Project

This project is an individual work. All implementation work should be done in Python 3.x and rely on the libraries written for scientific computing, namely NumPy, SciPy and Matplotlib. The files submitted should include your Python scripts and any additional files (figures, ...) that you judge necessary. All figures should be self-explanatory (labels on every axis, legend, ...). **Some level of testing is expected in the implementation and you will be specifically asked questions on the tests.** This project is largely an adaptation of the code written during the programming sessions so that you should re-use some of the code already written.

1. **Geometry.** We consider the following domain

$$\Omega := ((0, 2\pi) \times (0, 2\pi)) \setminus ([\pi/2, 3\pi/2] \times [\pi/2, 3\pi/2]), \tag{1}$$

with boundary $\Gamma = \Gamma_N \cup \Gamma_D$ where Γ_N denotes the boundary of $(0, 2\pi) \times (0, 2\pi)$ and Γ_D denotes the boundary of $(\pi/2, 3\pi/2) \times (\pi/2, 3\pi/2)$.

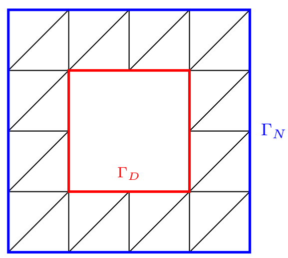

Figure 1: Meshed domain Ω with its boundaries Γ_N and Γ_D.

(a) Write a routine `GenerateMesh` adapted from the one from TP2 that generates a uniform structured triangular mesh for Ω. The outputs should be: `vtx` (coordinate array) and `elt` (connectivity array) for the mesh of the domain Ω in the format of TP2.

The mesh should be constructed by first generating a uniform triangular mesh of the full square $(0, 2\pi) \times (0, 2\pi)$. The number of points in each direction should be of the form $N = (4n + 1)$ for some $n \in \mathbb{N}$. Then the nodes and the (open) triangles with a non-trivial intersection with the square $(\pi/2, 3\pi/2) \times (\pi/2, 3\pi/2)$ should be removed. See Figure 2. The labels of the nodes and the elements should be updated, so that the labels of the nodes range from 0 to $N_\Omega - 1$ where N_Ω is the number of nodes in the mesh of Ω.

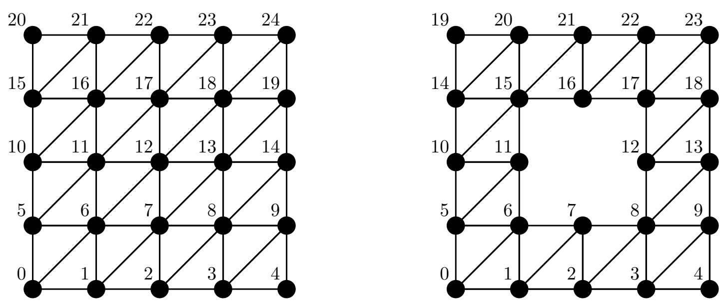

Figure 2: Examples of meshes for $n = 1$: initial square mesh (left) and mesh of Ω (right) with the labels of the nodes. Note that the labels of the mesh of Ω have changed compared to the mesh of the square.

(b) Write a routine `PlotMesh` that can represent a triangular mesh of the domain Ω and plot the new mesh.

(c) Adapt the previous routine the represent the unit outward normal vector $\mathbf{n}$ on the whole boundary Γ of Ω.

2. **PDE problem.** Recall that Γ_N denotes the boundary of $(0, 2\pi) \times (0, 2\pi)$ and Γ_D denotes the boundary of $(\pi/2, 3\pi/2) \times (\pi/2, 3\pi/2)$. Let

$$\mu(x, y) := \begin{cases} 1, & \text{if } y < \pi, \\ 2, & \text{if } y > \pi. \end{cases} \tag{2}$$

Let $p, q \in \mathbb{N}$,

$$f(x, y) := (4(p^2 + q^2) + \mu) \sin(2px) \sin(2qy), \qquad \forall (x, y) \in \Omega, \tag{3}$$

$$u_{\text{ex}}(x, y) := \sin(2px) \sin(2qy), \qquad \forall (x, y) \in \Omega. \tag{4}$$

We consider the following second-order PDE model:

$$\begin{cases} \text{Find } u \in H^1(\Omega) \text{ such that :} & \\ -\Delta u + \mu u = f, & \text{in } \Omega, \\ u = 0, & \text{on } \Gamma_D, \\ \partial_{\mathbf{n}} u = \partial_{\mathbf{n}} u_{\text{ex}}, & \text{on } \Gamma_N. \end{cases} \tag{5}$$

We want to compute an approximate solution u_h of the above problem using a conforming Galerkin method. The finite dimensional approximation space V_h is constructed using $\mathbb{P}_1$ Lagrange finite elements on triangular meshes.

(a) Write the variational formulation associated to (5).

(b) Write two routines to assemble the elementary matrices associated to each term appearing in the bilinear form for the numerical method considered, on the model of what was done in TP5.

(c) Write the routine to assemble the full matrix of the associated linear system.

(d) Write a routine to assemble (a numerical approximation of) the term associated to the Neumann boundary condition for the right-hand-side vector of the linear system.

(e) Write a routine to assemble (a numerical approximation of) the term associated to the source f for the right-hand-side vector of the linear system.

3. **Resolution.**

(a) Check that u_{ex} is the exact solution of the problem.

(b) Solve numerically (5).

(c) Write a routine `PlotApproximation` that can represent a piecewise affine field $v_h \in V_h$ in the domain Ω for some triangular mesh. Using this routine, represent the numerical solution u_h from the previous question and the associated error $u_h - \Pi_h u_{\text{ex}}$, where Π_h is the global interpolation operator.

(d) Plot the convergence of both errors

$$\frac{\|u_h - \Pi_h u_{\text{ex}}\|_{L^2(\Omega)}}{\|\Pi_h u_{\text{ex}}\|_{L^2(\Omega)}}, \qquad \frac{\|u_h - \Pi_h u_{\text{ex}}\|_{H^1(\Omega)}}{\|\Pi_h u_{\text{ex}}\|_{H^1(\Omega)}}, \tag{6}$$

with respect to the mesh parameter h for various uniform mesh refinements. What is the order of convergence for both errors?