ME 471/AE 420/CSE 451: Programming Assignment 5
Spring 2017

Due: Friday, April 21, 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)
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", "PrintSolution" and "PrintPostProcessingResults" functions. Make sure they are positioned only after the variables for the functions are defined and contain the desired values. For instance, "PrintEQNUM" should come after equation module, "PrintMatrices" should come after assembly module, "PrintSolution" should come after solve module and "PrintPostProcessingResults" should come after computing the fluxes (at element and nodes). This will ensure your assignment to be graded properly. While it is entirely your choice to leave all the print functions towards the end of your code, it is highly recommended to follow the flow we have suggested, as this will allow partial credit for the results generated up to the point right before your code crashes.

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 **05-FEA-Heat**. You should use PA4 as a starting point for your 2D heat transfer FEA program.

Your program should read a file named "input.dat" using the distributed "ReadInput" function. The program should also create the output files using the distributed output functions as described above. With your distribution, you will receive 4 different input files and all the output files inside folders Example1, Example2, etc. The problem descriptions are available on the website under the Schedule page. Similar to other programming assignments, we will be checking your code using a different configuration.

As in previous assignments, your FEA program will calculate the value of nodal temperatures. You will receive a function called "EvaluateGradientField" that evaluates heat flux at nodes, element centroids and integration points, as discussed in lecture. You will need to complete some missing parts for this function to work (as indicated in the file).

You will also receive in your distribution a couple of Matlab post-processing files to help you visualize the results: "PlotVectors" and "PlotFieldonMesh". **Before you commit your program, you need to make sure these functions are NOT invoked, since the grading script runs on a terminal without access to display functionality. Make sure to comment the corresponding lines.**

Grading schedule

We will start running the grading script from Tuesday April 18th at 02:00 pm and 00:01 am until Friday April 21st. 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.