🔨 If we use *more than one step*, we still need **auxiliary formulas** to close the iteration.

Linear Multistep Methods in Boundary Value Form

- Part of the problem, are those triangular matrices. Can we do something?
- If we use *more than one step*, we still need **auxiliary formulas** to close the iteration.
- We could distribute the conditions differently, that is, not all on the initial data.

$$\sum_{j=-\nu}^{\mu-\nu} \alpha_{j+\nu}\mathbf{u}_{n+j} = h\sum_{j=-\nu}^{\mu-\nu} \beta_{j+\nu}\mathbf{f}_{n+j}, \quad n = \nu, \ldots, m-k+\nu.$$

- k steps,
- ν initial conditions, and
- $\mu - \nu$ final conditions,
- Described by $\rho(z) = z^\nu \sum_{j=-\nu}^{k-\nu} \alpha_{j+\nu} z^j$, and $\sigma(z) = z^\nu \sum_{j=-\nu}^{k-\nu} \beta_{j+\nu} z^j$.

Linear Multistep Methods in Boundary Value Form

- Part of the problem, are those triangular matrices. Can we do something?
- If we use *more than one step*, we still need **auxiliary formulas** to close the iteration.
- We could distribute the conditions differently, that is, not all on the initial data.

$$\sum_{j=-\nu}^{\mu-\nu} \alpha_{j+\nu}\mathbf{u}_{n+j} = h\sum_{j=-\nu}^{\mu-\nu} \beta_{j+\nu}\mathbf{f}_{n+j}, \quad n = \nu, \dots, m-k+\nu.$$

- k steps,
- ν initial conditions, and
- $\mu - \nu$ final conditions,
- Described by $\rho(z) = z^\nu \sum_{j=-\nu}^{k-\nu} \alpha_{j+\nu} z^j$, and $\sigma(z) = z^\nu \sum_{j=-\nu}^{k-\nu} \beta_{j+\nu} z^j$.
- How does this change matrices and stability?

Linear Multistep Methods in Boundary Value Form

If we collect the matrices for the inner steps of a *scalar* ODE, we get

$$A_m = \begin{bmatrix} \alpha_\nu & \cdots & \alpha_k & & & \\ \vdots & \ddots & & \ddots & & \\ \alpha_0 & & \ddots & & \ddots & \\ & \ddots & & \ddots & & \alpha_k \\ & & \ddots & & \ddots & \vdots \\ & & & \alpha_0 & \cdots & \alpha_\nu \end{bmatrix}_{(m-\nu)\times(m-\nu)}, B_m = \begin{bmatrix} \beta_\nu & \cdots & \beta_k & & & \\ \vdots & \ddots & & \ddots & & \\ \beta_0 & & \ddots & & \ddots & \\ & \ddots & & \ddots & & \beta_k \\ & & \ddots & & \ddots & \vdots \\ & & & \beta_0 & \cdots & \beta_\nu \end{bmatrix}_{(m-\nu)\times(m-\nu)}$$

and the vectors

$$\mathbf{u} = (u_\nu, \cdots, u_{m-1})^T, \quad \mathbf{f} = (f_\nu, \cdots, f_{m-1})^T.$$

Linear Multistep Methods in Boundary Value Form

If we collect the matrices for the inner steps of a *scalar* ODE, we get A_m, B_m, and the vectors

$$\mathbf{u} = (u_\nu, \cdots, u_{m-1})^T, \quad \mathbf{f} = (f_\nu, \cdots, f_{m-1})^T.$$

Finding the system

$$A_m\mathbf{u} - hB_m\mathbf{f} = -\begin{bmatrix} \sum_{j=0}^{\nu-1}(\alpha_j y_j - h\beta_j f_j) \\ \vdots \\ a_0 y_{\nu-1} - h\beta_0 f_{\nu-1} \\ 0 \\ \vdots \\ 0 \\ \alpha_k y_m - h\beta_k f_m \\ \vdots \\ \sum_{j=1}^{\mu}(\alpha_{\nu+j} y_{m-1+j} - h\beta_{\nu_1+j} f_{m-1+j}). \end{bmatrix}$$

- A_m and B_m are Toeplitz matrices with *lower bandwidth* ν and *upper bandwidth* μ.
- We still need **auxiliary formulas** to fix the $\nu + \mu - 1$ starting/ending values.

Convergence and stability

Before concluding the construction, let's focus on *convergence* and *stability*.

Convergence and stability

Before concluding the construction, let's focus on *convergence* and *stability*.

❓ Did we gain anything by moving on to a more difficult problem?

Convergence and stability

Before concluding the construction, let's focus on *convergence* and *stability*.

❓ Did we gain anything by moving on to a more difficult problem?

$S_{\nu,\mu}$-polynomial (Brugnano and Trigiante 1998, Definition 4.4.2)

A polynomial $p(z)$ of degree $k = \nu + \mu$ is an $S_{\nu,\mu}$-polynomial if its roots are such that

$$|z_1| \leq |z_2| \leq \cdots \leq |z_\nu| < 1 < |z_{\nu+1}| \leq \cdots \leq |z_\nu|.$$

Convergence and stability

Before concluding the construction, let's focus on *convergence* and *stability*.

❓ Did we gain anything by moving on to a more difficult problem?

$S_{\nu,\mu}$-polynomial (Brugnano and Trigiante 1998, Definition 4.4.2)

A polynomial $p(z)$ of degree $k = \nu + \mu$ is an $S_{\nu,\mu}$-polynomial if its roots are such that

$$|z_1| \leq |z_2| \leq \cdots \leq |z_\nu| < 1 < |z_{\nu+1}| \leq \cdots \leq |z_\nu|.$$

$N_{\nu,\mu}$-polynomial (Brugnano and Trigiante 1998, Definition 4.4.3)

A polynomial $p(z)$ of degree $k = \nu + \mu$ is an $S_{\nu,\mu}$-polynomial if its roots are such that

$$|z_1| \leq |z_2| \leq \cdots \leq |z_\nu| \leq 1 < |z_{\nu+1}| \leq \cdots \leq |z_\nu|.$$

being **simple** the roots of unit modulus.

Convergence and stability

Before concluding the construction, let's focus on *convergence* and *stability*.

❓ Did we gain anything by moving on to a more difficult problem?

$S_{\nu,\mu}$-polynomial (Brugnano and Trigiante 1998, Definition 4.4.2)

A polynomial $p(z)$ of degree $k = \nu + \mu$ is an $S_{\nu,\mu}$-polynomial if its roots are such that

$$|z_1| \leq |z_2| \leq \cdots \leq |z_\nu| < 1 < |z_{\nu+1}| \leq \cdots \leq |z_\nu|.$$

$N_{\nu,\mu}$-polynomial (Brugnano and Trigiante 1998, Definition 4.4.3)

A polynomial $p(z)$ of degree $k = \nu + \mu$ is an $S_{\nu,\mu}$-polynomial if its roots are such that

$$|z_1| \leq |z_2| \leq \cdots \leq |z_\nu| \leq 1 < |z_{\nu+1}| \leq \cdots \leq |z_\nu|.$$

being **simple** the roots of unit modulus.

👁 If $\nu = k$ ($\mu = 0$), these are the conditions for LMF 0-stability!

Convergence and stability

Let $a_{-\nu} a_{\mu} \neq 0$ and

$$T_n = \begin{bmatrix} a_0 & \cdots & a_{\mu} & & & \\ \vdots & \ddots & & \ddots & & \\ a_{-\nu} & & \ddots & & \ddots & \\ & \ddots & & \ddots & & a_{\mu} \\ & & \ddots & & \ddots & \vdots \\ & & & a_{-\nu} & \cdots & a_0 \end{bmatrix},$$

we consider the polynomial

$$p(z) = \sum_{i=-\nu}^{\mu} a_i z^{\nu+i}.$$

Convergence and stability

Let $a_{-\nu}a_{\mu} \neq 0$ and

$$T_n = \begin{bmatrix} a_0 & \cdots & a_{\mu} & & & \\ \vdots & \ddots & & \ddots & & \\ a_{-\nu} & & \ddots & & \ddots & \\ & \ddots & & \ddots & & a_{\mu} \\ & & \ddots & & \ddots & \vdots \\ & & & a_{-\nu} & \cdots & a_0 \end{bmatrix},$$

we consider the polynomial

$$p(z) = \sum_{i=-\nu}^{\mu} a_i z^{\nu+i}.$$

Lemma (Brugnano and Trigiante 1998, Lemma 4.4.4)

If the polynomial $p(z)$ associated with the matrix T_n is an $N_{\nu,\mu}$-polynomial, then T_n^{-1} has entries $t_{i,j}^{(-1)}$ such that

1. $|t_{i,j}^{(-1)}| \leq \gamma$ independent of N, for $i \geq j$,
2. $|t_{i,j}^{(-1)}| \leq \eta\xi^{j-i}$ for $i < j$, where $\eta > 0$ and $0 < \xi, < 1$ are independent of N.

Convergence and stability

Let $a_{-\nu}a_{\mu} \neq 0$ and

$$T_n = \begin{bmatrix} a_0 & \cdots & a_{\mu} & & & \\ \vdots & \ddots & & \ddots & & \\ a_{-\nu} & & \ddots & & \ddots & \\ & \ddots & & \ddots & & a_{\mu} \\ & & \ddots & & \ddots & \vdots \\ & & & a_{-\nu} & \cdots & a_0 \end{bmatrix},$$

we consider the polynomial

$$p(z) = \sum_{i=-\nu}^{\mu} a_i z^{\nu+i}.$$

Lemma (Brugnano and Trigiante 1998, Lemma 4.4.4)

If the polynomial $p(z)$ associated with the matrix T_n is an $N_{\nu,\mu}$-polynomial, then T_n^{-1} has entries $t_{i,j}^{(-1)}$ such that

1. $|t_{i,j}^{(-1)}| \leq \gamma$ independent of N, for $i \geq j$,
2. $|t_{i,j}^{(-1)}| \leq \eta\xi^{j-i}$ for $i < j$, where $\eta > 0$ and $0 < \xi < 1$ are independent of N.

$$\|T_n^{-1}\| \leq \gamma C_n + \eta\Delta_n,$$

with $C_n = \begin{bmatrix} 1 & & \\ \vdots & \ddots & \\ 1 & \cdots & 1 \end{bmatrix}$,

Convergence and stability

Let $a_{-\nu}a_{\mu} \neq 0$ and

$$T_n = \begin{bmatrix} a_0 & \cdots & a_{\mu} & & & \\ \vdots & \ddots & & \ddots & & \\ a_{-\nu} & & \ddots & & \ddots & \\ & \ddots & & \ddots & & a_{\mu} \\ & & \ddots & & \ddots & \vdots \\ & & & a_{-\nu} & \cdots & a_0 \end{bmatrix},$$

we consider the polynomial

$$p(z) = \sum_{i=-\nu}^{\mu} a_i z^{\nu+i}.$$

Lemma (Brugnano and Trigiante 1998, Lemma 4.4.4)

If the polynomial $p(z)$ associated with the matrix T_n is an $N_{\nu,\mu}$-polynomial, then T_n^{-1} has entries $t_{i,j}^{(-1)}$ such that

1. $|t_{i,j}^{(-1)}| \leq \gamma$ independent of N, for $i \geq j$,
2. $|t_{i,j}^{(-1)}| \leq \eta\xi^{j-i}$ for $i < j$, where $\eta > 0$ and $0 < \xi < 1$ are independent of N.

$$\|T_n^{-1}\| \leq \gamma C_n + \eta\Delta_n,$$

with Δ_n the upper triangular Toeplitz matrix with last column $(\xi^{n-1}, \ldots, \xi^2, \xi, 0)^T$.

Convergence and stability

Theorem (Brugnano and Trigiante 1998, Theorem 4.4.3)

Ignoring the effect of round-off errors, a BVM with (ν, μ)-boundary conditions is convergent if it is consistent and the polynomial $\rho(z)$ is an $N_{\nu,\mu}$-polynomial.

Convergence and stability

Theorem (Brugnano and Trigiante 1998, Theorem 4.4.3)

Ignoring the effect of round-off errors, a BVM with (ν, μ)-boundary conditions is convergent if it is consistent and the polynomial $\rho(z)$ is an $N_{\nu,\mu}$-polynomial.

To reproduce the "0-stable + consistent $\Rightarrow$ convergence" framework, we define:

$0_{\nu,\mu}$-stability (Brugnano and Trigiante 1998, Definition 4.5.1)

A BVM with (ν, μ)-boundary conditions is $0_{\nu,\mu}$-stable if the corresponding polynomial $\rho(z)$ is an $N_{\nu,\mu}$-polynomial.

Convergence and stability

Theorem (Brugnano and Trigiante 1998, Theorem 4.4.3)

Ignoring the effect of round-off errors, a BVM with (ν, μ)-boundary conditions is convergent if it is consistent and the polynomial $\rho(z)$ is an $N_{\nu,\mu}$-polynomial.

To reproduce the "0-stable + consistent $\Rightarrow$ convergence" framework, we define:

$0_{\nu,\mu}$-stability (Brugnano and Trigiante 1998, Definition 4.5.1)

A BVM with (ν, μ)-boundary conditions is $0_{\nu,\mu}$-stable if the corresponding polynomial $\rho(z)$ is an $N_{\nu,\mu}$-polynomial.

(ν, μ)-Absolute stability (Brugnano and Trigiante 1998, Definition 4.7.1)

A BVM with (ν, μ)-boundary conditions is ν, μ-Absolutely stable for a given complex number q it the polynomial $\pi(z, q) = \rho(z) - q\sigma(z)$, is an $S_{\nu,\mu}$-polynomial.

Convergence and stability

We have a degree of **arbitrariness** in deciding how and how many initial / final conditions to set. Clearly ν has to be at least one (we do have an initial condition of our IVP), then for the remaining we have to let (ν, μ)-Absolute stability guide us.

Convergence and stability

We have a degree of **arbitrariness** in deciding how and how many initial / final conditions to set. Clearly ν has to be at least one (we do have an initial condition of our IVP), then for the remaining we have to let (ν, μ)-Absolute stability guide us.

Correct use a consistent LMF is *correctly used* in $q \in \mathbb{C}^-$, where $\pi(z, q)$ is an $S_{\nu,\mu}$-polynomial, if ν conditions are imposed at the initial points, and μ conditions are posed at the end of the interval.

Convergence and stability

We have a degree of **arbitrariness** in deciding how and how many initial / final conditions to set. Clearly ν has to be at least one (we do have an initial condition of our IVP), then for the remaining we have to let (ν, μ)-Absolute stability guide us.

Correct use a consistent LMF is *correctly used* in $q \in \mathbb{C}^-$, where $\pi(z, q)$ is an $S_{\nu,\mu}$-polynomial, if ν conditions are imposed at the initial points, and μ conditions are posed at the end of the interval.

To have a livable life, one always consider family of methods for which the boundary of the (ν, μ)-Absolutely stability region is a *regular Jordan curve*. More specifically, having that

$$\mathcal{A}_{\nu,\mu} = \{q \in \mathbb{C} : \pi(z, q) \text{ is an } S_{\nu,\mu}\text{-polynomial}\},$$

has the origin on its boundary and is possibly equal to the whole $\mathbb{C}^-$.

A gallery of formulas

It is possible to reformulate many LMFs in this new format

A gallery of formulas

It is possible to reformulate many LMFs in this new format

- BDF $\Rightarrow$ Generalized-BDF (GBDF): $\sum_{i=0}^{k} \alpha_i u_{n+i} = hf_{n+j}$, $j \in \{0, 1, \ldots, k\}$

A gallery of formulas

It is possible to reformulate many LMFs in this new format

- BDF $\Rightarrow$ Generalized-BDF (GBDF): $\sum_{i=0}^{k} \alpha_i u_{n+i} = h f_{n+j}$, $j \in \{0, 1, \ldots, k\}$
  - A method of this form is $0_{\nu,k-\nu}$-stable and $A_{\nu,k-\nu}$-stable for

$$\nu = \begin{cases} \frac{k+2}{2}, & \text{for even } k, \\ \frac{k+1}{2}, & \text{for odd } k. \end{cases}$$

  $\Rightarrow$ with this choice we no longer have the constraint of having at most $k = 6$ steps of the standard BDF!

- Adams-Moulton Methods $\Rightarrow$ GAMM $u_{n+j} - u_{n+j-1} = h \sum_{i=0}^{k} \beta_i f_{n+i}$

A gallery of formulas

It is possible to reformulate many LMFs in this new format

- BDF $\Rightarrow$ Generalized-BDF (GBDF): $\sum_{i=0}^{k} \alpha_i u_{n+i} = h f_{n+j}$, $j \in \{0, 1, \dots, k\}$
  - A method of this form is $0_{\nu,k-\nu}$-stable and $A_{\nu,k-\nu}$-stable for

$$\nu = \begin{cases} \frac{k+2}{2}, & \text{for even } k, \\ \frac{k+1}{2}, & \text{for odd } k. \end{cases}$$

  $\Rightarrow$ with this choice we no longer have the constraint of having at most $k = 6$ steps of the standard BDF!

- Adams-Moulton Methods $\Rightarrow$ GAMM $u_{n+j} - u_{n+j-1} = h \sum_{i=0}^{k} \beta_i f_{n+i}$
  - A method of this form is $0_{\nu,k-\nu}$-stable and $A_{\nu,k-\nu}$-stable for

$$\nu = \begin{cases} \frac{k+1}{2}, & \text{for odd } k, \\ \frac{k}{2}, & \text{for even } k. \end{cases}$$

- See the book (Brugnano and Trigiante 1998) for other possible generalizations.

Additional formulas

We need **additional formulas** for the $k - 1 = \nu + \mu - 1$ boundary values.

Additional formulas

We need **additional formulas** for the $k-1=\nu+\mu-1$ boundary values.

$$A_m = \begin{bmatrix} 1 & \dots & 0 & & & & \\ \alpha_0^{(1)} & \dots & \alpha_k^{(1)}, & & & & \\ \vdots & & \vdots & & & & \\ \alpha_0^{(\nu-1)} & \dots & \alpha_k^{(\nu-1)} & & & & \\ \alpha_0 & \dots & \alpha_k & & & & \\ & \alpha_0 & \dots & \alpha_k & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \alpha_0 & \dots & \alpha_k & \\ & & & \alpha_0^{(m-k+\nu+1)} & \dots & \alpha_k^{(m-k+\nu+1)} & \\ & & & \vdots & & \vdots & \\ & & & \alpha_0^{(m)} & \dots & \alpha_k^{(m)} & \end{bmatrix},$$

Additional formulas

We need **additional formulas** for the $k-1=\nu+\mu-1$ boundary values.

$$B_m = \begin{bmatrix} 0 & \dots & 0 & & & \\ \beta_0^{(1)} & \dots & \beta_k^{(1)}, & & & \\ \vdots & & \vdots & & & \\ \beta_0^{(\nu-1)} & \dots & \beta_k^{(\nu-1)} & & & \\ \beta_0 & \dots & \beta_k & & & \\ & \beta_0 & \dots & \beta_k & & \\ & & \ddots & \ddots & \ddots & \\ & & & \beta_0 & \dots & \beta_k \\ & & & \beta_0^{(m-k+\nu+1)} & \dots & \beta_k^{(m-k+\nu+1)}, \\ & & & \vdots & & \vdots \\ & & & \beta_0^{(m)} & \dots & \beta_k^{(m)} \end{bmatrix}.$$

Additional formulas

- We need **additional formulas** for the $k-1=\nu+\mu-1$ boundary values.
  - If we know how to compute them, then we end up having to solve the **matrix equation**

$$M_n U A_m^T - h L_n U B_m^T = F,$$

or the **linear system**

$$(A_m \otimes M_n - h B_m \otimes L_n)\mathbf{u} = \mathbf{f}, \text{ where } \mathrm{vec}(U) = \mathbf{u}, \ \mathrm{vec}(F) = \mathbf{f}.$$

Additional formulas

- We need **additional formulas** for the $k-1=\nu+\mu-1$ boundary values.
  - If we know how to compute them, then we end up having to solve the **matrix equation**

$$M_n U A_m^T - h L_n U B_m^T = F,$$

or the **linear system**

$$(A_m \otimes M_n - h B_m \otimes L_n)\mathbf{u} = \mathbf{f}, \text{ where } \mathrm{vec}(U) = \mathbf{u}, \ \mathrm{vec}(F) = \mathbf{f}.$$

- Let us build everything for using GBDFs and our fractional-in-space problem.

Generalized BDF

First we need to compute $\rho(z)$ and $\sigma(z)$

```
function [ro,si] = rosi_bdf( k, j )
b  = zeros(k+1,1); b(2) = 1;
ro = vsolve( -j:k-j, b(:) );
si = zeros( k+1, 1 ); si( j+1 ) = 1;
end
```

- Coefficients are computed by **imposing consistency** of maximal order p:

$$\sum_{j=0}^{k} (j^s \alpha_j - s j^{s-1} \beta_j) = 0,$$

$s = 0, 1, \ldots, p.$

Generalized BDF

First we need to compute $\rho(z)$ and $\sigma(z)$

```
function [ro,si] = rosi_bdf( k, j )
b  = zeros(k+1,1); b(2) = 1;
ro = vsolve( -j:k-j, b(:) );
si = zeros( k+1, 1 ); si( j+1 ) = 1;
end
```

- Coefficients are computed by **imposing consistency** of maximal order p:

$$\sum_{j=0}^{k} (j^s \alpha_j - s j^{s-1} \beta_j) = 0,$$

$s = 0, 1, \ldots, p.$

```
function f = vsolve( x, b )
f = b;
n = length( x )-1;
for k = 1:n
 for i = n+1:-1:k+1
  f(i) = f(i) - x(k)*f(i-1);
 end
end
for k = n:-1:1
 for i = k+1:n+1
  f(i) = f(i)/( x(i) - x(i-k) );
 end
 for i = k:n
  f(i) = f(i) - f(i+1);
 end
end
end
```

Generalized BDF

Then we use the `ro_si` routine to build the A_m and B_m matrices

```
function [a,b] = mab( k, n )
nu = fix( (k+2)/2 );
a  = spalloc( n, n+1, (k+1)*n );
b = a;
for i = 1:nu
 [ro,si] = rosi_bdf( k, i );
 a(i,1:k+1) = ro.';
 b(i,1:k+1) = si.';
end
for i = nu+1:n-(k-nu)
 a(i,i+1+(-nu:k-nu)) = ro.';
 b(i,i+1+(-nu:k-nu)) = si.';
end
```

```
j = nu;
for i = n-(k-nu)+1:n
 j = j + 1;
 [ro,si] = rosi_bdf( k, j );
 a(i,n+1+(-k:0)) = ro.';
 b(i,n+1+(-k:0)) = si.';
end
end
```

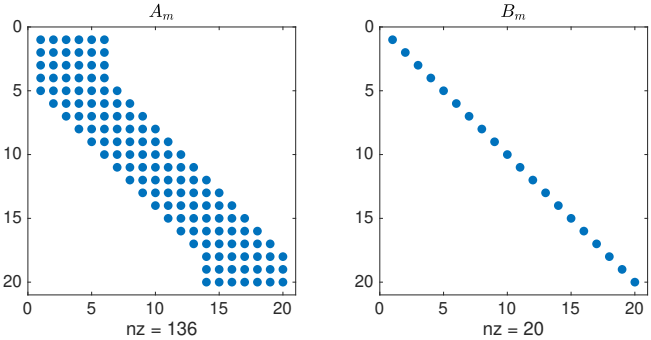
- `for i = 1:nu; end`, initial conditions,
- `for i = nu+1:n-(k-nu); end`, Toepltiz part,
- `for i = n-(k-nu)+1:n; end`, final conditions.

Generalized BDF

We can use the routine to generate

```
[Alpha,Beta] = mab(k,m); A = Alpha(:,2:m+1); B = Beta(:,2:m+1);
```

and visualize them



The first column contains the coefficients needed to compute the right-hand-side.

Generalized BDF

We now need to build the right-hand-side

```
nk=n*(m+1);
b=zeros(nk,1);  % Allocate the space for one more than needed
for j=1:m  % Use the source to build the rhs:
 b(1+j*n:(j+1)*n)=f(x,t0+j*h);
end
b(n+1:n*(m+1))=h*kron(Beta,speye(n))*b; % Correct with the betas coeff.s
b(1:n)=u0;                  % First block as the initial condition
% Correction coefficients:
Am = kron(Alpha(:,1),speye(n))-h*kron(Beta(:,1),L);
b(n+1:nk)=b(n+1:nk)-Am*u0; % Finish building RHS
```

And then we can solve the linear system

```
Mat = kron(A,M) - h*kron(B,L); rhs = b(n+1:nk);
u = Mat\rhs;
```

Generalized BDF

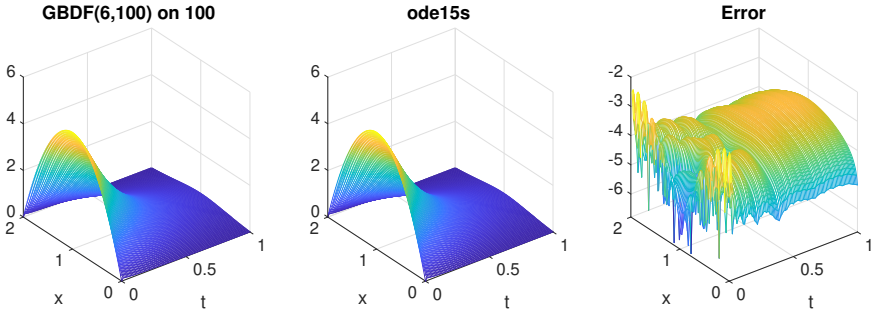
We can compare the solution with `ode15s`:

```
U = [u0,reshape(u,n,m)]; t = t0:h:tf;
[TT,UU] = ode15s(@(t,y) L*y +
↪  f(x.',t),t,u0);
E = abs(U-reshape(UU,m+1,n).');
figure(2)
subplot(1,3,1)
mesh(t,x,U);
xlabel('t');
ylabel('x');
title('GBDF(6,100) on 100')
```

```
subplot(1,3,2)
mesh(t,x,reshape(UU,m+1,n).')
xlabel('t');
ylabel('x');
title('ode15s')
subplot(1,3,3)
mesh(t,x,log10())
xlabel('t');
ylabel('x');
title('Error')
```

Generalized BDF

We can compare the solution with `ode15s`:



❓ What happens if we attempt solution via our matrix-equation solver?

Generalized BDF

We can solve it by doing:

```
maxit = 100;
tol = 1e-9;
[LL,UL] = lu(-h*L);
[LA,UA] = lu(A);
[X1,X2,res]=kpik_sylv(-h*L,LL,UL,A,
↪  LA,UA,C1,C2,maxit,tol);
```

Using our non-symmetric test problem with variable coefficients and fractional order α.

Generalized BDF

We can solve it by doing:

```
maxit = 100;
tol = 1e-9;
[LL,UL] = lu(-h*L);
[LA,UA] = lu(A);
[X1,X2,res]=kpik_sylv(-h*L,LL,UL,A,
↪  LA,UA,C1,C2,maxit,tol);
```

Using our non-symmetric test problem with variable coefficients and fractional order α.

k	m	n	IT	Res.
2	32	64	16	1.08e-15
2	64	128	23	2.16e-10
2	128	256	30	4.72e-10
2	256	512	38	9.20e-10
2	512	1024	49	7.31e-10
2	1024	2048	62	7.82e-10
2	2048	4096	78	8.06e-10
2	4096	8192	97	9.24e-10

Generalized BDF

We can solve it by doing:

```
maxit = 100;
tol = 1e-9;
[LL,UL] = lu(-h*L);
[LA,UA] = lu(A);
[X1,X2,res]=kpik_sylv(-h*L,LL,UL,A,
↪  LA,UA,C1,C2,maxit,tol);
```

Using our non-symmetric test problem with variable coefficients and fractional order α.

k	m	n	IT	Res.
3	32	64	15	7.18e-10
3	64	128	20	9.80e-10
3	128	256	26	7.77e-10
3	256	512	34	4.21e-10
3	512	1024	43	5.75e-10
3	1024	2048	54	8.05e-10
3	2048	4096	68	8.84e-10
3	4096	8192	85	9.87e-10

Generalized BDF

We can solve it by doing:

```
maxit = 100;
tol = 1e-9;
[LL,UL] = lu(-h*L);
[LA,UA] = lu(A);
[X1,X2,res]=kpik_sylv(-h*L,LL,UL,A,
↪  LA,UA,C1,C2,maxit,tol);
```

Using our non-symmetric test problem with variable coefficients and fractional order α.

k	m	n	IT	Res.
4	32	64	16	1.19e-14
4	64	128	24	3.22e-10
4	128	256	31	4.05e-10
4	256	512	39	6.97e-10
4	512	1024	50	6.20e-10
4	1024	2048	63	7.70e-10
4	2048	4096	79	9.05e-10
4	4096	8192	99	9.05e-10

Generalized BDF

We can solve it by doing:

```
maxit = 100;
tol = 1e-9;
[LL,UL] = lu(-h*L);
[LA,UA] = lu(A);
[X1,X2,res]=kpik_sylv(-h*L,LL,UL,A,
↪  LA,UA,C1,C2,maxit,tol);
```

Using our non-symmetric test problem with variable coefficients and fractional order α.

k	m	n	IT	Res.
5	32	64	16	1.72e-14
5	64	128	22	2.96e-10
5	128	256	28	4.90e-10
5	256	512	36	5.56e-10
5	512	1024	46	5.53e-10
5	1024	2048	58	7.10e-10
5	2048	4096	73	8.04e-10
5	4096	8192	91	9.75e-10

Generalized BDF

We can solve it by doing:

```
maxit = 100;
tol = 1e-9;
[LL,UL] = lu(-h*L);
[LA,UA] = lu(A);
[X1,X2,res]=kpik_sylv(-h*L,LL,UL,A,
↪  LA,UA,C1,C2,maxit,tol);
```

Using our non-symmetric test problem with variable coefficients and fractional order α.

k	m	n	IT	Res.
6	32	64	16	3.46e-14
6	64	128	24	4.70e-10
6	128	256	31	5.73e-10
6	256	512	40	4.78e-10
6	512	1024	50	9.39e-10
6	1024	2048	64	7.69e-10
6	2048	4096	81	7.31e-10
6	4096	8192	100	1.10e-09

Generalized BDF

We can solve it by doing:

```
maxit = 100;
tol = 1e-9;
[LL,UL] = lu(-h*L);
[LA,UA] = lu(A);
[X1,X2,res]=kpik_sylv(-h*L,LL,UL,A,
↪  LA,UA,C1,C2,maxit,tol);
```

Using our non-symmetric test problem with variable coefficients and fractional order α.

k	m	n	IT	Res.
7	32	64	16	6.13e-15
7	64	128	22	6.60e-10
7	128	256	29	4.78e-10
7	256	512	37	7.04e-10
7	512	1024	47	8.47e-10
7	1024	2048	60	7.66e-10
7	2048	4096	76	7.36e-10
7	4096	8192	95	8.46e-10

Generalized BDF

We can solve it by doing:

```
maxit = 100;
tol = 1e-9;
[LL,UL] = lu(-h*L);
[LA,UA] = lu(A);
[X1,X2,res]=kpik_sylv(-h*L,LL,UL,A,
↪  LA,UA,C1,C2,maxit,tol);
```

Using our non-symmetric test problem with variable coefficients and fractional order α.

k	m	n	IT	Res.
8	32	64	16	2.46e-14
8	64	128	24	5.41e-10
8	128	256	31	7.57e-10
8	256	512	40	6.53e-10
8	512	1024	51	7.34e-10
8	1024	2048	65	6.98e-10
8	2048	4096	82	7.42e-10
8	4096	8192	100	1.56e-09

Generalized BDF

We can solve it by doing:

```
maxit = 100;
tol = 1e-9;
[LL,UL] = lu(-h*L);
[LA,UA] = lu(A);
[X1,X2,res]=kpik_sylv(-h*L,LL,UL,A,
↪  LA,UA,C1,C2,maxit,tol);
```

Using our non-symmetric test problem with variable coefficients and fractional order α.

k	m	n	IT	Res.
8	32	64	16	2.46e-14
8	64	128	24	5.41e-10
8	128	256	31	7.57e-10
8	256	512	40	6.53e-10
8	512	1024	51	7.34e-10
8	1024	2048	65	6.98e-10
8	2048	4096	82	7.42e-10
8	4096	8192	100	1.56e-09

- The solution seems to be robust with respect to k,

Generalized BDF

We can solve it by doing:

```
maxit = 100;
tol = 1e-9;
[LL,UL] = lu(-h*L);
[LA,UA] = lu(A);
[X1,X2,res]=kpik_sylv(-h*L,LL,UL,A,
↪  LA,UA,C1,C2,maxit,tol);
```

Using our non-symmetric test problem with variable coefficients and fractional order α.

k	m	n	IT	Res.
8	32	64	16	2.46e-14
8	64	128	24	5.41e-10
8	128	256	31	7.57e-10
8	256	512	40	6.53e-10
8	512	1024	51	7.34e-10
8	1024	2048	65	6.98e-10
8	2048	4096	82	7.42e-10
8	4096	8192	100	1.56e-09

- The solution seems to be robust with respect to k,
- We still have a small increase with n and m.

Structured preconditioner

Let's now look for a different approach.

Structured preconditioner

Let's now look for a different approach.

- We can do matrix vector products with the system matrix without assembling the matrix:

```
function [y] = Mprod(A,B,L,h,x)
[sp1,~] = size(A);
[m,~] = size(L);
X = reshape(x,m,sp1);
Y = X*A' - h*(L*X*B');
y = reshape(Y,m*sp1,1);
end
```

Structured preconditioner

Let's now look for a different approach.

- We can do matrix vector products with the system matrix without assembling the matrix:

```
function [y] = Mprod(A,B,L,h,x)
[sp1,~] = size(A);
[m,~] = size(L);
X = reshape(x,m,sp1);
Y = X*A' - h*(L*X*B');
y = reshape(Y,m*sp1,1);
end
```

- The linear system is **not symmetric**: we can use either GMRES or Flexible-GMRES to solve it.

Structured preconditioner

Let's now look for a different approach.

- We can do matrix vector products with the system matrix without assembling the matrix:

```
function [y] = Mprod(A,B,L,h,x)
[sp1,~] = size(A);
[m,~] = size(L);
X = reshape(x,m,sp1);
Y = X*A' - h*(L*X*B');
y = reshape(Y,m*sp1,1);
end
```

- The linear system is **not symmetric**: we can use either GMRES or Flexible-GMRES to solve it.
- We just need to *figure out* a *preconditioner*.

Structured preconditioner

The 🔑 idea is *again* using a preconditioner that has the same structure:

$$P = \breve{A}_m \otimes M_n - h\breve{B}_m \otimes \tilde{L}_n,$$

- This idea comes from (Bertaccini 2000, 2001; Bertaccini and Ng 2001),

Structured preconditioner

The idea is *again* using a preconditioner that has the same structure:

$$P = \breve{A}_m \otimes M_n - h\breve{B}_m \otimes \tilde{L}_n,$$

- This idea comes from (Bertaccini 2000, 2001; Bertaccini and Ng 2001),
- How do we select the approximations $\breve{A}_m$, $\breve{B}_m$ and $\tilde{L}_n$?

Structured preconditioner

The 🔑 idea is *again* using a preconditioner that has the same structure:

$$P = \breve{A}_m \otimes M_n - h\breve{B}_m \otimes \tilde{L}_n,$$

- This idea comes from (Bertaccini 2000, 2001; Bertaccini and Ng 2001),
- How do we select the approximations $\breve{A}_m$, $\breve{B}_m$ and $\tilde{L}_n$?
  - A_m, B_m are Toeplitz + low-rank $\Rightarrow$ Circulant or Fast-Transform preconditioners,
  - $\tilde{L}_n$ has the *quasi-Toeplitz structure* we have seen, so we can use some of the techniques we had already seen for this; (Bertaccini and Durastante 2018).

Structured preconditioner

The idea is *again* using a preconditioner that has the same structure:

$$P = \breve{A}_m \otimes M_n - h\breve{B}_m \otimes \tilde{L}_n,$$

- This idea comes from (Bertaccini 2000, 2001; Bertaccini and Ng 2001),
- How do we select the approximations $\breve{A}_m$, $\breve{B}_m$ and $\tilde{L}_n$?
  - A_m, B_m are Toeplitz + low-rank $\Rightarrow$ Circulant or Fast-Transform preconditioners,
  - $\tilde{L}_n$ has the *quasi-Toeplitz structure* we have seen, so we can use some of the techniques we had already seen for this; (Bertaccini and Durastante 2018).
- It would be good to also have a **parallel way of applying the preconditioner**.

Structured preconditioner

- If $\breve{A}_m$ and $\breve{B}_m$ are *circulant–like approximations* of the Toeplitz (+ "low rank") matrices A_m and B_m, and the mass matrix is the identity, then we can express the **eigenvalues** of P as

$$\phi_i - h\psi_i\lambda_j, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n,$$

where

- $\{\phi_i\}$ are the eigenvalues of the circulant–like approximation $\breve{A}$,
- $\{\psi_i\}$ are the eigenvalues of the circulant–like approximation $\breve{B}$,
- $\{\lambda_j\}$ are the eigenvalues of the selected approximation of J_n.

Structured preconditioner

- If $\breve{A}_m$ and $\breve{B}_m$ are *circulant–like approximations* of the Toeplitz (+ "low rank") matrices A_m and B_m, and the mass matrix is the identity, then we can express the **eigenvalues** of P as

$$\phi_i - h\psi_i\lambda_j, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n,$$

  where
  - $\{\phi_i\}$ are the eigenvalues of the circulant–like approximation $\breve{A}$,
  - $\{\psi_i\}$ are the eigenvalues of the circulant–like approximation $\breve{B}$,
  - $\{\lambda_j\}$ are the eigenvalues of the selected approximation of J_n.
- What circulant-like approximation do we want?

Structured preconditioner

- If $\breve{A}_m$ and $\breve{B}_m$ are *circulant–like approximations* of the Toeplitz (+ "low rank") matrices A_m and B_m, and the mass matrix is the identity, then we can express the **eigenvalues** of P as

$$\phi_i - h\psi_i\lambda_j, \qquad i = 1, \ldots, m, \quad j = 1, \ldots, n,$$

  where
  - $\{\phi_i\}$ are the eigenvalues of the circulant–like approximation $\breve{A}$,
  - $\{\psi_i\}$ are the eigenvalues of the circulant–like approximation $\breve{B}$,
  - $\{\lambda_j\}$ are the eigenvalues of the selected approximation of J_n.
- What circulant-like approximation do we want?
  - An idea could be using Strang approximation (Gu et al. 2015)

$$P_{\mathfrak{s}} = \mathfrak{s}(A_m) \otimes I_m - h\mathfrak{s}(B_m) \otimes L_n,$$

Structured preconditioner

$$P_{\mathfrak{s}} = \mathfrak{s}(A_m) \otimes I_m - h\mathfrak{s}(B_m) \otimes L_n,$$

$$\mathfrak{s}(A) = \begin{bmatrix} \alpha_\nu & \cdots & \alpha_k & & & & \alpha_0 & \cdots & \alpha_{\nu-1} \\ \vdots & \ddots & & \ddots & & & & \ddots & \vdots \\ \alpha_0 & & \ddots & & \ddots & & & & \alpha_0 \\ & \ddots & & \ddots & & \ddots & & 0 & \\ & & \ddots & & \ddots & & \ddots & & \\ & 0 & & \ddots & & \ddots & & \ddots & \\ \alpha_k & & & & \ddots & & \ddots & & \alpha_k \\ \vdots & \ddots & & & & \ddots & & \ddots & \vdots \\ \alpha_{\nu+1} & \cdots & \alpha_k & & & & \alpha_0 & \cdots & \alpha_\nu \end{bmatrix},$$

- $\mathfrak{s}(B)$ can be built analogously.

Structured preconditioner

$$P_{\mathfrak{s}} = \mathfrak{s}(A_m) \otimes I_m - h\mathfrak{s}(B_m) \otimes L_n,$$

$$\mathfrak{s}(A) = \begin{bmatrix} \alpha_\nu & \cdots & \alpha_k & & & & \alpha_0 & \cdots & \alpha_{\nu-1} \\ \vdots & \ddots & & \ddots & & & & \ddots & \vdots \\ \alpha_0 & & \ddots & & \ddots & & & & \alpha_0 \\ & \ddots & & \ddots & & \ddots & & 0 & \\ & & \ddots & & \ddots & & \ddots & & \\ & 0 & & \ddots & & \ddots & & \ddots & \\ \alpha_k & & & & \ddots & & \ddots & & \alpha_k \\ \vdots & \ddots & & & & \ddots & & \ddots & \vdots \\ \alpha_{\nu+1} & \cdots & \alpha_k & & & & \alpha_0 & \cdots & \alpha_\nu \end{bmatrix},$$

- $\mathfrak{s}(B)$ can be built analogously.
- $\mathfrak{s}(A)$ is singular due to the consistency condition.

Structured preconditioner

$$P_{\tilde{\mathfrak{s}}} = \tilde{\mathfrak{s}}(A)_m \otimes I_n - h\tilde{\mathfrak{s}}(B)_m \otimes L_n.$$

$$\mathfrak{s}(A) = \begin{bmatrix} \alpha_\nu & \cdots & \alpha_k & & & & \alpha_0 & \cdots & \alpha_{\nu-1} \\ \vdots & \ddots & & \ddots & & & & \ddots & \vdots \\ \alpha_0 & & \ddots & & \ddots & & & & \alpha_0 \\ & \ddots & & \ddots & & \ddots & & 0 & \\ & & \ddots & & \ddots & & \ddots & & \\ & 0 & & \ddots & & \ddots & & \ddots & \\ \alpha_k & & & & \ddots & & \ddots & & \alpha_k \\ \vdots & \ddots & & & & \ddots & & \ddots & \vdots \\ \alpha_{\nu+1} & \cdots & \alpha_k & & & & \alpha_0 & \cdots & \alpha_\nu \end{bmatrix},$$

- $\mathfrak{s}(B)$ can be built analogously.
- $\mathfrak{s}(A)$ is singular due to the consistency condition.
- It is a single 0 eigenvalue, so we can move it by a rank 1 perturbation: $\tilde{\mathfrak{s}}(\cdot)$.

Structured preconditioner

$$P_{\tilde{\mathfrak{s}}} = \tilde{\mathfrak{s}}(A)_m \otimes I_n - h\tilde{\mathfrak{s}}(B)_m \otimes L_n.$$

$$\mathfrak{s}(A) = \begin{bmatrix} \alpha_\nu & \cdots & \alpha_k & & & \alpha_0 & \cdots & \alpha_{\nu-1} \\ \vdots & \ddots & & \ddots & & & \ddots & \vdots \\ \alpha_0 & & \ddots & & \ddots & & & \alpha_0 \\ & \ddots & & \ddots & & \ddots & 0 & \\ & & \ddots & & \ddots & & & \\ & 0 & & \ddots & & \ddots & & \\ \alpha_k & & & & \ddots & & \ddots & \alpha_k \\ \vdots & \ddots & & & & \ddots & & \vdots \\ \alpha_{\nu+1} & \cdots & \alpha_k & & & \alpha_0 & \cdots & \alpha_\nu \end{bmatrix},$$

Proposition (Bertaccini 2001, Proposition 4.1)

If L has eigenvalues μ_r such that $\Re(\mu_r) < -\delta < 0$, $r = 1, \ldots, m$. Then the preconditioner $P_{\tilde{\mathfrak{s}}}$ is invertible for $A_{\nu,k-\nu}$-stable formulae.

Structured preconditioner

❓ What can we say about the **clustering properties** of this preconditioner?

Theorem (Bertaccini 2000, Theorem 4.1)

Let $\mathcal{M} = A_m \otimes I_n - hB_m \otimes L_n$ for an $A_{\nu,k-\nu}$-stable formulae with k steps. Let P be the block circulant preconditioner

$$P = \breve{A}_m \otimes M_n - h\breve{B}_m \otimes L_n.$$

Then, for fixed $\delta > 0$, there exists $C_\delta \geq 0$, $m_\delta \geq k$ such that, for all $m \geq m_\delta$ ($m+1$ is the size of A and B),

$$P^{-1}M = I + M_\delta^{(1)} + M_\delta^{(2)},$$

where $\mathrm{rank}(M_\delta^{(2)}) \leq n[2(k+1) + C_\delta]$ and $\|M_\delta^{(1)}\|_2 \leq \delta c_L$ does not depend on m. If P is defined as Strang's circulant preconditioner, then $C_\delta = \|M_\delta^{(1)}\| = 0$.

Structured preconditioner

Another available choice is using instead $\{\omega\}$-Circulant matrices, i.e.,

$$P_\omega = \omega(A_m) \otimes I_n - h\omega(B_m) \otimes L_n,$$

$$\omega(A_m) = \begin{bmatrix} \alpha_\nu & \cdots & \alpha_k & & & & \omega\alpha_0 & \cdots & \omega\alpha_{\nu-1} \\ \vdots & \ddots & & \ddots & & & & \ddots & \vdots \\ \alpha_0 & & \ddots & & \ddots & & & & \omega\alpha_0 \\ & \ddots & & \ddots & & \ddots & & 0 & \\ & & \ddots & & \ddots & & \ddots & & \\ & 0 & & \ddots & & \ddots & & \ddots & \\ \omega\alpha_k & & & & \ddots & & \ddots & & \alpha_k \\ \vdots & \ddots & & & & \ddots & & \ddots & \vdots \\ \omega\alpha_{\nu+1} & \cdots & \omega\alpha_k & & & & \alpha_0 & \cdots & \alpha_\nu \end{bmatrix},$$

- $\omega(B_m)$ is defined similarly.
- The usual choice is setting $\omega = -1$, i.e., the skew-circulant preconditioner.

Stuctured preconditioner: application

To apply

$$P_\omega^{-1}\mathbf{v} = (\omega(A_m) \otimes I_n - h\omega(B_m) \otimes L_n)^{-1}\mathbf{v},$$

We can use the **diagonalization** of $\omega(A_m)$ and $\omega(B_m)$, i.e.,

$$P_\omega^{-1}\mathbf{v} = (F\Omega \otimes I_n)^{-1}(\Lambda_A \otimes I_n - h\Lambda_B \otimes L_n)^{-1}(\Omega^H F^H \otimes I_n)^{-1}\mathbf{v}.$$

1. Compute $\mathbf{w} = (\Omega^* F^* \otimes I_m)^{-1}\mathbf{v} = -V\Omega^{-H}F$,
2. Solve $(\Lambda_A \otimes I_n - h\Lambda_B \otimes L_n)^{-1}\mathbf{w}$ by solving

   $$(\lambda_i(A)I_n - h\lambda_i(B)L_n)\mathbf{z}_i = \mathbf{w}_i, \quad i = 1, \ldots, m$$

   with $\mathrm{vec}([\mathbf{w}_1, \ldots, \mathbf{w}_m]) = \mathbf{w}$, and similarly for $\mathbf{z}$,
3. Compute $\mathbf{y} = (F\Omega \otimes I_n)^{-1}\mathbf{z} = -ZF^H\Omega^{-1}$.

Stuctured preconditioner: application

To apply

$$P_\omega^{-1}\mathbf{v} = (\omega(A_m) \otimes I_n - h\omega(B_m) \otimes L_n)^{-1}\mathbf{v},$$

We can use the **diagonalization** of $\omega(A_m)$ and $\omega(B_m)$, i.e.,

$$P_\omega^{-1}\mathbf{v} = (F\Omega \otimes I_n)^{-1}(\Lambda_A \otimes I_n - h\Lambda_B \otimes L_n)^{-1}(\Omega^H F^H \otimes I_n)^{-1}\mathbf{v}.$$

1. Compute $\mathbf{w} = (\Omega^* F^* \otimes I_m)^{-1}\mathbf{v} = -V\Omega^{-H}F$,
2. Solve $(\Lambda_A \otimes I_n - h\Lambda_B \otimes L_n)^{-1}\mathbf{w}$ by solving

$$(\lambda_i(A)I_n - h\lambda_i(B)L_n)\mathbf{z}_i = \mathbf{w}_i, \quad i = 1, \ldots, m$$

   with $\mathrm{vec}([\mathbf{w}_1, \ldots, \mathbf{w}_m]) = \mathbf{w}$, and similarly for $\mathbf{z}$,
3. Compute $\mathbf{y} = (F\Omega \otimes I_n)^{-1}\mathbf{z} = -ZF^H\Omega^{-1}$.

❗ This step is **embarrassingly parallel**!

Numerical example

We use our favorite test problem with the space variant, nonsymmetric fractional operator in space and $\alpha = 1.5$, using GMRES(20) with a tolerance of `1e-9` using the P_{-1} preconditioner.

$k = 2$			$k = 3$			$k = 4$			$k = 5$			$k = 6$		
n	m	lt	n	m	lt	n	m	lt	n	m	lt	n	m	lt
64	32	30	64	32	32	64	32	35	64	32	38	64	32	46
128	64	31	128	64	33	128	64	38	128	64	45	128	64	53
256	128	31	256	128	34	256	128	39	256	128	48	256	128	58
512	256	31	512	256	34	512	256	39	512	256	50	512	256	62
1024	512	30	1024	512	33	1024	512	37	1024	512	49	1024	512	60

Numerical example

We use our favorite test problem with the space variant, nonsymmetric fractional operator in space and $\alpha = 1.5$, using GMRES(20) with a tolerance of `1e-9` using the P_{-1} preconditioner.

$k = 2$			$k = 3$			$k = 4$			$k = 5$			$k = 6$		
n	m	lt	n	m	lt	n	m	lt	n	m	lt	n	m	lt
64	32	30	64	32	32	64	32	35	64	32	38	64	32	46
128	64	31	128	64	33	128	64	38	128	64	45	128	64	53
256	128	31	256	128	34	256	128	39	256	128	48	256	128	58
512	256	31	512	256	34	512	256	39	512	256	50	512	256	62
1024	512	30	1024	512	33	1024	512	37	1024	512	49	1024	512	60

👁 Reduced ☺ **iteration dependence**, but paid with ☹ **full memory price**!

Further modifications

We can further approximate the preconditioner by selecting instead of L_n in

$$P_\omega^{-1}\mathbf{v} = (\omega(A_m) \otimes I_n - h\omega(B_m) \otimes L_n)^{-1}\mathbf{v},$$

a suitable approximation, e.g.,

- $g_k(L_n)$ a bandwidth k approximation of the dense L_n matrix, i.e., using the information on the decay of the coefficients (Bertaccini and Durastante 2018).
- A *structured preconditioner* based on GLT theory.

Further modifications

We can further approximate the preconditioner by selecting instead of L_n in

$$P_\omega^{-1}\mathbf{v} = (\omega(A_m) \otimes I_n - h\omega(B_m) \otimes L_n)^{-1}\mathbf{v},$$

a suitable approximation, e.g.,

- $g_k(L_n)$ a bandwidth k approximation of the dense L_n matrix, i.e., using the information on the decay of the coefficients (Bertaccini and Durastante 2018).
- A *structured preconditioner* based on GLT theory.

Open areas of research

- Efficient solution strategies for the $\lambda_i(A)I_n - h\lambda_i(B)L_n$ systems,
- Load-balancing issues for parallelism,
- Optimal poles selection for the matrix-equation based solvers,
- Multigrid solvers/preconditioners for $(A_m \otimes M_n - hB_m \otimes L_n)\mathbf{u} = \mathbf{f}$.

Tensor Equations

A *different approach* that can be of interest is to use another structure.

Tensor Equations

A *different approach* that can be of interest is to use another structure.

Let us suppose that L_n is obtained as the discretization of a *multidimensional fractional operator*, i.e.,

$$L_n = \sum_{i=1}^{\ell} \left(K^{-}_{m,\ell} \bigotimes_{p=1}^{i-1} I \otimes G^{(\ell)}_{n^{1/\ell}} \otimes \bigotimes_{p=1}^{\ell-1} I + K^{+}_{n,\ell} \bigotimes_{p=1}^{i-1} I \otimes {G^{(\ell)}_{n^{1/\ell}}}^{T} \otimes \bigotimes_{p=1}^{\ell-1} I \right)$$

where $K^{\pm}_{m,\ell}$ have also a Kronecker tensor structure whenever the functions $\{\kappa_j\}_{j=1}^{\ell}$ are separable in the x_j variables.

Tensor Equations

A *different approach* that can be of interest is to use another structure.

Let us suppose that L_n is obtained as the discretization of a *multidimensional fractional operator*, i.e.,

$$L_n = \sum_{i=1}^{\ell} \left(K^{-}_{m,\ell} \bigotimes_{p=1}^{i-1} I \otimes G^{(\ell)}_{n^{1/\ell}} \otimes \bigotimes_{p=1}^{\ell-1} I + K^{+}_{n,\ell} \bigotimes_{p=1}^{i-1} I \otimes {G^{(\ell)}_{n^{1/\ell}}}^{T} \otimes \bigotimes_{p=1}^{\ell-1} I \right)$$

where $K^{\pm}_{m,\ell}$ have also a Kronecker tensor structure whenever the functions $\{\kappa_j\}_{j=1}^{\ell}$ are separable in the x_j variables.

The matrix: $\mathcal{M} = A_m \otimes I_n - hB_m \otimes L_n$ has now a lot of **redundant information**!

Tensor Equations: thou shalt compress!

As we have done for the *hierarchical formats*, we want

- A **compressed representation** of $\mathcal{M}$, possibly with a number of parameters that grows poly-logarithmically with the overall size...
- A **fast BLAS-like toolbox** to solve our problem in this format.

Tensor Equations: thou shalt compress!

As we have done for the *hierarchical formats*, we want

- A **compressed representation** of $\mathcal{M}$, possibly with a number of parameters that grows poly-logarithmically with the overall size...
- A **fast BLAS-like toolbox** to solve our problem in this format.

There exists *many formats* for which this is possible, *e.g.*, the CANDECOMP/PARAFAC (CP) decomposition, the Tucker format, the Tensor Train (TT), the TT-Tucker, *etc.*; see (Kolda and Bader 2009).

Tensor Equations: thou shalt compress!

As we have done for the *hierarchical formats*, we want

- A **compressed representation** of $\mathcal{M}$, possibly with a number of parameters that grows poly-logarithmically with the overall size...
- A **fast BLAS-like toolbox** to solve our problem in this format.

There exists *many formats* for which this is possible, *e.g.*, the CANDECOMP/PARAFAC (CP) decomposition, the Tucker format, the Tensor Train (TT), the TT-Tucker, *etc.*; see (Kolda and Bader 2009).

We focus on the **T**ensor-**T**rain format, since it has a simple enough toolbox to work with: TT-Toolbox.

Tensor-Train

❓ But what is a *tensor*?

Tensor-Train

But what is a *tensor*?

A **tensor** is a **multidimensional array**, $a \in \mathbb{R}$ is a 0-tensor, $\mathbf{v} \in \mathbb{R}^{n_1}$ is a 1-tensor, $A \in \mathbb{R}^{n_1 \times n_2}$ is a 2-tensor, $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a 3-tensor, ...

Tensor-Train

But what is a *tensor*?

- A **tensor** is a **multidimensional array**, $a \in \mathbb{R}$ is a 0-tensor, $\mathbf{v} \in \mathbb{R}^{n_1}$ is a 1-tensor, $A \in \mathbb{R}^{n_1 \times n_2}$ is a 2-tensor, $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a 3-tensor, ...
- A **tensor** is a **multilinear maps** with respect to a fixed finite-dimensional $\mathbb{R}$ vector space V

$$\mathcal{A} : \underbrace{V^* \times \cdots \times V^*}_{p} \times \underbrace{V \times \cdots \times V}_{q} \to \mathbb{R},$$

Tensor-Train

❓ But what is a *tensor*?

- A **tensor** is a **multidimensional array**, $a \in \mathbb{R}$ is a 0-tensor, $\mathbf{v} \in \mathbb{R}^{n_1}$ is a 1-tensor, $A \in \mathbb{R}^{n_1 \times n_2}$ is a 2-tensor, $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a 3-tensor, ...
- A **tensor** is a **multilinear maps** with respect to a fixed finite-dimensional $\mathbb{R}$ vector space V

$$\mathcal{A} : \underbrace{V^* \times \cdots \times V^*}_{p} \times \underbrace{V \times \cdots \times V}_{q} \to \mathbb{R},$$

- A **tensor** is an **element of the tensor product** of vector spaces

$$\mathcal{A} \in \underbrace{V \times \cdots \times V}_{p} \otimes \underbrace{V^* \otimes \cdots \otimes V^*}_{q}.$$

Tensor-Train

❓ But what is a *tensor*?

- A **tensor** is a **multidimensional array**, $a \in \mathbb{R}$ is a 0-tensor, $\mathbf{v} \in \mathbb{R}^{n_1}$ is a 1-tensor, $A \in \mathbb{R}^{n_1 \times n_2}$ is a 2-tensor, $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a 3-tensor, ...
- A **tensor** is a **multilinear maps** with respect to a fixed finite-dimensional $\mathbb{R}$ vector space V

$$\mathcal{A} : \underbrace{V^* \times \cdots \times V^*}_{p} \times \underbrace{V \times \cdots \times V}_{q} \to \mathbb{R},$$

- A **tensor** is an **element of the tensor product** of vector spaces

$$\mathcal{A} \in \underbrace{V \times \cdots \times V}_{p} \otimes \underbrace{V^* \otimes \cdots \otimes V^*}_{q}.$$

The definition we select depends on the operations we want to perform.

Tensor-Train[1]

Let us start from trying to describe a *vector* associated with our discretization matrix $\mathcal{M}$.

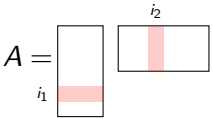
[1] For part of this material, a sincere thanks to Stefano Massei.

Tensor-Train[1]

Let us start from trying to describe a *vector* associated with our discretization matrix $\mathcal{M}$.

A rank-k matrix $A = U_1 U_2^T$ each entry is a dot product of vectors of length k

$$A(i_1, i_2) = U_1(i_1, :) \cdot U_2(:, i_2),$$



where the two indices select the left and right vectors.

[1] For part of this material, a sincere thanks to Stefano Massei.

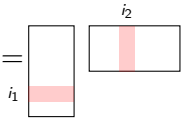
Tensor-Train[1]

Let us start from trying to describe a *vector* associated with our discretization matrix $\mathcal{M}$.

A rank-k matrix $A = U_1 U_2^T$ each entry is a dot product of vectors of length k

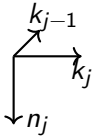
$$A(i_1, i_2) = U_1(i_1, :) \cdot U_2(:, i_2),$$

$A =$

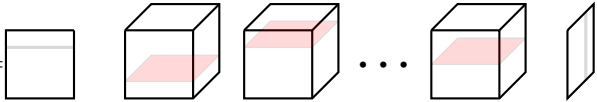


where the two indices select the left and right vectors. In a tensor of order d we insert $d-2$ matrices between the two vectors:

$$\mathcal{T}(i_1, \ldots, i_d) = U_1(i_1,\ :) \cdot U_2(:,\ :,\ i_2) \cdot\ \ldots\ \cdot U_{d-1}(:,\ :,\ i_{d-1}) \cdot U_d(:,\ i_d)$$



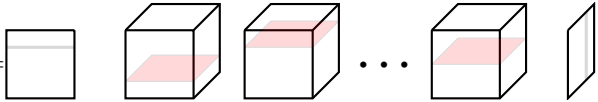
$\mathcal{T}(i_1, \ldots, i_d) =$



[1] For part of this material, a sincere thanks to Stefano Massei.

Tensor-Train

More formally, a tensor $\mathcal{T}$ is in TT decomposition if it can be written as

$\mathcal{T}(i_1, \ldots, i_d) =$ 

- Smallest possible tuple $(k_1, \ldots, k_{d-1})$ is called the **TT-rank** of $\mathcal{T}$.
- $U_j \in \mathbb{C}^{k_{j-1} \times n_j \times k_j}$ are called the **TT cores** of $\mathcal{T}$ (with $k_0 = k_d = 1$).
- If TT ranks are not large $\rightsquigarrow$ high compression ratio as d grows.
- TT decomposition multilinear with respect to the cores.

Tensor-Train

More formally, a tensor $\mathcal{T}$ is in TT decomposition if it can be written as

$$\mathcal{T}(i_1, \ldots, i_d) = \sum_{j_1=1}^{k_1} \cdots \sum_{j_{d-1}=1}^{k_{d-1}} U_1(i_1, j_1) U_2(j_1, i_2, j_2) \ldots U_d(j_{d-1}, i_d).$$

- Smallest possible tuple $(k_1, \ldots, k_{d-1})$ is called the **TT-rank** of $\mathcal{T}$.
- $U_j \in \mathbb{C}^{k_{j-1} \times n_j \times k_j}$ are called the **TT cores** of $\mathcal{T}$ (with $k_0 = k_d = 1$).
- If TT ranks are not large $\rightsquigarrow$ high compression ratio as d grows.
- TT decomposition multilinear with respect to the cores.

Tensor-Train

More formally, a tensor $\mathcal{T}$ is in TT decomposition if it can be written as

$$\mathcal{T}(i_1, \ldots, i_d) =$$

- Smallest possible tuple $(k_1, \ldots, k_{d-1})$ is called the **TT-rank** of $\mathcal{T}$.
- $U_j \in \mathbb{C}^{k_{j-1} \times n_j \times k_j}$ are called the **TT cores** of $\mathcal{T}$ (with $k_0 = k_d = 1$).
- If TT ranks are not large $\rightsquigarrow$ high compression ratio as d grows.
- TT decomposition multilinear with respect to the cores.

Tensor-Train

More formally, a tensor $\mathcal{T}$ is in TT decomposition if it can be written as

$$\mathcal{T}(i_1, \ldots, i_d) = \sum_{j_1=1}^{k_1} \cdots \sum_{j_{d-1}=1}^{k_{d-1}} U_1(i_1, j_1) U_2(j_1, i_2, j_2) \ldots U_d(j_{d-1}, i_d).$$

- Smallest possible tuple $(k_1, \ldots, k_{d-1})$ is called the **TT-rank** of $\mathcal{T}$.
- $U_j \in \mathbb{C}^{k_{j-1} \times n_j \times k_j}$ are called the **TT cores** of $\mathcal{T}$ (with $k_0 = k_d = 1$).
- If TT ranks are not large $\rightsquigarrow$ high compression ratio as d grows.
- TT decomposition multilinear with respect to the cores.

If for any $1 \leq \mu \leq d - 1$ we group the first μ factors and last $d - \mu$ factors then

$$\mathcal{T}(i_1, \ldots, i_\mu, i_{\mu+1}, \ldots, i_d),$$

is the matrix-matrix product of two (large) matrices.

TT decomposition and matrix factorizations

The **μth unfolding** of $\mathcal{T} \in \mathcal{C}^{n_1 \times \cdots \times n_d}$ is obtained by arranging the entries in a matrix

$$T^{<\mu>} \in \mathbb{C}^{(n_1 \cdots n_\mu) \times (n_{\mu+1} \cdots n_d)}$$

where the corresponding index map is given by

$$\begin{aligned} \mathrm{ind} : \mathbb{N}^{n_1 \times \cdots \times n_d} &\rightarrow \mathbb{N}^{(n_1 \cdots n_\mu) \times (n_{\mu+1} \cdots n_d)} \\ \mathrm{ind}(i_1, \ldots, i_d) &= (i_{row}, i_{col}), \end{aligned}$$

where

$$i_{row} = 1 + \sum_{s=1}^{\mu} (i_s - 1) \prod_{t=1}^{s-1} n_t,$$

$$i_{col} = 1 + \sum_{s=\mu+1}^{d} (i_s - 1) \prod_{t=\mu+1}^{s-1} n_t$$

TT decomposition and matrix factorizations

We can compute the **compression** of the **tensor** by computing the SVD of the *unfoldings*.

Lemma (Oseledets 2011)

The **TT rank** of a tensor $\mathcal{T}$ is given by

$$\text{tt-rank}(\mathcal{T}) = (\text{rank}(T^{<1>}), \ldots, \text{rank}(T^{<d-1>})).$$

Input: Tensor $\mathcal{T}$, ranks $k_1, \ldots, k_d$)
Output: $U_1, \ldots, U_d$.
$k_0 = k_d = 1$;
for $\mu = 1, \ldots, d-1$ **do**
  Reshape $\mathcal{T}$ into $T^{<2>} \in \mathbb{C}^{k_{\mu-1} n_\mu \times (n_{\mu+1} \ldots n_d)}$;
  Compute rank-k_μ approximation $T^{<2>} \approx U \Sigma V^T$ (e.g. via SVD);
  Reshape U into $U_\mu \in \mathbb{C}^{k_{\mu-1} \times n_\mu \times k_\mu}$;
  Update $\mathcal{T}$ via $T^{<2>} \leftarrow U^T X^{<2>} = \Sigma V^T$;
end
Set $U_d = \mathcal{T}$;

Algorithm 1: $\text{TT-SVD}(\mathcal{T}, k_1, \ldots, k_d)$

The **proof** is obtained by simply following the steps of the algorithm.

We can use *tolerances* instead of fixed ranks.

TT decomposition and matrix factorizations

And we can estimate the resulting error using the best approximation properties of the SVD.

Theorem (Oseledets 2011)

Let $\mathcal{T}_{SVD}$ denote the tensor in TT decomposition obtained from TT-SVD. Then

$$\|\mathcal{T} - \mathcal{T}_{SVD}\| \leq \sqrt{\epsilon_1^2 + \cdots + \epsilon_d^2}$$

where

$$\epsilon_\mu^2 = \| T^{<\mu>} - U\Sigma V^T \|_F^2 = \sigma_{k_\mu+1}^2 + \sigma_{k_\mu+2}^2 + \dots \ .$$

- We can modify the algorithm to accommodate different compression algorithms than the SVD,
- We can also compute the approximation via sketching algorithms, and avoiding using all the entries of $\mathcal{T}$.

TT-Matrices and matrix-vector products

If a **vector of length** $N = n_1 \times \ldots \times n_d$ is treated as a d**–dimensional tensor** with mode sizes n_k, and represented in TT–format, the **matrices acting on it** have the form

$$\mathcal{M}(i_1, \ldots, i_d, j_1, \ldots, j_d) = M_1(i_1, j_1) \ldots M(i_d, j_d), \qquad M_k(i_k, j_k) \in \mathbb{R}^{r_{k-1} \times r_k},$$

TT-Matrices and matrix-vector products

If a **vector of length** $N = n_1 \times \ldots \times n_d$ is treated as a d**–dimensional tensor** with mode sizes n_k, and represented in TT–format, the **matrices acting on it** have the form

$$\mathcal{M}(i_1, \ldots, i_d, j_1, \ldots, j_d) = M_1(i_1, j_1) \ldots M(i_d, j_d), \qquad M_k(i_k, j_k) \in \mathbb{R}^{r_{k-1} \times r_k},$$

- the first block indexes $i_1, \ldots, i_d$ enumerate the rows,

TT-Matrices and matrix-vector products

If a **vector of length** $N = n_1 \times \ldots \times n_d$ is treated as a d**–dimensional tensor** with mode sizes n_k, and represented in TT–format, the **matrices acting on it** have the form

$$\mathcal{M}(i_1, \ldots, i_d, j_1, \ldots, j_d) = M_1(i_1, j_1) \ldots M(i_d, j_d), \qquad M_k(i_k, j_k) \in \mathbb{R}^{r_{k-1} \times r_k},$$

- the first block indexes $i_1, \ldots, i_d$ enumerate the rows,
- the second block indexes $j_1, \ldots, j_d$ enumerate the columns.

Given $\mathcal{M}$ in TT–format, and a vector $\mathcal{X}$ in TT–format with cores X_k, and entries $X(j_1, \ldots, j_d)$ then the **matrix–vector multiplication** amounts to the following sum

$$\mathcal{Y}(i_1, \ldots, i_d) = \sum_{j_1, \ldots j_d} \mathcal{M}(i_1, \ldots, i_d, j_1, \ldots, j_d) \mathcal{X}(j_1, \ldots, j_d) = Y_1(i_1) \ldots Y_d(i_d),$$

where $Y_k(i_k) = \sum_{j_k} M_k(i_k, j_k) \otimes X_k(j_k)$

TT-Matrices and matrix-vector products

If a **vector of length** $N = n_1 \times \ldots \times n_d$ is treated as a d**–dimensional tensor** with mode sizes n_k, and represented in TT–format, the **matrices acting on it** have the form

$$\mathcal{M}(i_1, \ldots, i_d, j_1, \ldots, j_d) = M_1(i_1, j_1) \ldots M(i_d, j_d), \qquad M_k(i_k, j_k) \in \mathbb{R}^{r_{k-1} \times r_k},$$

- the first block indexes $i_1, \ldots, i_d$ enumerate the rows,
- the second block indexes $j_1, \ldots, j_d$ enumerate the columns.

Given $\mathcal{M}$ in TT–format, and a vector $\mathcal{X}$ in TT–format with cores X_k, and entries $X(j_1, \ldots, j_d)$ then the **matrix–vector multiplication** amounts to the following sum

$$\mathcal{Y}(i_1, \ldots, i_d) = \sum_{j_1, \ldots j_d} \mathcal{M}(i_1, \ldots, i_d, j_1, \ldots, j_d) \mathcal{X}(j_1, \ldots, j_d) = Y_1(i_1) \ldots Y_d(i_d),$$

where $Y_k(i_k) = \sum_{j_k} M_k(i_k, j_k) \otimes X_k(j_k)$ ⚠ The ranks of $\mathcal{Y}$ are the product of the ranks of the matrix and of the vector! So we need to **recompress** after every matrix-vector product.

TT-representation for our case

- We can use the same routine as before to *represent* the two BVM matrices,

```
%% Time-dependent operator
kval = 5;          % Grid power
m = 2^kval;        % Number of time
↪  steps
k = 2;
[Alpha,Beta] = mab(k,m);
A = Alpha(:,2:m+1);
B = Beta(:,2:m+1);
t0 = 0;
tf = 1;
h = (tf-t0)/m;
tA = tt_matrix(full(A),1e-14);
tA = tt_reshape(tA,2*ones(kval,2));
tB = tt_eye(2,kval);
```

TT-representation for our case

- We can use the same routine as before to *represent* the two BVM matrices,
- We build a tensor in which all the modes have size 2, this is usually called a Quantized-TT (QTT) formulation:

```
tA=tt_matrix(full(A),1e-14);
tA=tt_reshape(tA,2*ones(kval,2));
```

```
%% Time-dependent operator
kval = 5;          % Grid power
m = 2^kval;        % Number of time
↪  steps
k = 2;
[Alpha,Beta] = mab(k,m);
A = Alpha(:,2:m+1);
B = Beta(:,2:m+1);
t0 = 0;
tf = 1;
h = (tf-t0)/m;
tA = tt_matrix(full(A),1e-14);
tA = tt_reshape(tA,2*ones(kval,2));
tB = tt_eye(2,kval);
```

TT-representation for our case

- We can use the same routine as before to *represent* the two BVM matrices,
- We build a tensor in which all the modes have size 2, this is usually called a Quantized-TT (QTT) formulation:

```
tA=tt_matrix(full(A),1e-14);
tA=tt_reshape(tA,2*ones(kval,2));
```

- If we look at the values of k and maximal $\mathrm{tt\text{-}rank}$ we find:

k	2	3	4	5	6	7	8
$\max(\mathrm{tt\text{-}rank}(\mathcal{A}))$	3	5	6	7	7	7	9

```
%% Time-dependent operator
kval = 5;          % Grid power
m = 2^kval;        % Number of time
↪  steps
k = 2;
[Alpha,Beta] = mab(k,m);
A = Alpha(:,2:m+1);
B = Beta(:,2:m+1);
t0 = 0;
tf = 1;
h = (tf-t0)/m;
tA = tt_matrix(full(A),1e-14);
tA = tt_reshape(tA,2*ones(kval,2));
tB = tt_eye(2,kval);
```

TT-representation for our case

- We can act similarly also for the space operator.

```
%% Compression of the space part
tL = tt_matrix(L,1e-14);
tL = tt_reshape(tL,2*ones(kval+1,2));
tM = tt_eye(2,kval+1);
%% Final assembly
tMat = tkron(tA,tM)-h*tkron(tB,tL);
```

TT-representation for our case

- We can act similarly also for the space operator.
- We could be way more clever in the representation of these matrices, these are diagonal times Toeplitz, and we could do something specialized, *e.g.*, (Kazeev, Khoromskij, and Tyrtyshnikov 2013).

```
%% Compression of the space part
tL = tt_matrix(L,1e-14);
tL = tt_reshape(tL,2*ones(kval+1,2));
tM = tt_eye(2,kval+1);
%% Final assembly
tMat = tkron(tA,tM)-h*tkron(tB,tL);
```

TT-representation for our case

- We can act similarly also for the space operator.
- We could be way more clever in the representation of these matrices, these are diagonal times Toeplitz, and we could do something specialized, *e.g.*, (Kazeev, Khoromskij, and Tyrtyshnikov 2013).

```
%% Compression of the space part
tL = tt_matrix(L,1e-14);
tL = tt_reshape(tL,2*ones(kval+1,2));
tM = tt_eye(2,kval+1);
%% Final assembly
tMat = tkron(tA,tM)-h*tkron(tB,tL);
```

- Now that we have everything in this format, how can we solve our problem?

TT-representation for our case

- We can act similarly also for the space operator.
- We could be way more clever in the representation of these matrices, these are diagonal times Toeplitz, and we could do something specialized, *e.g.*, (Kazeev, Khoromskij, and Tyrtyshnikov 2013).

```
%% Compression of the space part
tL = tt_matrix(L,1e-14);
tL = tt_reshape(tL,2*ones(kval+1,2));
tM = tt_eye(2,kval+1);
%% Final assembly
tMat = tkron(tA,tM)-h*tkron(tB,tL);
```

- Now that we have everything in this format, how can we solve our problem?

TT-GMRES An option is to rephrase our favorite Krylov method using the TT arithmetic, (Dolgov 2013) and adapt what we know to build a preconditioner (Bertaccini and Durastante 2019).

TT-representation for our case

- We can act similarly also for the space operator.
- We could be way more clever in the representation of these matrices, these are diagonal times Toeplitz, and we could do something specialized, *e.g.*, (Kazeev, Khoromskij, and Tyrtyshnikov 2013).

```
%% Compression of the space part
tL = tt_matrix(L,1e-14);
tL = tt_reshape(tL,2*ones(kval+1,2));
tM = tt_eye(2,kval+1);
%% Final assembly
tMat = tkron(tA,tM)-h*tkron(tB,tL);
```

- Now that we have everything in this format, how can we solve our problem?

TT-GMRES An option is to rephrase our favorite Krylov method using the TT arithmetic, (Dolgov 2013) and adapt what we know to build a preconditioner (Bertaccini and Durastante 2019).

AMEn Use a *specialized solver* for linear systems in TT format (Dolgov and Savostyanov 2014).

✝ Concluding with an AMEn

Using AMEn (Dolgov and Savostyanov 2014) as

```
tx = amen_solve2(tMat,ttb,1e-6);
```

✝ Concluding with an AMEn

Using AMEn (Dolgov and Savostyanov 2014) as

```
tx = amen_solve2(tMat,ttb,1e-6);
```

k	m	n	IT	Residual	$\max(\text{tt-rank}(\mathcal{A}))$
2	64	128	9	2.231e-07	22
2	128	256	10	3.428e-07	26
2	256	512	14	5.925e-07	30
2	512	1024	22	3.957e-07	33
2	1024	2048	35	6.034e-07	37
2	2048	4096	47	6.968e-07	42

✝ Concluding with an AMEn

Using AMEn (Dolgov and Savostyanov 2014) as

```
tx = amen_solve2(tMat,ttb,1e-6);
```

k	m	n	IT	Residual	$\max(\text{tt-rank}(\mathcal{A}))$
3	64	128	8	2.252e-07	20
3	128	256	11	2.153e-07	24
3	256	512	15	2.138e-07	28
3	512	1024	18	2.950e-07	32
3	1024	2048	35	8.961e-07	36
3	2048	4096	50	3.821e-06	44

✝ Concluding with an AMEn

Using AMEn (Dolgov and Savostyanov 2014) as

```
tx = amen_solve2(tMat,ttb,1e-6);
```

k	m	n	IT	Residual	$\max(\text{tt-rank}(\mathcal{A}))$
3	64	128	8	2.252e-07	20
3	128	256	11	2.153e-07	24
3	256	512	15	2.138e-07	28
3	512	1024	18	2.950e-07	32
3	1024	2048	35	8.961e-07	36
3	2048	4096	50	3.821e-06	44

- Behavior is *similar* to the matrix-equation solver,

✝ Concluding with an AMEn

Using AMEn (Dolgov and Savostyanov 2014) as

```
tx = amen_solve2(tMat,ttb,1e-6);
```

k	m	n	IT	Residual	$\max(\text{tt-rank}(\mathcal{A}))$
3	64	128	8	2.252e-07	20
3	128	256	11	2.153e-07	24
3	256	512	15	2.138e-07	28
3	512	1024	18	2.950e-07	32
3	1024	2048	35	8.961e-07	36
3	2048	4096	50	3.821e-06	44

- Behavior is *similar* to the matrix-equation solver,
- We could play around with different **settings** and **options** of the AMEn solver.

Concluding with an AMEn

Using AMEn (Dolgov and Savostyanov 2014) as

```
tx = amen_solve2(tMat,ttb,1e-6);
```

k	m	n	IT	Residual	$\max(\text{tt-rank}(\mathcal{A}))$
3	64	128	8	2.252e-07	20
3	128	256	11	2.153e-07	24
3	256	512	15	2.138e-07	28
3	512	1024	18	2.950e-07	32
3	1024	2048	35	8.961e-07	36
3	2048	4096	50	3.821e-06	44

- Behavior is *similar* to the matrix-equation solver,
- We could play around with different **settings** and **options** of the AMEn solver.
- Studying the right combination of parameters, representation, setups is still an open problem for the BVM all-at-once approaches.

Conclusion and summary

- We have seen how to work with linear multistep methods in boundary value form,
- We have discussed some structured preconditioning strategy for the resulting linear systems,
- We have introduced the machinery for working with tensor equations in the Tensor Train format.
- There are *many* open problems and possibilities to do better here.

Next up

- Fractional Laplacians,
- Rational approximations and matrix functions,
- A couple of applications to complex network theory.

Bibliography I

Bertaccini, D. (2000). "A circulant preconditioner for the systems of LMF-based ODE codes". In: *SIAM J. Sci. Comput.* 22.3, pp. 767–786. ISSN: 1064-8275. DOI: 10.1137/S1064827599353476. URL: https://doi.org/10.1137/S1064827599353476.

— (2001). "Reliable preconditioned iterative linear solvers for some numerical integrators". In: *Numer. Linear Algebra Appl.* 8.2, pp. 111–125. ISSN: 1070-5325. DOI: 10.1002/1099-1506(200103)8:2<111::AID-NLA234>3.0.CO;2-Q. URL: https://doi.org/10.1002/1099-1506(200103)8:2%3C111::AID-NLA234%3E3.0.CO;2-Q.

Bertaccini, D. and F. Durastante (2019). "Block structured preconditioners in tensor form for the all-at-once solution of a finite volume fractional diffusion equation". In: *Appl. Math. Lett.* 95, pp. 92–97. ISSN: 0893-9659. DOI: 10.1016/j.aml.2019.03.028. URL: https://doi.org/10.1016/j.aml.2019.03.028.

Bertaccini, D. and F. Durastante (2018). "Limited memory block preconditioners for fast solution of fractional partial differential equations". In: *J. Sci. Comput.* 77.2, pp. 950–970. ISSN: 0885-7474. DOI: 10.1007/s10915-018-0729-3. URL: https://doi.org/10.1007/s10915-018-0729-3.

Bibliography II

- Bertaccini, D. and M. K. Ng (2001). "The convergence rate of block preconditioned systems arising from LMF-based ODE codes". In: *BIT* 41.3, pp. 433–450. ISSN: 0006-3835. DOI: 10.1023/A:1021906926616. URL: https://doi.org/10.1023/A:1021906926616.
- Brugnano, L. and D. Trigiante (1998). *Solving differential problems by multistep initial and boundary value methods*. Vol. 6. Stability and Control: Theory, Methods and Applications. Gordon and Breach Science Publishers, Amsterdam, pp. xvi+418. ISBN: 90-5699-107-8.
- Dolgov, S. V. (2013). "TT-GMRES: solution to a linear system in the structured tensor format". In: *Russian J. Numer. Anal. Math. Modelling* 28.2, pp. 149–172. ISSN: 0927-6467. DOI: 10.1515/rnam-2013-0009. URL: https://doi.org/10.1515/rnam-2013-0009.
- Dolgov, S. V. and D. V. Savostyanov (2014). "Alternating minimal energy methods for linear systems in higher dimensions". In: *SIAM J. Sci. Comput.* 36.5, A2248–A2271. ISSN: 1064-8275. DOI: 10.1137/140953289. URL: https://doi.org/10.1137/140953289.
- Gu, X.-M. et al. (2015). "Strang-type preconditioners for solving fractional diffusion equations by boundary value methods". In: *J. Comput. Appl. Math.* 277, pp. 73–86. ISSN: 0377-0427. DOI: 10.1016/j.cam.2014.08.011. URL: https://doi.org/10.1016/j.cam.2014.08.011.

Bibliography III

- Kazeev, V. A., B. N. Khoromskij, and E. E. Tyrtyshnikov (2013). "Multilevel Toeplitz matrices generated by tensor-structured vectors and convolution with logarithmic complexity". In: *SIAM J. Sci. Comput.* 35.3, A1511–A1536. ISSN: 1064-8275. DOI: 10.1137/110844830. URL: https://doi.org/10.1137/110844830.
- Kolda, T. G. and B. W. Bader (2009). "Tensor decompositions and applications". In: *SIAM Rev.* 51.3, pp. 455–500. ISSN: 0036-1445. DOI: 10.1137/07070111X. URL: https://doi.org/10.1137/07070111X.
- Oseledets, I. V. (2011). "Tensor-train decomposition". In: *SIAM J. Sci. Comput.* 33.5, pp. 2295–2317. ISSN: 1064-8275. DOI: 10.1137/090752286. URL: https://doi.org/10.1137/090752286.

An introduction to fractional calculus

Fundamental ideas and numerics

Fabio Durastante
Università di Pisa
fabio.durastante@unipi.it
fdurastante.github.io

November, 2022

Nonlocal operators (Andreu-Vaillo et al. 2010)

Let $\Omega \subset \mathbb{R}^n$ denote a **bounded** and **open** domain.
The action of a **nonlocal diffusion** operator $\mathcal{L}$ on $u(\mathbf{x}) : \Omega \to \mathbb{R}$ is defined as

$$\mathcal{L}u(\mathbf{x}) = 2\int_{\mathbb{R}^n} (u(\mathbf{y}) - u(\mathbf{x}))\gamma(\mathbf{x},\mathbf{y})\,\mathrm{d}\mathbf{y}, \qquad \forall\, \mathbf{x} \in \Omega \subseteq \mathbb{R}^n.$$

- the *volume* Ω is non-zero,
- the *kernel* $\gamma(\mathbf{x},\mathbf{y}) : \Omega \times \Omega \to \mathbb{R}$ is **nonnegative** and **symmetric**.

Nonlocal operators (Andreu-Vaillo et al. 2010)

Let $\Omega \subset \mathbb{R}^n$ denote a **bounded** and **open** domain.
The action of a **nonlocal diffusion** operator $\mathcal{L}$ on $u(\mathbf{x}) : \Omega \to \mathbb{R}$ is defined as

$$\mathcal{L}u(\mathbf{x}) = 2\int_{\mathbb{R}^n} (u(\mathbf{y}) - u(\mathbf{x}))\gamma(\mathbf{x},\mathbf{y})\,\mathrm{d}\mathbf{y}, \qquad \forall\, \mathbf{x} \in \Omega \subseteq \mathbb{R}^n.$$

- the *volume* Ω is non-zero,
- the *kernel* $\gamma(\mathbf{x},\mathbf{y}) : \Omega \times \Omega \to \mathbb{R}$ is **nonnegative** and **symmetric**.

The first **interesting equation** is the *nonlocal steady-state*

$$\begin{cases} -\mathcal{L}u = f, & \text{on } \Omega, \\ u = 0, & \text{on } \Omega_{\mathcal{I}}, \end{cases}$$

the **equality constraint** should be defined in general on an *interaction volume* $\Omega_{\mathcal{I}}$ that is **disjoint** from Ω; typically $\Omega_{\mathcal{I}} = \mathbb{R}^n \setminus \Omega \equiv \Omega^c$.

Fractional Laplacian

We are interested in a particular *nonlocal* operator $\mathcal{L}$ called the **Fractional Laplacian**.

Fractional Laplacian

We are interested in a particular *nonlocal* operator $\mathcal{L}$ called the **Fractional Laplacian**.

Fractional Laplacian

The fractional Laplacian is the pseudo-differential operator with Fourier symbol $\mathfrak{F}$ satisfying

$$(-\Delta)^{\alpha}\mathfrak{u}(\xi) = |\xi|^{2\alpha}\hat{u}(\xi), \quad 0 < \alpha \leq 1,$$

where $\hat{u}$ denotes the *Fourier transform* of u.

Fractional Laplacian

We are interested in a particular *nonlocal* operator $\mathcal{L}$ called the **Fractional Laplacian**.

Fractional Laplacian

The fractional Laplacian is the pseudo-differential operator with Fourier symbol $\mathfrak{F}$ satisfying

$$(-\Delta)^{\alpha}\mathfrak{u}(\xi) = |\xi|^{2\alpha}\hat{u}(\xi), \quad 0 < \alpha \leq 1,$$

where $\hat{u}$ denotes the *Fourier transform* of u.

Fractional Laplacian: integral formulation

An equivalent characterization of the fractional Laplacian is given by

$$(-\Delta)^{\alpha}u = c_{n,\alpha}\int_{\mathbb{R}^n}\frac{u(\mathbf{x})-u(\mathbf{y})}{|\mathbf{y}-\mathbf{x}|^{n+2\alpha}}\,\mathrm{d}\mathbf{y}, \qquad 0<\alpha<1, \;\; c_{n,\alpha} = \alpha 2^{2\alpha}\frac{\Gamma((n+2)/2)}{\Gamma(1/2)\Gamma(1-\alpha)}.$$

Fractional Laplacian

We are interested in a particular *nonlocal* operator $\mathcal{L}$ called the **Fractional Laplacian**.

Fractional Laplacian

The fractional Laplacian is the pseudo-differential operator with Fourier symbol $\mathfrak{F}$ satisfying

$$(-\Delta)^{\alpha}\mathfrak{u}(\xi) = |\xi|^{2\alpha}\hat{u}(\xi), \quad 0 < \alpha \leq 1,$$

where $\hat{u}$ denotes the *Fourier transform* of u.

Fractional Laplacian: integral formulation

An equivalent characterization of the fractional Laplacian is given by

$$-\mathcal{L} = (-\Delta)^{\alpha}, \quad 0 < \alpha < 1, \qquad \gamma(\mathbf{x}, \mathbf{y}) = \frac{c_{n,\alpha}}{2|\mathbf{y} - \mathbf{x}|^{n+2\alpha}} \; \forall \, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

Fractional Laplacian

We are interested in a particular *nonlocal* operator $\mathcal{L}$ called the **Fractional Laplacian**.

Fractional Laplacian

The fractional Laplacian is the pseudo-differential operator with Fourier symbol $\mathfrak{F}$ satisfying

$$(-\Delta)^{\alpha}\mathfrak{u}(\xi) = |\xi|^{2\alpha}\hat{u}(\xi), \quad 0 < \alpha \leq 1,$$

where $\hat{u}$ denotes the *Fourier transform* of u.

Fractional Laplacian: integral formulation

An equivalent characterization of the fractional Laplacian is given by

$$-\mathcal{L} = (-\Delta)^{\alpha}, \quad 0 < \alpha < 1, \qquad \gamma(\mathbf{x}, \mathbf{y}) = \frac{c_{n,\alpha}}{2|\mathbf{y} - \mathbf{x}|^{n+2\alpha}} \; \forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

We can play around with the definitions...

Fractional Laplacian (10 equivalent definitions)

We denote by $\mathbb{L}^p$ ($p \in [1, \infty)$) the Lebesgue spaces, $\mathcal{C}_0$ the space of continuous functions vanishing at infinity, and with $\mathcal{C}_{bu}$ the space of bounded uniformly continuous functions.

Theorem (Kwaśnicki 2017, Theorem 1.1)

Let $\mathcal{X}$ be any of the spaces $\mathbb{L}^p$, $p \in [1, \infty)$, $\mathcal{C}_0$ or $\mathcal{C}_{bu}$, and let $f \in \mathcal{X}$, $\beta = 2\alpha$. The following definitions of $\mathcal{L}f \in \mathcal{X}$ are equivalent:

(a) Fourier definition:

$$\mathcal{F}(\mathcal{L}f)(\xi) = -|\xi|^\beta \mathcal{F}f(\xi)$$

(if $\mathcal{X} = \mathbb{L}^p$, $p \in [1, 2]$);

(b) distributional definition:

$$\int_{\mathbb{R}^d} \mathcal{L}f(y)\varphi(y)dy = \int_{\mathbb{R}^d} f(x)L\varphi(x)dx$$

for all Schwartz functions φ, with $\mathcal{L}\varphi$ defined, for example, as in (a);

Fractional Laplacian (10 equivalent definitions)

We denote by $\mathbb{L}^p$ ($p \in [1, \infty)$) the Lebesgue spaces, $\mathcal{C}_0$ the space of continuous functions vanishing at infinity, and with $\mathcal{C}_{bu}$ the space of bounded uniformly continuous functions.

Theorem (Kwaśnicki 2017, Theorem 1.1)

Let $\mathcal{X}$ be any of the spaces $\mathbb{L}^p$, $p \in [1, \infty)$, $\mathcal{C}_0$ or $\mathcal{C}_{bu}$, and let $f \in \mathcal{X}$, $\beta = 2\alpha$. The following definitions of $\mathcal{L}f \in \mathcal{X}$ are equivalent:

(c) Bochner's[1] definition:

$$\mathcal{L}f = \frac{1}{|\Gamma(-\frac{\beta}{2})|} \int_0^\infty (e^{t\Delta} f - f) t^{-1-\beta/2} dt,$$

with the Bochner's integral of an $\mathcal{X}$-valued function;

[1] Bochner's integral extends the definition of Lebesgue integral to functions that take values in a Banach space, as the limit of integrals of simple functions.

Fractional Laplacian (10 equivalent definitions)

We denote by $\mathbb{L}^p$ ($p \in [1, \infty)$) the Lebesgue spaces, $\mathcal{C}_0$ the space of continuous functions vanishing at infinity, and with $\mathcal{C}_{bu}$ the space of bounded uniformly continuous functions.

Theorem (Kwaśnicki 2017, Theorem 1.1)

Let $\mathcal{X}$ be any of the spaces $\mathbb{L}^p$, $p \in [1, \infty)$, $\mathcal{C}_0$ or $\mathcal{C}_{bu}$, and let $f \in \mathcal{X}$, $\beta = 2\alpha$. The following definitions of $\mathcal{L}f \in \mathcal{X}$ are equivalent:

(d) Balakrishnan's definition:

$$\mathcal{L}f = \frac{\sin \frac{\beta\pi}{2}}{\pi} \int_0^\infty \Delta(sI - \Delta)^{-1} f \, s^{\beta/2-1} ds,$$

(e) singular integral definition:

$$\mathcal{L}f = \lim_{r \to 0^+} \frac{2^\beta \Gamma(\frac{d+\beta}{2})}{\pi^{d/2} |\Gamma(-\frac{\beta}{2})|} \int_{\mathbb{R}^d \setminus B(x,r)} \frac{f(\cdot + z) - f(\cdot)}{|z|^{d+\beta}} dz,$$

with the limit in $\mathcal{X}$;

Fractional Laplacian (10 equivalent definitions)

We denote by $\mathbb{L}^p$ ($p \in [1,\infty)$) the Lebesgue spaces, $\mathcal{C}_0$ the space of continuous functions vanishing at infinity, and with $\mathcal{C}_{bu}$ the space of bounded uniformly continuous functions.

Theorem (Kwaśnicki 2017, Theorem 1.1)

Let $\mathcal{X}$ be any of the spaces $\mathbb{L}^p$, $p \in [1,\infty)$, $\mathcal{C}_0$ or $\mathcal{C}_{bu}$, and let $f \in \mathcal{X}$, $\beta = 2\alpha$. The following definitions of $\mathcal{L}f \in \mathcal{X}$ are equivalent:

(f) Dynkin's definition:

$$\mathcal{L}f = \lim_{r\to 0^+} \frac{2^\beta \Gamma(\frac{d+\beta}{2})}{\pi^{d/2}|\Gamma(-\frac{\beta}{2})|} \int_{\mathbb{R}^d\setminus\overline{B}(x,r)} \frac{f(\cdot+z)-f(\cdot)}{|z|^d(|z|^2-r^2)^{\beta/2}}\,dz,$$

with the limit in $\mathcal{X}$;

Fractional Laplacian (10 equivalent definitions)

We denote by $\mathbb{L}^p$ ($p \in [1, \infty)$) the Lebesgue spaces, $\mathcal{C}_0$ the space of continuous functions vanishing at infinity, and with $\mathcal{C}_{bu}$ the space of bounded uniformly continuous functions.

Theorem (Kwaśnicki 2017, Theorem 1.1)

Let $\mathcal{X}$ be any of the spaces $\mathbb{L}^p$, $p \in [1, \infty)$, $\mathcal{C}_0$ or $\mathcal{C}_{bu}$, and let $f \in \mathcal{X}$, $\beta = 2\alpha$. The following definitions of $\mathcal{L}f \in \mathcal{X}$ are equivalent:

(g) quadratic form definition: $\langle \mathcal{L}f, \varphi \rangle = \mathcal{E}(f, \varphi)$ for all φ in the Sobolev space $H^{\beta/2}$, where

$$\mathcal{E}(f, g) = \frac{2^\beta \Gamma(\frac{d+\beta}{2})}{2\pi^{d/2} |\Gamma(-\frac{\beta}{2})|} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \frac{(f(y) - f(x))(\overline{g(y)} - \overline{g(x)})}{|x - y|^{d+\beta}} \, dx dy$$

(if $\mathcal{X} = \mathbb{L}^2$);

Fractional Laplacian (10 equivalent definitions)

We denote by $\mathbb{L}^p$ ($p \in [1, \infty)$) the Lebesgue spaces, $\mathcal{C}_0$ the space of continuous functions vanishing at infinity, and with $\mathcal{C}_{bu}$ the space of bounded uniformly continuous functions.

Theorem (Kwaśnicki 2017, Theorem 1.1)

Let $\mathcal{X}$ be any of the spaces $\mathbb{L}^p$, $p \in [1, \infty)$, $\mathcal{C}_0$ or $\mathcal{C}_{bu}$, and let $f \in \mathcal{X}$, $\beta = 2\alpha$. The following definitions of $\mathcal{L}f \in \mathcal{X}$ are equivalent:

(h) semigroup definition:

$$\mathcal{L}f = \lim_{t \to 0^+} \frac{P_t f - f}{t},$$

where $P_t f = f * p_t$ and $\mathcal{F}p_t(\xi) = e^{-t|\xi|^\beta}$;

(i) definition as the inverse of the Riesz potential:

$$\frac{\Gamma(\frac{d-\beta}{2})}{2^\beta \pi^{d/2} \Gamma(\frac{\beta}{2})} \int_{\mathbb{R}^d} \frac{\mathcal{L}f(\cdot + z)}{|z|^{d-\beta}} \, dz = -f(\cdot)$$

(if $\beta < d$ and $\mathcal{X} = \mathbb{L}^p$, $p \in [1, \frac{d}{\beta})$);

Fractional Laplacian (10 equivalent definitions)

We denote by $\mathbb{L}^p$ ($p \in [1, \infty)$) the Lebesgue spaces, $\mathcal{C}_0$ the space of continuous functions vanishing at infinity, and with $\mathcal{C}_{bu}$ the space of bounded uniformly continuous functions.

Theorem (Kwaśnicki 2017, Theorem 1.1)

Let $\mathcal{X}$ be any of the spaces $\mathbb{L}^p$, $p \in [1, \infty)$, $\mathcal{C}_0$ or $\mathcal{C}_{bu}$, and let $f \in \mathcal{X}$, $\beta = 2\alpha$. The following definitions of $\mathcal{L}f \in \mathcal{X}$ are equivalent:

(j) definition through harmonic extensions:

$$\begin{cases} \Delta_x u(x, y) + \beta^2 c_\beta^{2/\beta} y^{2-2/\beta} \partial_y^2 u(x, y) = 0 \quad \text{for } y > 0, \\ u(x, 0) = f(x), \\ \partial_y u(x, 0) = \mathcal{L}f(x), \end{cases}$$

where $c_\beta = 2^{-\beta} |\Gamma(-\frac{\beta}{2})| / \Gamma(\frac{\beta}{2})$ and where $u(\cdot, y)$ is a function of class $\mathcal{X}$ which depends continuously on $y \in [0, \infty)$ and $\|u(\cdot, y)\|_{\mathcal{X}}$ is bounded in $y \in [0, \infty)$.

Fractional Laplacian (10 equivalent definitions)

We denote by $\mathbb{L}^p$ ($p \in [1, \infty)$) the Lebesgue spaces, $\mathcal{C}_0$ the space of continuous functions vanishing at infinity, and with $\mathcal{C}_{bu}$ the space of bounded uniformly continuous functions.

Theorem (Kwaśnicki 2017, Theorem 1.1)

Let $\mathcal{X}$ be any of the spaces $\mathbb{L}^p$, $p \in [1, \infty)$, $\mathcal{C}_0$ or $\mathcal{C}_{bu}$, and let $f \in \mathcal{X}$, $\beta = 2\alpha$. The following definitions of $\mathcal{L}f \in \mathcal{X}$ are equivalent.
In addition, in (c), (e), (f), (h) and (j), convergence in the uniform norm can be relaxed to pointwise convergence to a function in $\mathcal{X}$ when $\mathcal{X} = \mathcal{C}_0$ or $\mathcal{X} = \mathcal{C}_{bu}$. Finally, for $\mathcal{X} = \mathbb{L}^p$ with $p \in [1, \infty)$, norm convergence in (e), (f), (h) or (j) implies pointwise convergence for almost all x.

- Convergence properties described here are for the *full-space definitions* of the fractional Laplace operator $\mathcal{L}$.
- We can invent **numerical methods** starting from each of these definitions.

Fractional Laplacian: equations on bounded domains

If Ω is **bounded** we can modify our first definition as follows.

- Take $u : \Omega \to \mathbb{R}$ and extend it to zero outside of Ω:

$$(-\Delta)^\alpha \tilde{u} = f \text{ in } \Omega, \qquad \tilde{u} = 0 \text{ in } \Omega^c = \mathbb{R}^n \setminus \Omega.$$

where

$$(-\Delta)^\alpha \tilde{u} = c_{n,\alpha} \int_{\mathbb{R}^n} \frac{\tilde{u}(\mathbf{x}) - \tilde{u}(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|^{n+2s}} \, \mathrm{d}\mathbf{y}$$

and thus $\tilde{u}$ is the extension by zero to $\mathbb{R}^n$ of a function $u : \Omega \to \mathbb{R}$ in $\mathbb{L}^2(\Omega)$.

Fractional Laplacian: equations on bounded domains

If Ω is **bounded** we can modify our first definition as follows.

- Take $u : \Omega \to \mathbb{R}$ and extend it to zero outside of Ω:

$$(-\Delta)^\alpha \tilde{u} = f \text{ in } \Omega, \qquad \tilde{u} = 0 \text{ in } \Omega^c = \mathbb{R}^n \setminus \Omega.$$

where

$$(-\Delta)^\alpha \tilde{u} = c_{n,\alpha} \int_{\mathbb{R}^n} \frac{\tilde{u}(\mathbf{x}) - \tilde{u}(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|^{n+2s}} \, \mathrm{d}\mathbf{y}$$

and thus $\tilde{u}$ is the extension by zero to $\mathbb{R}^n$ of a function $u : \Omega \to \mathbb{R}$ in $\mathbb{L}^2(\Omega)$.

Stochastic interpretation.

As we have seen when discussing the other derivatives, we can interpret also the Fractional Laplacian in a stochastic way. Indeed, one can prove that it is the infinitesimal generator of a 2α-stable Lévy process. The **boundary conditions** means that the particles are killed upon reaching Ω^c.

Fractional Laplacian: equations on bounded domains

The second definition relies instead on **spectral theory**.

- Recall that $-\Delta : \mathcal{D}(-\Delta) \subset \mathbb{L}^2(\Omega) \to \mathbb{L}^2(\Omega)$ is an unbounded, positive and closed operator with dense domain $\mathcal{D}(-\Delta) = \mathbb{H}_0^1(\Omega) \cap \mathbb{H}^2(\Omega)$ with a compact inverse.

Fractional Laplacian: equations on bounded domains

The second definition relies instead on **spectral theory**.

- Recall that $-\Delta : \mathcal{D}(-\Delta) \subset \mathbb{L}^2(\Omega) \to \mathbb{L}^2(\Omega)$ is an unbounded, positive and closed operator with dense domain $\mathcal{D}(-\Delta) = \mathbb{H}_0^1(\Omega) \cap \mathbb{H}^2(\Omega)$ with a compact inverse.
- There is a *countable* collection of eigenpairs $\{\lambda_k, \varphi_k\}_{k\in\mathbb{N}} \subset \mathbb{R}^+ \times \mathbb{H}_0^1(\Omega)$ such that $\{\varphi_k\}_{k\in\mathbb{N}}$ is an **orthonormal basis** of $\mathbb{L}^2(\Omega)$ (and of $\mathbb{H}_0^1(\Omega)$).

Fractional Laplacian: equations on bounded domains

The second definition relies instead on **spectral theory**.

- Recall that $-\Delta : \mathcal{D}(-\Delta) \subset \mathbb{L}^2(\Omega) \to \mathbb{L}^2(\Omega)$ is an unbounded, positive and closed operator with dense domain $\mathcal{D}(-\Delta) = \mathbb{H}_0^1(\Omega) \cap \mathbb{H}^2(\Omega)$ with a compact inverse.
- There is a *countable* collection of eigenpairs $\{\lambda_k, \varphi_k\}_{k\in\mathbb{N}} \subset \mathbb{R}^+ \times \mathbb{H}_0^1(\Omega)$ such that $\{\varphi_k\}_{k\in\mathbb{N}}$ is an **orthonormal basis** of $\mathbb{L}^2(\Omega)$ (and of $\mathbb{H}_0^1(\Omega)$).
- The **fractional power of the Dirichlet Laplacian** can thus be defined $\forall\, u \in \mathcal{C}_0^\infty$ as

$$(-\Delta)^\alpha u = \sum_{k=1}^{+\infty} \lambda_k^\alpha u_k \varphi_k, \qquad u_k = \langle w, \varphi_k \rangle_{\mathbb{L}^2(\Omega)} = \int_\Omega w \varphi_k \, \mathrm{d}x, \quad k \in \mathbb{N}$$

Fractional Laplacian: equations on bounded domains

The second definition relies instead on **spectral theory**.

- Recall that $-\Delta : \mathcal{D}(-\Delta) \subset \mathbb{L}^2(\Omega) \to \mathbb{L}^2(\Omega)$ is an unbounded, positive and closed operator with dense domain $\mathcal{D}(-\Delta) = \mathbb{H}^1_0(\Omega) \cap \mathbb{H}^2(\Omega)$ with a compact inverse.
- There is a *countable* collection of eigenpairs $\{\lambda_k, \varphi_k\}_{k\in\mathbb{N}} \subset \mathbb{R}^+ \times \mathbb{H}^1_0(\Omega)$ such that $\{\varphi_k\}_{k\in\mathbb{N}}$ is an **orthonormal basis** of $\mathbb{L}^2(\Omega)$ (and of $\mathbb{H}^1_0(\Omega)$).
- The **fractional power of the Dirichlet Laplacian** can thus be defined $\forall\, u \in \mathcal{C}^\infty_0$ as

$$(-\Delta)^\alpha u = \sum_{k=1}^{+\infty} \lambda_k^\alpha u_k \varphi_k, \qquad u_k = \langle w, \varphi_k \rangle_{\mathbb{L}^2(\Omega)} = \int_\Omega w \varphi_k \, \mathrm{d}x, \quad k \in \mathbb{N}$$

Extension

This definition of $(-\Delta)^\alpha$ can be extended by density to

$$\mathbb{H}^\alpha(\Omega) = \left\{ w = \sum_{k=1}^{+\infty} w_k \varphi_k \ : \ \sum_{k=1}^{+\infty} \lambda_k^s w_k^2 < +\infty \right\}.$$

☹ definitions on bounded domains aren't equivalent!

The **integral definition** of the Fractional Laplacian in

$$(-\Delta)^\alpha \tilde{u} = f \text{ in } \Omega, \qquad \tilde{u} = 0 \text{ in } \Omega^c = \mathbb{R}^n \setminus \Omega,$$

and the **spectral definition**

$$(-\Delta)^\alpha u = \sum_{k=1}^{+\infty} \lambda_k^\alpha u_k \varphi_k, \qquad u_k = \langle w, \varphi_k \rangle_{\mathbb{L}^2(\Omega)} = \int_\Omega w \varphi_k \, \mathrm{d}x, \quad k \in \mathbb{N},$$

are **NOT EQUIVALENT**!

Differences

Their difference is **positive** and **positivity preserving** (Musina and Nazarov 2014, Theorems 1 and 2). Furthermore, if we call $d(x, \partial\Omega)$ the distance for $x \in \Omega$ to the boundary $\partial\Omega$ we find

$$\text{(integral) } u(x) \approx d(x, \partial\Omega)^\alpha + v(x), \quad \text{(spectral) } u(x) \approx \begin{cases} d(x, \partial\Omega)^{2\alpha} + v(x), & \alpha \in (0, 1/2), \\ d(x, \partial\Omega) + v(x), & \alpha \in (1/2, 1), \end{cases}$$

for a smooth $v(x)$.

Equations of interest

Selecting the **right definition** for the problem the setting one has in mind (finite domain, infinite domain, ...) we can formulate several PDE with this new operator.

Diffusion-reaction $\partial_t u + (-\Delta)^\alpha u + c(t,x)u = 0$, Domain $(0,+\infty) \times \mathbb{R}^n$,

Quasi-geostrophic $\partial_t \theta + u \cdot \nabla\theta + \kappa(-\Delta)^\alpha \theta = f$, Domain $[0, T] \times \mathbb{R}^2$,

Cahn-Hilliard $\partial_t u + (-\Delta)^\alpha(-\varepsilon^2 \Delta u + f(u)) = 0$, Domain $(O, T] \times (0, 2\pi)^2$,

Porous medium $\partial_t u + (-\Delta)^\alpha(|u|^{m-1}\,\mathrm{sign}(u)) = 0$, Domain $(0,+\infty) \times \mathbb{R}^n$,

Schrödinger $i\hbar\partial_t \psi = D_\alpha(-\hbar^2\Delta)^\alpha \psi + V(r,t)\psi$, Domain $(r,t) \in \mathbb{R}^3 \times (0,+\infty)$,

Ultrasound $c_0^{-2}\partial_t^2 p = \nabla^2 p - \{\tau\partial_t(-\Delta)^\alpha + \eta(-\Delta)^{\alpha+1/2}\}p$, Domain $(-\infty,+\infty) \times \mathbb{R}^n$.

See **the review** (Lischke et al. 2020) for an updated *list of references*.

The Spectral Fractional Laplacian

Let us focus on problem using the **spectral Fractional Laplacian**

$$(-\Delta)^\alpha u = \sum_{k=1}^{+\infty} \lambda_k^\alpha u_k \varphi_k, \qquad u_k = \langle w, \varphi_k \rangle_{\mathbb{L}^2(\Omega)} = \int_\Omega w \varphi_k \, \mathrm{d}x, \quad k \in \mathbb{N}.$$

❓ How can we obtain **reliable numerical methods**?

The Spectral Fractional Laplacian

Let us focus on problem using the **spectral Fractional Laplacian**

$$(-\Delta)^\alpha u = \sum_{k=1}^{+\infty} \lambda_k^\alpha u_k \varphi_k, \qquad u_k = \langle w, \varphi_k \rangle_{\mathbb{L}^2(\Omega)} = \int_\Omega w \varphi_k \, \mathrm{d}x, \quad k \in \mathbb{N}.$$

❓ How can we obtain **reliable numerical methods**?

The Matrix-Transfer Technique

The idea from (Ilic et al. 2005, 2006) goes as follows, suppose that we have a *discretization scheme* for $-\Delta$ on Ω. That is, we can build $A_n = -\Delta_h \approx -\Delta$ on a discrete Ω_h ($h \to 0$ for $n \to +\infty$), then:

$$(-\Delta)^\alpha \approx (-\Delta_h)^\alpha = A_n^\alpha,$$

i.e., we have to compute a **matrix function** of (sparse) matrix discretizing the ordinary Laplacian on the domain of interest.

The Finite Difference Example

The simplest example we can think of is using **finite differences** on $\Omega = [0, 1]$ to solve for

$$\begin{cases} (-\Delta)^\alpha u = f(x), & x \in (0, 1), \\ u(0) = u(1) = 0. \end{cases}$$

This can be rewritten as

$$A_n = \frac{1}{h^2} T_{n-2}(2 - 2\cos(\theta)), \; h = \frac{1}{n-1},$$

on the grid $\{x_j = jh\}_{j=0}^n$, and solved on the inner nodes

$$\mathbf{u}_n(2 : n-1) = A_n^{-\alpha} \mathbf{f}(2 : n-1),$$

via *diagonalization*.

The Finite Difference Example

The simplest example we can think of is using **finite differences** on $\Omega = [0,1]$ to solve for

$$\begin{cases} (-\Delta)^\alpha u = f(x), & x \in (0,1), \\ u(0) = u(1) = 0. \end{cases}$$

This can be rewritten as

$$A_n = \frac{1}{h^2} T_{n-2}(2 - 2\cos(\theta)), \; h = \frac{1}{n-1},$$

on the grid $\{x_j = jh\}_{j=0}^n$, and solved on the inner nodes

$$\mathbf{u}_n(2:n-1) = A_n^{-\alpha}\mathbf{f}(2:n-1),$$

via *diagonalization*.

```
n = 100; h = 1/(n-1);
x = linspace(0,1,n).';
e  = ones(n-2,1);
An = spdiags([-e,2*e,-e]/h^2,-1:1,
↪  n-2,n-2);
f  = sin(pi*x);
u = [0;An\f(2:n-1);0];
[U,L,V] = eig(full(An));
ualpha = @(alpha)
↪  [0;V'\(L.^alpha\(U\f(2:n-1)));0];
```

The Finite Difference Example

The simplest example we can think of is using **finite differences** on $\Omega = [0, 1]$ to solve for

$$\begin{cases} (-\Delta)^\alpha u = f(x), & x \in (0, 1), \\ u(0) = u(1) = 0. \end{cases}$$

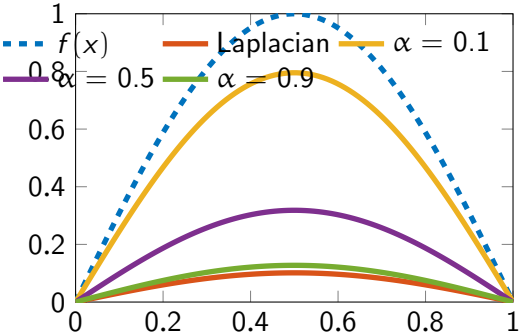
This can be rewritten as

$$A_n = \frac{1}{h^2} T_{n-2}(2 - 2\cos(\theta)), \; h = \frac{1}{n-1},$$

on the grid $\{x_j = jh\}_{j=0}^n$, and solved on the inner nodes

$$\mathbf{u}_n(2 : n-1) = A_n^{-\alpha} \mathbf{f}(2 : n-1),$$

via *diagonalization*.



The general case

- Somebody usually gets angry if we start diagonalizing stuff...

The general case

Somebody usually gets angry if we start diagonalizing stuff... so the right way to do is going for a *matrix function times vector* computation.

- We need to compute $g(z) = z^{-\alpha}$ for $\alpha \in (0, 1)$,
- on a matrix A_n that is either **symmetric and positive definite**, or of a matrix that is *similar* to an SPD matrix,
- A_n has also a condition number that grows (at least quadratically) with its size, i.e., is ill-conditioned.

The general case

Somebody usually gets angry if we start diagonalizing stuff... so the right way to do is going for a *matrix function times vector* computation.

- We need to compute $g(z) = z^{-\alpha}$ for $\alpha \in (0, 1)$,
- on a matrix A_n that is either **symmetric and positive definite**, or of a matrix that is *similar* to an SPD matrix,
- A_n has also a condition number that grows (at least quadratically) with its size, i.e., is ill-conditioned.

What method do we select?

The general case

Somebody usually gets angry if we start diagonalizing stuff... so the right way to do is going for a *matrix function times vector* computation.

- We need to compute $g(z) = z^{-\alpha}$ for $\alpha \in (0, 1)$,
- on a matrix A_n that is either **symmetric and positive definite**, or of a matrix that is *similar* to an SPD matrix,
- A_n has also a condition number that grows (at least quadratically) with its size, i.e., is ill-conditioned.

What method do we select?

- A_n is sparse and, if we deal with a regular uniform grid maybe also Toeplitz, a Lanczos **polynomial Krylov** with fast convergence would be perfect if it reaches convergence with a number of iteration independent of the size n.

The general case

Somebody usually gets angry if we start diagonalizing stuff... so the right way to do is going for a *matrix function times vector* computation.

- We need to compute $g(z) = z^{-\alpha}$ for $\alpha \in (0, 1)$,
- on a matrix A_n that is either **symmetric and positive definite**, or of a matrix that is *similar* to an SPD matrix,
- A_n has also a condition number that grows (at least quadratically) with its size, i.e., is ill-conditioned.

What method do we select?

- A_n is sparse and, if we deal with a regular uniform grid maybe also Toeplitz, a Lanczos **polynomial Krylov** with fast convergence would be perfect if it reaches convergence with a number of iteration independent of the size n.

Is this the case?

The Polynomial Krylov Method

If we use a polynomial Krylov subspace

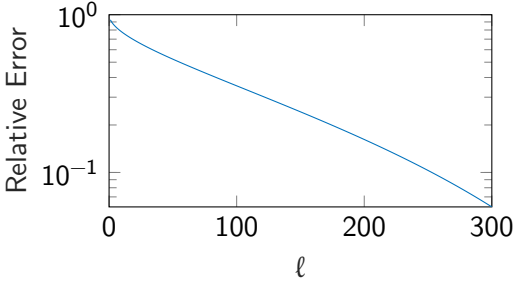
$$\mathcal{K}_\ell(A_n, v) = \mathrm{Span}\{\mathbf{v}, A_n\mathbf{v}, \ldots, A_n^{\ell-1}\mathbf{v}\}$$

to solve the problem, then the behavior is controlled by the approximation property

$$\|\mathbf{x} - \mathbf{x}_\ell\| \leq C \cdot \min_{p(z)\in\mathbb{P}_{\ell-1}} \max_{z\in\Lambda(A_n)} |p(z) - z^{-\alpha}|$$

for $\mathbb{P}_{\ell-1}$ the set of polynomial of degree $\leq \ell$, and C a constant *independent* of A and ℓ.

```
ytrue = mpower(full(An),-alpha)*b;
[Q,H] = arnoldi(An,b,l);
for j=1:l
 y = Q(:,1:j)*(mpower(H(1:j,1:j),
↪  -alpha)*(Q(:,1:j)'*b));
 err(j) = norm(y-ytrue)./norm(ytrue);
end
```



Rational Krylov Method

We need **better functions** for our approximation problem, i.e., *rational functions*!

Rational Krylov Method

We need **better functions** for our approximation problem, i.e., *rational functions*!

A general framework

Given a set of scalars $\{\sigma_1, \ldots, \sigma_{k-1}\} \subset \overline{\mathbb{C}}$ (the extended complex plane), that are not eigenvalues of A, let

$$q_{k-1}(z) = \prod_{j=1}^{k-1} (\sigma_j - z).$$

The rational Krylov subspace of order k associated with A, $\mathbf{v}$ and q_{k-1} is defined by

$$\mathcal{Q}_k(A, \mathbf{v}) = [q_{k-1}(A)]^{-1} \mathcal{K}_k(A, \mathbf{v}), \qquad \mathcal{K}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A\mathbf{v}, \ldots, A^{k-1}\mathbf{v}\}.$$

Rational Krylov Method

We need **better functions** for our approximation problem, i.e., *rational functions*!

A general framework

Given a set of scalars $\{\sigma_1, \ldots, \sigma_{k-1}\} \subset \overline{\mathbb{C}}$ (the extended complex plane), that are not eigenvalues of A, let

$$q_{k-1}(z) = \prod_{j=1}^{k-1} (\sigma_j - z).$$

The rational Krylov subspace of order k associated with A, $\mathbf{v}$ and q_{k-1} is defined by

$$\mathcal{Q}_k(A, \mathbf{v}) = [q_{k-1}(A)]^{-1} \mathcal{K}_k(A, \mathbf{v}), \qquad \mathcal{K}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A\mathbf{v}, \ldots, A^{k-1}\mathbf{v}\}.$$

A matrix expression

Given $\{\mu_1, \ldots, \mu_{k-1}\} \subset \overline{\mathbb{C}}$ such that $\sigma_j \neq \mu_j^{-2}$, we define the matrices

$$C_j = (\mu_j \sigma_j A - I)\,(\sigma_j I - A)^{-1}, \text{ and } \mathcal{Q}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, C_1\mathbf{v}, \ldots, C_{k-1} \cdots C_2 C_1 \mathbf{v}\}.$$

Rational Krylov Method

A matrix expression

Given $\{\mu_1, \ldots, \mu_{k-1}\} \subset \overline{\mathbb{C}}$ such that $\sigma_j \neq \mu_j^{-2}$, we define the matrices

$$C_j = (\mu_j \sigma_j A - I)\,(\sigma_j I - A)^{-1}, \text{ and } \mathcal{Q}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, C_1\mathbf{v}, \ldots, C_{k-1} \cdots C_2 C_1 \mathbf{v}\}.$$

Polynomial Krylov $\mathcal{W}_k(A, \mathbf{v}) = \mathcal{K}_k(A, \mathbf{v})$ set $\mu_j = 1$ and $\sigma_j = \infty$ for each j,

Rational Krylov Method

A matrix expression

Given $\{\mu_1, \ldots, \mu_{k-1}\} \subset \overline{\mathbb{C}}$ such that $\sigma_j \neq \mu_j^{-2}$, we define the matrices

$$C_j = (\mu_j \sigma_j A - I)\,(\sigma_j I - A)^{-1}, \text{ and } \mathcal{Q}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, C_1\mathbf{v}, \ldots, C_{k-1} \cdots C_2 C_1 \mathbf{v}\}.$$

Polynomial Krylov $\mathcal{W}_k(A, \mathbf{v}) = \mathcal{K}_k(A, \mathbf{v})$ set $\mu_j = 1$ and $\sigma_j = \infty$ for each j,

Extended Krylov $\mathcal{W}_{2k-1}(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A^{-1}\mathbf{v}, A\mathbf{v}, \ldots, A^{-(k-1)}\mathbf{v}, A^{k-1}\mathbf{v}\}$, set

$$(\mu_j, \sigma_j) = \begin{cases} (1, \infty), & \text{for } j \text{ even}, \\ (0, 0), & \text{for } j \text{ odd}. \end{cases}$$

Rational Krylov Method

A matrix expression

Given $\{\mu_1, \ldots, \mu_{k-1}\} \subset \overline{\mathbb{C}}$ such that $\sigma_j \neq \mu_j^{-2}$, we define the matrices

$$C_j = (\mu_j \sigma_j A - I)\,(\sigma_j I - A)^{-1}, \text{ and } \mathcal{Q}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, C_1\mathbf{v}, \ldots, C_{k-1} \cdots C_2 C_1 \mathbf{v}\}.$$

Polynomial Krylov $\mathcal{W}_k(A, \mathbf{v}) = \mathcal{K}_k(A, \mathbf{v})$ set $\mu_j = 1$ and $\sigma_j = \infty$ for each j,

Extended Krylov $\mathcal{W}_{2k-1}(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A^{-1}\mathbf{v}, A\mathbf{v}, \ldots, A^{-(k-1)}\mathbf{v}, A^{k-1}\mathbf{v}\}$, set

$$(\mu_j, \sigma_j) = \begin{cases} (1, \infty), & \text{for } j \text{ even}, \\ (0, 0), & \text{for } j \text{ odd}. \end{cases}$$

Shift-And-Invert $\mathcal{W}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, (\sigma I - A)^{-1}\mathbf{v}, \ldots, (\sigma I - A)^{-(k-1)}\mathbf{v}\}$, take $\mu_j = 0$ and $\sigma_j = \sigma$ for each j,

Rational Krylov Method

A matrix expression

Given $\{\mu_1, \ldots, \mu_{k-1}\} \subset \overline{\mathbb{C}}$ such that $\sigma_j \neq \mu_j^{-2}$, we define the matrices

$$C_j = (\mu_j \sigma_j A - I)\,(\sigma_j I - A)^{-1}, \text{ and } \mathcal{Q}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, C_1\mathbf{v}, \ldots, C_{k-1} \cdots C_2 C_1 \mathbf{v}\}.$$

Polynomial Krylov $\mathcal{W}_k(A, \mathbf{v}) = \mathcal{K}_k(A, \mathbf{v})$ set $\mu_j = 1$ and $\sigma_j = \infty$ for each j,

Extended Krylov $\mathcal{W}_{2k-1}(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A^{-1}\mathbf{v}, A\mathbf{v}, \ldots, A^{-(k-1)}\mathbf{v}, A^{k-1}\mathbf{v}\}$, set

$$(\mu_j, \sigma_j) = \begin{cases} (1, \infty), & \text{for } j \text{ even}, \\ (0, 0), & \text{for } j \text{ odd}. \end{cases}$$

Shift-And-Invert $\mathcal{W}_k(A, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, (\sigma I - A)^{-1}\mathbf{v}, \ldots, (\sigma I - A)^{-(k-1)}\mathbf{v}\}$, take $\mu_j = 0$ and $\sigma_j = \sigma$ for each j,

We are left our usual problem: **how do we select the poles?**

Pole Selection Strategies

Given a function $g(z)$ we find an **explicit (minimal) rational approximation**:

$$g(z) = \frac{P_\ell(z)}{Q_q(z)}, \; P_\ell \in \mathbb{P}_\ell[x], \; Q_q \in \mathbb{P}_q[x],$$

and use its poles for the RK-Method.

- Reasonably easy to get worst case scenario bounds;
- If we want an approximation of the same class with more poles we *usually* need to redo everything from scratch;
- There exist **brute force** algorithm to get such approximations.

Pole Selection Strategies

Given a function $g(z)$ we find an **explicit (minimal) rational approximation**:

$$g(z) = \frac{P_\ell(z)}{Q_q(z)}, \; P_\ell \in \mathbb{P}_\ell[x], \; Q_q \in \mathbb{P}_q[x],$$

and use its poles for the RK-Method.

- Reasonably easy to get worst case scenario bounds;
- If we want an approximation of the same class with more poles we *usually* need to redo everything from scratch;
- There exist **brute force** algorithm to get such approximations.

Obtain *optimal* rational approximation by solving **best approximation formulations**.

- It is only possible for certain combinations of complex plane regions and function classes;
- If one gets an explicit solution of the best approximation, there is usually a way to get nested poles;
- In some cases is possible **brute force** our way through it.

Pole Selection Strategies

Given a function $g(z)$ we find an **explicit (minimal) rational approximation**:

$$g(z) = \frac{P_\ell(z)}{Q_q(z)}, \; P_\ell \in \mathbb{P}_\ell[x], \; Q_q \in \mathbb{P}_q[x],$$

and use its poles for the RK-Method.

- Reasonably easy to get worst case scenario bounds;
- If we want an approximation of the same class with more poles we *usually* need to redo everything from scratch;
- There exist **brute force** algorithm to get such approximations.

Obtain *optimal* rational approximation by solving **best approximation formulations**.

- It is only possible for certain combinations of complex plane regions and function classes;
- If one gets an explicit solution of the best approximation, there is usually a way to get nested poles;
- In some cases is possible **brute force** our way through it.

Direct rational approximations

Sometimes it may be worth our while to use directly $g(A_n)\mathbf{v} = Q_q(A_n)^{-1}P_\ell(A_n)\mathbf{v}$.

Best Uniform Rational Approximation (BURA)

We try to find the poles by solving the $\min\text{-}\max$ problem

$$\max_{t\in[0,1]} |t^\alpha - r_{\alpha,k}(t)| = \min_{r_k(t)\in\mathbb{R}_{k,k}} \max_{t\in[0,1]} |t^\alpha - r_k(t)|, \qquad \alpha \in (0,1),$$

for $r_k(t)$ a (k,k)-rational function.

Best Uniform Rational Approximation (BURA)

We try to find the poles by solving the $\min\text{-}\max$ problem

$$\max_{t\in[0,1]} |t^\alpha - r_{\alpha,k}(t)| = \min_{r_k(t)\in\mathbb{R}_{k,k}} \max_{t\in[0,1]} |t^\alpha - r_k(t)|, \qquad \alpha \in (0,1),$$

for $r_k(t)$ a (k,k)-rational function.

Theorem (Stahl 2003, Theorem 1)

$$E_{\alpha,k} = \max_{t\in[0,1]} |t^\alpha - r_{\alpha,k}(t)| = 4^{\alpha+1}|\sin(\alpha\pi)|e^{-2\pi\sqrt{\alpha k}}.$$

Best Uniform Rational Approximation (BURA)

We try to find the poles by solving the $\min\text{-}\max$ problem

$$\max_{t\in[0,1]} |t^\alpha - r_{\alpha,k}(t)| = \min_{r_k(t)\in\mathbb{R}_{k,k}} \max_{t\in[0,1]} |t^\alpha - r_k(t)|, \qquad \alpha \in (0,1),$$

for $r_k(t)$ a (k,k)-rational function.

Theorem (Stahl 2003, Theorem 1)

$$E_{\alpha,k} = \max_{t\in[0,1]} |t^\alpha - r_{\alpha,k}(t)| = 4^{\alpha+1}|\sin(\alpha\pi)|e^{-2\pi\sqrt{\alpha k}}.$$

- The matrix is approximated as $A^{-\alpha} \approx \lambda_{1,h}^{-\alpha} r_{\alpha,k}(\lambda_{1,h}A^{-1})$.

Best Uniform Rational Approximation (BURA)

We try to find the poles by solving the $\min\text{-}\max$ problem

$$\max_{t\in[0,1]} |t^\alpha - r_{\alpha,k}(t)| = \min_{r_k(t)\in\mathbb{R}_{k,k}} \max_{t\in[0,1]} |t^\alpha - r_k(t)|, \qquad \alpha \in (0,1),$$

for $r_k(t)$ a (k,k)-rational function.

Theorem (Stahl 2003, Theorem 1)

$$E_{\alpha,k} = \max_{t\in[0,1]} |t^\alpha - r_{\alpha,k}(t)| = 4^{\alpha+1}|\sin(\alpha\pi)|e^{-2\pi\sqrt{\alpha k}}.$$

- The matrix is approximated as $A^{-\alpha} \approx \lambda_{1,h}^{-\alpha} r_{\alpha,k}(\lambda_{1,h}A^{-1})$.
- But how do we compute $r_{\alpha,k}(t)$ in practice?

Best Uniform Rational Approximation (BURA)

We try to find the poles by solving the $\min\text{-}\max$ problem

$$\max_{t\in[0,1]} |t^\alpha - r_{\alpha,k}(t)| = \min_{r_k(t)\in\mathbb{R}_{k,k}} \max_{t\in[0,1]} |t^\alpha - r_k(t)|, \qquad \alpha \in (0,1),$$

for $r_k(t)$ a (k,k)-rational function.

Theorem (Stahl 2003, Theorem 1)

$$E_{\alpha,k} = \max_{t\in[0,1]} |t^\alpha - r_{\alpha,k}(t)| = 4^{\alpha+1}|\sin(\alpha\pi)|e^{-2\pi\sqrt{\alpha k}}.$$

- The matrix is approximated as $A^{-\alpha} \approx \lambda_{1,h}^{-\alpha} r_{\alpha,k}(\lambda_{1,h}A^{-1})$.
- But how do we compute $r_{\alpha,k}(t)$ in practice?
- There is no *explicit solution*, thus we need to use a **numerical method**.

Best Uniform Rational Approximation (BURA)

The *workhorse* for computing BURA is the **Remez algorithm** (Braess 1986, § 6.B)

- Determine the points at which the error of the BURA equioscillates.
- Starting with a *suitable initial guess*, it iteratively determines a rational approximation passing through these points while shifting one or more toward a nearby local maximum.
- Implementation is delicate matter, observe we want both stability and possibly quadratic convergence.

Chose $P^{(0)}/Q^{(0)} \in \mathbb{R}_{m,n}$ and l points $\{x_i^1\}_{i=1}^l$;
$k \leftarrow 1$;
while *not satisfied* **do**

Determine $P^{(k)}/Q^{(k)} \in \mathbb{R}_{m,n}$ and $\eta_k \in \mathbb{R}$ such that for $i = 1, 2, \ldots, l$

$$f(x_i^k) - P^{(k)}(x_i^k)/Q^{(k)}(x_i^k) = (-1)^i \eta_k$$

Determine $x_1^{k+1} < x_2^{k+1} < \cdots < x_l^{k+1}$ such that for $i = 1, 2, \ldots, l$

$$s(-1)^i (f - P^{(k)}/Q^{(k)})(x_i^{k+1}) \geq |\eta_k|,$$

and that for one $i \in \{1, 2, \ldots, l\}$ the left-hand side equals $\|f - P^{(k)}/Q^{(k)}\|$, $s = \pm 1$;
$k \leftarrow k + 1$;

end

Best Uniform Rational Approximation (BURA)

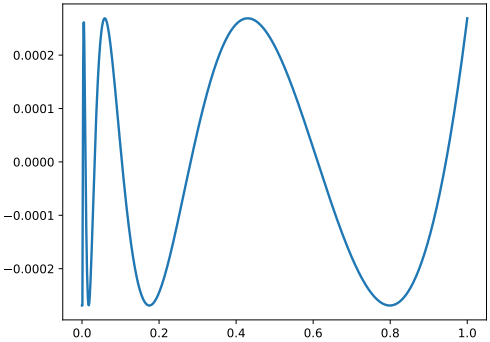
A **recent** and **available implementation** is given in the Python `baryrat` package, see (Hofreither 2021).

```
import numpy as np
import baryrat

alpha = 0.5
def f(x): return x**alpha
r = baryrat.brasil(f, [0,1], 5)
```

That gives us the `r.poles(0)`:

$$\sigma = \{-3.21294874e+00, -1.62633499e-01, -1.27958136e-02, -6.62129541e-04, -1.22326563e-05\}.$$



$k = 5,\ \alpha = 1/2$

Best Uniform Rational Approximation (BURA)

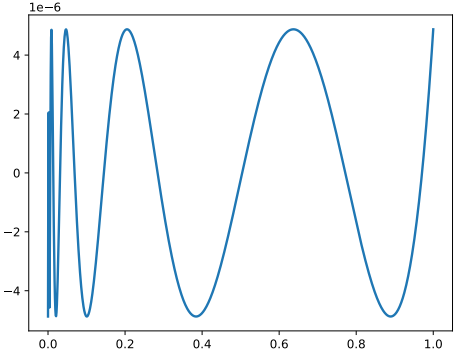
A **recent** and **available implementation** is given in the Python `baryrat` package, see (Hofreither 2021).

```
import numpy as np
import baryrat

alpha = 0.5
def f(x): return x**alpha
r = baryrat.brasil(f, [0,1], 5)
```

That gives us the `r.poles(0)`:

$$\sigma = \{-3.21294874e+00, -1.62633499e-01, -1.27958136e-02, -6.62129541e-04, -1.22326563e-05\}.$$



$k = 10,\ \alpha = 1/2$

Best Uniform Rational Approximation (BURA)

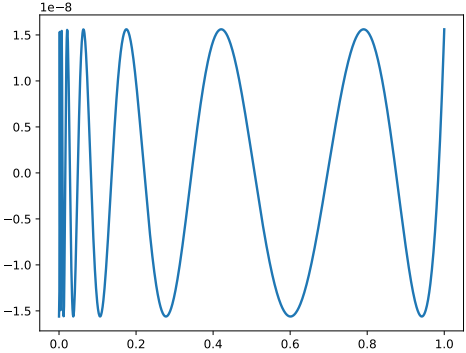
A **recent** and **available implementation** is given in the Python `baryrat` package, see (Hofreither 2021).

```
import numpy as np
import baryrat

alpha = 0.5
def f(x): return x**alpha
r = baryrat.brasil(f, [0,1], 5)
```

That gives us the `r.poles(0)`:

$$\sigma = \{-3.21294874e+00, -1.62633499e-01, -1.27958136e-02, -6.62129541e-04, -1.22326563e-05\}.$$



$k = 20,\ \alpha = 1/2$

Best Uniform Rational Approximation (BURA)

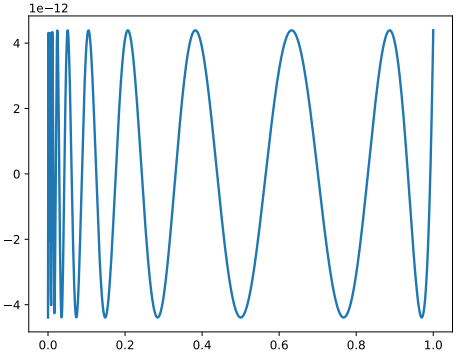
A **recent** and **available implementation** is given in the Python `baryrat` package, see (Hofreither 2021).

```
import numpy as np
import baryrat

alpha = 0.5
def f(x): return x**alpha
r = baryrat.brasil(f, [0,1], 5)
```

That gives us the `r.poles(0)`:

$$\sigma = \{-3.21294874e+00, -1.62633499e-01, -1.27958136e-02, -6.62129541e-04, -1.22326563e-05\}.$$



$k = 40,\ \alpha = 1/2$

Best Uniform Rational Approximation (BURA)

One can couple the error analysis with the one coming from the discretization of the Laplacian to get overall results (Harizanov et al. 2020).

Best Uniform Rational Approximation (BURA)

One can couple the error analysis with the one coming from the discretization of the Laplacian to get overall results (Harizanov et al. 2020).

Theorem (Harizanov et al. 2020, Theorem 4.2).

Let $\Omega \subset \mathbb{R}^2$ and suppose that the solution is in $\mathbb{H}^2(\Omega) \cup \mathbb{H}_0^1(\Omega)$ and satisfies $\|(-\Delta)^{-\alpha} f\|_{\mathbb{H}^2(\Omega)} \leq c\|f\|$. Then for $f \in \mathbb{H}^{1+\gamma}(\Omega)$, $\gamma > 0$, the solution $\mathbf{u}_h$ given by

$$\mathbf{u}_h = \lambda_{1,h}^{-\alpha}(\lambda_{1,h}A^{-1})^{\alpha} I_h f, \quad A = M_n^{-1}A_n, \quad I_h \text{ Interpolation},$$

satisfies

$$\|(-\Delta)^{-\alpha} f - \mathbf{u}_h\| \leq C(h^{2\alpha} + h^{1+\gamma})\|f\|_{\mathbb{H}^{1+\gamma}(\Omega)}.$$

Best Uniform Rational Approximation (BURA)

One can couple the error analysis with the one coming from the discretization of the Laplacian to get overall results (Harizanov et al. 2020).

Theorem (Harizanov et al. 2020, Theorem 4.2).

Let $\Omega \subset \mathbb{R}^2$ and suppose that the solution is in $\mathbb{H}^2(\Omega) \cup \mathbb{H}_0^1(\Omega)$ and satisfies $\|(-\Delta)^{-\alpha} f\|_{\mathbb{H}^2(\Omega)} \leq c\|f\|$. Then for $f \in \mathbb{H}^{1+\gamma}(\Omega)$, $\gamma > 0$, the solution $\mathbf{u}_h$ given by

$$\mathbf{u}_h = \lambda_{1,h}^{-\alpha}(\lambda_{1,h}A^{-1})^{\alpha} I_h f, \quad A = M_n^{-1}A_n, \quad I_h \text{ Interpolation},$$

satisfies

$$\|(-\Delta)^{-\alpha} f - \mathbf{u}_h\| \leq C(h^{2\alpha} + h^{1+\gamma})\|f\|_{\mathbb{H}^{1+\gamma}(\Omega)}.$$

Using **lumped FEM**, it is possible to have the error of the **fully discrete scheme** (Harizanov et al. 2020, Corollary 4.3), and then balance the discretization and the BURA error.

Best Uniform Rational Approximation (BURA)

One can couple the error analysis with the one coming from the discretization of the Laplacian to get overall results (Harizanov et al. 2020).

Theorem (Harizanov et al. 2020, Theorem 4.2).

Let $\Omega \subset \mathbb{R}^2$ and suppose that the solution is in $\mathbb{H}^2(\Omega) \cup \mathbb{H}_0^1(\Omega)$ and satisfies $\|(-\Delta)^{-\alpha} f\|_{\mathbb{H}^2(\Omega)} \leq c\|f\|$. Then for $f \in \mathbb{H}^{1+\gamma}(\Omega)$, $\gamma > 0$, the solution $\mathbf{u}_h$ given by

$$\mathbf{u}_h = \lambda_{1,h}^{-\alpha}(\lambda_{1,h} A^{-1})^{\alpha} I_h f, \quad A = M_n^{-1} A_n, \quad I_h \text{ Interpolation},$$

satisfies

$$\|(-\Delta)^{-\alpha} f - \mathbf{u}_h\| \leq C(h^{2\alpha} + h^{1+\gamma})\|f\|_{\mathbb{H}^{1+\gamma}(\Omega)}.$$

Using **lumped FEM**, it is possible to have the error of the **fully discrete scheme** (Harizanov et al. 2020, Corollary 4.3), and then balance the discretization and the BURA error.

The intend usage of these scheme is *outside* of a Krylov method.

Quadrature-based approaches

Another viable approach is to use a *rational approximation* based on a **quadrature formula**.

- There is more than a *connection* between **quadrature formulas** and **rational approximations**.
- Padé approximants can be viewed as formal Gaussian quadrature methods (Brezinski 1980, Page 34).
- This connection was already know to Gauß

  C. F. Gauss, Methodus nova integralium valores per approximationem inveniendi, Comment. Soc. Reg. Scient. Gotting. Recent., 1814

C.F.Gauß
(1777-1855)

Quadrature-based approaches

Another viable approach is to use a *rational approximation* based on a **quadrature formula**.

- There is more than a *connection* between **quadrature formulas** and **rational approximations**.
- Padé approximants can be viewed as formal Gaussian quadrature methods (Brezinski 1980, Page 34).
- This connection was already know to Gauß

  C. F. Gauss, Methodus nova integralium valores per approximationem inveniendi, Comment. Soc. Reg. Scient. Gotting. Recent., 1814

C.F.Gauß
(1777-1855)

The idea is always the same **1.** Find an integral representation of the function of interest. **2.** Find a change of variables that makes a Gauss-type weight appears. **3.** Rational approximation is obtained by the Gauss quadrature formula. **4.** The error analysis relies on the analysis for the formula.

The Gauss-Jacobi approach

This is an idea from (Aceto, Bertaccini, et al. 2019; Aceto and Novati 2018).

The Gauss-Jacobi approach

This is an idea from (Aceto, Bertaccini, et al. 2019; Aceto and Novati 2018).
We do **step 1** by looking through a book:

Proposition (Bhatia 1997, example V.1.10, 21, section 5.5.5)

Let $A \in \mathbb{R}^{n \times n}$ be such that $\Lambda(A) \subset \mathbb{C} \setminus (-\infty, 0]$. For $\alpha \in (0, 1)$ the following representation holds

$$A^{\alpha} = \frac{\sin(\alpha\pi)}{\alpha\pi} A \int_0^{\infty} \left(\rho^{1/\alpha} I + A\right)^{-1} \mathrm{d}\rho.$$

The Gauss-Jacobi approach

This is an idea from (Aceto, Bertaccini, et al. 2019; Aceto and Novati 2018).
We do **step 1** by looking through a book:

Proposition (Bhatia 1997, example V.1.10, 21, section 5.5.5)

Let $A \in \mathbb{R}^{n\times n}$ be such that $\Lambda(A) \subset \mathbb{C} \setminus (-\infty, 0]$. For $\alpha \in (0,1)$ the following representation holds

$$A^\alpha = \frac{\sin(\alpha\pi)}{\alpha\pi} A \int_0^\infty \left(\rho^{1/\alpha} I + A\right)^{-1} \mathrm{d}\rho.$$

Now do **step 2**, i.e., a *change of variables*:

$$\rho^{1/\alpha} = \tau \frac{1-t}{1+t}, \qquad \tau > 0.$$

The Gauss-Jacobi approach

By plugging the change of variables in the integral, we find

$$A^\alpha = \frac{2\sin(\alpha\pi)\tau^\alpha}{\pi} A \int_{-1}^{1} (1-t)^{\alpha-1}(1+t)^{-\alpha} \left(\tau(1-t)I + (1+t)A\right)^{-1} \, \mathrm{d}t.$$

The Gauss-Jacobi approach

By plugging the change of variables in the integral, we find

$$A^\alpha = \frac{2\sin(\alpha\pi)\tau^\alpha}{\pi} A \int_{-1}^{1} (1-t)^{\alpha-1}(1+t)^{-\alpha} \left(\tau(1-t)I + (1+t)A\right)^{-1} \, \mathrm{d}t.$$

We made the **weights of the Gauss-Jacobi quadrature** appear, thus

$$\left(\frac{1}{\tau}A\right)^{\aleph} \approx \frac{1}{\tau}A \sum_{j=1}^{k} \frac{2\sin(\alpha\pi)}{\pi} \frac{\omega_j}{1+\theta_j} \left(\frac{1-\theta_j}{1+\theta_j} + \frac{1}{\tau}A\right)^{-1},$$

- ω_j and θ_j are, respectively, the weights and nodes of the Gauss–Jacobi quadrature formula with weight function $(1-t)^{\alpha-1}(1+t)^{-\alpha}$,
- we should use *error analysis* to fix the τ parameter.
- From (Frommer, Güttel, and Schweitzer 2014, Lemma 4.4) we know that the k-point Gauss-Jacobi quadrature corresponds to the $(k-1, k)$-Padé approximant of $(z/\tau)^{\alpha-1}$ centered at 1.

The Gauss-Jacobi approach

As we have seen from the BURA example, we may be interested in $g(z) = z^{-\alpha}$, $\alpha \in (0,1)$, but it is easy to rewrite the approximation as

$$z^{-\alpha/2} \approx \sum_{j=1}^{k} \frac{2\sin(\alpha\pi)\tau^{1-\alpha/2}}{\pi} \frac{\omega_j}{1+\theta_j} \left(\frac{\tau(1-\theta_j)}{1+\theta_j} + z \right)^{-1} \triangleq R_{k-1,k}(z), \quad \tau > 0$$

- ω_j and θ_j are the weights and nodes of the Gauss–Jacobi quadrature formula with weight $(1-x)^{-\alpha}(1+x)^{\alpha-1}$.
- If we rearrange the expression we then find

$$R_{k-1,k}(z) = \frac{p_{k-1}(z)}{q_k(z)} = \frac{\chi \prod_{r=1}^{k-1}(z+\epsilon_r)}{\prod_{j=1}^{k}(z+\eta_j)}, \quad \chi = \frac{\eta_k}{\tau^\alpha} \frac{\binom{k+\alpha/2-1}{k-1}}{\binom{k-\alpha}{k}} \prod_{j=1}^{k-1} \frac{\eta_j}{\epsilon_j}.$$

for

$$\epsilon_r = \tau \frac{1-\zeta_r}{1+\zeta_r}, \quad r = 1,2,\ldots,k-1, \qquad \eta_j = \frac{\tau(1-\theta_j)}{1+\theta_j}, \quad j = 1,2,\ldots,k.$$

The Gauss-Jacobi approach

To fix the $\tau > 0$ parameter we need the error analysis from (Aceto and Novati 2019) to bound the *truncation error*:

$$E_{k-1,k}(\lambda/\tau) \triangleq (\lambda/\tau)^{-\alpha} - R_{k-1,k}(\lambda/\tau).$$

- When working with these expression, usually one can manipulate and express them in terms of *Gauss-Hypergeometric functions*, then use their asymptotic to produce the bound, *e.g.*, in this case

$$z = 1 - \frac{\lambda}{t}, \quad (1-z)^{-\alpha} = {}_2F_1\left(\begin{matrix} 1, \alpha \\ 1 \end{matrix} ; z \right), \quad |\arg(1-z)| < \pi.$$

The Gauss-Jacobi approach

To fix the $\tau > 0$ parameter we need the error analysis from (Aceto and Novati 2019) to bound the *truncation error*:

$$E_{k-1,k}(\lambda/\tau) \triangleq (\lambda/\tau)^{-\alpha} - R_{k-1,k}(\lambda/\tau).$$

- When working with these expression, usually one can manipulate and express them in terms of *Gauss-Hypergeometric functions*, then use their asymptotic to produce the bound, *e.g.*, in this case

$$z = 1 - \frac{\lambda}{t}, \quad (1-z)^{-\alpha} = {}_2F_1\left(\begin{matrix} 1, \alpha \\ 1 \end{matrix} ; z \right), \quad |\arg(1-z)| < \pi.$$

Proposition (Aceto and Novati 2019, Proposition 2)

For large values of k, the following representation for the truncation error holds

$$E_{k-1,k}(\lambda/\tau) = 2\sin(\alpha\pi)(\lambda/\tau)^{-\alpha} \left[\frac{\sqrt{\lambda} - \sqrt{\tau}}{\sqrt{\lambda} + \sqrt{\tau}} \right]^{2k} (1 + O(1/k)).$$

The Gauss-Jacobi approach

Theorem (Aceto and Novati 2019, Theorem 2)

If $\mathcal{L}$ is a self-adjoint positive operator on a separable Hilbert space $\mathbb{H}$ with spectrum $\Lambda(\mathcal{L}) \subset [c, +\infty)$, $c > 0$ having a compact inverse, then

$$\left\| \mathcal{L}^{-\alpha} - \tau_k^{-\alpha} R_{k-1,k}\left(\frac{1}{\tau_k}\mathcal{L}\right) \right\|_{\mathbb{H}\to\mathbb{H}} \leq 2\sin(\alpha\pi) c^{-\alpha} \left(\frac{2k\sqrt{e}}{\alpha}\right)^{-4\alpha} \left[2\ln\left(\frac{2k}{\alpha}\right) + 1\right]^{2\alpha} \left(1 + O(k^{-2})\right),$$

for

$$\tau_k = c\left(\frac{\alpha}{2ke}\right)^2 \exp\left(2W\left(\frac{4k^2 e}{\alpha^2}\right)\right),$$

where W denotes the Lambert W-function.

It becomes increasingly difficult if the spectrum is close to the branch point of $z^{-\alpha}$.

The Gauss-Jacobi approach (bounded operators)

If $\mathcal{L}_N$ is a **bounded operator**, i.e., $\Lambda(\mathcal{L}_N) \in [c, \lambda_N]$ then the $\min\text{-}\max$ problem for $|E_{k-1,k}(\lambda/\tau)|$ have two different solutions for *small* and *large* values of k.
We call $\overline{\lambda} = \frac{\tau}{\alpha^2}(k + \sqrt{k^2+1})^2$

$\overline{\lambda} < \lambda_N$ (k smalll) The previous estimate is still good, i.e.,

$$\tau_k = c\left(\frac{\alpha}{2ke}\right)^2 \exp\left(2W\left(\frac{4k^2e}{\alpha^2}\right)\right),$$

$\overline{\lambda} > \lambda_N$ (k large) then

$$\hat{\tau}_k = \left(-\frac{\alpha\sqrt{(}\lambda_N)}{8k}\ln\left(\frac{\lambda_N}{c}\right) + \sqrt{\left(\frac{\alpha\sqrt{\lambda_N}}{8k}\ln\left(\frac{\lambda_N}{c}\right)\right)^2 + \sqrt{c\lambda_N}}\right)^2.$$

The Gauss-Jacobi approach (bounded operators)

Theorem (Aceto and Novati 2019, Theorem 3)

Let $\overline{k}$ be such that for each $k \geq \overline{k}$ we have $\overline{\lambda} = \overline{\lambda}(k) > \lambda_N$. Then for each $k \geq \overline{k}$, taking $\tau = \hat{\tau}_k$, the following bound holds

$$\left\| \mathcal{L}_N^{-\alpha} - \hat{\tau}_k^{-\alpha} R_{k-1,k}\left(\frac{1}{\hat{\tau}_k}\mathcal{L}_N\right) \right\|_2 \leq 2\sin(\alpha\pi)(c\lambda_N)^{-\alpha/2}\exp\left(-4k\left(\frac{c}{\lambda_N}\right)^{1/4}\right)(1+O(k^{-1})).$$

- The bound gets worse when we refine the discretization of the differential operator!
- The choice of τ is better than the asymptotically selected value $\tau_\infty = \sqrt{c\Lambda_N}$.

The Gauss-Jacobi approach (bounded operators)

Theorem (Aceto and Novati 2019, Theorem 3)

Let $\overline{k}$ be such that for each $k \geq \overline{k}$ we have $\overline{\lambda} = \overline{\lambda}(k) > \lambda_N$. Then for each $k \geq \overline{k}$, taking $\tau = \hat{\tau}_k$, the following bound holds

$$\left\| \mathcal{L}_N^{-\alpha} - \hat{\tau}_k^{-\alpha} R_{k-1,k} \left(\frac{1}{\hat{\tau}_k} \mathcal{L}_N \right) \right\|_2 \leq 2 \sin(\alpha\pi)(c\lambda_N)^{-\alpha/2} \exp\left(-4k \left(\frac{c}{\lambda_N} \right)^{1/4} \right) (1 + O(k^{-1})).$$

- The bound gets worse when we refine the discretization of the differential operator!
- The choice of τ is better than the asymptotically selected value $\tau_\infty = \sqrt{c\Lambda_N}$.

The choice is made as

$$\tau_{k,N} = \begin{cases} \tau_k, & k < \overline{k}, \\ \hat{\tau}_k, & k \geq \overline{k}, \end{cases} \qquad \text{for } \overline{k} = \left\lceil \frac{\alpha}{2\sqrt{2}} \sqrt{\ln\left(\frac{\lambda_N}{c} e^2 \right)} \left(\frac{\lambda_N}{c} \right)^{\frac{1}{4}} \right\rceil.$$

A Gauss-Laguerre approach (Aceto and Novati 2022)

We start again from an **integral representation** (Bonito and Pasciak 2015)

$$\mathcal{L}^{-\alpha} = \frac{2\sin(\alpha\pi)}{\pi}\int_0^{+\infty} t^{2\alpha-1}(\mathcal{I}+t^2\mathcal{L})^{-1}\mathrm{d}t, \qquad \alpha\in(0,1).$$

A Gauss-Laguerre approach (Aceto and Novati 2022)

We start again from an **integral representation** (Bonito and Pasciak 2015)

$$\mathcal{L}^{-\alpha} = \frac{2\sin(\alpha\pi)}{\pi}\int_0^{+\infty} t^{2\alpha-1}(\mathcal{I}+t^2\mathcal{L})^{-1}\mathrm{d}t, \qquad \alpha\in(0,1).$$

Then, we go for the **change of variables** $y=\ln t$ we obtain

$$\begin{aligned}\mathcal{L}^{-\alpha} =& \frac{2\sin(\alpha\pi)}{\pi}\int_{-\infty}^{+\infty} e^{2\alpha y}(\mathcal{I}+e^{2y}\mathcal{L})^{-1}\mathrm{d}y, \qquad \alpha\in(0,1).\\ =& \int_{-\infty}^{0} e^{2\alpha y}(\mathcal{I}+e^{2y}\mathcal{L})^{-1}\mathrm{d}y + \int_0^{+\infty} e^{2\alpha y}(\mathcal{I}+e^{2y}\mathcal{L})^{-1}\mathrm{d}y\end{aligned}$$

$$\begin{aligned}2\alpha y &= -x\\ 2(1-\alpha)y &= x\end{aligned} \quad \rightarrow = \frac{1}{2\alpha}\int_0^{+\infty} e^{-x}(\mathcal{I}+e^{-x/\alpha}\mathcal{L})^{-1}\mathrm{d}x + \frac{1}{2(1-\alpha)}\int_0^{+\infty} e^{-x}(e^{-x/(1-\alpha)}\mathcal{I}+\mathcal{L})^{-1}\mathrm{d}x.$$

A Gauss-Laguerre approach (Aceto and Novati 2022)

We start again from an **integral representation** (Bonito and Pasciak 2015)

$$\mathcal{L}^{-\alpha} = \frac{2\sin(\alpha\pi)}{\pi}\int_0^{+\infty} t^{2\alpha-1}(\mathcal{I}+t^2\mathcal{L})^{-1}\mathrm{d}t, \qquad \alpha\in(0,1).$$

Then, we go for the **change of variables** $y = \ln t$ we obtain

$$\mathcal{L}^{-\alpha} = \frac{\sin(\alpha\pi)}{\alpha\pi} I^{(1)}(\mathcal{L}) + \frac{\sin(\alpha\pi)}{(1-\alpha)\pi} I^{(2)}(\mathcal{L}),$$

for

$$I^{(1)}(\lambda) = \int_0^{+\infty} e^{-x}(1+e^{-x/\alpha}\lambda)^{-1}\mathrm{d}x, \qquad I^{(2)}(\lambda) = \int_0^{+\infty} e^{-x}(e^{-x/(1-\alpha)}+\lambda)^{-1}\mathrm{d}x.$$

A Gauss-Laguerre approach (Aceto and Novati 2022)

We start again from an **integral representation** (Bonito and Pasciak 2015)

$$\mathcal{L}^{-\alpha} = \frac{2\sin(\alpha\pi)}{\pi}\int_0^{+\infty} t^{2\alpha-1}(\mathcal{I}+t^2\mathcal{L})^{-1}\mathrm{d}t, \qquad \alpha\in(0,1).$$

Then, we go for the **change of variables** $y = \ln t$ we obtain

$$\mathcal{L}^{-\alpha} =$$

for

$$I^{(1)}(\lambda) = \int_0^{+\infty} e^{-x}(1+e^{-x/\alpha}\lambda)^{-1}\mathrm{d}x, \qquad I^{(2)}(\lambda) = \int_0^{+\infty} e^{-x}(e^{-x/(1-\alpha)}+\lambda)^{-1}\mathrm{d}x.$$

The weight $\omega(x) = e^{-x}$, is the weight of **Gauss-Laguerre** formulas.

A Gauss-Laguerre approach (Aceto and Novati 2022)

If we call the weights $w_j^{(n)}$ and nodes $\vartheta_j^{(n)}$ (in ascending order) of the Gauss-Laguerre formula, then we obtain the following $(2n-1, 2n)$ rational approximation:

$$\mathcal{L}^{-\alpha} \approx \frac{\sin(\alpha\pi)}{\alpha\pi} R_{n-1,n}^{(1)}(\mathcal{L}) + \frac{\sin(\alpha\pi)}{(1-\alpha)\pi} R_{n-1,n}^{(2)}(\mathcal{L}) \triangleq R_{2n-1,2n}(\mathcal{L}),$$

where

$$\begin{aligned} R_{n-1,n}^{(1)}(\lambda) &= \sum_{j=1}^{n} w_j^{(n)} \left(1 + e^{-\vartheta_j^{(n)}/\alpha}\lambda\right)^{-1}, \\ R_{n-1,n}^{(2)}(\lambda) &= \sum_{j=1}^{n} w_j^{(n)} \left(e^{-\vartheta_j^{(n)}/(1-\alpha)} + \lambda\right)^{-1}. \end{aligned}$$

A Gauss-Laguerre approach (Aceto and Novati 2022)

If we call the weights $w_j^{(n)}$ and nodes $\vartheta_j^{(n)}$ (in ascending order) of the Gauss-Laguerre formula, then we obtain the following $(2n-1, 2n)$ rational approximation:

$$\mathcal{L}^{-\alpha} \approx \frac{\sin(\alpha\pi)}{\alpha\pi} R_{n-1,n}^{(1)}(\mathcal{L}) + \frac{\sin(\alpha\pi)}{(1-\alpha)\pi} R_{n-1,n}^{(2)}(\mathcal{L}) \triangleq R_{2n-1,2n}(\mathcal{L}),$$

where

$$\begin{aligned} R_{n-1,n}^{(1)}(\lambda) &= \sum_{j=1}^{n} w_j^{(n)} \left(1 + e^{-\vartheta_j^{(n)}/\alpha}\lambda\right)^{-1}, \\ R_{n-1,n}^{(2)}(\lambda) &= \sum_{j=1}^{n} w_j^{(n)} \left(e^{-\vartheta_j^{(n)}/(1-\alpha)} + \lambda\right)^{-1}. \end{aligned}$$

Third step is using **error estimate for Gauss-Laguerre formulas** to get the bound.

A Gauss-Laguerre approach (Aceto and Novati 2022)

The analysis treats separately the two integrals and requires expressing the error as a *contour integral*:

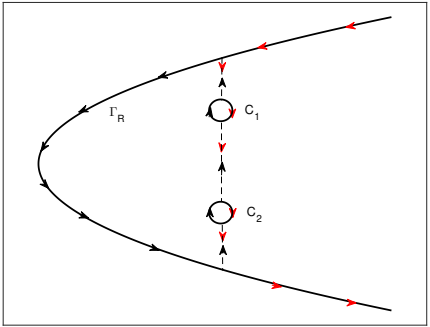
$$E_n(f) = \frac{1}{2\pi i} \int_\Gamma \frac{q_n(z)}{L_n(z)} f(z) \mathrm{d}z,$$

here $L_n(z)$ is the Laguerre polynomial, $q_n(z)$ is the so-called associated function defined by

$$q_n(z) = \int_0^{+\infty} \frac{e^{-x} L_n(x)}{z - x} \mathrm{d}x, \quad z \notin [0, +\infty),$$

and Γ is a contour containing $[0, +\infty)$ with the additional property that **no singularity** of $f(z)$ **lies on or within this contour**; see (Davis and Rabinowitz 1984, §4.6).

A Gauss-Laguerre approach (Aceto and Novati 2022)



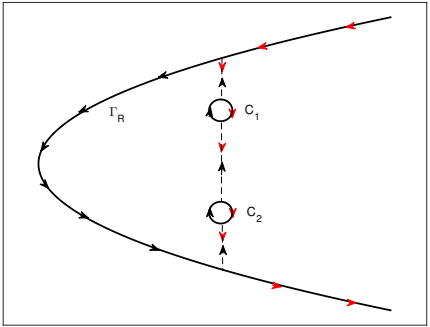
Denote with C_1 and C_2 two arbitrary small circles surrounding the two poles and define

$$\Gamma = \Gamma_R \cup C_1 \cup C_2.$$

The error can be written as

$$E_n(f) = \frac{1}{2\pi i}\left\{\int_{\Gamma_R} - \int_{C_1} - \int_{C_2}\right\}\frac{q_n(z)}{L_n(z)} f(z)\,dz.$$

A Gauss-Laguerre approach (Aceto and Novati 2022)



Denote with C_1 and C_2 two arbitrary small circles surrounding the two poles and define $\Gamma = \Gamma_R \cup C_1 \cup C_2$.

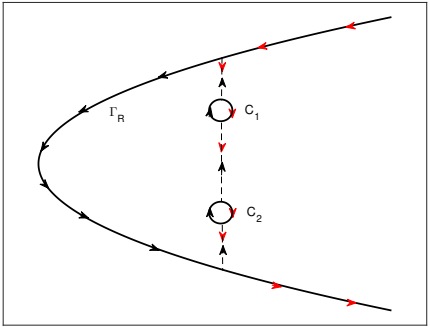
The error can be written as

$$E_n(f) = \frac{1}{2\pi i}\left\{\int_{\Gamma_R} - \int_{C_1} - \int_{C_2}\right\}\frac{q_n(z)}{L_n(z)} f(z)dz.$$

Then using:

$$\frac{q_n(z)}{L_n(z)} = 2\pi e^{-z}\left[\exp\left(\sqrt{-z}\right)\right]^{-2\sqrt{n}} \times$$
$$\times\left(1 + O\left(\frac{1}{n}\right)\right), \quad z \notin [0, +\infty),$$

A Gauss-Laguerre approach (Aceto and Novati 2022)



Denote with C_1 and C_2 two arbitrary small circles surrounding the two poles and define $\Gamma = \Gamma_R \cup C_1 \cup C_2$.

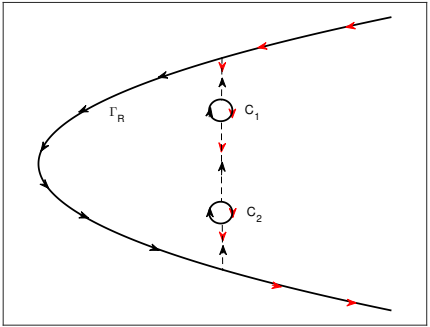
The error can be written as

$$E_n(f) = \frac{1}{2\pi i}\left\{\int_{\Gamma_R} - \int_{C_1} - \int_{C_2}\right\}\frac{q_n(z)}{L_n(z)} f(z)\,dz.$$

One arrives at

$$\begin{aligned}|E_n(f)| \leq & 4\pi \left|\mathrm{Res}\left(f(z), z_0\right) e^{-z_0}\right| \times \\ & \times \left[\exp\left(\mathrm{Re}\left(\sqrt{-z_0}\right)\right)\right]^{-2\sqrt{n}} \times \\ & \times \left(1 + O\left(\frac{1}{n}\right)\right).\end{aligned}$$

A Gauss-Laguerre approach (Aceto and Novati 2022)



Denote with C_1 and C_2 two arbitrary small circles surrounding the two poles and define $\Gamma = \Gamma_R \cup C_1 \cup C_2$.

The error can be written as

$$E_n(f) = \frac{1}{2\pi i}\left\{\int_{\Gamma_R} - \int_{C_1} - \int_{C_2}\right\}\frac{q_n(z)}{L_n(z)} f(z)dz.$$

One arrives at

$$\begin{aligned}|E_n(f)| \leq & 4\pi \left|\mathrm{Res}\left(f(z), z_0\right) e^{-z_0}\right| \times \\ & \times \left[\exp\left(\mathrm{Re}\left(\sqrt{-z_0}\right)\right)\right]^{-2\sqrt{n}} \times \\ & \times \left(1 + O\left(\frac{1}{n}\right)\right).\end{aligned}$$

Procedure

Apply the idea at $f(z) = (1 + e^{-z/\alpha}\lambda)^{-1}$, and $f(z) = (e^{-z/(1-\alpha)} + \lambda)^{-1}$. For the two integrals.

A Gauss-Laguerre approach (Aceto and Novati 2022)

Theorem (Aceto and Novati 2022, Proposition 5.3)

Let $R_{2n-1,2n}(\mathcal{L})$ be the Gauss-Laguerre rational approximation. Then, with respect to the operator norm in $\mathbb{H}$ we have for n large enough

$$\left\|\mathcal{L}^{-\alpha} - R_{2n-1,2n}(\mathcal{L})\right\| \leq 4\sin(\alpha\pi)\exp\left(-3\left(n\alpha^2\pi^2\right)^{1/3}\right)\left(1 + O\left(n^{-1/3}\right)\right).$$

A Gauss-Laguerre approach (Aceto and Novati 2022)

Theorem (Aceto and Novati 2022, Proposition 5.3)

Let $R_{2n-1,2n}(\mathcal{L})$ be the Gauss-Laguerre rational approximation. Then, with respect to the operator norm in $\mathbb{H}$ we have for n large enough

$$\left\|\mathcal{L}^{-\alpha} - R_{2n-1,2n}(\mathcal{L})\right\| \leq 4\sin(\alpha\pi)\exp\left(-3\left(n\alpha^2\pi^2\right)^{1/3}\right)\left(1 + O\left(n^{-1/3}\right)\right).$$

- The **convergence** is now **independent of the spectral information** of the matrix, we just need to scale A to have spectrum in $[1, +\infty)$.

A Gauss-Laguerre approach (Aceto and Novati 2022)

Theorem (Aceto and Novati 2022, Proposition 5.3)

Let $R_{2n-1,2n}(\mathcal{L})$ be the Gauss-Laguerre rational approximation. Then, with respect to the operator norm in $\mathbb{H}$ we have for n large enough

$$\left\|\mathcal{L}^{-\alpha} - R_{2n-1,2n}(\mathcal{L})\right\| \leq 4\sin(\alpha\pi)\exp\left(-3\left(n\alpha^2\pi^2\right)^{1/3}\right)\left(1 + O\left(n^{-1/3}\right)\right).$$

- The **convergence** is now **independent of the spectral information** of the matrix, we just need to scale A to have spectrum in $[1, +\infty)$.
- Truncation and balancing strategies can be applied to the quadratures observing that nodes and weights decay exponentially, i.e., apply

$$\mathcal{L}^{-\alpha} \approx \frac{\sin(\alpha\pi)}{\alpha\pi} R^{(1)}_{k_{n_1}-1,k_{n_1}}(\mathcal{L}) + \frac{\sin(\alpha\pi)}{(1-\alpha)\pi} R^{(2)}_{k_{n_2}-1,k_{n_2}}(\mathcal{L}).$$

Laplace-Stieltjes and Cauchy-Stieltjes functions

Functions expressed as Stieltjes integrals admit a representation of the form:

$$f(z) = \int_0^\infty g(t,z)\mu(t)\ \mathrm{d}t,$$

where

- $\mu(t)\mathrm{d}t$ is a (non-negative) on $[0,\infty]$, measure,
- $g(t,z)$ is integrable with respect to that measure.

Laplace-Stieltjes and Cauchy-Stieltjes functions

Functions expressed as Stieltjes integrals admit a representation of the form:

$$f(z) = \int_0^\infty g(t,z)\mu(t)\ \mathrm{d}t,$$

where

- $\mu(t)\mathrm{d}t$ is a (non-negative) on $[0,\infty]$, measure,
- $g(t,z)$ is integrable with respect to that measure.

Cauchy-Stieltjes

Let $f(z)$ be a function defined on $\mathbb{C}\setminus\mathbb{R}_-$. Then, $f(z)$ is a *Cauchy-Stieltjes* function if there is a positive measure $\mu(t)dt$ on $\mathbb{R}_+$ such that

$$f(z) = \int_0^\infty \frac{\mu(t)}{t+z}\ \mathrm{d}t.$$

Laplace-Stieltjes and Cauchy-Stieltjes functions

Functions expressed as Stieltjes integrals admit a representation of the form:

$$f(z) = \int_0^\infty g(t,z)\mu(t)\ \mathrm{d}t,$$

where

- $\mu(t)\mathrm{d}t$ is a (non-negative) on $[0,\infty]$, measure,
- $g(t,z)$ is integrable with respect to that measure.

Cauchy-Stieltjes

Let $f(z)$ be a function defined on $\mathbb{C}\setminus\mathbb{R}_-$. Then, $f(z)$ is a *Cauchy-Stieltjes* function if there is a positive measure $\mu(t)dt$ on $\mathbb{R}_+$ such that

$$f(z) = \int_0^\infty \frac{\mu(t)}{t+z}\ \mathrm{d}t.$$

The function we are interested in is of this class for $\alpha \in (0,1)$:

$$f(z) = z^{-\alpha} = \frac{\sin(\alpha\pi)}{\pi}\int_0^\infty \frac{t^{-\alpha}}{t+z}\ \mathrm{d}t.$$

In (Massei and Robol 2021) is given a general bound for the whole class of functions.

⏪ Back to Zolotarev

To **obtain the poles** we consider the approach of minimizing the expression of the error within the Krylov space for the entire class of functions: we **return to Zolotarev**.

⏪ Back to Zolotarev

To **obtain the poles** we consider the approach of minimizing the expression of the error within the Krylov space for the entire class of functions: we **return to Zolotarev**.

🔧 Let us write **compactly**: $\mathcal{W} = \mathcal{K}(A, \mathbf{v}, \Psi)$ for the rational Krylov subspace with poles Ψ. Then we can write the approximation error as:

$$\|\mathbf{x}_{\mathcal{W}} - \mathbf{x}\|_2 \leq 2 \cdot \|\mathbf{v}\|_2 \cdot \min_{r(z) \in \frac{\mathbb{P}_\ell}{\Psi}} \max_{z \in [a,b]} |f(z) - r(z)|.$$

where $\mathbf{x}_{\mathcal{W}} = Uf(U^H AU)U^H \mathbf{v}$ for U an orthonormal basis of $\mathcal{W}$, and $\mathbf{x} = f(A)\mathbf{v}$.

Back to Zolotarev

To **obtain the poles** we consider the approach of minimizing the expression of the error within the Krylov space for the entire class of functions: we **return to Zolotarev**.

Let us write **compactly**: $\mathcal{W} = \mathcal{K}(A, \mathbf{v}, \Psi)$ for the rational Krylov subspace with poles Ψ. Then we can write the approximation error as:

$$\|\mathbf{x}_{\mathcal{W}} - \mathbf{x}\|_2 \leq 2 \cdot \|\mathbf{v}\|_2 \cdot \min_{r(z) \in \frac{\mathbb{P}_\ell}{\Psi}} \max_{z \in [a,b]} |f(z) - r(z)|.$$

where $\mathbf{x}_{\mathcal{W}} = Uf(U^H AU)U^H \mathbf{v}$ for U an orthonormal basis of $\mathcal{W}$, and $\mathbf{x} = f(A)\mathbf{v}$.

Now comes the clever observation, the function we want to approximate is of the form

$$f(A)v = \int_0^\infty g(t, A)\mu(t)\, \mathrm{d}t, \qquad g(t, A) \in \{e^{-tA}, (tI + A)^{-1}\}$$

$\Rightarrow$ Since the **projection is linear** we need poles to approximate uniformly well (in t) the matrix exponentials and resolvents.

Cauchy-Stieltjes functions

For Cauchy-Stieltjes function, we just need the result for the resolvent function.

Theorem (Massei and Robol 2021, Theorem 1)

Let A be Hermitian positive definite with spectrum contained in $[a, b]$ and U be an orthonormal basis of $\mathcal{U}_{\mathcal{R}} = \mathcal{K}_\ell(A, v, \Psi)$. Then, $\forall t \in [0, \infty)$, we have the following inequality:

$$\|(tI + A)^{-1}\mathbf{v} - U(tI + A_\ell)^{-1}\mathbf{v}_\ell\|_2 \leq \frac{2}{t + a}\|\mathbf{v}\|_2 \min_{r(z) \in \frac{\mathcal{P}_\ell}{\Psi}} \frac{\max_{z \in [a,b]} |r(z)|}{\min_{z \in (-\infty, 0]} |r(z)|}$$

where $A_\ell = U^H A U$ and $\mathbf{v}_\ell = U^H v$.

Cauchy-Stieltjes functions

For Cauchy-Stieltjes function, we just need the result for the resolvent function.

Theorem (Massei and Robol 2021, Theorem 1)

Let A be Hermitian positive definite with spectrum contained in $[a, b]$ and U be an orthonormal basis of $\mathcal{U}_{\mathcal{R}} = \mathcal{K}_\ell(A, v, \Psi)$. Then, $\forall t \in [0, \infty)$, we have the following inequality:

$$\|(tI + A)^{-1}\mathbf{v} - U(tI + A_\ell)^{-1}\mathbf{v}_\ell\|_2 \leq \frac{2}{t + a}\|\mathbf{v}\|_2 \min_{r(z) \in \frac{\mathcal{P}_\ell}{\Psi}} \frac{\max_{z \in [a,b]} |r(z)|}{\min_{z \in (-\infty, 0]} |r(z)|}$$

where $A_\ell = U^H A U$ and $\mathbf{v}_\ell = U^H v$.

We got back to our favorite 4th problem of Zolotarev! Than we do not know how to solve in close form in general...

Cauchy-Stieltjes functions

For Cauchy-Stieltjes function, we just need the result for the resolvent function.

Theorem (Massei and Robol 2021, Theorem 1)

Let A be Hermitian positive definite with spectrum contained in $[a, b]$ and U be an orthonormal basis of $\mathcal{U}_{\mathcal{R}} = \mathcal{K}_\ell(A, v, \Psi)$. Then, $\forall t \in [0, \infty)$, we have the following inequality:

$$\|(tI + A)^{-1}\mathbf{v} - U(tI + A_\ell)^{-1}\mathbf{v}_\ell\|_2 \leq \frac{2}{t + a}\|\mathbf{v}\|_2 \min_{r(z) \in \frac{\mathcal{P}_\ell}{\Psi}} \frac{\max_{z \in [a,b]} |r(z)|}{\min_{z \in (-\infty, 0]} |r(z)|}$$

where $A_\ell = U^H A U$ and $\mathbf{v}_\ell = U^H v$.

- We got back to our favorite 4th problem of Zolotarev! Than we do not know how to solve in close form in general...
- this is not the general case, this is the case of two intervals $[a, b]$ and $(-\infty, 0]$ ☺

Solving this particular Zolotarev instance

The Zolotarev constant

Let $\Psi = \{\psi_1, \ldots, \psi_\ell\} \subset \overline{\mathbb{C}}$ be a finite set, and I_1, I_2 closed subsets of $\overline{\mathbb{C}}$. Then, we define

$$\theta_\ell(I_1, I_2, \Psi) = \min_{r(z) \in \frac{\mathcal{P}_\ell}{\Psi}} \frac{\max_{I_1} |r(z)|}{\min_{I_2} |r(z)|}.$$

Solving this particular Zolotarev instance

Theorem (Zolotarev)

Let $I = [a, b]$, with $0 < a < b$. Then

$$\min_{\Psi \subset \overline{\mathbb{C}},\ |\Psi| = \ell} \theta_\ell(I, -I, \Psi) \leq 4\rho_{[a,b]}^{\ell}, \qquad \rho_{[a,b]} = \exp\left(-\frac{\pi^2}{\log(4\kappa)}\right), \qquad \kappa = \frac{b}{a}.$$

In addition, the optimal rational function $r_\ell^{[a,b]}(z)$ that realizes the minimum has the form

$$r_\ell^{[a,b]}(z) = \frac{p_\ell^{[a,b]}(z)}{p_\ell^{[a,b]}(-z)}, \qquad p_\ell^{[a,b]}(z) = \prod_{j=1}^{\ell}(z + \psi_{j,\ell}^{[a,b]}), \qquad \psi_{j,\ell}^{[a,b]} \in -I.$$

We denote by $\Psi_\ell^{[a,b]} = \{\psi_{1,\ell}^{[a,b]}, \ldots, \psi_{\ell,\ell}^{[a,b]}\}$ the set of poles of $r_\ell^{[a,b]}(z)$.

Solving this particular Zolotarev instance

Theorem (Zolotarev)

Let $I = [a, b]$, with $0 < a < b$. Then

$$\min_{\Psi \subset \overline{\mathbb{C}},\ |\Psi| = \ell} \theta_\ell(I, -I, \Psi) \leq 4\rho_{[a,b]}^{\ell}, \qquad \rho_{[a,b]} = \exp\left(-\frac{\pi^2}{\log(4\kappa)}\right), \qquad \kappa = \frac{b}{a}.$$

In addition, the optimal rational function $r_\ell^{[a,b]}(z)$ that realizes the minimum has the form

$$r_\ell^{[a,b]}(z) = \frac{p_\ell^{[a,b]}(z)}{p_\ell^{[a,b]}(-z)}, \qquad p_\ell^{[a,b]}(z) = \prod_{j=1}^{\ell}(z + \psi_{j,\ell}^{[a,b]}), \qquad \psi_{j,\ell}^{[a,b]} \in -I.$$

We denote by $\Psi_\ell^{[a,b]} = \{\psi_{1,\ell}^{[a,b]}, \ldots, \psi_{\ell,\ell}^{[a,b]}\}$ the set of poles of $r_\ell^{[a,b]}(z)$.

⚠ This solution is for $I_1 = [a, b]$ and $I_2 = [-b, -a]$: we had $[a, b]$ and $(-\infty, 0]$!

Solving this particular Zolotarev instance

The Zolotarev constant

Let $\Psi = \{\psi_1, \ldots, \psi_\ell\} \subset \overline{\mathbb{C}}$ be a finite set, and I_1, I_2 closed subsets of $\overline{\mathbb{C}}$. Then, we define

$$\theta_\ell(I_1, I_2, \Psi) = \min_{r(z) \in \frac{\mathcal{P}_\ell}{\Psi}} \frac{\max_{I_1} |r(z)|}{\min_{I_2} |r(z)|}.$$

For any I_1, I_2 be subsets of the complex plane, and $\Psi \subset \overline{\mathbb{C}}$ we have

shift invariance For any $t \in \mathbb{C}$, it holds $\theta_\ell(I_1 + t, I_2 + t, \Psi + t) = \theta(I_1, I_2, \Psi)$.

monotonicity $\theta_\ell(I_1, I_2, \Psi)$ is monotonic with respect to the inclusion on the parameters I_1 and I_2: $I_1 \subseteq I_1', I_2 \subseteq I_2' \implies \theta_\ell(I_1, I_2, \Psi) \leq \theta_\ell(I_1', I_2', \Psi)$.

Möbius invariance If $M(z)$ is a Möbius transform, that is a rational function $M(z) = (\alpha z + \beta)/(\gamma z + \delta)$ with $\alpha\delta \neq \beta\gamma$, then

$$\theta_\ell(I_1, I_2, \Psi) = \theta_\ell(M(I_1), M(I_2), M(\Psi)).$$

⚠ This solution is for $I_1 = [a, b]$ and $I_2 = [-b, -a]$: we had $[a, b]$ and $(-\infty, 0]$!

Solving this particular Zolotarev instance

We just need to build the right Möbius transform to map

$$(-\infty, 0] \cup [a, b] \mapsto -I \cup I, \quad I = [a', b'],\ 0 < a' < b'.$$

Lemma (Massei and Robol 2021, Lemma 4)

The Möbius transformation

$$T_C(z) = \frac{\Delta + z - b}{\Delta - z + b}, \qquad \Delta = \sqrt{b^2 - ab},$$

maps $[-\infty, 0] \cup [a, b]$ into $[-1, -\widehat{a}] \cup [\widehat{a}, 1]$, with $\widehat{a} = \frac{\Delta + a - b}{\Delta - a + b} = \frac{b - \Delta}{\Delta + b}$. The inverse map $T_C(z)^{-1}$ is given by:

$$T_C^{-1}(z) = \frac{(b + \Delta)z + b - \Delta}{1 + z}.$$

Moreover, for any $0 < a < b$ it holds $\widehat{a}^{-1} \leq \frac{4b}{a}$, and therefore $\rho_{[\widehat{a},1]} \leq \rho_{[a,4b]}$.

Cauchy-Stieltjes functions

- We map the interval $[a, b]$ to $[\widehat{a}, 1]$,
- solve *explicitly* the Zolotarev problem there,
- read the poles for our problem.

Proposition (Massei and Robol 2021, Corollary 4)

Let $f(z)$ be a Cauchy-Stieltjes function, A be Hermitian positive definite with spectrum contained in $[a, b]$, U be an orthonormal basis of $\mathcal{K}_\ell(A, v, \Psi_{C,\ell}^{[a,b]})$ with $\Psi_{C,\ell}^{[a,b]}$ given by

$$\Psi_{C,\ell}^{[a,b]} = T_C^{-1}(\Psi_\ell^{[\widehat{a},1]})$$

and $\mathbf{x}_\ell = Uf(A_\ell)v_\ell$ with $A_\ell = U^H AU$ and $\mathbf{v}_\ell = U^H \mathbf{v}$. Then

$$\|f(A)\mathbf{v} - \mathbf{x}_\ell\|_2 \leq 8f(a)\|\mathbf{v}\|_2 \rho_{[a,4b]}^\ell = 8f(a)\exp\left(-\ell \frac{\pi^2}{\log(16b/a)}\right).$$

Nesting the poles

The poles built this way are still **not nested**. In (Massei and Robol 2021) a technique called method of equidistributed sequences (EDS) is proposed to generate them:

1. Select $\zeta \in \mathbb{R}^+ \setminus \mathbb{Q}$ and generate the sequence $\{s_j\}_{j\in\mathbb{N}} = \{0,\ \zeta - \lfloor\zeta\rfloor,\ 2\zeta - \lfloor 2\zeta\rfloor,\ 3\zeta - \lfloor 3\zeta\rfloor,\ \dots\}$, where $\lfloor\cdot\rfloor$ indicates the greatest integer less than or equal to the argument; this sequence has as asymptotic distribution (in the sense of EDS) the Lebesgue measure on $[0,1]$.
2. Compute the sequence $\{t_j\}_{j\in\mathbb{N}}$ such that $g(t_j) = s_j$ where

$$g(t) = \frac{1}{2M}\int_{a^2}^{t} \frac{dy}{\sqrt{(y-a^2)y(1-y)}}, \qquad M = \int_0^1 \frac{dy}{\sqrt{(1-y^2)(1-(1-a^2)y^2)}},$$

3. Define $\widetilde{\sigma}_j = \sqrt{t_j}$.

The EDS associated with $\Psi_\ell^{[a,b]}, \Psi_{C,\ell}^{[a,b]}$ are obtained by applying either a scaling or the Möbius transformation to the EDS for $\Psi_\ell^{[a,1]}$.

Brute force approaches

It is also possible to try and solve numerically rational approximation problems.

RKFIT (Berljafa and Güttel 2017) Is an iterative method for solving rational Least-Square problems, $\{A, F\} \in \mathbb{C}^{n\times n}$ and $\mathbf{b} \in \mathbb{C}^n$ find a ration function r such that

$$\|F\mathbf{b} - r(A)\mathbf{b}\|_2^2 \to \min .$$

AAA (Nakatsukasa, Sète, and Trefethen 2018) Find a representation of the rational approximant in barycentric form with interpolation at certain support points while performing a greedy selection of them to avoid exponential instabilities.

If we have an idea of *where* the approximation should work, these approaches deliver reasonably good results.

An Application to Complex Networks

The **spectral definition** makes the procedure ideal also in more *exotic* cases.

An Application to Complex Networks

The **spectral definition** makes the procedure ideal also in more *exotic* cases.

- A *weighted directed graph* (digraph) is a pair $G = (V, E, W)$, where $V = \{v_1, \ldots, v_n\}$ is a **set of nodes** (or vertices), and $E \subseteq V \times V$ is a **set** of **ordered pairs** of nodes called **edges**, and $W \in \mathbb{R}^{n \times n}$ such that $(W)_{i,j} \neq 0$ iff $(v_i, v_j) \in E$.

An Application to Complex Networks

The **spectral definition** makes the procedure ideal also in more *exotic* cases.

- A *weighted directed graph* (digraph) is a pair $G = (V, E, W)$, where $V = \{v_1, \dots, v_n\}$ is a **set of nodes** (or vertices), and $E \subseteq V \times V$ is a **set** of **ordered pairs** of nodes called **edges**, and $W \in \mathbb{R}^{n \times n}$ such that $(W)_{i,j} \neq 0$ iff $(v_i, v_j) \in E$.
- We call *in–degrees* and *out–degrees*

$$d_i^{(\text{in})} = \deg_{\text{in}}(v_i) = \sum_{j\,:\,(v_j, v_i) \in E} w_{j,i},$$

$$d_i^{(\text{out})} = \deg_{\text{out}}(v_i) = \sum_{j\,:\,(v_i, v_j) \in E} w_{i,j},$$

In matrix language

- If all the weights are equal to one, the **adjacency matrix** $A \in \mathbb{R}^{n \times n}$ is

$$(A)_{i,j} = a_{i,j} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

otherwise, $A \equiv W$.

An Application to Complex Networks

The **spectral definition** makes the procedure ideal also in more *exotic* cases.

- A *weighted directed graph* (digraph) is a pair $G = (V, E, W)$, where $V = \{v_1, \dots, v_n\}$ is a **set of nodes** (or vertices), and $E \subseteq V \times V$ is a **set** of **ordered pairs** of nodes called **edges**, and $W \in \mathbb{R}^{n \times n}$ such that $(W)_{i,j} \neq 0$ iff $(v_i, v_j) \in E$.
- We call *in–degrees* and *out–degrees*

$$d_i^{(\text{in})} = \deg_{\text{in}}(v_i) = \sum_{j\,:\,(v_j, v_i) \in E} w_{j,i},$$

$$d_i^{(\text{out})} = \deg_{\text{out}}(v_i) = \sum_{j\,:\,(v_i, v_j) \in E} w_{i,j},$$

In matrix language

- If all the weights are equal to one, the **adjacency matrix** $A \in \mathbb{R}^{n \times n}$ is

$$(A)_{i,j} = a_{i,j} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

  otherwise, $A \equiv W$.
- **Degree** diagonal **matrices**

$$\begin{aligned} D_{\text{in}} &= \operatorname{diag}(\deg_{\text{in}}(v_1), \dots, \deg_{\text{in}}(v_n)) \\ &= \operatorname{diag}(d_1^{(\text{in})}, \dots, d_n^{(\text{in})}), \\ D_{\text{out}} &= \operatorname{diag}(\deg_{\text{out}}(v_1), \dots, \deg_{\text{out}}(v_n)) \\ &= \operatorname{diag}(d_1^{(\text{out})}, \dots, d_n^{(\text{out})}). \end{aligned}$$

Laplacians on Graphs

Undirected case

Let $G = (V, E)$ be a weighted undirected graph with weight matrix W, weighted degree matrix D and weighted incidence matrix B. Then the **graph Laplacian** L of G is

$$L = D - W.$$

The *normalized random walk* version of the graph Laplacian is

$$D^{-1}L = I - D^{-1}W,$$

where I is the identity matrix. Observe that $D^{-1}W$ is a row–stochastic matrix, i.e. it is nonnegative with row sums equal to 1. The *normalized symmetric* version is

$$D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}.$$

If G is unweighted then $W = A$ in the above definitions. Here we assume that every vertex has nonzero degree.

Laplacians on Graphs

Directed case

Let $G = (V, E, W)$ be a weighted directed graph, with degree matrices D_{out} and D_{in} The **nonnormalized directed graph Laplacian** L_{out} and L_{in} of G are

$$L_{\text{out}} = D_{\text{out}} - W, \qquad L_{\text{in}} = D_{\text{in}} - W.$$

⚠ To define the **normalized versions**, we need to invert either the D_{in} or the D_{out} matrices, but the absence of isolated vertices is no longer sufficient to ensure this!

Laplacians on Graphs

Directed case

Let $G = (V, E, W)$ be a weighted directed graph, with degree matrices D_{out} and D_{in} The **nonnormalized directed graph Laplacian** L_{out} and L_{in} of G are

$$L_{\text{out}} = D_{\text{out}} - W, \qquad L_{\text{in}} = D_{\text{in}} - W.$$

⚠ To define the **normalized versions**, we need to invert either the D_{in} or the D_{out} matrices, but the absence of isolated vertices is no longer sufficient to ensure this!

It is **interesting to look at diffusion on graphs**:

$$\text{find } u : [0, T] \longrightarrow \mathbb{R}^n$$

$$\text{s.t.} \quad \begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t} u(t) = -\kappa L_{\cdot/\text{in}/\text{out}} u(t), & t \in (0, T], \\ u(0) = u_0, & \text{prescribed}, \end{cases}$$

Laplacians on Graphs

Directed case

Let $G = (V, E, W)$ be a weighted directed graph, with degree matrices D_{out} and D_{in} The **nonnormalized directed graph Laplacian** L_{out} and L_{in} of G are

$$L_{\text{out}} = D_{\text{out}} - W, \qquad L_{\text{in}} = D_{\text{in}} - W.$$

⚠ To define the **normalized versions**, we need to invert either the D_{in} or the D_{out} matrices, but the absence of isolated vertices is no longer sufficient to ensure this!

It is **interesting to look at diffusion on graphs**:

$$\text{find } u : [0, T] \longrightarrow \mathbb{R}^n$$

$$\text{s.t.} \quad \begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t} u(t) = -\kappa L^{\alpha}_{\cdot/\text{in/out}} u(t), & t \in (0, T], \\ u(0) = u_0, & \text{prescribed}, \end{cases} \qquad \alpha \in (0, 1],$$

$\Rightarrow$ it *could be* interesting to look at **fractional diffusion** on graphs.

Laplacians on Graphs

If G is *undirected*, i.e., $L = L^T$, everything follows by **diagonalization**, see, e.g., (Riascos and Mateos 2014).

Laplacians on Graphs

If G is *undirected*, i.e., $L = L^T$, everything follows by **diagonalization**, see, e.g., (Riascos and Mateos 2014).
If L is either L_{out} or L_{in} this **needs more care**.

Laplacians on Graphs

If G is *undirected*, i.e., $L = L^T$, everything follows by **diagonalization**, see, e.g., (Riascos and Mateos 2014).
If L is either L_{out} or L_{in} this **needs more care**.

- L_{out} is a **singular** M**-matrix**,
- $L_{\text{out}}\mathbf{1} = \mathbf{0}$,
- 0 is a *semisimple* eigenvalue of L_{out}.

Laplacians on Graphs

If G is *undirected*, i.e., $L = L^T$, everything follows by **diagonalization**, see, e.g., (Riascos and Mateos 2014).
If L is either L_{out} or L_{in} this **needs more care**.

- L_{out} is a **singular** M**-matrix**,
- $L_{\text{out}}\mathbf{1} = \mathbf{0}$,
- 0 is a *semisimple* eigenvalue of L_{out}.

⚖ We need to prove that L_{out}^{α} can be **defined** and **respect all the properties**.

Laplacians on Graphs

If G is *undirected*, i.e., $L = L^T$, everything follows by **diagonalization**, see, e.g., (Riascos and Mateos 2014).

If L is either L_{out} or L_{in} this **needs more care**.

- L_{out} is a **singular** M**-matrix**,
- $L_{\text{out}}\mathbf{1} = \mathbf{0}$,
- 0 is a *semisimple* eigenvalue of L_{out}.

We need to prove that L_{out}^{α} can be **defined** and **respect all the properties**.

Proposition (Benzi, Bertaccini, et al. 2020)

Given a weighted graph $G = (V, E, W)$ and its Laplacian with respect to the out degree L_{out}, the function $f(x) = x^{\alpha}$ is defined on the spectrum of L_{out} and induces a matrix function for all $\alpha \in (0, 1]$.

Laplacians on Graphs

If G is *undirected*, i.e., $L = L^T$, everything follows by **diagonalization**, see, e.g., (Riascos and Mateos 2014).
If L is either L_{out} or L_{in} this **needs more care**.

- L_{out} is a **singular** M**-matrix**,
- $L_{\text{out}}\mathbf{1} = \mathbf{0}$,
- 0 is a *semisimple* eigenvalue of L_{out}.

We need to prove that L_{out}^{α} can be **defined** and **respect all the properties**.

Proposition (Benzi, Bertaccini, et al. 2020)

If A is a singular M-matrix with 0 as a semisimple eigenvalue, then there exists a determination of A^{α} for every $\alpha \in (0, 1]$ that is a singular M-matrix.

Laplacians on Graphs

If G is *undirected*, i.e., $L = L^T$, everything follows by **diagonalization**, see, e.g., (Riascos and Mateos 2014).
If L is either L_{out} or L_{in} this **needs more care**.

- L_{out} is a **singular** M**-matrix**,
- $L_{\text{out}}\mathbf{1} = \mathbf{0}$,
- 0 is a *semisimple* eigenvalue of L_{out}.

We need to prove that L_{out}^{α} can be **defined** and **respect all the properties**.

Proposition (Benzi, Bertaccini, et al. 2020)

If A is a singular M-matrix with 0 as a semisimple eigenvalue, then there exists a determination of A^{α} for every $\alpha \in (0, 1]$ that is a singular M-matrix.

We could also investigate the **the decay of the entries** of the fractional power, but leave the subject aside and refer to (Benzi, Bertaccini, et al. 2020).

Laplacian on Graphs: computation

⚠ For the computation of the products $L_{\text{out}}^{\alpha}\mathbf{v}$ it is necessary to **modify the strategies** we have seen: all the bounds and constructions required that 0 was not in the spectrum.

Laplacian on Graphs: computation

For the computation of the products $L_{\text{out}}^{\alpha}\mathbf{v}$ it is necessary to **modify the strategies** we have seen: all the bounds and constructions required that 0 was not in the spectrum.

In (Benzi and Simunec 2022) different strategies for accommodating this feature of L_{out} are investigated:

1. Use a **rank-one** shift, since the right and left eigenvectors $\mathbf{1}$ and $\vec{z}$ of L_{out} can be easily computed, we compute

$$f(L^T)\mathbf{b} = f(L^T + \theta\mathbf{z}\mathbf{1}^T)\mathbf{b} + [f(0) - f(\theta)]\mathbf{z}, \text{ for any } \theta > 0,$$

and in the rational Krylov subspace we solve the linear system at the same cost at which we solve the ones for L^T via Sherman-Morrison:

$$(L^T + \theta\mathbf{z}\mathbf{1}^T - \xi I)^{-1} = (L^T - \xi I)^{-1} + \frac{\theta}{\xi(\theta - \xi)}\mathbf{z}\mathbf{1}^T.$$

Laplacian on Graphs: computation

⚠ For the computation of the products $L_{\text{out}}^{\alpha}\mathbf{v}$ it is necessary to **modify the strategies** we have seen: all the bounds and constructions required that 0 was not in the spectrum.

🔨 In (Benzi and Simunec 2022) different strategies for accommodating this feature of L_{out} are investigated:

1. Use a **rank-one** shift, since the right and left eigenvectors $\mathbf{1}$ and $\vec{z}$ of L_{out} can be easily computed, we compute

$$f(L^T)\mathbf{b} = f(L^T + \theta\mathbf{z}\mathbf{1}^T)\mathbf{b} + [f(0) - f(\theta)]\mathbf{z}, \text{ for any } \theta > 0,$$

and in the rational Krylov subspace we solve the linear system at the same cost at which we solve the ones for L^T by doing

$$(L^T + \theta\mathbf{z}\mathbf{1}^T - \xi I)^{-1}\mathbf{w} = \boldsymbol{\psi} + \frac{\mathbf{1}^T\mathbf{w}}{\theta - \xi}\mathbf{z} \text{ and } (L^T - \xi I)\boldsymbol{\psi} = \mathbf{w} - (\mathbf{1}^T\mathbf{w})\mathbf{z},$$

to **avoid cancellation** for $\xi \approx 0$.

Laplacian on Graphs: computation

⚠ For the computation of the products $L_{\text{out}}^{\alpha}\mathbf{v}$ it is necessary to **modify the strategies** we have seen: all the bounds and constructions required that 0 was not in the spectrum.

🔨 In (Benzi and Simunec 2022) different strategies for accommodating this feature of L_{out} are investigated:

- Use a **rank-one** shift, since the right and left eigenvectors $\mathbf{1}$ and $\vec{z}$ of L_{out} can be easily computed, we compute

$$f(L^T)\mathbf{b} = f(L^T + \theta\mathbf{z}\mathbf{1}^T)\mathbf{b} + [f(0) - f(\theta)]\mathbf{z}, \text{ for any } \theta > 0.$$

- Project L on the $n-1$ dimensional subspace $\mathcal{S} = \mathrm{Span}\{\mathbf{1}\}^{\perp} = \mathrm{Range}(\tilde{Q})$ and compute

$$\begin{aligned} f(L^T)\mathbf{b} &= f(L^T)\mathbf{v} + \beta f(L^T)\mathbf{z} && \leftarrow 0 \neq \beta = \mathbf{1}^T\mathbf{b} \text{ and } \mathbf{b} = \mathbf{v} + \beta\mathbf{z} \text{ for } \mathbf{v} \perp \mathbf{1} \\ &= Qf(Q^TL^TQ)Q^T\mathbf{v} + \beta f(0)\mathbf{z} && \leftarrow QQ^T = I - \mathbf{1}\mathbf{1}^T/n,\ Q = [\tilde{Q}, \mathbf{1}/\sqrt{n}]. \end{aligned}$$

❗ Q can be built so that $\{Q, Q^T\}\mathbf{v}$ costs $O(n)$.

A gallery of open problems

"When sorrows come, they come
not single spies, but in battalions"
Hamlet, Act IV, Scene V.

- Of the many problems we have discussed along the way, one that came back many times was the **selection of optimal poles** for the different matrix-equation/Rational Krylov based solvers (e.g., *all-at-once*, multi-dimensional approaches);

A gallery of open problems

"When sorrows come, they come
not single spies, but in battalions"
Hamlet, Act IV, Scene V.

- Of the many problems we have discussed along the way, one that came back many times was the **selection of optimal poles** for the different matrix-equation/Rational Krylov based solvers (e.g., *all-at-once*, multi-dimensional approaches);
- Inventing **reduced memory** methods for the integration of fractional partial differential equations in time and space, i.e.,

$$ {}^{CA}D_t^{\alpha}\mathbf{u} = \mathcal{L}(\mathbf{u}; t), \qquad \mathcal{L} \text{ non linear, and fractional;} $$

A gallery of open problems

"When sorrows come, they come not single spies, but in battalions"
Hamlet, Act IV, Scene V.

- Of the many problems we have discussed along the way, one that came back many times was the **selection of optimal poles** for the different matrix-equation/Rational Krylov based solvers (e.g., *all-at-once*, multi-dimensional approaches);
- Inventing **reduced memory** methods for the integration of fractional partial differential equations in time and space, i.e.,

$$ {}^{CA}D_t^{\alpha}\mathbf{u} = \mathcal{L}(\mathbf{u}; t), \qquad \mathcal{L} \text{ non linear, and fractional;} $$

- **Error analysis** entangling convergence of the Rational Krylov method and Finite Element (Isogeometric) Discretizations for FPDEs;

A gallery of open problems

"When sorrows come, they come not single spies, but in battalions"
Hamlet, Act IV, Scene V.

- Of the many problems we have discussed along the way, one that came back many times was the **selection of optimal poles** for the different matrix-equation/Rational Krylov based solvers (e.g., *all-at-once*, multi-dimensional approaches);
- Inventing **reduced memory** methods for the integration of fractional partial differential equations in time and space, i.e.,

$$ {}^{CA}D_t^{\alpha}\mathbf{u} = \mathcal{L}(\mathbf{u}; t), \qquad \mathcal{L} \text{ non linear, and fractional;} $$

- **Error analysis** entangling convergence of the Rational Krylov method and Finite Element (Isogeometric) Discretizations for FPDEs;
- Solving FPDEs on **unlimited spatial domains**.

Fractional Schrödinger equation

As we have discussed at the beginning of the lecture, there are several formulations of the Fractional Laplacian that should be naturally considered on the whole space.

An example is the **Schrödinger equation**

$$i\hbar^{\beta}\,{}^{CA}D^{\beta}\psi = -D_{\alpha}(-\hbar^2\Delta)^{\alpha/2}\psi + V(\mathbf{x},t)\psi,$$

that is naturally defined on the whole space.

To treat it numerically, the usual procedure is to couple it with **artificial boundary conditions of absorbing type**. It might be of interest to have **numerical methods** that can work with **infinite** or **semi-infinite matrices** that do not need this artificial correction.

Conclusions

- We focused on *few discretization*, there are many other viable approaches (*collocation*, *finite elements*, *IgA*,...). *Most* of the reasoning we did can be **adapted to these other cases**.
- There are **other classical problems** that admits a fractional extension, *e.g.*, optimal control, model order reduction, eigenvalue problems,...

"*The universe (which others call the Library) is composed of an indefinite and perhaps infinite number of hexagonal galleries, with vast air shafts between, surrounded by very low railings. From any of the hexagons one can see, interminably, the upper and lower floors. The distribution of the galleries is invariable.*"

Jorge Luis Borges, The Library of Babel.

Bibliography I

- Aceto, L., D. Bertaccini, et al. (2019). "Rational Krylov methods for functions of matrices with applications to fractional partial differential equations". In: *J. Comput. Phys.* 396, pp. 470–482. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2019.07.009. URL: https://doi.org/10.1016/j.jcp.2019.07.009.
- Aceto, L. and P. Novati (2018). "Efficient implementation of rational approximations to fractional differential operators". In: *J. Sci. Comput.* 76.1, pp. 651–671. ISSN: 0885-7474. DOI: 10.1007/s10915-017-0633-2. URL: https://doi.org/10.1007/s10915-017-0633-2.
- — (2019). "Rational approximations to fractional powers of self-adjoint positive operators". In: *Numer. Math.* 143.1, pp. 1–16. ISSN: 0029-599X. DOI: 10.1007/s00211-019-01048-4. URL: https://doi.org/10.1007/s00211-019-01048-4.
- — (2022). "Fast and accurate approximations to fractional powers of operators". In: *IMA J. Numer. Anal.* 42.2, pp. 1598–1622. ISSN: 0272-4979. DOI: 10.1093/imanum/drab002. URL: https://doi.org/10.1093/imanum/drab002.

Bibliography II

- Andreu-Vaillo, F. et al. (2010). *Nonlocal diffusion problems*. Vol. 165. Mathematical Surveys and Monographs. American Mathematical Society, Providence, RI; Real Sociedad Matemática Española, Madrid, pp. xvi+256. ISBN: 978-0-8218-5230-9. DOI: 10.1090/surv/165. URL: https://doi.org/10.1090/surv/165.
- Benzi, M., D. Bertaccini, et al. (2020). "Non-local network dynamics via fractional graph Laplacians". In: *J. Complex Netw.* 8.3, cnaa017, 29. ISSN: 2051-1310. DOI: 10.1093/comnet/cnaa017. URL: https://doi.org/10.1093/comnet/cnaa017.
- Benzi, M. and I. Simunec (2022). "Rational Krylov methods for fractional diffusion problems on graphs". In: *BIT* 62.2, pp. 357–385. ISSN: 0006-3835. DOI: 10.1007/s10543-021-00881-0. URL: https://doi.org/10.1007/s10543-021-00881-0.
- Berljafa, M. and S. Güttel (2017). "The RKFIT algorithm for nonlinear rational approximation". In: *SIAM J. Sci. Comput.* 39.5, A2049–A2071. ISSN: 1064-8275. DOI: 10.1137/15M1025426. URL: https://doi.org/10.1137/15M1025426.

Bibliography III

- Bhatia, R. (1997). *Matrix analysis*. Vol. 169. Graduate Texts in Mathematics. Springer-Verlag, New York, pp. xii+347. ISBN: 0-387-94846-5. DOI: 10.1007/978-1-4612-0653-8. URL: https://doi.org/10.1007/978-1-4612-0653-8.
- Bonito, A. and J. E. Pasciak (2015). "Numerical approximation of fractional powers of elliptic operators". In: *Math. Comp.* 84.295, pp. 2083–2110. ISSN: 0025-5718. DOI: 10.1090/S0025-5718-2015-02937-8. URL: https://doi.org/10.1090/S0025-5718-2015-02937-8.
- Braess, D. (1986). *Nonlinear approximation theory*. Vol. 7. Springer Series in Computational Mathematics. Springer-Verlag, Berlin, pp. xiv+290. ISBN: 3-540-13625-8. DOI: 10.1007/978-3-642-61609-9. URL: https://doi.org/10.1007/978-3-642-61609-9.
- Brezinski, C. (1980). *Padé-type approximation and general orthogonal polynomials*. Vol. 50. International Series of Numerical Mathematics. Birkhäuser Verlag, Basel-Boston, Mass., p. 250. ISBN: 3-7643-1100-2.

Bibliography IV

- Davis, P. J. and P. Rabinowitz (1984). *Methods of numerical integration*. Second. Computer Science and Applied Mathematics. Academic Press, Inc., Orlando, FL, pp. xiv+612. ISBN: 0-12-206360-0.
- Frommer, A., S. Güttel, and M. Schweitzer (2014). "Efficient and stable Arnoldi restarts for matrix functions based on quadrature". In: *SIAM J. Matrix Anal. Appl.* 35.2, pp. 661–683. ISSN: 0895-4798. DOI: 10.1137/13093491X. URL: https://doi.org/10.1137/13093491X.
- Harizanov, S. et al. (2020). "Analysis of numerical methods for spectral fractional elliptic equations based on the best uniform rational approximation". In: *J. Comput. Phys.* 408, pp. 109285, 21. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2020.109285. URL: https://doi.org/10.1016/j.jcp.2020.109285.
- Hofreither, C. (2021). "An algorithm for best rational approximation based on barycentric rational interpolation". In: *Numer. Algorithms* 88.1, pp. 365–388. ISSN: 1017-1398. DOI: 10.1007/s11075-020-01042-0. URL: https://doi.org/10.1007/s11075-020-01042-0.
- Ilic, M. et al. (2005). "Numerical approximation of a fractional-in-space diffusion equation. I". In: *Fract. Calc. Appl. Anal.* 8.3, pp. 323–341. ISSN: 1311-0454.

Bibliography V

- Ilic, M. et al. (2006). "Numerical approximation of a fractional-in-space diffusion equation. II. With nonhomogeneous boundary conditions". In: *Fract. Calc. Appl. Anal.* 9.4, pp. 333–349. ISSN: 1311-0454.
- Kwaśnicki, M. (2017). "Ten equivalent definitions of the fractional Laplace operator". In: *Fract. Calc. Appl. Anal.* 20.1, pp. 7–51. ISSN: 1311-0454. DOI: 10.1515/fca-2017-0002. URL: https://doi.org/10.1515/fca-2017-0002.
- Lischke, A. et al. (2020). "What is the fractional Laplacian? A comparative review with new results". In: *J. Comput. Phys.* 404, pp. 109009, 62. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2019.109009. URL: https://doi.org/10.1016/j.jcp.2019.109009.
- Massei, S. and L. Robol (2021). "Rational Krylov for Stieltjes matrix functions: convergence and pole selection". In: *BIT* 61.1, pp. 237–273. ISSN: 0006-3835. DOI: 10.1007/s10543-020-00826-z. URL: https://doi.org/10.1007/s10543-020-00826-z.
- Musina, R. and A. I. Nazarov (2014). "On fractional Laplacians". In: *Comm. Partial Differential Equations* 39.9, pp. 1780–1790. ISSN: 0360-5302. DOI: 10.1080/03605302.2013.864304. URL: https://doi.org/10.1080/03605302.2013.864304.

Bibliography VI

- Nakatsukasa, Y., O. Sète, and L. N. Trefethen (2018). "The AAA algorithm for rational approximation". In: *SIAM J. Sci. Comput.* 40.3, A1494–A1522. ISSN: 1064-8275. DOI: 10.1137/16M1106122. URL: https://doi.org/10.1137/16M1106122.
- Riascos, A. and J. Mateos (2014). "Fractional dynamics on networks: Emergence of anomalous diffusion and Lévy flights". In: *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 90.3. cited By 49. DOI: 10.1103/PhysRevE.90.032809. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84907266357&doi=10.1103%2fPhysRevE.90.032809&partnerID=40&md5=be06b3148ba7bc17a50f52854beb9fac.
- Stahl, H. R. (2003). "Best uniform rational approximation of x^α on $[0,1]$". In: *Acta Math.* 190.2, pp. 241–306. ISSN: 0001-5962. DOI: 10.1007/BF02392691. URL: https://doi.org/10.1007/BF02392691.