

How to destroy $Ax=b$?

Florian De Bel

March 11, 2025

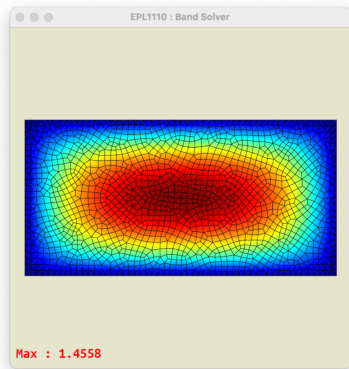
2 worlds

Direct solver

Iterative solver

Band Solver

A low storage and fast solver for sparse matrix compared to a dense Gaussian elimination (both are in the homework)



Other idea explains in the syllabus: Frontal solver

Cholesky Decomposition

The decomposition is valid only for positive definite matrix and follows

$$A = LL^T$$

with L a lower triangular matrix.

\Rightarrow Crout or Banachiewicz algorithm

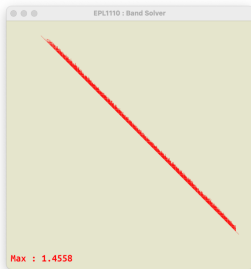
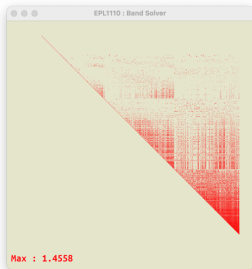
Then we use a forward and a backward substitution to get the solution of the system

$$Ly = b \quad \text{and} \quad L^T x = y$$

Fill in problem

Reordering the matrix to reduce the fill in

- In X or Y
- Reverse Cuthill-McKee (RCMK)
- Approximate Minimum Degree (AMD)
- Nested Dissection (ND)



Iterative Solver

- Richardson iteration
- Jacobi iteration
- Conjugate Gradient
- GMRES
- Multigrid

Preconditioning

The goal is to decrease the condition number of the matrix using a preconditioner to have less iterations. The system becomes

$$P_1^{-1}AP_2^{-1}y = P_1^{-1}b$$

with

$$x = P_2^{-1}z$$

where P_1 and P_2 are the left and right preconditioner

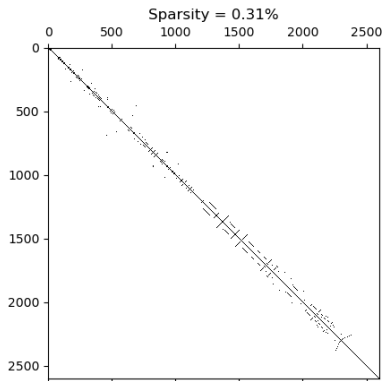
$$\kappa(A) = \frac{\lambda_{max}}{\lambda_{min}}$$

Preconditioner:

- Diagonal
- Incomplete LU (iLU0, iLUk)

Low storage

- COO
- CSR
- CSC
- ELL
- BSR



Storage system

$$A = \begin{bmatrix} 1 & 0 & 4 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 6 & 3 & 0 \\ 0 & 0 & 5 & 1 \end{bmatrix}$$

COO:

$$data = [1, 4, 2, 6, 3, 5, 1]$$

$$row = [0, 0, 1, 2, 2, 3, 3]$$

$$col = [0, 2, 1, 1, 2, 2, 3]$$

CSR:

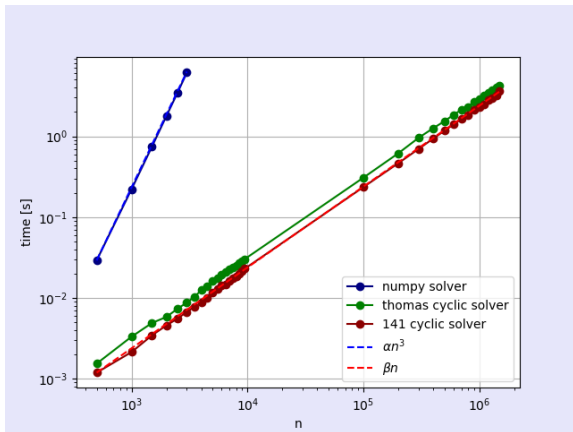
$$data = [1, 4, 2, 6, 3, 5, 1]$$

$$row = [0, 2, 3, 5, 7]$$

$$col = [0, 2, 1, 1, 2, 2, 3]$$

Speed diagnostic

Use log-log plot to show the convergence of your solver !



Grand Prix International 2025 de l'Elément le Plus Fini

Pour la première fois en 2025, l'équipe didactique a décidé d'étendre le matériel autorisé au-delà d'un simple coeur. Tout processeur ou GPU branché à la machine de test sera utilisable! Pour accéder à ces ressources, les langages suivants sont autorisés :

- Le langage C et ses bibliothèques standard (`pthread`, `#pragma omp parallel for`, ...)
- Le langage C++ et toute la STL (`std::execution::unseq`, `std::experimental::simd`, ...)
- Cuda et ses bibliothèques, à l'exception de cuBLAS, cuFFT, cuSPARSE, cuSOLVER (sinon le meilleur code tient en 2 lignes et ce n'est pas rigolo :-)
- OpenCL (3.0) et OpenGL (4.6)

L'utilisation de toute autre bibliothèque externe est formellement interdite. Néanmoins le choix du standard C/C++ est libre (C99/C11/gnu99,...)

Les tests de performance s'effectueront sur *Bobby*, le nouveau serveur MEMA. *Bobby* est équipé de :

- Un processeur Intel core i5-13500
- De 32GB de RAM
- D'une carte graphique NVIDIA RTX A4000 (compute capability 8.6)
- Les compilateurs disponibles sont: gcc-12 (g++-12), clang14 (clang++-14), nvcc 12.8