Math 5521 Computational Project 1

Due Spring 2018

The goal of this project is to test the convergence of the mixed FEM with Raviart-Thomas elements, using FreeFem++.

- Run the example LaplaceRT.edp in the examples++-tutorial directory and learn what it does. It is discussed in the FreeFem manual (see 9.1.7 in latest manual, or search PDF for LaplaceRT.edp). I found a typo in their discussion of the variational formulation. Also read about RT elements in the manual and other FreeFem basics, like mesh, fespace, problem, int1d, int2d commands.

- You will modify the code to solve various problems of the general form

$$
\begin{aligned}
-\nabla \cdot K \nabla p &= f \text{ in } \Omega, \\
p &= g \text{ on } \partial\Omega.
\end{aligned}
$$

- Use RT0 spaces. Note the option in the variational form for a penalization term

$$\epsilon \int_\Omega p\, q\, dx.$$

  Use this with $\epsilon = 1.0e - 8$, but get rid of the GMRES linear solver options and switch to 'solver=UMFPACK'. It will be much easier for larger problems.

- Make a copy of the code with name project1a.edp to solve with $K \equiv 1$, true solution $p(x, y) = x^3 + y^3$ on $\Omega = (0, 1)^2$. Choosing the true solution is a standard trick for testing code, called a "manufactured solution". It simply means that you plug your function p into the PDE and calculate what f must be. Use this f in your code; $g = p$ is obviously known on the boundary.

- In the same file, allow for the second option

$$K = \frac{1}{1 + 10(x^2 + y^2)}.$$

  You could have two sets of definitions for your data and just comment-out the one you are not using when you run the code.

- Make a copy named project1b.edp and change Ω to be the L-shaped domain obtained by removing the upper-right quarter from the unit square. Take $K = 1$, $f = 1$ and $g = 0$ (then you don't know the true solution). Note: there is code for the L-shaped domain in /examples++/demo.edp that you can use.

- Make a copy named project1c.edp with $\Omega = (0, 1)^2$, $f = 0$, $g = (1 - x)$ (on the boundary), and discontinuous permeability

$$K = \begin{cases} 100, & 0 < x, y < 1/2 \text{ or } 1/2 < x, y < 1 \\ 1, & \text{otherwise} \end{cases}$$

  Hint: func $K = 1 + 99 * (x < 0)$ evaluates to 100 if $x < 0$, 1 otherwise.

- For each problem above, define h to be the length of an element edge that lies on $\partial\Omega$, which we can make uniform, and run with $h = 1/10, 1/20, 1/40, 1/80$. Compute the errors

  - err(p,h)= $\|p - p_h\|_{L^2}$
  - err($\vec{u}$,h)=$\|\vec{u} - \vec{u}_h\|_{L^2}$
  - err($\nabla \cdot \vec{u}$,h)=$\|\nabla \cdot (\vec{u} - \vec{u}_h)\|_{L^2}$

  When you know the true solution, these can be computed directly. Otherwise, you use a "reference solution" on a fine grid. That is, compute using $h = 1/160$ in addition to the other h-values above and use the fine-mesh solution as your "true solution" to compute errors. In the integrals for the errors, integrate over the finer mesh. Note that you could solve first (once) on the reference mesh, then execute a loop to solve on the other meshes and compute the errors.

- Tabulate the errors and the convergence rates

$$\frac{\log\left(err(\cdot, 2h)/err(\cdot, h)\right)}{\log(2)}, \ h \leq \frac{1}{20}.$$

- Also plot the pressure errors for all cases (it is enough that your code shows the plots when run, you don't need to print them out). It helps to declare an extra function on the reference grid in the pressure space that holds the error; if you try to plot the difference of pressures on two different meshes directly in the plot command, FreeFem will complain.

- Write up a discussion of your results, speaking to the convergence behavior observed and the distributions of pressure errors.

- Submit your code and write-up to me. It is fine for your code to present some of your work for you upon execution, as long as the convergence data is organized in appearance (don't make me do work to figure out what your errors and rates are or I will not accept your submission).