



Regularization of inverse problems by an approximate matrix-function technique

Stefano Cipolla¹ · Marco Donatelli² · Fabio Durastante³

Received: 15 February 2020 / Accepted: 20 January 2021 / Published online: 11 April 2021
© The Author(s) 2021

Abstract

In this work, we introduce and investigate a class of matrix-free regularization techniques for discrete linear ill-posed problems based on the approximate computation of a special matrix-function. In order to produce a regularized solution, the proposed strategy employs a regular approximation of the Heavyside step function computed into a *small* Krylov subspace. This particular feature allows our proposal to be independent from the structure of the underlying matrix. If on the one hand, the use of the Heavyside step function prevents the amplification of the noise by suitably filtering the responsible components of the spectrum of the discretization matrix, on the other hand, it permits the correct reconstruction of the signal inverting the remaining part of the spectrum. Numerical tests on a gallery of standard benchmark problems are included to prove the efficacy of our approach even for problems affected by a high level of noise.

Keywords Regularization · Matrix function · Krylov methods

Mathematics Subject Classification (2010) 65F22 · 65F60 · 65F20

✉ Stefano Cipolla
stefano.cipolla87@gmail.com

Marco Donatelli
marco.donatelli@uninsubria.it

Fabio Durastante
f.durastante@na.iac.cnr.it

¹ Department of Mathematics, University of Padua, Via Trieste, 63, 35121, Padua, Italy

² Department of Science and High Technology, University of Insubria, Via Valleggio, 11, 22100, Como, Italy

³ Consiglio Nazionale delle Ricerche, Istituto per le Applicazioni del Calcolo “M. Picone”, Via Pietro Castellino n. 111, 80131, Naples, Italy

1 Introduction

Object of this work is the numerical solution of the problem

$$A_{n,m}\bar{\mathbf{x}}_m = \mathbf{g}_n \equiv \bar{\mathbf{g}}_n + \boldsymbol{\varepsilon}, \quad A_{n,m} \in \mathbb{R}^{n \times m}, \quad \mathbf{x}_m \in \mathbb{R}^m, \quad \mathbf{g}_n, \boldsymbol{\varepsilon} \in \mathbb{R}^n, \quad n, m > 0, \quad (1)$$

obtained from the discretization of an ill-posed problem in the Hadamard sense [1], in which the right-hand side \mathbf{g}_n is the sum of the true signal $\bar{\mathbf{g}}_n$ and of a noise vector $\boldsymbol{\varepsilon}$ with $\|\boldsymbol{\varepsilon}\| = \delta$.

A vast source of problems of this type is indeed the class of *inverse problems*, such as the numerical solution of Fredholm integral equations of the first kind with nondegenerate kernels, deblurring problems for images, and, more generally, the application of linear models for the reconstruction of corrupted signals.

The usual techniques for the solution of linear systems cannot be applied directly to (1) since their straightforward application would end up in the recovery of a non-physical/noisy corrupted solution [2]. A vast amount of ad-hoc techniques have been proposed in the literature to overcome this problem and they are broadly referred to as *regularization techniques*. In particular, we cite two strategies related to our proposal: regularizing preconditioning and hybrid methods.

Regularizing preconditioners have been developed to accelerate the convergence of iterative methods likes Landweber or CGLS, without spoiling the computed solution avoiding to amplify the high frequencies, where the noise lives. Many proposals define the preconditioner in a trigonometric matrix algebra [2–6], that can be applied only to shift-invariant operators. Since the preconditioner should preserve the structure of the coefficient matrix $A_{n,m}$ (see [7]), we follow the more general approach to define it in a Krylov subspace as suggested in [8, 9].

The combination of a direct solution of the original problem in a small size Krylov subspace is usually referred as a hybrid method. In more detail, hybrid methods combine the iterative construction of the Krylov subspace with the exact solution of the Tikhonov problem projected into the computed Krylov subspace [10–13]. The small size of the Krylov space allows to efficiently solve the projected Tikhonov problem.

In this work, tackling problem (1) through the normal equation

$$A_{n,m}^T A_{n,m} \bar{\mathbf{x}}_m = A_{n,m}^T \mathbf{g}_n, \quad (2)$$

we propose and investigate a class of regularization techniques based on a Matrix-Function approach. In detail, a regularized approximation of the solution of problem (1) is obtained as

$$\mathbf{x}_{m,\alpha}^\delta = f_\alpha(A_{m,n}^T A_{m,n}) A_{m,n}^T \mathbf{g}_n, \quad (3)$$

where $f_\alpha(z)$ is a suitable regularization of the inverse function applied to $A_{m,n}^T A_{m,n}$, i.e., $f_\alpha(A_{m,n}^T A_{m,n}) A_{m,n}^T A_{m,n} \approx I$ and, at the same time, $f_\alpha(A_{m,n}^T A_{m,n})$ does not propagate the noise $\boldsymbol{\varepsilon}$ presents in \mathbf{g}_n .

The matrix function f_α is defined having the property

$$f_\alpha(\lambda)\lambda \approx h_\alpha(\lambda), \quad (4)$$

where $h_\alpha(\lambda)$ is a differentiable approximation of the Heavyside step function in α , i.e.,

$$\text{given } \beta > 0, \quad h_\alpha(\lambda) = \begin{cases} 1, & \lambda > \alpha + \beta^{-1}, \\ 1/2, & \lambda = \alpha, \\ 0, & \lambda < \alpha - \beta^{-1}. \end{cases} \tag{5}$$

In this way, we are able to filter the spectrum of the inverse of $A_{m,n}^T A_{m,n}$ clustering to zero the eigenvalues smaller than α and inverting the remaining ones. As a result, the eigenvectors of the eigenvalues clustered to zero are neglected in the approximate solution and the propagation of the noise due to the ill-conditioning is avoided.

In this paper, we consider a smooth approximation of the Heavyside step function (5) previously proposed in [14, 15] for the analysis of electronic structures. Moreover, in order to obtain a fast matrix function evaluation, we consider a Krylov subspace method of polynomial type based on either the Arnoldi or the Lanczos decomposition, according to the properties of the matrix $A_{m,n}$. Hence, we obtain a hybrid method combining the approximate Heavyside step function with the iterative regularization due to the iterative computation of the Krylov subspace. Our proposal is independent of the particular structure of $A_{n,m}$ and it allows to work in a matrix-free framework simply requiring only the matrix-vector product operation.

The parameter α will be fixed according to the noise level assumed to be known, while the computation of the Krylov basis is stopped according to the combination of the discrepancy principle with a new criterion based on the stagnation of the norm of the residual. The several numerical results presented include different kinds of inverse problems proving that the proposed framework is general and robust. To enhance the reproducibility of the presented results, the codes of the new algorithms are publicly available (see <https://github.com/Cirdans-Home/IRfun>).

The paper is organized as follows. In Section 2 we discuss the construction of the approximation (3) connecting it to the classic works in regularization [1]. In Section 3 we define our proposal for the actual computation of our matrix-function regularization and we provide two suitable stopping criteria. In Section 4 we apply the proposed procedures to several test problems proving the goodness of our proposal. Finally, Section 5 is devoted to some final comments.

2 Motivations and literature review

2.1 Filter-based regularization methods

In this section we briefly summarize the framework of filter-based regularization methods. The notation and results are entirely borrowed from [1] and the interested reader can find there further details. Let us consider a compact linear operator $A : \mathcal{X} \rightarrow \mathcal{Y}$ between Hilbert spaces \mathcal{X}, \mathcal{Y} and denote with $A^* : \mathcal{Y} \rightarrow \mathcal{X}$ the adjoint operator of A . We denote by A^\dagger the Moore-Penrose inverse of A , and by $D(A^\dagger)$ its domain.

We are interested in an *ill-posed* (in the Hadamard-sense) linear operator equation of the form

$$Ax = g. \tag{6}$$

The following results hold:

Theorem 1 *Let $g \in D(A^\dagger)$. Then, $Ax = g$ has a unique best-approximate solution, which is given by*

$$x^\dagger := A^\dagger g.$$

We have, moreover, that

Theorem 2 *Let $g \in D(A^\dagger)$; $x \in \mathcal{X}$ is a least-square solution of $Ax = g$ if and only if the normal equation*

$$A^*Ax = A^*g \tag{7}$$

holds.

From the above results it follows that x^\dagger is the solution of (7), i.e.,

$$A^\dagger = (A^*A)^\dagger A^*.$$

When considering problem (6) where only an approximation g^δ of g is available and δ is the noise level, due to the unboundedness of A^\dagger , the solution $A^\dagger g^\delta$ is not a good approximation of $x^\dagger = A^\dagger g$ (we suppose g to be attainable). This follows by the following simple observation: consider the singular value expansion (s.v.e.) of A ($\sigma_i; v_i, u_i)_{i \in \mathbb{N}}$, then if $g \in D(A^\dagger)$, we have

$$A^\dagger g = \sum_{i=1}^{\infty} \frac{\langle g, u_i \rangle}{\sigma_i} v_i, \tag{8}$$

which clearly shows that errors in $\langle g, u_i \rangle$ are propagated with a factor of $1/\sigma_i$.

In practice, problem (6) is approximated by a family of neighboring well-posed problems [1].

Definition 1 By regularization operator for A^\dagger we call any family of operators

$$\{R_\alpha\}_{\alpha \in (0, \alpha_0)} : \mathcal{Y} \rightarrow \mathcal{X}, \quad \alpha_0 \in (0, +\infty] \tag{9}$$

with the following properties:

1. $R_\alpha : \mathcal{Y} \rightarrow \mathcal{X}$ is bounded for every α ;
2. For every $g \in D(A^\dagger)$ there exists a *rule choice* $\alpha : \mathbb{R}_+ \times \mathcal{Y} \rightarrow (0, \alpha_0) \subset \mathbb{R}$, $\alpha = \alpha(\delta, g^\delta)$, such that

$$\limsup_{\delta \rightarrow 0} \{\alpha(\delta, g^\delta) : g^\delta \in \mathcal{Y}, \|g - g^\delta\| \leq \delta\} = 0,$$

and

$$\limsup_{\delta \rightarrow 0} = \{\|R_{\alpha(\delta, g^\delta)} g^\delta - A^\dagger g\| : g^\delta \in \mathcal{Y}, \|g - g^\delta\| \leq \delta\} = 0.$$

Thus, a regularization method consists of a regularization operator and a parameter choice rule which is convergent in the sense that, if the regularization parameter is chosen according to that rule, then the regularized solutions converge as the noisy level tends to zero. Moreover, we have

Proposition 1 *Let, for all $\alpha > 0$, R_α be a continuous operator. Then, the family $\{R_\alpha\}$ is a regularization for A^\dagger if*

$$R_\alpha \rightarrow A^\dagger \text{ pointwise on } D(A^\dagger) \text{ as } \alpha \rightarrow 0. \tag{10}$$

We consider here a class of linear regularization methods based on spectral theory for selfadjoint compact linear operators. The basic idea is the following one: let $\{E_\lambda\}$ be a spectral family for A^*A . The best-approximate solution $x^\dagger = A^\dagger g$ can be characterized as

$$x^\dagger = \int \frac{1}{\lambda} dE_\lambda A^* g.$$

When the above integral does not exist due to the fact that $1/\lambda$ has a pole in 0, then the problem $Ax = g$ is ill-posed, and the idea is now to replace $1/\lambda$ by a parameter dependent family of functions $f_\alpha(\lambda)$, i.e.,

$$x_\alpha := \int f_\alpha(\lambda) dE_\lambda A^* g =: f_\alpha(A^*A)A^* g,$$

i.e., we are considering

$$R_\alpha := \int f_\alpha(\lambda) dE_\lambda A^* \tag{11}$$

as regularization operator for A^\dagger .

The following result gives sufficient conditions for R_α as in (11), to be a regularization operator for A^\dagger :

Theorem 3 *Let, for all $\alpha > 0$ and an $\varepsilon > 0$, $f_\alpha : [0, \|A\|^2 + \varepsilon) \rightarrow \mathbb{R}$ fulfills the following assumptions: f_α is piece-wise continuous, and there exists a $C > 0$ such that*

$$|\lambda f_\alpha(\lambda)| \leq C$$

and

$$\lim_{\alpha \rightarrow 0} f_\alpha(\lambda) = \frac{1}{\lambda}$$

for all $\lambda \in (0, \|A\|^2]$. Then, for all $g \in D(A^\dagger)$

$$\lim_{\alpha \rightarrow 0} f_\alpha(A^*A)A^* g = A^\dagger g.$$

If $y \notin D(A^\dagger)$, then $\lim_{\alpha \rightarrow 0} \|f_\alpha(A^*A)A^* g\| = +\infty$.

2.2 Tikhonov regularization and Heavyside step function

A special choice for f_α which fulfills the assumptions of Theorem 3 is

$$f_\alpha(\lambda) := \frac{1}{\lambda + \alpha}.$$

In this case we have

$$x_\alpha^\delta = f_\alpha(A^*A)A^*g^\delta = \sum_{i=1}^\infty \frac{\sigma_i}{\sigma_i^2 + \alpha} \langle g^\delta, u_i \rangle v_i. \tag{12}$$

Formula (12), if on the one hand, clearly shows the *regularization capabilities* for this choice of $f_\alpha(\lambda)$ since errors in $\langle g^\delta, u_i \rangle$ are not propagated with factors $1/\sigma_i$ but with factors $\sigma_i/(\sigma_i^2 + \alpha)$, on the other hand, it shows the limitation of choosing a uniform shift of all the singular values: this choice is responsible of an over-damping of the largest singular values.

Our proposal synthesizes and moves from the above observation. Let us consider (12) for a generic f_α , we have

$$x_\alpha^\delta = f_\alpha(A^*A)A^*g^\delta = \sum_{i=1}^\infty f_\alpha(\sigma_i^2)\sigma_i \langle g^\delta, u_i \rangle v_i. \tag{13}$$

In order to provide a regularized solution x_α^δ , using (8), the function f_α should be s.t.

$$f_\alpha(\sigma_i^2)\sigma_i = \begin{cases} 1/\sigma_i, & \text{if } \sigma_i^2 > \alpha \\ 0, & \text{if } \sigma_i^2 < \alpha, \end{cases}$$

i.e., it should be

$$f_\alpha(\sigma_i^2)\sigma_i^2 \approx h_\alpha(\sigma_i^2),$$

from which we recover (5). Observe that the function

$$f_\alpha(\lambda) = \frac{h_\alpha(\lambda)}{\lambda} \tag{14}$$

clearly satisfies the hypothesis of Theorem 3, and hence

$$R_\alpha := \int \frac{h_\alpha(\lambda)}{\lambda} dE_\lambda A^*$$

is a regularizing operator for A^\dagger .

Remark 1 We remark in passing [1], that when the operator A is already selfadjoint positive semidefinite, one will not use $R_\alpha := f_\alpha(A^*A)A^*$, but apply the theory of regularization methods for equations with selfadjoint operators, where R_α would be just $f_\alpha(A)$.

Moreover, from now on, as it is customary, we will suppose that an approximation $A_{n,m}$ of A in finite dimensional subspaces is available, i.e., that we have a discretization of the regularization operator [1]. In the following part we focus then on the construction of a regularization approach of *hybrid type* in which the regularization

is effected by both the parameters defining the function $f_\alpha(\lambda)$ and the choice of the projection subspace (see, e.g., [11–13]).

2.3 Regularizing preconditioners

In this section, for the sake of simplicity, we will suppose that $A_n \equiv A_{n,n}$ is a symmetric positive definite matrix. Then, using Remark 1, instead of solving problem in (2), we can suppose to solve

$$A_n \bar{\mathbf{x}}_n = \mathbf{g}_n \equiv \bar{\mathbf{g}}_n + \boldsymbol{\varepsilon}, \quad A_n \in \mathbb{R}^{n \times n}, \quad \mathbf{g}_n, \boldsymbol{\varepsilon} \in \mathbb{R}^n, \quad n > 0, \quad (15)$$

During the last twenty years, intense research has been devoted the computation of an approximate solution of (15) by coupling the classic preconditioned Krylov iterative methods with preconditioners P_n able to prevent the propagation of the noisy corrupted components contained in \mathbf{g}_n , i.e.,

$$P_n^{-1} A_n \bar{\mathbf{x}}_n = P_n^{-1} \mathbf{g}_n, \quad (16)$$

where P_n is a *regularizing preconditioner* [2–4, 16]. These can be heuristically intended as preconditioners P_n with a dual role; on the one hand they have to be able to speed up the convergence in the well-conditioned space of the spectra of A_n , and, simultaneously, on the other, they need to slow down the restoration of the most corrupted components of \mathbf{g}_n .

Just to fix ideas in the right order, we stress that this is not the only class of preconditioners used in regularization problems, indeed there exist ample casuistry in which the objective is the same one of classic preconditioning, i.e., accelerating the solution of the underlying linear system and this happens, for example, when looking for the solution of the linear systems arising in the Tikhonov regularization, consider, e.g., [5, 6].

The class of regularizing *matrix algebras preconditioners* P_n is a well studied class of regularizing preconditioners and can be described in a very general setting (see [3, Definition 3.1]).

Such definition is built up mimicking the procedure that is usually followed to produce regularizing preconditioner for matrices A_n of Toeplitz type, i.e.,

$$[A_n(\kappa)]_{r,s} = a_{r-s}, \quad a_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} \kappa(x) \exp(-ikx) dx,$$

for $k \in \mathbb{Z}$ and κ a function in \mathbb{L}^1 with one (or more) root(s). In this case the P_n^{-1} preconditioners are nothing more than matrices generated from a family of bounded functions approximating the unbounded function $1/\kappa$. Specifically, if they are selected from an algebra \mathcal{M}_n of matrices simultaneously diagonalized by an orthogonal transform U_n [17, 18], i.e.,

$$\mathcal{M}_n = \{M_n \in \mathbb{C}^{n \times n} : M_n = U_n \text{diag}(\mathbf{z})U_n^*, \quad \mathbf{z} \in \mathbb{C}^n, \quad U_n^*U_n = I_n\}, \quad (17)$$

then the construction of P_n can be achieved by applying a suitable filter f_α to the diagonal term in the Schur decomposition of some suitably chosen matrix $\bar{M}_n \in \mathcal{M}_n$ (e.g., the projection of A_n). This is indeed nothing more than the computation of a

filtering matrix-function $f_\alpha(\lambda)$ on the particular \overline{M} chosen, since what is then built is simply

$$P_n := f_\alpha(\overline{M}_n) = U_n \operatorname{diag}(f_\alpha(\mathbf{z}))U_n^*. \tag{18}$$

To this class of preconditioners belong all the combinations that can be built taking \mathcal{M}_n as a trigonometric algebra and f_α a suitable filtering function in the sense of [3, Definition 3.1]. The steps needed in this approach include a careful selection of the algebra (17) in which the problem is projected, and the selection of an appropriate filter function; both the choices are strictly connected with the structure of the sequence of matrices $\{A_n\}_n$, and severely effect the quality of the restored solution. An analogous observation holds for all those *regularizing structured* preconditioners [7, 19], in which the preconditioner shares the same structure of the underlying matrix.

As a matter of fact, in many applications, it is not possible to retrace a useful structure in A_n in order to devise a proper matrix algebra \mathcal{M}_n .

The approach we propose in this work, synthesizing from the above presented techniques and from [8, 9], aims to be applied independently from the particular structure of A_n avoiding the necessity of devising an opportune matrix algebra \mathcal{M}_n . In particular, it allows to work with the matrices $\{A_n\}_n$ in a matrix-free framework, i.e., gathering *information* from the matrices just focusing on the matrix-vector product operation (Krylov-type techniques).

2.4 The matrix-function technique

To give a precise idea of the general framework of our approach, if A_n is symmetric and positive definite, we have

$$\begin{aligned} f_\alpha(A_n)\mathbf{g}_n &= Uf_\alpha(\Lambda_{A_n})U^H\mathbf{g}_n = \sum_{j=1}^n f_\alpha(\lambda_j)(\mathbf{u}_j^T\mathbf{g}_n)\mathbf{u}_j \\ &= \sum_{j:\lambda_j \leq \alpha} f_\alpha(\lambda_j)(\mathbf{u}_j^T\mathbf{g}_n)\mathbf{u}_j + \sum_{j:\lambda_j > \alpha} f_\alpha(\lambda_j)(\mathbf{u}_j^T\mathbf{g}_n)\mathbf{u}_j, \end{aligned} \tag{19}$$

where α is a suitable threshold parameter and $f : \mathbb{R}_+ \rightarrow \mathbb{R}$.

Since the eigencomponents related to the $\{j : \lambda_j \leq \alpha\}$ are those responsible for the propagation of the noise contained in \mathbf{g}_n —while the $\{j : \lambda_j > \alpha\}$ are the ones for which it is possible to reconstruct the signal without incurring in a noise propagation phenomenon—we devise the use of a $f_\alpha(\lambda)$ such that

$$f_\alpha(\lambda) \approx h_\alpha(\lambda)\lambda^{-1}, \tag{20}$$

for $h_\alpha(\lambda)$ the Heavyside step function in α as in (5). Accordingly to this choice, we are setting in (19)

$$\begin{aligned} \sum_{j:\lambda_j \leq \alpha} f_\alpha(\lambda_j)(\mathbf{u}_j^T\mathbf{g}_n)\mathbf{u}_j &\approx 0, \\ \sum_{j:\lambda_j > \alpha} f_\alpha(\lambda_j)(\mathbf{u}_j^T\mathbf{g}_n)\mathbf{u}_j &\approx \sum_{j:\lambda_j > \alpha} \frac{1}{\lambda_j}(\mathbf{u}_j^T\mathbf{g}_n)\mathbf{u}_j. \end{aligned}$$

To build such matrix function we need a suitable regular approximation of the Heavy-side step function. This is a problem that has been addressed in a completely different setting for the analysis of electronic structures in quantum chemistry and solid state physics [14, 15]; we use hence the approximation

$$f_\alpha(A_n) = h_\alpha(A_n)A_n^{-1} \approx \frac{1}{2} [1 + \tanh(\beta(A_n - \alpha I_n))] A_n^{-1}, \quad \alpha, \beta > 0. \quad (21)$$

Since we want to avoid any occurrences of the computation of $A_n^{-1}\mathbf{x}$ in this context, for computing $f_\alpha(A_n)\mathbf{g}_n$, we decide to recur to a Krylov subspace method of polynomial type based on either the Lanczos decomposition, if A_n are symmetric matrices, or the Arnoldi decomposition, if the A_n are nonsymmetric.

2.5 Fixing the parameters

As for all regularization methods, we need a suitable way of fixing the various parameters defining the method. In our case, we have to discuss the choice of the α , β for $f_\alpha(\lambda)$.

From Fig. 1, the effects of the two parameters are clear. The value of α regulates which part of the spectrum we are filtering, and the β how sharp is the threshold process. Of course, the choice of α and β should depend on the noise level and on the decay speed of the eigenvalues/singular values of A_n .

A reasonable heuristic for this choice is represented by a value of α that is slightly smaller than the level of noise and by a value of β that is such that $1/\beta \ll \alpha$, in order to avoid the instances represented by the dashed lines in Fig. 1, where some small eigenvalues end up being magnified in the inversion procedure.

Theorem 4 *Given the regularization problem in (1) and the regularizing function $f_\alpha(A_n)$ in (21), when $\delta \rightarrow 0$, for the parameter choice $\alpha \propto \delta$ and $1/\beta < \alpha$, we find that*

1. *If $A_{n,m} = A_n$, then $f_\alpha(A_n)\mathbf{g}_n \rightarrow \bar{\mathbf{g}}_n$,*
2. *otherwise, $f_\alpha(A_{n,m}^T A_{n,m}) A_{n,m}^T \mathbf{g}_n$ converges to the least-square solution of (1).*

Proof We simply need to observe that, by (20) and (5), $f_\alpha(\lambda) \rightarrow \lambda^{-1}$ when $\delta \rightarrow 0$ for the parameter choice $\alpha \propto \delta$ and $1/\beta < \alpha$. □

3 Computing the matrix function

For the sake of readability, from this section on, we fix the dimension of the problem n . We will write A instead A_n , $\bar{\mathbf{x}}$ instead $\bar{\mathbf{x}}_n$ and \mathbf{g} instead \mathbf{g}_n .

The core of our approach is the computation $f_\alpha(A)\mathbf{g}$ and for this task, we will exploit Krylov subspace methods. This section is devoted to a careful review of these techniques.

We have seen in (18) that for diagonalizable matrices, computing $f_\alpha(A)\mathbf{g}$ amounts to the computation of the function f_α on the eigenvalues of the matrix A . This procedure can be defined also in a more general setting [20], for an arbitrary matrix A and

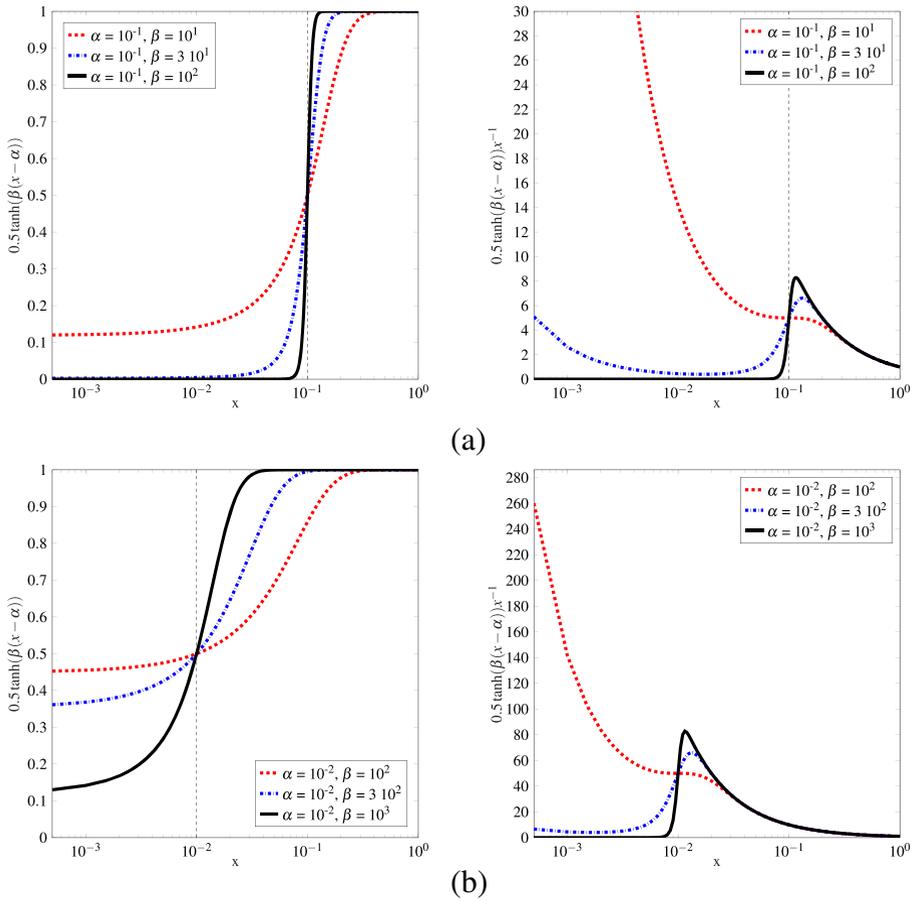


Fig. 1 Regularized version of $h_\alpha(\lambda)$ (left) and $f_\alpha(\lambda)$ function (right) for $\alpha = 10^{-1}, 10^{-2}$ and several values of β

filter function f analytic inside a closed contour Γ enclosing the spectrum of A : the matrix function-vector product $f(A)\mathbf{g}$ can be defined as

$$f(A)\mathbf{g} = \frac{1}{2\pi i} \int_{\Gamma} f(z)(zI - A)^{-1}\mathbf{g} dz. \tag{22}$$

An efficient class of algorithms for computing (22) is represented by projection methods. Let V_k be an orthogonal matrix whose columns $\mathbf{v}_1, \dots, \mathbf{v}_k$ span an arbitrary subspace, then we can approximate $f(A)\mathbf{g}$ on that subspace as

$$\begin{aligned} f(A)\mathbf{g} &= \frac{1}{2\pi i} \int_{\Gamma} f(z)(zI - A)^{-1}\mathbf{g} dz \\ &\approx \frac{1}{2\pi i} \int_{\Gamma} f(z)V_k(zI - V_k^T A V_k)^{-1}V_k^T \mathbf{g} dz \\ &= V_k f(V_k^T A V_k)V_k^T \mathbf{g}. \end{aligned} \tag{23}$$

In this work we are interested in projection methods such that the matrix V_k spans the k th Krylov subspace

$$\mathcal{K}_k(A, \mathbf{g}) = \text{Span}\{\mathbf{g}, A\mathbf{g}, \dots, A^{k-1}\mathbf{g}\}.$$

In the case of generic square matrix the Arnoldi procedure with modified Gram-Schmidt builds an orthonormal basis V_k of $\mathcal{K}_k(A, \mathbf{g})$ satisfying the so-called Arnoldi relation

$$AV_k = V_k H_k + h_{k+1,k} \mathbf{v}_{k+1} \mathbf{e}_k^T, \tag{24}$$

where $H_k = (h_{i,j})_{i,j}$ is an upper Hessenberg matrix of size $k \times k$. Finally, the approximation in (23) is computed as

$$\mathbf{f}_k = \gamma V_k f(H_k) \mathbf{e}_1, \quad \gamma = \|\mathbf{g}\|, \quad \mathbf{e}_1 = (1, 0, \dots, 0)^T \in \mathbb{R}^k.$$

In Algorithm 1 we give a synthetic presentation for the matrix-function-times-vector computation which encompasses a reorthogonalization method.

Algorithm 1 Matrix function computation (Arnoldi–Modified Gram-Schmidt based).

```

Input:  $A \in \mathbb{R}^{n \times n}$ ,  $\mathbf{g} \in \mathbb{R}^n$ ,  $m_{Max}$ , function  $f$ .
Output:  $\mathbf{f}_m \approx f(A)\mathbf{g}$ .
1  $\gamma \leftarrow \|\mathbf{g}\|_2$ ;
2  $\mathbf{v}_1 \leftarrow \mathbf{g}/\gamma$ ;
3 for  $k = 1 : m_{Max}$  do
4    $\mathbf{w}_k \leftarrow A \mathbf{v}_k$ ;                                /* Modified Gram-Schmidt */
5   for  $i = 1 : k$  do
6      $h_{i,k} \leftarrow \langle \mathbf{w}_k, \mathbf{v}_i \rangle$ ;
7      $\mathbf{w}_k \leftarrow \mathbf{w}_k - h_{i,k} \mathbf{v}_i$ ;
8   end
9   for  $i = 1 : k$  do
10     $\tau \leftarrow \langle \mathbf{v}_i, \mathbf{w}_k \rangle$ ;                    /* Reorthogonalization */
11     $\mathbf{w}_k \leftarrow \mathbf{w}_k - \tau \mathbf{v}_i$ ;
12     $h_{i,k} \leftarrow h_{i,k} + \tau$ ;
13  end
14   $h_{k+1,k} \leftarrow \|\mathbf{w}_k\|_2$ ;
15   $V_k = [\mathbf{v}_1 | \dots | \mathbf{v}_k]$ ;
16   $\mathbf{f}_k \leftarrow \gamma V_k f(H_k) \mathbf{e}_1$ ;                /* Schur-Parlett */
17  if  $h_{k+1,k} = 0$  then
18     $\mathbf{f}_m := \mathbf{f}_k$ ;
19    exit;
20  end
21   $\mathbf{v}_{k+1} \leftarrow \mathbf{w}_k/h_{k+1,k}$ ;
22 end

```

When the matrix A is symmetric, Algorithm 1 can be greatly simplified by using the Lanczos procedure to generate the basis of the Krylov subspace $\mathcal{K}_k(A, \mathbf{g})$. By

this procedure we build an orthonormal basis V_k of $\mathcal{K}_k(A, \mathbf{g})$ satisfying a modified version of the Arnoldi relation (24)

$$AV_k = V_k T_k + \alpha_{k+1} \mathbf{v}_{k+1} \mathbf{e}_k^T, \tag{25}$$

where $T_k = \text{diag}(\beta, \alpha, \beta)$ is a symmetric tridiagonal matrix of size $k \times k$. Finally, the approximation in (23) is computed as

$$\mathbf{f}_k = \gamma V_k f(T_k) \mathbf{e}_1, \quad \gamma = \|\mathbf{g}\|, \quad \mathbf{e}_1 = (1, 0, \dots, 0)^T \in \mathbb{R}^k.$$

As for the nonsymmetric case, also the Lanczos algorithm can suffer from a loss of orthogonality in the computed vectors, thus the Algorithm 2 includes also a full reorthogonalization step.

Algorithm 2 Matrix function computation (Lanczos–Based).

```

Input:  $A \in \mathbb{R}^{n \times n}$ ,  $\mathbf{g} \in \mathbb{R}^n$ ,  $m_{Max}$ , function  $f$ .
Output:  $\mathbf{f}_m \approx f(A)\mathbf{g}$ .
1  $\gamma \leftarrow \|\mathbf{g}\|_2$ ;
2  $\mathbf{v}_1 \leftarrow \mathbf{g}/\gamma$ ;
3  $\hat{\mathbf{w}}_1 \leftarrow A\mathbf{v}_1$ ;
4  $\alpha_1 \leftarrow \langle \hat{\mathbf{w}}_1, \mathbf{v}_1 \rangle$ ;
5  $\mathbf{w}_1 = \hat{\mathbf{w}}_1 - \alpha_1 \mathbf{v}_1$ ;
6 for  $k = 2 : m_{Max}$  do
7    $\beta_j \leftarrow \|\mathbf{w}_{j-1}\|_2$ ;
8   if  $\beta_j \neq 0$  then
9      $\mathbf{v}_j = \mathbf{w}_{j-1}$ ;
10     $V_k = [\mathbf{v}_1 | \dots | \mathbf{v}_k]$ ;
11     $\mathbf{f}_k \leftarrow \gamma V_k f(T_k) \mathbf{e}_1$ ;
12  else
13     $\mathbf{f}_m := \mathbf{f}_k$ ;
14    exit;
15  end
16   $\mathbf{v}_k \leftarrow (I_k - V_{k-1} V_{k-1}^T) \mathbf{v}_k$ ;          /* Reorthogonalization */
17   $\hat{\mathbf{w}}_j \leftarrow A\mathbf{v}_j$ ;
18   $\alpha_j \leftarrow \langle \hat{\mathbf{w}}_j, \mathbf{v}_1 \rangle$ ;
19   $\mathbf{w}_j = \hat{\mathbf{w}}_j - \alpha_j \mathbf{v}_j - \beta_j \mathbf{v}_{j-1}$ ;
20 end

```

As a matter of fact, we will use both Algorithm 1 and 2 in an iterative fashion and we need to provide a stopping rule to completely specify our proposal. In the next Section 3.1 we discuss and clarify this issue.

We conclude this section by highlighting a connection between our proposal and the standard approach with hybrid Krylov method. Specifically, we stress that under some hypotheses on the Krylov subspace the methods produce the same iterates.

Proposition 2 *If $V_k f_\alpha(V_k^T A V_k) V_k^T \mathbf{e}_1 \equiv V_k(V_k^T A V_k)^{-1} V_k^T \mathbf{e}_1$ then*

- \mathbf{f}_k coincides with the k th iteration of the CG Algorithm, if A is SPD, and Algorithm 2 is used;
- \mathbf{f}_k coincides with the k th iteration of the CGLS Algorithm, if f_α is computed on $A^T A$, and Algorithm 2 is used;
- \mathbf{f}_k coincides with the k th iteration of the FOM Algorithm, if f_α is computed on A , and Algorithm 1 is used.

Proof All the properties follow straightforwardly from the standard construction of the CG, and CGLS Algorithm with the Lanczos orthogonalization procedure, and of FOM, for the Arnoldi orthogonalization procedure. We refer to [21] for the construction of the relative algorithms. \square

3.1 The stopping criterion

A well-known stopping criterion for regularization is represented by the *discrepancy principle*. If we consider problem (1) and we denote by \mathbf{f}_k an approximation of the solution $\bar{\mathbf{x}}$ obtained in an iterative fashion, the basic idea behind this criterion is to stop the iteration of the chosen method as soon as the norm of the residual $\mathbf{r}_k = \mathbf{g} - A\mathbf{f}_k$ is sufficiently small, typically of the same size of δ , i.e., the norm of the perturbation ε in the right-hand side of (1). In our setting, $\mathbf{f}_k = V_k f_\alpha(V_k^T A V_k) V_k^T \mathbf{g}$, and we can easily monitor the residuals

$$\|\mathbf{r}_k\| = \|\mathbf{g} - A\mathbf{f}_k\| = \|\mathbf{g} - AV_k f_\alpha(V_k^T A V_k) V_k^T \mathbf{g}\|. \tag{26}$$

The discrepancy principle, based on this quantities, reads as

$$\text{select the smallest } k \text{ such that } \|\mathbf{r}_k\|/\|\mathbf{g}\| \leq \frac{\eta \cdot \delta}{\|\mathbf{g}\|}, \tag{27}$$

where $\eta \geq 1$ and we call `NoiseLevel` the relative noise level $\|\boldsymbol{\varepsilon}\|/\|\mathbf{g}\| \equiv \delta/\|\mathbf{g}\|$. Observe now that, since f_α does not coincide, in general, with the function $1/\lambda$, the quantity $\|\mathbf{r}_k\|$ is not guaranteed to converge to 0 as k increases, as it happens in the linear system solution framework. In general, it will *stabilize* when the dimension of the Krylov space increases: when $k \rightarrow n$, \mathbf{f}_k will be an increasingly better approximation of $f(A)\mathbf{g}$ and hence $\|\mathbf{g} - AV_k f_\alpha(V_k^T A V_k) V_k^T \mathbf{g}\|$ will converge to the quantity $\|\mathbf{g} - Af(A)\mathbf{g}\| \neq 0$ since $f(A) \neq A^{-1}$.

In order to improve the stopping capabilities of our regularized reconstruction algorithm, we need to add a further stopping criterion. With this in mind, we consider the sequence of the residuals $\{c_k := \|\mathbf{r}_k\| - \|\mathbf{r}_{k-1}\|\}_k$ which is such that

$$\|\|\mathbf{r}_k\| - \|\mathbf{r}_{k-1}\|\| \leq \|\mathbf{r}_k - \mathbf{r}_{k-1}\| = \|A\mathbf{f}_k - A\mathbf{f}_{k-1}\|, \tag{28}$$

and thus goes to zero as k increases. We select as a stopping criterion the following:

$$\text{select the smallest } k \text{ such that } \|\|\mathbf{r}_k\| - \|\mathbf{r}_{k-1}\|\|/\|\mathbf{g}\| \leq \frac{\eta \cdot \delta}{\|\mathbf{g}\|}. \tag{29}$$

Heuristically this choice is supported by the fact that, due to the noise presence, it is not possible to discern among reconstructions giving rise to residuals which differ

for a quantity of the same order of the noise level. Observe, moreover, that from (28), it is clear that the stopping residual (29) will be satisfied. Finally, we point out that performing the stopping check using the form in (29) is very cheap in space and time once the norm of residuals has been computed.

We stress again that the method we are using generates a solution that is in the Krylov subspace $\mathcal{K}_k(A, \mathbf{b})$, or either $\mathcal{K}_k(A^T A, A^T \mathbf{b})$, for the fixed choice of the parameter α made in Section 2.5. Using analogous techniques to the one in [11–13], it is possible to devise a suitable hybrid choice that selects adaptively an optimal threshold α within the given Krylov space of order k , i.e., we could connect the choice of k and α in an adaptive way.

In the following Algorithm 3 we present the full pseudo-code for our proposal in the case in which the Arnoldi procedure is used; the case for the Lanczos based procedure is obtained similarly from Algorithm 2.

Algorithm 3 Regularization using matrix function.

Input: $A \in \mathbb{R}^{n \times n}$, $\mathbf{g} \in \mathbb{R}^n$, m_{Max} , function f , $\mathbf{f}_0 = 0$, η , NoiseLevel := NL.
Output: \mathbf{x}_m regularized solution of $A\mathbf{x} = \mathbf{g}$.

```

1  $\gamma = \|\mathbf{g}\|$ ;
2  $\mathbf{v}_1 = \mathbf{g}/\gamma$ ;
3 for  $k = 1 : m_{Max}$  do
4    $\mathbf{w}_k \leftarrow A \mathbf{v}_k$ ; /* Modified Gram-Schmidt */
5   for  $i = 1 : k$  do
6      $h_{i,k} \leftarrow \langle \mathbf{w}_k, \mathbf{v}_i \rangle$ ;
7      $\mathbf{w}_k \leftarrow \mathbf{w}_k - h_{i,k} \mathbf{v}_i$ ;
8   end
9   for  $i = 1 : k$  do
10     $\tau \leftarrow \langle \mathbf{v}_i, \mathbf{w}_k \rangle$ ; /* Reorthogonalization */
11     $\mathbf{w}_k \leftarrow \mathbf{w}_k - \tau \mathbf{v}_i$ ;
12     $h_{i,k} \leftarrow h_{i,k} + \tau$ ;
13  end
14   $h_{k+1,k} \leftarrow \|\mathbf{w}_k\|_2$ ;
15   $V_k = [\mathbf{v}_1 | \dots | \mathbf{v}_k]$ ;
16   $\mathbf{x}_k \leftarrow \gamma V_k f(H_k) \mathbf{e}_1$ ; /* Schur-Parlett */
17   $\mathbf{r}_k \leftarrow \mathbf{g} - A \mathbf{x}_k$ ;
18  if  $h_{k+1,k} = 0$  or  $\frac{\|\mathbf{r}_k\|}{\|\mathbf{g}\|} \leq \eta NL$  or  $\frac{\|\|\mathbf{r}_k\| - \|\mathbf{r}_{k-1}\|\|}{\|\mathbf{g}\|} \leq \eta NL$ ; /* Stop Check */
19  then
20     $\mathbf{x}_m := \mathbf{x}_k$ ;
21    exit;
22  end
23   $\mathbf{v}_{k+1} \leftarrow \mathbf{w}_k/h_{k+1,k}$ ;
24 end

```

4 Numerical experiments

All the numerical experiments are performed on a Linux machine with an Intel® Xeon® Platinum 8176 CPU, 2.10 GHz, with 84 Gb of RAM. The code is written and executed in Matlab 9.3.0.713579 (R2017b). The regularization routines used for comparison, and the test problems are generated with the packages AIR Tools II [22] and IR Tools [23]. The algorithm presented here, together with the example files generating the examples, is available as the IRfun MATLAB function on <https://github.com/Cirdans-Home/IRfun>.

In order to study the effectiveness of our stopping criteria, we organize the tests comparing the best achievable PSNR within maximum allotted iterations with the results obtained employing the stopping criteria discussed in Section 3.1. Moreover, some testing on the stability of the algorithm with respect to the choice of the optimization parameter α in $f_\alpha(x)$ is investigated here.

The methods we used for comparison are the standard Krylov methods for regularization implemented in IR Tools [23], i.e., the CGLS, Preconditioned CGLS [4], Range Restricted GMRES (RR-GMRES), the hybrid GMRES, GMRES with ℓ_1 penalty [24], and hybrid LSQR method [25]. Moreover, we consider also the solution with the Tikhonov regularization method in which we solve the auxiliary linear system with the CGLS method and compute the optimal parameter λ by means of the L-curve criterion using of the SVD of the matrix A . We report, moreover, the computational time for every method. To conclude, let us stress the fact that our proposal is a linear method exploiting the information from Krylov subspaces: for this reason, in the comparisons, we restrict ourselves only to the linear methods mentioned above not taking into account nonlinear techniques.

4.1 Deblurring problems: $f_\alpha(A)$ for A symmetric

In all the examples in this section, the parameters α , and β for the matrix function regularization are selected as $\alpha = \delta \times 10^{-1}$, and $\beta = 10^9$. We perform a fixed number of iterations (100) for each method, independently from the fact that a stopping criterion is satisfied or not; for all the comparison methods the discrepancy principle is used to pick the k at which the iterations are halted, while for our proposal we use the stopping criteria described in Section 3.1, i.e., the minimum between the two that satisfy the (27) and (29). The parameter λ for the Tikhonov method is set by using the SVD/L-curve criterion and the CGLS method is used to obtain the solution of the resulting linear system. We report in every table, both, the number of iterations and the achieved PSNR: the ones in brackets are relative to the best-reconstructed solution while the others, correspond to the results obtained by applying the stopping criteria. For the Tikhonov-CGLS method the two quantities coincide since the regularization is obtained by the shift λ , and the linear system is then solved to the highest accuracy. For the preconditioned CGLS we use the algebra preconditioner $P_{(i)}(A) := \mathcal{L}_{A^{2i}}^{-1} \mathcal{L}_A^{-1}$ introduced by the authors in [4], where \mathcal{L}_A is the projection

of the matrix A on the space $\text{sd } F$ of matrices simultaneously diagonalized by the two-level Fourier transform

$$\mathcal{L} := \text{sd } F = \{F \text{diag}(\mathbf{z})F^* : \mathbf{z} \in \mathbb{C}^n\}.$$

Specifically, we always test the various preconditioners obtained for $i = 1, \dots, 8$, and take the one giving the best results. Observe that in this way we always consider also the classic optimal and super-optimal preconditioners (see [4, 17]).

Satellite The first example is the ‘satellite’ image (Fig. 2) with a mild, medium, and severe Gaussian blur generated by the `PRblurgauss()` function. Since the images terminate with a zero boundary, we have used the zero Dirichlet boundary conditions to assemble the matrix A . To the right-hand side we add Gaussian noise of level

$$\delta = \{10^{-3}, 10^{-2}, 10^{-1}, 2 \times 10^{-1}, 3 \times 10^{-1}, 5 \times 10^{-1}\} \quad (30)$$

by means of the function `PRnoise(b, ‘gauss’, delta)`.

For the sake of completeness, for this particular problem set, we compare our approach using as preconditioner for the CGLS method an approximate inverse Toeplitz preconditioner proposed in [19]. To have a fair comparison with the CGLS and the PCGLS methods we apply the `fftPrec()` from the Regularization Tools [26] package to the CGLS method (the associate threshold for `fftPrec()` has been set using the generalized cross validation method). In the following, this method will be denoted as `fftPCGLS`.

The results are collected in Tables 1, if reorthogonalization is used for the Lanczos method, and in Table 2 without reorthogonalization.

An example of the reconstructed ‘satellite’ by the various algorithms is instead given in Fig. 3.

The first observation we can make on these cases is that the stopping criteria work efficiently on all the test cases. As a matter of fact, the PSNR for the best-reconstructed signal, and the one obtained from the stopping are comparable.

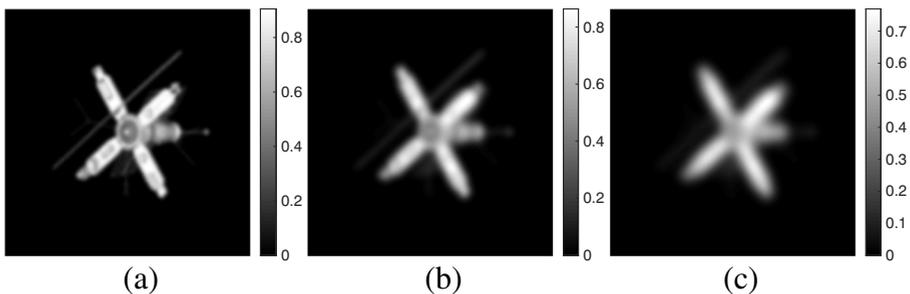


Fig. 2 ‘satellite’ test problem with different Gaussian blur generated by the `PRblurgauss()` function. **a** mild. **b** medium. **c** severe

Table 1 'satellite' test problem, three types of Gaussian blur (mild, medium, severe), different levels of noise δ . The parameter for the Tikhonov method is set by using the SVD/L-curve criterion, the parameter $\beta = 10^9$

(a) mild											
δ	CGLS			PCGLS			RR-GMRES				
	IT	T(s)	PSNR	$P_{\epsilon_1}(A)$	IT	T(s)	PSNR	$f_{\alpha}(A)\mathbf{g}_n$	IT	T(s)	PSNR
1.0e-03	100 (100)	2.0e+00	28.92 (28.92)	2	2 (74)	1.6e-01	27.90 (29.23)	25 (41)	4.2e-01	29.05 (29.39)	
1.0e-02	20 (44)	4.7e-01	27.65 (28.10)	4	2 (176)	1.3e-01	25.70 (28.03)	9 (14)	1.0e-01	27.69 (28.12)	
1.0e-01	4 (7)	1.9e-01	25.77 (26.10)	3	1 (3)	3.6e-02	26.03 (26.15)	3 (4)	4.6e-02	25.78 (26.08)	
2.0e-01	2 (4)	4.4e-02	24.70 (25.32)	3	1 (2)	1.0e-01	25.45 (25.45)	2 (3)	3.4e-02	24.92 (25.33)	
3.0e-01	2 (3)	4.3e-02	24.52 (24.75)	4	1 (2)	3.7e-02	24.76 (24.76)	2 (2)	4.1e-02	24.66 (24.66)	
5.0e-01	1 (2)	7.5e-02	23.28 (24.06)	4	1 (2)	1.1e-01	24.25 (24.25)	1 (2)	7.7e-02	23.28 (24.01)	
fftPCGLS											
δ	Tikhonov (CGLS)			$f_{\alpha}(A)\mathbf{g}_n$							
	IT	T(s)	PSNR	λ	IT	T(s)	PSNR	α	IT	T(s)	PSNR
1.0e-03	100 (89)	2.4e+00	26.36 (26.87)	9.06e-04	100	5.2e+00	28.92	1.0e-04	39 (38)	2.6e+00	29.36 (29.37)
1.0e-02	71 (10)	1.8e+00	21.64 (26.69)	9.33e-03	100	5.7e+00	27.49	1.0e-03	19 (14)	3.4e-01	27.24 (28.09)
1.0e-01	17 (1)	5.9e-01	16.67 (24.26)	7.97e-02	100	5.7e+00	25.32	1.0e-02	7 (5)	1.1e-01	25.06 (26.08)
2.0e-01	12 (1)	4.4e-01	15.24 (22.71)	1.40e-01	88	4.7e+00	24.52	2.0e-02	8 (4)	1.2e-01	17.50 (25.28)
3.0e-01	11 (1)	4.8e-01	13.06 (20.57)	1.89e-01	67	3.6e+00	23.85	3.0e-02	9 (3)	4.2e-01	11.23 (24.65)
5.0e-01	9 (1)	4.3e-01	10.80 (17.69)	2.63e-01	50	1.4e+00	22.92	5.0e-02	9 (3)	1.4e-01	6.62 (23.88)

Table 1 (continued)

δ	hybrid GMRES			hybrid-FGMRES			hybrid L-SQR		
	IT	T(s)	PSNR	IT	T(s)	PSNR	IT	T(s)	PSNR
1.0e-03	16 (32)	2.9e-01	28.81 (29.02)	19 (22)	3.1e-01	29.56 (29.61)	95 (100)	4.6e+00	28.78 (28.80)
1.0e-02	12 (14)	1.8e-01	27.72 (27.75)	9 (7)	2.9e-01	27.92 (28.08)	20 (100)	4.2e-01	27.50 (28.01)
1.0e-01	8 (6)	1.3e-01	25.80 (25.86)	4 (2)	1.4e-01	22.06 (24.75)	56 (23)	1.7e+00	25.99 (26.09)
2.0e-01	7 (4)	1.1e-01	24.62 (24.89)	3 (1)	1.1e-01	18.15 (23.21)	39 (14)	8.0e-01	24.97 (25.15)
3.0e-01	7 (3)	1.4e-01	23.40 (23.96)	2 (1)	8.9e-02	18.24 (21.71)	2 (11)	1.0e-01	24.35 (24.40)
5.0e-01	3 (2)	7.3e-02	21.88 (22.16)	2 (1)	2.8e-02	14.17 (18.89)	2 (2)	2.8e-01	23.50 (23.50)

(b) medium

δ	CGLS			PCGLS			RR-GMRES			
	IT	T(s)	PSNR	$P_{(i)}(A)$	IT	T(s)	PSNR	IT	T(s)	PSNR
1.0e-03	100 (100)	1.9e+00	25.27 (25.27)	1	3 (21)	1.9e-01	25.37 (25.84)	36 (68)	1.1e+00	25.60 (25.93)
1.0e-02	37 (91)	1.5e+00	24.87 (25.11)	3	2 (158)	1.4e-01	23.11 (25.06)	14 (22)	1.5e-01	24.95 (25.13)
1.0e-01	6 (17)	9.7e-02	23.13 (23.84)	2	1 (3)	3.6e-02	23.73 (23.90)	5 (8)	1.6e-01	23.43 (23.85)
2.0e-01	3 (9)	1.5e-01	22.49 (23.15)	3	1 (5)	3.7e-02	22.66 (23.16)	3 (5)	4.6e-02	22.69 (23.16)
3.0e-01	2 (6)	4.1e-02	22.13 (22.76)	3	1 (3)	3.9e-02	22.60 (22.77)	2 (4)	3.4e-02	22.27 (22.78)
5.0e-01	1 (4)	3.7e-02	20.79 (22.36)	3	1 (2)	3.4e-02	22.37 (22.37)	1 (3)	6.5e-02	20.79 (22.38)

Table 1 (continued)

δ	fifPCGLS			Tikhonov (CGLS)			$f_\alpha(A)g_n$				
	IT	T(s)	PSNR	λ	IT	T(s)	PSNR	α	IT	T(s)	PSNR
1.0e-03	100 (6)	2.3e+00	20.61 (22.72)	8.67e-04	100	5.3e+00	25.27	1.0e-04	37 (60)	1.0e+00	25.66 (25.92)
1.0e-02	100 (25)	2.3e+00	20.46 (23.53)	8.38e-03	100	3.9e+00	25.12	1.0e-03	22 (20)	4.1e-01	25.11 (25.12)
1.0e-01	17 (3)	5.8e-01	19.06 (21.80)	6.96e-02	100	5.1e+00	23.88	1.0e-02	7 (8)	2.8e-01	23.71 (23.81)
2.0e-01	8 (1)	3.9e-01	19.05 (21.03)	1.21e-01	98	5.4e+00	23.19	2.0e-02	6 (6)	8.8e-02	23.13 (23.13)
3.0e-01	5 (1)	3.1e-01	18.58 (19.87)	1.62e-01	76	4.3e+00	22.79	3.0e-02	6 (5)	8.8e-02	22.32 (22.74)
5.0e-01	4 (1)	3.0e-01	16.55 (17.67)	2.19e-01	58	1.5e+00	22.27	5.0e-02	6 (4)	9.0e-02	19.99 (22.35)
δ	hybrid GMRES			hybrid-FGMRES			hybrid LSQR				
	IT	T(s)	PSNR	IT	T(s)	PSNR	IT	T(s)	IT	PSNR	
1.0e-03	16 (35)	1.7e-01	25.29 (25.39)	20 (25)	5.9e-01	25.85 (25.94)	100 (100)	6.0e+00	25.22 (25.22)		
1.0e-02	14 (14)	2.0e-01	24.57 (24.57)	9 (7)	9.2e-02	24.74 (24.81)	25 (100)	1.3e+00	24.36 (24.73)		
1.0e-01	8 (6)	1.3e-01	22.95 (22.99)	4 (2)	1.2e-01	20.99 (22.45)	7 (100)	1.9e-01	22.88 (23.81)		
2.0e-01	7 (4)	1.2e-01	22.12 (22.33)	3 (1)	9.7e-02	18.12 (21.19)	81 (84)	2.4e+00	23.17 (23.17)		
3.0e-01	7 (3)	1.2e-01	21.29 (21.78)	2 (1)	7.4e-02	17.85 (20.30)	2 (65)	1.0e-01	21.99 (22.79)		
5.0e-01	3 (3)	1.9e-01	20.36 (20.36)	2 (1)	2.9e-02	14.17 (18.28)	2 (63)	3.0e-01	21.54 (22.27)		

Table 1 (continued)

(c) severe											
δ	CGLS			PCGLS			RR-GMRES			PSNR	T(s)
	IT	T(s)	PSNR	$P_{(i)}(A)$	IT	T(s)	PSNR	IT	T(s)		
1.0e-03	100 (100)	2.2e+00	23.26 (23.26)	1	2 (52)	1.6e-01	23.19 (24.25)	49 (92)	1.3e+00	24.05 (24.47)	
1.0e-02	47 (100)	1.2e+00	22.88 (23.20)	2	2 (94)	1.4e-01	22.38 (23.37)	16 (32)	4.2e-01	22.95 (23.39)	
1.0e-01	7 (22)	3.0e-01	22.08 (22.40)	2	1 (4)	3.6e-02	22.10 (22.33)	5 (10)	1.6e-01	22.14 (22.41)	
2.0e-01	4 (11)	6.5e-02	21.68 (22.14)	5	1 (35)	3.6e-02	19.52 (22.10)	3 (6)	1.2e-01	21.68 (22.15)	
3.0e-01	3 (8)	5.4e-02	21.38 (21.99)	6	1 (28)	1.1e-01	18.82 (21.96)	2 (5)	3.5e-02	20.97 (21.99)	
5.0e-01	1 (6)	2.5e-02	19.16 (21.76)	5	1 (13)	1.1e-01	19.52 (21.76)	1 (4)	2.8e-02	19.16 (21.75)	
ffitPCGLS											
δ	IT	T(s)	PSNR	Tikhonov (CGLS)			$f_{\alpha}(A)g_n$	IT	T(s)	PSNR	
				λ	IT	T(s)	α				
1.0e-03	100 (8)	2.3e+00	10.81 (20.57)	7.32e-04	100	5.4e+00	1.0e-04	12 (83)	1.9e-01	23.44 (24.46)	
1.0e-02	100 (39)	2.3e+00	19.07 (20.18)	7.66e-03	100	2.5e+00	1.0e-03	26 (29)	1.3e+00	23.33 (23.37)	
1.0e-01	17 (6)	5.8e-01	19.69 (20.83)	5.89e-02	100	2.9e+00	1.0e-02	7 (10)	2.9e-01	22.27 (22.40)	
2.0e-01	10 (3)	4.6e-01	18.90 (19.65)	1.02e-01	100	3.6e+00	2.0e-02	7 (7)	2.8e-01	22.15 (22.15)	
3.0e-01	8 (1)	4.0e-01	16.78 (18.29)	1.38e-01	84	3.9e+00	3.0e-02	6 (6)	8.8e-02	21.97 (21.97)	
5.0e-01	5 (1)	3.3e-01	15.78 (16.97)	1.92e-01	63	3.3e+00	5.0e-02	6 (5)	2.4e-01	21.48 (21.73)	

Table 1 (continued)

δ	hybrid GMRES			hybrid-FGMRES			hybrid LSQR		
	IT	T(s)	PSNR	IT	T(s)	PSNR	IT	T(s)	PSNR
1.0e-03	18 (37)	2.1e-01	23.38 (23.45)	19 (100)	6.0e-01	24.14 (24.83)	100 (100)	6.0e+00	23.15 (23.15)
1.0e-02	14 (15)	5.1e-01	22.64 (22.64)	9 (9)	9.4e-02	23.08 (23.08)	23 (100)	6.1e-01	22.52 (22.75)
1.0e-01	8 (6)	3.3e-01	21.79 (21.88)	4 (2)	4.8e-02	19.95 (21.34)	7 (100)	4.9e-01	21.81 (22.35)
2.0e-01	5 (4)	2.5e-01	21.18 (21.22)	4 (1)	4.6e-02	15.46 (19.98)	100 (100)	2.6e+00	22.13 (22.13)
3.0e-01	7 (4)	1.2e-01	20.06 (20.52)	3 (1)	1.1e-01	14.36 (19.40)	2 (97)	1.1e-01	20.55 (21.94)
5.0e-01	8 (3)	3.2e-01	18.32 (19.20)	2 (1)	2.9e-02	14.36 (17.86)	2 (75)	1.0e-01	20.22 (21.66)

Table 2 ‘satellite’ test problem, three types of Gaussian blur (mild, medium, severe), different levels of noise δ without reorthogonalization (i.e., $\text{Reorth} = \text{‘off’}$). To be compared with the results in Table 1

		$f_{\alpha}(A)g_n$											
		mild				medium				severe			
δ	α	IT	T(s)	PSNR	IT	T(s)	PSNR	IT	T(s)	PSNR	IT	T(s)	PSNR
1.0e-03	1.0e-04	40 (39)	1.1e+00	29.35 (29.36)	35 (61)	8.3e-01	25.62 (25.92)	12 (82)	2.2e-01	23.43 (24.49)			
1.0e-02	1.0e-03	14 (14)	3.3e-01	28.06 (28.06)	14 (20)	2.6e-01	24.95 (25.10)	14 (29)	2.6e-01	22.87 (23.38)			
1.0e-01	1.0e-02	6 (5)	1.4e-01	25.87 (26.09)	6 (8)	1.0e-01	23.50 (23.80)	6 (9)	1.2e-01	22.17 (22.39)			
2.0e-01	2.0e-02	5 (4)	1.0e-01	24.49 (25.29)	5 (6)	8.7e-02	23.05 (23.15)	5 (7)	9.9e-02	21.98 (22.13)			
3.0e-01	3.0e-02	5 (3)	9.2e-02	21.84 (24.64)	4 (5)	6.4e-02	22.64 (22.77)	5 (6)	8.9e-02	21.93 (21.99)			
5.0e-01	5.0e-02	4 (3)	8.1e-02	21.47 (23.75)	4 (4)	6.6e-02	22.43 (22.43)	4 (5)	6.5e-02	21.61 (21.72)			

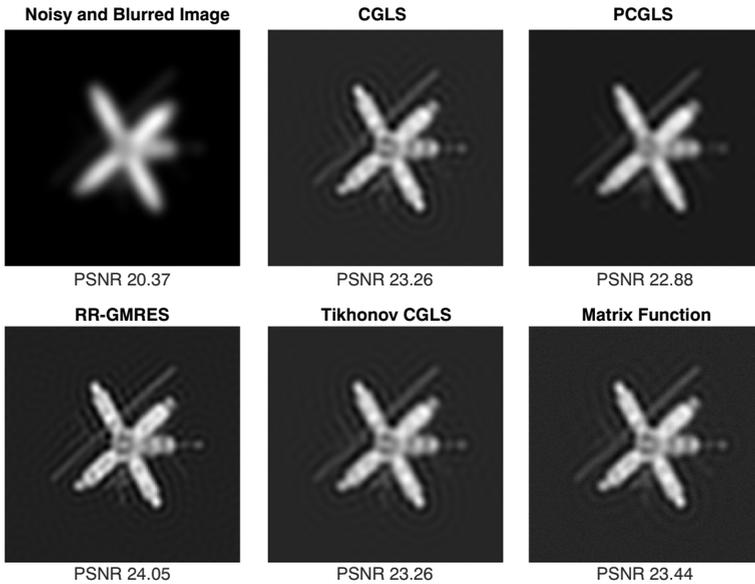


Fig. 3 Reconstructed signals for the ‘satellite’ test problem, for $\delta = 5e - 1$ and medium level of blur

For high levels of noise ($\delta \geq 2e - 1$) and severe blur level the combination of the matrix function routine and the stopping criteria deliver better results than the comparison methods in the majority of the cases. In particular, in the comparison with the hybrid Krylov methods, we should mention the fact that, in some cases, the latter achieves slightly better results in terms of combination of PSNR/stopping results but at the cost of having a higher execution time. Generally, we can observe that the timings of our proposal are smaller than the one needed for solving the problem with the Tikhonov approach and are comparable with the ones for the other Krylov-type methods. There is some improvement on the timings when no reorthogonalization is used at the cost of a slightly minor performance in terms of PSNR.

We are also interested in investigating the sensitiveness of the proposed approach with respect to the regularization parameters α and β of Section 2.5 in terms of achieved PSNR. In Fig. 4 we report the PSNR of the best reconstruction in 100 iterations obtained by the matrix function iterative regularization with the function $f_\alpha(A)$, and varying the parameter α around the selected value in Section 2.5, i.e., $\alpha = \delta \times 10^{-1}$.

We repeat the test for all the blur levels in Fig. 2 and the noise levels δ in (30). We report, moreover, the same stability test for β . What we observe is that the selected parameters lay, always, in a flat zone of the graph. This implies that even if we slightly vary them the resulting restoration quality is unchanged.

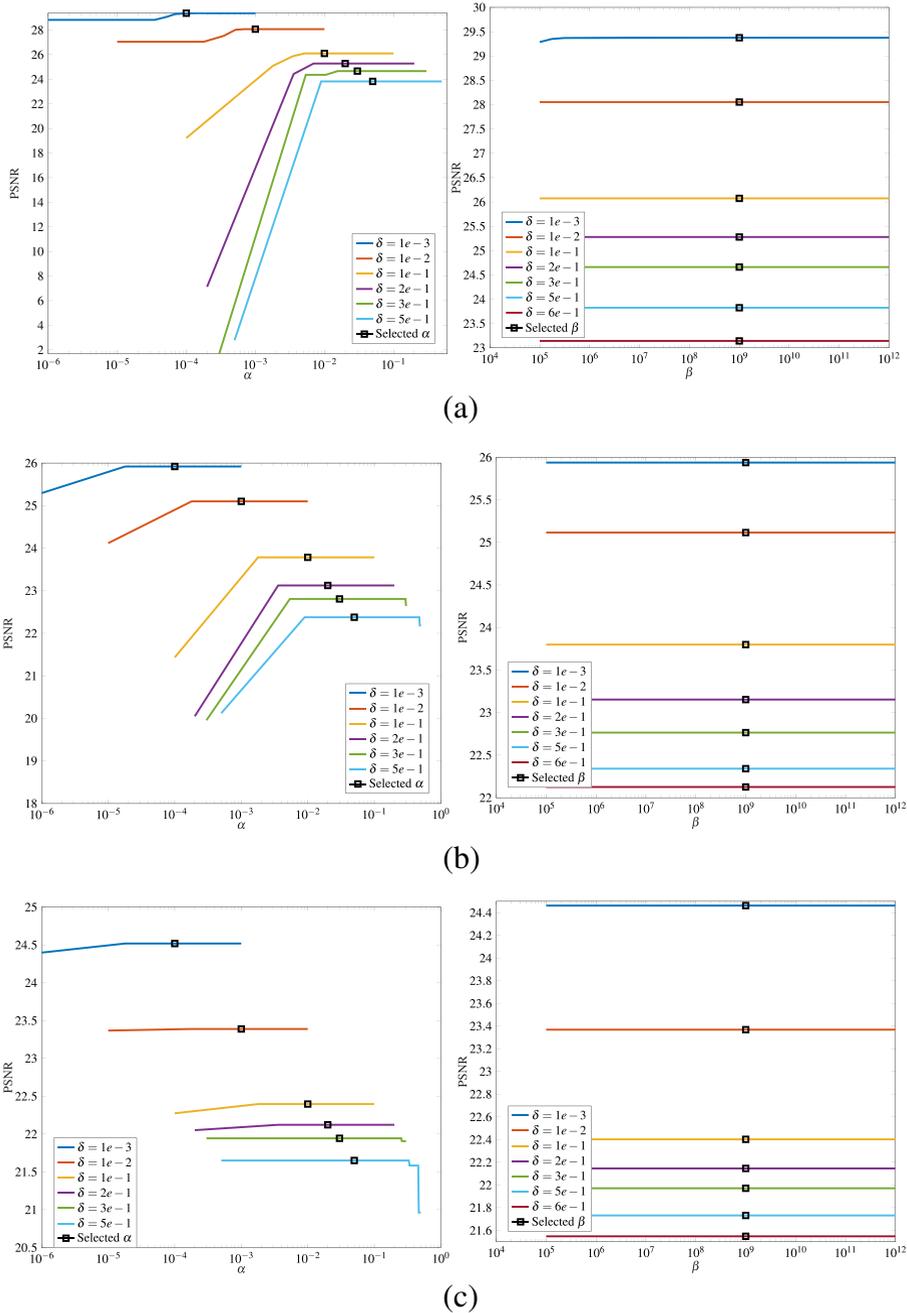


Fig. 4 Sensibility with respect to the choice of α and β for the matrix-function regularization algorithm based on the Lanczos procedure with reorthogonalization for the ‘satellite’ test problem. The sensibility with respect to β is evaluated for the value of α selected for the experiments. **a** mild. **b** medium. **c** severe

4.2 Space variant problems: $f_\alpha(A)$

We consider also the regularization of problems whose solution is not spatially invariant, namely we consider the first-kind Fredholm integral equation from Phillips [27]. This problem defines the function

$$\phi(x) = \begin{cases} 1 + \cos(x\pi/3), & |x| < 3, \\ 0, & |x| \geq 3, \end{cases}$$

and tries to recover it by means of the kernel $K(s, t) = \phi(s - t)$, with right-hand side $g(s) = (6 - |s|)(1 + .5 \cos(s\pi/3)) + 9/(2\pi) \sin(|s|\pi/3)$ on the interval $[-6, 6]$. The second problem of this class we decide to investigate is the discretization of 1D gravity surveying model problem [28] in which a mass distribution $f(t) = \sin(\pi t) + 0.5 \sin(2\pi t)$, is located at depth $d = 0.25$, while the vertical component of the gravity field $g(s)$ is measured at the surface. The resulting problem is again a first-kind Fredholm integral equation, this time with kernel $K(s, t) = d(d^2 + (s - t)^2)^{-\frac{3}{2}}$, in which the discretization of the source $g(t)$ is obtained as $\mathbf{g} = A\mathbf{x}$, where A is computed by means of a mid-point quadrature rule on the interval $[0, 1]$. We perturb again the right-hand side by the noise in (30), and give the solution in Table 3.

The “–” reported for the RR-GMRES algorithm in Table 3b occurs when the algorithm generates a Hessenberg matrix that is numerically singular, and thus halts without giving back a result.

In Fig. 5 we report the PSNR of the best reconstruction in 100 iteration obtained by the matrix function iterative regularization with the function $f_\alpha(A)$, and varying the parameter α around the selected value in Section 2.5, i.e., $\alpha = \delta \times 10^{-1}$, from which we observe again that the selected parameter is located in a flat zone, i.e., small variations do not alter the resulting PSNR for the best reconstruction.

4.3 Tomography problem: $f(A_{n,m}^T A_{n,m})$ for $A_{n,m}$ rectangular

Tomography problems are imaging problems in which the image has to be reconstructed from some of its sections obtained through the use of a penetrating wave, and it literally means a “slice view”. In general, we could expect to have to solve a problem of the form

$$\text{find } \mathbf{x}_m \in \mathbb{R}^m \text{ s.t. } A_{n,m}\mathbf{x}_m = \mathbf{g}_n, \quad n \neq m, \quad A_{n,m} \in \mathbb{R}^{n \times m}, \quad \mathbf{g}_n \in \mathbb{R}^n. \quad (31)$$

Using the least-square interpretation of problem (31), we can then compute our regularized solution as

$$\mathbf{x}_m = f_\alpha(A_{n,m}^T A_{n,m})A_{n,m}^T \mathbf{g}_n,$$

for the f_α given in (21) by means of the matrix function algorithm based on the Lanczos orthogonalization process.

Parallel tomography we consider the “line model” for creating a 2D X-ray tomography test problem with an $N \times N$ pixel domain, using p parallel rays

Table 3 Space variant problems: Phillips and Gravity for different levels of noise δ in (30), $\eta = 1.01$, and using reorthogonalization

(a) Phillips											
δ	CGLS		PCGLS		RR-GMRES		PSNR		T(s)		PSNR
	IT	$P_{(i)}(A)$	IT	$P_{(i)}(A)$	IT	$P_{(i)}(A)$	PSNR	IT	PSNR	T(s)	
1.0e-03	9 (10)	7.5e-02	62.52 (62.95)	7	6 (25)	1.1e-02	54.03 (66.36)	7 (10)	2.8e-02	60.71 (62.77)	
1.0e-02	5 (7)	1.2e-02	52.62 (52.84)	1	3 (6)	6.4e-03	52.43 (53.84)	4 (7)	1.5e-02	52.86 (54.62)	
1.0e-01	3 (5)	1.2e-02	41.62 (51.28)	1	2 (5)	4.7e-03	48.52 (52.94)	3 (4)	1.4e-02	42.49 (51.52)	
2.0e-01	3 (5)	1.1e-02	41.51 (49.02)	3	2 (6)	4.8e-03	37.44 (52.35)	3 (4)	9.2e-03	42.31 (48.62)	
3.0e-01	2 (6)	1.0e-02	34.63 (49.54)	2	1 (6)	3.0e-03	36.06 (49.56)	2 (4)	6.5e-03	35.50 (47.82)	
5.0e-01	1 (6)	9.0e-03	29.89 (45.13)	2	1 (5)	2.9e-03	36.77 (45.70)	1 (4)	6.4e-03	29.89 (44.76)	
6.0e-01	1 (3)	5.4e-03	29.74 (38.63)	2	1 (7)	2.9e-03	35.44 (39.47)	1 (3)	6.5e-03	29.74 (38.60)	
(b) Gravity											
δ	Tikhonov (CGLS)		$f_{\alpha}(A)g_n$		PSNR		T(s)		IT		PSNR
	λ	IT	T(s)	IT	PSNR	IT	T(s)	IT	PSNR		
1.0e-03	6.51e-03	27	2.8e+00	1.0e-04	46.16	14 (14)	5.5e-02	53.61 (53.61)	53.61 (53.61)		
1.0e-02	5.57e-02	17	4.9e-01	1.0e-03	47.54	6 (7)	1.2e-02	53.02 (53.76)	53.02 (53.76)		
1.0e-01	2.93e-01	13	4.8e-01	1.0e-02	51.32	6 (5)	9.9e-03	50.83 (50.87)	50.83 (50.87)		
2.0e-01	5.02e-01	13	5.1e-01	2.0e-02	46.72	5 (8)	6.4e-03	48.19 (50.20)	48.19 (50.20)		
3.0e-01	6.57e-01	12	4.5e-01	3.0e-02	44.74	5 (8)	1.0e-02	48.19 (48.32)	48.19 (48.32)		
5.0e-01	8.29e-01	12	4.9e-01	5.0e-02	43.41	5 (5)	5.4e-03	46.17 (46.17)	46.17 (46.17)		
6.0e-01	9.30e-01	11	4.3e-01	6.0e-02	37.82	5 (4)	6.5e-03	37.44 (38.60)	37.44 (38.60)		

Table 3 (continued)

δ	Hybrid GMRES			Hybrid-FGMRES			Hybrid LSQR		
	IT	T(s)	PSNR	IT	T(s)	PSNR	IT	T(s)	PSNR
1.0e-03	6 (100)	6.4e-02	53.77 (62.31)	4 (14)	2.7e-02	46.52 (56.41)	6 (46)	4.7e-02	52.86 (62.36)
1.0e-02	7 (100)	3.0e-02	52.58 (55.29)	3 (6)	1.7e-02	37.19 (52.07)	7 (29)	4.2e-02	52.40 (55.29)
1.0e-01	100 (100)	1.6e-01	51.49 (51.49)	3 (6)	1.5e-02	25.84 (40.94)	22 (44)	5.7e-02	50.28 (51.67)
2.0e-01	100 (100)	1.7e-01	46.14 (46.14)	3 (5)	1.2e-02	21.19 (37.29)	34 (46)	6.8e-02	46.22 (46.45)
3.0e-01	100 (100)	1.5e-01	43.38 (43.38)	3 (5)	9.9e-03	16.32 (34.12)	2 (27)	1.9e-02	34.09 (43.42)
5.0e-01	100 (100)	1.6e-01	40.71 (40.71)	2 (10)	1.2e-02	15.44 (28.91)	2 (34)	1.8e-02	32.63 (42.00)
6.0e-01	3 (100)	1.5e-02	27.75 (37.09)	2 (10)	6.7e-03	14.13 (26.75)	2 (32)	2.0e-02	32.14 (37.54)

(b) Gravity										
δ	CGLS			PCGLS			RR-GMRES			
	IT	T(s)	PSNR	$P_0(A)$	IT	T(s)	PSNR	IT	T(s)	PSNR
1.0e-03	8 (10)	1.4e-02	37.90 (46.70)	3	5 (17)	1.0e-02	30.07 (38.33)	-	-	-
1.0e-02	6 (8)	8.8e-03	32.16 (37.52)	6	3 (12)	9.0e-03	22.95 (35.03)	-	-	-
1.0e-01	4 (5)	8.0e-03	25.94 (26.03)	2	2 (6)	5.1e-03	17.88 (31.33)	-	-	-
2.0e-01	2 (4)	5.9e-03	16.38 (25.02)	2	1 (6)	4.4e-03	16.27 (26.40)	-	-	-
3.0e-01	2 (5)	7.7e-03	16.88 (24.01)	2	1 (5)	3.5e-03	15.82 (26.16)	-	-	-
5.0e-01	1 (4)	5.6e-03	11.54 (24.99)	3	2 (4)	5.2e-03	20.84 (23.06)	-	-	-
6.0e-01	1 (4)	4.9e-03	11.70 (24.63)	2	1 (5)	3.9e-03	16.14 (28.69)	-	-	-

Table 3 (continued)

δ	Tikhonov (CGLS)			$f_\alpha(A)g_n$					
	λ	IT	T(s)	PSNR	α	IT	T(s)	PSNR	
1.0e-03	5.87e-03	15	3.5e-01	35.01	1.0e-04	10 (10)	1.2e-02	43.68 (43.68)	
1.0e-02	4.70e-02	13	3.7e-01	39.54	1.0e-03	9 (9)	1.2e-02	38.67 (38.67)	
1.0e-01	3.81e-01	11	3.8e-01	25.11	1.0e-02	6 (5)	6.9e-03	24.13 (26.58)	
2.0e-01	6.48e-01	10	3.6e-01	24.29	2.0e-02	5 (5)	5.1e-03	25.82 (25.82)	
3.0e-01	6.69e-01	10	3.9e-01	21.22	3.0e-02	5 (5)	5.5e-03	22.19 (22.19)	
5.0e-01	1.13e+00	9	3.6e-01	19.96	5.0e-02	5 (4)	9.2e-03	21.17 (21.18)	
6.0e-01	1.15e+00	9	3.5e-01	21.13	6.0e-02	5 (5)	6.4e-03	22.63 (22.63)	
δ	hybrid GMRES			hybrid-FGMRES			hybrid L-SQR		
	IT	T(s)	PSNR	IT	T(s)	PSNR	IT	T(s)	PSNR
	17 (45)	2.7e-02	37.51 (39.45)	7 (15)	1.5e-02	23.42 (33.83)	17 (100)	4.3e-02	38.15 (42.68)
	63 (47)	8.3e-02	11.44 (34.40)	5 (10)	8.4e-03	15.01 (32.12)	22 (93)	4.6e-02	35.07 (36.53)
	67 (47)	9.6e-02	11.24 (25.85)	3 (6)	7.9e-03	6.89 (19.39)	21 (23)	4.6e-02	26.19 (26.19)
	60 (45)	8.0e-02	11.72 (21.83)	3 (9)	1.1e-02	-1.20 (18.55)	2 (99)	1.6e-02	16.18 (24.94)
	61 (44)	9.4e-02	11.10 (18.83)	3 (10)	7.0e-03	-6.83 (15.95)	18 (94)	5.7e-02	20.03 (21.06)
	62 (53)	8.6e-02	11.47 (17.81)	2 (11)	6.1e-03	-1.76 (11.90)	21 (92)	5.3e-02	19.98 (21.11)
	60 (47)	8.0e-02	11.30 (17.59)	2 (9)	5.5e-03	-4.47 (12.22)	21 (35)	5.0e-02	21.18 (21.86)

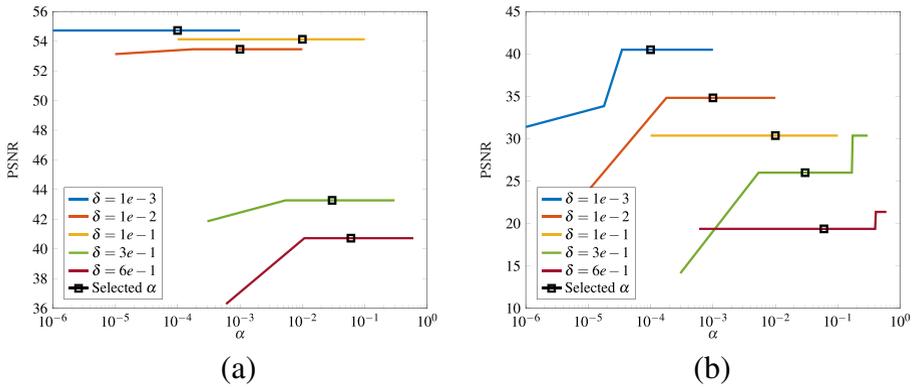


Fig. 5 Sensibility with respect to the choice of the α parameter for the matrix-function regularization algorithm with reorthogonalization for the ‘Phillips’ and ‘Gravity’ test problems

for each angle θ generated by the `paralleltomo` routine from the AIR-TOLSII package. To the right-hand side we add Gaussian noise of level δ as in (30). In this example the maximum number of iterations is fixed at 800. The results are collected in Table 4. The relaxation parameters for the Kaczmarz, the random Kaczmarz, Cimmino, and Landweber methods are set with the training routine `train_relaxpar(A,b,x_true,@method,20)`, while the threshold parameter τ is computed with `train_dpme(A,b1,x_true,@method,'DP',NoiseLevel,10,20,options)`.

On this set of test problems, we observe different behaviors for the considered methods allowing us to observe a couple of interesting facts. Firstly, we observe that for higher levels of noise the discrepancy principle combined with the Kaczmarz, random Kaczmarz, Landweber and Cimmino methods do not work well, i.e., the iteration at which it stops the method is very far from the one for which the best PSNR is achieved even though the discrepancy principle gives better results when combined with the standard CGLS algorithm.

Secondly, it is interesting to compare the best achievable reconstructions obtained by CGLS and by our proposal: the best obtained PSNR is the same in the two cases. This is due to the fact that the ill-conditioning of the matrices $A_{n,m}$ obtained for this test problem is the result of very large singular values and not to the presence of decaying ones. This behavior is in accordance with Theorem 4, i.e., the function $f_\alpha(x)$ is not “cutting out” the singular values with relatively small magnitude since they are not present, hence the solution computed by our method coincides with the one provided by the CGLS method. Even if on this dataset our proposal does not compare favorably with CGLS in terms of execution time, the results here obtained suggest its typical use-case, i.e., our proposal should be employed for problems exhibiting a fast decay of the smallest singular value to zero.

Table 4 ‘parallelotomo’ test problem, different levels of noise δ , $\eta = 1.01$. The relaxation parameters for Kaczmarz, random Kaczmarz, Cimmino, and Landweber methods are set with `train.relaxpar(A,b,x,true,@method,20)`, while the threshold parameter τ is computed by `train.dpme(A,b1,x,true,@method,'DP',NoiseLevel,10,20,options)`

(a) Regularization results												
δ	$f_a(A_n^T, A_{n,m})A_n^T \mathbf{g}_n$			Kaczmarz			Random Kaczmarz			Cimmino		
	IT	T(s)	PSNR	IT	T(s)	PSNR	IT	T(s)	PSNR	IT	T(s)	PSNR
1.0e-03	28 (193)	3.8e+01	26.73 (28.98)	20 (800)	8.9e+01	27.13 (28.74)	21 (777)	9.2e+01	27.38 (28.96)	Cimmino		
1.0e-02	18 (30)	3.0e+01	25.96 (26.30)	317 (17)	1.3e+03	22.59 (26.09)	800 (18)	3.5e+03	19.56 (26.01)	IT		
1.0e-01	9 (6)	1.2e+01	19.78 (20.60)	800 (2)	3.4e+03	3.73 (19.97)	800 (4)	3.4e+03	4.14 (19.45)	IT		
2.0e-01	7 (4)	9.5e+00	15.96 (18.29)	800 (1)	3.4e+03	-1.91 (17.52)	800 (9)	3.5e+03	5.65 (18.03)	IT		
3.0e-01	7 (4)	1.1e+01	12.22 (17.22)	800 (1)	3.4e+03	-5.43 (15.96)	800 (1)	3.4e+03	-4.25 (15.64)	IT		
5.0e-01	7 (3)	9.6e+00	7.29 (16.06)	800 (1)	3.4e+03	-9.87 (13.02)	800 (1)	3.4e+03	-8.69 (13.03)	IT		
Landweber												
δ	IT	T(s)	PSNR	IT	T(s)	PSNR	IT	T(s)	PSNR	Cimmino		
1.0e-03	800 (800)	9.1e+02	26.83 (26.83)	56 (193)	6.4e+01	27.96 (28.98)	800 (800)	8.2e+02	27.02 (27.02)	IT		
1.0e-02	800 (800)	7.4e+02	26.32 (26.32)	14 (30)	1.3e+01	25.65 (26.30)	800 (687)	7.4e+02	26.17 (26.17)	IT		
1.0e-01	800 (44)	7.4e+02	15.58 (20.73)	6 (6)	5.9e+00	20.60 (20.60)	800 (33)	7.5e+02	13.57 (20.26)	IT		
2.0e-01	800 (21)	7.9e+02	9.89 (18.43)	4 (4)	4.9e+00	18.29 (18.29)	800 (16)	8.8e+02	7.78 (17.88)	IT		
3.0e-01	800 (15)	7.4e+02	6.42 (17.19)	3 (4)	3.3e+00	17.11 (17.22)	800 (11)	7.4e+02	4.47 (16.56)	IT		
5.0e-01	800 (9)	7.3e+02	2.20 (15.77)	2 (3)	2.3e+00	15.97 (16.06)	800 (6)	7.2e+02	0.21 (15.03)	IT		

Table 4 (continued)

(b) Trained parameters											
δ	Kaczmarz		Random Kaczmarz		Cimmino		Landweber				
	ω	τ	ω	τ	ω	τ	ω	τ	ω	τ	
1.0e-03	8.0e-01	2.8e+01	1.2e+00	4.9e+00	3.2e+02	1.0e-03	1.7e-04	1.0e-03	1.7e-04	1.0e-03	
1.0e-02	5.8e-01	9.8e+00	7.6e-01	4.3e+00	3.2e+02	1.0e-02	1.7e-04	1.0e-02	1.7e-04	1.0e-02	
1.0e-01	2.8e-01	6.1e+00	2.6e-01	7.0e+00	3.2e+02	1.0e-01	1.7e-04	1.0e-01	1.7e-04	1.0e-01	
2.0e-01	2.6e-01	6.6e+00	3.4e-02	7.5e+01	3.1e+02	2.0e-01	1.7e-04	2.0e-01	1.7e-04	2.0e-01	
3.0e-01	2.6e-01	6.6e+00	1.9e-01	9.8e+00	2.9e+02	3.0e-01	1.7e-04	3.0e-01	1.7e-04	3.0e-01	
5.0e-01	2.6e-01	6.6e+00	1.9e-01	9.8e+00	2.8e+02	5.1e-01	1.6e-04	5.1e-01	1.6e-04	5.1e-01	

5 Conclusion and future perspectives

In this work we have introduced a hybrid Krylov method for the regularization of discrete inverse problems through the usage of matrix functions computed with Krylov methods. This construction generalizes the approach used for structured regularizing preconditioners to cases in which the opportune structure is not easily devised. The theoretical justification of the given approach is based on spectral filtering framework for the inverse, or the pseudo-inverse, of an ill-posed operator. We have discussed a heuristic for the stopping criterion and for the selection of the parameters of the filter that does not require further computations to be made on the system matrix, needing only the knowledge of the norm of the additive noise, and we plan to extend to a fully adaptive and automatic choice of the parameters in the style of [11–13].

The numerical results show that the proposed methods behave consistently throughout different applications, and are able to deal with cases in which the noise level is high ($> 20\%$). The comparison with the standard Krylov-type methods is promising in terms of achieved PSNR and timings.

Moreover, the proposed methods behave better than both the Tikhonov method and the fixed point methods (Kaczmarz, Cimmino, Landweber) with trained parameters in several test cases.

Matter of future investigations is the usage of rational Krylov methods for the computation of the various matrix functions and the possible exploitation of different filtering functions within the same framework.

Funding Open Access funding provided by Università degli Studi di Padova within the CRUI-CARE Agreement. This work has been partially funded by the 2019 GNCS-INDAM Project “Tecniche innovative e parallele per sistemi lineari e non lineari di grandi dimensioni, funzioni ed equazioni matriciali ed applicazioni”.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of inverse problems, vol. 375. Springer Science & Business Media (1996)
2. Hanke, M., Nagy, J., Plemmons, R.: Preconditioned iterative regularization for ill-posed problems. In: Numerical linear algebra (Kent, OH, 1992), pp. 141–163. de Gruyter, Berlin (1993)
3. Estatico, C.: A classification scheme for regularizing preconditioners, with application to Toeplitz systems. *Linear Algebra Appl.* **397**, 107–131 (2005). <https://doi.org/10.1016/j.laa.2004.10.006>

4. Cipolla, S., Di Fiore, C., Durastante, F., Zellini, P.: Regularizing properties of a class of matrices including the optimal and the superoptimal preconditioners. *Numer. Linear Algebra Appl.* **26**(2). <https://doi.org/10.1002/nla.2225>, <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85058701148&doi=10.1002fnla.2225&partnerID=40&md5=ba11b64a73f577ff5405c1f3e696d885> (2019)
5. Donatelli, M., Hanke, M.: Fast nonstationary preconditioned iterative methods for ill-posed problems, with application to image deblurring. *Inverse Probl.* **29**(9), 095008, 16 (2013). <https://doi.org/10.1088/0266-5611/29/9/095008>
6. Buccini, A.: Regularizing preconditioners by non-stationary iterated Tikhonov with general penalty term. *Appl. Numer. Math.* **116**, 64–81 (2017). <https://doi.org/10.1016/j.apnum.2016.07.009>
7. Dell’Acqua, P., Donatelli, M., Estatico, C., Mazza, M.: Structure preserving preconditioners for image deblurring. *J. Sci. Comput.* **72**(1), 147–171 (2017). <https://doi.org/10.1007/s10915-016-0350-2>
8. Brezinski, C., Novati, P., Redivo-Zaglia, M.: A rational Arnoldi approach for ill-conditioned linear systems. *J. Comput. Appl. Math.* **236**(8), 2063–2077 (2012). <https://doi.org/10.1016/j.cam.2011.09.032>
9. Novati, P., Redivo-Zaglia, M., Russo, M.R.: Preconditioning linear systems via matrix function evaluation. *Appl. Numer. Math.* **62**(12), 1804–1818 (2012). <https://doi.org/10.1016/j.apnum.2012.07.001>
10. O’Leary, D.P., Simmons, J.A.: A bidiagonalization-regularization procedure for large scale discretizations of ill-posed problems. *SIAM J. Sci. Stat. Comput.* **2**(4), 474–489 (1981). <https://doi.org/10.1137/0902037>
11. Lewis, B., Reichel, L.: Arnoldi-Tikhonov regularization methods. *J. Comput. Appl. Math.* **226**(1), 92–102 (2009). <https://doi.org/10.1016/j.cam.2008.05.003>
12. Gazzola, S., Novati, P.: Automatic parameter setting for Arnoldi-Tikhonov methods. *J. Comput. Appl. Math.* **256**, 180–195 (2014). <https://doi.org/10.1016/j.cam.2013.07.023>
13. Gazzola, S., Novati, P., Russo, M.R.: On Krylov projection methods and Tikhonov regularization. *Electron. Trans. Numer. Anal.* **44**, 83–123 (2015)
14. Benzi, M., Boito, P., Razouk, N.: Decay properties of spectral projectors with applications to electronic structure. *SIAM Rev.* **55**(1), 3–64 (2013). <https://doi.org/10.1137/100814019>
15. Liang, W., Saravanan, C., Shao, Y., Baer, R., Bell, A.T., Head-Gordon, M.: Improved fermi operator expansion methods for fast electronic structure calculations. *J. Chem. Phys.* **119**(8), 4117–4125 (2003)
16. Hanke, M.: Conjugate gradient type methods for ill-posed problems. Pitman Research Notes in Mathematics Series. Longman Scientific & Technical, Harlow, vol. 327 (1995)
17. Di Benedetto, F., Serra Capizzano, S.: A unifying approach to abstract matrix algebra preconditioning. *Numer. Math.* **82**(1), 57–90 (1999). <https://doi.org/10.1007/s002110050411>
18. Cipolla, S., Di Fiore, C., Tudisco, F., Zellini, P.: Adaptive matrix algebras in unconstrained minimization. *Linear Algebra Appl.* **471**, 544–568 (2015). <https://doi.org/10.1016/j.laa.2015.01.010>
19. Hanke, M., Nagy, J.: Inverse Toeplitz preconditioners for ill-posed problems. *Linear Algebra Appl.* **284**(1-3), 137–156 (1998). [https://doi.org/10.1016/S0024-3795\(98\)10046-0](https://doi.org/10.1016/S0024-3795(98)10046-0). ILAS Symposium on Fast Algorithms for Control, Signals and Image Processing (Winnipeg, MB, 1997)
20. Higham, N.J.: *Functions of matrices. Society for Industrial and Applied Mathematics (SIAM), Philadelphia.* <https://doi.org/10.1137/1.9780898717778>. Theory and computation (2008)
21. Saad, Y. *Iterative methods for sparse linear systems*, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2003). <https://doi.org/10.1137/1.9780898718003>
22. Hansen, P.C., Jørgensen, J.S.: AIR Tools II: algebraic iterative reconstruction methods, improved implementation. *Numer. Algorithms.* **79**(1), 107–137 (2018). <https://doi.org/10.1007/s11075-017-0430-x>
23. Gazzola, S., Hansen, P.C., Nagy, J.G.: IR Tools: a MATLAB package of iterative regularization methods and large-scale test problems. *Numer. Algorithms.* <https://doi.org/10.1007/s11075-018-0570-7> (2018)
24. Calvetti, D., Lewis, B., Reichel, L.: GMRES, L-curves, and discrete ill-posed problems. *BIT* **42**(1), 44–65 (2002). <https://doi.org/10.1023/A:1021918118380>
25. Chung, J., Nagy, J.G., O’Leary, D.P.: A weighted-GCV method for Lanczos-hybrid regularization. *Electron. Trans. Numer. Anal.* **28**, 149–167 (2007/08)
26. Hansen, P.C.: Regularization Tools version 4.0 for Matlab 7.3. *Numer. Algorithms.* **46**(2), 189–194 (2007). <https://doi.org/10.1007/s11075-007-9136-9>

27. Phillips, D.L.: A technique for the numerical solution of certain integral equations of the first kind. *J. Assoc. Comput. Mach.* **9**, 84–97 (1962). <https://doi.org/10.1145/321105.321114>
28. Wing, G.M.: A primer on integral equations of the first kind. Society for Industrial and Applied Mathematics (SIAM), Philadelphia. <https://doi.org/10.1137/1.9781611971675>. The problem of deconvolution and unfolding, With the assistance of John D. Zahrt (1991)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.