Numerical Methods
Homework 2

Asad Ali

April 2015

1 Structure

The files included are as follows:

- driver.c: The main file, containing the test driver for the three functions. This outputs a file called "results" that displays 4 columns containing n, followed by the times for each of the 3 algorithms, respectively.
- heatsolvers.c/.h: The actual functions
- class-func.c/.h: A few general functions I've found helpful for homework assignments
- nrutil.c/.h: files for the rref routine

To compile, just use "make", which produces the executable "heat_test". "make clean" will remove the executable and any object files.

2 Equations

For FTCS, the equation I derived was:

$$C = \frac{\alpha \Delta x}{\Delta x^2}$$
$$T_{ijk}^{new} = (1-6C)T_{ijk} + C(T_{i+1,jk} + T_{i-1,jk} + T_{i,j-1,k} + T_{i,j+1,k} + T_{ij,k+1} + T_{ij,k-1}) + S \quad (1)$$

For Crank-Nicolson, the equation was:

$$C = \frac{\alpha \Delta x}{2\Delta x^2}$$
$$(1+6C)T_{ijk}^{new} - C(T_{i+1,jk}^{new} + ...) = (1-6C)T_{ijk} + C(T_{i+1,jk} + ...) + S \quad (2)$$

For ADI, the general equation was:

$$C = \frac{\alpha \Delta x}{3\Delta x^2}$$
$$(1+2C)T_{ijk}^{new} - C(T_{i+1,jk}^{new} + ...) = (1-4C)T_{ijk} + C(T_{i+1,jk} + ...) + \frac{S}{3} \quad (3)$$

With the "new" section on the left consisting of one dimension per calculation, and the section on the right consisting of the other two dimensions.

3 Algorithms

I created a struct called "simulation", which contains l_x, l_y, l_z, Δx, Δy, Δz, $S(x,y,z)$, and the boundary conditions for a simulation. Each function takes a simulation, timestep, and number of timesteps.

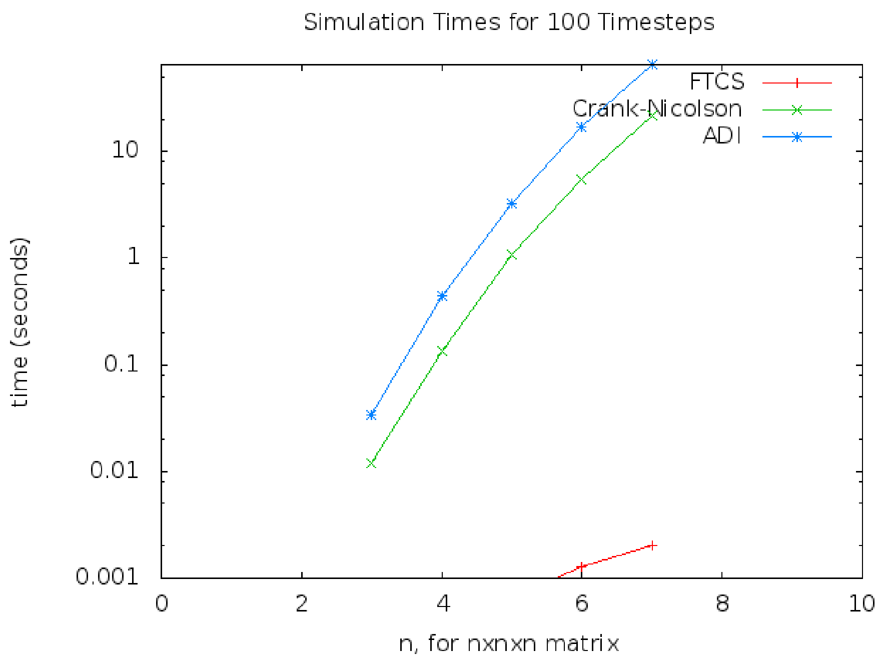

Figure 1: Time for each simulation run

The FTCS algorithm was a straightforward loop of a single equation through the whole grid for each timestep.

For Crank-Nicolson and ADI, I implemented static values for matrices A such that $Ax = b$. I then used rref to solve for the unknown values, since I don't want to have to explain to my children why some people live an "LU Decomposition lifestyle." As such, these algorithms run horribly slow, as God intended. Since my own rref routine had some bugs, I straight jacked y'all's and did some minor edits to the way it handles indexing.

4 Results

As expected, the FTCS algorithm ran orders of magnitude faster than the other two. It may have some stability conditions, but when I ran the other two I could actually hear myself aging. I imagine that writing some faster $Ax = b$ solvers will help, since crunching all those matrices is clearly what's slowing things down.