High Performance Linear Algebra

Lecture 0: Bureaucracy and Contents

Ph.D. program in High Performance Scientific Computing

Fabio Durastante   Pasqua D'Ambra   Salvatore Filippone

November 5, 2025

Dipartimento di Matematica
Università di Pisa

About this course: general information

First some *bureaucratic* information about the course:

- Course webpage: fdurastante.github.io/courses/hpla2025.html
- Lecture slides: fdurastante.github.io/courses/hpla2025.html#lectures
- The course consists of **30 hours** of lectures, divided into **15 lectures** of **2 hours** each

The **exam** will consist in a **project work** to be presented at the end of the course. This will involve the implementation and performance analysis of some linear algebra algorithms, or the performance analysis of existing libraries, possibly in relation to a specific application. The choice of the project topic will also depend on your Ph.D. research topic, so to make it more interesting and useful for you.

What is Linear Algebra?

Linear Algebra is a branch of mathematics concerned with:

- **Vector spaces** and linear transformations
- Systems of **linear equations**, **matrices**, **vectors**
- Key concepts: determinants, eigenvalues, eigenvectors, singular values

Applications: computer graphics, machine learning, optimization, physics

Numerical Linear Algebra focuses on:

- Solving LA problems using numerical methods on computers
- Development of **efficient**, **stable**, and **accurate** algorithms
- Essential for *large-scale* problems where exact solutions are impractical

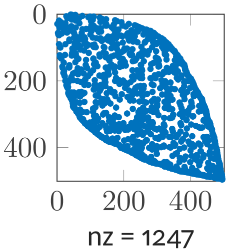
Problem 1: Linear Systems

Consider the Poisson equation (PDE):

$$-\Delta u = f \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \partial\Omega$$

Discretization approach:

- Divide domain into grid: $N = n_1 \times n_2 \times n_3$ points
- Use finite difference approximation for derivatives
- Results in sparse linear system: $A\mathbf{u} = \mathbf{f}$
- $A \in \mathbb{R}^{N \times N}$ is **sparse**
- Most elements are zero
- N is typically very large



Sparse matrix pattern

Find scalar λ and vector $\mathbf{v}$ such that:

$$P\mathbf{v} = \lambda\mathbf{v}$$

Example: Markov Chains

- Transition matrix $P \in \mathbb{R}^{N\times N}$ ($P_{i,j} \geq 0$, rows sum to 1)
- Evolution: $\mathbf{p}_{\ell+1} = P\mathbf{p}_{\ell}$
- Stationary distribution $\boldsymbol{\pi}$ satisfies:

$$\boldsymbol{\pi}^\top = \boldsymbol{\pi}^\top P, \quad \boldsymbol{\pi}^\top\mathbf{1} = 1$$

- Finding $\boldsymbol{\pi}$ is an eigenvalue problem for large N

Problem 3: Matrix Equations

Sylvester equation: $AX + XB = C$

Application: Model Reduction in Control Theory

LTI dynamical system:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \quad \mathbf{y}(t) = C\mathbf{x}(t)$$

Balanced truncation approach:

1. Compute Gramians via Lyapunov equations:

$$AP + PA^\top + BB^\top = 0 \quad \text{(controllability)}$$
$$A^\top Q + QA + C^\top C = 0 \quad \text{(observability)}$$

2. Solve Sylvester equation: $AT + TS = B$
3. Efficient algorithms needed for large dimensions

Problem 4: Machine Learning

Linear Regression:

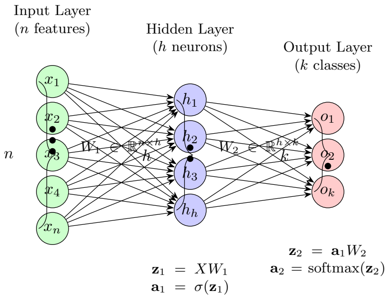
- Data: $X \in \mathbb{R}^{m \times n}$, targets: $\mathbf{y} \in \mathbb{R}^m$
- Find coefficients: $\min_\beta \|X\beta - \mathbf{y}\|_2^2$

Neural Networks:

- Weights as matrices: $W_1 \in \mathbb{R}^{n \times h}$, $W_2 \in \mathbb{R}^{h \times k}$
- Forward pass:

$$\mathbf{a}_1 = \sigma(XW_1)$$
$$\mathbf{a}_2 = \text{softmax}(\mathbf{a}_1 W_2)$$

- Training relies on matrix operations



Neural network

Take Home Message

Key Point

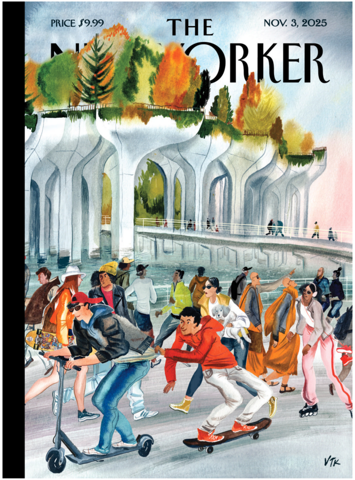
Applied mathematics is fundamentally about solving combinations of linear algebra problems.

Modern challenges:

- Ever larger problem sizes
- Need for reliable results in reasonable time
- Requirements: *efficient*, *scalable*, *parallel* algorithms

$\Rightarrow$ This motivates high-performance numerical linear algebra!

Straight from the press



*The math is unwieldy, and is especially dependent on an operation known as matrix multiplication. "Beauty is the first test: there is no permanent place in the world for ugly mathematics," the mathematician G. H. Hardy wrote, in 1940. But **matrix multiplication**, to which our civilization is now devoting so many of its marginal resources, **has all the elegance of a man hammering a nail into a board**. It is possessed of neither beauty nor symmetry: in fact, in matrix multiplication, a times b is not the same as b times a. As the matrices increase in size, the arithmetic requires great computational power to solve.*

S. Witt — The New Yorker, November 3rd, 2025

Nonetheless...

I will try to convince you that:

There is both beauty and elegance in Matrix Multiplication!

Fill the following poll: date.dm.unipi.it/hpladate to select your preferred time slot for the course.