CS601: Software Development for Scientific Computing

Programming Assignment 2 - Software for Computation on Grids
Due: 16/11/2024

This is an individual assignment. The objective of this assignment is to familiriaze with grid codes and develop a generalized software solution for computing on grids.

1 Problem Statement

There are two parts to this assignment:

- **Part I** The questions are indicated in `FEMain.cpp` provided as part of the starter files. As an additional exercise, you must write code comments, in all the files in `inc` and `src` folders provided and generate documentation via Doxygen.
- **Part II**:
  1. Design and implement a class `RDomain` as a *sub-class* of the `Domain` class outlined in the lecture. Add methods and attributes to the `RDomain` class that you think are necessary. The class should have the capability of generating simple grids on the domain (Rectangular and 1D). An outline of the `Domain` is given below:

```
1    class Domain{
2    private:
3    //...
4    public:
5    //..
6    virtual void PrintGrid(string outputFileName) const=0;
7    //...
8    };
```

     - Using an object of the `RDomain` class, a user must be able to print the grid to a file via the member function `PrintGrid`. Do NOT change the signature of `PrintGrid` in your implementation. While implementing the `PrintGrid` function, use the File-IO functions `fopen`, `fwrite`, and `fclose` (or the corresponding `ofstream` class) to output the grid coordinates in *binary* format. You should write all the X-coordinates first followed by all the Y-coordinates. Do not write anything else to the file. Refer to the man pages of `fopen` and `fwrite` for details. Example usage of File-IO functions is given below:

```
1    #include<cstdio>
2    FILE *fp;
3    /*initialize the file handle to open the file outfile.bin in binary
         mode for writing.*/
4    fp =fopen("outfile.bin","wb");
5    /*write m double-precision values stored at contiguous memory
         locations starting from x*/
6    fwrite(x,sizeof(double),m,fp);
7    fclose(fp); //close the file handle
```

     - Using an object of the `RDomain` class, a user must be able to specify the step size(s) of the space variables.
  2. Design and implement a class `GridFn` for implementing the grid function modeling the 1D heat diffusion problem discussed in class (refer slides). Use the three-point stencil. Also, assume that that the two ends of the metal bar are held at constant temperature of 0 and the initial temperatre distribution is given by $f(x) = x\sqrt{(l-x)^3}$. Note that the time-step δt should neither be part

of the grid (`RDomain` class) nor the grid function(`GridFn` class). Instead, time-step should be a higher-level parameter.

3. Design and implement a class `Solution` for implementing the numerical computation of approximating the solving of a PDE over a grid. Specifically, implement your `Solution` class using the classes designed in 1 and 2. For a test case, assume that α, thermal diffusivity, is 1, l(length of the rod)=1.2, δx=0.4, and δt=0.1 as a test case. Design your classes in such a way that the following outline of the `Solution` is possible:

```
1       Create domain
2       Create grid function to operate on a domain
3       Create a solution and prepare to compute the solution
4       Set initial conditions
5       Iterate:
6         compute the grid function till the solution converges or a maximum
              number of steps has been reached.
7         (optional) Set boundary conditions
```

2 What you need to do and submit

- clone the $PA2$ repository that contains all the files that you need to get started with your assignment (link provided in the discussion forum).
- Modify the `Makefile` provided to print team info upon executing `make team` command.
- For part I, modify `FEMain.cpp` and all other files with your code comments. Also, `make doc` should build the documentation.
- For part II, include all your source code as briefed earlier. Modify the `Makefile` provided such that `make part2` should build your code.
- A shell script (this must be written in bash) called `runme` that builds your Part II using the `Makefile` and executes it using the parameters provided. This script should take in three arguments: first, the input length of the rod, l, then the time-step δt, and space-step δx.

You must tag your source code and submit as you have done previously. The tag name to be used is: ***`cs601pa2submission`****. All tag names are case-sensitive..*