CAAM 336 · DIFFERENTIAL EQUATIONS

Homework 2 · Solutions

Posted Wednesday 10 September 2014. Due 5pm Wednesday 17 September 2014.

1. [50 points: 8 points for (a), 12 points for (c), 10 points for (b), (d), (e)] The 1D heat equation with $\kappa = 1$ over the interval $[0, 1]$ is given by

$$\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x}2 = 0$$

with boundary conditions and initial condition

$$u(0,t) = u(1,t) = 0 \qquad t > 0,$$
$$u(x,0) = \sin(\pi x).$$

As we've seen in class, *centered* finite difference approximations are more accurate than both forwards/backwards difference approximations. To this end, we would like to find a way to leverage central differences for our approximation of the time derivative $\frac{\partial u}{\partial t}$.

The trick to doing so is to write down the finite difference equations in space and time at the point $(x_i, t_{j+1/2})$

$$\frac{\partial u}{\partial t}(x_i, t_{j+1/2}) = \frac{\partial u}{\partial x}2(x_i, t_{j+1/2}).$$

We can then proceed in two steps:

- Central differences *in time* then gives us

$$\frac{\partial u}{\partial t}(x_i, t_{j+1/2}) \approx \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{dt}$$

as an approximation for $\frac{\partial u(x_i, t_{j+1/2})}{\partial t}$, where $dt = t_{j+1} - t_j$ is time step.

- To approximate the term $\frac{\partial u}{\partial x}2(x_i, t_{j+1/2})$ we can average our finite difference approximations for $\frac{\partial u}{\partial x}2(x_i, t_j + 1)$ and $\frac{\partial u}{\partial x}2(x_i, t_j)$: defining $u(x_i, t_j) = u_i^n$, we can set

$$\frac{\partial u}{\partial x}2(x_i, t_{j+1/2}) \approx \frac{1}{2}\left[\frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{h^2} + \frac{u_{i+1}^{j} - 2u_i^{j} + u_{i-1}^{j}}{h^2}\right].$$

where $h = x_{i+1} - x_i$ is the grid spacing/mesh size in x.

Notice now that, if we combine the above two approximations, we no longer have any terms involving $t_{j+1/2}$! We have just defined the *Crank-Nicolson* scheme for u_i^j

$$\frac{u_i^{j+1} - u_i^j}{dt} = \frac{1}{2}\left[\frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{h^2} + \frac{u_{i+1}^{j} - 2u_i^{j} + u_{i-1}^{j}}{h^2}\right]$$

Turn to the next page for the rest of Problem 3.

(a) We know that $\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}$ is an $O(\Delta t^2)$ approximation to $\frac{\partial u(x,t+\Delta t/2)}{\partial t}$. Show that

$$\frac{1}{2}\left[\frac{u(x+\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x-\Delta x,t+\Delta t)}{\Delta x^2}+\frac{u(x+\Delta x,t)-2u(x,t)+u(x-\Delta x,t)}{\Delta x^2}\right]$$

is an $O(\Delta x^2)$ approximation to $\frac{\partial u(x,t+\frac{\Delta}{2})}{\partial x}2$. With this, we can conclude Crank-Nicolson is a second order accurate approximation to the PDE in both space and time, or that

$$\left|\frac{\partial u(x,t+\Delta t/2)}{\partial t}-\frac{\partial u(x,t+\Delta t/2)}{\partial x}2-\text{Crank-Nicolson formula}\right|=O(\Delta t^2+\Delta x^2).$$

(b) Write the Crank-Nicolson scheme as an update step

$$\mathbf{u}^{j+1}=(\mathbf{I}+\mathbf{A})^{-1}(\mathbf{I}-\mathbf{B})\mathbf{u}^j,$$

specifying exactly what the matrices $\mathbf{A}$ and $\mathbf{B}$ are.

(c) As with any timestepping method, we can rewrite the Crank-Nicolson scheme as

$$\mathbf{u}^{j+1}=((\mathbf{I}+\mathbf{A})^{-1}(\mathbf{I}-\mathbf{B}))^{j+1}\mathbf{u}^0.$$

Show that Crank-Nicolson scheme is unconditionally stable by showing that, for eigenvalues λ_i of $\mathbf{A}^{-1}\mathbf{B}$,

$$\lambda_i^j<\infty,\quad \text{for any } j>0.$$

(Hint: $\mathbf{I}+\mathbf{A}$ and $\mathbf{I}-\mathbf{B}$ should have the same eigenvectors.)

(d) Create a Matlab script that implements the Crank-Nicolson method. Compute the numerical solution at points x_i and times t_j and plot the computed solution values u_i^j for $i=0,...,N+1$ and $j=0,10,50$ where $N=8,16,32$.

(e) Given that $u(x,t)=e^{-\pi^2 t}\sin(\pi x)$ is the exact solution for the above problem, plot the error at each point $|u_{\text{exact}}(x_i,t_j)-u_i^j)|$, for $i=0,...,N+1$ and $j=0,10,50$ for $N=8,16,32$ for 3 successive time steps (use $dt=h$).

Solution.

(a) To estimate the truncation error (TE) for Crank-NicolsonŠs method, we first recall TaylorŠs theorem with remainder, which states that a function $u(x)$ can be expanded in a series about the point c:

$$u(x)=u(c)+u_x(c)(x-c)+\frac{u_{xx}(c)}{2!}(x-c)^2+\frac{u_{xxx}(c)}{3!}(x-c)^3++\frac{u^{(n+1)}(c)(\xi)}{n!}(x-c)^{n+1}$$

where ξ is between x and c. The last term is referred to as the remainder term or truncation error.

We write the equation at the point $(x_i,t_{j+\frac{1}{2}})$. To better understanding let see Crank-Nicolson Stencil.

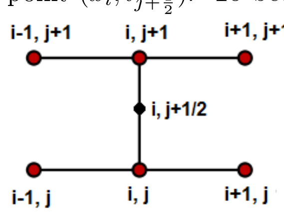


Figure 1: Crank-Nicolson Stencil

Taylor series expansion of $u(x,t+\Delta t)$ around the point $(x,t+\frac{\Delta t}{2})$ is

$$u(x,t+\Delta t) = u(x,t+\frac{\Delta t}{2}) + u_t(x,t+\frac{\Delta t}{2})\left(\frac{\Delta t}{2}\right) + \frac{u_{tt}(x,t+\frac{\Delta t}{2})}{2!}\left(\frac{\Delta t}{2}\right)^2 + \frac{u_{ttt}(x,t+\frac{\Delta t}{2})}{3!}\left(\frac{\Delta t}{2}\right)^3 + \frac{u_{tttt}(x,t+\frac{\Delta t}{2})}{4!}\left(\frac{\Delta t}{2}\right)^4 \dots$$

Similarly Taylor series expansion of $u(x,t)$ around the point $(x,t+\frac{\Delta t}{2})$ is

$$u(x,t) = u(x,t+\frac{\Delta t}{2}) - u_t(x,t+\frac{\Delta t}{2})\left(\frac{\Delta t}{2}\right) + \frac{u_{tt}(x,t+\frac{\Delta t}{2})}{2!}\left(\frac{\Delta t}{2}\right)^2 - \frac{u_{ttt}(x,t+\frac{\Delta t}{2})}{3!}\left(\frac{\Delta t}{2}\right)^3 + \frac{u_{tttt}(x,t+\frac{\Delta t}{2})}{4!}\left(\frac{\Delta t}{2}\right)^4 \dots$$

Taking difference of these two equations we get

$$u(x,t+\Delta t) - u(x,t) = 2u_t(x,t+\frac{\Delta t}{2})\left(\frac{\Delta t}{2}\right) + 2\frac{u_{ttt}(x,t+\frac{\Delta t}{2})}{3!}\left(\frac{\Delta t}{2}\right)^3 \dots$$

Dividing by Δt both sides gives

$$\boxed{\frac{u(x,t+\Delta t) - u(x,t)}{\Delta t} = u_t(x,t+\frac{\Delta t}{2}) + \frac{1}{4}\frac{u_{ttt}(x,t+\frac{\Delta t}{2})}{3!}(\Delta t)^2 \dots} \tag{1}$$

implying that the truncation error of the time derivative is $TE_t = \frac{1}{24}u_{ttt}(x,\eta)(\Delta t)^2$ such that $\eta \in (t,t+\Delta t)$. This implies the first order central finite difference formula for $\frac{\partial u}{\partial t}$ is 2nd order accurate i.e., $O(\Delta t^2)$ accurate.
To approximate the term $u_{xx}(x,t+\frac{\Delta t}{2})$ we use the average of the second centered differences for $u_{xx}(x,t+\Delta t)$ and $u_{xx}(x,t)$;
First of all let's find $u_{xx}(x,t+\Delta t)$. Then Taylor series expansion of $u(x+\Delta x,t+\Delta t)$ around the point $(x,t+\Delta t)$ is (Note that we are expanding in x direction at the $t+\Delta t$ th time level)

$$u(x+\Delta x,t+\Delta t) = u(x,t+\Delta t) + u_x(x,t+\Delta t)(\Delta x) + \frac{u_{xx}(x,t+\Delta t)}{2!}(\Delta x)^2 + \frac{u_{xxx}(x,t+\Delta t)}{3!}(\Delta x)^3 + \frac{u_{xxxx}(x,t+\Delta t)}{4!}(\Delta x)^4 + \frac{u_{xxxxx}(x,t+\Delta x)}{5!}(\Delta x)^5 \dots$$

Similarly Taylor series expansion of $u(x-\Delta x,t+\Delta t)$ around the point $(x,t+\Delta t)$ is

$$u(x-\Delta x,t+\Delta t) = u(x,t+\Delta t) - u_x(x,t+\Delta t)(\Delta x) + \frac{u_{xx}(x,t+\Delta t)}{2!}(\Delta x)^2 - \frac{u_{xxx}(x,t+\Delta t)}{3!}(\Delta x)^3 + \frac{u_{xxxx}(x,t+\Delta t)}{4!}(\Delta x)^4 - \frac{u_{xxxxx}(x,t+\Delta x)}{5!}(\Delta x)^5 \dots$$

Adding last two equation gives

$$u(x+\Delta x,t+\Delta t) + u(x-\Delta x,t+\Delta t) = 2u(x,t+\Delta t) + u_{xx}(x,t+\Delta t)(\Delta x)^2 + 2\frac{u_{xxxx}(x,t+\Delta t)}{4!}(\Delta x)^4 \dots$$

Subtracting $2u(x,t+\Delta t)$ from both sides and dividing by Δx^2 gives

$$\underbrace{\frac{u(x+\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x-\Delta x,t+\Delta t)}{\Delta x^2}}_{=U^{j+1}} = u_{xx}(x,t+\Delta t)+\frac{u_{xxxx}(x,t+\Delta t)}{12}(\Delta x)^2\ldots$$

Now we will find Taylor series expansion of $u(x+\Delta x,t)$ and $u(x-\Delta x,t)$ around the point (x,t) (Note that this time we are getting Taylor series expansion in x direction at the t th time level). By repeating same procedure as $t+\Delta t$ th time level we get

$$\underbrace{\frac{u(x+\Delta x,t)-2u(x,t+\Delta t)+u(x-\Delta x,t)}{\Delta x^2}}_{=U^{j}} = u_{xx}(x,t)+\frac{u_{xxxx}(x,t)}{12}(\Delta x)^2\ldots$$

Now taking average of the second centered differences for $u_{xx}(x,t+\Delta t)$ and $u_{xx}(x,t)$ we will find approximation for $u_{xx}(x,t+\frac{\Delta}{2})$

$$\boxed{\frac{1}{2}(U^{j+1}+U^j) = u_{xx}(x,t+\frac{\Delta t}{2})+\frac{u_{xxxx}(x,t+\frac{\Delta t}{2})}{12}(\Delta x)^2\ldots} \tag{2}$$

implying that the truncation error of the 2nd order space derivative is $TE_{xx}=\frac{1}{12}u_{xxxx}(\xi,\eta)(\Delta x)^2$ such that $\eta\in(t,t+\Delta t)$ and $\xi\in(x-\Delta x,x+\Delta x)$. This implies the 2nd order central finite difference formula for $\frac{\partial^2 u}{\partial x^2}$ is 2nd order accurate i.e., $O(\Delta x^2)$ accurate.
Both (1) and (2) implies(subtracting (2) from (1))

$$\underbrace{u_t(x,t+\frac{\Delta t}{2})-u_{xx}(x,t+\frac{\Delta t}{2})}_{PDE}+TE_t-TE_{xx} = \underbrace{\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}-\frac{1}{2}(U^{j+1}+U^j)}_{CrankNicolson}$$

From here we can conclude

$$|PDE-CrankNicolson| = O(\Delta x^2+\Delta t^2)$$

(b) From the problem Crank-Nicolson scheme given by following formula

$$\frac{u_i^{j+1}-u_i^j}{dt} = \frac{1}{2}\left[\frac{u_{i+1}^{j+1}-2u_i^{j+1}+u_{i-1}^{j+1}}{h^2}+\frac{u_{i+1}^{j}-2u_i^{j}+u_{i-1}^{j}}{h^2}\right]$$

Define $r=\frac{dt}{2dx^2}$ Then it turns out

$$u_i^{j+1}-u_i^j = r\left[u_{i+1}^{j+1}-2u_i^{j+1}+u_{i-1}^{j+1}+u_{i+1}^{j}-2u_i^{j}+u_{i-1}^{j}\right]$$

Rearranging the term gives

$$(1+2r)u_i^{j+1}-r(u_{i+1}^{j+1}+u_{i-1}^{j+1}) = (1-2r)u_i^j+r(u_{i+1}^j+u_{i-1}^j)$$

This leads to following matrix equation (since we had homogeneous Dirichlet boundary conditions we do not have any contribution to our system from Dirichlet boundary)

$$\underbrace{\begin{bmatrix} 1+2r & -r & & & \\ -r & 1+2r & -r & & \\ & -r & 1+2r & \ddots & \\ & & \ddots & \ddots & -r \\ & & & -r & 1+2r \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} u_1^{j+1} \\ u_2^{j+1} \\ \vdots \\ u_{N-1}^{j+1} \\ u_N^{j+1} \end{bmatrix}}_{U^{j+1}} = \underbrace{\begin{bmatrix} 1-2r & r & & & \\ r & 1-2r & r & & \\ & r & 1-2r & \ddots & \\ & & \ddots & \ddots & r \\ & & & r & 1-2r \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} u_1^{j} \\ u_2^{j} \\ \vdots \\ u_{N-1}^{j} \\ u_N^{j} \end{bmatrix}}_{U^{j}}$$

Then we have system of equations $\mathbf{LU^{j+1}} = \mathbf{MU^j}$ or $U^{j+1} = \mathbf{L^{-1}MU^j}$. Now we want to write $\mathbf{L}$ as sum of identity matrix I.

$$\mathbf{L} = \mathbf{I} + \mathbf{A}$$

where $\mathbf{A}$ is

$$r\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix}$$

Similarly we want to write $\mathbf{M}$ as sum of identity matrix I.

$$\mathbf{M} = \mathbf{I} - \mathbf{B}$$

where $\mathbf{B}$ is

$$r\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix}$$

We can conclude $\mathbf{A} = \mathbf{B}$.

(c) For the motivation first remember for any timestepping method,

$$\mathbf{u}^{j+1} = ((\mathbf{I} + \mathbf{A})^{-1}(\mathbf{I} - \mathbf{B}))^{j+1}\mathbf{u}^0$$

where u^0 is initial value and it contain some error with our assumption. Define error $e^0 = |u^0 - u^0_*|$ where u_* is exact solution of the problem. Then

$$e^j = |u^j - u^j_*| = |((\mathbf{I} + \mathbf{A})^{-1}(\mathbf{I} - \mathbf{B}))^j\mathbf{e}^0|.$$

Now taking norm of both sides

$$\begin{aligned} \|e^j\| &= \|((\mathbf{I} + \mathbf{A})^{-1}(\mathbf{I} - \mathbf{B}))^j\mathbf{e}^0\| \\ \text{(by matrix and vector norm property)} &\leq \|((\mathbf{I} + \mathbf{A})^{-1})^j\|\|(\mathbf{I} - \mathbf{B})^j\|\|\mathbf{e}^0\| \\ \text{(we know that}\|A^j\| \leq \|A\|^j) &\leq \|(\mathbf{I} + \mathbf{A})^{-1}\|^j\|\mathbf{I} - \mathbf{B})\|^j\|\mathbf{e}^0\| \end{aligned}$$

We might define matrix norm as follows

$$\|\mathbf{A}\| = \sqrt{\rho(\mathbf{AA^T})} = \sqrt{\mathbf{max}|\lambda_\mathbf{i}|^\mathbf{2}} = \mathbf{max}|\lambda_\mathbf{i}|$$

where $\rho(\mathbf{A})$ is called spectral radius of $\mathbf{A}$ which means maximum eigenvalues λ of the matrix $\mathbf{A}$. Then we can say

$$\|e^j\| \leq \left(|\lambda_{(\mathbf{I}+\mathbf{A})^{-1}}||\lambda_{(\mathbf{I}-\mathbf{B})}|\right)^j \|\mathbf{e}^0\|$$

If max eigenvalue of $(\mathbf{I} + \mathbf{A}^{-1})(\mathbf{I} - \mathbf{B})$ in modulus is less than one, then $e^j \to 0$ for $j \to \infty$ Now, by formula eigenvalues of the $N \times N$ matrices $\mathbf{A}$ and $\mathbf{B}$ is $r(2 + 2\cos\frac{i\pi}{N+1}) = 4r\cos^2\frac{i\pi}{2(N+1)}$

Then

$$|\lambda_{(\mathbf{I}+\mathbf{A})^{-1}}| = |(1+4r\cos^2\frac{i\pi}{2(N+1)})^{-1}| = \frac{1}{|1+4r\cos^2\frac{i\pi}{2(N+1)}|}$$

and

$$|\lambda_{(\mathbf{I}-\mathbf{B})}| = |(1-4r\cos^2\frac{i\pi}{2(N+1)})|$$

Therefore

$$|\lambda_{(\mathbf{I}+\mathbf{A})^{-1}}||\lambda_{(\mathbf{I}-\mathbf{B})}| = \frac{|(1-4r\cos^2\frac{i\pi}{2(N+1)})|}{|1+4r\cos^2\frac{i\pi}{2(N+1)}|} \leq 1$$

It is easy to see that above ratio is always less than one without any restriction on $r > 0$. Then we say that Crank-Nicolson is unconditionally stable.

Note: Students does not have to explain motivation part

(d) From part b we know that by solving following linear system we get solution successive time steps.

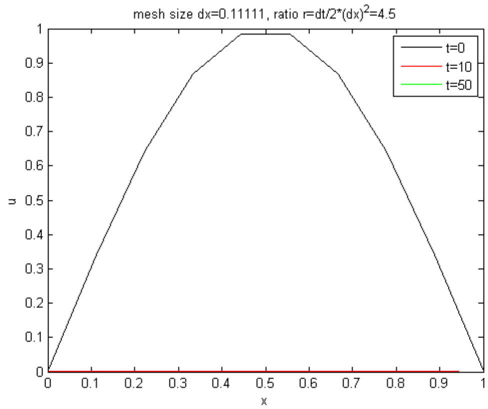
$$U^{j+1} = (\mathbf{I}+\mathbf{A})^{-1}(\mathbf{I}-\mathbf{B})U^j$$

From the graph note that when we use bigger time step heat equation converge to steady state case.

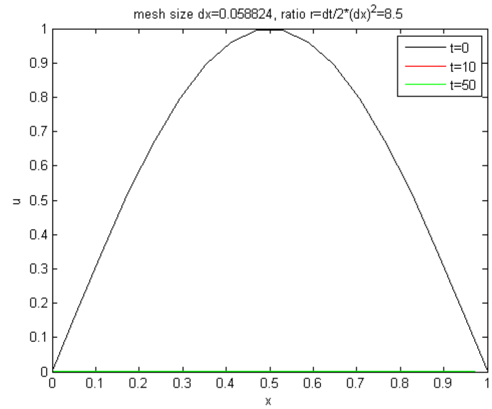
Included is Matlab code that can be used to generate the finite difference solution and the error between it and the exact solution.

```
%% Heat equation u_t=u_xx - finite difference scheme - Crank Nicolson method

%%
%   This program integrates the heat equation  u_t - u_xx = 0
%   on the interval [0,1] using finite difference approximation
%   via Crank Nicolson method. The implicit set of equations are solved at
%   each time step



clear all, clc, clf
%% Initial and Boundary conditions
M=32;
dx = (1-0)/(M+1);
dt = dx;

% number of time iterations
K =100;

% final time of the computation
Tf = K*dt;

% initial conditions
u0 = @(x) sin(pi*x);



% the mesh ratio
r = dt/(2*dx^2);

tvals=0:dt:Tf;
xvals=0:dx:1;

ue= sin(pi*xvals);

N=length(tvals);
J=length(xvals);
% Note: the original index j runs from j = 1 ( x = 0) to j = J ( x = 1).
```

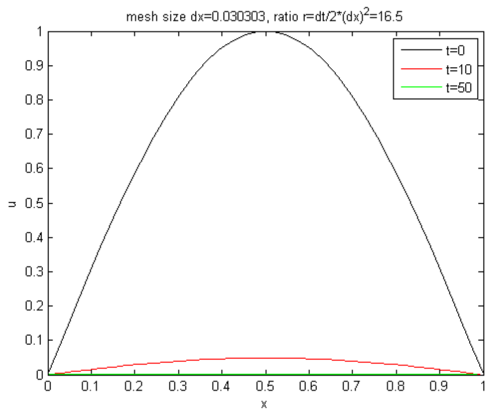
```
u=zeros(J,N);

u(:,1)=u0(xvals);


E = ones(J ,1);
D = spdiags([-E 2*E -E],[-1,0,1],J,J);
I = speye(J);

A = I+ r*D;
B = I- r*D;

A(1,:) = 0; A(1,1) = 1;
A(J,:) = 0; A(J,J) = 1;



%% Time iteration

n=0;
for m = 1:K-1
    n=n+1; % counter for iteration
    rhs=B*u(:,n);
    rhs([1,J])=0;
    u(:,n+1) = A\rhs;
    u(1,n+1) = 0;
    u(J,n+1) = 0;
end




%% Plot the final results

figure(1)
plot(xvals,u(:,1),'k');hold on %soluton at t=0
xlabel x; ylabel u;
title(strcat('mesh size dx= ',num2str(dx),...
    ', ratio r=dt/2*(dx)^2= ',num2str(r)))
plot(xvals,u(:,11),'r');hold on
plot(xvals,u(:,51),'g');hold on
legend('t=0', 't=10','t=50')
hold off



figure(2)
surf(xvals, tvals, u')
xlabel x; ylabel t; zlabel u
title(strcat('mesh size dx= ',num2str(dx),...
    ', ratio r=dt/2*(dx)^2= ',num2str(r)))

figure(3)
plot(xvals,abs(exp(-pi^2*tvals(1))*ue-u(:,1)'),'b');hold on %soluton at t=0
xlabel  x; ylabel |error|;
title(strcat('mesh size dx= ',num2str(dx),...
    ', ratio r=dt/2*(dx)^2= ',num2str(r)))
plot(xvals,abs(exp(-pi^2*tvals(11))*ue-u(:,11)'),'r');hold on
plot(xvals,abs(exp(-pi^2*tvals(51))*ue-u(:,51)'),'g');hold on
legend('t=0', 't=10','t=50')
hold off

figure(4)
semilogy(xvals,abs(exp(-pi^2*tvals(1))*ue-u(:,1)'),'b');hold on %soluton at t=0
xlabel x; ylabel error;
```



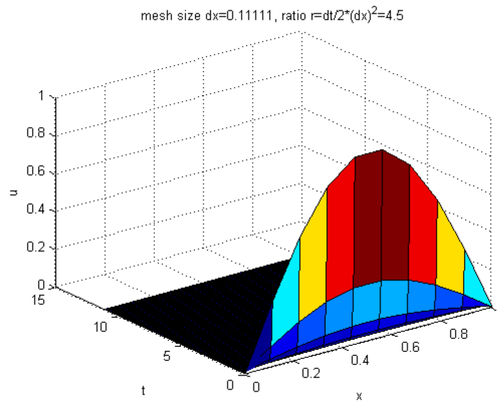
(a) FD solutions for various time level when $N = 8$



(b) FD solutions for various time level when $N = 16$



(c) FD solutions for various time level when $N = 32$



(d) FD surface plot when $N = 8$

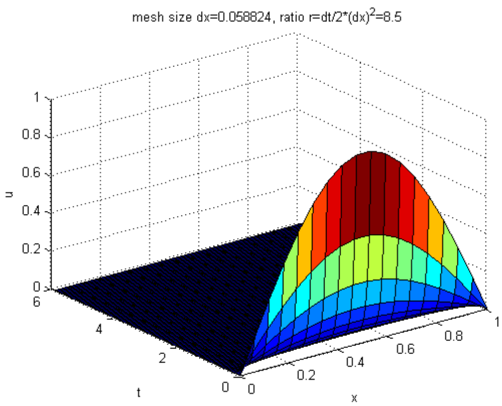
```
title(strcat('mesh size dx= ',num2str(dx),...
    ', ratio r=dt/2*(dx)^2= ',num2str(r)))
semilogy(xvals,abs(exp(-pi^2*tvals(11))*ue-u(:,11)'),'r');hold on
semilogy(xvals,abs(exp(-pi^2*tvals(51))*ue-u(:,51)'),'g');hold on
legend('t=0', 't=10','t=50')
hold off
```

Note: Students does not have to give both FD 2D-plots and surfaces, one of them should be enough. Also the students who fix dt and find solution is also excepted as a right solution. The plots with fix dt is given end of that question.

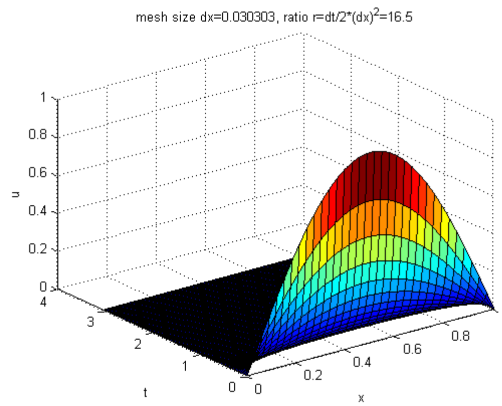
(e) In that part note that $dt = h = \frac{1}{N+1}$ whenever we change N, dt is also changing. For example when $N = 32$ error is scaling with 10^{-4}, $N = 16$ error is scaling with 10^{-5} and $N = 8$ error is scaling with 10^{-6} after 50 time step. What's happening here, when $N = 32$, $dt = 1/33$, after 50 time step we get total time $t = dt * 100 = 1,5151$ similarly when $N = 16$, $dt = 1/17$ and after 50 time step we get total time $t = dt * 100 = 2,94$. This means that, actually we are calculating error later time step that's why, as we get bigger mesh size, error is getting better because we are finding error later time. It might be better comparison if we would fix dt and looking at error when we proceed in time.

We added also the plot when $dt = 0.001$ for the given method at that part. Here we can see we have better approximation when we increase N.

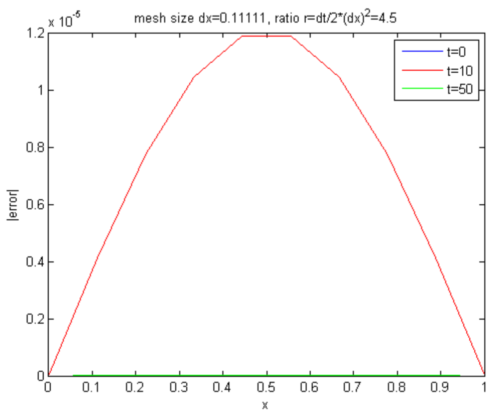
Note: It is not expect from students to give above explanation and for the graph they do not have to plot loglog graph. Error plot would be enough. Also the students who fix dt and find error is



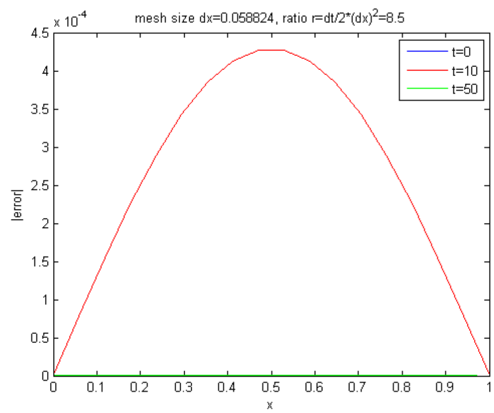
(e) FD surface plot when $N = 16$



(f) FD surface plot when $N = 32$

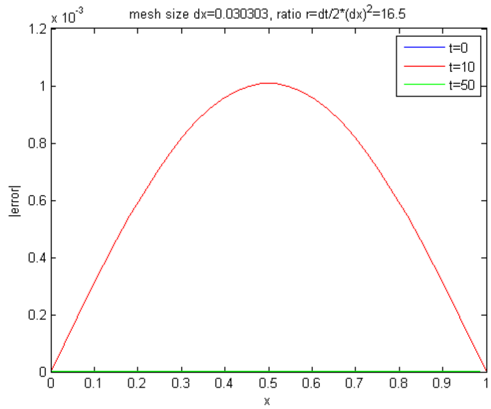


(g) FD error for various time level when $N = 8$

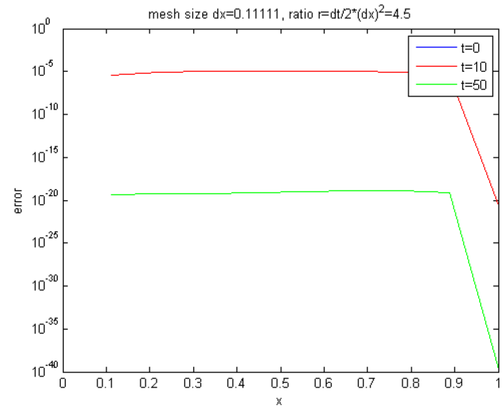


(h) FD error for various time level when $N = 16$

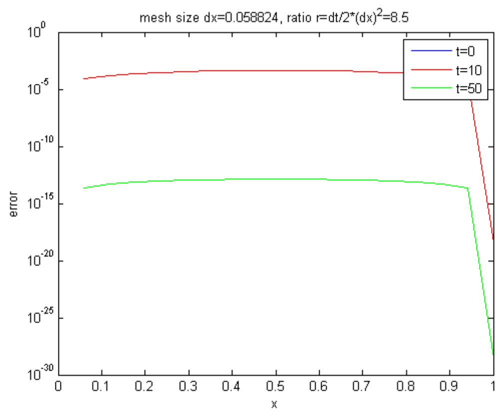
also excepted as a right solution.



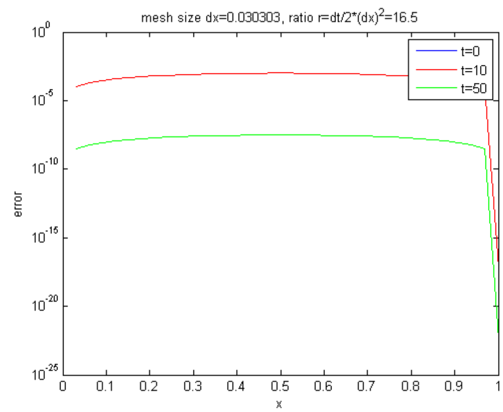
(i) FD error for various time level when $N = 32$



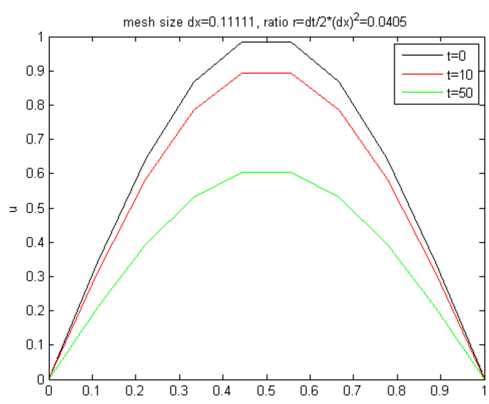
(j) FD logerror for various time level when $N = 8$



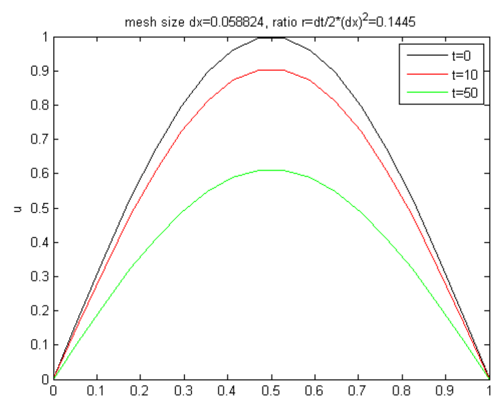
(k) FD logerror for various time level when $N = 16$



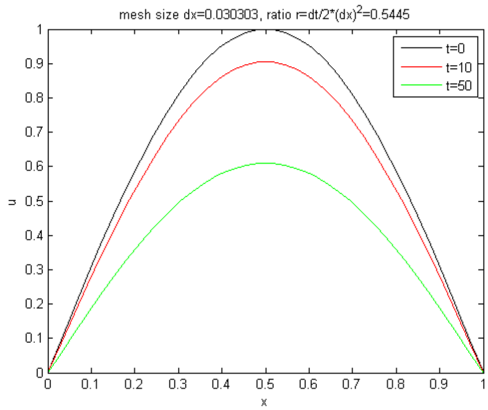
(l) FD logerror for various time level when $N = 32$



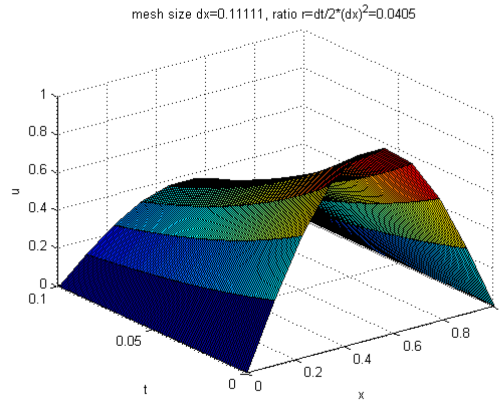
(m) FD solutions for various time level when $N = 8$ and $dt = 0.001$



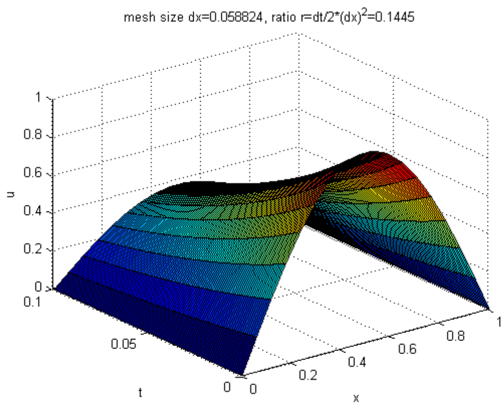
(n) FD solutions for various time level when $N = 16$ and $dt = 0.001$



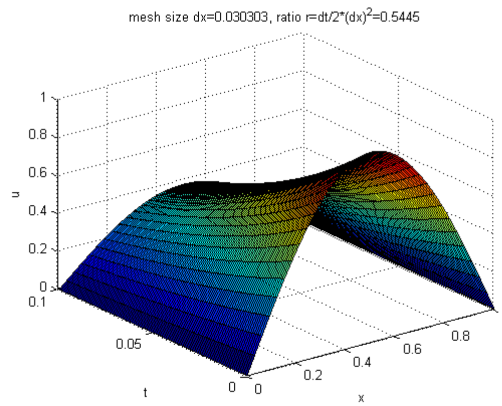
(o) FD solutions for various time level when $N = 32$ and $dt = 0.001$



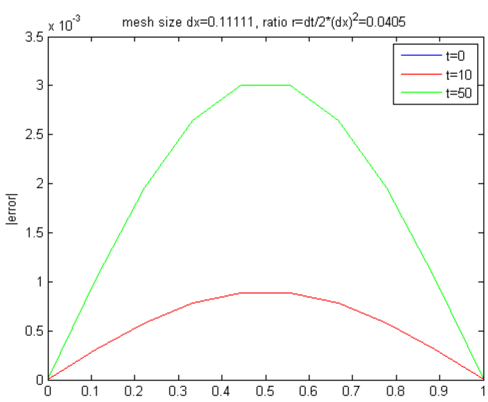
(p) FD surface plot when $N = 8$ and $dt = 0.001$



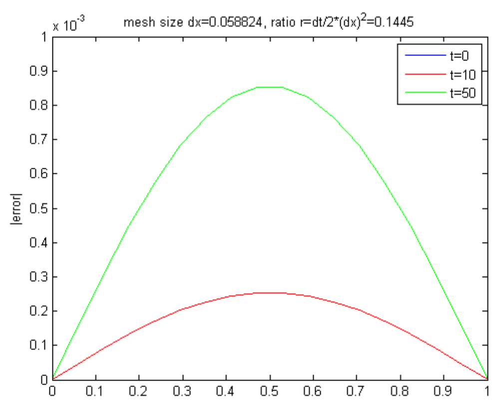
(q) FD surface plot when $N = 16$ and $dt = 0.001$



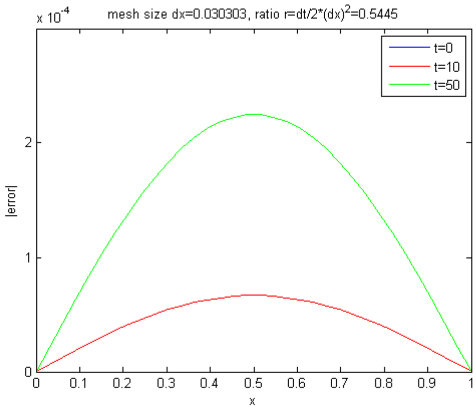
(r) FD surface plot when $N = 32$ and $dt = 0.001$



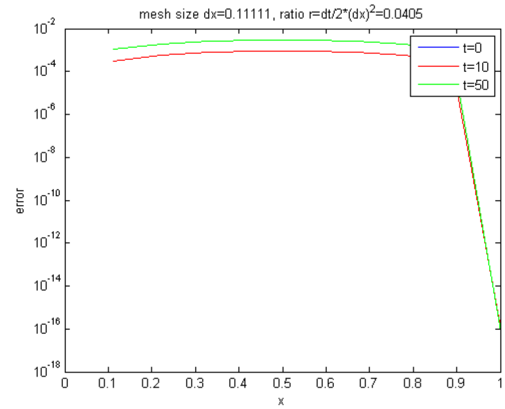
(s) FD error for various time level when $N = 8$ and $dt = 0.001$



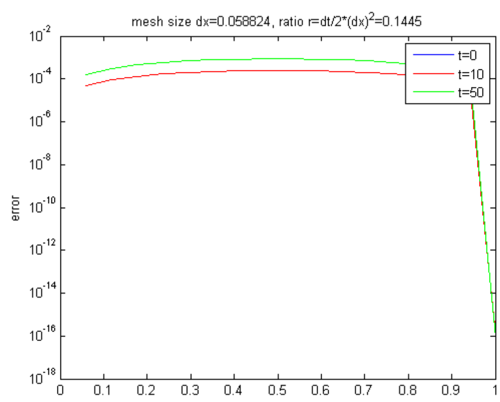
(t) FD error for various time level when $N = 16$ and $dt = 0.001$



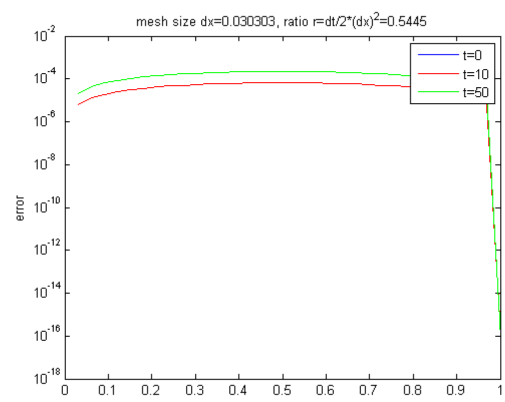
(u) FD error for various time level when $N = 32$ and $dt = 0.001$



(v) FD logerror for various time level when $N = 8$ and $dt = 0.001$



(w) FD logerror for various time level when $N = 16$ and $dt = 0.001$



(x) FD logerror for various time level when $N = 32$ and $dt = 0.001$