Assignment 5

ISC 4232/5935

Due: November 29 at 11:59pm

Programming Language: any sensible language
Submit write-up, code, and any auxiliary files to bquaife@fsu.edu
Undergraduates: Submit questions except 2d, 2e, and 2f
Graduates: Submit all questions

---

1. The goal of this question is to write a FEM code for the two-point BVP

$$-\frac{d}{dx}\left(p(x)\frac{du}{dx}\right) + q(x)u(x) = f(x), \quad x \in (a, b),$$

with homogeneous Dirichlet conditions $u(0) = u(1) = 0$, and a uniform grid. Your code will also compute the norms of the errors of the solution's derivative $\|(u - u^h)'\|$ and the solution $\|u - u_h\|$.

You should have a main routine, say `fem1d` which should call appropriate functions to set up the geometry, the coefficient matrix and right hand side, solve the linear system, and then call an error routine to compute the norm of the error. Each routine will be written and tested separately.

   (a) The first step is to write a geometry routine which sets up the coordinate of each node, an array which associates local nodes to global nodes, and arrays for applying the quadrature rule. On the first and last elements, there will only be one local node because of the homogeneous Dirichlet boundary condition. Write a routine which satisfies the following criteria.
   Input:

   - left and right endpoints of domain
   - number of nodes in x-direction

   Output:

   - number of elements
   - number of unknowns
   - an array which gives the x-coordinate for each node
   - an array which associates each local node of an element with its global node; the array will be dimensioned by the number of elements and the number of local nodes per element (2 for linear elements)
   - an array which gives the area of each element to be used as the weight in the quadrature rule; for a uniform grid, this array will be constant.
   - an array which gives the quadrature point in each element; use a one-point Gauss rule; array should be dimensioned by the number of elements.

   Set up all the arrays for a uniform grid with 9 nodes on [0, 2]. Print out the number of elements and number of unknowns and all of your arrays.

(b) The next step is to write a routine to evaluate the linear basis function at a given point. Write a routine which works for a uniform grid and satisfies the following criteria
Input:

- the point x where you want to evaluate the basis function
- the node number the basis function is centered at, i.e., where it is one
- an array of the x-coordinates of each node

Output:

- the value of the basis function at the given point
- the value of the derivative of the basis function at the given point.

Test your code by evaluating the basis function and its derivative centered at $x = 0.5$ at the points $x = 0.25, 0.375, 0.5, 0.675, 0.75$ using the grid set up in Part 1a. Print your results.

(c) We are now ready to write a routine which assembles the matrix using a one-point Gauss quadrature rule. The routine should be written so that it first calculates all contributions to the matrix and right hand side over the first element, then the second element, etc. The basic structure of the assembly routine should be as follows:

- loop over the number of elements
- loop over the number of quadrature points
- loop over the number of local nodes to determine test function and equation number
- loop over the number of local nodes to determine trial function and unknown or column number
- compute entry, say a_{ij}, times the quadrature weight and add into appropriate position of matrix.

For the general two-point BVP with $p(x) = 1$ and $q(x) = 1$ we know the exact entries of the coefficient matrix for a uniform grid; these are given in the notes. Output your matrix for your uniform grid in Part 1a and compare with the exact value. Are they the same? Why or why not?

(d) The right hand side of our linear system can be assembled in an analogous way to the matrix. Modify your assembly routine to assemble the right hand side. The routine should now output your right hand side and matrix. Print out your right hand side vector for $f(x) = 1$ for your uniform grid in Part 1a and compare with the exact answer.

2. Apply your code to solve

$$\begin{aligned} -u''(x) &= \pi^2 \sin(\pi x) \quad x \in (0, 1), \\ u(0) &= 0, \\ u(1) &= 0. \end{aligned}$$

3. The last routine that we need is the error routine so that we can perform a convergence study from Question 2. Write a routine to compute the integral of the square of the error, i.e.,

$$\int_a^b (u(x) - u^h(x))^2 dx.$$

This integral over the entire domain should be computed over each interval and summed so that the structure of the code will be similar to that of the assembly—we loop over the number of elements, then the number of quadrature points. Computing the exact solution $u(x)$ at the quadrature point is trivial but we also need to know the value of u^h at the quadrature points. These values can be evaluated by computing

$$u^h(x) = \sum_{j=1}^{n} \mu_j \phi_j(x)$$

However, we know that the only terms which contribute to this sum are the particular basis functions which are nonzero over the current element; these are just two basis functions—the one associated with the left endpoint of the element and the right endpoint of the element. Consequently to calculate uh we need to loop over the local nodes associated with that element and compute the two terms in the series. Write the error routine using a two-point Gauss quadrature rule for evaluating the integral. Test your error routine with the problem in Question 2. Solve the problem for $h = 1/4, 1/8, \ldots, 1/64$, output your errors, and compute the rate of convergence. Make sure that you are achieving second-order convergence.

4. (**Grad Students Only**) Modify your code to use continuous piecewise quadratic basis functions and use the two-point Gaussian quadrature rule to compute the matrix and vector entries. Test your results for the problem in 2. Output your results as before and compare with using linear basis functions.