ME 471/AE 420/CSE 451: Programming Assignment 3
Spring 2017

Due: Friday, March 10, 2017 at 11:59pm (subversion)

General Instructions for Programming Assignments

To complete your submission, follow the steps below:

1. Go to your working directory (for example, cd ME471-Programming-Assignments)
2. Download assignment

   (svn checkout https://subversion.ews.illinois.edu/svn/sp17-me471/your_netid/Assignment-Folder-Name) or (svn update) if you already have a working directory
3. Write your FEA code
4. In case you create new files (.m, .cpp, .h), you will need to use svn add (svn add schedule files and directories in your working copy for addition to the repository.)
5. Upload the changes (svn commit -m "COMMIT_MESSAGE")

Before you commit your work, make sure all the files are following these guidelines:

1. Matlab users:
   (a) Do not change the name of the main file (MainFile.m). The grading script will execute this file.
   (b) Do not modify the following lines in the main file:
   // ==========
   // DO NOT MODIFY THE LINE BELOW!! //Autograding script will search for this variable definition
   filename = 'input.dat';
   // ==========
   Of course, you are free to modify the name of the input file when working on your local machine, but make sure the filename variable is set to 'input.dat' before you commit.
   (c) Do not delete the contents of the C-Code folder (mainly the Makefile file)
2. For C++ users:
   (a) Do not modify the variable EXENAME inside the Makefile. The grading script will execute the file defined by EXENAME.
   (b) Do not modify the following lines in the main file:
   // ==========
   // DO NOT MODIFY THE LINE BELOW!! //Autograding script will search for this variable definition
   string filename = "input.dat";
   // ==========
   Of course, you are free to modify the name of the input file when working on your local machine, but make sure the filename variable is set to 'input.dat' before you commit.

3. IMPORTANT: Do not modify the "PrintEQNUM", "PrintMatrices" and "PrintSolution" functions. Make sure they are positioned only after the variables for the functions are defined and contain the desired values. "PrintEQNUM" should come after equation module, "PrintMatrices" should come after assembly module, and "PrintSolution" should come after solve module. This will ensure your assignment will be graded properly.

It is good practice to commit regularly and frequently. For example, commit when you are done writing a function. This allows both simpler commit messages and greater confidence in the repository.

Instructions

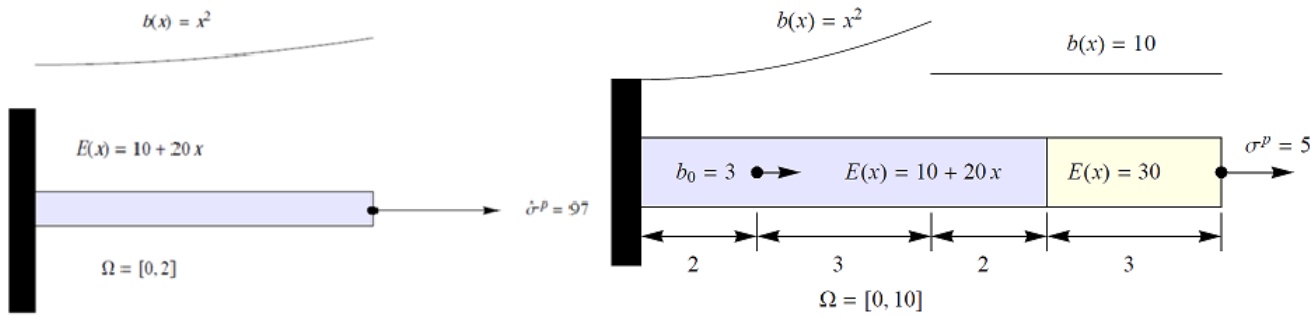
Download (checkout) the assignment folder **03-FEA-1D-IsoparametricElements**. Modify your 1D finite element program to approximate the response of a one-dimensional elastostatic system using isoparametric elements, as discussed in lectures 13-14. Your code should also accommodate heterogeneous modulus of elasticity $E(x)$ and non-uniform body load $b(x)$.

Your program should read a file named "input.dat" that has the format depicted in Fig.1 and also discussed in lecture. The program should use the same "PrintOutput" functions used in previous assignments.

You should also be able to reproduce the results depicted in Figure 2. The results were obtained using six quadratic elements of the same size ($h = 1$), where the center node is located in the middle of the element (even node spacing for each element). Figure 2 shows the final results using 2 and 3 gauss quadrature points for the numerical integration. Similar to other programming assignments, we will be checking your code using a different 1D rod configuration. For this assignment, we will use two different input files to check your code. The grade will be the average between the two of them.

Grading schedule

We will start running the grading script from Tuesday Feb 28th at 02:00 pm and 00:01 am until Friday March 3rd and from Tuesday March 7th 02:00 pm and 00:01 am until Friday March 10th. In case you don't get a satisfactory score, you can change your code, make another commit, and your code will be re-graded. You final score will be the score of your last run.



Input.dat for domain [0,2]

```
5  2  1  1  2  2
0.0
0.2
0.7
1.3
2.0
1  2  3  UMAT  DLOAD
3  4  5  UMAT  DLOAD
5  97
1  0.0
```

```
UUR =  0
       1.68355
       4.34477
       6.35695
       7.96953
```

For domain[0,10]

Results for 4 linear elements, Ng=2

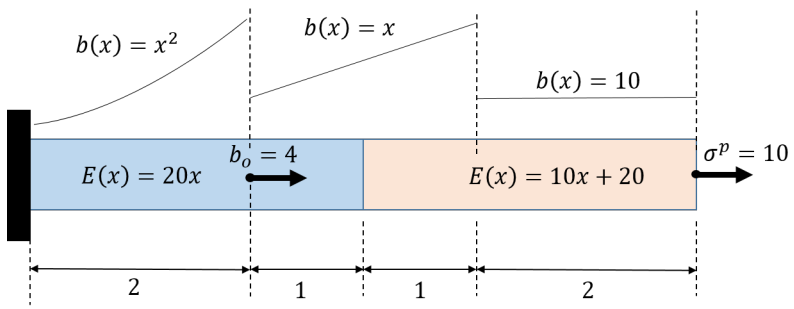
```
UUR =
  0.00000000
  6.60000000
  9.59062500
 10.28293269
 12.28293269

PUR =
-99.66666667
 19.25000000
 34.75000000
 25.00000000
 20.00000000
```

Results for 4 quadratic elements, Ng=2

```
NODE DISPLACEMENT
  0.00000000
  5.20289855
  7.77101449
  9.87378089
 11.00462105
 11.42311012
 11.71038646
 13.08538646
 13.71038646
NODE FORCE
-99.66666667
  1.77777778
  5.36111111
 26.00000000
 15.08333333
 13.33333333
  8.33333333
 20.00000000
 10.00000000
```

Figure 1: Left box: Input file for the region [0,2] and displacement results; intermediate results for this same domain appear in lecture notes. Right box: Final results for the (0,10) domain when using 4 linear and quadratic elements, and 2 integration points. The mesh with quadratic elements uses the following nodal locations: (0.0, 1.0, 2.0, 3.5, 5.0, 6.0, 7.0, 8.5, 10.0). Analytical solution for the entire rod in HW#2.



Ng=3

```
UUR =
  0.00000000
  4.80000000
  6.39500000
  7.24346154
  7.83307692
  8.22860057
  8.54084719
  8.85269112
  9.12220078
  9.34255652
  9.50955256
  9.63047244
  9.71133298

FUR =
 -42.66666667
  0.20000000
  0.30000000
  1.53333333
  4.98333333
  1.66666667
  1.00000000
  2.33333333
  2.33333333
  6.66666667
  3.33333333
  6.66666667
 11.66666667
```

Ng=2

```
UUR =
  0.00000000
  4.80104167
  6.39583333
  7.24447115
  7.83397436
  8.22949801
  8.54174463
  8.85358855
  9.12309822
  9.34345395
  9.51045000
  9.63136988
  9.71223041

FUR =
 -42.66666667
  0.22222222
  0.27777778
  1.55555556
  4.97222222
  1.66666667
  1.00000000
  2.33333333
  2.33333333
  6.66666667
  3.33333333
  6.66666667
 11.66666667
```

Figure 2: Another example to help you debug the code. Domain discretized into six quadratic elements of the same size ($h = 1$). Final results using 2 and 3 integration points.