MATH 440/540 - Applied Mathematics I

Final Project (for grad. students)
Date assigned: April 15, 2009
Due date: **May 6, 2009**

- Include a cover page and *this* problem sheet
- Include the printout of your program(s) for completeness

PROBLEMS:

1. Consider the following initial value problem :

$$\frac{d^2u}{dx^2} + 2\frac{du}{dx} - 3u = 0 \tag{1}$$

$$u(0) = 2; \frac{du}{dx} = -2$$

   (a) Solve IVP (1) analytically using appropriate method.
   (b) Transform (1) into the system of two first order equations.
   (c) Solve the resulting system numerically using explicit Euler method for systems.
   (d) Plot the numerical and analytical solutions for $x \in [0, 2]$.
   (e) Compute the maximum norm of the difference between numerical and analytical solutions.

2. Consider the following heat equation:

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} \ = 0, \ x \in [0, 1], \ t > 0, \tag{2}$$

with initial and boundary conditions given by

$$u(x, 0) = \begin{cases} x, & \text{if } 0 \leq x \leq 0.5, \\ 1 - x, & \text{if } 0.5 < x \leq 1. \end{cases} \tag{3}$$

$$u(0, t) = u(1, t) = 0, \ t >= 0. \tag{4}$$

Numerically solve (2)-(4) with the diffusion coefficient $D = 1/2$.

Use the following difference scheme:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2}$$

Take $\Delta x = 0.1$ and compute the numerical solution of (2)-(4) with $\Delta t = 1/50, 1/100, 1/200$.

Questions.

- Numerically check the order of accuracy in time by showing the error between exact and numerical solution for different values of Δt.
- Compare plots of the exact and numerical solution for each scheme at $\Delta t = 1/50, 1/100, 1/200$.
- Explain the results of the computations.

Exact solution for 1D diffusion equation:

$$v(x,t) = \sum_{k=1}^{\infty} \frac{4}{(k\pi)^2} \sin(\frac{k\pi}{2}) \sin(k\pi x) e^{-D(k\pi)^2 t}, \tag{5}$$

where t is the time, x is the space coordinate and D is the diffusion coefficient.

Matlab code for computing exact solution is given below. In this code the summation has 13 terms (instead of ∞), the time variable is T (so the final time is 1 in this code), *xcoord* is the space coordinate, J is number of space intervals.

```
T=1;
xcoord=linspace(0,1,J+1)
vxt=zeros(J+1,1);
vxt_sum=zeros(1,J+1);
for k=1:13
   vxt=4*(1/((k*pi).^2))*sin(k*pi/2)*sin(k*pi*xcoord)*exp(-D*(k*pi).^2*T);
   vxt_sum=vxt_sum+vxt;
   sol=vxt_sum
end
```