

Information on this module

Instructor Federico Poloni — federico.poloni@unipi.it.
Room 343 DN.

Office hours (ricevimento): Friday at 11 (but **not** tomorrow, because exams; replacement date Wed 23 9-11).

Where? Join the team on MS Teams with code ya2wxug, for now. Different times (or a face-to-face appointment in Pisa) can be arranged on request.

Content

Prof. Frangioni's part: solve **all** the problems: $\min_S f(x)$.

This part: solve a **very specific** problem:

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2.$$

“Solve a linear system (or find something close)”

Why is this still relevant? It is a problem we can understand well:

- ▶ Specialized algorithms for direct solution;
- ▶ Focus on maximum accuracy attainable;
- ▶ Scales well to large dimensions;
- ▶ ‘Features’ useful for theoretical understanding: e.g., principal components.

Obligatory XKCD



<https://xkcd.com/1838/>

Books

Books (for this part)

- ▶ Trefethen–Bau, *Numerical linear algebra*. **Recommended**: we will follow more its approach.
- ▶ Demmel, *Applied numerical linear algebra*. Alternative source for other explanations, further exercises. . .
- ▶ Eldén, *Matrix Methods in Data Mining and Pattern Recognition*, <https://epubs.siam.org/doi/book/10.1137/1.9780898718867>. Not very detailed for our purposes, but gives good insight on some topics. You can access it from free from the university network (or via <https://vpn.unipi.it>).

Languages

Matlab: proprietary language specialized in matrix computations. Clunky language with a big Visual Basic-like IDE, great libraries and syntactic sugar for numerics.

Python: great simple language with decent numerics libraries stapled onto it.

Fortran (yes, that's still a thing), **C/C++**: sometimes needed for high performance, because they don't have the 'interpreted language problem'.

Julia: newer language (same age as Go / Rust): stable syntax, but toolchain not too mature. It tries to combine the advantages of the others (syntactic sugar for numerics + libraries + fast for loops).

Under the core: usually the same libraries (**Blas/Lapack**) for basic operations: summing vectors, multiplying matrices. . .

How to install Matlab

Available without charge to Unipi students: info on <http://matlab.sid.unipi.it>.

TL;DR

- ▶ Create an account on www.mathworks.com with your @studenti.unipi.it e-mail;
- ▶ Add the 'activation key' found on http://doc.sid.unipi.it/images/1/15/Istruzioni_Installazione_MatLab_Student_2017.pdf;
- ▶ Download and install.

Suggested to save disk space and/or bandwidth: install only a few 'toolboxes'. For this course: **Matlab, Symbolic Toolbox, Optimization Toolbox.**

You may want also Statistics and Machine Learning Toolbox, Neural Network Toolbox.

Matlab introduction

Command window

```
>> a = 19
```

```
a =
```

```
    19
```

```
>> b = 5
```

```
b =
```

```
     5
```

```
>> a+b
```

```
ans =
```

```
    24
```

```
>> c = 2*a + b; % semicolons suppress output
```

Accuracy of operations

Default data type: double.

Warning: operations are only accurate 'up to 16 digits'.

```
>> (1/98) * 98 - 1  
ans =  
-1.1102e-16
```

(exponential notation: stands for -1.1102×10^{-16})

Understanding these approximations will be an important part of this module.