

System of linear equations

Numerical methods used to solve such problem allow to introduce and experiment on [Time_complexity](#), considering cubic time behavior of standard algorithms and *i.e.* quadratic time solutions using LU decomposition.

Theory

- [System_of_linear_equations](#)
- [Gaussian_elimination](#), Gauss and Gauss-Jordan eliminations (diagonalization, triangularization)
- [Pivot_element](#), pivoting
- [LU_decomposition](#)
 - [Triangular_matrix#Forward_and_back_substitution](#)
- Time complexity analysis
 - Hint : in Python, use the timeit module

Exercices and applications

- Exercices :
 - write a python function for diagonalisation with partial pivoting
 - random numbers → linear systems
 - comparison with numpy standard library
 - measurements of execution time to check cubic complexity

1D problems with neighbours

- Thermal diffusion and chemical diffusion (transient or stationary) on a regular 1D space with equidistant steps. ODE equations can be written such a given evolution equation for node # i only implies nodes $i+1$ and $i-1$
- Using [tridiagonal Thomas algorithm](#) allows to save computational time thanks to n complexity
- ? Python library with Thomas algorithm

References :

- Numerical recipes, The Art of Scientific Computing 3rd Edition, William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, 2007, isbn: 9780521880688
 - <http://numerical.recipes/>
 - http://www2.units.it/ipi/students_area/imm2/files/Numerical_Recipes.pdf
 - <http://apps.nrbook.com/empanel/index.html#>

From: <https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link: https://dvillers.umons.ac.be/wiki/teaching:methcalchim:system_of_linear_equations?rev=1506587988

Last update: **2017/09/28 10:39**

