Elemental numerical methods

S1

A tridiagonal linear system

1. We wish to solve the differential equation
$$\begin{aligned} -u''(x) &= f(x), \quad x \in]0,1[, \\ u(0) = u(1) &= 0. \end{aligned}$$
For this, we propose the following finite difference scheme. A positive integer n is given. We set $\Delta x = 1/(n+1)$. The grid points are $x_i = i\Delta x$, $i = 0 \cdots n$. We look for approximations of $u(x_i) \simeq u_i$ for $i = 1 \cdots n$. We suppose that $u_0 = u_{n+1} = 0$. The set of equations is
$$\frac{-u_{i-1} + 2u_i - u_{i+1}}{\Delta x^2} = f(x_i), \quad i = 1 \cdots n.$$
Justifiy these equations thanks to a Taylor expansion.

2. Write the linear system $AU = F$ associated to this set of equations.

3. Write a program to solve the previous linear system using the given `factolu` subroutine.

4. Compare the solution of your program with an exact solution of the differential equation for several value of n. Plot on the same graphic the two solutions in order to demonstrate that your program performs correctly.

Morse and skyline storage

1. Recall what are the morse and the skyline storages of a matrix

2. Write a program that convert a matrix stored in the morse form into the skyline form. Check that your program works on several simple matrices (write on the screen all the arrays in order to verify that your programing is correct).

3. Solve the linear system of the previous tridiagonal system with the provided `sol` subroutine. Check the correct LU factorization and the correct solution by printing the arrays for a small matrix.

Solving the Laplace equation on a grid

In order to solve approximately the two-dimensional problem
$$\begin{aligned} -\Delta u &= f \quad \text{on } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$
where $\Omega$is the square $]0,1[\times]0,1[$, we consider the following finite difference approximation
$$\frac{4u_{ij} - u_{i-1j} - u_{i+1j} - u_{ij-1} - u_{ij+1}}{\Delta x^2} = f(x_i, y_j).$$
The space step is $\Delta x = 1/(n+1)$ and the grid points are $(x_i, y_j) = (i\Delta x, j\Delta x)$.

1. The finite difference approximation can be written also as a linear system $AU = F$. Write a program that compute the morse storage of the matrix A.

2. Convert the morse storage into a skyline storage and solve the previous linear system with the provided subroutine.

3. Check that your program is correct by comparing the numercial solution with an exact solution obtained with an adequately chosen f.

Iterative methods

1. Write a program that solves the previous linear system by the iterative method of Jacobi. When do you stop the iterations?

2. Same question with the conjugate gradient method.

3. What is the best method of the two previous methods?