MATH 6390 (Fall 2013) Project 5
Finite-Element Modeling of a Trace Gas Sensor
Due Thur Dec 19th 10am (NO EXCEPTIONS)

For this project, you may work in a group of TWO and submit a single co-authored report and python code.

The goal of this project is to model a photoacoustic spectroscopy trace gas sensor. The model is discussed in detail in the Lecture Notes on Finite-Element Modeling of Trace Gas Sensors Part I and Part II.

(A) Let D denote the unit disc, $D = \{x^2 + y^2 \leq 1\}$ in the plane and let ∂D be its boundary. Use FEniCS to find the eigenvalues $\lambda = k^2$ and eigenfunctions of the problem

$$\Delta u + \lambda u = 0 \qquad \text{in } D, \tag{1}$$

$$\frac{\partial u}{\partial n} = 0 \qquad \text{in } \partial D \tag{2}$$

by solving a generalized eigen-problem of the form $(A - \lambda B)\mathbf{U} = \mathbf{0}$.

You should create the mesh using the FEniCS function UnitCircle. For hints on how to use FEniCS to set up and solve the eigen-problem, see the FEniCS script demo_eigenvalue.py in the Ubuntu directory /usr/share/dolfin/demo/la/eigenvalue/python. You will also need to refer to the FEniCS Documentation for the SLEPcEigenSolver.

Compute **only** the 20 smallest eigenvalues, $\lambda = k^2$. (It would take a LONG time to compute all eigenvalues!) Include in your report labeled plots of eigenfunctions corresponding to the smallest 5 distinct nonzero eigenvalues.

Validate your eigenvalue computation by comparison to results obtained from the zeros of the derivatives of the Bessel-J functions. You can use the Matlab Exchange program zerobess.m to numerically compute values of k so that $J'_m(k) = 0$.

(B) Let $S(r) = \exp(-r^2/0.02)$. Use FEniCS to find the smallest eigenvalue λ_* of $-\Delta$ so that the solution of

$$\Delta u + \lambda u = S \qquad \text{in } D, \tag{3}$$

$$\frac{\partial u}{\partial n} = 0 \qquad \text{in } \partial D \tag{4}$$

gives a resonance of the system. To detect the presence of a resonance, you should choose a range of values for λ near an eigenvalue, λ_*, and calculate and plot the L^2-norm of the solution, u, as a function of λ. Explain why smaller eigenvalues than λ_* do not produce a resonance. (Hint: See pages 16 and 17 of the Lecture Notes.)

Note: To avoid potential problems calling FEniCS with different parameter values, I recommend defining a function CallFEniCS(Lambda) and calling this function from within a for-loop over possible values of Lambda.

See Section 1.1.11 of the FEniCS Tutorial for information on how to compute the L^2-norm

$$\|u\| = \sqrt{\int_D |u|^2 \, dx}. \tag{5}$$

(C) EXTRA CREDIT, since we are running out of time, though this is the point of the entire project. For the eigenvalue $\lambda_* = k_*^2$ you found in **(B)** use FEniCS to solve the problem

$$\Delta u \; + \; k_*^2 u \; = \; S \qquad \text{in } D, \tag{6}$$

$$\frac{\partial u}{\partial n} \; = \; ikAu \qquad \text{in } \partial D \tag{7}$$

where $A = a + ib = 10^{-3}(1+i)$. As in **(B)**, choose a range of values for λ near an eigenvalue, λ_*, and calculate and plot the L^2-norm of the solution, u, as a function of λ. What do you notice about the maximum value of this plot compared to that in **(B)**? What is the reason for the differences between the two plots?

Notice that the Robin boundary condition involves complex numbers. Since FEniCS does not support complex arithmetic, you will have to write all quantities in terms of their real and imaginary parts and derive a system of two PDEs. To see how to convert a system of two PDEs into a variational form see demo_mixed-posson.py in /usr/share/dolfin/demo/pde/mixed-poisson/python. In particular, you will need W = MixedFunctionSpace[V V] and you will need to use split().

Report

Submit a discussion of results as a pdf file produced using LaTeX and containing appropriate figures and tables. Also submit your FEniCS Python script(s).

Grading Scheme

(A) 15, **(B)** 15, **(C)** 20; Code comments and coding style 10, Discussion of results 10. Total 50 pts + 20 extra credit.