University of California, Berkeley
CEE C133/ME C180, Engineering Analysis Using the Finite Element Method
Spring 2010
Instructor: S. Govindjee
GSI: N. Hodge

Labs 4, 5: Programming 1D Finite Elements, Part 1

This week, you will start writing your own FE code. For concreteness, and the ability to generate exact solutions, we will be modeling a canonical problem: the deformation of a beam with body forces, and tractions. The strong form is

$$\begin{aligned} \frac{\partial}{\partial x}\left(A(x)E(x)\frac{\partial u}{\partial x}\right) + b(x) = 0, &\quad \text{in } \Omega = (0, L), \\ u = \bar{u}(x), &\quad \text{on } \Gamma_u, \\ AE\frac{\partial u}{\partial x} = \bar{F}(x), &\quad \text{on } \Gamma_q. \end{aligned}$$

This assignment will count for two labs. If you want to finish it all this week, feel free. The following lists contain some hints to help you along the way.

Week 1:

- The first step when doing these kinds of things is always to write out your solution method on paper. In this case, if you take a bit of care when writing out the analytical weak form, you will save yourself a lot of trouble.

  One part of this is accounting in the most general possible way for the boundary conditions. Since the weak form explicitly contains terms related to the force boundary conditions, the most general formulation should contain these, and then zeroed out if they are not present in any particular problem. While changing the displacement boundary conditions changes the space of trial solutions, it does not explicitly change any of the terms in the weak form. Accounting for the existence of displacement boundary conditions will be done in week 2, when dealing with the solution for the system of equations.

- I am a firm believer that writing things out as generally a possible, but in this case, it might be useful to be a bit more specific. So, note that you are to do this using *nel* linear elements (i.e., *nel* should be a modifiable parameter in your code), with the elements formulated in physical space, i.e., *not* in isoparametric (i.e., ξ) space. This will make your discretization of the weak form a bit easier.

- The next step is to set up the data structures and organize your code. I would recommend the following:

- The basic data needed is the location of the nodes (a 1D array), and the mapping between the elements/local node numbers to the global node numbers, LM (a 2D array).
- I would strongly suggest you break the calculation of the interpolation functions and their derivatives out into separate functions, in order to make it easier to make changes later in the semester.

  Likewise, it probably makes sense to break the calculation of A, E, and b into separate functions, so that you can handle expressions that vary in x.

- The next step is to loop over the elements to calculate $\mathbf{K}^e$ and $\mathbf{F}^e$. This actually will involve two loops (maybe): a loop over the elements, and an inner loop over the quadrature points (this may not be a loop if you use something like Simpson's rule, which has a small, fixed number of integration points). You can program any integration rule you want: midpoint, trapezoidal, Simpson, Gauss, whatever; you *cannot* use a built-in MATLAB function.

  For anyone who is interested, I have posted a MATLAB file to bspace called *gaussquad.m*. This provides the points and weights for various orders of Gauss quadrature up to 20.

Your assignment for week 1 is to turn in the following to me the following week in lab:

- Your derivations of the analytical and discretized weak forms.
- A printout of the element stiffness and force matrices for $L = 1$, $A = 1$, $E = 1 + x$, $b = 1$, $\bar{F} = 1$ on $\Gamma_q = \{0\}$, and *nel* (number of elements) $= 2$.

Week 2:

- The next step is to assemble the element contributions to the stiffness into the global stiffness. In general, this works as follows: for the element stiffness for a linear 1D element,

$$\mathbf{K}^e = \begin{bmatrix} K^e_{11} & K^e_{12} \\ K^e_{21} & K^e_{22} \end{bmatrix}$$

  and an LM matrix for a system of two elements,

$$LM = \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 2 & 3 \\ \hline \end{array}$$

  the components of the element stiffness array for $e = 2$ map to the global stiffness components as follows:

$$\mathbf{K}^{e=2} = \begin{bmatrix} K_{11}^{e=2} & K_{12}^{e=2} \\ K_{21}^{e=2} & K_{22}^{e=2} \end{bmatrix} \Rightarrow \begin{bmatrix} K_{LM(1,e=2),LM(1,e=2)} & K_{LM(1,e=2),LM(2,e=2)} \\ K_{LM(2,e=2),LM(1,e=2)} & K_{LM(2,e=2),LM(2,e=2)} \end{bmatrix} = \begin{bmatrix} K_{22} & K_{23} \\ K_{32} & K_{33} \end{bmatrix}.$$

Make sure that you have the "random" mesh type selected when you are testing the assembly of your element-wise stiffness matrices.

- Now, we need to apply the Dirichlet BCs. One way to do this is as follows:
  - Subtract the product of the column of $\mathbf{K}$ corresponding to the node at which the Dirichlet BC is applied times the value of the Dirichlet BC from $\mathbf{F}$.
  - Strike the row and column corresponding to the Dirichlet BC from the system.
  - Set the appropriate value of $\hat{\mathbf{u}}$ equal to the Dirichlet BC.
  - Writing out a simple example by hand might help you see what is going on here.

  However, this is not quite the way that we have implemented this in our code.
- Solve the system. Place the solved-for degrees of freedom into the appropriate locations of the solution vector $\hat{\mathbf{u}}$, if you used the first method above to apply the Dirichlet BCs (which we didn't).

Your assignment for week 2 (due the following week in lab) is as follows:

- Send me your code via email. You will be graded on, among other things, the assembly of the element stiffness matrices, so do this properly.
- Give or send me a *clear* and *complete* description of the (alternate) method that we are using to implement the Dirichlet BCs.
- Plot your solution versus the exact solution for the following data:
  - $nel = 1, 5, 10, 50$
  - $L = 10m$
  - The cross-section is circular, with radius of $100mm$.
  - The stiffness is $E = (E_0 + E_1 x)\,\frac{N}{m^2}$, where $|E_0| = |E_1| = 7e10$.
  - $u = 0m$ on $\Gamma_u = \{0\}$
  - $AE\frac{\partial u}{\partial x} = 500N$ on $\Gamma_q = \{L\}$
  - $b = (500 + 10x)\,N$.