HW1, Math 228B

Date due 1/27/2011

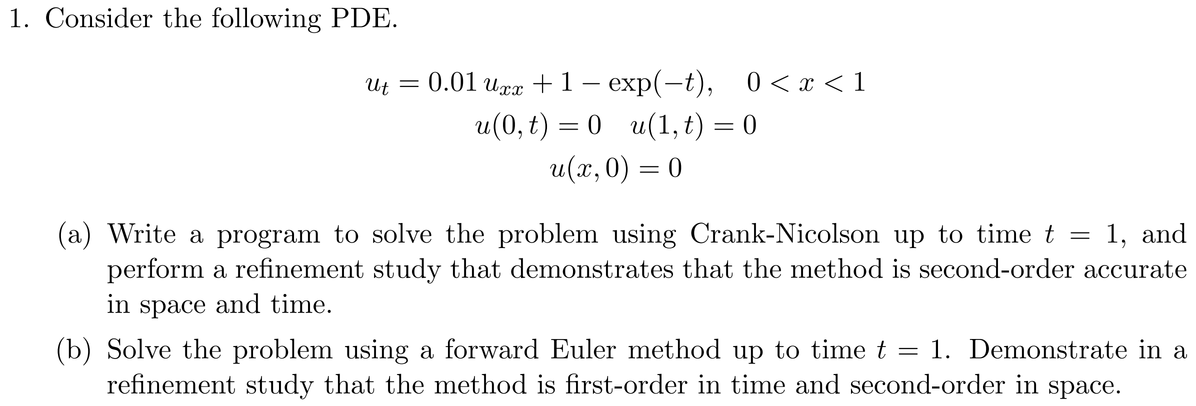
March 2011
University of California, Davis

Nasser M. Abbasi

March 2011 Compiled on February 27, 2021 at 2:13am

Contents

1 Problem1



1. Consider the following PDE.

$$u_t = 0.01\,u_{xx} + 1 - \exp(-t), \quad 0 < x < 1$$
$$u(0,t) = 0 \quad u(1,t) = 0$$
$$u(x,0) = 0$$

   (a) Write a program to solve the problem using Crank-Nicolson up to time $t = 1$, and perform a refinement study that demonstrates that the method is second-order accurate in space and time.

   (b) Solve the problem using a forward Euler method up to time $t = 1$. Demonstrate in a refinement study that the method is first-order in time and second-order in space.

Figure 1: Problem description

The goal of a refinement study is to perform a numerical experiment to determine the order of accuracy of a given finite difference scheme. The appendix of this problem contain a review of the idea behind refinement study.

The problem asked us to determine the order of accuracy in time and in space. A program implementing the above scheme was run a number of times, each time with a different initial value for the space and time step. To verify order of accuracy for the C-N scheme, the space and the time step were divided by 2 simultaneously before the start of each run. To verify order of accuracy for the forward Euler scheme, the space step was divided by 2 but the time was divided by 4. For both schemes , the program generated ratios of successive errors between the numerical solutions at the end of each run (1 second long run).

Convergence of this ratio to the value 4 implied the results we are asked to demonstrate.

In the following, the C-N and the forward Euler finite difference schemes are derived, then the numerical results presented, followed by a conclusion.

1.1 Part (a)

The method of lines (MOL) was used to implement the C-N scheme to solve for the numerical solution u. The equations are solved using Matlab's $u = A\backslash b$ where A is a sparse matrix (the system update matrix) constructed based on the C-N discretization. An efficient algorithm to solve for u in this scheme is Thomas algorithm version of Gaussian elimination. It is understood that this will automatically be done by Matlab "\" operator when it recognizes that the A matrix is a tridiagonal giving an $O(n)$ order for the solver where n is the number of unknowns.

Let the PDE be

$$d\ u_t - Du_{xx} + au = g(x,t) \tag{1}$$

$g(x,t)$ is an internal source with initial conditions as $u(x,t) = u0(x)$. The Dirichlet boundary conditions are

$$\begin{cases} u(0,t) = \alpha(t) \\ u(L,t) = \beta(t) \end{cases}$$

and Neumann boundary conditions are

$$\begin{cases} u_t(0,t) = \alpha(t) \\ u_t(L,t) = \beta(t) \end{cases}$$

The terms d, a above are constants, and D is the diffusion constant. For the C-N scheme (1) was discretized at point x_j with space step as h and with time step as k resulting in

$$d\,\frac{u_j^{n+1} - u_j^n}{k} = \frac{1}{2}\left(f_j^n + f_j^{n+1}\right)$$

Where f^n is the RHS of the PDE at time $t_n = nk$, so the above becomes

$$d\,\frac{u_j^{n+1} - u_j^n}{k} = \frac{1}{2}\Big[\left(Du_{xx} - au + g_j\right)^n + \left(Du_{xx} - au + g_j\right)^{n+1}\Big]$$

$$= \frac{1}{2}\left(D\frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{h^2} - au_j^n + g_j^n\right) +$$
$$\left(D\frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{h^2} - au^{n+1} + g_j^{n+1}\right)$$

$$= \frac{D}{2h^2}\left(u_{j-1}^n - 2u_j^n + u_{j+1}^n\right) + \frac{D}{2h^2}\left(u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}\right) -$$
$$\frac{a}{2}\left(u_j^n + u_j^{n+1}\right) + \frac{1}{2}\left(g_j^n + g_j^{n+1}\right)$$

collecting all terms at time $n+1$ to the left gives

$$u_j^{n+1}\left(1 + \frac{kD}{dh^2} + \frac{ak}{2d}\right) - \frac{kD}{2dh^2}u_{j-1}^{n+1} - \frac{kD}{2dh^2}u_{j+1}^{n+1} = u_j^n\left(1 - \frac{kD}{dh^2} - \frac{ak}{2d}\right) + u_{j-1}^n\frac{kD}{2dh^2} + u_{j+1}^n\frac{kD}{2dh^2} + \frac{k}{2d}\left(g_j^n + g_j^{n+1}\right)$$

Let

$$r_1 = \left(1 + \frac{kD}{dh^2} + \frac{ak}{2d}\right)$$
$$r_2 = \frac{kD}{2dh^2}$$
$$r_3 = \left(1 - \frac{kD}{dh^2} - \frac{ak}{2d}\right)$$
$$r_4 = \frac{k}{2d}$$

Then the above becomes

$$r_1 u_j^{n+1} - r_2 u_{j-1}^{n+1} - r_2 u_{j+1}^{n+1} = r_3 u_j^n + r_2 u_{j-1}^n + r_2 u_{j+1}^n + r_4 \left(g_j^n + g_j^{n+1} \right) \tag{2}$$

The above algebraic equation (2) is the C-N finite difference scheme for (1) and is valid for x_j at the internal points. Considering the case of both ends having Dirichlet boundary conditions, and using the following grid numbering[1]

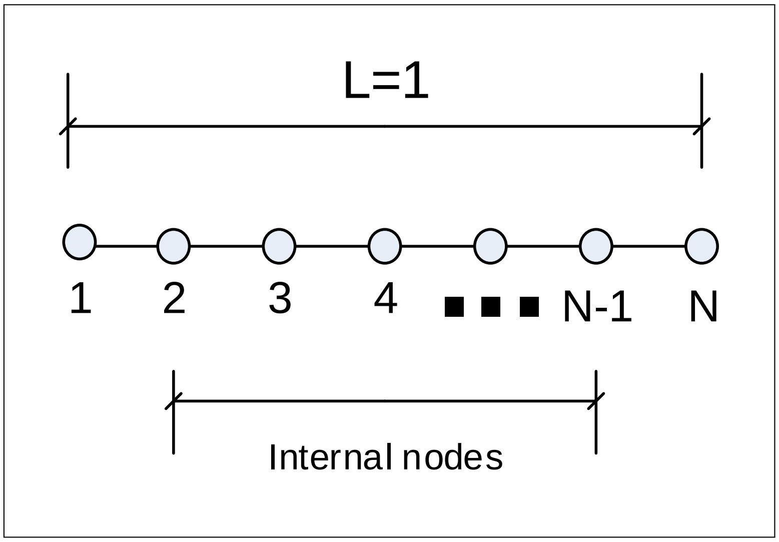


Figure 2: Problem grid format

Then (2) above is valid at the internal nodes numbered $j = 2 \cdots N-1$. Hence u_1^n will be the left boundary point and u_N^n will be the right boundary point. When the boundary conditions are Dirichlet, let $u_1^n = \alpha(t_n)$ and $u_N^n = \beta(t_n)$. Converting (2) to matrix form results in

$$\overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_1 & -r_2 & 0 & 0 & 0 & 0 \\ 0 & -r_2 & r_1 & -r_2 & 0 & 0 & 0 \\ 0 & 0 & -r_2 & r_1 & -r_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & \vdots & 0 \\ 0 & 0 & 0 & 0 & -r_2 & r_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}^{\mathbf{A}} \overbrace{\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ u_{N-2}^{n+1} \\ u_{N-1}^{n+1} \\ u_N^{n+1} \end{bmatrix}}^{\mathbf{x}} = \overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_3 & r_2 & 0 & 0 & 0 & 0 \\ 0 & r_2 & r_3 & r_2 & 0 & 0 & 0 \\ 0 & 0 & r_2 & r_3 & r_2 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & 0 & r_2 & r_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ u_4^n \\ \vdots \\ u_{N-1}^n \\ u_N^n \end{bmatrix} + \begin{bmatrix} \alpha^{n+1} \\ r_2 \tilde{\alpha} + r_4 \tilde{g}_2 \\ r_4 \tilde{g}_3 \\ r_4 \tilde{g}_4 \\ \vdots \\ r_2 \tilde{\beta} + r_4 \tilde{g}_{N-1} \\ \beta^{n+1} \end{bmatrix}}^{\mathbf{b}}$$

Where in (3) $\alpha(t_{n+1}) + \alpha(t_n) \equiv \tilde{\alpha}$ and $\beta(t_{n+1}) + \beta(t_n) \equiv \tilde{\beta}$ and $g^n + g^{n+1} \equiv \tilde{g}$.

Equation (3) is in the form $Au^{n+1} = b$. And u^{n+1} is solved for using $A \backslash b$. Notice that (3) is in the same form shown in class notes, which is

$$\left(I - \frac{Dk}{2} L \right) u^{n+1} = \left(I + \frac{Dk}{2} L \right) u^n + k f^{n+\frac{1}{2}} \tag{4}$$

[1] This is slightly different from the standard numbering format we used before.

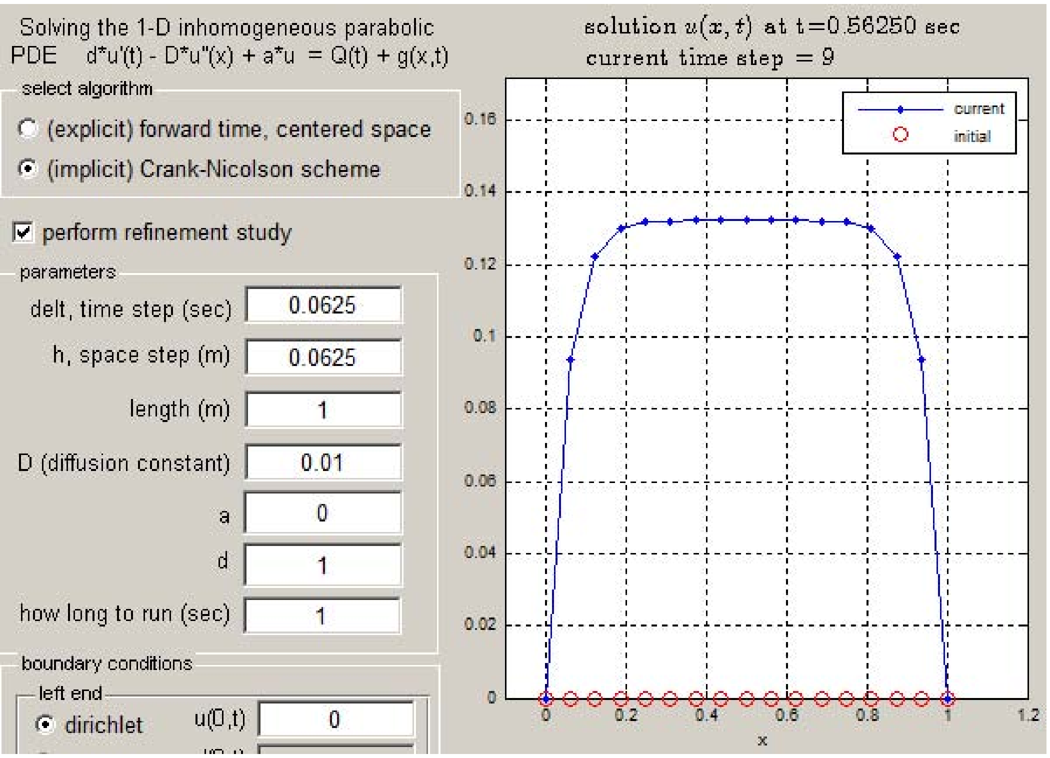
Where the update matrix $B = \left(I - \frac{Dk}{2}L\right)^{-1}\left(I + \frac{Dk}{2}L\right)$. L is the standard Laplace operator for $1D$ problem given by

$$\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$$

Notice that (3) compared to (4), has additional terms included in the RHS in order to support the general form the parabolic PDE. Equation (4) represents the diffusion pde $u_t - Du_{xx} = 0$.

u_j^n when $n = 0$ is obtained from initial conditions. The first step solves for u_j^1 which is then used in the second second step to solve for u_j^2 and so on, until the maximum time to solve for is reached. Since Dirichlet boundary conditions are used, u_1^n (the solution at the left edge) and u_N^n, the solution at the right edge are always known. The above system is solved only for the internal nodes. Next section shows the numerical results.

Result for part(a) The above scheme was implemented with a GUI added to make it easier to use these algorithms. The following plot shows the numerical solution at $t = 1$.



Solving the 1-D inhomogeneous parabolic PDE d*u'(t) - D*u"(x) + a*u = Q(t) + g(x,t)

select algorithm

○ (explicit) forward time, centered space

● (implicit) Crank-Nicolson scheme

☑ perform refinement study

parameters

delt, time step (sec) 0.0625

h, space step (m) 0.0625

length (m) 1

D (diffusion constant) 0.01

a 0

d 1

how long to run (sec) 1

boundary conditions

left end

● dirichlet u(0,t) 0

solution $u(x,t)$ at t=0.56250 sec

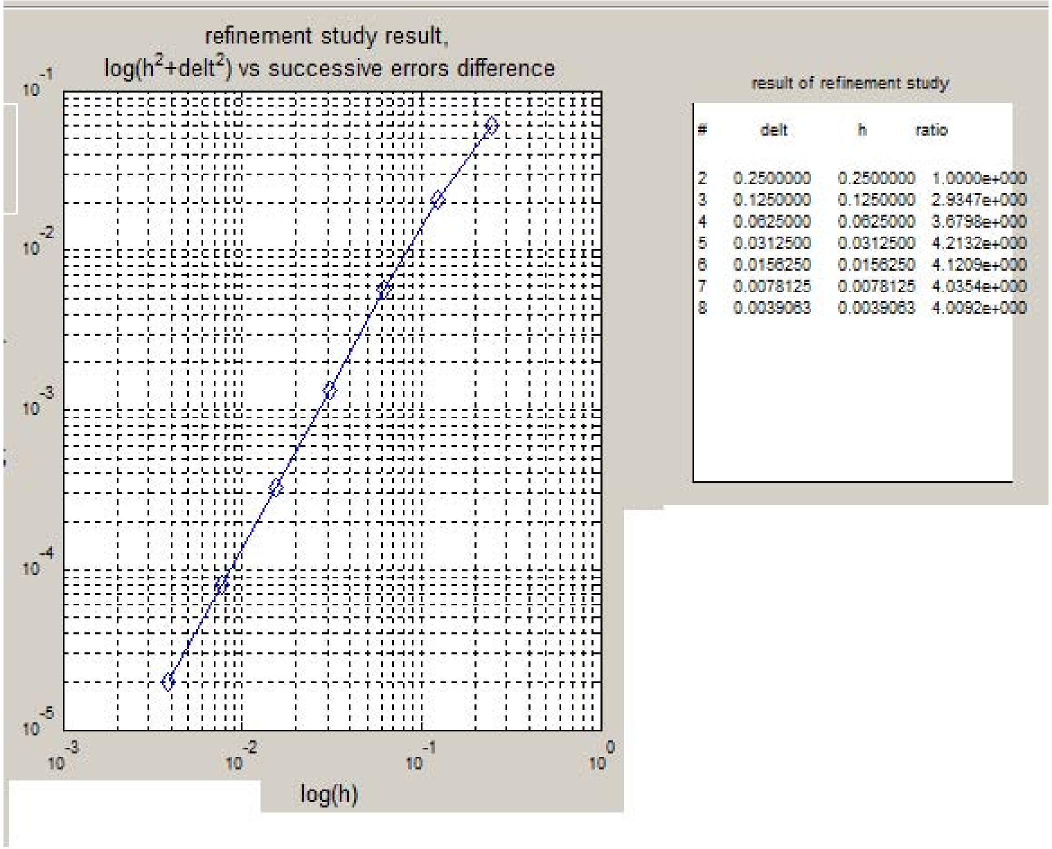
current time step = 9

Figure 3: shows the numerical solution at $t = 1$

The following is the ratio error table. This table shows that the ratio converged to 4.

#	delt	h	ratio
2	0.2500000	0.2500000	1.0000e+000
3	0.1250000	0.1250000	2.9347e+000
4	0.0625000	0.0625000	3.6798e+000
5	0.0312500	0.0312500	4.2132e+000
6	0.0156250	0.0156250	4.1209e+000
7	0.0078125	0.0078125	4.0354e+000
8	0.0039063	0.0039063	4.0092e+000

The following is the loglog plot of the above result. The x-axis represents h and the y-axis the difference in errors (absolute). The slope of the line is seen to be 2 implying a second order accuracy.



```
result of refinement study

#     delt        h           ratio

2     0.2500000   0.2500000   1.0000e+000
3     0.1250000   0.1250000   2.9347e+000
4     0.0625000   0.0625000   3.6798e+000
5     0.0312500   0.0312500   4.2132e+000
6     0.0156250   0.0156250   4.1209e+000
7     0.0078125   0.0078125   4.0354e+000
8     0.0039063   0.0039063   4.0092e+000
```

Figure 4: log log plot

Conclusion Since the ratio is 4 and since the time step and the space step were halved in each run, this implies C-N is second order in time and space.

1.2 Part (b)

Starting with the same PDE as in part (a)

$$d\ u_t - Du_{xx} + au = g(x,t) \tag{1}$$

All terms and boundary conditions and the solution domain are as shown in part (a).

For the forward Euler scheme (1) was discretized at point x_j with space step as h and with time step as k as follows

$$\begin{aligned} d\ \frac{u_j^{n+1} - u_j^n}{k} &= f_j^n \\ &= D\frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{h^2} - au_j^n + g_j^n \end{aligned}$$

Hence

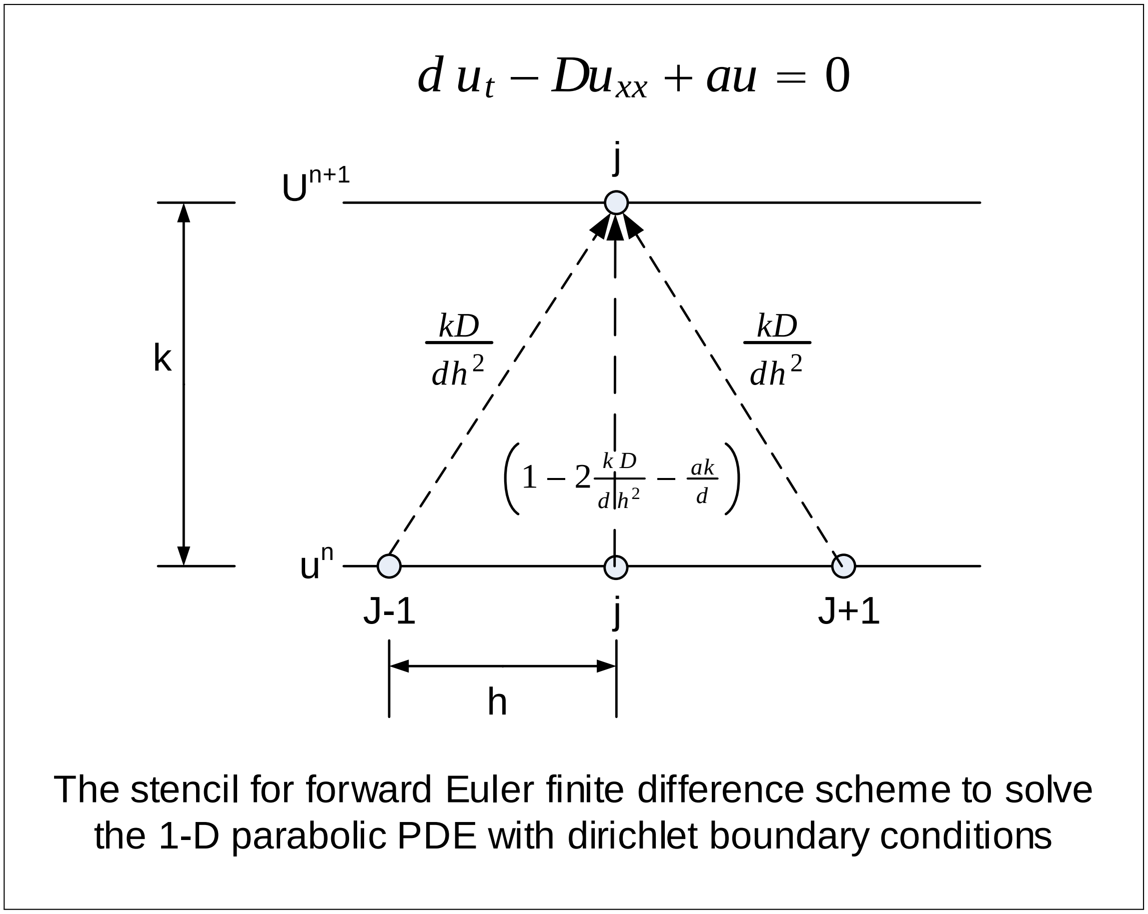
$$\begin{aligned} u_j^{n+1} &= u_j^n + \frac{kD}{dh^2}\left(u_{j-1}^n - 2u_j^n + u_{j+1}^n\right) - \frac{ak}{d}u_j^n + \frac{k}{d}g_j^n \\ &= \frac{kD}{dh^2}u_{j-1}^n + u_j^n\left(1 - 2\frac{k\,D}{d\,h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2}u_{j+1}^n + \frac{k}{d}g_j^n \end{aligned}$$

Let $r_1 = \left(1 - 2\frac{kD}{dh^2} - \frac{ak}{d}\right), r_2 = \frac{kD}{dh^2},, r_3 = \frac{k}{d}$, the above becomes

$$u_j^{n+1} = r_2 u_{j-1}^n + r_1 u_j^n + r_2 u_{j+1}^n + r_3 g_j^n \tag{2}$$

The above algebraic equation (2) is the forward Euler finite difference scheme for (1) and is valid for x_j at the internal points.

Therefore, the stencil for the forward Euler scheme for the 1D parabolic PDE is


$d\,u_t - Du_{xx} + au = 0$
j
U^{n+1}
k
$\frac{kD}{dh^2}$
$\frac{kD}{dh^2}$
$\left(1 - 2\frac{k\,D}{d\,h^2} - \frac{ak}{d}\right)$
u^n
J-1
j
J+1
h

The stencil for forward Euler finite difference scheme to solve the 1-D parabolic PDE with dirichlet boundary conditions

Figure 5: stencile for forward Euler

Considering the case of both ends having Dirichlet boundary conditions, and using the same numbering as in part (a) then (2) is valid at the internal nodes $j = 2 \cdots N - 1$.

u_1^n will be the left boundary point and u_N^n will be the right boundary point. Let $u_1^n = \alpha(t_n)$ and $u_N^n = \beta(t_n)$. Converting (2) to matrix form results in

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \\ \vdots \\ u_{N-2}^{n+1} \\ u_{N-1}^{n+1} \\ u_N^{n+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_1 & r_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_2 & r_1 & r_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & r_2 & r_1 & r_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_2 & r_1 & r_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & r_2 & r_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ u_4^n \\ \vdots \\ u_{N-2}^n \\ u_{N-1}^n \\ u_N^n \end{bmatrix} + \begin{bmatrix} \alpha^{n+1} \\ r_2\alpha^n + r_3 g_2^n \\ r_3 g_3^n \\ r_3 g_4^n \\ \vdots \\ r_3 g_{N-2}^n \\ r_2\beta^n + r_3 g_{N-1}^n \\ \beta^{n+1} \end{bmatrix}$$

And now u^{n+1} is found by direct matrix/vector multiplication as shown above. No matrix inversion is required in this case since this is an explicit method.

Looking at the stencil above, an idea is now suggested to determine stability directly from the stencil diagram. By imposing that the weight on each edge in the directed graph not exceed unity, and that the total algebraic sum of the weight of the edge also not exceed unity. This includes any combination of edges involved. If this is always the case, then u_j^{n+1} will always have an amplitude $\leq u_j^n$ since the weights are never more than 1 no matter what combinations are used. This idea is applied to this problem with $u_t + Du_{xx} = 0$, hence $a = 0$ and $d = 1$. This gives that following conditions on the edges shown in the stencil diagram above

$$\begin{cases} (1) \quad \frac{kD}{h^2} \leq 1 & \text{Condition on } j-1 \text{ or } j+1 \text{ separately} \\ (2) \quad 2\frac{kD}{h^2} \leq 1 & \text{Condition on } j-1 \text{ and } j+1 \text{ added together} \\ (3) \quad \left|1 - 2\frac{k\,D}{h^2}\right| \leq 1 & \text{Condition on the } j \text{ edge} \\ (4) \quad \left|\frac{kD}{h^2} + 1 - 2\frac{k\,D}{h^2}\right| \leq 1 & \text{Condition on the } j \text{ edge with either } j-1 \text{ or } j+1 \\ (5) \quad \left|2\frac{kD}{h^2} + 1 - 2\frac{k\,D}{h^2}\right| \leq 1 & \text{Condition that all edges sum to less than 1} \end{cases}$$

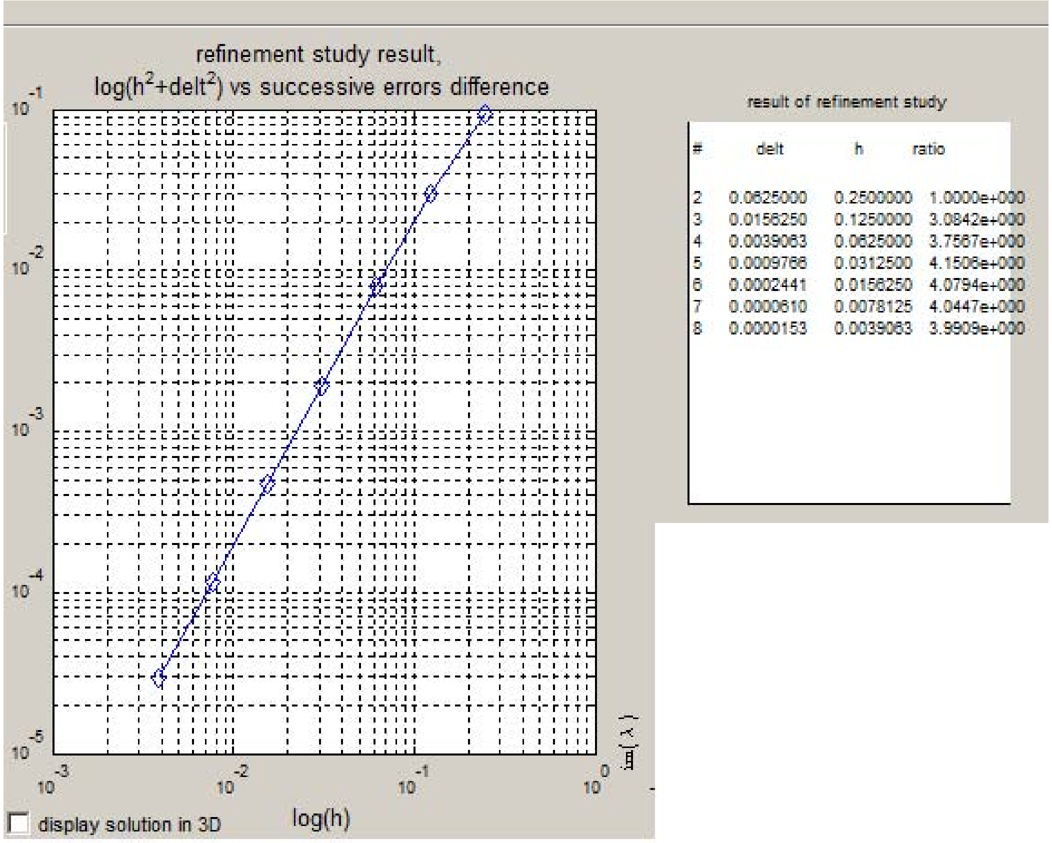
Condition (1) is weaker than (2), hence not considered. Condition (3) results in $\frac{kD}{h^2} \leq 1$ which is the same as (1). Condition (4) gives $\left|1 - \frac{k\,D}{h^2}\right| \leq 1$ or $\frac{k\,D}{h^2} \leq 2$ which is also weaker than (2). Condition (5) gives $1 \leq 1$ hence no information is obtained from it. Therefore, condition (2) remains, and that condition says that $\frac{kD}{h^2} \leq \frac{1}{2}$, which is the strongest condition. Hence, this is the absolute condition for stability for forward Euler. This agrees with the method to determine this using Von Neumann analysis.

Result for part(b) The forward Euler results are below. The space step was divided by 2 and the time step was divided by 4.

```
#       delt            h          ratio
2    0.0625000    0.2500000   1.0000e+000
```

3	0.0156250	0.1250000	3.0842e+000
4	0.0039063	0.0625000	3.7567e+000
5	0.0009766	0.0312500	4.1506e+000
6	0.0002441	0.0156250	4.0794e+000
7	0.0000610	0.0078125	4.0447e+000
8	0.0000153	0.0039063	3.9909e+000

The following is the corresponding loglog plot



result of refinement study

#	delt	h	ratio
2	0.0625000	0.2500000	1.0000e+000
3	0.0156250	0.1250000	3.0842e+000
4	0.0039063	0.0625000	3.7567e+000
5	0.0009766	0.0312500	4.1506e+000
6	0.0002441	0.0156250	4.0794e+000
7	0.0000610	0.0078125	4.0447e+000
8	0.0000153	0.0039063	3.9909e+000

Figure 6: corresponding loglog plot

Conclusion Since the ratio is 4 and since the time step was divided by 4 and the space step by 2, this implies forward Euler is first order in time and second order in space.

The appendix of this problem show the steady state analytical solution to the above PDE derived using Laplace transform method.

1.3 Problem 1 appendix

Review of refinement study process The idea behind refinement study is reviewed briefly. Assume the goal is to find the order of accuracy of a finite difference scheme with respect to the space step. The finite difference formula is first derived, and the exact solution is substituted into this formula. Terms that contain $u(x \pm h)$, are replaced by Taylor series approximation. The result is simplified, and the error term is found. An small example is given to illustrate the idea.

To find the order of accuracy in space using forward Euler finite difference approximation to a derivative

$$u'(x) = \frac{u(x+h) - u(x)}{h}$$

Each term for u in the RHS above is replaced by its exact value, using Taylor series expansion where needed, resulting in

$$\begin{aligned} u'(x) &= \frac{\left[u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{3!}u'''(x) + \cdots\right] - u(x)}{h} \\ &= u'(x) + \overbrace{\frac{h}{2}u''(x) + \frac{h^2}{3!}u'''(x) + \cdots}^{error} \end{aligned} \tag{1}$$

The error term is hence. It is the amount that the RHS differs from the LHS. The leading error term in (1) (the dominant term) is $\frac{h}{2}u''(x)$, but since x is a known value (the above is being evaluated at each grid point, hence x is known), then $u''(x)$ is some constant, and the leading error term in (1) is of the form Ch, where C is some constant. This is the same as saying that the error is of order h.

The above method can be used to find the order of the error in approximation when the exact solution is know. In problem (1), the exact solution is not given and was difficult to obtain. Hence, instead of finding the order of accuracy using the above method, it was found using a numerical experiment (refinement study).

In the refinement study the error itself is determined, and from the error profile (as h is changed), the order is determined. But this error is the error between successive numerical solutions.

Once the numerical error is found (after running the refinement study), then one method to find p (order of the error) is to take the logarithm resulting in

$$\begin{aligned} error &= Ch^p \\ \log(error) &= p\log(h) + \log C \\ &= p\log(h) + \text{constant} \end{aligned}$$

and this represents an equation of the line $Y = pX + k$, where p is the line slope which is the same as the order of accuracy. Hence, by generating different h values, and for each h determine the corresponding error, then p is found by measuring at the slope of line from the plot generated. If the slope is $p = 1$, then it is first order accuracy, and if the slope is $p = 2$, it is second order.

The above is a graphical method. Another method is as follows: Starting with some h value, the error e_{n-1} is found, then h is divided by half and the error, now called e_n is found again. The ratio $\frac{e_{n-1}}{e_n}$ is found. If p happened to be 2, then the ratio will come out to be 4. This is because $\frac{e_{n-1}}{e_n} = \frac{(h_{n-1})^p}{\left(\frac{h_{n-1}}{2}\right)^p} = 2^p$ and so if $p = 2$, then the ratio will be 4. If $p = 1$, then the ratio will be 2.

In the above description, errors are found using differences between successive solutions as follows

$$\begin{aligned} e_{n-1} &= |U_{n+1} - U_n| \\ e_n &= |U_n - U_{n-1}| \end{aligned}$$

The norm used to measure U, the approximate solution, is the Euclidean norm modified for the space grid

$$||U|| = \sqrt{h}\,||U||_2$$

Steady state analytical solution to the PDE The following shows the steps used to determine the steady state solution for

$$u_t = au_{xx} + f(t) \tag{1}$$

where $a = \frac{1}{100}$ and $f(t) = 1 - e^{-t}$ with initial conditions $u(0, t) = 0$ and boundary conditions $u(0, t) = 0$ and $u(1, t) = 0$.

The above is an inhomogeneous PDE (the source term $1 - e^{-t}$). The boundary conditions are homogeneous, and with zero initial conditions.

Since this is an inhomogeneous PDE, separation of variables can not be used. But the steady state solution (the particular solution) can be found using an integral transform approach. Integral transformation is first applied to the PDE, resulting in an ODE which is then solved in the new transformed space, and the solution in time domain is found by inverse transforming back.

Since the spatial domain in this problem is a bounded interval (from 0 to 1), Fourier transformation will not be used because the spatial domain is bounded and does not match the Fourier transformation domain (from $-\infty$ to $+\infty$), however, Laplace transformation (for $t > 0$) can be used as it matches the time domain of the problem.

Therefore, taking the Laplace transform of (1) w.r.t time gives

$$-u(x, 0) + sU(x, s) = a\frac{d^2U(x, s)}{dx^2} + \frac{1}{s} - \frac{1}{1 + s}$$

But $u(x, 0) = 0$, hence the resulting ODE is

$$a\frac{d^2U(x, s)}{dx^2} - sU(x, s) = \frac{1}{1 + s} - \frac{1}{s}$$

With the boundary conditions $U(0,s) = 0$ and $U(1,s) = 0$ obtained from the spatial domain. The above ODE is a second order, linear ODE, a inhomogeneous ODE that can be solved for $U(x,s)$, which results in the following (for the case $a = \frac{1}{100}$)

$$U(x,s) = \frac{-e^{-10x\sqrt{s}}\left(e^{10x\sqrt{s}-1} - 1\right)\left(e^{10x\sqrt{s}} - e^{10\sqrt{s}}\right)}{s^2(1+s)\left(1+e^{10\sqrt{s}}\right)}$$

The steady state solution can now be found using the limit theorem for Laplace transform, giving

$$\begin{aligned} u(x,\infty) &= \lim_{s->0} sU(x,s) \qquad (1) \\ &= 50x(1-x) \qquad (1) \end{aligned}$$

Here is a plot of the particular solution

```
x=0:0.01:1; plot(x,50*x.*(1-x))
```

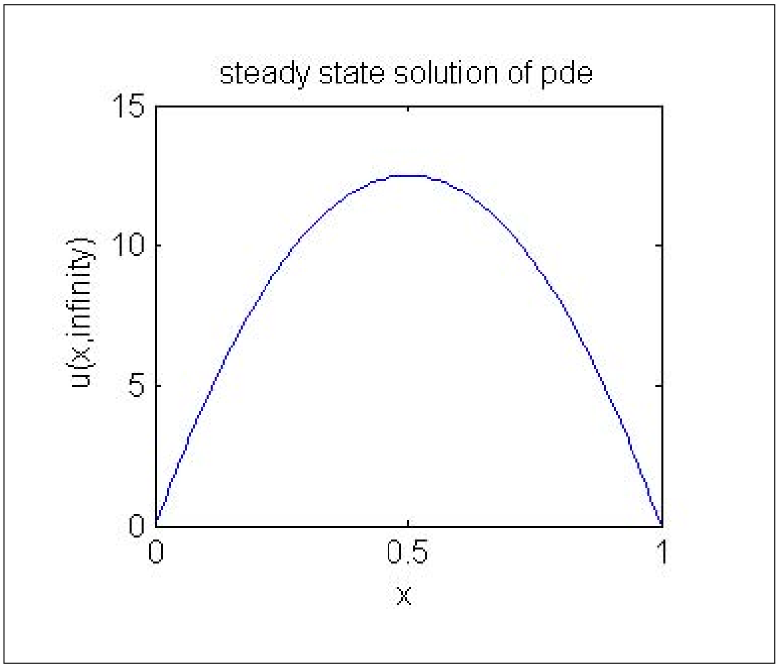


Figure 7: steady state plot. PDE solution

Derivation of forward Euler for periodic boundary conditions From part(b) above,

$$u_j^{n+1} = \frac{kD}{dh^2}u_{j-1}^n + u_j^n\left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2}u_{j+1}^n + \frac{k}{d}g_j^n$$

Periodic boundary conditions implies $u(0,t) = u(1,t)$, Hence u_{j-1} when j is the first note on the left is the same as node $N-1$. And u_{j+1} when j is the last node on the right is the same as node $j = 2$. As shown in the diagram below

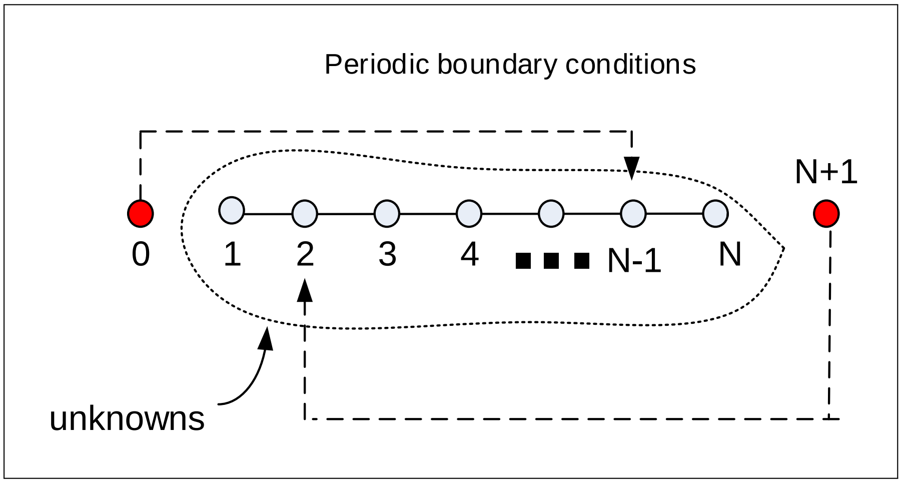


Figure 8: Grid format

Then

$$u_1^{n+1} = \frac{kD}{dh^2}u_{N-1}^n + u_1^n\left(1 - 2\frac{k}{d}\frac{D}{h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2}u_2^n$$

And

$$u_N^{n+1} = \frac{kD}{dh^2}u_{N-1}^n + u_N^n\left(1 - 2\frac{k}{d}\frac{D}{h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2}u_2^n$$

Let $r_1 = \left(1 - 2\frac{kD}{dh^2} - \frac{ak}{d}\right), r_2 = \frac{kD}{dh^2}, r_3 = \frac{k}{d}$, Hence the system can be written as

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \\ \vdots \\ u_{N-1}^{n+1} \\ u_N^{n+1} \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & 0 & 0 & 0 & r_2 & 0 \\ r_2 & r_1 & r_2 & 0 & 0 & 0 & 0 \\ 0 & r_2 & r_1 & r_2 & 0 & 0 & 0 \\ 0 & 0 & r_2 & r_1 & r_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & r_2 & r_1 & r_2 \\ 0 & r_2 & 0 & 0 & 0 & r_2 & r_1 \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ u_4^n \\ \vdots \\ u_{N-1}^n \\ u_N^n \end{bmatrix} + \begin{bmatrix} 0 \\ r_3 g_2^n \\ r_3 g_3^n \\ r_3 g_4^n \\ \vdots \\ r_3 g_{N-1}^n \\ 0 \end{bmatrix}$$

Derivation of forward Euler for Neumann boundary conditions both ends Using this numbering

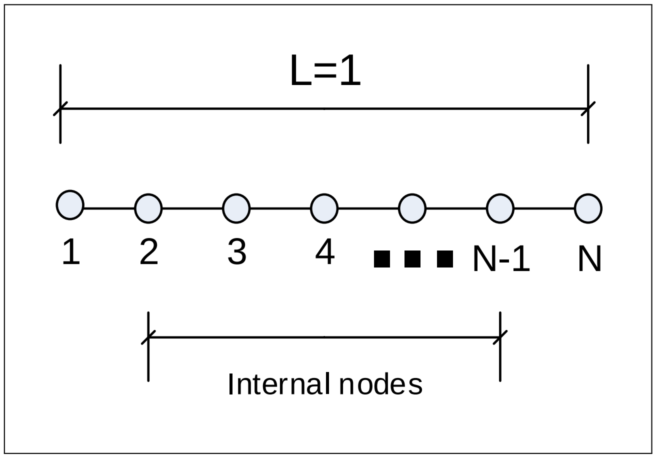


Figure 9: Grid format

Assume that $u_t = \alpha$ at node 1 and $u_t = \beta$ at node N (these are the Neumann boundary conditions).

Add a ghost node 0 to the left of node 1, and approximating $\alpha(t)$ gives

$$\alpha = \frac{u_0 - u_2}{2h}$$

hence

$$u_0 = 2h\alpha + u_2 \tag{1}$$

But the PDE for node 1 is

$$u_1^{n+1} = \frac{kD}{dh^2} u_0^n + u_1^n \left(1 - 2\frac{k\,D}{d\,h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2} u_2^n \tag{2}$$

Substitute (1) into (2) gives

$$\begin{aligned} u_1^{n+1} &= \frac{kD}{dh^2}(2h\alpha + u_2) + u_1^n \left(1 - 2\frac{k\,D}{d\,h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2} u_2^n \\ &= u_1^n \left(1 - 2\frac{k\,D}{d\,h^2} - \frac{ak}{d}\right) + u_2^n \left(2\frac{kD}{dh^2}\right) + 2h\alpha \frac{kD}{dh^2} \end{aligned}$$

Similarly for the right end. Add a ghost node $N+1$ to the right of node N, and approximating $\beta(t)$ gives

$$\beta = \frac{u_{N-1} - u_{N+1}}{2h}$$

hence

$$u_{N+1} = 2h\beta + u_{N-1} \tag{3}$$

But the PDE for node N is

$$u_N^{n+1} = \frac{kD}{dh^2}u_{N-1}^n + u_N^n\left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2}u_{N+1}^n \tag{4}$$

Substitute (3) into (4) gives

$$\begin{aligned} u_1^{n+1} &= \frac{kD}{dh^2}u_{N-1}^n + u_N^n\left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2}\left(2h\beta + u_{N-1}\right) \\ &= 2\frac{kD}{dh^2}u_{N-1}^n + u_N^n\left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + 2h\beta\frac{kD}{dh^2} \end{aligned}$$

Nodes $j = 2 \cdots N-1$ remain the same as before. In other words

$$u_j^{n+1} = \frac{kD}{dh^2}u_{j-1}^n + u_j^n\left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2}u_{j+1}^n + \frac{k}{d}g_j^n$$

Let $r_1 = \left(1 - 2\frac{kD}{dh^2} - \frac{ak}{d}\right), r_2 = \frac{kD}{dh^2},, r_3 = \frac{k}{d}$, Hence the system becomes (now nodes 1 and N are unknowns and added)

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \\ \vdots \\ u_{N-1}^{n+1} \\ u_N^{n+1} \end{bmatrix} = \begin{bmatrix} r_1 & 2r_2 & 0 & 0 & 0 & 0 & 0 \\ r_2 & r_1 & r_2 & 0 & 0 & 0 & 0 \\ 0 & r2 & r_1 & r_2 & 0 & 0 & 0 \\ 0 & 0 & r_2 & \ddots & r_2 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & r_2 & r_1 & r_2 \\ 0 & 0 & 0 & 0 & 0 & 2r_2 & r_1 \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ u_4^n \\ \vdots \\ u_{N-1}^n \\ u_N^n \end{bmatrix} + \begin{bmatrix} 2h\alpha^n\ r_2 \\ r_3 g_2^n \\ r_3 g_3^n \\ r_3 g_4^n \\ \vdots \\ r_3 g_{N-1}^n \\ 2h\beta^n\ r_2 \end{bmatrix}$$

Derivation of forward Euler for Neumann on left and Dirichlet on right Using this numbering

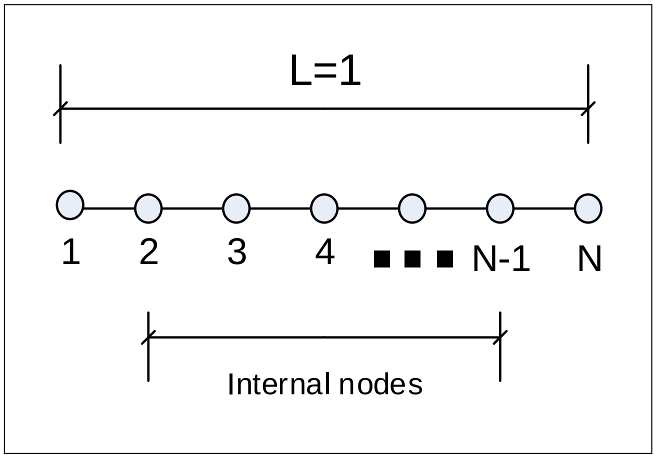


Figure 10: Grid format

Assume that $u_N = \beta$ at node N and $u_t = \alpha$ at node 1 (these are the Dirichlet and Neumann boundary conditions).

The unknowns in this case are nodes $1 \cdots N-1$, since u_N is known from Dirichlet boundary conditions. Add a ghost node 0 to the left of node 1, and approximating α gives

$$\alpha = \frac{u_0 - u_2}{2h}$$

hence

$$u_0 = 2h\alpha + u_2 \tag{3}$$

But the PDE for node 1 is

$$u_1^{n+1} = \frac{kD}{dh^2} u_0^n + u_1^n \left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2} u_2^n \tag{4}$$

Substitute (3) into (4) gives

$$\begin{aligned} u_1^{n+1} &= \frac{kD}{dh^2}(2h\alpha + u_2) + u_1^n \left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2} u_2^n \\ &= u_1^n \left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + \left(2\frac{kD}{dh^2}\right) u_2^n + 2\ h\ \alpha \frac{kD}{dh^2} \end{aligned}$$

Nodes $j = 2 \cdots N-1$ remain the same as before. In other words

$$u_j^{n+1} = \frac{kD}{dh^2}u_{j-1}^n + u_j^n\left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2}u_{j+1}^n + \frac{k}{d}g_j^n$$

Let $r_1 = \left(1 - 2\frac{kD}{dh^2} - \frac{ak}{d}\right), r_2 = \frac{kD}{dh^2}, r_3 = \frac{k}{d}$, Hence the system becomes

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ u_{N-1}^{n+1} \\ u_N^{n+1} \end{bmatrix} = \begin{bmatrix} r_1 & 2r_2 & 0 & 0 & 0 & 0 & 0 \\ r_2 & r_1 & r_2 & 0 & 0 & 0 & 0 \\ 0 & r_2 & r_1 & r_2 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & r_2 & r_1 & r_2 & 0 \\ 0 & 0 & 0 & 0 & r_2 & r_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ u_{N-1}^n \\ u_N^n \end{bmatrix} + \begin{bmatrix} 2\ h\ \alpha^n r_2 \\ r_3 g_2^n \\ r_3 g_3^n \\ r_3 g_4^n \\ \vdots \\ r_3 g_{N-1}^n + r_2 \beta^n \\ \beta^{n+1} \end{bmatrix}$$

Derivation of forward Euler for Neumann on right and Dirichlet on left Using this numbering

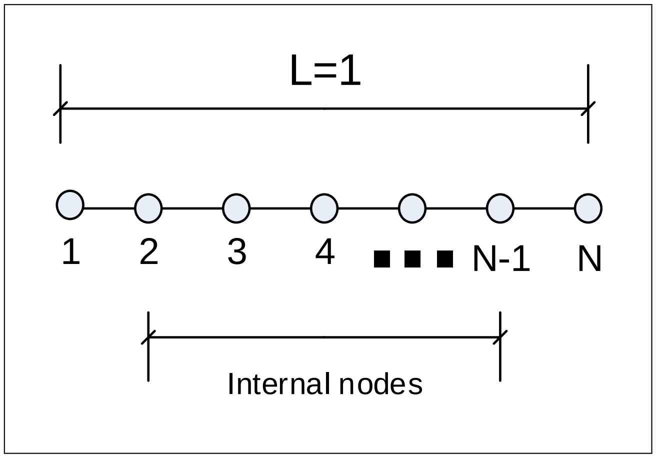


Figure 11: Grid format

Assume that $u_0 = \alpha$ at node 1 and $u_t = \beta$ at node N (these are the Dirichlet and Neumann boundary conditions).

The unknowns in this case are nodes $2 \cdots N$, since u_1 is known from Dirichlet boundary conditions. Add a ghost node $N+1$ to the right of node N, and approximating $\beta(t)$ gives

$$\beta = \frac{u_{N-1} - u_{N+1}}{2h}$$

hence

$$u_{N+1} = 2h\beta + u_{N-1} \tag{3}$$

But the PDE for node N is

$$u_N^{n+1} = \frac{kD}{dh^2} u_{N-1}^n + u_N^n \left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2} u_{N+1}^n \tag{4}$$

Substitute (3) into (4) gives

$$\begin{aligned} u_N^{n+1} &= \frac{kD}{dh^2} u_{N-1}^n + u_N^n \left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2}\left(2h\beta + u_{N-1}\right) \\ &= 2\frac{kD}{dh^2} u_{N-1}^n + u_N^n \left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + 2h\beta\frac{kD}{dh^2} \end{aligned}$$

Nodes $j = 2 \cdots N-1$ remain the same as before. In other words

$$u_j^{n+1} = \frac{kD}{dh^2} u_{j-1}^n + u_j^n \left(1 - 2\frac{k\ D}{d\ h^2} - \frac{ak}{d}\right) + \frac{kD}{dh^2} u_{j+1}^n + \frac{k}{d} g_j^n$$

Let $r_1 = \left(1 - 2\frac{kD}{dh^2} - \frac{ak}{d}\right), r_2 = \frac{kD}{dh^2},, r_3 = \frac{k}{d}$, Hence the system becomes

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \\ \vdots \\ u_{N-1}^{n+1} \\ u_N^{n+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_1 & r_2 & 0 & 0 & 0 & 0 \\ 0 & r_2 & r_1 & r_2 & 0 & 0 & 0 \\ 0 & 0 & r_2 & r_1 & r_2 & 0 & 0 \\ 0 & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & r_2 & r_1 & r_2 \\ 0 & 0 & 0 & 0 & 0 & 2r_2 & r_1 \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ u_4^n \\ \vdots \\ u_{N-1}^n \\ u_N^n \end{bmatrix} + \begin{bmatrix} \alpha^{n+1} \\ r_2\alpha^n + r_3 g_2^n \\ r_3 g_2^n \\ r_3 g_3^n \\ r_3 g_4^n \\ \vdots \\ r_3 g_{N-1}^n \\ 2h\beta^n\ r_2 \end{bmatrix}$$

Derivation of C-N for periodic boundary conditions From part(a) above,

$$u_j^{n+1}\left(1 + \frac{kD}{dh^2} + \frac{a\Delta t}{2d}\right) - \frac{kD}{2dh^2} u_{j-1}^{n+1} - \frac{kD}{2dh^2} u_{j+1}^{n+1} = u_j^n \left(1 - \frac{kD}{dh^2} - \frac{ak}{2d}\right) + u_{j-1}^n \frac{kD}{2dh^2} + u_{j+1}^n \frac{kD}{2dh^2} + \frac{k}{2d}\left(g_j^n + g_j^{n+1}\right)$$

Periodic boundary conditions implies $u(0,t) = u(1,t)$, Hence there is an extra one unknown (in addition to the internal nodes). Either $u(0,t)$ or $u(1,t)$ can be selected since they have the same value. When selecting the right end node, then u_{N+1} becomes an unknown to be added to the internal nodes. Using the following diagram

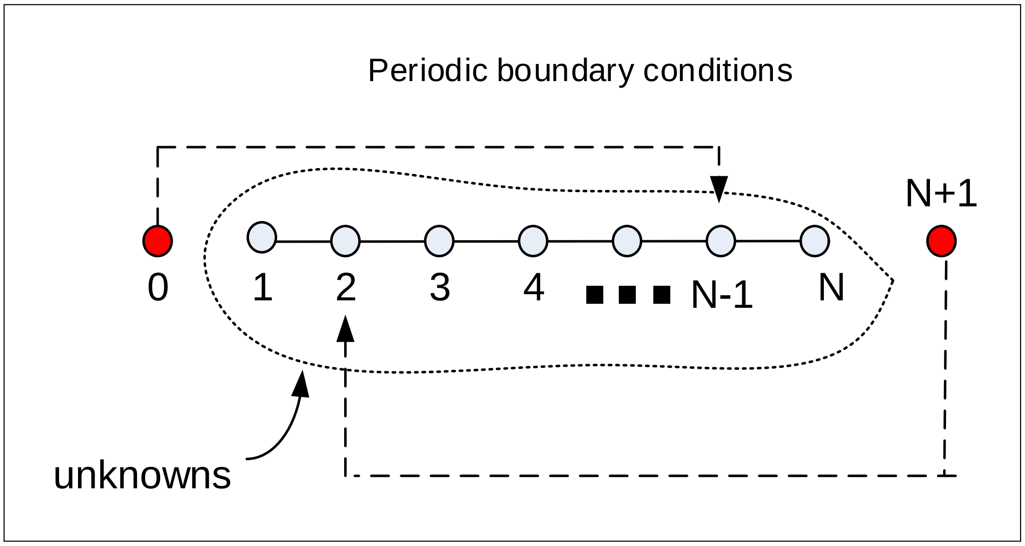


Figure 12: Grid format

Then for node 2

$$u_2^{n+1}\left(1+\frac{kD}{dh^2}+\frac{a\Delta t}{2d}\right)-\frac{kD}{2dh^2}u_1^{n+1}-\frac{kD}{2dh^2}u_3^{n+1}=u_2^n\left(1-\frac{kD}{dh^2}-\frac{ak}{2d}\right)+u_1^n\frac{kD}{2dh^2}+u_3^n\frac{kD}{2dh^2}+\frac{k}{2d}\left(g_2^n+g_2^{n+1}\right)$$

$$u_2^{n+1}\left(1+\frac{kD}{dh^2}+\frac{a\Delta t}{2d}\right)-\frac{kD}{2dh^2}u_N^{n+1}-\frac{kD}{2dh^2}u_3^{n+1}=u_2^n\left(1-\frac{kD}{dh^2}-\frac{ak}{2d}\right)+u_N^n\frac{kD}{2dh^2}+u_3^n\frac{kD}{2dh^2}+\frac{k}{2d}\left(g_2^n+g_2^{n+1}\right)$$

And for node N

$$u_N^{n+1}\left(1+\frac{kD}{dh^2}+\frac{a\Delta t}{2d}\right)-\frac{kD}{2dh^2}u_{N-1}^{n+1}-\frac{kD}{2dh^2}u_{N+1}^{n+1}=u_N^n\left(1-\frac{kD}{dh^2}-\frac{ak}{2d}\right)+u_{N-1}^n\frac{kD}{2dh^2}+u_{N+1}^n\frac{kD}{2dh^2}+\frac{k}{2d}\left(g_N^n+g_N^{n+1}\right)$$

$$u_N^{n+1}\left(1+\frac{kD}{dh^2}+\frac{a\Delta t}{2d}\right)-\frac{kD}{2dh^2}u_{N-1}^{n+1}-\frac{kD}{2dh^2}u_2^{n+1}=u_N^n\left(1-\frac{kD}{dh^2}-\frac{ak}{2d}\right)+u_{N-1}^n\frac{kD}{2dh^2}+u_2^n\frac{kD}{2dh^2}+\frac{k}{2d}\left(g_N^n+g_N^{n+1}\right)$$

Let

$$r_1=\left(1+\frac{kD}{dh^2}+\frac{ak}{2d}\right)$$

$$r_2=\frac{kD}{2dh^2}$$

$$r_3=\left(1-\frac{kD}{dh^2}-\frac{ak}{2d}\right)$$

$$r_4=\frac{k}{2d}$$

$$g^n+g^{n+1}\equiv\tilde{g}$$

Converting to matrix form gives

$$\overbrace{\begin{bmatrix} r_1 & -r_2 & 0 & 0 & -r_2 \\ -r_2 & r_1 & -r_2 & 0 & 0 \\ 0 & -r_2 & r_1 & 0 & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ -r_2 & 0 & 0 & -r_2 & r_1 \end{bmatrix}}^{\mathbf{A}} \overbrace{\begin{bmatrix} u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ u_{N-1}^{n+1} \\ u_N^{n+1} \end{bmatrix}}^{\mathbf{x}} = \overbrace{\begin{bmatrix} r_3 & r_2 & 0 & 0 & r_2 \\ r_2 & r_3 & r_2 & 0 & 0 \\ 0 & r_2 & r_3 & r_2 & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ r_2 & 0 & 0 & r_2 & r_3 \end{bmatrix} \begin{bmatrix} u_2^n \\ u_3^n \\ u_4^n \\ \vdots \\ u_N^n \end{bmatrix} + \begin{bmatrix} r_4\tilde{g}_2 \\ r_4\tilde{g}_3 \\ r_4\tilde{g}_4 \\ \vdots \\ r_4\tilde{g}_N \end{bmatrix}}^{\mathbf{b}} \tag{3}$$

$$u_1^{n+1} = u_N^{n+1}$$

Derivation of C-N for Neumann boundary conditions both ends Using this numbering

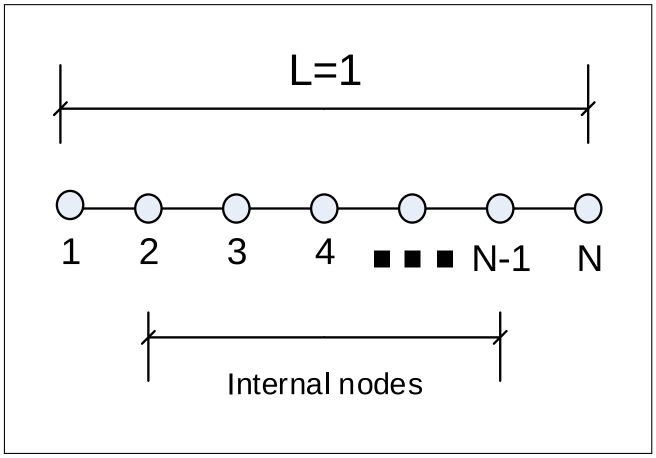


Figure 13: Grid format

Assuming that $u_t = \alpha$ at node 1 and $u_t = \beta$ at node N (these are the Neumann boundary conditions). Add a ghost node 0 to the left of node 1, and approximating α gives

$$\alpha = \frac{u_0 - u_2}{2h}$$

hence

$$u_0 = 2h\alpha + u_2 \tag{1}$$

But the PDE for node 1 is

$$r_1 u_1^{n+1} - r_2 u_0^{n+1} - r_2 u_2^{n+1} = r_3 u_1^n + r_2 u_0^n + r_2 u_2^n \tag{2}$$

substitute (1) into (2)

$$\begin{aligned} r_1 u_1^{n+1} - r_2\left(2h\alpha^{n+1} + u_2^{n+1}\right) - r_2 u_2^{n+1} &= r_3 u_1^n + r_2\left(2h\alpha^n + u_2^n\right) + r_2 u_2^n \\ r_1 u_1^{n+1} - 2r_2 u_2^{n+1} &= r_3 u_1^n + 2r_2 u_2^n + 2r_2 h\alpha^n + 2r_2 h\alpha^{n+1} \end{aligned}$$

Similarly for the right end. Add a ghost node $N+1$ to the right of node N, and approximating $\beta(t)$ gives

$$\beta = \frac{u_{N-1} - u_{N+1}}{2h}$$

hence

$$u_{N+1} = 2h\beta + u_{N-1} \tag{3}$$

But the PDE for node N is

$$r_1 u_N^{n+1} - r_2 u_{N-1}^{n+1} - r_2 u_{N+1}^{n+1} = r_3 u_N^n + r_2 u_{N-1}^n + r_2 u_{N+1}^n \tag{4}$$

substitute (3) into (4)

$$\begin{aligned} r_1 u_N^{n+1} - r_2 u_{N-1}^{n+1} - r_2\left(2h\beta^{n+1} + u_{N-1}^{n+1}\right) &= r_3 u_N^n + r_2 u_{N-1}^n + r_2\left(2h\beta + u_{N-1}^n\right) \\ r_1 u_N^{n+1} - 2r_2 u_{N-1}^{n+1} &= r_3 u_N^n + 2r_2 u_{N-1}^n + 2r_2 h\beta + 2r_2 h\beta^{n+1} \end{aligned}$$

Nodes $j = 2 \cdots N-1$ remain the same as before. In other words

$$u_j^{n+1}\left(1 + \frac{kD}{dh^2} + \frac{ak}{2d}\right) - \frac{kD}{2dh^2} u_{j-1}^{n+1} - \frac{kD}{2dh^2} u_{j+1}^{n+1} = u_j^n\left(1 - \frac{kD}{dh^2} - \frac{ak}{2d}\right) + u_{j-1}^n \frac{kD}{2dh^2} + u_{j+1}^n \frac{kD}{2dh^2} + \frac{k}{2d}\left(g_j^n + g_j^{n+1}\right)$$

Let

$$\begin{aligned} r_1 &= \left(1 + \frac{kD}{dh^2} + \frac{ak}{2d}\right) \\ r_2 &= \frac{kD}{2dh^2} \\ r_3 &= \left(1 - \frac{kD}{dh^2} - \frac{ak}{2d}\right) \\ r_4 &= \frac{k}{2d} \end{aligned}$$

gives

$$r_1 u_j^{n+1} - r_2 u_{j-1}^{n+1} - r_2 u_{j+1}^{n+1} = r_3 u_j^n + r_2 u_{j-1}^n + r_2 u_{j+1}^n + r_4\left(g_j^n + g_j^{n+1}\right)$$

Then the above becomes

$$\overbrace{\begin{bmatrix} r_1 & -2r_2 & 0 & 0 & 0 & 0 \\ -r_2 & r_1 & -r_2 & 0 & 0 & 0 \\ 0 & -r_2 & r_1 & -r_2 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & -r_2 & r_1 & -r_2 \\ 0 & 0 & 0 & 0 & -2r_2 & r_1 \end{bmatrix}}^{\mathbf{A}} \overbrace{\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ u_{N-1}^{n+1} \\ u_N^{n+1} \end{bmatrix}}^{\mathbf{x}} = \overbrace{\begin{bmatrix} r_3 & 2r_2 & 0 & 0 & 0 & 0 \\ r_2 & r_3 & r_2 & 0 & 0 & 0 \\ 0 & r_2 & r_3 & r_2 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & r_2 & r_3 & r_2 \\ 0 & 0 & 0 & 0 & 2r_2 & r_3 \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ u_{N-1}^n \\ u_N^n \end{bmatrix} + \begin{bmatrix} 2r_2 h\alpha^n + 2r_2 h\alpha^{n+1} \\ r_4\left(g_2^n + g_2^{n+1}\right) \\ r_4\left(g_3^n + g_3^{n+1}\right) \\ \vdots \\ r_4\left(g_{N-1}^n + g_{N-1}^{n+1}\right) \\ 2r_2 h\beta + 2r_2 h\beta^{n+1} \end{bmatrix}}^{\mathbf{b}}$$

Derivation of C-N for Neumann on left and Dirichlet on right Using this numbering

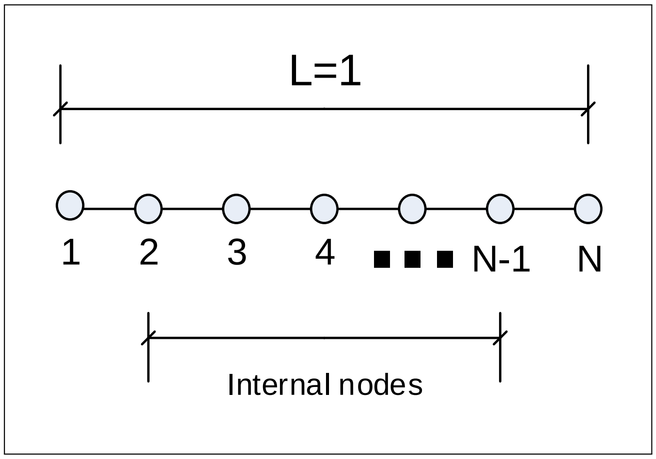


Figure 14: Grid format

Assume that $u_t = \alpha$ at node 1 and $u_N = \beta$ at node N (these are the Neumann and Dirichlet boundary conditions). Add a ghost node 0 to the left of node 1, and approximating α gives

$$\alpha = \frac{u_0 - u_2}{2h}$$

hence

$$u_0 = 2h\alpha + u_2 \tag{1}$$

But the PDE for node 1 is

$$r_1 u_1^{n+1} - r_2 u_0^{n+1} - r_2 u_2^{n+1} = r_3 u_1^n + r_2 u_0^n + r_2 u_2^n \tag{2}$$

substitute (1) into (2)

$$\begin{aligned} r_1 u_1^{n+1} - r_2\left(2h\alpha^{n+1} + u_2^{n+1}\right) - r_2 u_2^{n+1} &= r_3 u_1^n + r_2\left(2h\alpha^n + u_2^n\right) + r_2 u_2^n \\ r_1 u_1^{n+1} - 2r_2 u_2^{n+1} &= r_3 u_1^n + 2r_2 u_2^n + 2r_2 h\alpha^n + 2r_2 h\alpha^{n+1} \end{aligned}$$

Nodes $j = 2 \cdots N-1$ remain the same as before. In other words

$$u_j^{n+1}\left(1 + \frac{kD}{dh^2} + \frac{ak}{2d}\right) - \frac{kD}{2dh^2}u_{j-1}^{n+1} - \frac{kD}{2dh^2}u_{j+1}^{n+1} = u_j^n\left(1 - \frac{kD}{dh^2} - \frac{ak}{2d}\right) + u_{j-1}^n\frac{kD}{2dh^2} + u_{j+1}^n\frac{kD}{2dh^2} + \frac{k}{2d}\left(g_j^n + g_j^{n+1}\right)$$

Let

$$\begin{aligned} r_1 &= \left(1 + \frac{kD}{dh^2} + \frac{ak}{2d}\right) \\ r_2 &= \frac{kD}{2dh^2} \\ r_3 &= \left(1 - \frac{kD}{dh^2} - \frac{ak}{2d}\right) \\ r_4 &= \frac{k}{2d} \end{aligned}$$

gives

$$r_1 u_j^{n+1} - r_2 u_{j-1}^{n+1} - r_2 u_{j+1}^{n+1} = r_3 u_j^n + r_2 u_{j-1}^n + r_2 u_{j+1}^n + r_4\left(g_j^n + g_j^{n+1}\right)$$

Then the system becomes

$$\overbrace{\begin{bmatrix} r_1 & -2r_2 & 0 & 0 & 0 & 0 & 0 \\ -r_2 & r_1 & -r_2 & 0 & 0 & 0 & 0 \\ 0 & -r_2 & r_1 & -r_2 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & 0 \\ 0 & 0 & 0 & -r_2 & r_1 & -r_2 & 0 \\ 0 & 0 & 0 & 0 & -r_2 & r_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}^{\mathbf{A}} \overbrace{\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ u_{N-2}^{n+1} \\ u_{N-1}^{n+1} \\ u_N^{n+1} \end{bmatrix}}^{\mathbf{x}} = \overbrace{\begin{bmatrix} r_3 & 2r_2 & 0 & 0 & 0 & 0 & 0 \\ r_2 & r_3 & r_2 & 0 & 0 & 0 & 0 \\ 0 & r_2 & r_3 & r_2 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & 0 \\ 0 & 0 & 0 & r_2 & r_3 & r_2 & 0 \\ 0 & 0 & 0 & 0 & 2r_2 & r_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ u_{N-2}^n \\ u_{N-1}^n \\ u_N^n \end{bmatrix} + \begin{bmatrix} 2r_2 h\alpha^n + 2r_2 h\alpha^{n+1} \\ r_4\left(g_2^n + g_2^{n+1}\right) \\ r_4\left(g_3^n + g_3^{n+1}\right) \\ \vdots \\ r_4\left(g_{N-2}^n + g_{N-2}^{n+1}\right) \\ r_4\left(g_{N-1}^n + g_{N-1}^{n+1}\right) + r_2\beta^n \\ \beta^{n+1} \end{bmatrix}}^{\mathbf{b}}$$

Derivation of C-N for Neumann on right and Dirichlet on left Using this numbering

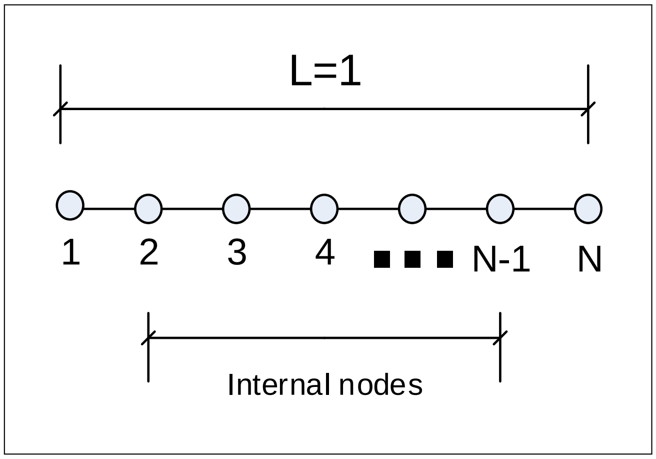


Figure 15: Grid format

Assume that $u_1 = \alpha$ at node 1 and $u_t = \beta$ at node N (these are the Dirichlet and Neumann boundary conditions). Add a ghost node $N+1$ to the right of node N, and approximating β gives

$$\beta = \frac{u_{N-1} - u_{N+1}}{2h}$$

hence

$$u_{N+1} = 2h\beta + u_{N-1} \tag{1}$$

But the PDE for node N is

$$r_1 u_N^{n+1} - r_2 u_{N-1}^{n+1} - r_2 u_{N+1}^{n+1} = r_3 u_N^n + r_2 u_{N-1}^n + r_2 u_{N+1}^n \tag{2}$$

substitute (1) into (2)

$$\begin{aligned} r_1 u_N^{n+1} - r_2 u_{N-1}^{n+1} - r_2\left(2h\beta^{n+1} + u_{N-1}^{n+1}\right) &= r_3 u_N^n + r_2 u_{N-1}^n + r_2\left(2h\beta^n + u_{N-1}^n\right) \\ r_1 u_N^{n+1} - 2r_2 u_{N-1}^{n+1} &= r_3 u_N^n + 2r_2 u_{N-1}^n + 2r_2 h\beta^n + 2r_2 h\beta^{n+1} \end{aligned}$$

Nodes $j = 2 \cdots N-1$ remain the same as before. In other words

$$u_j^{n+1}\left(1 + \frac{kD}{dh^2} + \frac{ak}{2d}\right) - \frac{kD}{2dh^2} u_{j-1}^{n+1} - \frac{kD}{2dh^2} u_{j+1}^{n+1} = u_j^n\left(1 - \frac{kD}{dh^2} - \frac{ak}{2d}\right) + u_{j-1}^n \frac{kD}{2dh^2} + u_{j+1}^n \frac{kD}{2dh^2} + \frac{k}{2d}\left(g_j^n + g_j^{n+1}\right)$$

Let

$$\begin{aligned}
r_1 &= \left(1 + \frac{kD}{dh^2} + \frac{ak}{2d}\right) \\
r_2 &= \frac{kD}{2dh^2} \\
r_3 &= \left(1 - \frac{kD}{dh^2} - \frac{ak}{2d}\right) \\
r_4 &= \frac{k}{2d}
\end{aligned}$$

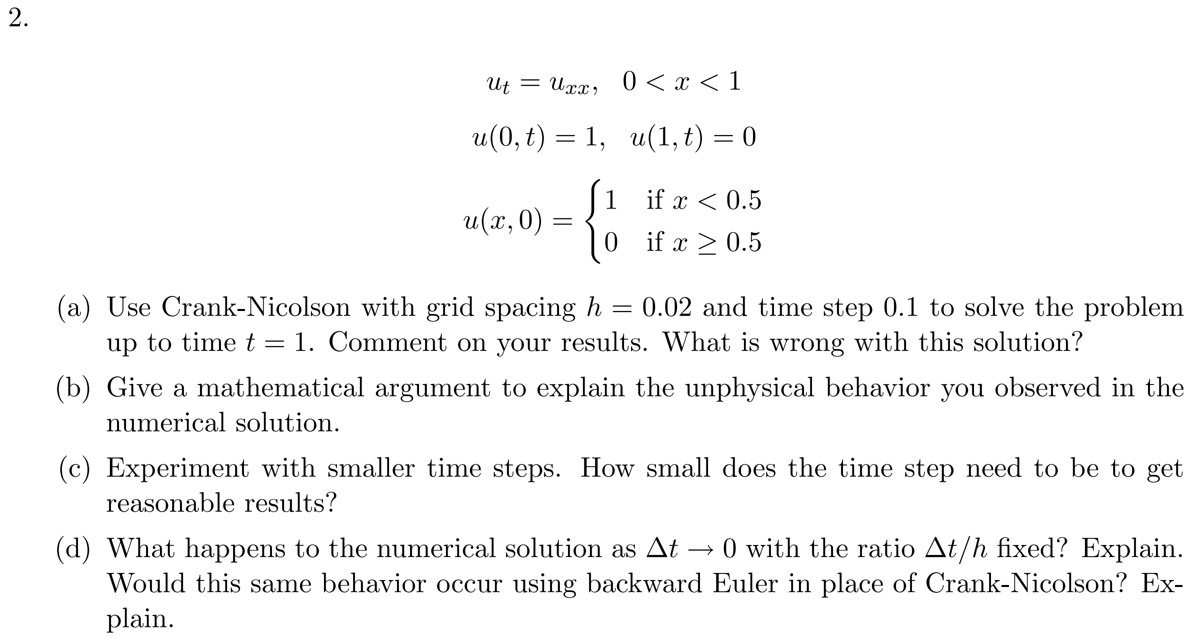
gives

$$r_1 u_j^{n+1} - r_2 u_{j-1}^{n+1} - r_2 u_{j+1}^{n+1} = r_3 u_j^n + r_2 u_{j-1}^n + r_2 u_{j+1}^n + r_4\left(g_j^n + g_j^{n+1}\right)$$

Then the system becomes

$$\overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_1 & -r_2 & 0 & 0 & 0 & 0 \\ 0 & -r_2 & r_1 & -r_2 & 0 & 0 & 0 \\ 0 & 0 & -r_2 & r_1 & -r_2 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & 0 & -r_2 & r_1 & -r_2 \\ 0 & 0 & 0 & 0 & 0 & -2r_2 & r_1 \end{bmatrix}}^{\mathbf{A}} \overbrace{\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \\ \vdots \\ u_{N-1}^{n+1} \\ u_N^{n+1} \end{bmatrix}}^{\mathbf{x}} = \overbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r_3 & r_2 & 0 & 0 & 0 & 0 \\ 0 & r_2 & r_3 & r_2 & 0 & 0 & 0 \\ 0 & 0 & r_2 & r_3 & r_2 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & 0 & r_2 & r_3 & r_2 \\ 0 & 0 & 0 & 0 & 0 & 2r_2 & r_3 \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ \vdots \\ u_{N-2}^n \\ u_{N-1}^n \end{bmatrix} + \begin{bmatrix} \alpha^{n+1} \\ r_4\left(g_2^n + g_2^{n+1}\right) + r_2\alpha^n \\ r_4\left(g_3^n + g_3^{n+1}\right) \\ r_4\left(g_4^n + g_4^{n+1}\right) \\ \vdots \\ r_4\left(g_{N-2}^n + g_{N-2}^{n+1}\right) \\ 2r_2h\beta^n + 2r_2h\beta^{n+1} \end{bmatrix}}^{\mathbf{b}}$$

2 Problem 2



2.

$$u_t = u_{xx}, \quad 0 < x < 1$$

$$u(0,t) = 1, \quad u(1,t) = 0$$

$$u(x,0) = \begin{cases} 1 & \text{if } x < 0.5 \\ 0 & \text{if } x \geq 0.5 \end{cases}$$

(a) Use Crank-Nicolson with grid spacing $h = 0.02$ and time step 0.1 to solve the problem up to time $t = 1$. Comment on your results. What is wrong with this solution?

(b) Give a mathematical argument to explain the unphysical behavior you observed in the numerical solution.

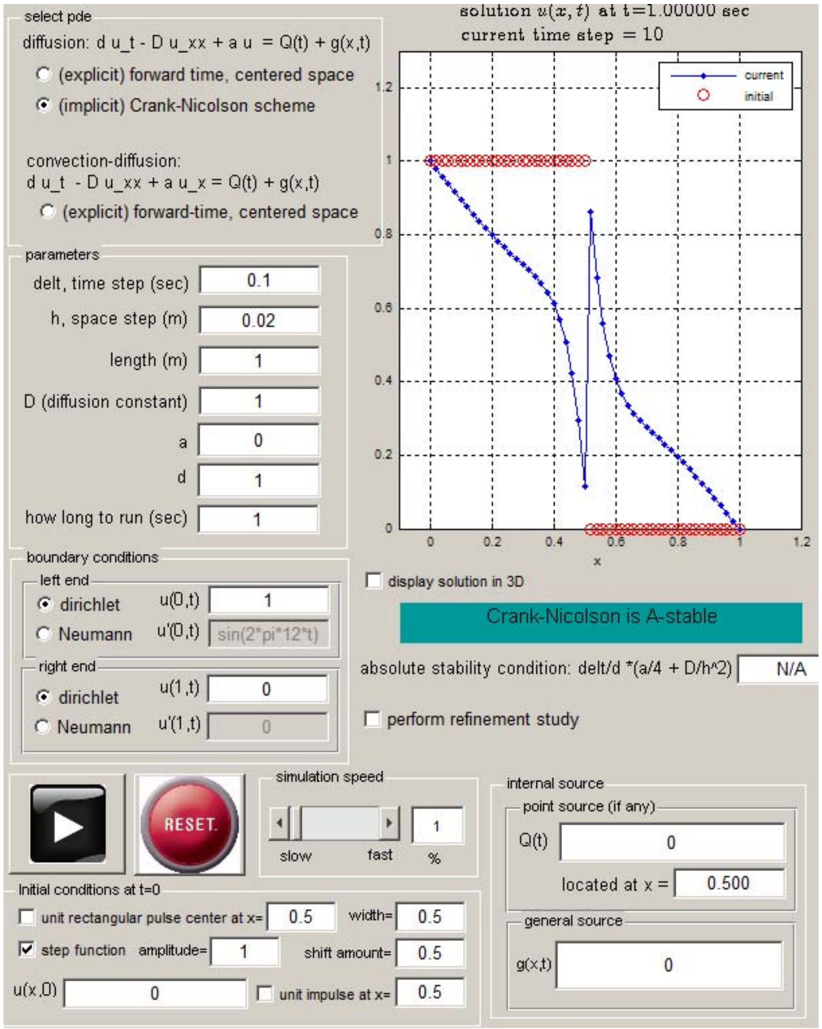
(c) Experiment with smaller time steps. How small does the time step need to be to get reasonable results?

(d) What happens to the numerical solution as $\Delta t \to 0$ with the ratio $\Delta t/h$ fixed? Explain. Would this same behavior occur using backward Euler in place of Crank-Nicolson? Explain.

Figure 16: Problem statement

2.1 Part(a)

The C-N scheme was programmed in Matlab and then run on the above problem. The following shows the result



select pde

diffusion: d u_t - D u_xx + a u = Q(t) + g(x,t)

○ (explicit) forward time, centered space

● (implicit) Crank-Nicolson scheme

convection-diffusion:
d u_t - D u_xx + a u_x = Q(t) + g(x,t)

○ (explicit) forward-time, centered space

parameters

delt, time step (sec) 0.1

h, space step (m) 0.02

length (m) 1

D (diffusion constant) 1

a 0

d 1

how long to run (sec) 1

boundary conditions

left end

● dirichlet u(0,t) 1

○ Neumann u'(0,t) sin(2*pi*12*t)

right end

● dirichlet u(1,t) 0

○ Neumann u'(1,t) 0

□ display solution in 3D

Crank-Nicolson is A-stable

absolute stability condition: delt/d *(a/4 + D/h^2) N/A

□ perform refinement study

simulation speed 1 %

internal source

point source (if any)

Q(t) 0

located at x = 0.500

general source

g(x,t) 0

Initial conditions at t=0

□ unit rectangular pulse center at x= 0.5 width= 0.5

☑ step function amplitude= 1 shift amount= 0.5

u(x,0) 0 □ unit impulse at x= 0.5

Figure 17: C-N scheme solution result

In the above plot, the red line represents initial conditions (the step function shifted to the right by 0.5) and the blue line represents the final numerical solution at time $t = 1$ second. The following plot is a closer look at the grid near $x = \frac{1}{2}$ showing the initial conditions

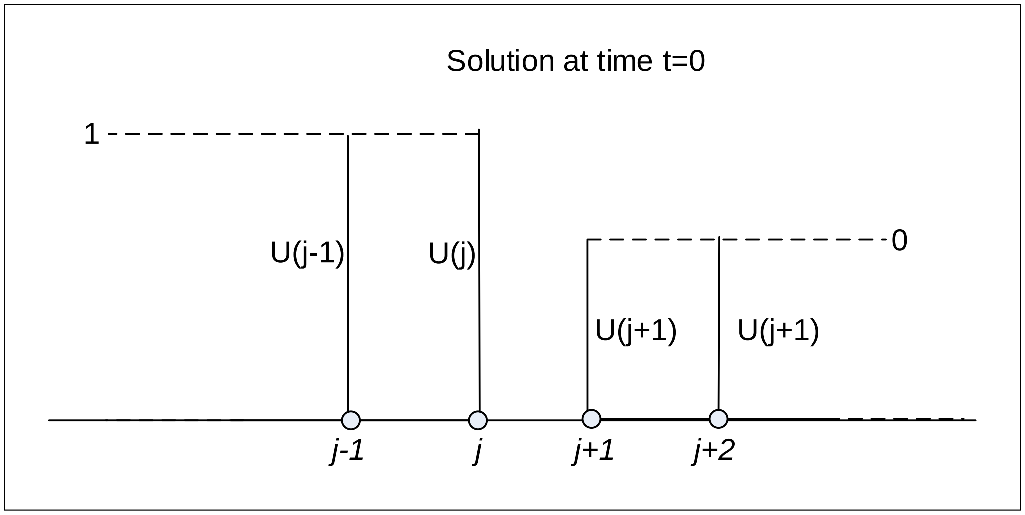


Figure 18: the grid near $x = \frac{1}{2}$ showing the initial conditions

It is clear the numerical solution is not accurate as it does not match what is expected to occur physically which is for initial data to diffuse. The initial data contained high spatial frequency that should have been smoothed out rapidly. The final numerical solution is not smooth and contain high spatial frequency components which should have been attenuated by the time the run is completed.

The exact solution in the Fourier space is $\hat{u}(\xi,t) = \hat{u}(\xi,0)e^{-D\xi^2 t}$, where $\hat{u}(\xi,0)$ is the spectrum or Fourier coefficients of the initial condition $u(x,0)$. This shows that modes with large spatial frequency (large wave number ξ) will attenuate the fastest due to the negative exponential decay effect. But this was not observed in the above numerical solution.

C-N is a stable scheme (A-stable), but can be inaccurate if the time step used is large relative to the space step or if initial conditions contain large spatial frequency components. In C-N, the time step needs to be about the same order of value as the space step for the scheme to give accurate numerical results. (This is because in C-N, the order of accuracy of space and time are the same, as was found in problem 1).

Therefore, it appears that C-N scheme does not handle discontinuities in initial conditions well as this result shows.

In the next part, the amplification factor for C-N is determined, and a mathematical explanation for the above result is given.

2.2 Part (b)

The C-N scheme for $u_t = Du_{xx}$ is given by

$$-ru_{j-1}^{n+1} + u_j^{n+1}(1+2r) - ru_{j+1}^{n+1} = ru_{j-1}^{n} + u_j^{n}(1-2r) + ru_{j+1}^{n} \tag{1}$$

Where $r = \frac{\Delta tD}{2h^2}$. Von Neumann analysis is used to determine the magnification factor[2]. Assume $u_j^n = e^{i\xi x_j}$, and $u_j^{n+1} = g(\xi)e^{i\xi x_j}$ then (1) becomes[3]

$$-r\left(g\; e^{i\xi x_j}e^{-i\xi h}\right) + g\; e^{i\xi x_j}(1+2r) - r\left(g\; e^{i\xi x_j}e^{i\xi h}\right) = r\left(e^{i\xi x_j}e^{-i\xi h}\right) + e^{i\xi x_j}(1-2r) + re^{i\xi x_j}e^{i\xi h}$$

$$g(-2r\cos(\xi h) + 1 + 2r) = 2r\cos(\xi h) + 1 - 2r$$

$$g(\xi) = \frac{1 + 2r\cos(\xi h) - 2r}{1 - 2r\cos(\xi h) + 2r}$$

Hence, the magnification factor is

$$g(\xi) = \frac{1 + \frac{D\Delta t}{h^2}(\cos(\xi h) - 1)}{1 - \frac{D\Delta t}{h^2}(\cos(\xi h) - 1)}$$

Let ξ be written as ξ_p in the above in order to examine g in terms of specific wave number ξ_p (which has units of radians per unit length). The above becomes

$$g\left(\xi_p\right) = \frac{1 + \frac{D\Delta t}{h^2}\left(\cos\left(\xi_p h\right) - 1\right)}{1 - \frac{D\Delta t}{h^2}\left(\cos\left(\xi_p h\right) - 1\right)} \tag{2}$$

But $\xi_p = p\pi$ and $p = 1 \cdots N$, with $h = \frac{1}{N+1}$where the line was discretized using the standard grid convention

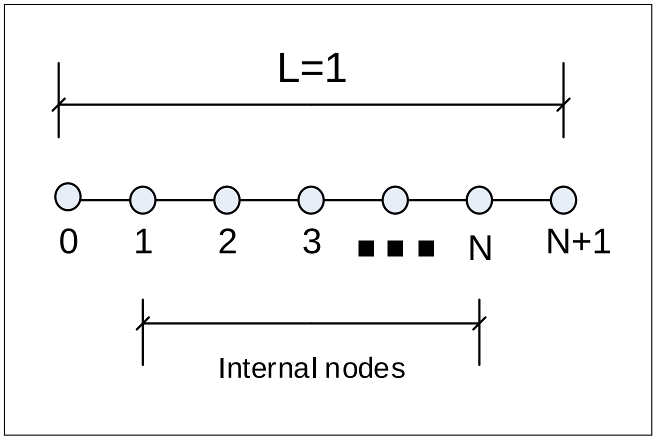


Figure 19: Grid format

Therefore $\cos\left(\xi_p h\right) = \cos\left(p\pi h\right) = \cos\left(\frac{p\pi}{N+1}\right)$. The largest frequency occurs when $p = N$, because then h is smallest, and the smallest frequency occurs when $p = 1$. Hence, there are a total of N Fourier modes when representing initial data as Fourier series. Now the magnification factor in (2) becomes

[2] The magnification factor is the term $g(\xi)$ in the expression relating $\hat{u}^{n+1}$ to $\hat{u}^n$ in the expression $\hat{u}^{n+1} = g(\xi)$ $\hat{u}^n$

[3] It is possible to derive the amplification factor using direct application of fourier transform, but the procedure is longer. The final result will be the same. The appendix of this problem contain this derivation.

$$g\left(\xi_p\right) = \frac{1 + \frac{D\Delta t}{h^2}\left(\cos\left(p\pi h\right) - 1\right)}{1 - \frac{D\Delta t}{h^2}\left(\cos\left(p\pi h\right) - 1\right)} \tag{3}$$

But since $\left|\cos\left(p\pi h\right)\right| \leq 1$, then $g\left(\xi_p\right)$ is less than 1 in magnitude for any p, implying that C-N is stable. To determine the magnitude of $g\left(\xi_p\right)$ when the mode has the largest frequency, let $p\pi h = \frac{N}{N+1}\pi \approx \pi$ in (3), resulting in

$$g(\xi_N) = \frac{1 - 2\frac{D\Delta t}{h^2}}{1 + 2\frac{D\Delta t}{h^2}} \tag{4}$$

When the time step $\Delta t \gg h$, then $\frac{D\Delta t}{h^2} \gg 1$, and in the limit $\left|g(\xi_N)\right| \rightarrow 1$. This shows that large frequency modes will decay very slowly because $g\left(\xi_p\right)$ is now close to 1. No attenuation will occur between each application of the update matrix or between each time step.

The above explains the result seen in part (a). Large frequency components did not decay fast as was expected, because the time step used was much larger than the space step. The problem asked us to use $\Delta t = 0.1$ and $h = 0.01$, which gives $\frac{D\Delta t}{h^2} = \frac{0.1}{0.01^2} = 1000$ and hence $g(\xi_N) \rightarrow \left|\frac{1-2000}{1+2000}\right| \rightarrow 0.9999$, and since this is almost one, then large frequency modes did not attenuate with each time step. The amplification factor needs to be small for attenuation to occur fast.

The following is a plot of $g\left(\xi_p\right)$ showing how the amplification factor changes as function of p for the case of $\Delta t = 0.1$ and $h = 0.01$, and $D = 1$. It shows what was found above, that at large frequency where p is close to N will have a correspondingly large

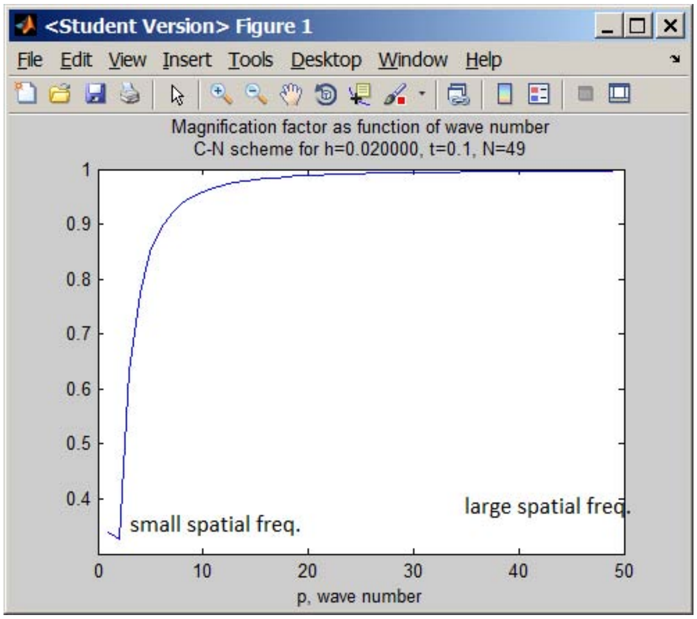

<Student Version> Figure 1
File Edit View Insert Tools Desktop Window Help
Magnification factor as function of wave number
C-N scheme for h=0.020000, t=0.1, N=49
1
0.9
0.8
0.7
0.6
0.5
0.4
small spatial freq.
large spatial freq.
0
10
20
30
40
50
p, wave number

Figure 20: corrected plot of amplification factor

To determine what value of $\frac{D\Delta t}{h^2}$ is required to make the large frequency mode decay right away, let $\frac{D\Delta t}{h^2} = \frac{1}{2}$ in (4), this gives $g(\xi_N) = 0$, which implies that large frequency mode will be knocked out right away. Here is a plot of $g(\xi_N) = \frac{1-2\frac{D\Delta t}{h^2}}{1+2\frac{D\Delta t}{h^2}}$ as a function of $\frac{D\Delta t}{h^2}$ showing that when $\frac{D\Delta t}{h^2} = 0$ then the magnification factor is minimum. This is only for mode $p = N$.

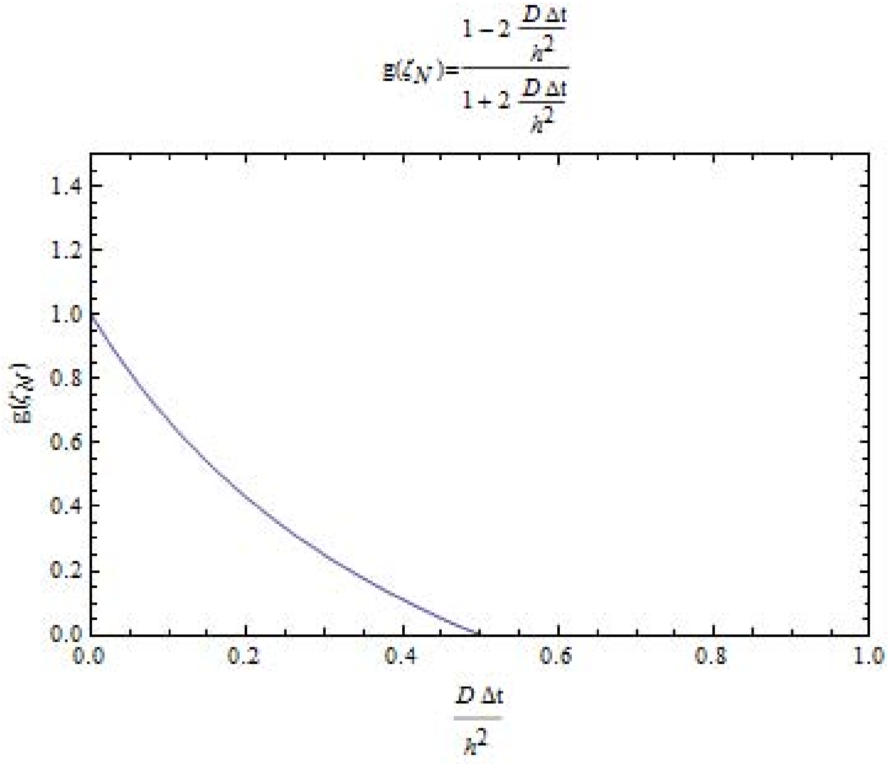


Figure 21: for mode $p = N$

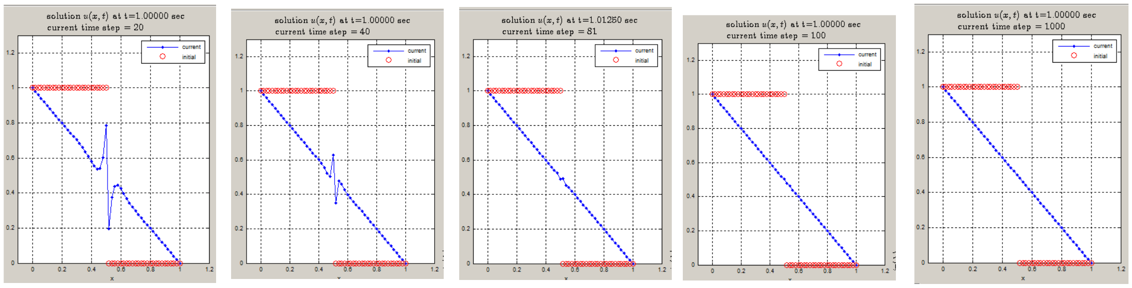
Conclusion If initial data contained large difference in value over very short distances (in other words, large spatial frequencies) such as given in this problem, producing discontinuity in data and its space derivative, and when the time step is large compared to the space step, then the numerical solution produced by C-N will not be accurate since large frequency modes will not attenuate.

To compensate for large frequency present in initial data, the ratio$\frac{D\Delta t}{h^2}$needs to be made close to 0.5 as possible. It might be better not to use C-N at all in such case and look for a scheme which does not have this problem.

notice that condition that $\frac{D\Delta t}{h^2} = \frac{1}{2}$ found above, is the same value for the upper limit for the absolute stability condition for the forward Euler discretization scheme for the 1D diffusion problem.

2.3 Part(c)

The time step was reduced and the program was run for 1 second. When the time step was reduced all the way to $\Delta t = 0.01$ then the final solution appeared smooth every where and in particular at $x = 0.5$. The following diagram illustrates this.



0.05 sec 0.025 sec 0.0125 sec 0.01 sec 0.001 sec

Figure 22: Final solution at $x = 0.5$

When using time step of 0.01 sec, 100 steps are used. From the last part it was found that $g(\xi_N) = \frac{1-2\frac{D\Delta t}{h^2}}{1+2\frac{D\Delta t}{h^2}}$, hence $|g| = \left|\frac{1-2\frac{0.01}{0.02^2}}{1+2\frac{0.01}{0.02^2}}\right| = 0.960\,78$, and therefore $|g|^{100} \to 0.960\,78^{100} \approx 0$ showing that by the 100^{th} iteration, the large mode frequency have completely smoothed out as verified by the above plots.

In the plot below, the magnification factor $|g(\xi)|$ is shown for $t = 0.01$ and $h = 0.02$ showing that at $p = 50$, $g(\xi) = 0.960\,78$. Compare this value with the one used in part(b) which was 0.999.

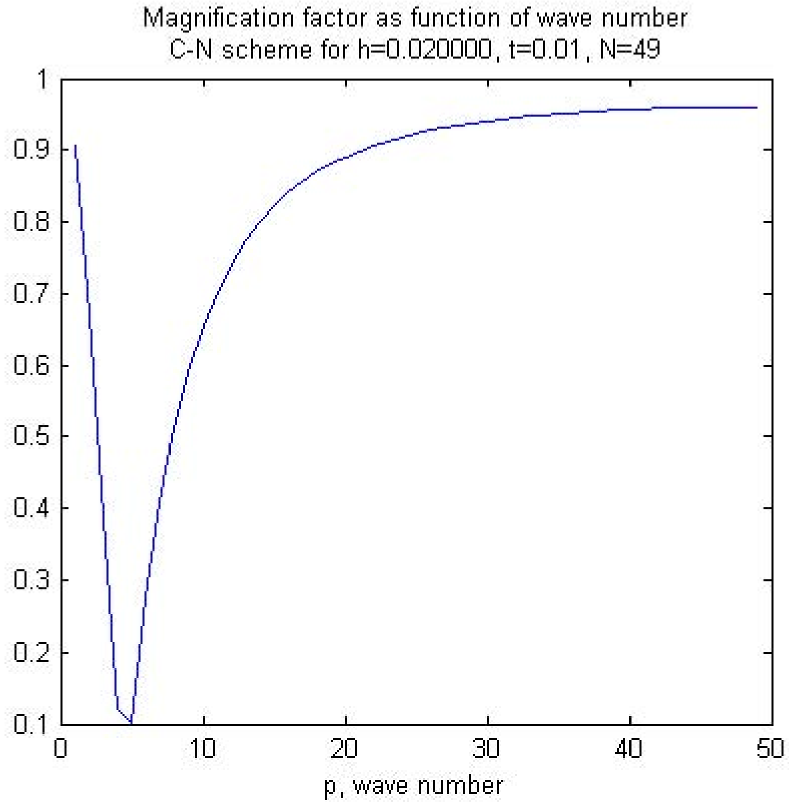


Figure 23: Comparing with the one used in part(b) which was 0.999.

To obtain smooth solution immediately after one time step, the required time step to accomplish this, can be determined from the condition for optimal amplification factor found in the last part which is given by $\frac{D\Delta t}{h^2} = \frac{1}{2}$. From this relation, and when $h = 0.02$ and $D = 1$, the time step will be $\Delta t = 0.0002$. This value of time step produces a smooth solution immediately (one step). This was confirmed, and here is the numerical solution, after only one time step, using $\Delta t = 0.0002, h = 0.02$ and $D = 1$

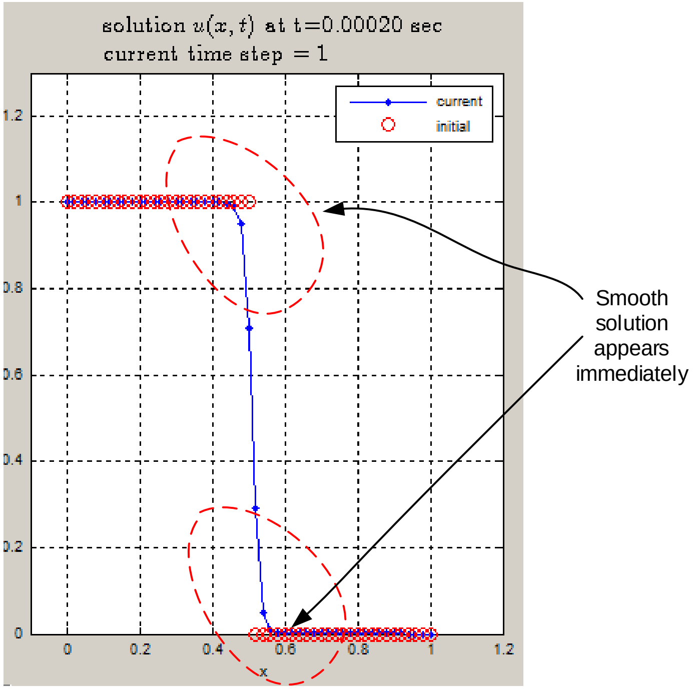


Figure 24: numerical solution, after only one time step, using $\Delta t = 0.0002, h = 0.02$ and $D = 1$

It is also possible to determine which Δt achieves a given specific attenuation of the high mode. Suppose it is required to attenuate the high mode to 0.001 of its initial amplitude at the end of 1 second run. Therefore, this means that

$$
\begin{aligned}
\left|g(\xi_N)\right|^{\frac{1}{\Delta t}} &= 0.001\\
\left|\frac{1-2\frac{D\Delta t}{h^2}}{1+2\frac{D\Delta t}{h^2}}\right|^{\frac{1}{\Delta t}} &= 0.001
\end{aligned}
$$

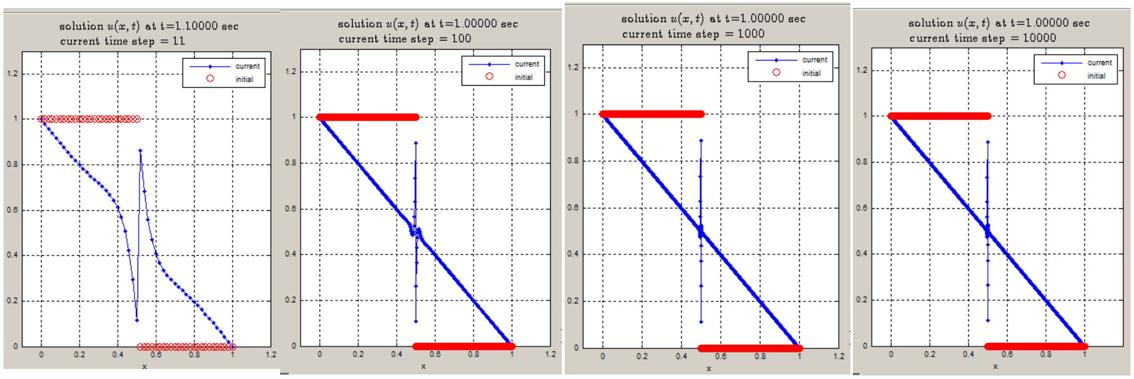
Taking logs, and using $h = 0.02$ and $D = 1$ results in

$$
\begin{aligned}
\frac{1}{\Delta t}\log\left(\frac{1-5000\Delta t}{1+5000\Delta t}\right) &= -3\\
\log(1-5000\Delta t) - \log(1+5000\Delta t) &= -3\Delta t
\end{aligned}
$$

The above is not a linear equation, but can be numerically solved for the root Δt. For the above example, Δt came out to be 0.00761 seconds. This means that when using $\Delta t = 0.00761$ sec, $h = 0.02$, and $D = 1$, then the largest frequency harmonic will have its amplitude attenuated to 0.01% of its original value after 1 second run.

2.4 Part(d)

Making the time step smaller and smaller, while keeping the ratio $\frac{\Delta t}{h}$ fixed, produces the following result (all runs are for one second). In this example, the ratio was kept at 5. The following sequence of ratios are used $\left\{\frac{0.1}{0.02}, \frac{0.01}{0.002}, \frac{0.001}{0.0002}, \frac{0.0001}{0.00002}\right\}$ to generate the following solution after one second run



Det=0.1, h=0.02

Det=0.01, h=0.002

Det=0.001, h=0.0002

Det=0.0001, h=0.00002

Figure 25: solution after one second run

Now, recall from part (b) that the magnification factor for the largest Fourier mode was given by

$$g(\xi_N) = \frac{1 - 2\frac{D\Delta t}{h^2}}{1 + 2\frac{D\Delta t}{h^2}}$$

When $\frac{\Delta t}{h}$ is held constant, say C, then the above becomes

$$g(\xi_N) = \frac{1 - Ch^{-1}}{1 + Ch^{-1}}$$

In the limit, as $h \to 0$ then $g(\xi_N) \to 1$, which implies, as was found in part(b), that large Fourier modes (high p values) will not be attenuated. This is confirmed by the plots above.

Using backward Euler in place of C-N When using backward Euler. The finite difference scheme for $u_t = Du_{xx}$ becomes

$$\begin{aligned} \frac{u_j^{n+1} - u_j^n}{\Delta t} &= f\left(u^{n+1}\right) \\ &= D\frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{h^2} \end{aligned} \tag{1}$$

Applying Von Neumann analysis, let $u_j^n = e^{i\xi x_j}$, and $u_j^{n+1} = g(\xi)e^{i\xi x_j}$, then (1) becomes

$$\begin{aligned}
g\ e^{i\xi x_j}e^{i\xi h} -\ e^{i\xi x_j} &= \frac{D\Delta t}{h^2} g\left(e^{i\xi x_j}e^{-i\xi h} - 2e^{i\xi x_j} + e^{i\xi x_j}e^{i\xi h}\right) \\
g\ e^{i\xi h} - 1 &= \frac{D\Delta t}{h^2} g\left(e^{-i\xi h} - 2 + e^{i\xi h}\right) \\
g\left(e^{i\xi h} - \frac{D\Delta t}{h^2}(2\cos(\xi h) - 2)\right) = 1 & \\
g(\xi) &= \frac{1}{\left(e^{i\xi h} - \frac{D\Delta t}{h^2}(2\cos(\xi h) - 2)\right)}
\end{aligned}$$

Therefore

$$g\left(\xi_p\right) = \frac{1}{\left(e^{i\xi_p h} - \frac{D\Delta t}{h^2}\left(2\cos\left(\xi_p h\right) - 2\right)\right)}$$

But $\xi_p = p\pi$ and $p = 1 \cdots N$, and to evaluate what happens to $g\left(\xi_p\right)$ at the largest spatial frequencies, let $\xi_p = N\pi$ and the above becomes

$$g(\xi_N) = \frac{1}{\left(e^{iN\pi h} - \frac{D\Delta t}{h^2}\left(2\cos\left(\frac{N}{N+1}\pi\right) - 2\right)\right)}$$

But $\frac{N}{N+1} \approx 1$ and $e^{iN\pi h} \approx e^{i\pi} = \cos(\pi) + i\sin\pi = -1$, then the above becomes

$$g(\xi_N) = \frac{1}{\left(-1 - \frac{D\Delta t}{h^2}(2\cos\pi - 2)\right)}$$

Therefore

$$g(\xi_N) = \frac{1}{4\frac{D\Delta t}{h^2} - 1}$$

When $\Delta t \gg h$ then $\left|g(\xi_N)\right| \to \frac{1}{1-1} \to 0$ which implies that large frequency modes will be knocked out fast. This is opposite to the situation observed using C-N. Hence backward Euler does not have the same problem with large spatial frequencies in initial data. But notice that when $\frac{D\Delta t}{h^2} = \frac{1}{2}$, now $g(\xi_N) = \frac{1}{2-1} = 1$, which means that large frequency mode will not decay or decay very slowly, This is also opposite of what was found for C-N.

2.5 Appendix for problem 2

Derivation of part (b) by direct application of DFT (the harder way) The C-N scheme for $u_t = Du_{xx}$ given by

$$-ru_{j-1}^{n+1} + u_j^{n+1}(1 + 2r) - ru_{j+1}^{n+1} = ru_{j-1}^n + u_j^n(1 - 2r) + ru_{j+1}^n \tag{2}$$

Assuming the problem is on the whole real line, then

$$u_j^n = \frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{u}^n(\xi) e^{i\xi x_j} d\xi \tag{1A}$$

and

$$u_j^{n+1} = \frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{u}^{n+1}(\xi) e^{i\xi x_j} d\xi \tag{1B}$$

Where $\hat{u}^n(\xi)$ is the discrete Fourier transform (DFT) of u_j^n. In what follows, $\hat{u}^n$ is written instead of $\hat{u}^n(\xi)$ to make it easier to read the equations. The C-N finite difference scheme for $u_t = Du_{xx}$ is given by

$$-ru_{j-1}^{n+1} + u_j^{n+1}(1+2r) - ru_{j+1}^{n+1} = ru_{j-1}^n + u_j^n(1-2r) + ru_{j+1}^n \tag{2}$$

Where $r = \frac{\Delta t D}{2h^2}$. Substitute (1A) and (1B) into (2), but leaving u_j^{n+1} as is gives

$$-r\left(\frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{u}^{n+1} e^{i\xi\left(x_j - h\right)}\ d\xi\right) + u_j^{n+1}(1+2r) - r\left(\frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{u}^{n+1} e^{i\xi\left(x_j + h\right)} d\xi\right) =$$

$$r\left(\frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{u}^n e^{i\xi\left(x_j - h\right)} d\xi\right) + (1-2r)\left(\frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{u}^n e^{i\xi x_j} d\xi\right) + r\left(\frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{u}^n e^{i\xi\left(x_j + h\right)} d\xi\right)$$

or

$$\frac{-r}{\sqrt{2\pi}}\left(\int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{u}^{n+1} e^{i\xi x_j}\ \left(e^{i\xi h} + e^{-i\xi h}\right) d\xi\right) + u_j^{n+1}(1+2r) =$$

$$\frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \left(r\ \hat{u}^n\ e^{i\xi x_j}\ e^{-i\xi h} + \hat{u}^n\ e^{i\xi x_j} - 2\ r\ \hat{u}^n\ e^{i\xi x_j} + r\ \hat{u}^n\ e^{i\xi x_j}\ e^{i\xi h}\right) d\xi$$

Simplify

$$\begin{aligned} u_j^{n+1}(1+2r) &= \frac{r}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{u}^{n+1} e^{i\xi x_j}\ \left(e^{i\xi h} + e^{-i\xi h}\right) d\xi + \frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{u}^n e^{i\xi x_j} \left[r\left(e^{i\xi h} + e^{-i\xi h}\right) + 1 - 2r\right] d\xi \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{u}^{n+1}\ 2r\cos(\xi h) e^{i\xi x_j} d\xi + \frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \hat{u}^n (2r(\cos(\xi h) - 1) + 1) e^{i\xi x_j} d\xi \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} \left(\hat{u}^n (2r(\cos(\xi h) - 1) + 1) + 2r\cos(\xi h) \hat{u}^{n+1}\right) e^{i\xi x_j} d\xi \end{aligned}$$

Hence

$$\begin{aligned} DFT\left(u_j^{n+1}(1+2r)\right) &= \left(\hat{u}^n(2r(\cos(\xi h)-1)+1)+2r\cos(\xi h)\hat{u}^{n+1}\right) \\ DFT\left(u_j^{n+1}\right) &= \frac{1}{(1+2r)}\left(\hat{u}^n(2r(\cos(\xi h)-1)+1)+2r\cos(\xi h)\hat{u}^{n+1}\right) \end{aligned}$$

This implies that

$$\hat{u}^{n+1} = \frac{1}{(1+2r)}\left(\hat{u}^n(2r(\cos(\xi h)-1)+1)+2r\cos(\xi h)\hat{u}^{n+1}\right)$$

Solving for $\hat{u}^{n+1}$ gives

$$\begin{aligned} \hat{u}^{n+1} - \frac{2r\cos(\xi h)\hat{u}^{n+1}}{(1+2r)} &= \frac{1}{(1+2r)}\hat{u}^n(2r(\cos(\xi h)-1)+1) \\ \hat{u}^{n+1}\left(\frac{(1+2r)-2r\cos(\xi h)}{(1+2r)}\right) &= \frac{\hat{u}^n(2r(\cos(\xi h)-1)+1)}{(1+2r)} \end{aligned}$$

Hence

$$\begin{aligned} \hat{u}^{n+1} &= \frac{(2r(\cos(\xi h)-1)+1)}{(1+2r)-2r\cos(\xi h)}\hat{u}^n \\ &= \frac{1+2r\cos(\xi h)-2r}{1-2r\cos(\xi h)+2r}\hat{u}^n \end{aligned}$$

Therefore $\hat{u}^{n+1} = g(\xi)\hat{u}^n$, where $g(\xi) = \frac{1+2r\cos(\xi h)-2r}{1-2r\cos(\xi h)+2r}$, and since $r = \frac{\Delta t D}{2h^2}$, then

$$g(\xi) = \frac{1+\frac{D\Delta t}{h^2}\left(\cos\left(p\pi h\right)-1\right)}{1-\frac{D\Delta t}{h^2}\left(\cos\left(p\pi h\right)-1\right)}$$

where $\xi_p = p\pi$, is the wave number.

Another derivation for the magnification factor The magnification factor $g(\xi)$ found above is the same as the eigenvalue of the update matrix of the C-N scheme. From

$$\left(I - \frac{D\Delta t}{2}L\right)u^{n+1} = \left(I + \frac{D\Delta t}{2}L\right)u^n$$

or

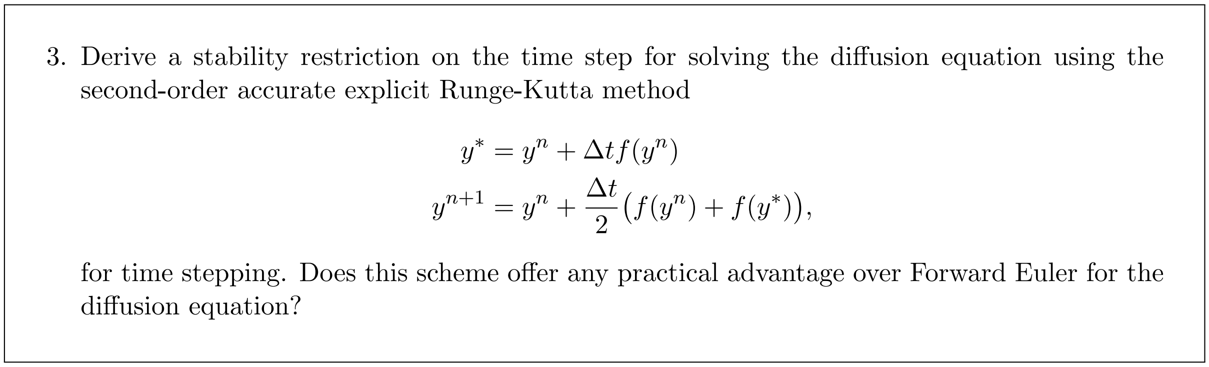
$$u^{n+1} = \left(I - \frac{D\Delta t}{2}L\right)^{-1}\left(I + \frac{D\Delta t}{2}L\right)u^n$$

Where L is the 1D Laplacian grid operator which has eigenvalues $\lambda_p = \frac{2}{h^2}\left(\cos\left(p\pi h\right)-1\right)$, hence, let μ_p be the eigenvalue of B above, then

$$\begin{aligned}\mu_p &= \frac{1+\frac{D\Delta t}{2}\lambda_p}{1-\frac{D\Delta t}{2}\lambda_p}\\ &= \frac{1+\frac{D\Delta t}{2}\frac{2}{h^2}\left(\cos\left(p\pi h\right)-1\right)}{1-\frac{D\Delta t}{2}\frac{2}{h^2}\left(\cos\left(p\pi h\right)-1\right)}\\ &= \frac{1+\frac{D\Delta t}{h^2}\left(\cos\left(p\pi h\right)-1\right)}{1-\frac{D\Delta t}{h^2}\left(\cos\left(p\pi h\right)-1\right)}\end{aligned}$$

Which is what was found in part(b)

3 Problem 3



3. Derive a stability restriction on the time step for solving the diffusion equation using the second-order accurate explicit Runge-Kutta method

$$\begin{aligned} y^* &= y^n + \Delta t f(y^n) \\ y^{n+1} &= y^n + \frac{\Delta t}{2}\big(f(y^n) + f(y^*)\big), \end{aligned}$$

for time stepping. Does this scheme offer any practical advantage over Forward Euler for the diffusion equation?

Figure 26: Problem statement

The finite difference scheme for the diffusion problem is shown the appendix of this problem.

To obtain the absolute stability restriction, let

$$\begin{aligned} y^{n+1} - y^n &= \frac{\Delta t}{2}\Big(f\big(y^n\big) + f\big(y^*\big)\Big) \\ y^{n+1} &= y^n + \frac{\Delta t}{2}\big(\lambda y^n + \lambda y^*\big) \\ &= y^n + \frac{\Delta t}{2}\Big(\lambda y^n + \lambda\big(y^n + \Delta t \lambda y^n\big)\Big) \\ &= y^n + \frac{\Delta t}{2}\lambda y^n + \frac{\Delta t}{2}\lambda\big(y^n + \Delta t \lambda y^n\big) \\ &= y^n + \frac{\Delta t}{2}\lambda y^n + \frac{\Delta t}{2}\lambda y^n + \frac{(\lambda \Delta t)^2}{2} y^n \\ &= \left(1 + \Delta t \lambda + \frac{(\lambda \Delta t)^2}{2}\right) y^n \end{aligned}$$

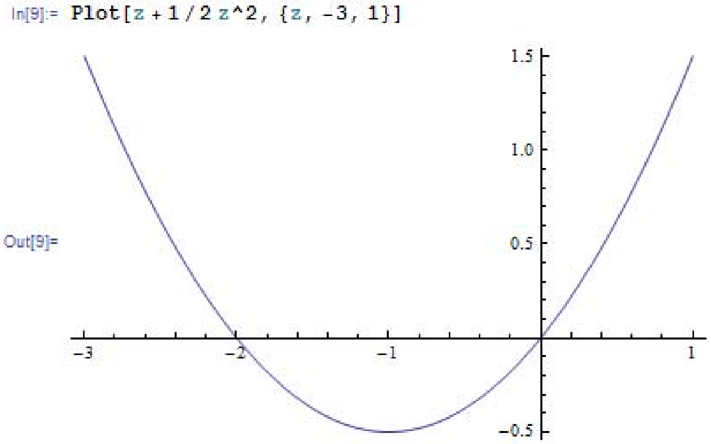
Assuming $\Delta t \lambda = z$

$$y^n + 1 = \left(1 + z + \frac{1}{2}z^2\right) y^n$$

Hence $R(z) = 1 + z + \frac{1}{2}z^2$ and for absolute stability it is required that $|R(z)| \leq 1$ which leads to

$$\begin{aligned} -1 \leq 1 + z + \frac{1}{2}z^2 \leq 1 \\ -2 \leq z + \frac{1}{2}z^2 \leq 0 \end{aligned}$$

A plot of $z + \frac{1}{2}z^2$ shows that $-2 \leq z \leq 0$



```
In[9]:= Plot[z + 1 / 2 z^2, {z, -3, 1}]
```

Figure 27: absolute stability region

The above gives the interval of absolute stability for the eigenvalues. To obtain the region of absolute stability, assume λ can be complex in general (complex eigenvalue), which results in a disk of radius 1 centered at -1. This is the same region of stability as Forward Euler.

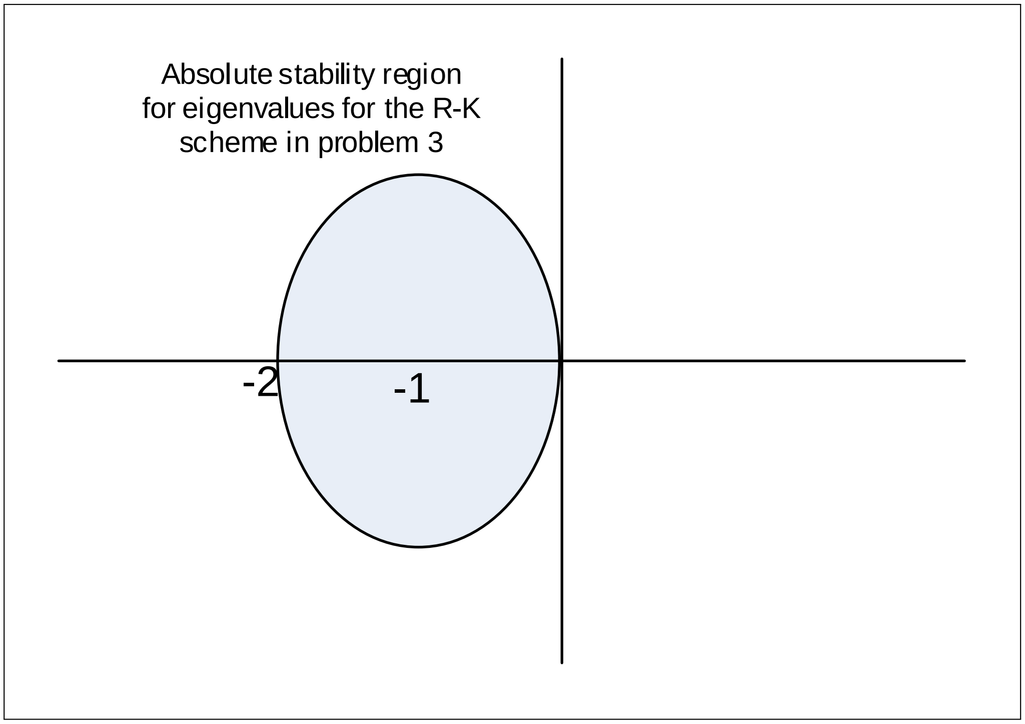


Figure 28: region of absolute stability

Notice that since $R(0) = 1$ and $\frac{dR(z)}{dz} = 1$, then this method is also consistent and first order accurate in time.

To answer the question about any advantage of this method over forward Euler. Recalling that In forward Euler

$$u^{n+1} = Bu^n$$

Where B is the update matrix given by $B = I + D\Delta t L_x$, where L_x is the 1-D Laplacian operator for u_{xx} with Dirichlet boundary conditions, with eigenvalue $\lambda = \frac{2}{h^2}\left(\cos\left(p\pi h\right) - 1\right)$ where $p = 1 \cdots N$ using the standard grid convention used before.

Let μ be the eigenvalue of the above update matrix B, hence

$$\mu = 1 + \frac{2D\Delta t}{h^2}\left(\cos\left(p\pi h\right) - 1\right)$$

For stability, $|u| \leq 1$ hence

$$\left|1 + \frac{2D\Delta t}{h^2}\left(\cos\left(p\pi h\right) - 1\right)\right| \leq 1$$

$$\left|1 - 4\frac{D\Delta t}{h^2}\right| \leq 1$$

Simplifying, this gives

$$0 \leq \frac{D\Delta t}{h^2} \leq \frac{1}{2}$$

To compare the above with the R-K scheme in this problem, since

$$\begin{aligned} \frac{u^* - u^n}{\Delta t} &= Lu^n \\ \frac{u^{n+1} - u^n}{\Delta t} &= \frac{1}{2}(Lu^n + Lu^*) \end{aligned} \tag{1}$$

Expanding gives

$$\begin{aligned} u^{n+1} &= u^n + \frac{\Delta t}{2}(Lu^n + L(u^n + \Delta t L u^n)) \\ &= u^n + \frac{\Delta t}{2}\left(Lu^n + Lu^n + \Delta t L^2 u^n\right) \\ &= \left(I + \frac{\Delta t}{2}\left(2L + \Delta t L^2\right)\right)u^n \\ &= Bu^n \end{aligned}$$

The eigenvalue of L is $\frac{2}{h^2}\left(\cos\left(p\pi h\right) - 1\right)$ but since $\cos A - 1 = -2\sin^2\frac{A}{2}$, hence $\cos\left(p\pi h\right) - 1 = -2\sin^2\frac{p\pi h}{2}$ and the eigenvalue is written as $-\frac{4}{h^2}\sin^2\frac{p\pi h}{2}$.

let $\sin^2\frac{p\pi h}{2} \equiv \omega$, therefore the eigenvalue of L becomes $\lambda = -\frac{4\omega}{h^2}$.

Using this notation, the above update matrix B for this scheme will have the following eigenvalue

$$\begin{aligned} \mu &= 1 + \frac{\Delta t}{2}\left(-D\frac{8\omega}{h^2} + \Delta t\left(-D\frac{4\omega}{h^2}\right)^2\right) \\ &= 1 - \frac{4D\Delta t\omega}{h^2} + \Delta^2 t\frac{8D^2\omega^2}{h^4} \end{aligned}$$

For stability, $\left|\mu\right| \leq 1$. Maximum μ will occur at $\omega = 1$ implying that $\sin^2 \frac{p\pi h}{2} = 1$ or $p = N$, resulting in

$$\left|1 - \frac{4D\Delta t}{h^2} + \Delta^2 t \frac{8D^2}{h^4}\right| \leq 1$$

Therefore

$$\begin{aligned}
-1 \leq 1 - \frac{4D\Delta t}{h^2} + \Delta^2 t \frac{8D^2}{h^4} &\leq 1 \\
-2 \leq -\frac{4D\Delta t}{h^2} + \Delta^2 t \frac{8D^2}{h^4} &\leq 0 \\
0 \leq \frac{4D\Delta t}{h^2} - \Delta^2 t \frac{8D^2}{h^4} &\leq 2 \\
0 \leq \frac{D\Delta t}{h^2} - \Delta^2 t \frac{2D^2}{h^4} &\leq \frac{1}{2} \\
0 \leq \frac{D\Delta t}{h^2} - 2\left(\frac{D\Delta t}{h^2}\right)^2 &\leq \frac{1}{2}
\end{aligned}$$

This shows that with the given RK scheme, stability implies the condition $0 \leq \frac{D\Delta t}{h^2} - \Delta^2 t \frac{2D^2}{h^4} \leq \frac{1}{2}$. This is compared to $0 \leq \frac{D\Delta t}{h^2} \leq \frac{1}{2}$ for forward Euler.

What does this mean in terms of the time step? Will this allow the use of a larger time step than with FE while keep absolute stability?

Assume $D = 1$, This is a table showing the maximum value of Δt allowed for different h values

scheme	$h = 1$	$h = 0.1$	$h = 0.01$
FE $0 \leq \frac{\Delta t}{h^2} \leq \frac{1}{2}$	0.5	0.05	0.005
RK $0 \leq \frac{\Delta t}{h^2} - \frac{2\Delta^2 t}{h^4} \leq \frac{1}{2}$	0.5	0.05	0.005

Hence, the largest time step does not change with this scheme when compared to forward Euler. It seems based on the above, that the explicit Runge-Kutta scheme for solving the diffusion PDE does not offer any advantage in handling the stiffness of the PDE since the time step remained constrained by h^2 as with FE. It seems that explicit schemes are not suitable for stiff problems

3.1 Appendix for problem 3

3.2 Derivation of the update matrix for 2 step R-K for diffusion problem with Dirichlet B.C.

Given

$$y_t = D y_{xx}$$

Then

$$\frac{y_i^* - y_i^n}{\Delta t} = D\frac{y_{i-1}^n - 2y_i^n + y_{i+1}^n}{h^2}$$

$$y_i^* = y_i^n + \Delta t\overbrace{\frac{D}{h^2}\left(y_{i-1}^n - 2y_i^n + y_{i+1}^n\right)}^{f\left(y^n\right)}$$

Hence

$$\begin{aligned}
2\frac{y_i^{n+1} - y_i^n}{\Delta t} &= \left(f\left(y^n\right) + f\left(y^*\right)\right) \\
&= \left[\frac{D}{h^2}\left(y_{i-1}^n - 2y_i^n + y_{i+1}^n\right)\right] + \left[\frac{D}{h^2}\left(y_{i-1}^* - 2y_i^* + y_{i+1}^*\right)\right]
\end{aligned}$$

$$\begin{aligned}
&= \frac{D}{h^2}\left(y_{i-1}^n - 2y_i^n + y_{i+1}^n\right) + \\
&\frac{D}{h^2}\left[y_{i-1}^n + \Delta t\frac{D}{h^2}\left(y_{i-2}^n - 2y_{i-1}^n + y_i^n\right)\right] - \\
&2\frac{D}{h^2}\left[y_i^n + \Delta t\frac{D}{h^2}\left(y_{i-1}^n - 2y_i^n + y_{i+1}^n\right)\right] + \\
&\frac{D}{h^2}\left[y_{i+1}^n + \Delta t\frac{D}{h^2}\left(y_i^n - 2y_{i+1}^n + y_{i+2}^n\right)\right]
\end{aligned}$$

Expand and simplify

$$\begin{aligned}
2\frac{y_i^{n+1} - y_i^n}{\Delta t} &= \frac{D}{h^2}y_{i-1}^n\left(1 + 1 - 2\Delta t\frac{D}{h^2} - 2\Delta t\frac{D}{h^2}\right) \\
&+ \frac{D}{h^2}y_i^n\left(-2 + \Delta t\frac{D}{h^2} - 2 - 4\Delta t\frac{D}{h^2} + \Delta t\frac{D}{h^2}\right) \\
&+ \frac{D}{h^2}y_{i+1}^n\left(1 - 2\Delta t\frac{D}{h^2} + 1 - 2\Delta t\frac{D}{h^2}\right) \\
&+ \frac{D}{h^2}y_{i+2}^n\left(\Delta t\frac{D}{h^2}\right)
\end{aligned}$$

$$= \frac{D}{h^2}\left[y_{i-1}^n\left(2 - 4\Delta t\frac{D}{h^2}\right) + y_i^n\left(-4 - 2\Delta t\frac{D}{h^2}\right) + y_{i+1}^n\left(2 - 4\Delta t\frac{D}{h^2}\right) + y_{i+2}^n\left(\Delta t\frac{D}{h^2}\right)\right]$$

Hence

$$
\begin{aligned}
y_i^{n+1} &= y_i^n + \frac{D\Delta t}{2h^4}\left[y_{i-1}^n\left(2h^2 - 4\Delta tD\right) + y_i^n\left(-4h^2 - 2\Delta tD\right) + y_{i+1}^n\left(2h^2 - 4\Delta tD\right) + y_{i+2}^n(\Delta tD)\right] \\
&= y_i^n + \frac{D^2\Delta^2 t}{2h^4}\left[2y_{i-1}^n\left(\frac{h^2}{\Delta tD} - 2\right) - 2y_i^n\left(\frac{2h^2}{\Delta tD} + 1\right) + 2y_{i+1}^n\left(\frac{h^2}{\Delta tD} - 2\right) + y_{i+2}^n\right]
\end{aligned}
$$

$$
\begin{aligned}
&= y_{i-1}^n\left(\frac{D^2\Delta^2 t}{2h^4}\left(\frac{h^2}{\Delta tD} - 2\right)\right) + y_i^n\left(1 - \frac{D^2\Delta^2 t}{h^4}\left(\frac{2h^2}{\Delta tD} + 1\right)\right) + \\
&y_{i+1}^n\left(\frac{D^2\Delta^2 t}{h^4}\left(\frac{h^2}{\Delta tD} - 2\right)\right) + y_{i+2}^n\frac{D^2\Delta^2 t}{2h^4}
\end{aligned}
$$

Let $\frac{D^2\Delta^2 t}{h^4} = r_1$ and let $\frac{h^2}{\Delta tD} = r_2$ then[4]

$$
\begin{aligned}
y_i^{n+1} &= y_{i-1}^n\left(\frac{r_1}{2}(r_2 - 2)\right) + y_i^n(1 - r_1(2r_2 + 1)) + y_{i+1}^n(r_1(r_2 - 2)) + y_{i+2}^n\frac{r_1}{2} \\
&= y_{i-1}^n\left(\frac{r_1 r_2}{2} - r_1\right) + y_i^n(1 - 2r_1 r_2 - r_1) + y_{i+1}^n(r_1 r_2 - 2r_1) + y_{i+2}^n\frac{r_1}{2}
\end{aligned}
$$

Using this 1-D grid

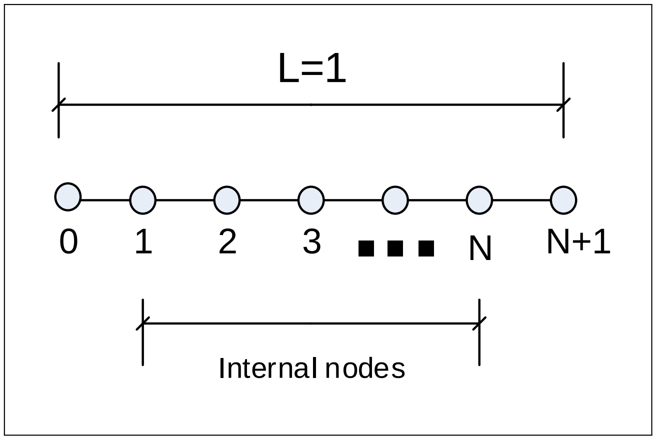


Figure 29: Grid format

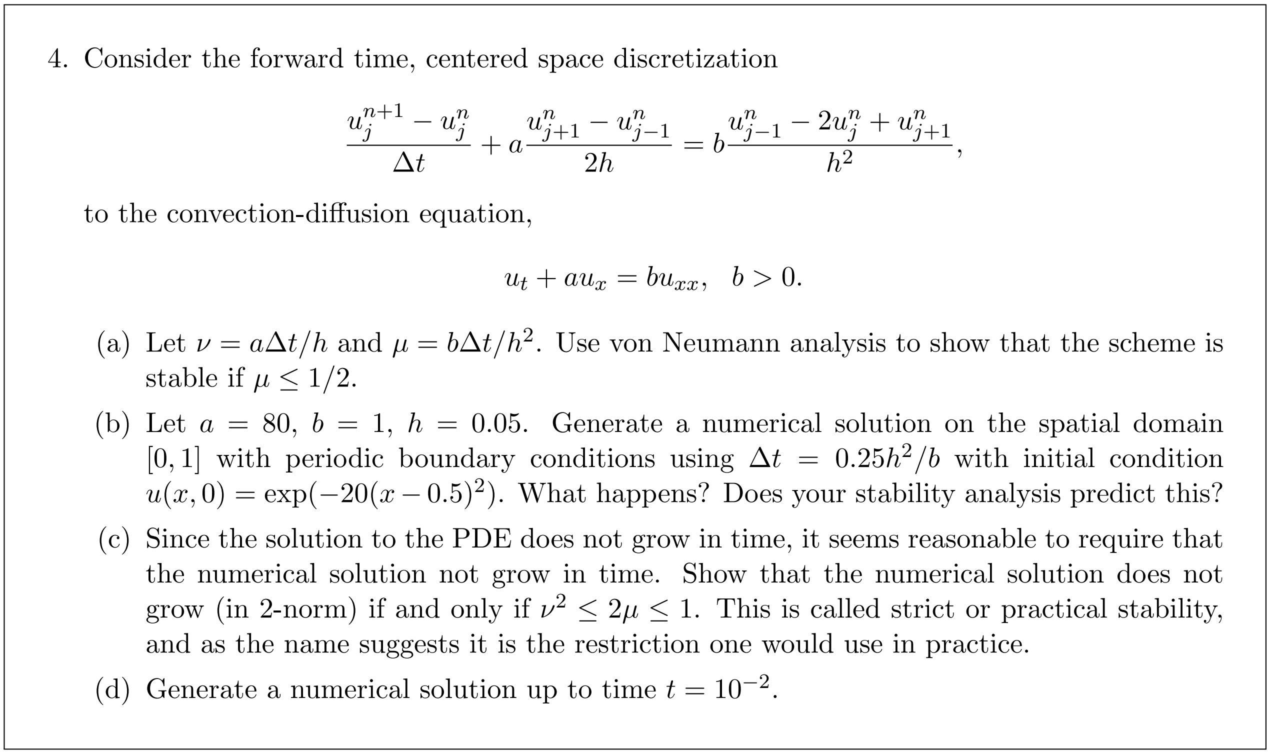
[4] Notice that units of D are *meter*^{2} per second, hence $\frac{h^2}{\Delta tD}$ is dimensionless, so we are ok.

The matrix form of the scheme becomes

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ u_{N-1}^{n+1} \\ u_N^{n+1} \end{bmatrix} = \begin{bmatrix} (1-2r_1r_2-r_1) & (r_1r_2-2r_1) & \frac{r_1}{2} & 0 & 0 & 0 \\ \left(\frac{r_1r_2}{2}-r_1\right) & (1-2r_1r_2-r_1) & (r_1r_2-2r_1) & \frac{r_1}{2} & 0 & 0 \\ 0 & \left(\frac{r_1r_2}{2}-r_1\right) & (1-2r_1r_2-r_1) & (r_1r_2-2r_1) & \frac{r_1}{2} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \left(\frac{r_1r_2}{2}-r_1\right) & (1-2r_1r_2-r_1) & (r_1r_2-2r_1) \\ 0 & 0 & 0 & 0 & \left(\frac{r_1r_2}{2}-r_1\right) & (1-2r_1r_2-r_1) \end{bmatrix} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ u_N^n \end{bmatrix} + \begin{bmatrix} \left(\frac{r_1r_2}{2}-r_1\right)u_0^n \\ 0 \\ 0 \\ \vdots \\ y_{N+1}^n \frac{r_1}{2} \\ (r_1r_2-2r_1)u_{N+1}^n \end{bmatrix}$$

There is a problem above at node N as two additional nodes are needed to its right, but only one node exist. Need to look more into this later, as this part is not required for this HW.

4 Problem 4



4. Consider the forward time, centered space discretization

$$\frac{u_j^{n+1}-u_j^n}{\Delta t}+a\frac{u_{j+1}^n-u_{j-1}^n}{2h}=b\frac{u_{j-1}^n-2u_j^n+u_{j+1}^n}{h^2},$$

to the convection-diffusion equation,

$$u_t+au_x=bu_{xx},\quad b>0.$$

(a) Let $\nu=a\Delta t/h$ and $\mu=b\Delta t/h^2$. Use von Neumann analysis to show that the scheme is stable if $\mu\le 1/2$.

(b) Let $a=80$, $b=1$, $h=0.05$. Generate a numerical solution on the spatial domain $[0,1]$ with periodic boundary conditions using $\Delta t=0.25h^2/b$ with initial condition $u(x,0)=\exp(-20(x-0.5)^2)$. What happens? Does your stability analysis predict this?

(c) Since the solution to the PDE does not grow in time, it seems reasonable to require that the numerical solution not grow in time. Show that the numerical solution does not grow (in 2-norm) if and only if $\nu^2\le 2\mu\le 1$. This is called strict or practical stability, and as the name suggests it is the restriction one would use in practice.

(d) Generate a numerical solution up to time $t=10^{-2}$.

Figure 30: Problem statement

4.1 Part (a)

The PDE is $u_t+au_x=bu_{xx}$, with $b>0$, the forward time, centered space discretization is

$$\frac{u_j^{n+1}-u_j^n}{\Delta t}+a\frac{u_{j+1}^n-u_{j-1}^n}{2h}=b\frac{u_{j-1}^n-2u_j^n+u_{j+1}^n}{h^2}\tag{1}$$

Applying Von Neumann analysis, let $u_j^n=e^{i\xi x_j}$ and $u_j^{n+1}=g(\xi)e^{i\xi x_j}$, then (1) becomes

$$\begin{aligned}\frac{ge^{i\xi x_j}-e^{i\xi x_j}}{\Delta t}+a\frac{e^{i\xi x_j}e^{i\xi h}-e^{i\xi x_j}e^{-i\xi h}}{2h}&=b\frac{e^{i\xi x_j}e^{-i\xi h}-2e^{i\xi x_j}+e^{i\xi x_j}e^{i\xi h}}{h^2}\\ ge^{i\xi x_j}-e^{i\xi x_j}+\frac{a\Delta t}{2h}\left(e^{i\xi x_j}e^{i\xi h}-e^{i\xi x_j}e^{-i\xi h}\right)&=\frac{b\Delta t}{h^2}\left(e^{i\xi x_j}e^{-i\xi h}-2e^{i\xi x_j}+e^{i\xi x_j}e^{i\xi h}\right)\\ g-1+\frac{\nu}{2}\left(e^{i\xi h}-e^{-i\xi h}\right)&=\mu\left(e^{-i\xi h}-2+e^{i\xi h}\right)\\ g(\xi)&=1-\frac{\nu}{2}\left(e^{i\xi h}-e^{-i\xi h}\right)+\mu\left(e^{-i\xi h}-2+e^{i\xi h}\right)\end{aligned}$$

Hence[5]

$$g(\xi)=1+2\mu(\cos(\xi h)-1)-i\nu\sin(\xi h)$$

[5] Notice that the first oder derivatives (or odd order in general) produces eigenvalues that are complex, and the even order ones produce real eigenvalues.

But $\cos A - 1 = -2\sin^2\frac{A}{2}$, hence $\cos(\xi h) - 1 = -2\sin^2\frac{\xi h}{2}$ and the above becomes

$$g(\xi) = 1 - 4\mu\sin^2\left(\frac{\xi h}{2}\right) - i\nu\sin(\xi h)$$

Therefore

$$\left|g(\xi)\right|^2 = \left[1 - 4\mu\sin^2\left(\frac{\xi h}{2}\right)\right]^2 + \nu^2\sin^2(\xi h) \tag{2}$$

Using the trig identity

$$\sin^2(\xi h) = 4\sin^2\left(\frac{\xi h}{2}\right)\left(1 - \sin^2\left(\frac{\xi h}{2}\right)\right)$$

then (2) becomes

$$\left|g(\xi)\right|^2 = \left[1 - 4\mu\sin^2\left(\frac{\xi h}{2}\right)\right]^2 + 4\nu^2\sin^2\left(\frac{\xi h}{2}\right)\left(1 - \sin^2\left(\frac{\xi h}{2}\right)\right)$$

Let $\sin^2\left(\frac{\xi h}{2}\right) \equiv \omega$ in the above

$$\left|g(\xi)\right|^2 = \left(1 - 4\mu\omega\right)^2 + 4\nu^2\omega(1 - \omega)$$

The maximum of $\left|g\left(\xi_p\right)\right|$ occurs when $\xi h \approx \pi$, making $\omega = 1$, hence from above, the maximum of $\left|g(\xi)\right|^2$ is reduced to $1 - 4\mu$, and then for stability

$$\left|1 - 4\mu\right| \leq 1$$

therefore

$$\begin{aligned} -1 &\leq 1 - 4\mu \leq 1 \\ -2 &\leq -4\mu \leq 0 \\ 0 &\leq 4\mu \leq 2 \end{aligned}$$

Hence

$$0 \leq \mu \leq \frac{1}{2}$$

The above result can also be derived using the stencil diagram method. The stencil diagram for the above scheme (for internal nodes only) is

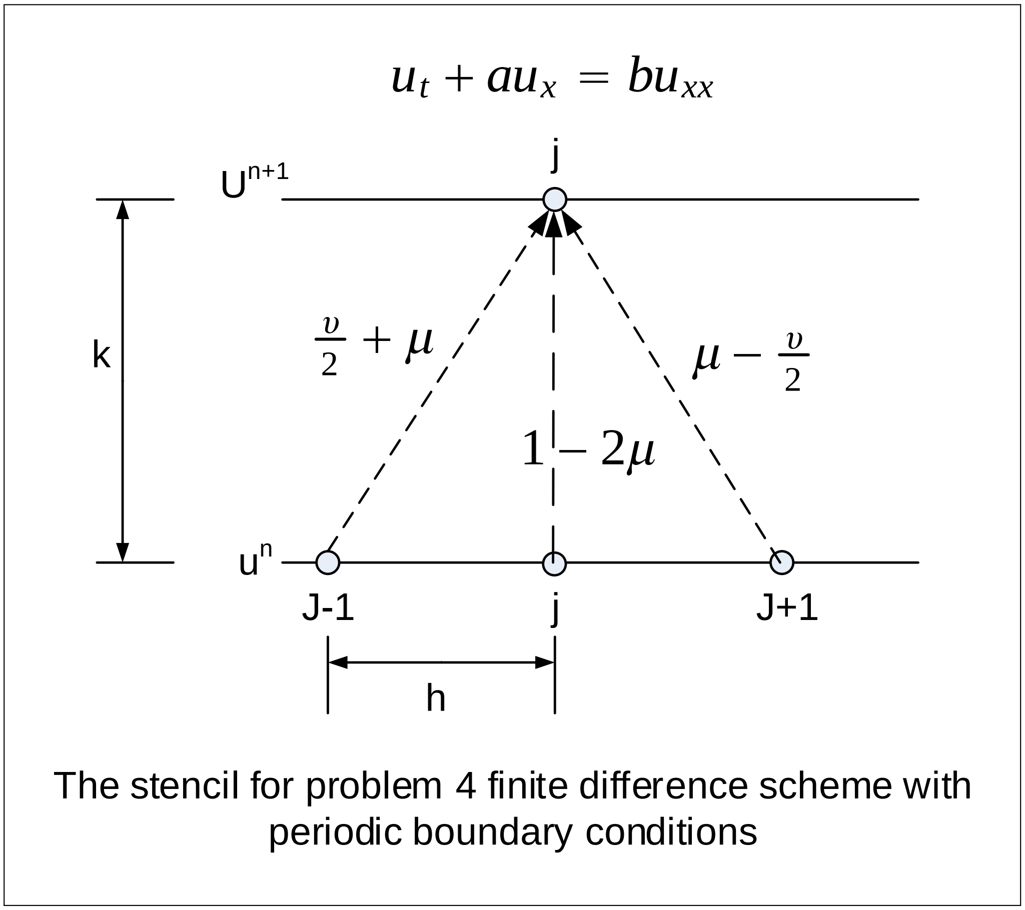


Figure 31: Grid used

As was done in problem 1, By imposing that the weight on each edge in the above directed graph not exceed unity, and that the total algebraic sum of the weight of the edge also not exceed unity. This includes any combination of edges involved. For if this was the case, then u_j^{n+1} will always have an amplitude $\leq u_j^n$. Applying this to the above diagram gives

$$\begin{cases} (1) & \frac{v}{2}+\mu \leq 1 & \text{Condition on } j-1 \text{ edge} \\ (2) & \left|\mu-\frac{v}{2}\right| \leq 1 & \text{Condition on } j+1 \text{ edge} \\ (3) & 2\mu \leq 1 & \text{Condition on } j-1 \text{ added to } j+1 \text{ edge} \\ (4) & \left|1-2\mu\right| \leq 1 & \text{Condition on the } j \text{ edge} \\ (5) & \left|1+\frac{v}{2}-\mu\right| \leq 1 & \text{Condition on the } j \text{ edge added to } j-1 \\ (6) & \left|1-\frac{v}{2}-\mu\right| \leq 1 & \text{Condition on the } j \text{ edge added to } j+1 \\ (7) & 1 \leq 1 & \text{Condition that all edges sum to less than 1} \end{cases}$$

Condition (7) gives no information. Condition (4) gives $\mu \leq 1$ and hence weaker than (3), condition (5) is the same as (2) and gives $\mu - \frac{v}{2} \leq 1$, condition (6) gives $\frac{v}{2} + \mu \leq 2$ which is weaker than condition (1). Hence the following are remaining conditions (3),(1),(2).

$$\begin{cases} (1) & \frac{v}{2}+\mu \leq 1 & \text{Condition on } j-1 \text{ edge} \\ (2) & \left|\mu-\frac{v}{2}\right| \leq 1 & \text{Condition on } j+1 \text{ edge} \\ (3) & 2\mu \leq 1 & \text{Condition on } j-1 \text{ added to } j+1 \text{ edge} \end{cases}$$

From the above, only condition (3) can provide useful information, which is that $\mu \leq \frac{1}{2}$, which is what was found using Von Neumann analysis.

4.2 Part (b)

The scheme was implemented. The source code is in the appendix of this problem. The grid used is the standard grid

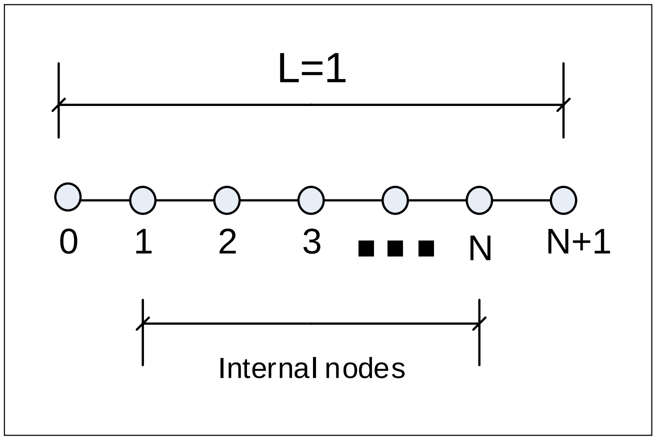


Figure 32: Grid used

In the following, I will be use the following PDE $cu_t + au_x = bu_{xx}$ (where$c$ was added a parameter for the advection term).

Periodic boundary conditions implies $u(0,t) = u(1,t)$, Hence there is an extra one unknown (in addition to the internal nodes). Either $u(0,t)$ or $u(1,t)$ can be selected since they have the same value. When selecting the right end node, then u_{N+1} becomes an unknown to be added to the internal nodes. Using the following diagram

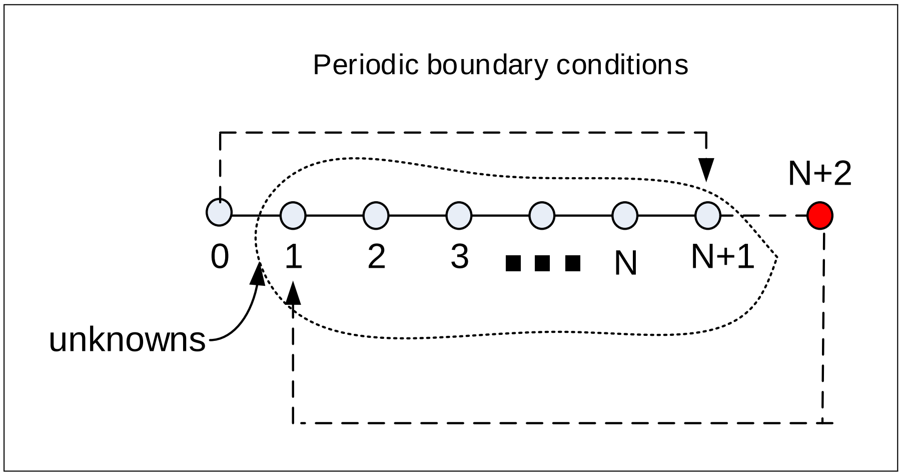


Figure 33: Grid used

The forward time, centered space discretization is

$$c\frac{u_j^{n+1}-u_j^n}{\Delta t}+a\frac{u_{j+1}^n-u_{j-1}^n}{2h}=b\frac{u_{j-1}^n-2u_j^n+u_{j+1}^n}{h^2} \tag{1}$$

Therefore, for nodes $2\cdots N$, the finite difference scheme is

$$\begin{aligned}c\frac{u_j^{n+1}-u_j^n}{\Delta t}+a\frac{u_{j+1}^n-u_{j-1}^n}{2h}&=b\frac{u_{j-1}^n-2u_j^n+u_{j+1}^n}{h^2}\\ u_j^{n+1}&=u_j^n-a\Delta t\frac{u_{j+1}^n-u_{j-1}^n}{2ch}+b\Delta t\frac{u_{j-1}^n-2u_j^n+u_{j+1}^n}{ch^2}\\ u_j^{n+1}&=u_j^n-\frac{a\Delta t}{2ch}\left(u_{j+1}^n-u_{j-1}^n\right)+\frac{b\Delta t}{ch^2}\left(u_{j-1}^n-2u_j^n+u_{j+1}^n\right)\\ u_j^{n+1}&=u_{j-1}^n\left(\frac{a\Delta t}{2ch}+\frac{b\Delta t}{ch^2}\right)+u_j^n\left(1-2\frac{b\Delta t}{ch^2}\right)+u_{j+1}^n\left(-\frac{a\Delta t}{2ch}+\frac{b\Delta t}{ch^2}\right)\end{aligned}$$

Let $v=\frac{a\Delta t}{ch}$, $\mu=\frac{b\Delta t}{ch^2}$, the above becomes

$$u_j^{n+1}=u_{j-1}^n\left(\frac{v}{2}+\mu\right)+u_j^n\left(1-2\mu\right)+u_{j+1}^n\left(\mu-\frac{v}{2}\right)$$

Node $j=1$ gives

$$\begin{aligned}u_1^{n+1}&=u_0^n\left(\frac{v}{2}+\mu\right)+u_1^n\left(1-2\mu\right)+u_2^n\left(\mu-\frac{v}{2}\right)\\ &=u_{N+1}^n\left(\frac{v}{2}+\mu\right)+u_1^n\left(1-2\mu\right)+u_2^n\left(\mu-\frac{v}{2}\right)\end{aligned}$$

And for node $j=N+1$

$$\begin{aligned}u_{N+1}^{n+1}&=u_N^n\left(\frac{v}{2}+\mu\right)+u_{N+1}^n\left(1-2\mu\right)+u_{N+2}^n\left(\mu-\frac{v}{2}\right)\\ &=u_N^n\left(\frac{v}{2}+\mu\right)+u_{N+1}^n\left(1-2\mu\right)+u_1^n\left(\mu-\frac{v}{2}\right)\end{aligned}$$

The full system can now be written in matrix form

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ u_N^{n+1} \\ u_{N+1}^{n+1} \end{bmatrix} = \begin{bmatrix} \left(1-2\mu\right) & \left(\mu-\frac{v}{2}\right) & 0 & 0 & 0 & \left(\frac{v}{2}+\mu\right) \\ \left(\mu+\frac{v}{2}\right) & \left(1-2\mu\right) & \left(\mu-\frac{v}{2}\right) & 0 & 0 & 0 \\ 0 & \left(\mu+\frac{v}{2}\right) & \left(1-2\mu\right) & \left(\mu-\frac{v}{2}\right) & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \left(\mu+\frac{v}{2}\right) & \left(1-2\mu\right) & \left(\mu-\frac{v}{2}\right) \\ \left(\mu-\frac{v}{2}\right) & 0 & 0 & 0 & \left(\mu+\frac{v}{2}\right) & \left(1-2\mu\right) \end{bmatrix} \begin{bmatrix} u_1^{n} \\ u_2^{n} \\ u_3^{n} \\ \vdots \\ u_N^{n} \\ u_{N+1}^{n} \end{bmatrix}$$

The above can also be written as

$$\begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ u_N^{n+1} \\ u_{N+1}^{n+1} \end{bmatrix} = \overbrace{\left(I - \frac{a\Delta t}{2ch} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & -1 & \ddots & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} + \frac{b\Delta t}{ch^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 1 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 1 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \right)}^{\textit{update matrix}} \begin{bmatrix} u_1^{n} \\ u_2^{n} \\ u_3^{n} \\ \vdots \\ u_N^{n} \\ u_{N+1}^{n} \end{bmatrix}$$

u_j^0 are taken from initial conditions. The above is in the form

$$u^{n+1} = Bu^n$$

and is implemented directly as above in the code. Using the numerical values given in the problem

$$\Delta t = \frac{0.25h^2}{b} = 0.25\frac{(0.05)^2}{1} = 0.000625$$

and

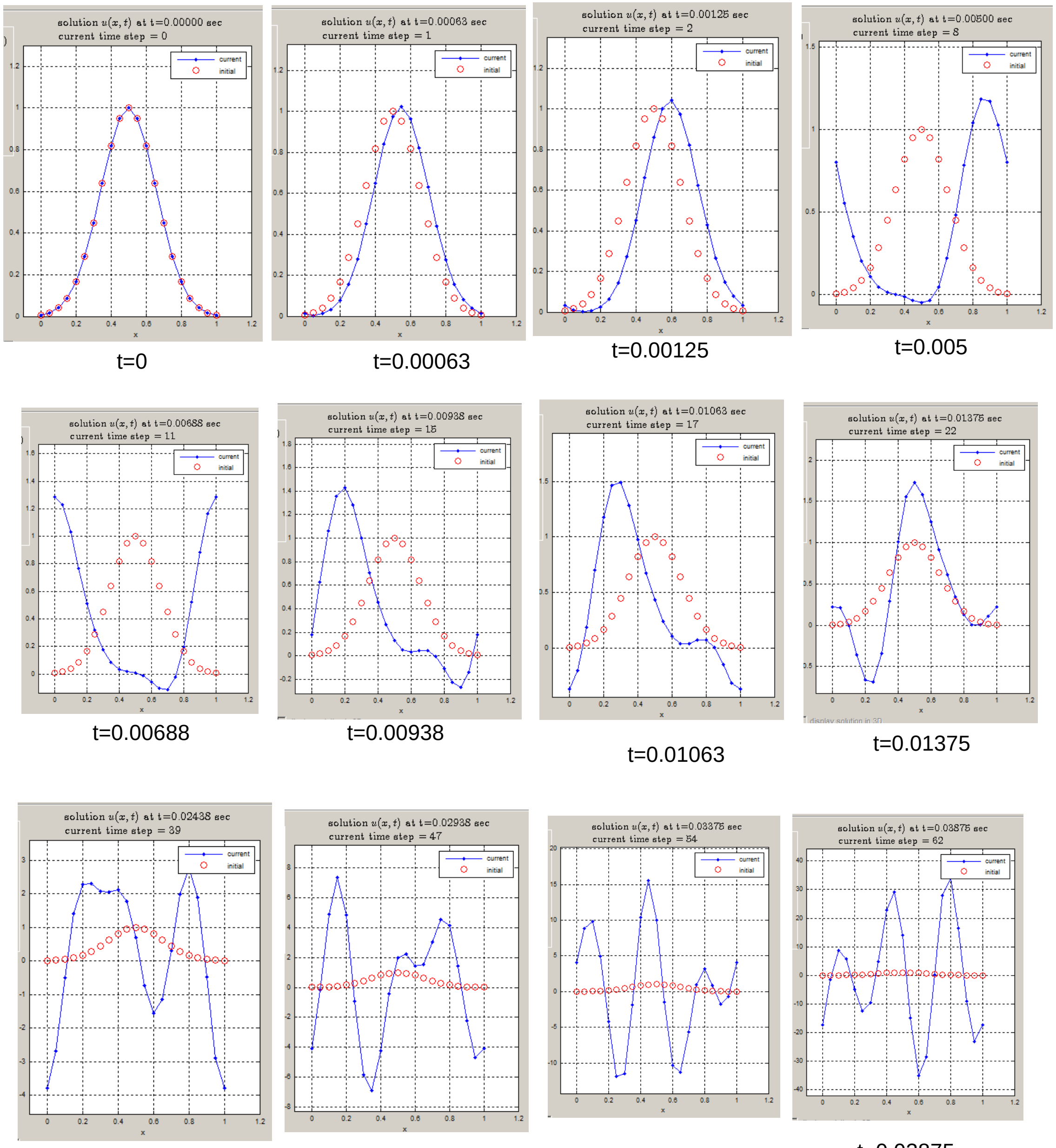
$$v = \frac{a\Delta t}{h} = \frac{80(0.000625)}{0.05} = 1$$

and

$$\mu = \frac{b\Delta t}{h^2} = \frac{0.000625}{(0.05)^2} = 0.25$$

since $\mu \leq \frac{1}{2}$ the solution is expected to be stable since this is the condition derived in part (a).

The following is the result of running the program. The numerical solution grew with time. Here are few snap shots taken at increasing time steps showing the problem. After only about 10 time steps, the numerical solution can be seen to grow more than the initial conditions



t=0 t=0.00063 t=0.00125 t=0.005

t=0.00688 t=0.00938 t=0.01063 t=0.01375

t=0.02438 t=0.02938 t=0.03375 t=0.03875

Figure 34: few snap shots taken at increasing time steps showing the problem

The stability analysis that was done did not predict this based on the value of μ which was $\leq \frac{1}{2}$.

4.3 Part(c)

The eigenvalues u_p of the update matrix derived in part (b) are

$$u_p = 1 - i\frac{a\Delta t}{h}\sin\left(\pi ph\right) + \frac{2b\Delta t}{h^2}\left(\cos\left(\pi ph\right) - 1\right)$$

But $v = \frac{a\Delta t}{h}$, $\mu = \frac{b\Delta t}{h^2}$, then the above becomes

$$u_p = 1 - iv\sin\left(\pi ph\right) + 2\mu\left(\cos\left(\pi ph\right) - 1\right)$$

Let $\sin^2\left(\frac{\xi h}{2}\right) \equiv \omega$, then the above can be written as

$$\left|u_p\right|^2 = \left(1 - 4\mu\omega\right)^2 + 4\nu^2\omega(1-\omega)$$

As was done in part (a).

From part (a), it was found that when $\omega = 1$, this resulted in the condition of stability being $0 \le \mu \le \frac{1}{2}$, and when $\omega = 0$, the maximum eigenvalue is 1. To find the condition of $\left|u_p\right| \le 1$ for the full range of ω, first expand $\left|u_p\right|^2$ into a quadratic in ω and minimize

$$\begin{aligned}\left|u_p\right|^2 &= 1 + 16\mu^2\omega^2 - 8\mu\omega + 4\nu^2\omega - 4\nu^2\omega^2\\ &= 1 - 4\omega\left(2\mu - \nu^2\right) + 4\omega^2\left(4\mu^2 - \nu^2\right)\end{aligned}$$

Since $\omega = \sin^2\left(\frac{\pi ph}{2}\right)$, then ω values are from $0\cdots 1$. Since the maximum eigenvalue occurs when $\omega = 0$, then $\omega = 0$ is the maximum point of the quadratic $1-4\omega\left(2\mu - \nu^2\right)+4\omega^2\left(4\mu^2 - \nu^2\right)$, hence the slope of this quadratic at $\omega = 0$ must be negative. But the slope is

$$\begin{aligned}\frac{d}{d\omega}\left|u_p\right|^2 &= \frac{d}{d\omega}\left(1 - 4\omega\left(2\mu - \nu^2\right) + 4\omega^2\left(4\mu^2 - \nu^2\right)\right)\\ &= -4\left(2\mu - \nu^2\right) + 8\omega\left(4\mu^2 - \nu^2\right)\end{aligned}$$

For the above to be negative (so that the eigenvalue remain below 1) implies that $2\mu - \nu^2$ must be positive, i.e.

$$2\mu - \nu^2 \ge 0$$

or

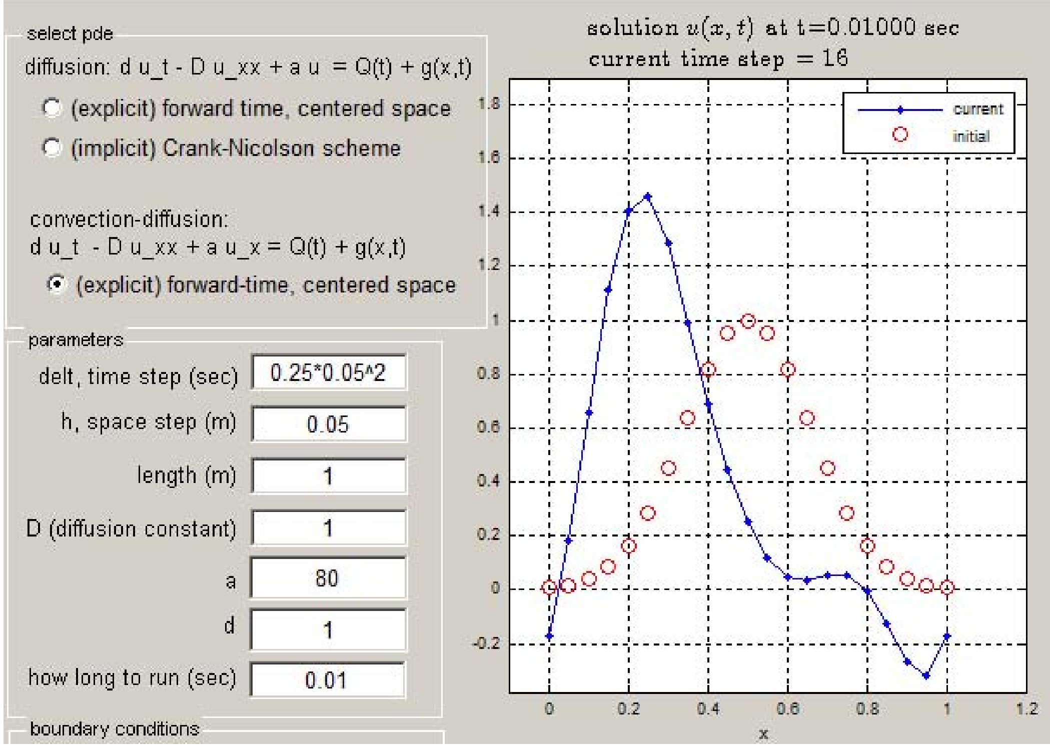
$$\nu^2 \le 2\mu$$

And since from part(a) it was found that $\mu \le \frac{1}{2}$, or $2\mu \le 1$ then the above become

$$\nu^2 \le 2\mu \le 1$$

4.4 Part(d)

The solution for $\Delta t = 0.000625$, $h = 0.05$, $a = 80$, $b = 1$, with $u(x,0) = \exp\left(-20(x-0.5)^2\right)$ at $t = 0.01$ seconds is



select pde

diffusion: d u_t - D u_xx + a u = Q(t) + g(x,t)

○ (explicit) forward time, centered space

○ (implicit) Crank-Nicolson scheme

convection-diffusion:
d u_t - D u_xx + a u_x = Q(t) + g(x,t)

● (explicit) forward-time, centered space

parameters

delt, time step (sec): 0.25*0.05^2

h, space step (m): 0.05

length (m): 1

D (diffusion constant): 1

a: 80

d: 1

how long to run (sec): 0.01

boundary conditions

solution $u(x,t)$ at t=0.01000 sec
current time step = 16

Figure 35: Part c final solution

The value of the numerical solution at that time is

```
-0.1727
 0.1817
 0.6603
 1.1158
 1.4092
 1.4609
 1.2868
 0.9910
 0.6912
 0.4441
 0.2527
 0.1168
 0.0451
 0.0337
 0.0532
```

```
  0.0563
 -0.0013
 -0.1275
 -0.2680
 -0.3162
 -0.1727
```

4.5 Appendix for problem 4

derivation of the convection-diffusion using general terms The PDE is $cu_t + au_x = bu_{xx}$, with $b > 0$, the forward time, centered space discretization is

$$c\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_{j+1}^n - u_{j-1}^n}{2h} = b\frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{h^2} \tag{1}$$

Therefore, for nodes $2 \cdots N$, the finite difference scheme is

$$\begin{aligned}
c\frac{u_j^{n+1} - u_j^n}{\Delta t} + a\frac{u_{j+1}^n - u_{j-1}^n}{2h} &= b\frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{h^2} \\
u_j^{n+1} &= u_j^n - a\Delta t\frac{u_{j+1}^n - u_{j-1}^n}{2ch} + b\Delta t\frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{ch^2} \\
u_j^{n+1} &= u_j^n - \frac{a\Delta t}{2ch}\left(u_{j+1}^n - u_{j-1}^n\right) + \frac{b\Delta t}{ch^2}\left(u_{j-1}^n - 2u_j^n + u_{j+1}^n\right) \\
u_j^{n+1} &= u_{j-1}^n\left(\frac{a\Delta t}{2ch} + \frac{b\Delta t}{ch^2}\right) + u_j^n\left(1 - 2\frac{b\Delta t}{ch^2}\right) + u_{j+1}^n\left(-\frac{a\Delta t}{2ch} + \frac{b\Delta t}{ch^2}\right)
\end{aligned}$$

Let $v = \frac{a\Delta t}{ch}$, $\mu = \frac{b\Delta t}{ch^2}$, the above becomes

$$u_j^{n+1} = u_{j-1}^n\left(\frac{v}{2} + \mu\right) + u_j^n\left(1 - 2\mu\right) + u_{j+1}^n\left(\mu - \frac{v}{2}\right)$$

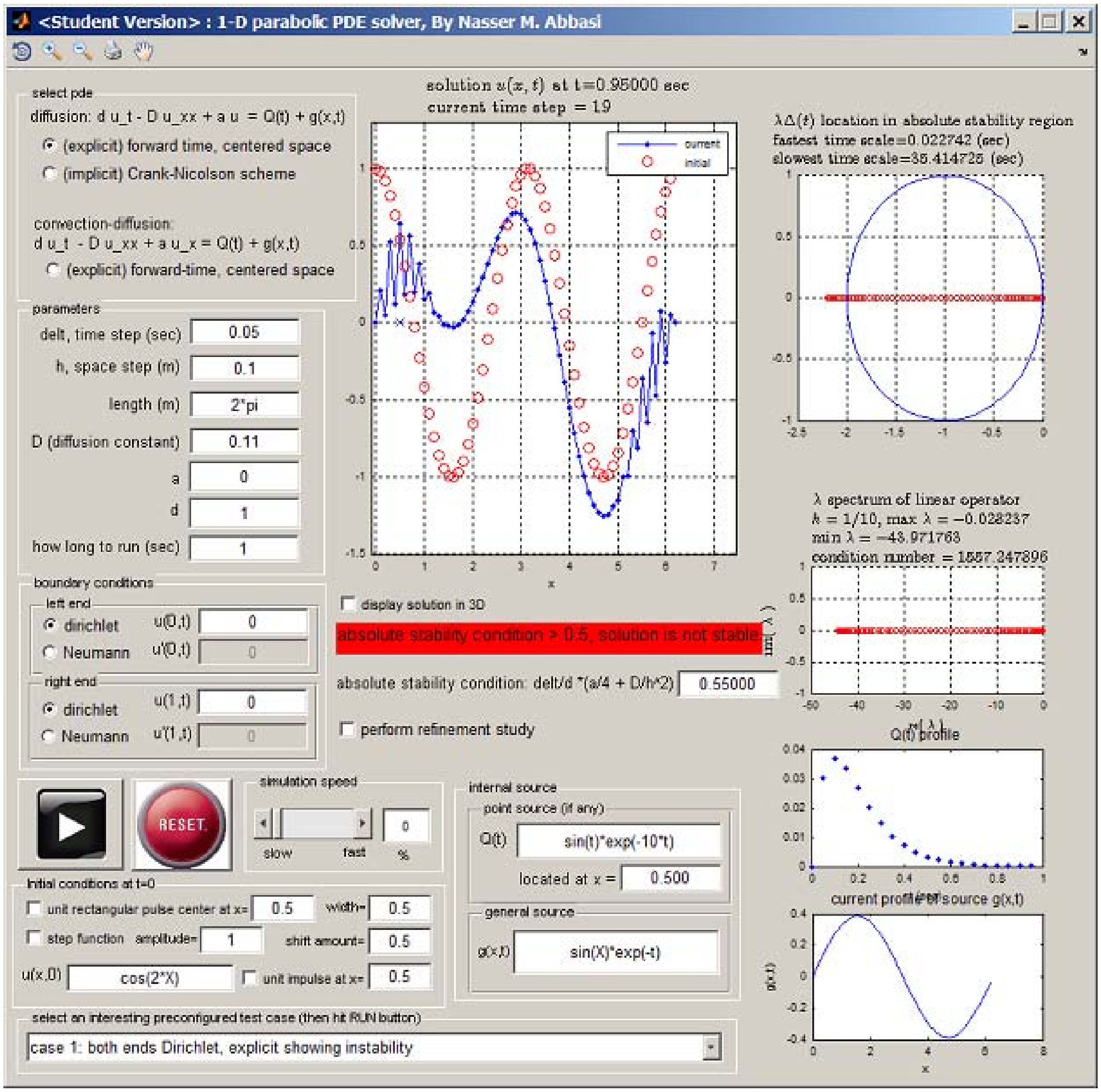
Node $j = 1$ gives

$$\begin{aligned}
u_1^{n+1} &= u_0^n\left(\frac{v}{2} + \mu\right) + u_1^n\left(1 - 2\mu\right) + u_2^n\left(\mu - \frac{v}{2}\right) \\
&= u_{N+1}^n\left(\frac{v}{2} + \mu\right) + u_1^n\left(1 - 2\mu\right) + u_2^n\left(\mu - \frac{v}{2}\right)
\end{aligned}$$

And for node $j = N + 1$

$$\begin{aligned}
u_{N+1}^{n+1} &= u_N^n\left(\frac{v}{2} + \mu\right) + u_{N+1}^n\left(1 - 2\mu\right) + u_{N+2}^n\left(\mu - \frac{v}{2}\right) \\
&= u_N^n\left(\frac{v}{2} + \mu\right) + u_{N+1}^n\left(1 - 2\mu\right) + u_1^n\left(\mu - \frac{v}{2}\right)
\end{aligned}$$

5 Screen shot of the GUI matlab application used for HW1



<Student Version> : 1-D parabolic PDE solver, By Nasser M. Abbasi

select pde

diffusion: d u_t - D u_xx + a u = Q(t) + g(x,t)

● (explicit) forward time, centered space

○ (implicit) Crank-Nicolson scheme

convection-diffusion:
d u_t - D u_xx + a u_x = Q(t) + g(x,t)

○ (explicit) forward-time, centered space

parameters

delt, time step (sec): 0.05

h, space step (m): 0.1

length (m): 2*pi

D (diffusion constant): 0.11

a: 0

d: 1

how long to run (sec): 1

boundary conditions

left end

● dirichlet u(0,t): 0

○ Neumann u'(0,t): 0

right end

● dirichlet u(1,t): 0

○ Neumann u'(1,t): 0

simulation speed: 0 %

Initial conditions at t=0

□ unit rectangular pulse center at x= 0.5 width= 0.5

□ step function amplitude= 1 shift amount= 0.5

u(x,0): cos(2*X) □ unit impulse at x= 0.5

select an interesting preconfigured test case (then hit RUN button)

case 1: both ends Dirichlet, explicit showing instability

□ display solution in 3D

absolute stability condition > 0.5, solution is not stable

absolute stability condition: delt/d *(a/4 + D/h^2): 0.55000

□ perform refinement study

internal source

point source (if any)

Q(t): sin(t)*exp(-10*t)

located at x = 0.500

general source

g(x,t): sin(X)*exp(-t)

solution $u(x,t)$ at t=0.95000 sec
current time step = 19

$\lambda\Delta(t)$ location in absolute stability region
fastest time scale=0.022742 (sec)
slowest time scale=35.414725 (sec)

λ spectrum of linear operator
$h = 1/10$, max $\lambda = -0.028237$
min $\lambda = -43.971763$
condition number = 1557.247896

Figure 36: Matlab program I developed for this HW

6 Matlab Source code developed for this HW

6.1 nma_math228b_build_HW1.m

```
function nma_math228b_build_HW1()

list = dir('*.m');

if isempty(list)
    fprintf('no matlab files found\n');
    return
end

for i=1:length(list)
    name=list(i).name;
    fprintf('processing %s\n',name)
    p0 = fdep(list(i).name,'-q');
    [pathstr, name_of_matlab_function, ext] = fileparts(name);

    %make a zip file of the m file and any of its dependency
    p1=dir([name_of_matlab_function '.fig']);
    if length(p1)==1
        files_to_zip =[p1(1).name;p0.fun];
    else
        files_to_zip =p0.fun;
    end

    zip([name_of_matlab_function '.zip'],files_to_zip)

end

end
```

6.2 nma_math228b_HW1.m

```
function nma_math_228b_HW1

t=0:0.1:100;
x=0.01:0.01:1;
sol=zeros(length(x),1);

for i=1:length(t)
   for j=1:length(x)
      sol(j)=  uh(x(j),t(i)) + up(x(j))   ;
   end
   plot(x,sol);
   title(sprintf('t=%f',t(i)));
   drawnow();
   pause(.01);

end
end


%--------------------------
function v=up(x)

v= 50* x * (1-x) ;
end
%-----------------------
function v=uh(x,t)
a=0.01;
sum=0; N=50;

for n=1:N
   sum=sum+( 400* an(n)*exp( -a * (n*pi)^2 *t ) * sin(n*pi*x) );
end
v=sum;
end

%-----------------------
function v=an(n)
v=( (-1)^n-1)/(n^3*pi^3);
end
```

6.3 nma_math228b_plot.m

```
function nma_math228b_plot()
close all;

t=0.01;
D=1;

N=49;
h=1/(N+1);
p=1:N;
z=p*pi*(1/(N+1));
g= (D*t/h^2)*(cos(z)-1);
g=abs((1+g)./(1-g));


plot(p,g)
title(sprintf('Magnification factor as function of wave number\nC-N scheme for h=%f, t=0.01,
xlabel('p, wave number');
```

```
ylabel('g(wave number)');
%


close all;
h=0.02;
t=0.001;
D=1;
r=t*D/(2*h^2);
z=-pi/h:0.01:pi/h;
g= abs((1+2*r*cos(z*h)-2*r)./(1-2*r*cos(z*h)+2*r));

plot(z*h,g)
title(sprintf('Magnification factor, C-N scheme for h=0.02, t=0.001'))
xlabel('zeta(radians)');
ylabel('g(zeta)');
xlim([-pi,pi])
ylim([0,1.2]);
set(gca,'XTick',-pi:pi/2:pi)
set(gca,'XTickLabel',{'-pi/h','-pi/2h','0','pi/2h','pi/h'})


close all;
h=0.02;
t=0.001;
D=1;
r=t*D/(2*h^2);
z=-pi/h:0.01:pi/h;
g= abs(1-(4*D*t/h^2)*sin(z*h/2).^2);

plot(z*h,g)
title(sprintf('Magnification factor, FE scheme for h=0.02, t=0.001\ndelt*D/h^2=%3.3f',t*D/h^2
xlabel('zeta(radians)');
ylabel('g(zeta)');
xlim([-pi,pi])
ylim([0,1.2]);
set(gca,'XTick',-pi:pi/2:pi)
%set(gca,'XTickLabel',{'-pi/h','-pi/2h','0','pi/2h','pi/h'})

end
```