Project 1: 2 D Unsteady Heat Equation Solver

Parallel Computing for Computational Mechanics Summer Term 2020

1 What worked well?

- implementing the K, F, and M element matrices
- mass lumping
- RMS value calculation
- error value calculation

2 Common Mistakes Coding

2.1 CalculateElementMatrices function

- The radius of the source term region was not specified correctly (wrong value)

```
for(int i=0; i<nen; i++)
{
    node = mesh->getElem(e)->getConn(i);
    x = xyz[node * nsd + xsd];
    y = xyz[node * nsd + ysd];
    radius = sqrt(pow(x,2) + pow(y,2));
    // specify radius here:
    if(radius < (rFlux - 1e-10))
    {
        mesh->getElem(e)->setF(i, F[i]);
    }
    mesh->getElem(e)->setM(i, M[i][i]);
    for(int j=0; j<nen; j++)
    {
        mesh->getElem(e)->setK(i,j,K[i][j]);
    }
}
```

2.2 ApplyDrichletBC function

- bigger/lesser equals comparison was used without a threshold

```
if (settings->getBC(1)->getType()==1)
{
    for(int i=0; i<nn; i++)
    {
        x = xyz[i*nsd+xsd];
        y = xyz[i*nsd+ysd];
        radius = sqrt(pow(x,2) + pow(y,2));
        // Introduce threshold of 1e-10 here:
        if (abs(radius-rOuter) <= 1E-10)
        {
            if(settings->getBC(1)->getType()==1)
```

```
            {
                mesh->getNode(i)->setBCtype(1);
                T[i] = settings->getBC(1)->getValue1();
            }
        }
        else
        {
            this->nnSolved += 1;
        }
```

2.3 explicitSolver function

- Residual was not divided by number of non-Dirichlet nodes

```
partialL2error = 0.0;
globalL2error = 0.0;
for(int i=0; i<nn; i++)
{
    pNode = mesh->getNode(i);
    if(pNode->getBCtype() != 1)
    {
        massTmp = massG[i];

        MT = MTnew[i];

        Tnew = MT/massTmp;

        partialL2error += pow(T[i]-Tnew,2);

        T[i] = Tnew;
        MTnew[i] = 0;
    }
}
// here you should divide by number of non-Dirichlet nodes
globalL2error = sqrt(partialL2error/this->nnSolved);
```

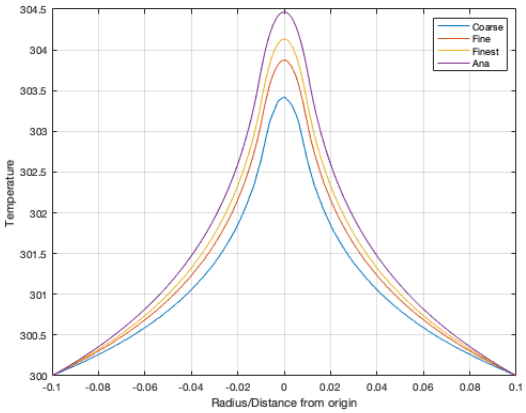
2.4 CompareAnalytical function

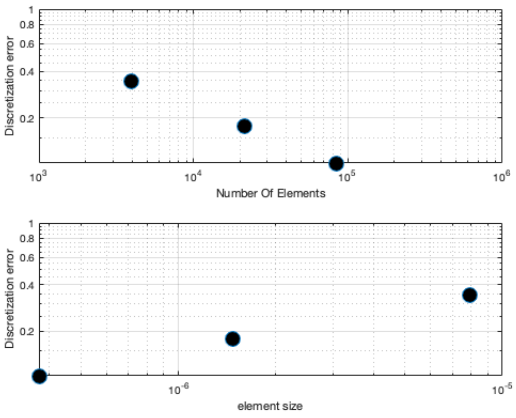
- discretization error was not divided by number of non-Dirichlet nodes

3 Common Mistakes Report

- Entire Sections of Text were copied from the assignment (Plagiarism, if you do not mark the cited sections)
- The convergence was not documented either in a plot, table or in written words.
- The methods were not explained.
- Code excerpts were presented without explaining what they are supposed to do or why you take a certain approach.

- Formulas were not embedded properly. A formula/mathematical equation can be treated like a sentence. When it is introduced in the text before, you start with a colon, you end it with a period. If you have multiple equations, each of them is separated by commas.
- Figures did not present the relevant info. The minimum should have the following two plots plus furthermore a view of the temperature distribution on the disk from paraview for each of the meshes.





4 Answer to Questions

– Q1: You should not assign a strict value of 0.1 or 1 when doing a smaller equals or larger equals comparison but allow for a threshold, as due to discretization (and even smaller) machine precision, nodes that are considered boundary nodes could be slightly off of the radius but still be the outer and therefore nominally the boundary node. You should mention the mesh size here.

– Q2: When increasing the time size, the simulation crashes at a certain value. This is due to the time discretization being explicit and not unconditionally stable. One helpful number is the critical CFL number that can lead to the simulation results diverging. You should mention what happens and WHY this happens.

5 IMPORTANT: FOLDER STRUCTURE FOR PROJECT 2

We will strictly enforce that the folder structure for project 2 is identical to what I will provide here now, as we need a consistent structure for running your code and grading your projects. You are supposed to hand one (1) single tar file, which is named **pr2.tar.gz**. This file should, when extracted contain one (1) pdf of your report (you can name that however you like) and a subfolder with the code named "code". In this subfolder code, please create two subfolders, one for each instance of code you are using to prevent data races asked in Question 3 of the assignment. Inside each of those codeN folders should then go all the files that are inside the folder *skeleton* that I uploaded to moodle with the changes you made to the CmakeLists.txt and the .cpp files in question. Each of these folders should be compilable and run on its own. You can just copy and paste all the files that you have not changed. Do not rename files if that prohibits the make process. You can use the same, optimized CmakeLists.txt with the flags you came up with for all the code subfolders. Make sure your code is compilable, this is a requirement.
So after unpacking **pr2.tar.gz**, the folder should look like this://

```
Report.pdf
Code
        Code1
                2D_Unsteady_Diffusion.cpp
                CMakeLists.txt
                ...
                test
                        run.j
                        settings.coarse.in
                        settings.fine.in
                        settings.finest.in
                        ...
        Code2
                2D_Unsteady_Diffusion.cpp
                CMakeLists.tx
                ...
                test
                        run.j
                        settings.coarse.in
                        settings.fine.in
                        settings.finest.in
                        ...
```