

Worcester Polytechnic Institute
Mechanical Engineering Department
ME 612 Computational Fluid Dynamics
Project 1. Due February 8, 2010

Time evolution of a one-dimensional equation

Write a program to compute the unsteady behavior of the following equation

$$\frac{\partial f}{\partial t} + f \frac{\partial f}{\partial x} = \nu \frac{\partial^2 f}{\partial x^2}$$

in a periodic domain of length 1 with $\nu = 0.01$.

Take the initial conditions to be $f(x, t = 0) = \sin(2\pi x) + 1.0$ and approximate the equation using a forward in time approximation for the time derivative, and centered approximations for the first and second derivative. Follow the evolution up to time 1.0 using at least three different grid resolutions.

I strongly recommend that you use MATLAB, although other programming languages are also allowable. You should be able to use the program shown in class as a template. (The program is attached)

You should hand in a short description of what you did and what you see (1-2 pages), a plot of the solution at 3-4 times for one resolution and then a plot of the solution at time 1.0 for at least three resolutions. You will find that the solution is only stable if Δt is sufficiently small. Find, by numerical experiments, the stability limit for at least two different grid resolutions.

Remember to be professional and concise.

The program from class

```
% one-dimensional advection-diffusion by the FTCS scheme
n=21; nstep=100; length=2.0; h=length/(n-1); dt=0.05; D=0.05;
f=zeros(n,1); y=zeros(n,1); ex=zeros(n,1); time=0.0;
for i=1:n, f(i)=0.5*sin(2*pi*h*(i-1)); end; % initial conditions
for m=1:nstep, m, time
    for i=1:n, ex(i)=exp(-4*pi*pi*D*time)*...
        0.5*sin(2*pi*(h*(i-1)-time)); end; % exact solution
    hold off; plot(f,'linewidth',2); axis([1 n -2.0, 2.0]); % plot solution
    hold on; plot(ex,'r','linewidth',2); pause; % plot exact solution
    y=f; % store the solution
    for i=2:n-1,
        f(i)=y(i)-0.5*(dt/h)*(y(i+1)-y(i-1))+...
            D*(dt/h^2)*(y(i+1)-2*y(i)+y(i-1)); % advect by centered differences
    end;
    f(n)=y(n)-0.5*(dt/h)*(y(2)-y(n-1))+...
        D*(dt/h^2)*(y(2)-2*y(n)+y(n-1)); % do endpoints for
    f(1)=f(n); % periodic boundaries
    time=time+dt;
end;
```