ME 282, 2023
R. K. Shukla
Assignment 3, Due Tuesday, **March** 23

1. Consider the leap frog scheme for numerical integration of the ODE

$$\frac{dy}{dt} = f(y,t),\ y(0) = y_0. \tag{1}$$

Analyze the stability of this scheme and express it in the form of a stability diagram.

Now consider application of leap frog method with central in space discretization of

$$u_t = bu_{xx}, u(0,t) = u(1,t) = 0,\ u(x,0) = u_0(x) \tag{2}$$

constructed using uniform grid points in [0,1] and the standard three-point, second order, discretization to approximate u_{xx} (let $x_i = i/N, i = 0, 1, \ldots, N$). Derive a stability criterion for this method using the stability diagram obtained from the first part.

2. Determine the order of accuracy and discuss the stability of the following difference scheme

$$\left(1 + \frac{1}{12}\delta_x^{(2)}\right)\left(u_j^{n+1} - u_j^n\right) = \frac{b\Delta t}{2h^2}\delta_x^{(2)}\left(u_j^{n+1} + u_j^n\right) + \frac{\Delta t}{2}\left[f_j^{n+1} + \left(1 + \frac{1}{6}\delta_x^{(2)}\right) f_j^n\right] \tag{3}$$

for the solution of $u_t = bu_{xx} + f$, where $\delta_x^{(2)} u_j^n = u_{j+1}^n - 2u_j^n + u_{j-1}^n$.

3. Consider the DuFort-Frankel scheme for the diffusion equation $u_t = bu_{xx}$, where $b > 0$, on a uniform mesh,

$$u_j^{n+1} = u_j^{n-1} + \frac{2b\Delta t}{h^2}\left(u_{j+1}^n - u_j^{n+1} - u_j^{n-1} + u_{j-1}^n\right). \tag{4}$$

Determine the restrictions on time step Δt and grid spacing h required to ensure convergence.

4. Write a computer code to solve

$$\begin{aligned} &u_t = \nabla^2 u + \sin(2\pi x)\sin(2\pi y)\sin(2\pi t) \quad \text{for} \quad (x,y) \in (0,1)^2, \\ &u(x,0,t) = u(x,1,t) = 0, \\ &u(0,y,t) = u(1,y,t) = 0, \end{aligned} \tag{5}$$

using the Crank-Nicolson method. Use the matrix diagonalization method discussed in the class to solve the resulting linear system of equations. Plot the L_2 and L_∞ errors as a function of number of mesh nodes N ($h_x = h_y = 1/N$) and the time step size Δt. What rate of convergence do you observe in space and in time?

(1)

ASSIGNMENT-3 -ME 282

-SAURABH SHARMA
21498

Solution 1 $\frac{dy}{dt} = f(y,t)$, $y(0) = y_0$

$\frac{dy}{dt} = \lambda u$ (Modal form) where λ represents eigenvalue of system. This equation is known as Dahlquist equation & is used to test stability.

Solution of the system : $y(t) = e^{\lambda_R t} e^{i\lambda_I t} y_0$ where, $\lambda_R = Re(\lambda)$, $\lambda_I = Im(\lambda)$

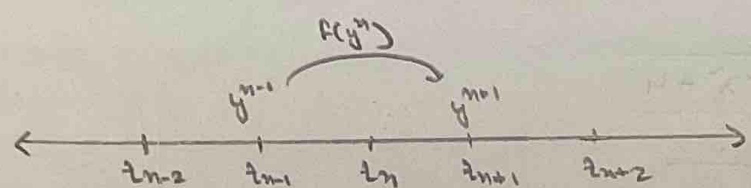
$\therefore \lambda = \lambda_R + i\lambda_I$

For a well posed system, $y(t)$ should be bounded & then :

$\lambda_R \leq 0$

Leap Frog Numerical Integration :

With respect to time, it will be centre in time discretization scheme. Leap denoting the jump over a node for evaluation of difference for time derivatives.



$$\frac{y(t_n+\Delta t) - y(t_n - \Delta t)}{2\Delta t} = f(y(t_n))$$

In modal form : $\frac{y^{n+1} - y^{n-1}}{2\Delta t} = \lambda y^n$

$$y^{n+1} - 2\lambda\Delta t y^n - y^{n-1} = 0 \qquad ..(1)$$

Lets define stability function R such that

$$R = \frac{y^{n+1}}{y^n} = \frac{y^n}{y^{n-1}}$$

Substituting :, $Ry^n - 2\lambda\Delta t y^n - \frac{y^n}{R} = 0$

$\Rightarrow (R^2 - 2\lambda\Delta t R - 1)\frac{y^n}{R} = 0$

Assuming $\frac{y^n}{R} \neq 0$: $R^2 - 2\lambda\Delta t R - 1 = 0$

Roots for quadratic equation which is also known as stability polynomial is given by

$$R(\lambda\Delta t) = \lambda\Delta t \pm \sqrt{(\lambda\Delta t)^2 + 1} \qquad ..(2)$$

This stability function can be used to determine stability criteria as

$$y^{n+1} = Ry^n = R^2 y^{n-1} = R^{n+1} y_0$$

Hence for y^{n+1} to be bounded : $|R(\lambda\Delta t)| \leq 1$ ...(3)

From equation (2) & (3): (2)

$$|\lambda\Delta t \pm \sqrt{(\lambda\Delta t)^2+1}| \leq 1$$

where when bounds of $\lambda\Delta t$ as per above inequality is presented on Argand diagram, it represents stability diagram.

Denoting $\lambda\Delta t = z$

$$R(z) = z \pm \sqrt{z^2+1}$$

$$|z \pm \sqrt{z^2+1}| \leq 1 \qquad \dots(4)$$

The corresponding stability diagram can be determined as follows:

Necessary & sufficient condition of stability is that

a.) All roots of stability polynomial satisfy equation $|R(z)| \leq 1$

b.) if $|R(z)| = 1$, it must be a simple root (not repeated)

CASE1 Roots purely real

Let $z = z_r$, $\operatorname{Im}(z) = 0$

Then using equation (4) for the 1st root:

$$|z_r + \sqrt{z_r^2+1}| \leq 1$$

$$\therefore z_r \leq 0$$

For 2nd root: $|z_r - \sqrt{z_r^2+1}| \leq 1$

$$\therefore z_r \geq 0$$

Both of the above equations should be satisfied simultaneously, but that's possible only when $z_r = 0$ which gives $|R(z)| = 1$, i.e., repeated roots of modulus 1, which violates the second criteria. Therefore, roots can't be purely real or have a real component.

CASE 2 Roots purely imaginary

Let $z = iz_i$, $\operatorname{Re}(z) = 0$

Thus, $R(z\pm) = iz_i \pm \sqrt{(iz_i)^2+1}$

$$\therefore R(z\pm) = iz_i \pm \sqrt{1-z_i^2}$$

If $z_i^2 < 1$, then square root is real, and for both the roots

$$R(z_\pm) = \sqrt{z_i^2+(1-z_i^2)} = 1$$

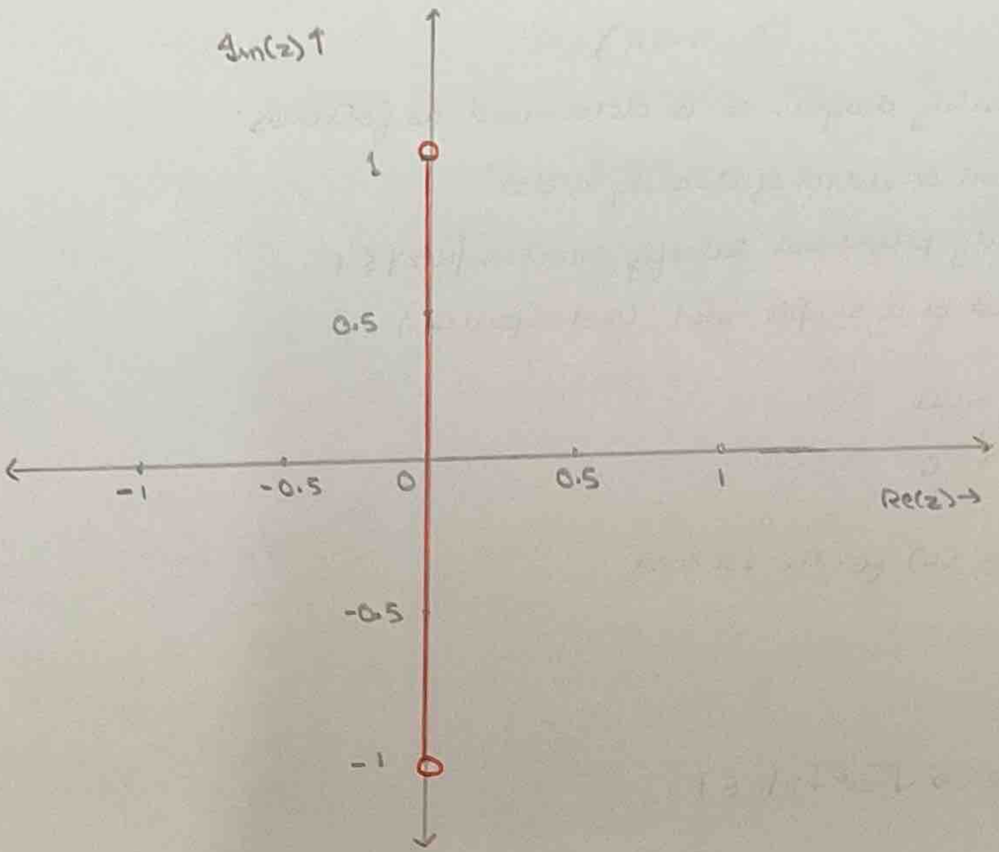
Here, $R(z_+) \neq R(z_-)$ will be true, even though $|R(z_\pm)| = 1$, only for strictly inequality

$$z_i^2 < 1 \Rightarrow |z_i| < 1$$

If $z_i = \pm 1$, then $R(z_+) = R(z_-)$ are repeated roots of modulus 1 which violates second criteria & if $|z_i| > 1$ both roots can't simultaneously satisfy $|R(z)| \leq 1$, violating first criteria

Therefore for stability region, $z \in (i, i)$

Hence stability diagram for leap frog iteration scheme is as below:



Stability diagram for leap frog scheme in z-plane

$u_t = b u_{xx}$, $u(0,t) = u(1,t) = 0$, $u(x,0) = u_0(x)$ (4)

Centred in time and standard 3 point , 2nd order discretization to approximate u_{xx}.

$$\frac{u_{i,j+1} - u_{i,j-1}}{2k} = \frac{b}{h^2}\left[u_{i-1,j} + u_{i+1,j} - 2u_{i,j}\right]$$

Let $\lambda = \frac{2kb}{h^2}$

$$u_{i,j+1} - u_{i,j-1} = \lambda\left(u_{i-1,j} + u_{i+1,j} - 2u_{i,j}\right)$$

$$u_{i,j+1} = u_{i,j-1} + \lambda\left(u_{i-1,j} + u_{i+1,j} - 2u_{i,j}\right)$$

Substitute : $u_{i,j} = g^j e^{imh\beta}$ where, $m = \sqrt{-1}$

$$g^{j+1} e^{mih\beta} = g^{j-1} e^{mih\beta} + \lambda\left(g^j e^{m(i-1)h\beta} + g^j e^{m(i+1)h\beta} - 2g^j e^{mih\beta}\right)$$

$$g = g^{-1} + \lambda\left(e^{-mh\beta} + e^{mh\beta} - 2\right)$$

$$g^2 - \lambda g\left(e^{-mh\beta} + e^{mh\beta} - 2\right) - 1 = 0$$

$$g^2 - \lambda g\left(2\cos(h\beta) - 2\right) - 1 = 0$$

$$g_\pm = \frac{2(\cos(h\beta)-1)\lambda \pm \sqrt{4(\cos(h\beta)-1)^2\lambda^2 + 4}}{2}$$

$$g_\pm = \lambda\left[\cos(h\beta)-1\right] \pm \sqrt{\lambda^2\left[\cos(h\beta)-1\right]^2 + 1}$$

Unconditionally unstable.

Alternate : $u_t = \frac{b}{h^2}\left[u_{j+1}^n - 2u_j^n + u_{j-1}^n\right]$

which can be written as system of equations as :

$$\frac{d\underset{\sim}{u}}{dt} = -\frac{b}{h^2}\underset{\sim}{A}\,\underset{\sim}{u} = \underset{\sim}{F}$$

$$\underset{\sim}{A} = \begin{bmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & \cdots & \cdots & 0 \\ 0 & \cdots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{bmatrix}$$

$\underset{\sim}{A}$ can be diagonalized & written as $\underset{\sim}{A} = \underset{\sim}{P}\,\underset{\sim}{\Lambda}\,\underset{\sim}{P}^{-1}$

where, $\underset{\sim}{\Lambda}$ & $\underset{\sim}{P}$ are obtained using Eigen values & vectors which we know are given by

$$\underset{\sim}{\Lambda} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_{N-1} \end{bmatrix}, \quad \lambda_k = 4\sin^2\left(\frac{\pi k}{2N}\right)$$

$$\underset{\sim}{P} = \left[\underset{\sim}{x}^{(1)}\ \underset{\sim}{x}^{(2)} \cdots \underset{\sim}{x}^{(N-2)}\ \underset{\sim}{x}^{(N-1)}\right], \quad \underset{\sim}{P}^{-1} = \underset{\sim}{P}^{T}$$

where, $\underset{\sim}{x}^{(k)}$ are eigenvectors given by

$x_j^{(k)} = \sin(k\pi x_j)$, $k = 1, 2, \ldots N-1$

Then, $\dfrac{d\underset{\sim}{u}}{dt} = \dfrac{-b}{h^2} \underset{\sim}{P} \underset{\sim}{\Lambda} \underset{\sim}{P}^{-1} \underset{\sim}{u}$

Pre-multiplying with P^{-1} : $\dfrac{d}{dt}(\underset{\sim}{P}^{-1}\underset{\sim}{u}) = -\dfrac{b}{h^2} \Lambda \underset{\sim}{P}^{-1} \underset{\sim}{u}$

Let $\underset{\sim}{P}^{-1}\underset{\sim}{u} = \underset{\sim}{v}$

$\dfrac{d\underset{\sim}{v}}{dt} = \dfrac{-b}{h^2} \underset{\sim}{\Lambda} \underset{\sim}{v}$

Or in scalar form : $\dfrac{dv_i}{dt} = \dfrac{-b}{h^2} \lambda_i v_i = -\dfrac{b}{h^2} 4\sin^2\left(\dfrac{\pi k}{2N}\right) v_i$ .. (5)

Comparing with Dahlquist equation :

$\lambda = -\dfrac{b}{h^2}\left[4\sin^2\left(\dfrac{\pi k}{2N}\right)\right]$

The above evaluated value is real & negative as $b > 0$

$\lambda_k = -\dfrac{4b}{h^2}\sin^2\left(\dfrac{k\pi}{2N}\right) < 0$ .. (6)

For the given differential equation to be stable, equation (5) should be stable. Hence, we'll pose stability criteria on eigen values by equation (6)

Hence, to determine stability, we can evaluate $\lambda\Delta t$ & use stability diagram obtained earlier for leapfrog time integration schemes

$z = \lambda\Delta t = \lambda_k \Delta t = -\dfrac{4b\Delta t}{h^2}\sin^2\left(\dfrac{k\pi}{2N}\right)$

We found that for leapfrog scheme to be stable, z should be purely imaginary & bound $z \in (-i, i)$ as depicted in stability diagram as well. But for considered centre in space discretisation, eigen values & hence $z = \lambda\Delta t$ is found to be purely real.

Therefore for any Δt or h, it will never fall inside stability region in stability diagram.

Hence, Leap frog scheme with given centre in space discretization is unconditionally stable.

Solution 2 Given equation : $u_t = bu_{xx} + f$

Scheme :

$$\left(1+\frac{1}{12}\delta_x^{(2)}\right)(u_j^{n+1}-u_j^n) = \frac{bk}{2h^2}\delta_x^{(2)}(u_j^{n+1}+u_j^n) + \frac{k}{2}\left[f_j^{n+1} + \left(1+\frac{1}{6}\delta_x^{(2)}\right)f_j^n\right]$$

where, $\delta_x^{(2)}u_j^n = u_{j+1}^n - 2u_j^n + u_{j-1}^n$

$\delta_x^{(2)}$ is a kind of operator which operates with u_j^n like given in above.

$$\delta_x^{(2)}u_j^{n+1} = u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}$$

$$\delta_x^{(2)}f_j^n = f_{j+1}^n - 2f_j^n + f_{j-1}^n$$

$$LHS = u_j^{n+1} - u_j^n + \frac{1}{12}\left[\delta_x^{(2)}u_j^{n+1} - \delta_x^{(2)}u_j^n\right]$$

$$= u_j^{n+1} - u_j^n + \frac{1}{12}\left[u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1} - u_{j+1}^n + 2u_j^n - u_{j-1}^n\right]$$

$$= \frac{1}{12}u_{j+1}^{n+1} + \frac{5}{6}u_j^{n+1} + \frac{1}{12}u_{j-1}^{n+1} - \frac{u_{j+1}^n}{12} - \frac{5}{6}u_j^n - \frac{1}{12}u_{j-1}^n = \frac{u_{j+1}^{n+1} + 10u_j^{n+1} + u_{j-1}^{n+1} - u_{j+1}^n - 10u_j^n - u_{j-1}^n}{12}$$

$$RHS = \frac{bk}{2h^2}\left[u_{j+1}^{n+1} + u_{j+1}^n - 2(u_j^{n+1}+u_j^n) + u_{j-1}^{n+1} + u_{j-1}^n\right]$$

$$+ \frac{k}{2}\left[f_j^{n+1} + f_j^n + \frac{1}{6}(f_{j+1}^n - 2f_j^n + f_{j-1}^n)\right]$$

$$\underbrace{\frac{u_j^{n+1}-u_j^n}{\Delta t}}_{(1)} + \frac{1}{12}\delta_x^{(2)}\frac{(u_j^{n+1}-u_j^n)}{\Delta t} = b\,\delta_x^{(2)}\frac{(u_j^{n+1}+u_j^n)}{2h^2} + \frac{f_j^{n+1}+f_j^n}{2} + \frac{\delta_x^{(2)}f_j^n}{12}$$

(2) (3) (4) (5)

Taylor series expansion for the terms :

$$u_j^{n+1} = u_j^n + \Delta t\frac{\partial u}{\partial t}\Big|_j^n + \frac{(\Delta t)^2}{2!}\frac{\partial^2 u}{\partial t^2}\Big|_j^n + \frac{(\Delta t)^3}{3!}\frac{\partial^3 u}{\partial t^3}\Big|_j^n + \dots$$

$$u_{j+1}^{n+1} = u_j^n + \Delta t\frac{\partial u}{\partial t}\Big|_j^n + h\frac{\partial u}{\partial x}\Big|_j^n + \frac{1}{2!}\left[(\Delta t)^2 u_{tt}\Big|_j^n + h\Delta t\,u_{xt}\Big|_j^n + h^2u_{xx}\Big|_j^n\right] + \dots$$

$$u_{j-1}^{n+1} = u_j^n + \Delta t\frac{\partial u}{\partial t}\Big|_j^n - hu_x\Big|_j^n + \frac{1}{2!}\left[(\Delta t)^2 u_{tt}\Big|_j^n - h\Delta t\,u_{xt}\Big|_j^n + h^2u_{xx}\Big|_j^n + \dots\right]$$

Rearranging (1) :

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} = u_j^n + O(\Delta t)$$

Now, $$\delta_x^{(2)}u_j^{n+1} = 2\left[\frac{h^2}{2!}u_{xx}\Big|_j^n + \frac{h^4}{4!}u_{xxxx}\Big|_j^n + \dots + \frac{\Delta t\,h^2}{3!}u_{xxx}\Big|_j^n + \frac{(\Delta t)^2h^2}{4!}u_{xxxx}\Big|_j^n + \frac{(\Delta t)^3h^2}{5!}u_{xxttt}\Big|_j^n + \dots\right]$$

$$\delta_x^{(2)}u_j^n = 2\left[\frac{h^2}{2!}u_{xx}\Big|_j^n + \frac{h^4}{4!}u_{xxxx}\Big|_j^n + \frac{h^6}{6!}u_{xxxxxx}\Big|_j^n\right]$$

Evaluating (2):

$$\frac{1}{12}\delta_x^{(2)}\frac{(u_j^{n+1}-u_j^n)}{\Delta t} = \frac{1}{6}\left[\frac{h^2}{3!}u_{xxt}\Big|_j^n + \frac{(\Delta t)h^2}{4!}u_{xxtt}\Big|_j^n + \frac{(\Delta t)^2h^2}{5!}u_{xxttt}\Big|_j^n + \dots\right]$$

$$\frac{1}{12}\delta_x^{(2)}\frac{(u_j^{n+1}-u_j^n)}{\Delta t} = O(h^2) + [O(\Delta t)h^2]$$

Evaluating (3):

$$b\,\delta_x^{(2)}\left(\frac{u_j^{n+1}+u_j^n}{\Delta t}\right) = bu_{xx}\Big|_j^n + 2b\left[\frac{h^2}{4!}u_{xxxx}\Big|_j^n + \frac{h^4}{6!}u_{xxxxxx}\Big|_j^n + \dots\right]$$
$$+ b\left[\frac{(\Delta t)}{3!}u_{xxt}\Big|_j^n + \frac{(\Delta t)^2}{4!}u_{xxtt}\Big|_j^n + \frac{(\Delta t)^3}{5!}u_{xxttt}\Big|_j^n + \dots\right]$$

$$b\,\delta_x^{(2)}\frac{(u_j^{n+1}+u_j^n)}{\Delta t} = bu_{xx}\Big|_j^n + O(h^2) + O(\Delta t)$$

Now taylor series for function f:

$$f_j^{n+1} = f\Big|_j^n + \Delta t\, f_t\Big|_j^n + \frac{(\Delta t)^2}{2!}f_{tt}\Big|_j^n + \frac{(\Delta t)^3}{3!}f_{ttt}\Big|_j^n + \dots$$

Evaluating (4):

$$\frac{f_j^{n+1}+f_j^n}{2} = f(x_j,t_n) + \frac{1}{2}\left[\Delta t\, f_t\Big|_j^n + \frac{\Delta t^2}{2}f_{tt}\Big|_j^n + \frac{(\Delta t)^3}{3!}f_{ttt}\Big|_j^n + \dots\right]$$

$$\therefore \frac{f_j^{n+1}+f_j^n}{2} = f(x_j,t_n) + O(\Delta t)$$

(5) is given by similar expression:

$$\delta_x^{(2)}f_j^n = 2\left[\frac{h^2}{2!}f_{xx} + \frac{h^4}{4!}f_{xxxx} + \frac{h^6}{6!}f_{xxxxxx}\right]\Big|_j^n$$

$$\therefore \frac{\delta_x^{(2)}f_j^n}{12} = O(h^2)$$

Substituting the terms:

$$u_t(x_j,t_n) = bu_{xx}(x_j,t_n) + f(x_j,t_n) + O(\Delta t) + O(h^2) + O((\Delta t)h^2)$$

Comparing with exact equation:

$$u_t(x_j,t_n) = bu_{xx}(x_j,t_n) + f(x_j,t_n)$$

Order of accuracy in space & time are given by $O(\Delta t)$ & $O(h^2)$

In difference form as given in question, substituting evaluated terms:

$$\left[1+\frac{1}{12}\delta_x^{(2)}\right](u_j^{n+1}-u_j^n) = \frac{b\Delta t}{2h^2}\delta_x^{(2)}(u_j^{n+1}+u_j^n) + \frac{\Delta t}{2}\left[f_j^{n+1} + \left(1+\frac{1}{6}\delta_x^{(2)}\right)f_j^n\right]$$
$$+ O(\Delta t)^2 + O((\Delta t)h^2)$$

Stability Analysis :

$$(u_j^{n+1}-u_j^n)+\frac{1}{12}\delta_x^{(2)}(u_j^{n+1}-u_j^n)=\frac{b\Delta t}{2h^2}\delta_x^{(2)}(u_j^{n+1}+u_j^n)+\frac{(\Delta t)(f_j^{n+1}+f_j^n)}{2}+\frac{(\Delta t)\delta_x^{(2)}f_j^n}{12}$$

$$u_j^{n+1}-u_j^n+\frac{1}{12}\left(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}-u_{j+1}^n+2u_j^n-u_{j-1}^n\right)=\frac{b\Delta t}{2h^2}\left(u_{j+1}^{n+1}-2u_j^{n+1}+u_{j-1}^{n+1}+u_{j+1}^n-2u_j^n+u_{j-1}^n\right)$$
$$+\left[\frac{(\Delta t)(f_j^{n+1}+f_j^n)}{2}+\frac{(\Delta t)(f_{j+1}^n-2f_j^n+f_{j-1}^n)}{12}\right]$$

Let $N=\frac{b\Delta t}{h^2}$

Von Neuman Stability Analysis : $u_j^n \sim g^n e^{ij\theta}$

where, g = amplification factor

$||g(\theta)|| \leq 1$ (for stability)

Substituting in terms :

$$(g^{n+1}e^{ij\theta}-g^n e^{ij\theta})+\frac{1}{12}\left(g^{n+1}e^{i(j+1)\theta}-2g^{n+1}e^{ij\theta}+g^{n+1}e^{i(j-1)\theta}-g^n e^{i(j+1)\theta}+2g^n e^{ij\theta}-g^n e^{i(j-1)\theta}\right)$$
$$=\frac{N}{2}\left[g^{n+1}e^{i(j+1)\theta}-2g^{n+1}e^{ij\theta}+g^{n+1}e^{i(j-1)\theta}+g^n e^{i(j+1)\theta}-2g^n e^{ij\theta}+g^n e^{i(j-1)\theta}\right]$$

$$\Rightarrow g=\frac{\frac{5}{6}+\frac{\cos\theta}{6}+N\cos\theta-N}{\frac{5}{6}+\frac{\cos\theta}{6}+N\cos\theta-N}$$

$$g=1+\frac{12(\cos\theta-1)N}{(5+\cos\theta)-6(\cos\theta-1)N}$$

For stability $||g||\leq 1$

$$\therefore \left|1+\frac{12(\cos\theta-1)N}{(5+\cos\theta)-6(\cos\theta-1)N}\right|\leq 1$$

$$\Rightarrow -2\leq\frac{12(\cos\theta-1)N}{5+\cos\theta-6(\cos\theta-1)N}\leq 0$$

Left side inequality : $(5+\cos\theta)\geq 0$ (always true)

Right side inequality : $\frac{12(\cos\theta-1)N}{5+\cos\theta-6(\cos\theta-1)N}\leq 0$

$\cos\theta\leq 1 \Rightarrow \cos\theta-1\leq 0 \Rightarrow$ Numerator <0

So, Denominator should always be positive

$(5+\cos\theta)-6(\cos\theta-1)N>0$

$\Rightarrow 5+\cos\theta > 6(\cos\theta-1)N$

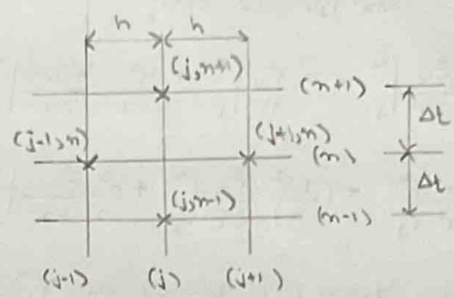
As, $\cos\theta-1\leq 0$, i.e., $6(\cos\theta-1)$ is negative

$$N>\frac{(5+\cos\theta)}{6(\cos\theta-1)}$$

For above inequality RHS will be always negative as numerator is positive & denominator negative for all θ values. As we know $N > 0$, therefore above inequality always hold

Solution (3)

Diffusion equation : $u_t = b u_{xx}$, $b > 0$



Dufort - Frankel Scheme :

$$u_j^{n+1} = u_j^{n-1} + \frac{2b\Delta t}{h^2}\left(u_{j+1}^{n} - u_j^{n+1} - u_j^{n-1} + u_{j-1}^{n}\right)$$

Expanding the terms by Taylor series :

$$u_j^{n+1} = u_j^n + (\Delta t)\frac{\partial u}{\partial t}\Big|_{(x_j, t_n)} + \frac{(\Delta t)^2}{2!}\frac{\partial^2 u}{\partial t^2}\Big|_{(x_j, t_n)} + \frac{(\Delta t)^3}{3!}\frac{\partial^3 u}{\partial t^3}\Big|_{(x_j, t_n)} + \frac{(\Delta t)^4}{4!}\frac{\partial^4 u}{\partial t^4}\Big|_{(x_j, t_n)} + \cdots$$

$$u_j^{n-1} = u_j^n - (\Delta t)\frac{\partial u}{\partial t}\Big|_{(x_j, t_n)} + \frac{(\Delta t)^2}{2!}\frac{\partial^2 u}{\partial t^2}\Big|_{(x_j, t_n)} - \frac{(\Delta t)^3}{3!}\frac{\partial^3 u}{\partial t^3}\Big|_{(x_j, t_n)} + \frac{(\Delta t)^4}{4!}\frac{\partial^4 u}{\partial t^4}\Big|_{(x_j, t_n)} + \cdots$$

$$u_{j+1}^m = u_j^n + h\frac{\partial u}{\partial x}\Big|_{(x_j, t_n)} + \frac{h^2}{2!}\frac{\partial^2 u}{\partial x^2}\Big|_{(x_j, t_n)} + \frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3}\Big|_{(x_j, t_n)} + \frac{h^4}{4!}\frac{\partial^4 u}{\partial x^4}\Big|_{(x_j, t_n)} + \cdots$$

$$u_{j-1}^n = u_j^n - h\frac{\partial u}{\partial x}\Big|_{(x_j, t_n)} + \frac{h^2}{2!}\frac{\partial^2 u}{\partial x^2}\Big|_{(x_j, t_n)} - \frac{h^3}{3!}\frac{\partial^3 u}{\partial x^3}\Big|_{(x_j, t_n)} + \frac{h^4}{4!}\frac{\partial^4 u}{\partial x^4}\Big|_{(x_j, t_n)} - \cdots$$

$$L(u) = \frac{\partial u}{\partial t} - b\frac{\partial^2 u}{\partial x^2} = 0$$

$$T_j^n = \frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} - \frac{b}{h^2}\left[u_{j-1}^n + u_{j+1}^n - (u_j^{n+1} + u_j^{n-1})\right] = \text{Term 1} - \cancel{\tfrac{b}{\Delta t}}\,[\text{Term 2}]$$

Let $\Delta t = k$

Solving Term 1 :

$$\frac{1}{2k}\Bigg[u_j^n + k\frac{\partial u}{\partial t}\Big|_j^n + \frac{k^2}{2}\frac{\partial^2 u}{\partial t^2}\Big|_j^n + \frac{k^3}{6}\frac{\partial^3 u}{\partial t^3}\Big|_j^n + \frac{k^4}{24}\frac{\partial^4 u}{\partial t^4}\Big|_j^n + O(k^5)$$
$$- u_j^n + k\frac{\partial u}{\partial t}\Big|_j^n - \frac{k^2}{2}\frac{\partial^2 u}{\partial t^2}\Big|_j^n + \frac{k^3}{6}\frac{\partial^3 u}{\partial t^3}\Big|_j^n - \frac{k^4}{24}\frac{\partial^4 u}{\partial t^4}\Big|_j^n + O(k^5)\Bigg]$$

$$= \frac{1}{2k}\left[2k\frac{\partial u}{\partial t}\Big|_j^n + \frac{2k^3}{6}\frac{\partial^3 u}{\partial t^3}\Big|_j^n + O(k^5)\right]$$

$$= \frac{\partial u}{\partial t}\Big|_j^n + \frac{k^2}{6}\frac{\partial^3 u}{\partial t^3}\Big|_j^n + O(k^4)$$

Solving Term 2 : $\frac{b}{h^2}\left[u_{j+1}^n + u_{j-1}^n - (u_j^{n+1} + u_j^{n-1})\right]$

$$\frac{b}{h^2}\Bigg\{u_j^n + h\frac{\partial u}{\partial x}\Big|_j^n + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}\Big|_j^n + \frac{h^3}{6}\frac{\partial^3 u}{\partial x^3}\Big|_j^n + \frac{h^4}{24}\frac{\partial^4 u}{\partial x^4}\Big|_j^n + \frac{h^5}{120}\frac{\partial^5 u}{\partial x^5}\Big|_j^n + O(h^6)$$
$$+ u_j^n - h\frac{\partial u}{\partial x}\Big|_j^n + \frac{h^2}{2}\frac{\partial^2 u}{\partial x^2}\Big|_j^n - \frac{h^3}{6}\frac{\partial^3 u}{\partial x^3}\Big|_j^n + \frac{h^4}{24}\frac{\partial^4 u}{\partial x^4}\Big|_j^n - \frac{h^5}{120}\frac{\partial^5 u}{\partial x^5}\Big|_j^n + O(h^6)$$
$$- \Big[u_j^n + k\frac{\partial u}{\partial t}\Big|_j^n + \frac{k^2}{2}\frac{\partial^2 u}{\partial t^2}\Big|_j^n + \frac{k^3}{6}\frac{\partial^3 u}{\partial t^3}\Big|_j^n + O(k^4)$$
$$+ u_j^n - k\frac{\partial u}{\partial t}\Big|_j^n + \frac{k^2}{2}\frac{\partial^2 u}{\partial t^2}\Big|_j^n - \frac{k^3}{6}\frac{\partial^3 u}{\partial t^3}\Big|_j^n + O(k^4)\Big]\Bigg\}$$

$$= \frac{b}{h^2}\left[2u_j^n + h^2\frac{\partial^2 u}{\partial x^2}\Big|_j^n + \frac{h^4}{12}\frac{\partial^4 u}{\partial x^4}\Big|_j^n + O(h^6) - \left(2u_j^n + k^2\frac{\partial^2 u}{\partial t^2}\Big|_j^n + O(k^4)\right)\right]$$

$$= \frac{b}{h^2}\left[h^2\frac{\partial^2 u}{\partial x^2}\Big|_j^n - k^2\frac{\partial^2 u}{\partial t^2}\Big|_j^n + \frac{h^4}{12}\frac{\partial^4 u}{\partial x^4}\Big|_j^n + O(h^6) + O(k^4)\right]$$

$$= b\left[\frac{\partial^2 u}{\partial x^2}\Big|_j^n - \left(\frac{k}{h}\right)^2\frac{\partial^2 u}{\partial t^2}\Big|_j^n + \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}\Big|_j^n + O(h^4) + O\left(\frac{k^4}{h^2}\right)\right]$$

Substituting Term 1 & Term 2 in T_i^n:

$$T_i^n = \frac{\partial u}{\partial t}\Big|_j^n + \frac{k^2}{6}\frac{\partial^3 u}{\partial t^3}\Big|_j^n - b\left[\frac{\partial^2 u}{\partial x^2}\Big|_j^n - \left(\frac{k}{h}\right)^2\frac{\partial^2 u}{\partial t^2}\Big|_j^n + \frac{h^2}{12}\frac{\partial^4 u}{\partial x^4}\Big|_j^n\right] + O\left(\frac{k^4}{h^2}, k^4, h^4\right)$$

$$= \left[\frac{\partial u}{\partial t} - b\frac{\partial^2 u}{\partial x^2} + b\left(\frac{k}{h}\right)^2\frac{\partial^2 u}{\partial t^2}\right]_j^n + \left[\frac{k^2}{6}\frac{\partial^3 u}{\partial t^3} + \frac{bh^2}{12}\frac{\partial^4 u}{\partial x^4}\Big|_j^n\right] + O\left(\frac{k^4}{h^2}, k^4, h^4\right)$$

For hyperbolic system, we best hope for $\frac{k}{h} \approx 1$

However if we use DuFort scheme with $k = 1$, the sol^n converge to $bu_{tt} + u_t = bu_{xx}$ (wave equation) and not the diffusion equation ($u_t = bu_{xx}$).

Scheme will only converge to $u_t = bu_{xx}$ if $\frac{k}{h} \to 0$

Stability Analysis : Let $N = \frac{k}{h^2}$

$$u_j^{n+1} - u_j^{n-1} = 2bN\left[u_{j+1}^n - (u_j^{n+1} + u_j^{n-1}) + u_{j-1}^n\right]$$

$$\Rightarrow (1+2bN)u_j^{n+1} - (1-2bN)u_j^{n-1} = 2bN(u_{j+1}^n + u_{j-1}^n)$$

Substitute $u_j^n = g^n e^{ijh\beta}$

$$(1+2bN)g^2 - (1-2bN) = 2bN(e^{ih\beta} + e^{-ih\beta})g$$

$$g_\pm = \frac{2bN\cos(h\beta) \pm \sqrt{1 - 4b^2N^2\sin^2(h\beta)}}{1+2bN}$$

CASE 1: $1 - 4b^2N^2 \geq 0 \Rightarrow |g_\pm| \leq \frac{2bN|\cos(h\beta)| + \sqrt{1}}{1+2bN} \leq \frac{2bN+1}{1+2bN} = 1$

CASE 2: $1 - 4b^2N^2 < 0 \Rightarrow |g_\pm|^2 = \frac{(2bN\cos(h\beta))^2 + 4b^2N^2\cos^2(h\beta) - 1}{1+2bN} = \frac{4b^2N^2 - 1}{(1+2bN)^2} = \frac{2bN-1}{1+2N} \leq 1$

∴ Du-Fort Frankel Scheme is explicit & unconditionally stable scheme for $u_t = bu_{xx}$.

Using Lax Richtmeyer Equivalence Theorem:

Consistency + Stability $\Leftrightarrow$ Convergence

DuFort Frankel Scheme:

(.) Problem is well posed (.) Unconditionally stable

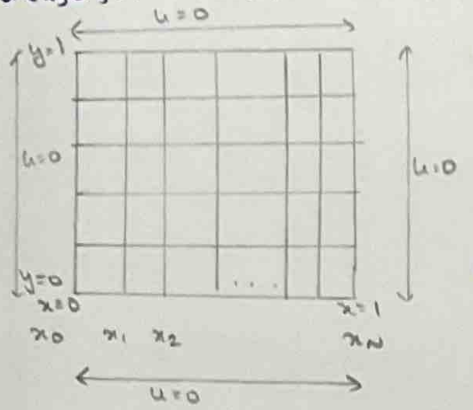
(.) Conditionally consistent when $\frac{\Delta t}{h^2}$ = constant

For convergence, required restriction in grid spacing h & time step Δt is $\frac{\Delta t}{h^2}$ = constant

Solution 4

Given : $u_t = \nabla^2 u + \sin(2\pi x)\sin(2\pi y)\sin(2\pi t)$ for $(x,y) \in (0,1)^2$

$u(x,0,t) = u(x,1,t) = 0$; $u(0,y,t) = u(1,y,t) = 0$



Similar to 2D diffusion ~~eqn~~ equation plus some additional terms

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \sin(2\pi x)\sin(2\pi y)\sin(2\pi t)$$

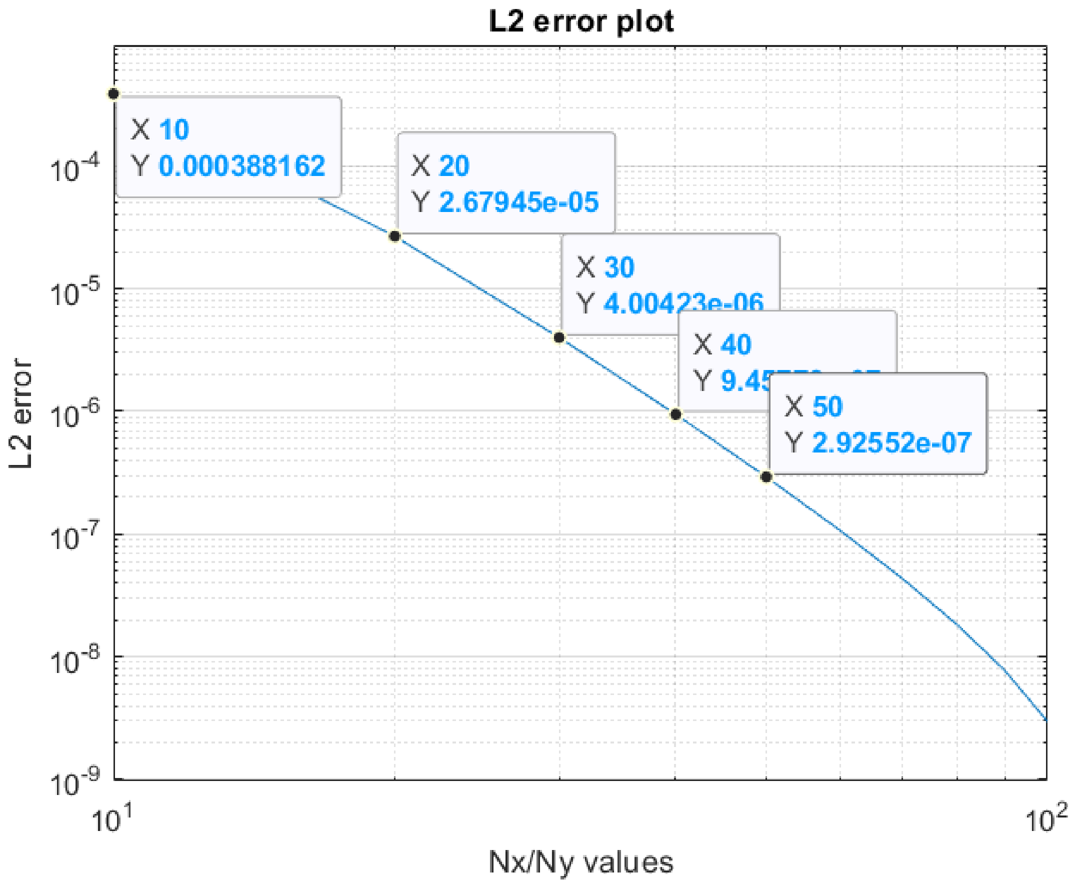
Crank Nicolson discretization :

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \frac{1}{2h^2}\Big[\left(u_{i+1,j}^{n+1} + u_{i-1,j}^{n+1} + u_{i,j+1}^{n+1} + u_{i,j-1}^{n+1} - 4u_{i,j}^{n+1}\right) + \left(u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n} - 4u_{i,j}^{n}\right)\Big] + \sin(2\pi x_i)\sin(2\pi y_i)\sin(2\pi t_n)$$

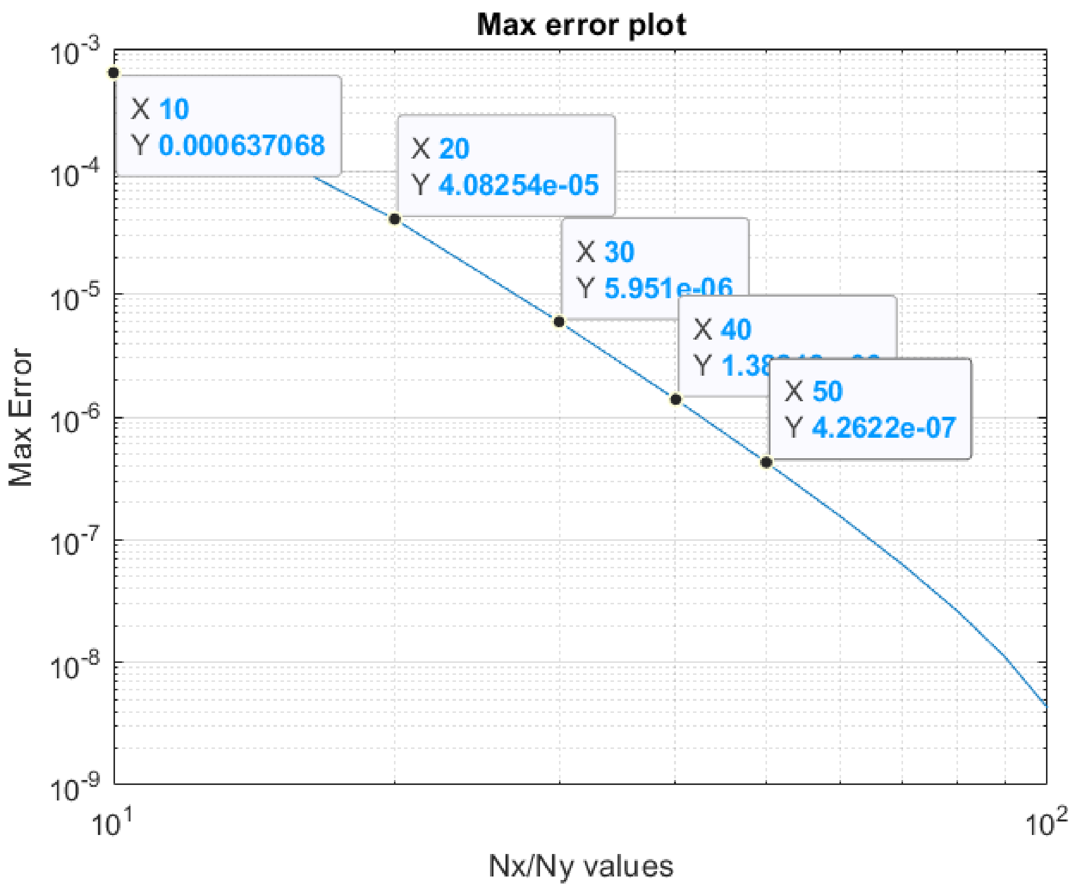
BC : $u_{i,0}^{n} = 0$, $u_{i,1}^{n} = 0$, $u_{0,j}^{n} = 0$, $u_{1,j}^{n} = 0$

Solution 4: (C++ code)

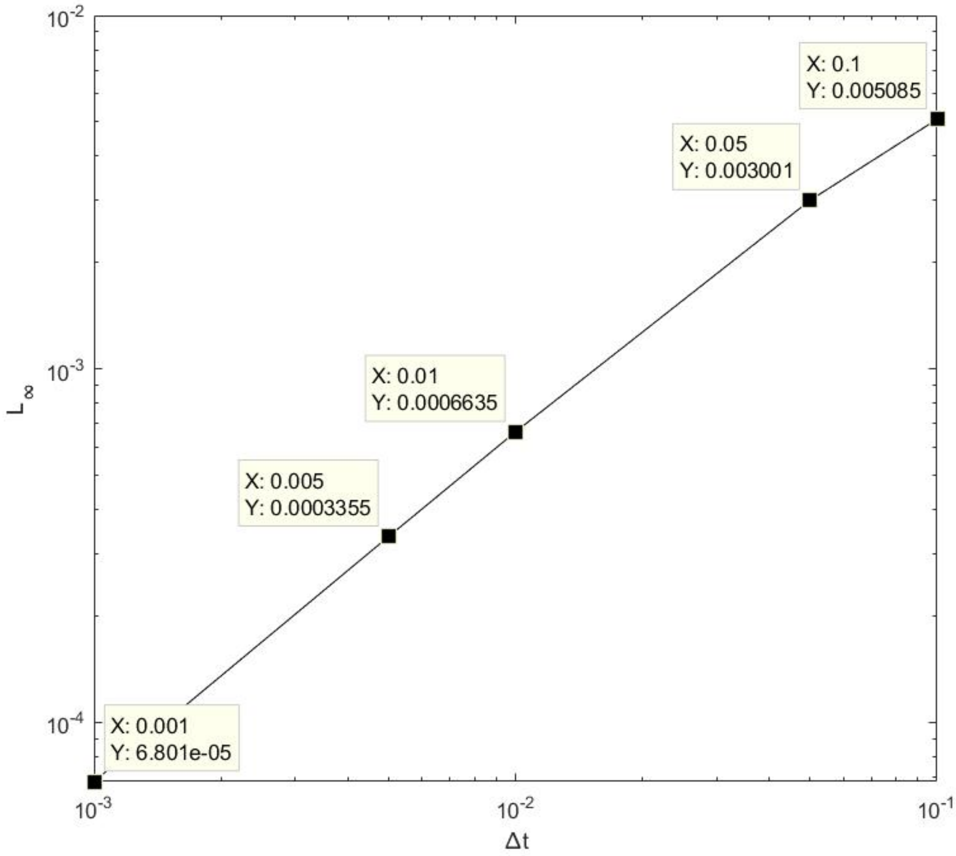
Error plots with dt=0.0001



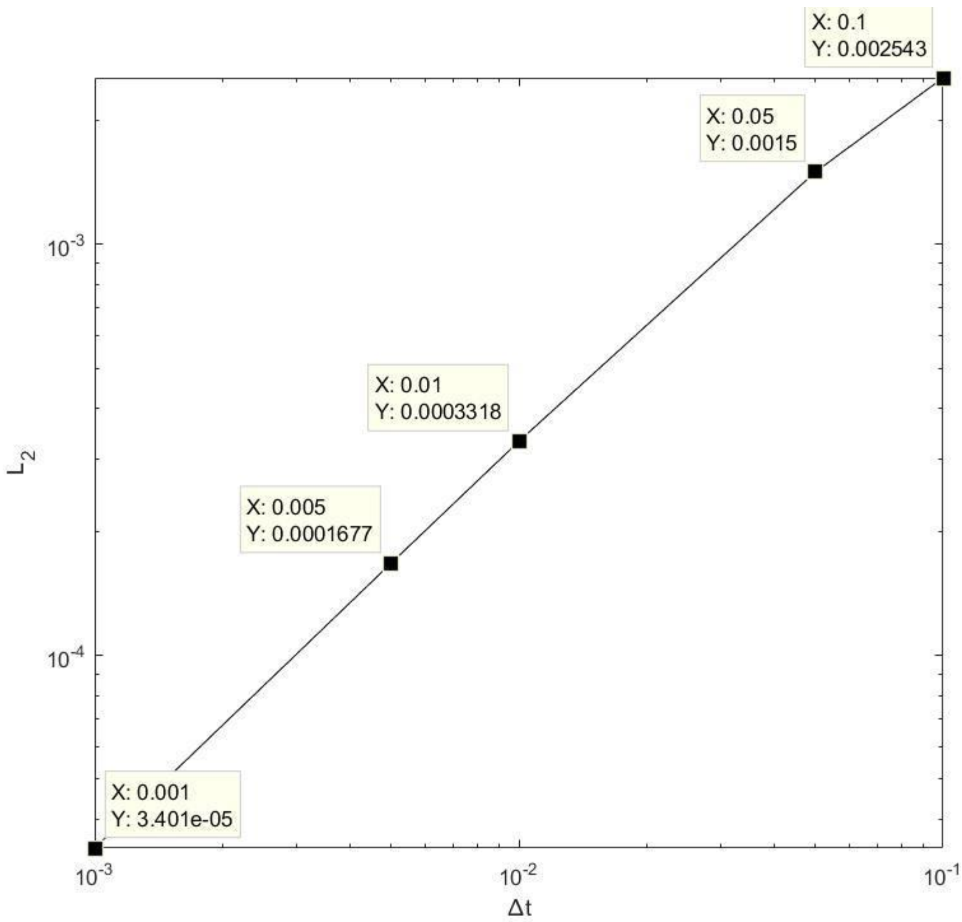
Rate of convergence in space= -4.688



Error Plots with N=200:



Rate of convergence = 0.9424



4.

$$u_t = \nabla u + \sin(2\pi x)\sin(2\pi y)\sin(2\pi t) \tag{4.1}$$

$$u_t = u_{xx} + u_{yy} + f(x,y)$$

$$(x,y) \in (0,1)^2$$

$$u(0,y,t) = u(1,y,t) = 0$$

$$u(x,0,t) = u(x,1,t) = 0$$

We discretize the domain into uniform grid points as below

$$(u_t)_{ij}^n = (u_{xx})_{ij}^n + \left(u_{yy}\right)_{ij}^n + f_{ij}^n$$

Where $i, j = 0,1, \dots N$

$$h = \Delta x = \Delta y = \frac{1}{N}, \qquad x_i = \frac{i}{N} = ih, \qquad y_j = \frac{j}{N} = jh$$

and $t_n = n\Delta t$

This can be approximated using standard 3-point discretization centre in space.

For forward in time discretisation i.e., FTCS

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} = \frac{u_{i-1.j}^n + u_{i,j-1}^n - 4u_{ij}^n + u_{i,j+1}^n + u_{i+1,j}^n}{h^2} + f_{ij}^n \tag{4.2}$$

For backward in time discretisation i.e., BTCS

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} = \frac{u_{i-1.j}^{n+1} + u_{i,j-1}^{n+1} - 4u_{ij}^{n+1} + u_{i,j+1}^{n+1} + u_{i+1,j}^{n+1}}{h^2} + f_{ij}^{n+1} \tag{4.3}$$

Therefore Crank-Nicholson Scheme is average of FTCS and BTCS, i.e.,

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} = \frac{u_{i-1.j}^n + u_{i,j-1}^n - 4u_{ij}^n + u_{i,j+1}^n + u_{i+1,j}^n}{2h^2} + \frac{u_{i-1.j}^{n+1} + u_{i,j-1}^{n+1} - 4u_{ij}^{n+1} + u_{i,j+1}^{n+1} + u_{i+1,j}^{n+1}}{2h^2} + \frac{f_{ij}^n + f_{ij}^{n+1}}{2} \tag{4.4}$$

$$\begin{aligned} u_{ij}^{n+1} + \mu\left(-u_{i-1.j}^{n+1} - u_{i,j-1}^{n+1} + 4u_{ij}^{n+1} - u_{i,j+1}^{n+1} - u_{i+1,j}^{n+1}\right) \\ = u_{ij}^n - \mu\left(-u_{i-1.j}^n - u_{i,j-1}^n + 4u_{ij}^n - u_{i,j+1}^n - u_{i+1,j}^n\right) + \frac{\Delta t}{2}\left(f_{ij}^n + f_{ij}^{n+1}\right) \end{aligned} \tag{4.5}$$

Where $i = 1,2 \dots, N-1$ and $j = 1,2 \dots, N-1$

$$where\ \mu = \frac{\Delta t}{2h^2} \tag{4.6}$$

This can be written as linear system of equations as

$$[\mathbf{I} + \mu\mathbf{A}]\mathbf{u}^{n+1} = [\mathbf{I} - \mu\mathbf{A}]\mathbf{u}^n + \mathbf{F} \tag{4.7}$$

Where **A** is a familiar matrix of which we have already evaluated Eigenvalues and Eigenvectors.

$$\mathbf{A} = \begin{bmatrix} 4 & -1 & 0 & \cdots & -1 & & & & & 0 & 0 \\ -1 & 4 & -1 & & & \ddots & & & & & 0 \\ \vdots & & \ddots & & & & & & & & \\ -1 & & & \ddots & & & & & & & \\ & \ddots & & \overbrace{\qquad}^{N\text{-}1} & \ddots & & \overbrace{\qquad}^{N\text{-}1} & \ddots & & & \\ & & -1 & \cdots & -1 & 4 & -1 & \cdots & -1 & & \\ & & & \ddots & & & \ddots & & & \ddots & \\ & & & & & & & \ddots & & & -1 \\ & & & & & & & & \ddots & & \vdots \\ 0 & & & & & \ddots & & & -1 & 4 & -1 \\ 0 & 0 & & & & & -1 & \cdots & 0 & -1 & 4 \end{bmatrix}_{(N-1)^2 \times (N-1)^2}, \mathbf{u}^n = \begin{bmatrix} \begin{pmatrix} u_{11}^n \\ u_{12}^n \\ \vdots \\ u_{1,N-1}^n \end{pmatrix} \\ \begin{pmatrix} u_{21}^n \\ \vdots \\ u_{2,N-1}^n \end{pmatrix} \\ \vdots \\ \vdots \\ \begin{pmatrix} \vdots \\ u_{ij}^n \\ \vdots \end{pmatrix} \\ \\ \vdots \\ \begin{pmatrix} u_{N-1,1}^n \\ u_{N-1,2}^n \\ \vdots \\ u_{N-1,N-1}^n \end{pmatrix} \end{bmatrix}$$

$$F_{ij} = \frac{\Delta t}{2}\left(f_{ij}^n + f_{ij}^{n+1}\right) = \frac{\Delta t}{2}\sin(2\pi x_i)\sin\left(2\pi y_j\right)\left\{\sin(2\pi t_n) + \sin\left(2\pi(t_n + \Delta t)\right)\right\} \tag{4.8}$$

This gives us an iterative update equation for each time step.

We can diagonalise **A** as

$$\mathbf{A} = \mathbf{P\Lambda P^{-1}} \tag{4.9}$$

Therefore, equivalently equation (4.7) can be written as

$$\mathbf{P\Lambda_1 P^{-1}u}^{n+1} = \mathbf{P\Lambda_2 P^{-1}u}^n + \mathbf{F}$$

where

$$\mathbf{\Lambda_1} = 1 + \mu\mathbf{\Lambda}, \qquad \mathbf{\Lambda_2} = 1 - \mu\mathbf{\Lambda} \tag{4.10}$$

Pre-multiplying with $\mathbf{P^{-1}}$

$$\mathbf{\Lambda_1 P^{-1}u}^{n+1} = \mathbf{\Lambda_2 P^{-1}u}^n + \mathbf{P^{-1}F}$$

If

$$\mathbf{P^{-1}u}^n = \mathbf{V}^n,\ \ \tilde{\mathbf{F}} = \mathbf{P^{-1}F} \tag{4.11}$$

The equation becomes of the form

$$\mathbf{\Lambda_1 V}^{n+1} = \mathbf{\Lambda_2 V}^n + \tilde{\mathbf{F}} \tag{4.12}$$

Which is decoupled now, and each equation can be written in scalar form as

$$\begin{gathered} \lambda_1^{(i,j)} v_{ij}^{n+1} = \lambda_2^{(i,j)} v_{ij}^{n+1} + \tilde{F}_{ij} \\ \left(1 + \mu\lambda^{(i,j)}\right) v_{ij}^{n+1} = \left(1 - \mu\lambda^{(i,j)}\right) v_{ij}^{n+1} + \tilde{F}_{ij} \\ v_{ij}^{n+1} = \frac{1 - \mu\lambda^{(i,j)}}{1 + \mu\lambda^{(i,j)}} v_{ij}^{n+1} + \frac{\tilde{F}_{ij}}{1 + \mu\lambda^{(i,j)}} \end{gathered} \tag{4.13}$$

The above one is the update iterative equation, where

$$\lambda^{(i,j)} = 4\left\{\sin^2\left(\frac{i\pi}{2N}\right) + \sin^2\left(\frac{j\pi}{2N}\right)\right\}$$

(4.14)

$$\mathbf{P} = \left[\left(\mathbf{X}^{(1,1)}, \mathbf{X}^{(1,2)} \ldots \mathbf{X}^{(1,N-1)}\right) \quad \left(\mathbf{X}^{(2,1)}, \mathbf{X}^{(2,2)} \ldots \mathbf{X}^{(2,N-1)}\right) \quad \cdots \quad \left(\mathbf{X}^{(N-1,1)}, \mathbf{X}^{(N-1,2)} \ldots \mathbf{X}^{(N-1,N-1)}\right)\right]$$

Where eigen vectors are given by

$$X_{ij}^{(p,q)} = \sin(p\pi x_i)\sin(q\pi y_j) = \sin\left(\frac{p\pi i}{N}\right)\sin\left(\frac{q\pi j}{N}\right)$$

(4.15)

We'll start by takin initial condition at time $t = 0$ as

$$u(x, y, 0) = 0$$

$$\mathbf{u}^{(0)} = 0$$

(4.16)

Therefore

$$\mathbf{V}^{(0)} = \mathbf{P}^{-1}\mathbf{u}^{(0)} = 0$$

At each iteration, the solution can be evaluated as

$$\mathbf{u}^n = \mathbf{P}\mathbf{V}^n$$

(4.17)

The MATLAB programming code to solve this system of equation is as follows:

```
clc
grid =[25 50 100 200 400];

L2=[];
Li=[];

for g=1:size(grid,2)

        N=grid(g);
        h=1/N;
        dt=0.001;
        T=0.6;

        %defining mu
        mu=dt/(2*h^2);

        %defining P matrix for diagonalisation
        P=zeros([(N-1)^2 (N-1)^2]);

        for p=1:(N-1)
          for q=1:(N-1)
            for i=1:(N-1)
              for j=1:(N-1)
                pp=(p-1)*(N-1);
                ii=(i-1)*(N-1);
                P(pp+q, ii+j)=(2/N)*sin(p*pi*i/N)*sin(q*pi*j/N);
              end
            end
          end
        end
```

```
        U=zeros([(N-1)^2, 1]);
        V=zeros([(N-1)^2, 1]);
        F=zeros([(N-1)^2, 1]);

        for t=0:dt:T

          %defining F
          for i=1:(N-1)
            for j=1:(N-1)
              F((i-1)*(N-1)+j,1)=0.5*dt*sin(2*pi*i/N)*sin(2*pi*j/N)*(sin(2*pi*t)+sin(2*pi*(t+dt)));
            end
          end

          Ftild= transpose(P)*F; %defining Ftild

          for i=1:(N-1)
            for j=1:(N-1)
              Lij=4*((sin(0.5*i*pi/N))^2+(sin(0.5*j*pi/N))^2);
              Fij=Ftild((i-1)*(N-1)+j,1);
              V((i-1)*(N-1)+j,1)=(V((i-1)*(N-1)+j,1)*(1-mu*Lij)+Fij)/(1+mu*Lij);
            end
          end
        end

U=P*V;
Us=reshape(U,(N-1),(N-1));
[X,Y]=meshgrid(h:h:(1-h),h:h:(1-h));
Uex=(2/(16*pi^2+1))*(sin(2*pi*T)+(-cos(2*pi*T)+exp(-8*T*pi^2))/(4*pi))*sin(2*pi*X).*sin(2*pi*Y);

        %L2 error
        Uerr=Us-Uex;
        L2(g)=sqrt(sum(sum(Uerr.^2)))/N;
        Li(g)=max(max(abs(Uerr)));
end

figure
loglog(grid,L2,'kO-');
xlabel('N');
ylabel('L_2');

figure
loglog(grid,Li,'kO-');
xlabel('N');
ylabel('L_∞');
```

Alternatively, Matrix bidiagonalization can also be used as follows by writing equation (4.5) as

$$u_{ij}^{n+1} - \frac{\Delta t}{2}\left\{\frac{u_{i-1,j}^{n+1} - 2u_{ij}^{n+1} + u_{i+1,j}^{n+1}}{h^2} + \frac{u_{i,j-1}^{n+1} - 2u_{ij}^{n+1} + u_{i,j+1}^{n+1}}{h^2}\right\}$$
$$= u_{ij}^{n} + \frac{\Delta t}{2}\left(\frac{u_{i-1,j}^{n} - 2u_{ij}^{n} + u_{i+1,j}^{n}}{h^2} + \frac{u_{i,j-1}^{n} - 2u_{ij}^{n} + u_{i,j+1}^{n}}{h^2}\right) + \frac{\Delta t}{2}\left(f_{ij}^{n} + f_{ij}^{n+1}\right)$$

This can be written as linear system of equations as

$$\mathbf{U}^{n+1} - \frac{\Delta t}{2}\left(\mathbf{D}_x^{(2)}\mathbf{U}^{n+1} + \mathbf{U}^{n+1}\mathbf{D}_y^{(2T)}\right) = \mathbf{U}^{n} + \frac{\Delta t}{2}\left(\mathbf{D}_x^{(2)}\mathbf{U}^{n} + \mathbf{U}^{n}\mathbf{D}_y^{(2T)}\right) + \mathbf{F} \tag{4.18}$$

Where 2nd order operators $\mathbf{D}^{(2)}$ are given by

$$\mathbf{D}^{(2)} = -\frac{1}{h^2}\mathbf{A}$$

$$(4.19)$$

Where we are already familiar with the tridiagonal matrix **A**

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ 0 & \dots & \ddots & \dots & 0 \\ 0 & \dots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix}_{(N-1)\times(N-1)}, \quad \mathbf{U}^n = \begin{bmatrix} u_{11}^n & u_{21}^n & & u_{N-1,1}^n \\ u_{12}^n & u_{22}^n & \dots & u_{N-1,2}^n \\ \vdots & \vdots & & \vdots \\ u_{1,N-1}^n & u_{2,N-1}^n & & u_{N-1,N-1}^n \end{bmatrix}_{(N-1)\times(N-1)} \tag{4.20}$$

And **F** is given by equation (4.8), just now it is a square matrix like **U** instead of a vector as was before.

Now, **A** can be diagonalised and written as

$$\mathbf{A} = \mathbf{P\Lambda P^{-1}}$$

Where **Λ** and **P** are obtained using Eigen values and vectors which we've already evaluated as follows:

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_{N-1} \end{bmatrix}, \qquad \lambda_k = 4\sin^2\left(\frac{\pi k}{2N}\right)$$

$$\mathbf{P} = [\mathbf{X}^{(1)} \quad \mathbf{X}^{(2)} \quad \cdots \quad \mathbf{X}^{(N-2)} \quad \mathbf{X}^{(N-1)}], \qquad \mathbf{P^{-1}} = \mathbf{P^T}, \qquad X_j^{(k)} = \sin(k\pi x_j), \qquad k = 1,2\dots,N-1$$

As we have taken uniform meshing, therefore from equation (4.19)

$$\mathbf{D}_x^{(2)} = -\frac{1}{h^2}\mathbf{P\Lambda}_x\mathbf{P^{-1}}, \qquad \mathbf{D}_y^{(2)} = -\frac{1}{h^2}\mathbf{P\Lambda}_y\mathbf{P^{-1}} \tag{4.21}$$

Substituting in (4.18)

$$\mathbf{U}^{n+1} + \frac{\Delta t}{2h^2}\left(\mathbf{P\Lambda}_x\mathbf{P^{-1}U}^{n+1} + \mathbf{U}^{n+1}\mathbf{P}^T\mathbf{\Lambda}_y\mathbf{P}^{-T}\right) = \mathbf{U}^n - \frac{\Delta t}{2h^2}\left(\mathbf{P\Lambda}_x\mathbf{P^{-1}U}^n + \mathbf{U}^n\mathbf{P}^T\mathbf{\Lambda}_y\mathbf{P}^{-T}\right) + \mathbf{F}$$

Pre-multiplying with $\mathbf{P^{-1}}$and post-multiplying with **P** , (also accounting $\mathbf{P}^T = \mathbf{P}$ as its symmetric)

$$\mathbf{P^{-1}U}^{n+1}\mathbf{P} + \mu\mathbf{\Lambda}_x\mathbf{P^{-1}U}^{n+1}\mathbf{P} + \mu\mathbf{P^{-1}U}^{n+1}\mathbf{P\Lambda}_y = \mathbf{P^{-1}U}^n\mathbf{P} - \mu\mathbf{\Lambda}_x\mathbf{P^{-1}U}^n\mathbf{P} - \mu\mathbf{P^{-1}U}^n\mathbf{P\Lambda}_y + \mathbf{P^{-1}FP}$$

If

$$\mathbf{P^{-1}U}^n\mathbf{P} = \mathbf{W}^n, \qquad \hat{\mathbf{F}} = \mathbf{P^{-1}FP}, \qquad \mathbf{U}^n = \mathbf{PW}^n\mathbf{P^{-1}} \tag{4.22}$$

The equation becomes of the form

$$\mathbf{W}^{n+1} + \mu\mathbf{\Lambda}_x\mathbf{W}^{n+1} + \mu\mathbf{W}^{n+1}\mathbf{\Lambda}_y = \mathbf{W}^n - \mu\mathbf{\Lambda}_x\mathbf{W}^n - \mu\mathbf{W}^n\mathbf{\Lambda}_y + \hat{\mathbf{F}} \tag{4.23}$$

Which is decoupled now, and each equation can be written as

$$w_{ij}^{n+1} + \mu\lambda_x^{(i)}w_{ij}^{n+1} + \mu\lambda_y^{(j)}w_{ij}^{n+1} = w_{ij}^n - \mu\lambda_x^{(i)}w_{ij}^n - \mu\lambda_y^{(j)}w_{ij}^n + \hat{F}_{ij}$$

$$\left\{1 + \mu\left(\lambda_x^{(i)} + \lambda_y^{(j)}\right)\right\}w_{ij}^{n+1} = \left\{1 - \mu\left(\lambda_x^{(i)} + \lambda_y^{(j)}\right)\right\}w_{ij}^{n+1} + \hat{F}_{ij}$$

$$w_{ij}^{n+1} = \frac{1 - \mu\left(\lambda_x^{(i)} + \lambda_y^{(j)}\right)}{1 + \mu\left(\lambda_x^{(i)} + \lambda_y^{(j)}\right)}w_{ij}^{n+1} + \frac{\hat{F}_{ij}}{1 + \mu\left(\lambda_x^{(i)} + \lambda_y^{(j)}\right)} \tag{4.24}$$

Which is very much like the equation (4.13). Here, the difference is in the dimensions of the matrix we are dealing with and arrangement of **u** & **F** matrix. Here in Matrix Bidiagonalization method, dimension of **P** is just $(N-1)\times(N-1)$ whilst in the previous method, it is $(N-1)^2 \times (N-1)^2$. Both the methods were attempted and found to give exact same results for error plots which will be discussed later.

The MATLAB programming code for this Bidiagonalization method is as follows:

```
clc
tgrid =[0.001 0.005 0.01 0.05 0.1];

L2=[];
Li=[];


for g=1:size(tgrid,2)
        N=200;
        h=1/N;
        dt=tgrid(g);
        T=0.6;

        %defining mu
        mu=dt/(2*h^2);

        %defining P matrix for diagonalisation
        P=zeros([(N-1) (N-1)]);

        for i=1:(N-1)
          for j=1:(N-1)
            P(i,j)=sqrt(2/N)*sin(j*pi*i/N);
          end
        end

U=zeros([(N-1) (N-1)]);
W= zeros([(N-1) (N-1)]);
F= zeros([(N-1) (N-1)]);

        for t=0:dt:T
          %defining F
          for i=1:(N-1)
            for j=1:(N-1)
              F(i,j)=0.5*dt*sin(2*pi*i/N)*sin(2*pi*j/N)*(sin(2*pi*t)+sin(2*pi*(t+dt)));
            end
          end
          Ftild= transpose(P)*F*P;        %defining Ftild
          for i=1:(N-1)
            for j=1:(N-1)
              Lij=4*((sin(0.5*i*pi/N))^2+(sin(0.5*j*pi/N))^2);
              W(i,j)=(W(i,j)*(1-mu*Lij)+Ftild(i,j))/(1+mu*Lij);
            end
          end
        end

        U=P*W*transpose(P);


[X,Y]=meshgrid(h:h:(1-h),h:h:(1-h));
Uex=(2/(16*pi^2+1))*(sin(2*pi*T)+(-cos(2*pi*T)+exp(-8*T*pi^2))/(4*pi))*sin(2*pi*X).*sin(2*pi*Y);

        %L2 error
        Uerr=U-Uex;
        L2(g)=sqrt(sum(sum(Uerr.^2)))/N;
        Li(g)=max(max(abs(Uerr)));

end
```

Exact Solution

Now to evaluate various errors, exact solution is required. We'll use Fourier Series to solve as follows:

$$u_t = \nabla u + \sin(2\pi x)\sin(2\pi y)\sin(2\pi t)$$

$$u_t = u_{xx} + u_{yy} + f(x,y)$$

$$(x,y) \in (0,1)^2$$

$$u(0,y,t) = u(1,y,t) = 0$$

$$u(x,0,t) = u(x,1,t) = 0$$

We'll assume the below form for $u\ and\ f$

$$u = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} a_{mn}(t)\sin(m\pi x)\sin(n\pi y) \tag{4.25}$$

$$f = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} f_{mn}(t)\sin(m\pi x)\sin(n\pi y)\,, \qquad f_{mn}(t) = \begin{cases}\sin(2\pi t)\,, & m,n = 2\\ 0, & m,n \neq 2\end{cases} \tag{4.26}$$

Substituting these in the governing equation, we get

$$u_t = u_{xx} + u_{yy} + f(x,y)$$

$$\begin{aligned}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\frac{da_{mn}}{dt}\sin(m\pi x)\sin(n\pi y)\\ = -\sum_{m=1}^{\infty}\sum_{n=1}^{\infty} a_{mn}(t)\sin(m\pi x)\sin(n\pi y)\,[(m\pi)^2 + (n\pi)^2]\\ + \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} f_{mn}(t)\sin(m\pi x)\sin(n\pi y)\end{aligned} \tag{4.27}$$

For $m,n \neq 2$

Comparing the coefficient gives

$$\frac{da_{mn}}{dt} = -a_{mn}[(m\pi)^2 + (n\pi)^2]$$

Integrating, we get

$$a_{mn}(t) = C_{mn}exp\{-(m^2 + n^2)\pi^2 t\} \tag{4.28}$$

Where constant C_{mn} is evaluated using initial condition at time $t = 0$. We have

$$u(x,y,0) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} a_{mn}(0)\sin(m\pi x)\sin(n\pi y) = 0$$

$$where, a_{mn}(0) = C_{mn}$$

$$\therefore \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} C_{mn}\sin(m\pi x)\sin(n\pi y) = 0$$

$$\therefore C_{mn} = 0, \qquad \forall\, m,n \neq 2 \tag{4.29}$$

For $m, n = 2$

$$\frac{da_{22}}{dt} = -a_{22}[(2\pi)^2 + (2\pi)^2] + \sin(2\pi t)$$

$$\frac{da_{22}}{dt} + 8\pi^2 a_{22} = \sin(2\pi t)$$

(4.30)

This can be solved by using Integrating factor

$$IF = \exp\left(\int 8\pi^2 dt\right) = \exp(8\pi^2 t)$$

Multiplying equation (4.30) with IF, we get

$$\exp(8\pi^2 t)\frac{da_{22}}{dt} + \exp(8\pi^2 t)\, 8\pi^2 a_{22} = \exp(8\pi^2 t)\sin(2\pi t)$$

$$\frac{d\,[\exp(8\pi^2 t)\, a_{22}]}{dt} = \exp(8\pi^2 t)\sin(2\pi t)$$

Integrating with respect to t

$$\exp(8\pi^2 t)\, a_{22} = \int_0^t \exp(8\pi^2 t)\sin(2\pi t)\, dt$$

LHS can be evaluated by using integration by parts twice, which gives

$$\exp(8\pi^2 t)\, a_{22} = \left[\frac{2}{16\pi^2 + 1}\left(\sin(2\pi t) - \frac{cos(2\pi t)}{4\pi}\right)\exp(8\pi^2 t)\right]_0^t$$

$$\exp(8\pi^2 t)\, a_{22} = \frac{2}{16\pi^2 + 1}\left(\sin(2\pi t) - \frac{cos(2\pi t)}{4\pi}\right)\exp(8\pi^2 t) + \frac{2}{(16\pi^2 + 1)}\frac{1}{4\pi}$$

$$a_{22} = \frac{2}{16\pi^2 + 1}\left(\sin(2\pi t) - \frac{cos(2\pi t)}{4\pi} + \frac{\exp(-8\pi^2 t)}{4\pi}\right)$$

(4.31)

Substituting back, we get

$$u(x, y, t) = \frac{2}{16\pi^2 + 1}\left(\sin(2\pi t) - \frac{cos(2\pi t)}{4\pi} + \frac{\exp(-8\pi^2 t)}{4\pi}\right)\sin(2\pi x)\sin(2\pi y)$$

(4.32)

This is the required exact solution for given governing equation.

Error Evaluation

Now L_2 and L_∞ error can be evaluated using below equation

$$e_2 = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(u_{ij}^{calculated} - u_{ij}^{exact}\right)^2}{N^2}}$$

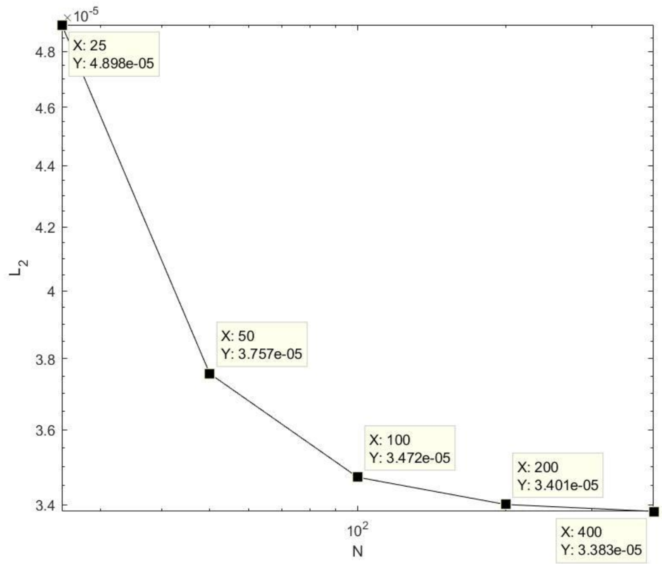
(4.33)

$$e_\infty = Maximum\left(\left|u_{ij}^{calculated} - u_{ij}^{exact}\right|\right)_{i,j=1}^{N}$$

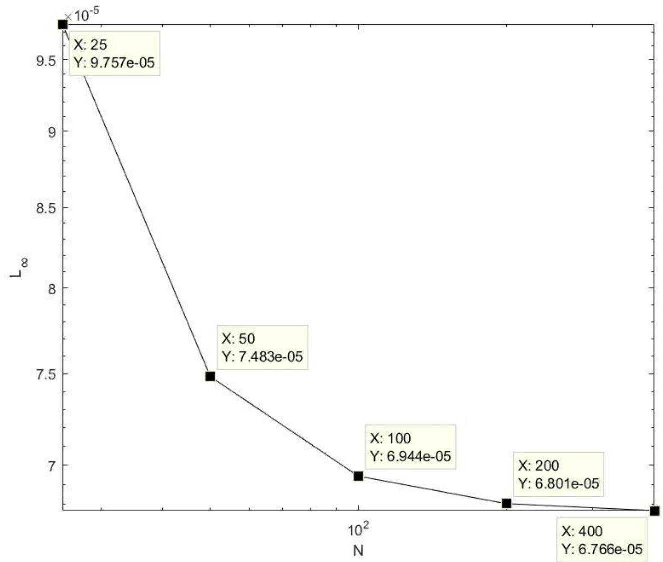
(4.34)

Rate of Convergence in Space

For various values of $N = 25, 50, 100, 200$ & 400, L_2 and L_∞ errors are evaluated for $\Delta t = 0.001$ and marching from time $t = 0\ to\ 0.6$. The plots obtained are as below



L_2 vs N log-log plot for $\Delta t = 0.001$



L_∞ vs N log-log plot for $\Delta t = 0.001$

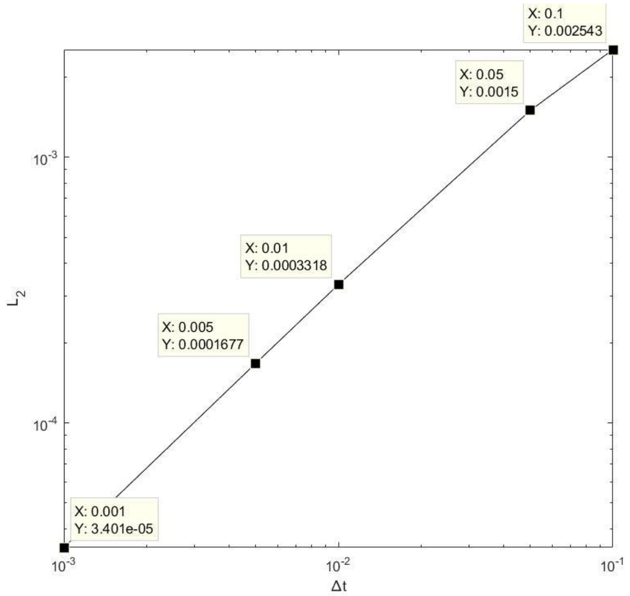
The rate of convergence i.e., slope of the $L_2\ vs\ N$ plot on a log-log scale when $N = 25$ to 50 is -0.3827 which is high compared to N beyond 50 till 400, where the slope is -0.0483.

For $L_\infty\ vs\ N$ plot on log-log scale slope is $-0.3827\ \&\ -0.0466$ for $N = 25$ to 50 & $N =$ 50 to 400 respectively.

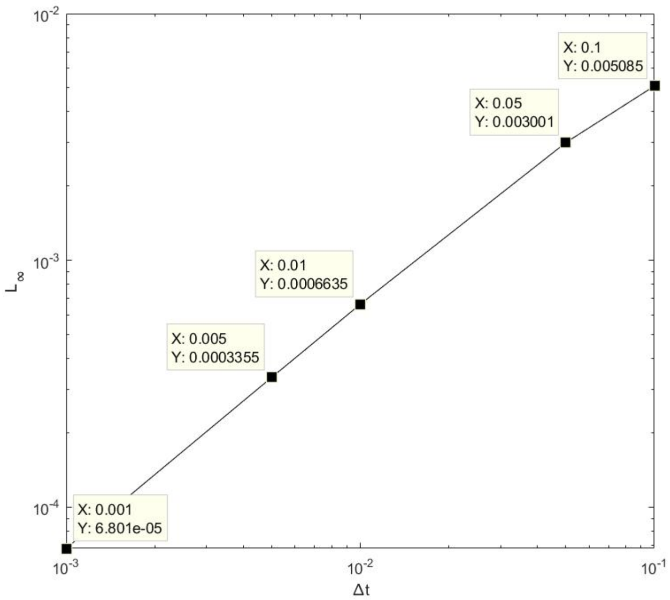
We observe from the graph that error decreases as we reduce mesh size i.e., increase the number of grid points. Hence slope has a negative value.

Rate of Convergence in Time

For various values of $\Delta t = 0.001, 0.005, 0.01, 0.05$ & 0.1 L_2 and L_∞ errors are evaluated for $N = 200$ and marching from $t = 0$ to 0.6. The plots obtained are as below



L_2 vs Δt log-log plot for $N = 200$



L_∞ vs Δt log-log plot for $N = 200$

The rate of convergence i.e., slope of the L_2 vs Δt plot on a log-log scale is 0.9424

For L_∞ vs Δt plot on log-log scale, the slope is same i.e., 0.9424.

We observe from the graph that error decreases as we reduce time step size, and hence slope has a positive value.