Contents

NMTFD1 Practical 1: Group 10

Isa Mammadli
Onkar Marathe

November 21, 2022

1 Problem introduction

For the dimensionless 2D heat equation:

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \tag{1}$$

$$[x, y] \in [0, 1] \times [0, 1], t \in [0, 0.16] \tag{2}$$

Initial and boundary conditions are:

$$w(x, y, 0) = 0, \tag{3}$$

$$\begin{aligned} &w(0, y, t) = 1 - y^3, \;\; w(1, y, t) = 1 - \sin\left(\frac{\pi}{2} y\right), \\ &w(x, 0, t) = 1, \;\; w(x, 1, t) = 0 \end{aligned} \tag{4}$$

2 Solutions: Discretization and Numerical Schemes

- **Discretize the equation by CDS scheme in space and the Crank-Nicolson method in time:**
  To begin with, the forward *time* step for w will be:

  $$\frac{\partial w}{\partial t} = \frac{w_{i,j}^{n+1} - w_{i,j}^{n}}{\Delta t} \tag{5}$$

  If we consider i to be vertical (y direction) and j to be horizontal (x direction), then we can write the second order derivative of w with respect to x as follows:

  $$\frac{\partial^2 w}{\partial x^2} = \frac{w_{i,j+1}^{n} - 2w_{i,j}^{n} + w_{i,j-1}^{n}}{\Delta x^2} \tag{6}$$

  Explicit Euler. Crank-Nicolson method becomes more intuitive when we start with deriving explicit Euler's method. Combining (5) and (6) along with second order derivative of w with respect to y-direction.

Table 1: Coefficients for Explicit-Euler scheme.

w	Coefficient for time step: n+1	Coefficients for time step: n	Compass direction
$w_{i,j}$	1	$(1 - 2\frac{\Delta t}{\Delta x^2} - 2\frac{\Delta t}{\Delta y^2})$	P
$w_{i,j+1}$	0	$\frac{\Delta t}{\Delta x^2}$	E
$w_{i,j-1}$	0	$\frac{\Delta t}{\Delta x^2}$	W
$w_{i+1,j}$	0	$\frac{\Delta t}{\Delta y^2}$	N
$w_{i-1,j}$	0	$\frac{\Delta t}{\Delta y^2}$	S

$$\frac{w_{i,j}^{n+1} - w_{i,j}^{n}}{\Delta t} = \frac{w_{i,j+1}^{n} - 2w_{i,j}^{n} + w_{i,j-1}^{n}}{\Delta x^2} + \frac{w_{i+1,j}^{n} - 2w_{i,j}^{n} + w_{i-1,j}^{n}}{\Delta y^2} \tag{7}$$

Equation (7) is called explicit Euler method. Clearly, all the terms on the right are of the time step n. Table 1 illustrates the coefficients for each term. You can see the new (i,j) element at (n+1) is calculated entirely based on values from n-th time step. Now, if w is assumed to be $\mathbb{R}^{m \times n}$ - a temperature field of m by n dimensions (or nodes). Then we can write w as column vector with $\mathbb{R}^{mn}$, while the coefficient matrix A_{eu} will be of $\mathbb{R}^{mn \times mn}$. ($A_{eu}$ here is a penta-diagonal matrix.)

$$w^{n+1} = A_{eu} w^{n} \tag{8}$$

Crank-Nicolson. In case of Crank-Nicolson, we write right hand side with (n+1)-th terms as well as with n-th time terms and then divide by 2 as shown below.

$$\begin{aligned}\frac{w_{i,j}^{n+1} - w_{i,j}^{n}}{\Delta t} = \frac{1}{2}[&\frac{w_{i,j+1}^{n+1} - 2w_{i,j}^{n+1} + w_{i,j-1}^{n+1}}{\Delta x^2} + \frac{w_{i+1,j}^{n+1} - 2w_{i,j}^{n+1} + w_{i-1,j}^{n+1}}{\Delta y^2} + \\ &\frac{w_{i,j+1}^{n} - 2w_{i,j}^{n} + w_{i,j-1}^{n}}{\Delta x^2} + \frac{w_{i+1,j}^{n} - 2w_{i,j}^{n} + w_{i-1,j}^{n}}{\Delta y^2}]\end{aligned} \tag{9}$$

Next step is to simplify (9) by collecting (n+1)-th factors to the left hand side and n-th time step elements to the right hand side. To be concise, this was summarized in Table 2, with corresponding coefficients for each time step.

Table 2: Coefficients for Crank-Nicolson scheme.

w	Coefficient for time step (A): n+1	Coefficients for time step (B): n	Compass direction
$w_{i,j}$	$(1+\frac{\Delta t}{\Delta x^2}+\frac{\Delta t}{\Delta y^2})$	$(1-\frac{\Delta t}{\Delta x^2}-\frac{\Delta t}{\Delta y^2})$	P
$w_{i,j+1}$	$-\frac{\Delta t}{2\Delta x^2}$	$\frac{\Delta t}{2\Delta x^2}$	E
$w_{i,j-1}$	$-\frac{\Delta t}{2\Delta x^2}$	$\frac{\Delta t}{2\Delta x^2}$	W
$w_{i+1,j}$	$-\frac{\Delta t}{2\Delta y^2}$	$\frac{\Delta t}{2\Delta y^2}$	N
$w_{i-1,j}$	$-\frac{\Delta t}{2\Delta y^2}$	$\frac{\Delta t}{2\Delta y^2}$	S

If the coefficients for (n+1)-th and n-th time steps are compiled in A and B matrices respectively, then the Crank-Nicolson can be written as in (10):

$$\begin{aligned} Aw^{n+1} &= Bw^n \\ w^{n+1} &= A^{-1}Bw^n \end{aligned} \tag{10}$$

A and B matrices are also penta-diagonal. Note that in above tables, coefficients are written for all elements (i,j) except for the boundary conditions. In case the w_{ij} represents some boundary conditions, then the coefficient will be 1 for (i,j) or P and 0 for remaining directions (N,S,E,W). In this way, the boundary values are fixed and A, B matrices are built.

3 Stability

- **Show under what conditions the Crank-Nicolson scheme is stable.** Using Equation (9), Von Neumann stability analysis will be conducted here. However, we will start with 1D Crank-Nicolson and then modify it for 2D and draw the conclusion for the stability.

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial x^2}$$

where

$$w = (x, t)$$

With the discretization:

$$\frac{w^{n+1} - w^n}{\Delta t} = \frac{1}{2}\left(\frac{w_{j+1}^{n+1} - 2w_j^{n+1} + w_{j-1}^{n+1}}{\Delta x^2} + \frac{w_{j+1} - 2w_j^n + w_{j-1}^n}{\Delta x^2}\right) \tag{11}$$

$$w(x,t) = G(k)e^{ikx}$$

Where the $G(k) = e^{-k^2 t}$ and called growth factor and k is wave number. For the stability of the numerical scheme we need $|G(k)| \leq 1$ (Reference [2]) In the discretized version.

$$\begin{aligned} w_j^n &= e^{ik\Delta xj} \\ w_j^{n+1} &= Ge^{ik\Delta xj} \end{aligned} \tag{12}$$

where i = complex number, j = grid point in x direction. Substituting (12) into (11), we get:

$$\begin{aligned} \frac{Ge^{ik\Delta xj} - e^{ik\Delta xj}}{\Delta t} = \frac{1}{2}(&\frac{Ge^{ik\Delta x(j+1)} - 2Ge^{ik\Delta xj} + Ge^{ik\Delta x(j-1)}}{\Delta x^2} + \\ &+ \frac{e^{ik\Delta x(j+1)} - 2e^{ik\Delta xj} + e^{ik\Delta x(j-beg1)}}{\Delta x^2}) \end{aligned} \tag{13}$$

Dividing both sides with $e^{ik\Delta xj}$, replacing $\frac{\Delta t}{\Delta x^2} = r$ and solving for G, we end up with:

$$G = \frac{1 - r(1 - \cos(k\Delta x))}{1 + r(1 - \cos(k\Delta x))} \tag{14}$$

Since absolute value of G as per (15) is always less or equal to 1, we can say Crank-Nicolson in 1D is **unconditionally stable**.

In terms of 2D heat equation with Crank-Nicolson scheme, Equation (12) will be modified with $w_j^n = e^{ik\Delta xj} \cdot e^{il\Delta yp}$. (Note that in the y direction, $y = i\Delta y$, but 'i' was replaced with p, to prevent confusion with complex number.) Assuming $\Delta x = \Delta y$, in 2D G(k) becomes as follows:

$$G = \frac{1 - r(2 - \cos(k\Delta x) - \cos(l\Delta x))}{1 + r(2 - \cos(k\Delta x) - \cos(l\Delta x))} \tag{15}$$

Thus, the same reasoning works for 2D, as well. Crank-Nicolson is **unconditionally stable**, hence Δt is not bounded for stability. However, this does not necessarily mean, we can take arbitrarily large time steps. Numerical accuracy and also for avoiding oscillations smaller time steps will be needed. As noted by Ferziger et al. ([1]), $\Delta t \leq \Delta x^2$ can be taken for guaranteed results, but larger time steps are possible and it is problem dependent. In our case, we see at higher time steps we can find non-oscillatory solutions.

4 Convergence

- **Show that the numerical results will converge when the grid is refined.** We know,

$$
\begin{aligned}
\frac{\partial w}{\partial x} &= \frac{\Delta w}{\Delta x} + T \\
T &= \frac{\partial w}{\partial x} - \frac{\Delta w}{\Delta x}
\end{aligned}
\tag{16}
$$

Where T is truncation error.

$$\lim_{\Delta x \to 0}(T) = \lim_{\Delta x \to 0}\left(\frac{\partial w}{\partial x} - \frac{\Delta w}{\Delta x}\right) = 0$$

Similarly,

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}$$

$$\frac{\partial w}{\partial t} - \frac{\partial^2 w}{\partial x^2} - \frac{\partial^2 w}{\partial y^2} = 0$$

i.e.

$$\frac{\partial w}{\partial t} - \frac{\partial^2 w}{\partial x^2} - \frac{\partial^2 w}{\partial y^2} = \frac{\Delta w}{\Delta t} - \frac{\Delta^2 w}{\Delta x^2} - \frac{\Delta^2 w}{\Delta y^2} + T$$

Here T combines all the truncation errors for x, y and t.

$$T = \left(\frac{\partial w}{\partial t} - \frac{\Delta w}{\Delta t}\right) + \left(\frac{\partial^2 w}{\partial x^2} - \frac{\Delta^2 w}{\Delta x^2}\right) + \left(\frac{\partial^2 w}{\partial y^2} - \frac{\Delta^2 w}{\Delta y^2}\right)$$

$$\lim_{\Delta t, \Delta x, \Delta y \to 0}(T) = \lim_{\Delta t \to 0}\left(\frac{\partial w}{\partial t} - \frac{\Delta w}{\Delta t}\right) + \lim_{\Delta x \to 0}\left(\frac{\partial^2 w}{\partial x^2} - \frac{\Delta^2 w}{\Delta x^2}\right) + \lim_{\Delta y \to 0}\left(\frac{\partial^2 w}{\partial y^2} - \frac{\Delta^2 w}{\Delta y^2}\right) \tag{17}$$

If we replace expressions in the brackets with Taylor expansions (Central difference in x and y directions):

$$\left(\frac{\partial w^2}{\partial x^2} - \frac{\Delta^2 w}{\Delta x^2}\right) = -\frac{\partial^4 w}{\partial x^4}\frac{\Delta x^2}{12} + HOT \tag{18}$$

Apply the same for Δy. In case of time, since it is first order derivative, applying forward difference:

$$\left(\frac{\partial w}{\partial t} - \frac{\Delta w}{\Delta t}\right) = -\frac{\partial^2 w}{\partial t^2}\frac{\Delta t}{2} - \frac{\partial^3 w}{\partial t^3}\frac{\Delta t^2}{6} - HOT \tag{19}$$

Substituting Equations (18) (for both x and y directions) and (19), we clearly see all the expressions in the brackets individually approach 0, as we choose arbitrarily smaller grid sizes and time steps. Hence, truncation error T approaches 0 and the numerical results converges. A numerical experiment was carried for this purpose, with Crank-Nicolson scheme, time step = 0.001 and number of grids were varied from 4 to 80, corresponding to grid sizes of 0.25 to 0.0125 units. Results are as given below. (In the plot 'h' represents the number of grids in x and y directions (equal for both) and 'dh' represents size of individual grids.)

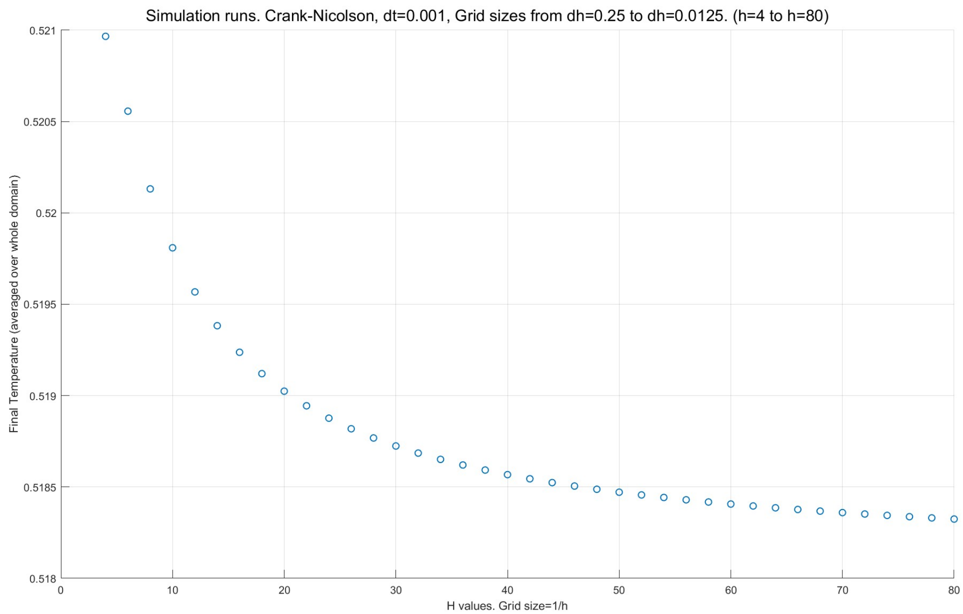

Figure 1: Grid size sensitivity and convergence

5 MATLAB codes/functions

- **MATLAB Program for Crank-Nicolson and Explicit Euler schemes.** Codes shall be explained during the presentation. As introduction, there are multiple functions and files that use those functions to build matrices, plot and run through time loop.
  - **Heat_Diffusion_2D_Euler** and **Heat_Diffusion_2D_CrankNicolson** are the main codes that execute time marching and plot the results as requested in the task sheet.
  - **CN_grid_sensitivity** is the numerical experiment that runs Crank Nicolson scheme for various grid sizes to confirm convergence.

- **construct_EU_B (for Explcit Euler) constructA** and **constructB (for CN)**, construct the coefficient matrices. They also use secondary functions like **ToVector, ToMatrix** to go back and forth between vector and matrix forms
- **plot_mesh** is also a secondary function that reduces the number of lines during plotting via surf() built-in function. It is capable of increasing mesh size 1 unit for surf() routine, to reflect boundary conditions with color map.

6 Discussion of results

This part should be read along with MATLAB codes and outputs.

- **Use h = 1/40 in x and y directions and t with different values: 0.01, 0.001 and 0.0001.**
  - for Explicit Euler stability conditions are $\Delta t \leq \frac{1}{8}(\Delta x^2 + \Delta y^2)$, resulting with $\Delta t \leq 1.5625 \cdot 10^{-4}$. Therefore, out of the listed time steps, 0.0001 will be stable. [1]
  - for Crank Nicolson, there is no bound in terms of stability as explained above and for the accuracy 0.001 is sufficient.
- **Two plots for ExEu and CN time evolution at x=y=0.4**. Since the heat sources are from bottom, left and right one can see from the results that due to high initial temperature gradient, temperature at this node increases very fast, but later stabilizes along with average temperature of whole domain. At the end lower temperature gradients causes lower heat transfer, hence getting closer to steady state. (Figures 2,3)

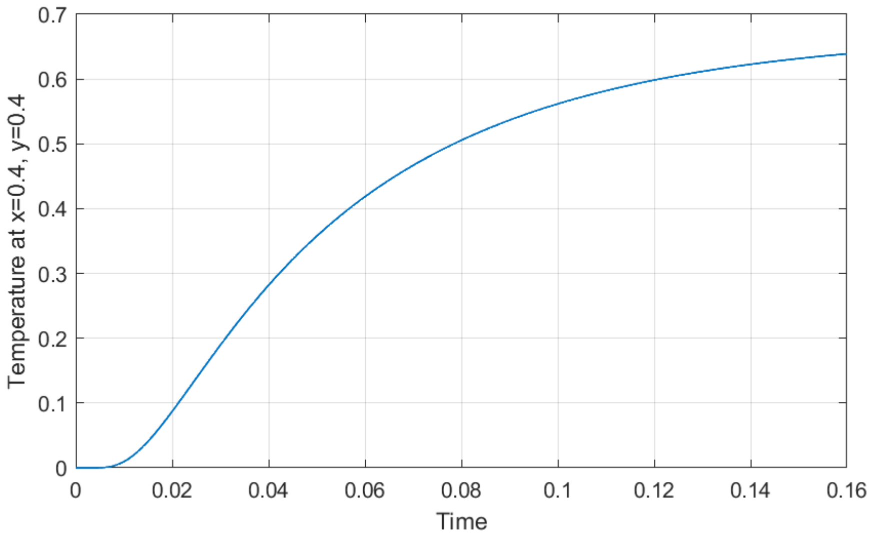

Figure 2: Explicit Euler with (41x41) grid, dt=1e-4

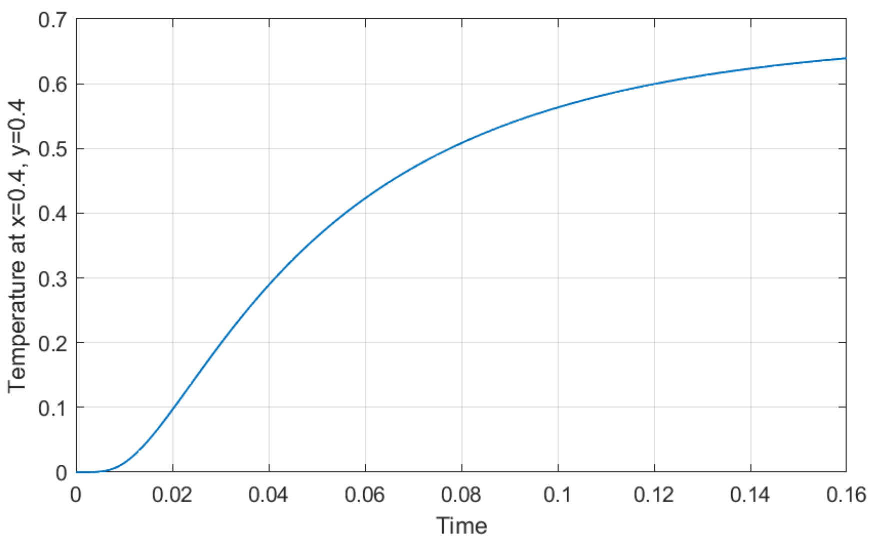

Figure 3: Crank Nicolson with (41x41) grid and dt=1e-3

- **Two plots for vertical profile in ExEu and CN at t=0.16, x=0.4**. Similarly, since the temperatures are fixed at top with zero and bottom with 1, one can see almost linear temperature gradient forms vertically at the end. At the final time step vertical profile at each successive x heavily depends on the proximity to left or right boundary conditions, kind of distance weighted average of two. (Figures 4,5)

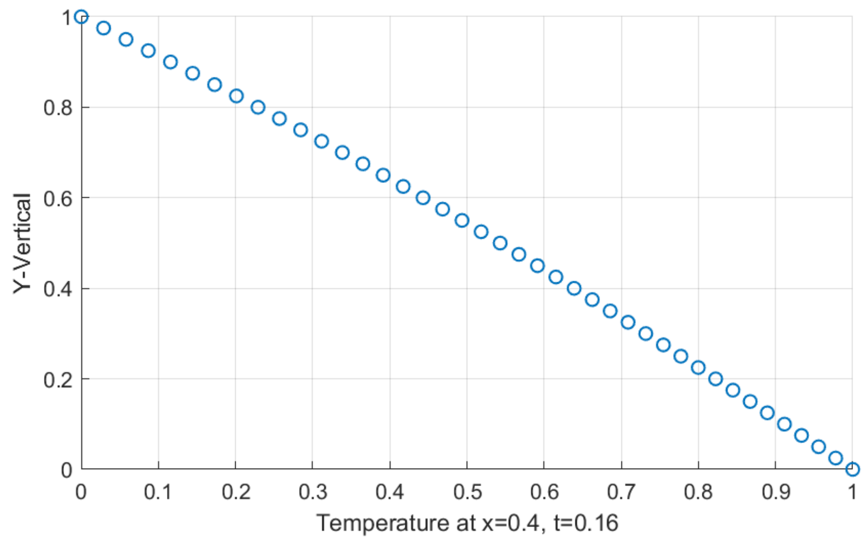

Figure 4: Explicit Euler with (41x41) grid, dt=1e-4

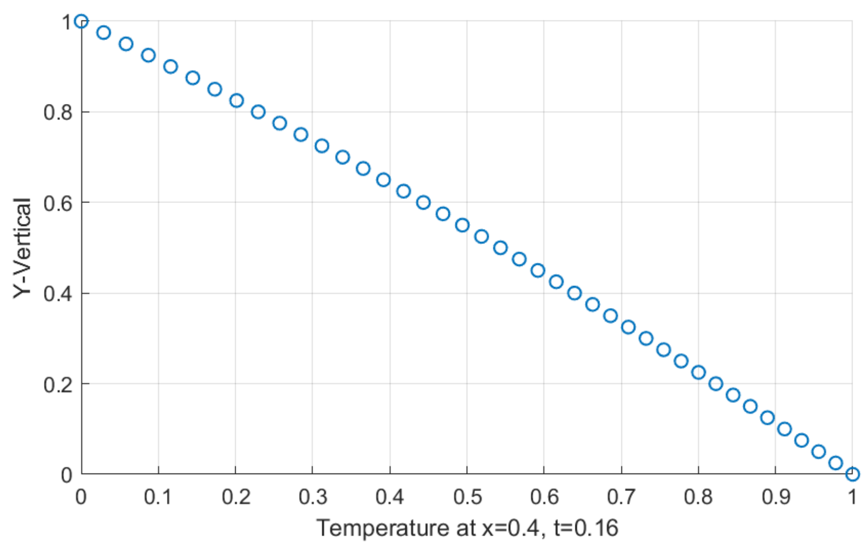

Figure 5: Crank Nicolson with (41x41) grid and dt=1e-3

- **Performance of two methods** From the multiple runs considered in MATLAB, with $\Delta t = 0.0001$ there is not much difference in the run time of Crank Nicolson and Explicit Euler schemes, both providing results in

2.6-2.8 seconds (this time includes building necessary matrices and time marching in both schemes). To better distinguish performance of two methods, series of runs can be made (e.g. 250 runs) and statistical distribution (mean, median) can be compared. However, one evident advantage of Crank Nicolson scheme is that it can converge with 0.001 time stepping in approximately 0.5 seconds, at which time step Explicit Euler is unstable.

- **Explain the physical phenomenon.** As noted above, overall temperature evolution over time is driven by boundary conditions and resulting temperature gradient in the x and y directions. Imposed boundary conditions cause heat to radiate from left, bottom and right very fast, forming 'U' shaped temperature field initially. Having higher temperatures on the left-top side than right-top causes field to reshape into 'L' view. At some point temperature in middle part of x direction reaches right boundary conditions, when a linear-like temperature profile emerges from from left to right. Fixed boundary conditions at the top prevents further heat transfer and temperature fields get closer to stabilization.

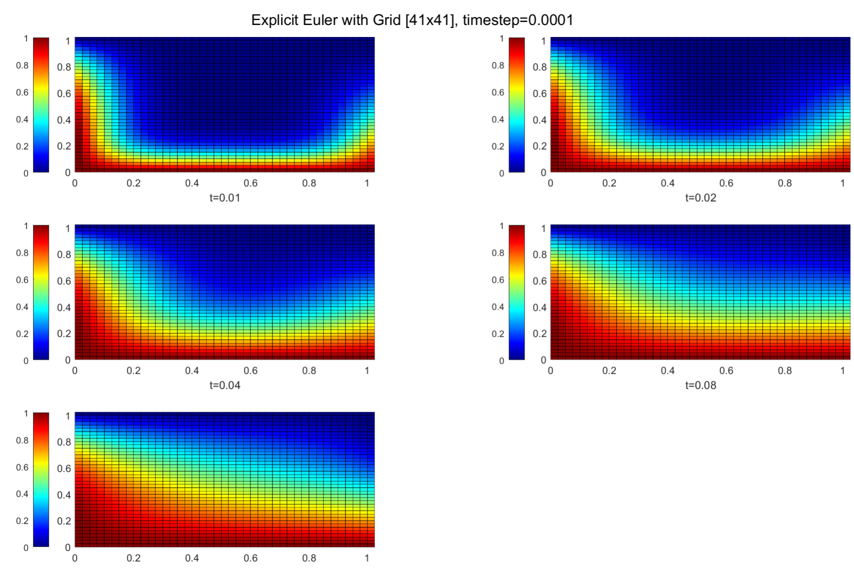

Figure 6: Explicit Euler with (41x41) grid, dt=1e-4

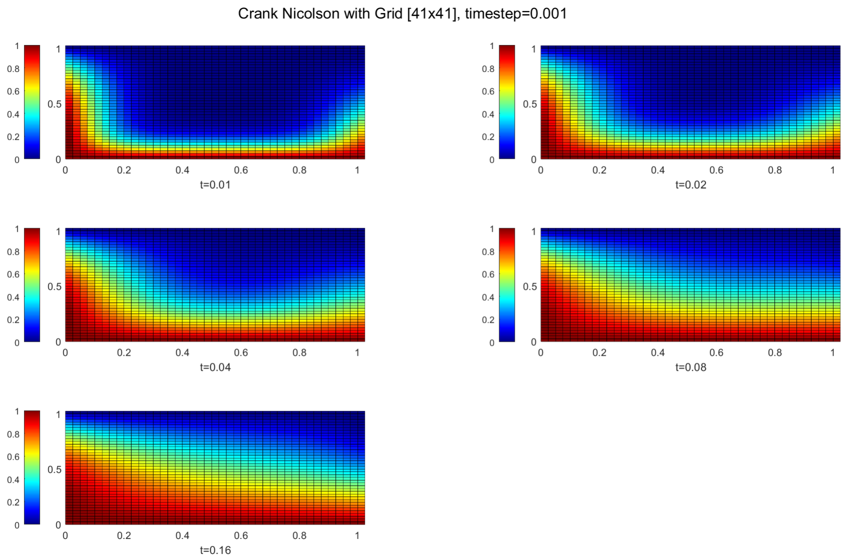

Figure 7: Crank Nicolson with (41x41) grid and dt=1e-3

References

[1] J.H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer Berlin Heidelberg, 2012.

[2] MIT OpenCourseWare. *Von Neumann Stability Analysis*. MIT OpenCourseWare, 2009.