Christian Rowsell (40131393)

Mech 588 – UBC Mechanical Engineering Winter 2023

Unsteady 2D Heat Equation on Curvilinear Mesh

Mech 588 Assignment 1

Programming Assignment:

2D Heat Equation in General Coordinates

Mech 588 — Spring, 2023

Due Date: February 10

This problem addresses issues that arise in solving the unsteady heat conduction equation, *Tt* =∇2*T*, on a two-dimensional non-uniform mesh.

1. Write a verification plan. Be sure to include tests for all of the key parts of your code, as well as global tests to confirm that your code correctly solves the PDE. Turn in your verification plan as part of your report, but you do not need to submit the results of your verification tests; if your code doesn’t pass a reasonable set of verification cases, it won’t solve the final problem, either.

The code can be broken up into the following broad categories. Each of these categories will be tested individually, and once they all work the separate pieces can be combined into a master code which will be tested further. To allow for easier testing later on, a data structure was created to store the solution variables, and output the value as a .vtk file. This allows for visualization of the solution vectors using Paraview, which can help detect bugs later on. A class named Solution is created, which stores the x and y coordinates, as well as the velocity and temperature values for each data point in the grid. A constructor was made which allocates memory for each of these vectors, and initializes the values to be 0 at every point.

1. Write and verify a code that solves the heat equation to second-order accuracy in both time and space. You should use implicit time advance.
2. The geometry for the final non-rectilinear mesh problem is shown below. The boundary conditions are as follows:
   * Along A, B, and C, *T* = 0.
   * Along D, *T* =−4(*y*−2)(*y*−3). This is a parabola with value of 0 at *y* = 2 and 3 and 1 at *y* = 5*/*2.

The initial condition for the problem is 0.

Diagram

Description automatically generated

Meshes will be provided on Canvas in 32×8, 64×16, and 128×32 sizes. In each case, *i* will run “tangentially” from *y* = 0 around to *x* = 0 and *j* will run radially from inside to outside. Data will be given for *x* and *y* for all cell corners on or inside the domain boundary.

Using these meshes, compute the solution *T* at time *t* = 0*.*1, and provide convincing evidence that your scheme is second-order accurate. In your write-up, be clear about what solver parameters you used to produce results that you present. (Roughly speaking, provide me with all the information I would need to be able to reproduce your results using your code if I were so inclined.)