Joyce Huang
sh2393

ECE 4960 Programming Assignment 5

Objective

The goal of this project is to implement a finite difference solver for diffusion equation for both the 1D and 2D system that can be described by a set of partial differential equations.

Diffusion Equations

1D Diffusion Equation

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

The diffusion equation exhibits the characteristic where it changes rapidly in the beginning, but after then evolution of u becomes slower. The diffusion equation eventually converges to a stationary solution u(x). Therefore, the numerical method required to solve this equation would require adaptive time steps as small ones are unnecessary in the stationary state.

2D Diffusion Equation

$$D_x \frac{\partial^2 c}{\partial x^2} + D_y \frac{\partial^2 c}{\partial y^2} = \frac{\partial c}{\partial t}$$

Implementation

The code inherited forward Euler, backward Euler, Trapezoidal Euler and Guass-Seidal algorithm from the previous program. Then, I created two solver files, one for the 1D diffusion equation and the other for the 2D diffusion equation. The main execution is handled in the main function, where it can specify which set of equations to solve. The program will calculate the error against the analytical solution. The main program then outputs all the resulting array to txt files, which can then be used by python for plotting. The program uses the eigen library.

Results

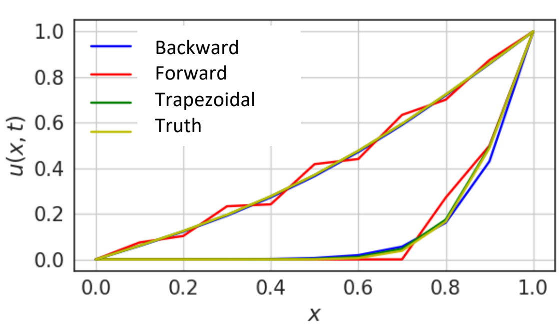

Figure 1. Time step = 0.52.

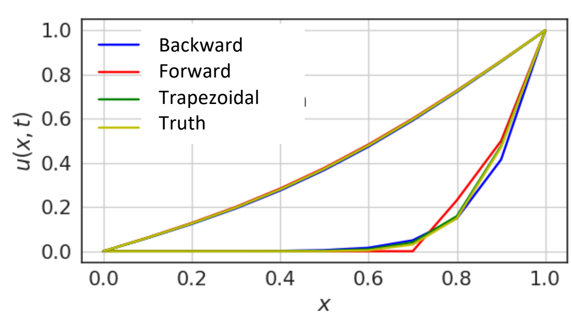

Figure 2. Time step = 0.48.

From the plotted results from the 1D equations, we observe that the forward Euler method is not always stable. When the time step is just slightly larger, the forward Euler method becomes unstable. On the other hand, both the backward Euler and the trapezoidal method is unconditionally stable.