Series 1

Numerical methods for PDEs
Last edited: March 1, 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`.

Exercise 1 Finite Differences for the Poisson equation in 1D

We consider the 1D Poisson equation with homogeneous Dirichlet boundary conditions:

$$\begin{aligned} -u''(x) &= f(x), \quad \forall x \in \Omega = (0,1) \\ u(0) &= u(1) = 0. \end{aligned} \tag{1}$$

We want to discretize eq. (1) using the Finite Differences method, namely the centered finite differences. To this aim, we subdivide the interval $[0,1]$ in $N+1$ subintervals using equispaced grid points

$$\{x_0 = 0, x_1, \ldots, x_N, x_{N+1} = 1\}.$$

The discretized problem can be written as a linear system

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

where $\mathbf{A}$ is a $N \times N$ matrix, $\mathbf{F}$ a $N \times 1$ vector and $\mathbf{u}$ the $N \times 1$ vector containing the unknowns $u_j \approx u(x_j)$, $j = 1, \ldots, N$, the approximate values of the function u at the grid points.
Let us denote by $h = |x_1 - x_0|$ the meshsize.

1a)

Using central finite differences, write the matrix $\mathbf{A}$ and the right hand side vector $\mathbf{F}$. For the right hand side, write it in terms of a generic force term $f(x)$ in (1).

1b)

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

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

where `typedef Eigen::SparseMatrix<double> SparseMatrix`. This function computes the matrix $\mathbf{A}$ for (2). Here the input parameter `N` denotes the number of *interior* grid points. Assume that the size of the input matrix `A` has not been initialized.

1c)

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

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

where `typedef double(*FunctionPointer)(double)` and `typedef Eigen::VectorXd Vector`. This function computes the right-handside $\mathbf{F}$ for (2). The input parameter `f` is the function pointer for the right-handside $f(x)$, `N` is again the number of interior grid points, and `dx` is the length of a cell. Assume that the size of the input vector `rhs` has not been initialized.

1d)

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

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

to solve the Poisson problem (1).
The input parameters `f` and `N` are as in subproblem **1c)**. The vector `u` is assumed to have not been initialized in size, and at the end of the routine it has to correspond to the array $\{u_h(x_j)\}_{j=1}^N$ containing the approximate values of the solution u at the interior grid points $\{x_j\}_{j=1}^N$.

Hint: Use the routines from subproblems **1b)** and **1c)**.

1e)

Run the routine `poissonSolve` for $f(x) = \sin(2\pi x)$ and $N = 50$ and plot the solution.

We saw in the lecture that the centered finite difference schemes satisfies is stable and consistent, and thus it converges to the exact solution u to (1) when the mesh is refined.
Here we are going to study the convergence of our scheme.

1f)

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

```
void poissonConvergence( std::vector<double>& errors, std::vector<int>& resolutions,
↪ FunctionPointer f)
```

,

to perform the convergence study. The input argument `f` is a function pointer to the right-hanside $f(x)$. The vectors `resolutions` and `errors`, assumed to be passed in input with uninitialized size, must contain, in output, the array $[N_1, \ldots, N_6]$ of numbers of degrees of freedom and the array $[e_1, \ldots, e_6]$ of computed errors, respectively.
As error between the discrete solution u_h and the exact solution u, we consider the maximum norm error $\|u - u_{h_i}\|_\infty = \max_{x\in[0,1]} |u - u_h|$, $i = 1, \ldots, 6$; we can compute this error just in an approximate way: we consider a very fine grid with meshsize $h_{ref} = \frac{1}{2^{10}}$ and grid points $x_0 = 0, x_1 = h_{ref}, \ldots, x_{N+1} = 1$ and approximate the maximum norm by

$$\|u - u_h\|_\infty \approx \max_{x_0 \ldots x_{N+1}} |u(x_i) - u_h(x_i)| \tag{3}$$

The standard steps for a convergence study then:

1. compute the *exact solution* u to (1);
2. start from a meshsize $h_1 = \frac{1}{4}$, corresponding to $N_1 = 3$ interior grid points;
3. compute the discrete solution u_{h_1} to (1);
4. compute the error $e_1 \approx \|u - u_{h_1}\|_\infty$; inside each mesh interval, consider the linear interpolant for u_h;
5. refine the grid, considering $h_2 = \frac{h_1}{2} = \frac{1}{8}$ and repeat the algorithm from step 3;
6. repeat the previous step till $h_6 = \frac{h_1}{2^5}$.

1g)

Run the routine `poissonConvergence` for $f(x) = \sin(2\pi x)$.
Make a double logarithmic plot of the errors $e_1, \ldots, e_6$ versus the resolutions $N_1, \ldots, N_6$. What do you observe? Which is the order of convergence?

Exercise 2 Finite Differences for the Porous Media Equation in 2D

In this problem we consider the finite differences (FD) discretization of the equation of porous media on the unit square:

$$\begin{aligned} -\nabla \cdot (\sigma \nabla u) &= f \quad \text{in } \Omega := (0,1)^2, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{4}$$

for a bounded and continuous function $f \in \mathcal{C}^0(\overline{\Omega})$. In a general formulation of the problem σ is a smooth function of u; for simplicity, here we take $\sigma : \Omega \longrightarrow \mathbb{R}$, $\sigma \in \mathcal{C}^1(\Omega)$.

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]

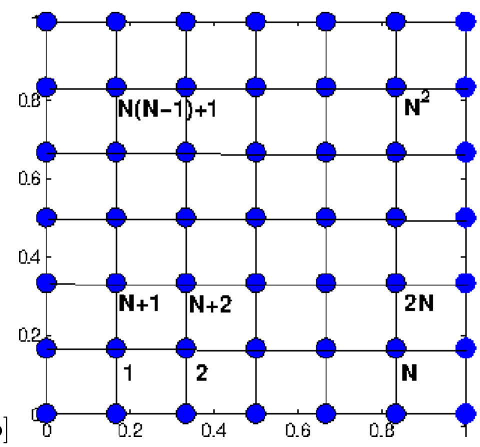

Figure 1: Lexicographic numbering of vertices of the equidistant tensor product mesh.

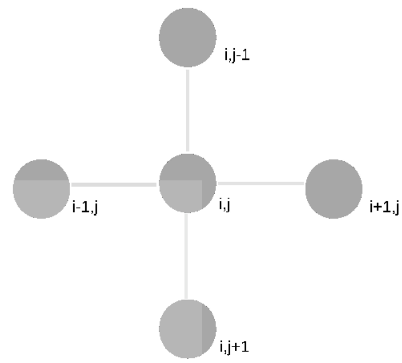

Figure 2: 5-point stencil used in this problem.

We consider the 5-point finite difference scheme described by the stencil shown in Fig. 2.

2a)

Write the system

$$\mathbf{A}\mathbf{u} = \mathbf{F} \tag{5}$$

corresponding to the discretization of (4) using the stencil in Fig. 2, specifying the matrix $\mathbf{A}$ and the vectors $\mathbf{F}$ and $\mathbf{u}$.

Hint: Consider starting the discretization as $\frac{\partial}{\partial x}(\sigma \frac{\partial u}{\partial x}) \approx \frac{1}{h}((\sigma \frac{\partial u}{\partial x})_{i+\frac{1}{2},j} - (\sigma \frac{\partial u}{\partial x})_{i-\frac{1}{2},j})$

2b)

(Core problem) In the template file `finite_difference.cpp`, implement the function

```
void createPorousMediaMatrix2D(SparseMatrix& A, FunctionPointer sigma, int N, double dx),
```

to construct the matrix $\mathbf{A}$ in (5), 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.

2c)

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 (5), with `typedef Eigen::VectorXd Vector` and `typedef double(*FunctionPointer)(double, double)`. The argument `f` is a function pointer to the function f in (4), `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.

Note: This is **not** a core problem and awards no points by itself, but a complete solution of exercise **2d)** requires the implementation of this function.

2d)

(Core problem) In the template file `finite_difference.cpp`, implement the function

```
Vector porousMediaSolve(FunctionPointer f, FunctionPointer f, int N),
```

to solve the system (5), which returns `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.

2e)

Plot the discrete solution that you get from subproblem **2d)** when

$$f(x, y) = 4\pi^2 \sin(2\pi x) \sin(2\pi y)(4 \cos(2\pi x) \cos(2\pi y) + \pi), \quad \sigma(x, y) = \frac{\pi}{2} + \cos(2\pi x) \cos(2\pi y)$$

and $N = 590$.

2f)

(Core problem) In the template file `finite_difference.cpp`, implement the function

```
void convergeStudy(FunctionPointer F, FunctionPointer sigma),
```

that uses the infinity norm, i.e.

$$\text{for } \mathbf{a} \in \mathbb{R}^d, \quad \|\mathbf{a}\|_\infty := \max_{0 \le i \le d} |\mathbf{a}_i|, \tag{6}$$

to compute the error between your discrete solution obtained with

$$f(x, y) = 4\pi^2 \sin(2\pi x)\sin(2\pi y)(4\cos(2\pi x)\cos(2\pi y) + \pi), \quad \sigma(x, y) = \frac{\pi}{2} + \cos(2\pi x)\cos(2\pi y)$$

for $N = 2^k$, $k = 3 \cdots 8$ and the exact solution $u(x, y) = \sin(2\pi x)\sin(2\pi y)$.

Hint: Use your code from **2d)**.

Exercise 3 Green's Formula

To derive the variational formulation from a PDE, we needed multi-dimensional integration by parts as expressed through Green's first formula. In this problem we study the derivation of Green's formula, thus practicing elementary vector analysis and the application of Gauss' theorem.

3a)

Now prove Green's formula for $\Omega \subset \mathbb{R}^2$

$$-\int_\Omega \nabla \cdot \mathbf{j}\, v \; d\mathbf{x} = -\int_{\partial\Omega} \mathbf{j} \cdot \mathbf{n}\, v \; dS + \int_\Omega \mathbf{j} \cdot \nabla\, v \; d\mathbf{x},$$

where $\mathbf{j} \in \mathcal{C}^1(\overline{\Omega})^2$ and $v \in \mathcal{C}^1(\overline{\Omega})$.

Hint: Use Gauss' theorem.

$$\int_\Omega \nabla \cdot \mathbf{F} \; d\mathbf{x} = \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} \;\; dS,$$

where $\mathbf{F} \in \mathcal{C}^1(\overline{\Omega})^2$.