Numerical Methods for PDEs
MATH 610 - 600
A. Bonito

Programming Assigment 2
Spring 2018

Exercise 1 100%

FEM for 2D Boundary Value Problems Using Linear Elements

Consider the following elliptic boundary value problem: find $u \in H^1(\Omega)$ satisfying

$$\begin{aligned} -\Delta u + q\, u &= f, \qquad \text{in} \qquad \Omega, \\ \frac{\partial u}{\partial \nu} &= g, \qquad \text{on} \qquad \partial\Omega, \end{aligned}$$

where q is a given strictly positive parameter, $f \in L^2(\Omega)$ and ν is the outside pointing unit normal of Ω.
Solve the below given problems by approximating the corresponding boundary value problem using continuous piecewise linear finite elements on a partitions $\mathcal{T}$ given by *Triangle*

`http://www.cs.cmu.edu/~quake/triangle.html`

Consider meshes such that for any $\tau \in \mathcal{T}$, $|\tau| \leqslant 1/n^2$, $n = 10, 20, 40$, where $|\tau|$ is the area of τ. For each of the compuational examples below assemble the matrix in sparse format, solve the system using direct sparse solver (there is one in Matlab) and report the wall clock for the execution of the whole problem.

Computational examples

1. Take $\Omega = (0,1) \times (0,1)$, $q = 5$, $g = 0$ and $f(x,y) = (5 + 10\pi^2)\cos(\pi x)\cos(3\pi y)$ with exact solution $u(x,y) = \cos(\pi x)\cos(3\pi y)$. Report the $log - log$ plots of the L^2 and H^1 errors.
2. Take $q = 1$, $g = 1$, $f \equiv 1$, Ω is a polygon with vertices $(0,0)$, $(0.5,0)$, $(1,1)$, $(0,2)$. Plot the solutions obtained. What happens if you take instead $q = 0$?
3. The domain $\Omega = (0,1)^2 \setminus \overline{\Omega}_1$ where

   $$\overline{\Omega}_1 = \{|x - 0.5| \leqslant 0.25 \quad \text{and} \quad |y - 0.5| \leqslant 0.25\}.$$

   Take $q = 1$, $g = 1$, $f(x,y) = xy$. Plot the solution.

Usefull Commands

First of all run *./triangle –help* to see a detailed explanation on how *Triangle* works.
You can download from the class website the file *cube.poly*. It contains the data for *Triangle* to generate a subdivision of the unit square. To do so proceed as follow:

1. run *triangle -pqa0.2 cube*. The switch "-p" specify that a ".poly" file is read, the "-q" switch is a quality switch and the "-a" switch specify the maximum area (in this case 0.2).
2. run *showme cube.1* to see your newly created subdivision.

Triangle generates three files "cube.1.poly", "cube.1.node" and "cube.1.ele". You probably do not need the file "cube.1.poly". Except for the first and last line, the files "cube.1.ele" and "cube.1.node" contain the list of triangles identified by the index of its vertices and the coordinates of each vertex (the forth column specify the boundary marker, i.e. whether or not the node is at the boundary).
To import the generated data in matlab you can do the following

1. delete from each files generated by *Triangle* the lines you do not want (first lines and last line).
2. use the *load* command of matlab to store the content of the file in an array.