Math 493: Homework 2 – Anderson – Summer 2009

Due: Wednesday, June 3, 2009

1. Consider the one dimensional heat equation

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2}, \\ u &= 0 \qquad \text{on } x = 0, 1, \\ u &= f(x) \qquad \text{on } t = 0, \end{aligned}$$

We shall be interested in the solution to this problem for $0 \le x \le 1$ and $0 \le t \le 1$. Consider $f(x) = \sin 2\pi x$.

(a) Find the exact solution.

(b) Consider a numerical solution approach to this problem using the spatial discretization $x_j = (j-1)/N$ for $j = 1, \ldots, N+1$ where N is the number of space intervals on $x \in [0,1]$ and the time discretization $t_i = (i-1)/M$ for $i = 1, \ldots, M+1$ where M is the number of time intervals on $t \in [0,1]$. Here the grid spacings in the x and t directions are $h = 1/N$ and $k = 1/M$. Define $r = k/h^2$. Noting that $\phi_1^i = \phi_{N+1}^i = 0$ for all i define a solution vector $\vec{\phi}^i = (\phi_2^i, \phi_3^i, \ldots, \phi_N^i)$ which approximates the true solution at the spatial grid points at time step i.

Specifically, write a Matlab code (or perhaps three Matlab codes) to solve this problem numerically using:

1. Forward Euler Scheme

$$\vec{\phi}^{i+1} = A_{FE}\vec{\phi}^i, \tag{1}$$

where $A_{FE} = I + rT_{N-1}$.

2. Backward Euler Scheme

$$A_{BE}\vec{\phi}^{i+1} = \vec{\phi}^i, \tag{2}$$

where $A_{BE} = I - rT_{N-1}$.

3. Crank-Nicolson Scheme

$$A_{CN}\vec{\phi}^{i+1} = B_{CN}\vec{\phi}^i, \tag{3}$$

where $A_{CN} = 2I - rT_{N-1}$ and $B_{CN} = 2I + rT_{N-1}$.

In all of these cases I is an $N-1 \times N-1$ identity matrix and T_{N-1} is an $N-1 \times N-1$ tridiagonal matrices with -2 on the main diagonal and 1 on the first upper and first lower diagonals.

Your code(s) should implement each of the above three schemes. Turn these in as this will be part of your grade on this assignment. The goal is to explore the basic properties of these schemes. Your code need not be optimized for speed and efficiency although you will probably notice things slowing down as the grids in space and time are refined so any efficiency you can include will probably pay off. You can use Matlab's built in commands to solve linear systems. Please avoid computing matrix inverses explicitly (i.e. don't use 'inv(A)').

The following questions ask you to use your Matlab code to explore issues such as error and stability. To treat the error, define an exact solution vector $\vec{\Phi}^i$ at the final time $t=1$ that has components $\Phi(x_j, t_i)$ for $j = 2, \ldots, N$. Define a relative error as

$$e = \|\vec{\phi}^i - \vec{\Phi}^i\|_2 / \|\vec{\Phi}^i\|_2$$

Note that these norms can be computed in Matlab using the `norm` command.

(c) Take $N = 20$. Using $M = 25, 50, 100, 200, 400, 800, 1600, 3200$ examine a plot of the solution approximation at $t = 1$ – for all three methods – and discuss the results (Do not turn in every plot but you might turn in a couple representative ones). Could you have predicted any of these observations? As you are doing this, make a table showing M, r and errors e_{FE}, e_{BE} and e_{CN} for each scheme. Can you observe the predicted order of accuracy in time for each of these schemes in this data?

(d) Take $N = 100$ and repeat the above calculations with $M = 25, 50, 100, 200, 400, 800$. Can you observe the predicted order of accuracy in time for each of these schemes in this data? How might the value of N influence your observations?

(e) If $N = 1000$ what is the minimum value of M required in theory to avoid numerical instability for Forward Euler? Try running a few values of M here (e.g. $M = 25, 50, 100$) and see if you can observe the second order accuracy trend in time in the Crank-Nicolson scheme. Discuss the results.

2. Write out a finite difference formula for the expression

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

that is second order accurate in x and y discretizations. Be sure to draw your mesh and identify all notation that you use.