The Finite Element Method for Problems in Physics

Coding Assignment 1

Consider the following differential equation of elastostatics, in strong form:

Find u satisfying

$$(E\,A\,u_{,x})_{,x} + f\,A = 0, \quad \text{in } (0, L),$$

for the following sets of boundary conditions and forcing function ($\bar{f}$ is a constant):

(i) $u(0) = g_1$, $u(L) = g_2$, $f = \bar{f}x$,

(ii) $u(0) = g_1$, $EAu_{,x} = h$ at $x = L$, $f = \bar{f}x$,

where $E = 10^{11}$ Pa, $A = 10^{-4}$ m^2, $\bar{f} = 10^{11}$ Nm^{-4}, $L = 0.1$ m, $g_1 = 0$, $g_2 = 0.001$ m, and $h = 10^6$ N.

Coding Instructions: Write a one-dimensional finite element code in C++ using the deal.II FEM library frameswork and the template provided to solve the given problem, following these requirements:

- Code (a) linear, (b) quadratic and (c) cubic order Lagrange polynomial basis functions.
- Include a function to calculate the L^2 norm of the error between the finite element solution (u^h) with the exact solution (u), given by $\sqrt{\int_\Omega (u - u^h)^2\,dx}$.
- All integration in K_{local}, F_{local}, and the L^2 norm of the error should be done by Gaussian quadrature (see Lecture 4.11), instead of using the analytical solution to the integrals shown in the lectures.
- For submission, use a 100 element mesh. (You will be provided with the six .vtk solution files for a 10 element mesh to help you debug your code. A table of the corresponding L^2 norms is included at the end of the instructions.)

You will be provided with four files to create and run your code (please refer to the recorded tutorials for an in-depth explanation):

- main1.cc (the source file that defines and solve the boundary value problem using your C++ class)
- FEM1.h (the template header file defining your finite element method C++ class)
- writeSolutions.h (used to create the solution files you will submit)
- CMakeLists.txt (used to create a Makefile to run your code, as shown in the first segment of the recorded introduction to C++)

Almost all of your coding will be done in FEM1.h, but do not modify any function names or the names of any class data structure. The only parts of main1.cc that should be modified are the function inputs that define the basis function order (1, 2, or 3), the problem number (1 or 2) which specifies which set of boundary conditions to use, and the number of elements in the mesh. Your FEM1.h file must run with the given main1.cc file. FEM1.h should calculate the results for part (i) or (ii), with linear, quadratic, or cubic basis functions, based on the inputs from main1.cc (i.e. FEM1.h should use "if" statements to determine which

	Part (i)	Part (ii)
Linear	1.66468e-07	1.66468e-07
Quadratic	1.81848e-09	1.81848e-09
Cubic	machine	precision

Table 1: L^2 norm of the error for a 10 element mesh.

problem to solve and what basis function order to use; do not create additional .h files for each problem and basis function order). Nothing should be changed in writeSolutions.h or CMakeLists.txt.

The convergence of the solution with respect to the domain discretization can be studied by varying the number of elements and finding the finite element solutions (e.g. 3, 10 , 100 and 1000 elements). Consider the slope of the log-log plot of the L^2 norm of error versus the reciprocal of the element length, $1/h_e$. Why might you not achieve an optimal convergence slope for higher order basis functions or a high number of elements?

Submission Instructions: Use a 100 element mesh for your submission. You should submit (through the Coursera website) a .zip file (name it CA1.zip) containing your FEM1.h file and the .h5 solution files. You will have a total of six .h5 solution files (using linear, quadratic, and cubic basis functions for problems (i) and (ii)). Each .h5 file will contain the solution vector and the L^2 norm of the error. Do not submit any files other than those listed here.