Each of the problems requires you to write some FEniCS code. Use the FEniCS programs I provided as models. Write neat, well organized code, and email me the complete source code and output for each problem. Besides generating the results I ask for, thoroughly debug your code on examples of your own construction. Do not hand these in, but make sure your code works!

1. Use FEniCS to solve the one-dimensional boundary value problem

$$-\epsilon^2 u'' + u = 2 \text{ on } (0,1), \quad u(0) = 1, \quad u'(1) = 0.$$

The exact solution is

$$u(x) = 2 - \frac{e^{-x/\epsilon} + e^{(x-2)/\epsilon}}{1 + e^{-2/\epsilon}}.$$

a) Plot the exact solution for $\epsilon = 0.1$ and $\epsilon = 0.001$ using FEniCS. To do this, define a FEniCS expression `uexp` for the solution and use the command `uex = interpolate(uexp, V)` to interpolate it into a Lagrange function space of degree 1 with, say, 200 elements, and then plot the result. You can use

```
fig = plot(uex)
fig.write_png('myfile')
```

to save the plot to myfile.png.

b) Write a FEniCS program with a routine `solvebvp(n, deg, epsilon)` which solves the boundary value problem using Lagrange finite elements of indicated degree on a uniform mesh of n elements. The return value from the routine should be u_h, the finite element solution (as a FEniCS `Function`). Using your routine, plot the approximate solution $u_h(x)$ obtained with Lagrange elements of degree 1 and $n = 50$ elements for both $\epsilon = 0.1$ and $\epsilon = 0.001$. Compare with the plots in a).

c) For $\epsilon = 0.001$ make a table showing the value of u_h and the error $u - u_h$ at $x = 0.05$ for Lagrange elements of degree 1 and Lagrange elements of degree 4 with n = 10, 20, 40, and 80.

2. Modify the program poisson_convergence2.py to study the convergence of the finite element method for the following mixed boundary value problem. The domain is the rectangle $(-1,1) \times (0,1)$. The PDE is $-\Delta u = 0$. On the left half of the portion of the boundary lying along the x-axis, i.e., on the interval $-1 < x < 0$, $y = 0$ the boundary condition is $\partial u/\partial n = 0$. On the remainder of the boundary the Dirichlet condition $u = u_0$ is imposed, where u_0 is the exact solution, given in polar coordinates by

$$u_0 = r^{1/2} \sin(\theta/2) \text{ on } \Omega.$$

a) Use Lagrange elements of degree 1 with a mesh of the size $(2n \times n) \times 2$ elements with $n = 8, 16, \ldots, 128$. For the "exact solution" needed to compute errors, use a mesh of $(256 \times 128) \times 2$

elements with polynomials of degree 5. Turn in the resulting table (and observe the rates of convergence).

b) Do the same with Lagrange elements of degree 2. How do the rates of convergence change when the degree is increased?

3. Use mshr to make a mesh of the domain consisting of the square $(-1, 1) \times (-1, 1)$ minus a disk around the origin of radius 0.1. Solve the problem $-\Delta u = 1$ on this domain with the Dirichlet condition $u = 0$ on the boundary of the disk and, on the boundary of the square, the Robin boundary condition

$$\frac{\partial u}{\partial n} + 2u = 2.$$

Adjust the parameter n in `generate_mesh(domain, n)` to obtain a mesh with $30,000$ elements (± 1000 elements), and use Lagrange cubics. (Note: The square may be constructed in mshr with `Rectangle(Point(-1., -1.), Point(1., 1.))`).

a) Plot the solution.

b) What is the maximum value obtained by the solution at a DOF?