Series 1 Warmup

Numerical methods for PDEs
Last edited: February 28, 2017
Due date: 2017-03-14 at 23:59

Template codes are available on the course's webpage at `https://moodle-app2.let.ethz.ch/course/view.php?id=3089`.

This is a warmup problem. You do **not need to hand in** this problem.

Exercise 1 Finite Differences for Poisson Equation in 2D

In this problem we consider the Finite Differences discretization of the Poisson problem on the unit square:

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega := (0,1)^2, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{1}$$

for a bounded and continuous function $f \in \mathcal{C}^0(\overline{\Omega})$.

We consider a regular tensor product grid with meshwidth $h := (N+1)^{-1}$ and we assume a lexicographic numbering of the interior vertices of the mesh as depicted in Fig.1.

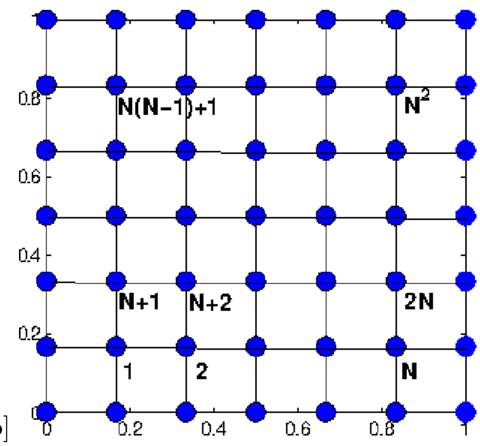
[b] 

Figure 1: Lexicographic numbering of vertices of the equidistant tensor product mesh.

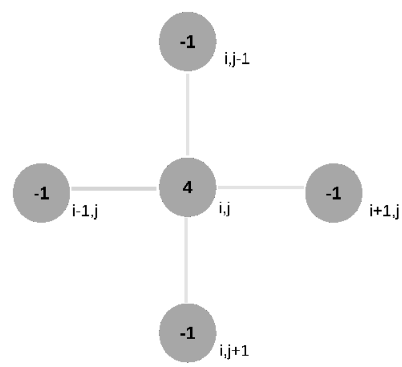


Figure 2: 5-point stencil used in this problem.

We consider the 5-point stencil finite difference scheme for the operator $-\Delta$ described by the 5-points stencil shown in Fig. 2.

1a)

Write the system

$$\mathbf{A}\mathbf{u} = \mathbf{F} \tag{2}$$

corresponding to the discretization of (1) using the stencil in Fig. 2, specifying the matrix $\mathbf{A}$ and the vectors $\mathbf{F}$ and $\mathbf{u}$.

1b)

In the template file `finite_difference.cpp`, implement the function

```
void createPoissonMatrix2D(SparseMatrix& A, int N),
```

to construct the matrix $\mathbf{A}$ in (2), where N denotes the number of interior grid points along one dimension, with `typedef Eigen::SparseMatrix<double> SparseMatrix`. Assume the matrix `A` to have an uninitialized size at the beginning.

1c)

In the template file `finite_difference.cpp`, implement the function

```
void createRHS(Vector& rhs, FunctionPointer f, int N, double dx),
```

to build the vector $\mathbf{F}$ in (2), with `typedef Eigen::VectorXd Vector` and `typedef double(*FunctionPointer)(double, double)`. The argument `f` is a function pointer to the function f in (1), `N` is the number of interior grid points and `dx` is cell width. Again, assume that the vector `rhs` has uninitialized size when passed in input.

1d)

In the template file `finite_difference.cpp`, implement the function

```
void poissonSolve(Vector& u, FunctionPointer f, int N),
```

to solve the system (2), with `u` the vector containing the values of the approximate solution at all the grid points, *including those on the boundary*, and the other arguments as in the previous subproblems.

1e)

Plot the discrete solution that you get from subproblem **1d)** for $f(x, y) = 8\pi^2 \sin(2\pi x)\sin(2\pi y)$ and $N = 50$, and compare it to the exact solution $u(x, y) = \sin(2\pi x)\sin(2\pi y)$.