A comparison of finite difference methods for solving the one- and two-dimensional diffusion equation

Erik Alexander Sandvik

December 19, 2019

Abstract

The one-dimensional diffusion equation is solved in using the Forward Euler, Backward Euler and Crank-Nicolson finite difference schemes. In addition, the two-dimensional diffusion equation is solved using the two-dimensional Forward Euler scheme. Analytical expressions are derived so that we can compare the finite difference schemes in the errors. We observe that the error in the Forward Euler, Backward Euler and Crank-Nicolson schemes are nearly identical at $\Delta x = 0.01,\ \alpha = 1/2$. We argue that our results suggest that there exists a lower bound in the error when the solution is changing with time. Von Neumann stability analysis is performed on all of the finite difference schemes. Our findings support the analytically derived stability criterion for the two-dimensional Forward Euler scheme. The lowest value of the alpha-parameter where the numerical results where observed to blow up was $\alpha = 0.267$.

1 Introduction

In this report we will solve the one- and two-dimensional diffusion equation using various finite difference schemes. In one dimension we use the Forward Euler, Backward Euler and Crank-Nicolson schemes. We solve the one-dimensional diffusion equation analytically to obtain an expression in which we can use to find the error in these finite difference schemes. We can then use the error to compare the finite difference schemes with each other. The boundary conditions and the initial condition are chosen so that we obtain a solution that is significantly curved and changing rapidly initially while becoming nearly straight and changing slowly at later times. This is to see how the finite difference schemes compare against each other under different circumstances. For the two-dimensional diffusion equation we use the two-dimensional Forward Euler scheme. Also here we will derive an analytical solution to which we can compare the numerical results. We will use Von Neumann stability analysis to analytically obtain the stability criteria for the various finite difference schemes. In particular we will attempt to verify the stability criterion of the two-dimensional Forward Euler scheme using our numerical results.

2 Analytical

2.1 Derivation of the heat equation

To derive the heat equation we consider some substance of volume V and with surface S. At any given time there is some heat flowing in or out of the substance. We define the heat flux density $\boldsymbol{q}$

as the amount of energy flow per unit area and per unit time. Fourier's law states that the heat flux density is proportional to the temperature gradient

$$\boldsymbol{q} = -k\boldsymbol{\nabla}T \tag{1}$$

The thermal conductivity k of the material is defined by Eq. (1). Integrating both sides of Eq. (1) over the surface S we obtain

$$\frac{\partial Q}{\partial t} = -\oiint d\boldsymbol{S}\cdot k\boldsymbol{\nabla}T \tag{2}$$

where dQ is the amount of heat that flows out of the substance during a time interval dt. Let U be the internal energy of the substance. Assuming no work is being done on the substance the first law of thermodynamics says that $dU = -dQ$. Using $dU = C_V dT$where C_V is the heat capacity of the material at constant volume we can rewrite the left side of Eq. (2) as

$$-C_V \frac{\partial T}{\partial t}$$

We can rewrite the right side of Eq. (2) using the divergence theorem as

$$-\iiint_V dV \;\; \boldsymbol{\nabla}\cdot(k\boldsymbol{\nabla}T)$$

Using $C_V = mc_V$ where m is the mass of the substance and c_V is the specific heat capacity of the material, Eq. (2) now reads

$$mc_V \frac{\partial T}{\partial t} = \iiint_V dV \;\; \boldsymbol{\nabla}\cdot(k\boldsymbol{\nabla}T) \tag{3}$$

By only considering an infinitesimal part of the substance with mass dm and volume dVwe can substitute $\iiint_V dV \to dV$ and $m \to dm = \rho dV$ in Eq. (3) where ρ is the density of the material. Dividing both sides by dV we obtain the heat equation

$$\rho c_V \frac{\partial T}{\partial t} = \boldsymbol{\nabla}\cdot(k\boldsymbol{\nabla}T) \tag{4}$$

A more stripped down version of Eq. (4) is the diffusion equation

$$\frac{\partial u}{\partial t} = \boldsymbol{\nabla}^2 u \tag{5}$$

where time and position are dimensionless.

2.2 Analytical solution in one dimension

The one-dimensional diffusion is

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x} \tag{6}$$

which we will solve analytically on the interval $x \in [0,\ 1]$ for $t > 0$ with the boundary conditions $u(0,t) = 0$, $u(1,t) = 1$ and the initial condition $u(x,0) = 0$.

We can think of $u(x,t)$ as the temperature distribution along a rod of length 1 with a constant heat source at one of the end points. From our daily life experience we know that the temperature distribution should reach a steady state after some amount of time, in which case we can set the left side of Eq. (6) to zero and obtain

$$u\left(x, t \to \infty\right) = ax + b$$

i.e the temperature distribution approaches some first degree polynomial. The coefficients a and b are determined by the boundary conditions. In our case the temperature distribution will converge to

$$u\left(x, t \to \infty\right) = x \tag{7}$$

Eq. (7) satisfies the diffusion equation for all times t since the time derivative and the second order position derivative are both zero, but it does not satisfy our choice of the initial condition. But since the diffusion equation is linear, a sum of two solutions is also a solution. We can therefore try a solution on the form

$$u(x,t) = x + f(x,t)$$

where $f(x,t)$ has to satisfy $f(0,t) = f(1,t) = 0$ and $f(x) \equiv f(x,0) = -x$. We can find $f(x,t)$by using the method of separation of variables and assume that it can be written on the form $f(x,t) = F(x)G(t)$. Inserting it into Eq. (6) we obtain

$$F\frac{dG}{dt} = G\frac{d^2F}{dx^2}$$

Or equivalently

$$\frac{1}{G}\frac{dG}{dt} = \frac{1}{F}\frac{d^2F}{dx^2} \tag{8}$$

Since the left side is a function of t alone and the right side is a function of x alone, both sides of Eq. (8) must be a constant. It's convenient to call this constant $-k^2$ for reasons which will be apparent in a moment. We can then split Eq. (8) into two ordinary differential equations and obtain the respective general solutions

$$\frac{dG}{dt} = -k^2 G \rightarrow G(t) = e^{-k^2 t}$$

$$\frac{d^2 F}{dx^2} = -k^2 F \rightarrow F(x) = A\sin(kx) + B\cos(kx)$$

There is generally a constant in front of the exponential in $G(t)$, but it might as well be absorbed into $F(x)$. Choosing a constant on the form $-k^2$ ensures that there is no time dependence as $t \rightarrow \infty$.

To find the solution $f(x,t) = F(x)G(t)$ which satisfies $f(0,t) = f(1,t) = 0$ it is convenient to set $t = 0$ so that $G = 1$ and $F(0) = F(1) = 0$. From the first boundary condition we get

$$A\sin(0) + B\cos(0) = A\cdot 0 + B \cdot 1 = B = 0$$

And from $F(1) = 0$ we get

$$A\sin(k) = 0 \rightarrow k = n\pi$$

where n is some positive integer. n could be a negative integer too, but since $\sin(x)$ is an odd function, $\sin(-|n|\pi x) = -\sin(|n|\pi x)$ where the minus sign might as well be absorbed into A. A solution $f(x,t)$ which satisfies $f(0,t) = f(1,t) = 0$ is thus on the form

$$f(x,t) = A\sin(n\pi x)e^{-n^2\pi^2 t}$$

but it does not satisfy $f(x,0) = f(x) = -x$ for any n. Again we can exploit the linearity of the diffusion equation and write $f(x,t)$ as

$$f(x,t) = \sum_{n=1}^{\infty} A_n \sin(n\pi x)e^{-n^2\pi^2 t} \tag{9}$$

which is a Fourier series. The functions $\sin(n\pi x)$ are orthogonal with respect to the inner product

$$2\int_0^1 dx \sin(m\pi x)\sin(n\pi x) = \delta_{nm}$$

If we set $t = 0$ in Eq. (9) and "multiply" both sides by $2\int_0^1 dx \sin(m\pi x)$ we obtain

$$A_n = -2\int_0^1 dx \sin(n\pi x)x = (-1)^n \frac{2}{n\pi}$$

The analytical solution can thus be written as

$$u(x,t) = x + \sum_{n=1}^{\infty} (-1)^n \frac{2}{n\pi} \sin(n\pi x)e^{-n^2\pi^2 t} \tag{10}$$

2.3 Analytical solution in two dimensions

The two-dimensional diffusion equation is

$$\frac{\partial u(x,y,t)}{\partial t} = \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} \tag{11}$$

which for simplicity will be solved on the area $x, y \in [0,1]$ with the boundary conditions $u|_{x=0} = u|_{y=0} = u|_{x=1} = u|_{y=1} = 0$ and with initial condition $f(x,y) \equiv u(x,y,0)$ (it'll soon be apparent what's a convenient pick for the initial condition).

Again we use the method of separation of variables and assume a solution on the form $u(x,y,t) = X(x)Y(y)T(t)$. Substituting this into Eq. (11) we obtain

$$XY\frac{dT}{dt} = YT\frac{d^2X}{dx^2} + XY\frac{d^2Y}{dy^2}$$

Dividing both sides by XYT we get

$$\frac{1}{T}\frac{dT}{dt} = \frac{1}{X}\frac{d^2X}{dx^2} + \frac{1}{Y}\frac{d^2Y}{dy^2} \tag{12}$$

where the left side is a function of t alone and the right side is a function of position alone. So both sides is equal to some constant $-k^2$. But we could move the t-term over to the right and either the x-or y-term over to the left, in which case the left side would be a function of either x or y alone. So all terms in Eq. (12) must be constants.

Let the x-term be $-p^2$ and the y-term be $-q^2$ where $p^2 + q^2 = k^2$. Then we get the three ordinary differential equations

$$\frac{d^2X}{dx^2} = -p^2X \to X(x) = A\sin(px) + B\cos(px)$$

$$\frac{d^2Y}{dy^2} = -q^2Y \to Y(y) = C\sin(qy) + D\cos(qy)$$

$$\frac{dT}{dt} = -k^2T \to T(t) = e^{-k^2t}$$

where the coefficient in front of the exponential in $T(t)$ has been absorbed into $X(x)Y(y)$.

Analogous to the one-dimensional case, the boundary conditions are satisfied by choosing $B = D = 0$ and $p = m\pi,\ q = n\pi$ for some positive integers m and n. Again by exploiting the linearity of the diffusion equation

$$u(x,y,t) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} A_{m,n}\sin(m\pi x)\sin(n\pi y)e^{-\pi^2\left(m^2+n^2\right)t}$$

Setting $t = 0$ and "multiplying" both sides by $4\int_0^1 dx \int_0^1 dy \sin(m'\pi x)\sin(n'\pi y)$ we obtain

$$A_{m,n} = 4\int_0^1 dx \int_0^1 dy\ f(x,y)\sin(m\pi x)\sin(n\pi x)$$

In particular, if $f(x,y) = \sin(\pi x)\sin(\pi y)$ only $A_{1,1} = 1$ is nonzero. In this case the analytical solution is

$$u(x,y,t) = \sin(\pi x)\sin(\pi y)e^{-2\pi^2 t} \tag{13}$$

where the factor of 2 in the exponential comes from $\left(m^2+n^2\right) = 1^2 + 1^2 = 2$.

3 Numerical methods

3.1 Numerical solution to tridiagonal matrix equations

The two implicit schemes Backward Euler and Crank-Nicolson requires us to solve a tridiagonal matrix equation, so we'll take a moment to go through how this can be done efficiently. Consider the matrix equation

$$\begin{bmatrix} 1 & & & & & \\ a_1 & d_1 & b_1 & & & \\ & & \ddots & & & \\ & & a_{n-2} & d_{n-2} & b_{n-2} \\ & & & & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

The indices has been chosen to match the standard array indexing in the C/C++ and Python programming languages. The matrix is not completely tridiagonal; a 1 has been added to the top left and the bottom right elements to allow us to choose any boundary condition. Omitting these elements correspond to setting $y_0 = y_{n-1} = 0$, i.e Dirichlet boundary conditions. Multiplying out the matrix we get a set of linear equations where the first few are

$$x_0 = y_0 \tag{14}$$

$$a_1x_0 + d_1x_1 + b_1x_2 = y_1 \tag{15}$$

$$a_2x_1 + d_2x_2 + b_2x_3 = y_2$$

What we're going to do is to eliminate the a_i in every equation using the previous equation. Multiplying Eq. (14) by a_1 and subtracting the result from Eq. (15) we get

$$d_1x_1 + b_1x_2 = \tilde{y}_1 \tag{16}$$

$$a_2x_1 + d_2x_2 + b_2x_3 = y_2 \tag{17}$$

Where $\tilde{y}_1 \equiv y_1 - a_1 y_0$. Multiplying Eq. (16) by a_2/d_1 and subtracting the result from Eq. (17) we get

$$\left(d_2 - \frac{a_2}{d_1} b_1\right) x_2 + b_2 x_3 = y_2 - \frac{a_2}{d_1} \tilde{y}_1$$

Defining $\tilde{d}_2 \equiv d_2 - \frac{a_2}{d_1} b_1$ and $\tilde{y}_2 \equiv y_2 - \frac{a_2}{d_1} \tilde{y}_1$ the next few equations are

$$\tilde{d}_2 x_2 + b_2 x_3 = \tilde{y}_2 \tag{18}$$

$$a_3 x_2 + d_3 x_3 + b_3 x_4 = y_3 \tag{19}$$

Notice that Eq. (18) and (19) are on the same form as Eq. (18) and Eq. (17). Repeating this process using the recurrence relations

$$\tilde{d}_i = d_i - \frac{a_i}{\tilde{d}_{i-1}} b_{i-1}, \ \tilde{d}_1 = d_1 \tag{20}$$

$$\tilde{y}_i = y_i - \frac{a_i}{\tilde{d}_{i-1}} \tilde{y}_{i-1}, \ \tilde{y}_1 = y_1 - a_0 y_0 \tag{21}$$

for $i = 2, \ 3, \ \ldots n-2$ we have effectively rewritten the matrix equation as

$$\begin{bmatrix} 1 & & & & \\ & \tilde{d}_1 & b_1 & & \\ & & \ddots & & \\ & & & \tilde{d}_{n-2} & b_{n-2} \\ & & & & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ \tilde{y}_1 \\ \vdots \\ \tilde{y}_{n-2} \\ y_{n-1} \end{bmatrix}$$

Eq. (20) and (21) are a type of forward substitution algorithms. The last few equations are now

$$x_{n-3} \tilde{d}_{n-3} + b_{n-3} x_{n-2} = \tilde{y}_{n-3}$$

$$x_{n-2} \tilde{d}_{n-2} + b_{n-2} x_{n-1} = \tilde{y}_{n-2} \tag{22}$$

$$x_{n-1} = y_{n-1}$$

Solving Eq. (22) for x_{n-2} we get

$$x_{n-2} = \frac{\tilde{y}_{n-2} - b_{n-2} x_{n-1}}{\tilde{d}_{n-2}}$$

It's now easy to see that we can solve for all the x_i's using the backward substitution algorithm

$$x_i = \frac{\tilde{y}_i - b_i x_{i+1}}{\tilde{d}_i}, \;\; x_{n-1} = y_{n-1} \tag{23}$$

for $i = n-2, \; n-3, \; \ldots, \; 1$ (x_0 is already given by Eq. (14)).

3.2 The Forward Euler scheme

Consider a function $f(x)$ and its Taylor expansion around the point $x = a$.

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2 + \mathcal{O}\left(x^3\right) \tag{24}$$

We want to find an approximation to the first and second derivative of $f(x)$ at an arbitrary point x. If h is a small number we can accomplish this by substituting $a \to x$ and $x - a \to h$ in Eq. (24) to obtain

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \mathcal{O}(h^3) \tag{25}$$

Solving for $f'(x)$ we get

$$f'(x) = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(-h) \tag{26}$$

Truncating all the terms containing powers of h larger than or equal to 1 we get an approximation to the first derivative which is first-order accurate, because the truncation error $\mathcal{O}(-h)$ is approximately proportional to the first power of h. To obtain a similar expression to Eq. (26) for the second derivative we can substitute $a \to x$ and $x - a \to -h$ in Eq. (24).

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2}h^2 + \mathcal{O}(-h^3) \tag{27}$$

Adding Eq. (25) and (27) together we get

$$f(x+h) + f(x-h) = 2f(x) + f''(x)h^2 + \mathcal{O}(h^4)$$

Solving for $f''(x)$:

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \mathcal{O}(-h^2) \tag{28}$$

where the approximation to the second derivative is second-order accurate. We can now write Eq. (6) as

$$\frac{u(x,t+\Delta t) - u(x,t)}{\Delta t} + \mathcal{O}(\Delta t) = \frac{u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)}{\Delta x^2} + \mathcal{O}\left(\Delta x^2\right) \tag{29}$$

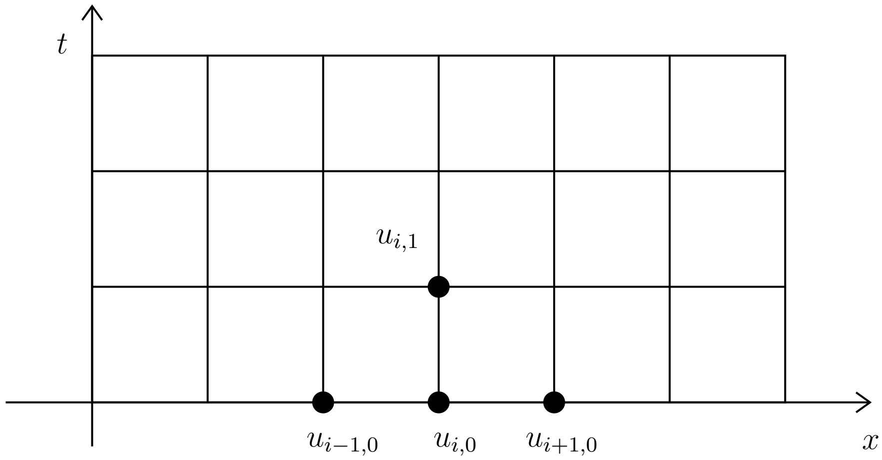
We'll now discretize the domain $x \in [0, 1]$ into points $x_i = i\Delta x$ and times $t \in [0,\ T]$ into $t_i = j\Delta t$ for $i = 0,\ 1,\ \ldots,\ N-1$ and $j = 0,\ 1,\ \ldots, n-1$ where $x_{N-1} = 1$ and $t_{n-1} = T$. The step sizes are then given by

$$\Delta x = \frac{1}{N-1},\ \Delta t = \frac{T}{n-1}$$

Substituting $u_{i,j} \equiv u(x_i, t_j)$ into Eq. (29) and sloppily throwing away the truncation errors we can write

$$u_{i,j+1} = \alpha u_{i+1,j} + (1-2\alpha)u_{i,j} + \alpha u_{i-1,j} \tag{30}$$

where $\alpha \equiv \Delta t/\Delta x^2$. Eq. (30) is the Forward Euler scheme. It's an explicit scheme, meaning that there is only a single unknown quantity explicitly given in terms of known quantities.



This is illustrated in the figure above which shows the numerical solution $u_{i,j}$ as a function of x and t. All quantities $u_{i,\ 0}$ for $i = 0,\ \ldots,\ N-1$ on the first row are known because they come from the initial condition and the boundary conditions. Thus every single quantity $u_{i,1}$ for $i = 1,\ \ldots,\ N-2$ can be calculated. After that every single quantity on the third row can be calculated, and so on. This shows that when we implement Eq. (30) in our program the inner loop has to loop over position and the outer loop has to loop over time.

3.3 The Backward Euler scheme

The Backward Euler scheme looks just a tiny bit different from the Forward Euler scheme, but the way we actually obtain the numerical solution $u_{i,j}$ is very different. We still use Eq. (28) to approximate the second-order position derivative in Eq. (6), but the new approximation to the time-derivative is obtained by using Eq. (27)

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \mathcal{O}(h) \tag{31}$$

We can use Eq. (28) and Eq. (31) to write Eq. (6) as

$$\frac{u(x,t)-u(x,t-\Delta t)}{\Delta t}+\mathcal{O}(\Delta t)=\frac{u(x+\Delta x,t)-2u(x,t)+u(x-\Delta x,t)}{\Delta x^2}+\mathcal{O}\left(\Delta x^2\right) \tag{32}$$

We discretize the domain $x \in [0,1]$ and the times $t \in [0,\ T]$ similarly to what we did in the previous section. Again substituting $u_{i,j} \equiv u(x_i,t_j)$ and introducing $\alpha = \Delta t/\Delta x^2$ we can put all u_j's and u_{j-1}'s on separate sides of the equality sign and write Eq. (32) as

$$u_{i,j-1} = -\alpha u_{i+1,j} + (1+2\alpha)u_{i,j} - \alpha u_{i-1,j} \tag{33}$$

If we set $j=1$ we see that $u_{i,j-1}$ is the only one known of all the $u_{i,j}$'s in Eq. (33). So a single equation is not going to cut it. What we can do is to write Eq. (33) as a system of linear equations

$$\begin{gathered}
u_{0,j} = u_{0,j-1}\\
-\alpha u_{0,j} + (1+2\alpha)u_{1,j} - \alpha u_{2,j} = u_{1,j-1}\\
\vdots\\
-\alpha u_{N-3,j} + (1+2\alpha)u_{N-2,j} - \alpha u_{N-1,j} = u_{N-2,j-1}\\
u_{N-1,j} = u_{N-1,j-1}
\end{gathered}$$

which we can write as the matrix equation

$$\begin{bmatrix} 1 & & & & \\ -\alpha & 1+2\alpha & -\alpha & & \\ & & \ddots & & \\ & & -\alpha & 1+2\alpha & -\alpha \\ & & & & 1 \end{bmatrix}\begin{bmatrix} u_{0,j} \\ u_{1,j} \\ \vdots \\ u_{N-1,j}\end{bmatrix} = \begin{bmatrix} u_{0,j-1} \\ u_{1,j-1} \\ \vdots \\ u_{N-1,j-1}\end{bmatrix}$$

and solve using the recurrence relations in Eq. (20), (21) and (23). The fact that we have to solve a system of linear equations to obtain the numerical solution $u_{i,j}$ makes the Backward Euler scheme an implicit scheme.

3.4 The Crank-Nicolson scheme

We can derive the Crank-Nicolson scheme in a simple way using the Forward and Backward Euler schemes, but first we have to rewrite the Backward Euler scheme slightly, shifting it up one time step:

$$\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t} = \frac{u(x+\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x-\Delta x,t+\Delta t)}{\Delta x^2}$$

What we're going to do now is to approximate the first time derivative of and the second position derivative of $u(x,t)$ at the point $(x, t+\Delta t/2)$ by taking the average of the Forward and Backward Euler schemes. The approximation to the time derivative is

$$\frac{1}{2}\left[\overbrace{\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}}^{\text{Forward Euler}}+\overbrace{\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}}^{\text{Rewritten Backward Euler}}\right]=\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t} \tag{34}$$

and the approximation to the second order position derivative is

$$\frac{1}{2}\left[\overbrace{\frac{u(x+\Delta x,t)-2u(x,t)+u(x-\Delta x,t)}{\Delta x^2}}^{\text{Forward Euler}}+\overbrace{\frac{u(x+\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x-\Delta x,t+\Delta t)}{\Delta x^2}}^{\text{Rewritten Backward Euler}}\right] \tag{35}$$

Since the approximation to the time derivative is centered at $t+\Delta t/2$ and evaluated at t and $t+\Delta t$ it corresponds to

$$f'(t)=\frac{f(t+\Delta t)-f(t-\Delta t)}{\Delta t}+\mathcal{O}\left(\Delta t^2\right) \tag{36}$$

which is second-order accurate in Δt. The approximation to the second-order position derivative is however centered at x and evaluated at x and $x \pm \Delta x$, so it is still "only" second-order accurate in Δx^2 as with the Forward and Backward Euler schemes.

Discretizing Eq. (34) and (35) similarly to what we did with the Forward and Backward Euler schemes and substituting $u_{i,j} \equiv u(x_i, t_j)$ we can bring it all together.

$$\frac{u_{i,j+1}-u_{i,j}}{\Delta t}=\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}+u_{i+1,j+1}-2u_{i,j+1}+u_{i-1,j+1}}{2\Delta x^2}$$

Again introducing $\alpha \equiv \Delta t/\Delta x^2$ and moving all u_j's and u_{j+1}'s on separate sides of the equality sign we obtain

$$-\alpha u_{i-1,j}+2(1+\alpha)u_{i,j}-\alpha u_{i+1,j}=\alpha u_{i-1,j-1}+2(1-\alpha)u_{i,j-1}+\alpha u_{i+1,j-1} \tag{37}$$

Setting $j=1$ we see that there are three unknowns $u_{i-1,1}, u_{i,1}$ and $u_{i+1,1}$ in Eq. (37), so we need to write it as a system of linear equations.

$$u_{0,j}=u_{0,j-1}$$

$$-\alpha u_{0,j}+2(1+\alpha)u_{1,j}-\alpha u_{2,j}=\alpha u_{0,j-1}+2(1-\alpha)u_{1,j-1}+\alpha u_{2,j-1}$$

$$\vdots$$

$$-\alpha u_{N-3,j} + 2(1+\alpha)u_{N-2,j} - \alpha u_{N-1,j} = \alpha u_{N-3,j-1} + 2(1-\alpha)u_{N-2,j-1} + \alpha u_{N-1,j-1}$$

$$u_{N-1,j} = u_{N-1,j-1}$$

This can be written as the matrix equation

$$(2I + \alpha B)\boldsymbol{u}_j = (2I - \alpha B)\boldsymbol{u}_{j-1} \tag{38}$$

$$\begin{bmatrix} 1 & & & & \\ \alpha & 2(1-\alpha) & \alpha & & \\ & & \ddots & & \\ & & \alpha & 2(1-\alpha) & \alpha \\ & & & & 1 \end{bmatrix} \begin{bmatrix} u_{0,j-1} \\ u_{1,j-1} \\ \vdots \\ u_{n-1,j-1} \\ u_{N-1,j-1} \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ -\alpha & 2(1+\alpha) & -\alpha & & \\ & & \ddots & & \\ & & -\alpha & 2(1+\alpha) & -\alpha \\ & & & & 1 \end{bmatrix} \begin{bmatrix} u_{0,j} \\ u_{1,j} \\ \vdots \\ u_{n-1,j} \\ u_{N-1,j} \end{bmatrix}$$

where I is the identity matrix and

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

Defining $\tilde{\boldsymbol{u}}_{j-1} \equiv (2I - \alpha B)\boldsymbol{u}_{j-1}$ we can write Eq. (38) as

$$(2I + \alpha B)\boldsymbol{u}_j = \tilde{\boldsymbol{u}}_{j-1}$$

Since $2I + \alpha B$ is on the same form as the matrix discussed in section 3.1 we can use the recurrence relations in Eq. (20), (21) and (23) to solve for $\boldsymbol{u}_j$.

3.5 The two-dimensional Forward Euler scheme

To solve the two-dimensional diffusion equation (Eq. (11)) we approximate the two second order position derivatives by Eq. (28) while the time derivative is approximated by Eq. (26). Here the index i represents x, j represents y and n represents t.

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

We'll use the same step sizes in the x-and y-coordinates $h \equiv \Delta x = \Delta y$. Defining $\alpha \equiv \Delta t / h^2$ and solving for $u_{i,j}^{n+1}$ we get

$$u_{i,j}^{n+1} = u_{i,j}^n + \alpha\left(u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 4u_{i,j}^n\right) \tag{39}$$

Setting $n = 0$ makes it clear that $u_{i,j}^{n+1}$ is the only unknown in Eq. (39) since all the $u^{n=0}$ are known from the initial condition. The two-dimensional Forward Euler scheme is thus also an explicit scheme.

4 Stability analysis

In this section we will use Von Neumann stability analysis to determine for what values of Δx and Δt the various schemes discussed in the previous sections are stable, but first we need to go though some definitions.

A finite difference scheme is said to be *convergent* if the numerical solution of the scheme converges to the exact analytical solution of the partial differential equation as Δx and Δt approaches zero.

A finite difference scheme is said to be *consistent* if it is at least first order accurate in Δx and Δt. The truncation errors are $\mathcal{O}\left(\Delta x^{\beta}\right)$ and $\mathcal{O}\left(\Delta t^{\gamma}\right)$ where $\beta, \gamma \geq 1$.

A finite difference scheme is said to be *unconditionally stable* if the error, that is the difference between the numerical solution and the exact solution, do not tend to infinity for any choice of $\Delta x$and Δt.

A finite difference scheme is said to be *conditionally stable* if the error does not tend to infinity for some, but not all choices of Δx and Δt.

These definitions are brought together by the Lax-Equivalence theorem which states that a finite difference scheme is convergent if and only if it is consistent and stable [1]. All the finite difference schemes discussed in the previous sections are consistent, so to show that they are convergent we'll use Von Neumann stability analysis to figure out if and when they are stable.

To do this it is convenient to write the general solution of the diffusion equation on the more compact form

$$f(x,t) = \sum_{k=-\infty}^{\infty} A_k e^{ikx} e^{-k^2 t}$$

But it suffices to use only one of the functions in the linear combination

$$f_k(x,t) = A_k e^{ikx} e^{-k^2 t}$$

Using discrete points $x_j = j\Delta x$ and times $t_n = n\Delta t$ we can write

$$f_k(x_j, t_n) = A_k e^{ikj\Delta x} e^{-k^2 n\Delta t}$$

$$= A_k e^{-k^2 n\Delta t} e^{ikj\Delta x} = \left(B_k e^{-k^2\Delta t}\right)^n e^{ikj\Delta x}$$

$$= \xi_a(k)^n e^{ikj\Delta x} \tag{40}$$

where $A_k = B_k^n$ and $\xi_a(k) \equiv B_k e^{-k^2\Delta t}$ is the analytical amplification factor. The various finite difference schemes may be thought of as difference equations where Eq. (40) is a solution. If $u_{j,n}$ is the numerical solution and $\epsilon_{j,n}$ is the error we can write

$$f_k(x_j, t_n) = u_{j,n} - \epsilon_{j,n} \tag{41}$$

where $\epsilon_{j,n} \equiv u_{j,n} - f_k(x_j, t_n)$. We can put $f_{j,n} \equiv f_k(x_j, t_n)$ into the Forward Euler scheme

$$f_{j,n+1} = \alpha f_{j+1,n} + (1-2\alpha) f_{j,n} + \alpha f_{j-1,n}$$

$$(u_{j,n+1} - \epsilon_{j,n+1}) = \alpha(u_{j+1,n} - \epsilon_{j+1,n}) + (1-2\alpha)(u_{j,n} - \epsilon_{j,n}) + \alpha(u_{j-1,n} - \epsilon_{j-1,n})$$

But the $u_{j,n}$'s are given exactly by the Forward Euler scheme (Eq. (30)), so we have the difference equation for the errors

$$\epsilon_{j,n+1} = \alpha\epsilon_{j+1,n} + (1-2\alpha)\epsilon_{j,n} + \alpha\epsilon_{j-1,n} \tag{42}$$

where $\epsilon_{j,n} = \xi(k)^n e^{ikj\Delta x}$ is a solution.

$\xi(k)$ is called the amplification factor and we see that the errors tend to infinity with each time step if $|\xi(k)| > 1$. The stability criterion for a finite difference scheme is thus$|\xi(k)| \leq 1$ for all k.

To solve for the amplification factor we can put $\epsilon_{j,n} = \xi(k)^n e^{ikj\Delta x}$ into Eq. (42)

$$\xi(k)^{n+1} e^{ikj\Delta x} = \alpha\xi(k)^n e^{ik(j+1)\Delta x} + (1-2\alpha)\xi(k)^n e^{ikj\Delta x} + \alpha\xi(k)^n e^{ik(j-1)\Delta x}$$

Dividing both sides by $\xi(k)^n e^{ikj\Delta x}$

$$\xi(k) = \alpha e^{ik\Delta x} + (1-2\alpha) + \alpha e^{-ik\Delta x}$$

$$= 1 - 2\alpha + 2\alpha\cos(k\Delta x) = 1 - 2\alpha\left[1 - \cos(k\Delta x)\right]$$

$$= 1 - 4\alpha\sin^2\left(\frac{k\Delta x}{2}\right)$$

By varying k we see that this expression can at most be 1 when the sine is zero and at least be $1-4\alpha$ when the sine is ± 1. Using the stability criterion $|\xi(k)| \leq 1$ we have

$$(1-4\alpha)^2 \leq 1$$

$$\alpha \leq \frac{1}{2}$$

The Forward Euler scheme is thus conditionally stable with stability criterion $\Delta t/\Delta x^2 \leq 1/2$. We can repeat this process for the other finite difference schemes. For the Backward Euler scheme we have (Eq. (33))

$$\epsilon_{j,n-1} = -\alpha\epsilon_{j+1,n} + (1+2\alpha)\epsilon_{j,n} - \alpha\epsilon_{j-1,n}$$

$$\xi(k)^{n-1} e^{ikj\Delta x} = -\alpha\xi(k)^n e^{ik(j+1)\Delta x} + (1+2\alpha)\xi(k)^n e^{ikj\Delta x} - \alpha\xi(k)^n e^{ik(j-1)\Delta x}$$

Dividing both sides by $\xi(k)^n e^{ikj\Delta x}$ we get

$$\xi(k)^{-1} = -\alpha e^{ik\Delta x} + (1+2\alpha) - \alpha e^{-ik\Delta x}$$

$$= 1 + 2\alpha - 2\alpha\cos(k\Delta x)$$

$$= 1 + 2\alpha\left[1 - \cos(k\Delta x)\right]$$

$$= 1 + 4\alpha\sin^2\left(\frac{k\Delta x}{2}\right)$$

This expression is at least 1 when the sine is zero, meaning that $|\xi(k)| \leq 1$ for all k. So the Backward Euler scheme is unconditionally stable. For the Crank Nicolson scheme we have (Eq. (37))

$$-\alpha\epsilon_{j-1,n} + 2(1+\alpha)\epsilon_{j,n} - \alpha\epsilon_{j+1,n} = \alpha\epsilon_{j-1,n-1} + 2(1-\alpha)\epsilon_{j,n-1} + \alpha\epsilon_{j+1,n-1}$$

$$-\alpha\xi(k)^n e^{ik(j-1)\Delta x} + 2(1+\alpha)\xi(k)^n e^{ikj\Delta x} - \alpha\xi(k)^n e^{ik(j+1)\Delta x}$$

$$= \alpha\xi(k)^{n-1} e^{ik(j-1)\Delta x} + 2(1-\alpha)\xi(k)^{n-1} e^{ikj\Delta x} + \alpha\xi(k)^{n-1} e^{ik(j+1)\Delta x}$$

Again dividing both sides by $\xi(k)^n e^{ikj\Delta x}$ we get

$$-\alpha e^{-ik\Delta x} + 2(1+\alpha) - \alpha e^{ik\Delta x}$$

$$= \alpha\xi(k)^{-1} e^{-ik\Delta x} + 2(1-\alpha)\xi(k)^{-1} + \alpha\xi(k)^{-1} e^{ik\Delta x}$$

$$2(1+\alpha) - 2\alpha\cos(k\Delta x) = \xi(k)^{-1}\left[2(1-\alpha) + 2\alpha\cos(k\Delta x)\right]$$

$$1 + \alpha\left[1 - \cos(k\Delta x)\right] = \xi(k)^{-1}\left\{1 - \alpha\left[1 - \cos(k\Delta x)\right]\right\}$$

$$1 + 2\alpha\sin^2\left(\frac{k\Delta x}{2}\right) = \xi(k)^{-1}\left[1 - 2\alpha\sin^2\left(\frac{k\Delta x}{2}\right)\right]$$

$$\xi(k) = \frac{1 - 2\alpha\sin^2(k\Delta x/2)}{1 + 2\alpha\sin^2(k\Delta x/2)}$$

This expression is 1 when the sines are zero. Otherwise the numerator is less than one and the denominator is larger than one, so it's easy to see that $|\xi(k)| \leq 1$ for all k. The Crank-Nicolson scheme is thus also unconditionally stable.

Finally we have the two-dimensional Euler forward scheme (Eq. (39)) where we assume that the solution is on the form $\epsilon_{j,l}^n = \xi(k)^n e^{ikjh} e^{iklh}$. Here the index l represent y while the indices j and n still represent x and t respectively

$$\epsilon_{j,l}^{n+1} = \epsilon_{j,l}^n + \alpha \left(\epsilon_{j+1,l}^n + \epsilon_{j-1,l}^n + \epsilon_{j,l+1}^n + \epsilon_{j,l-1}^n - 4\epsilon_{j,l}^n \right)$$

$$\xi(k)^{n+1} e^{ikjh} e^{iklh} = \xi(k)^n e^{ikjh} e^{iklh}$$

$$+\alpha \xi(k)^n \left(e^{ik(j+1)h} e^{iklh} + e^{ik(j-1)h} e^{iklh} + e^{ikjh} e^{ik(l+1)h} + e^{ikjh} e^{ik(l-1)h} - 4e^{ikjh} e^{iklh} \right)$$

Dividing both sides by $\xi(k)^n e^{ikjh} e^{iklh}$ we get

$$\xi(k) = 1 + \alpha \left(e^{ikh} + e^{-ikh} + e^{ikh} + e^{-ikh} - 4 \right)$$

$$= 1 + \alpha \left(4\cos(kh) - 4 \right)$$

$$= 1 - 4\alpha \left[1 - \cos(kh) \right]$$

$$= 1 - 8\alpha \sin^2 \left(\frac{kh}{2} \right)$$

This expression is at most 1 when the sine is zero and at least $1 - 8\alpha$ when the sine is one. The stability criterion gives

$$(1 - 8\alpha)^2 \leq 1$$

$$\alpha \leq \frac{1}{4}$$

So the two-dimensional Forward Euler scheme is conditionally stable with stability criterion $\Delta t / h^2 \leq 4$. The following table summarizes the results of this section.

Numerical scheme	**Stability criterion**
Forward Euler	$\Delta t / \Delta x^2 \leq 1/2$
Backward Euler	Stable for all $\Delta t,\ \Delta x$
Crank-Nicolson	Stable for all $\Delta t,\ \Delta x$
Two-dimensional Forward Euler	$\Delta t / h^2 \leq 1/4$

5 Procedure and initial considerations

In the following section we will compare the one-dimensional finite difference schemes Forward Euler, Backward Euler and Crank-Nicolson to the analytical solution in Eq. (10), which was obtained using the boundary conditions $u(0,t) = 0$, $u(1,t) = 1$ and the initial condition $u(x,0) = 0$. We do this qualitatively by animating the analytical and numerical solution in the same plot. We know that the solution will converge to a straight line sooner or later, so the question is when is the solution pretty much a straight line? This is when we should stop the numerical calculations.

To answer this the analytical solution has been animated in the time interval $t \in [0,\ 0.5]$.

We see that the analytical solution is pretty much straight after $t = 0.4$, so this is when we stop the numerical calculations.

The position step sizes in the numerical calculations will be $\Delta x = 0.1$ and $\Delta x = 0.01$. For the Forward Euler scheme the time step will be picked according to the stability criterion $\alpha \leq 1/2$, i.e $\Delta t = 5 \cdot 10^{-3}$ and $\Delta t = 5 \cdot 10^{-5}$. For the Backward Euler and Crank-Nicolson schemes the time step can be whatever as they are always stable. For the purpose of making an animation, $n = 400$ points in the interval $t \in [0,\ 0.4]$ is sufficient.

We will also compare these schemes with each other quantitatively by computing the error between the numerical solutions and the analytical solution and showing them in the same plot. The error will be computed according to

$$\epsilon(t_j) = \frac{\sqrt{\sum_{i=0}^{N-1} \left[\mathrm{FDS}(x_i, t_j)\ -\ u(x_i,\ t_j)\right]^2}}{\sqrt{\sum_{i=0}^{N-1} u(x_i, t_j)^2}} \tag{43}$$

where $u(x_i, t_i)$ is the analytical solution in Eq. (10) at position x_i and time t_j and FDS is the numerical result obtained from Forward Euler, Backward Euler or Crank-Nicolson. As the error is a function of time this will allow us to see how the finite difference schemes compare against each other when the solution is significantly curved and changing quickly as well as when the solution is nearly straight and changing slowly.

Here we will need to use the same step sizes in position and time. To ensure that the Forward Euler scheme is stable we use $\Delta x = 0.1,\ \Delta t = 5 \cdot 10^{-3}$ and $\Delta x = 0.01,\ \Delta t = 5 \cdot 10^{-5}$ for all the numerical schemes.

The analytical solution contains an infinite series which we will have to truncate, but the terms goes to zero very quickly. For a time t not close to zero the fifth term in the series is of magnitude $e^{-5^2}/5\ \sim\ 10^{-12}$, which is utterly negligible. However, for times t close to zero this term is of magnitude $1/5$, not at all negligible compared to the choice of boundary conditions. As a compromise we will truncate the infinite series at the 20th term to prevent the solution from looking too sine wavy initially.

The analytical solution when the infinite series is truncated at the 20th term is plotted at the times $t = 0$ and $t = 0.001$ in Figure 1. We see that the analytical solution looks somewhat sine wavy at $t = 0$, but that this is a non-issue at times $t \geq 0.001$. We do however need to keep this in mind when we plot the errors in the numerical results according to Eq. (43). It is expected that the error at times t close to zero will be artificially large.
In two dimensions we will compare the numerical results obtained from the two-dimensional Forward Euler scheme with the analytical expression in Eq. (13). Again we do this qualitatively by animating the numerical results and the analytical expression. The step sizes will again be

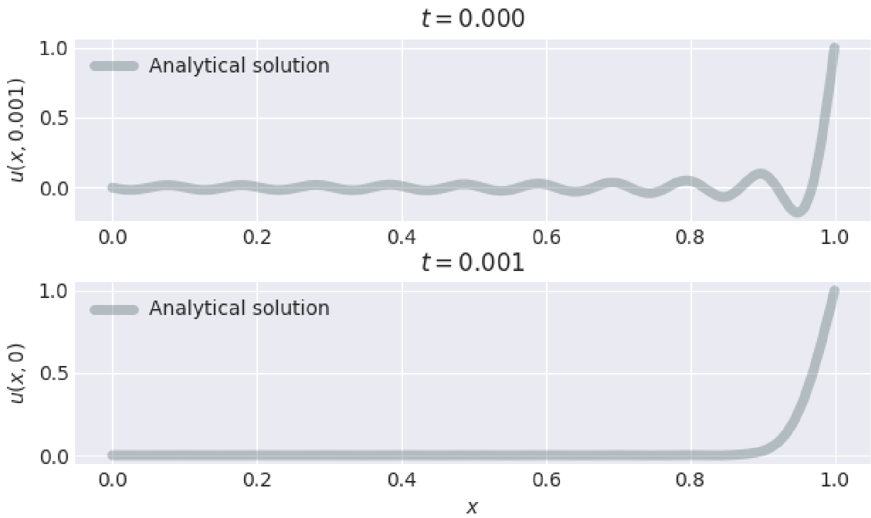


Figure 1: The analytical solution in Eq. 10 truncated at the 20th term at times $t = 0.000$ and $t = 0.001$.

$\Delta x = 0.1,\ 0.01$, but the time step sizes will be chosen according to the stability criterion $\alpha = 1/4$, i.e $\Delta t = 2.5 \cdot 10^{-3},\ 2.5 \cdot 10^{-5}$.

We will also investigate that our numerical results obtained from the two-dimensional Forward Euler scheme matches the stability criterion. We very well may not be getting unstable results for α just larger than $1/4$. We are therefore going to find the smallest α by trial and error such that the numerical results blows up to infinity. The whole process will not be documented in the results section. We are simply going to show the result when we have found this smallest α.

6 Results

6.1 Comparison of the one-dimensional finite difference schemes with the analytical solution

The numerical results obtained from the Forward Euler scheme, the Backward Euler scheme and the Crank Nicolson scheme has been animated for $\Delta x = 0.1$ and $\Delta x = 0.01$ together with the analytical solution. See the links in the table below.

Table 1: Links to animations of the numerical results obtained from the one-dimensional finite difference schemes.

Scheme	$\Delta x = 0.1$	$\Delta x = 0.01$
Forward Euler	Link	Link
Backward Euler	Link	Link
Crank-Nicolson	Link	Link

We see that all of the numerical results match the analytical solution rather nicely, perhaps with the exception of the Forward Euler scheme at $\Delta x = 0.1$ which doesn't look quite as nice for $t < 0.5$. It seems to be lagging behind the analytical solution somewhat but catches up when the analytical solution is nearly straight and changing slowly.

We seem to be getting some weird "shaking behavior" early on from the Crank-Nicolson scheme at $\Delta x = 0.01$, but it stops at $t = 0.04$ or so. This does not happen for $\Delta x = 0.1$. This is a surprising result considering that the α parameter is the same for the results from the Crank-Nicolson scheme at both $\Delta x = 0.1$ and $\Delta x = 0.01$. After the shaking behavior stops the numerical results seem to be fitting the analytical solution to a tee, so it seems unlikely that there is something wrong with our implementation of the Crank-Nicolson scheme. This must be a property of the scheme itself, so it would be interesting to investigate what causes the shaking behavior, for what choice of Δx and Δt and why there is shaking behavior only initially and only in a small part of the domain.

6.2 The error in the numerical results

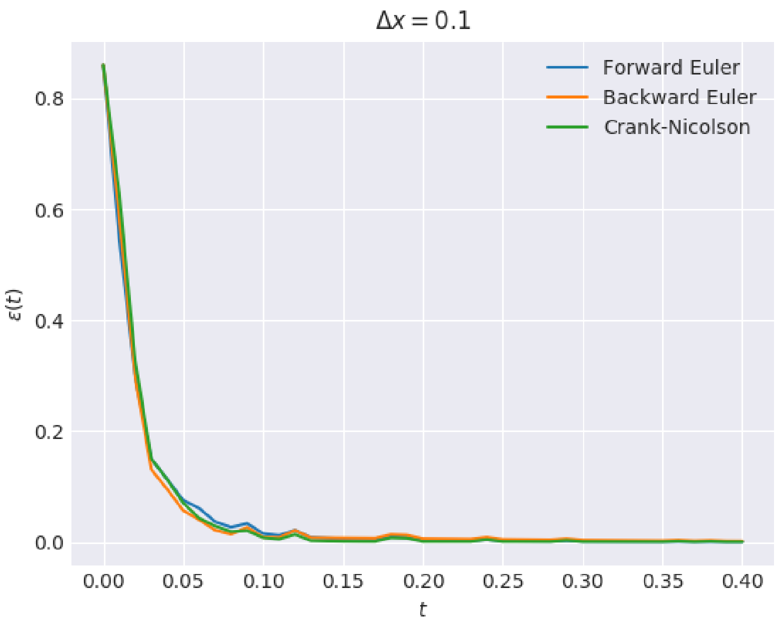


Figure 2: The error in the results obtained from the Forward Euler, Backward Euler and Crank-Nicolson schemes at $\Delta x = 0.1$. The error was calculated according to Eq. (43).

Figure 2 shows the error in the results obtained from the Forward Euler, Backward Euler and Crank-Nicolson schemes at $\Delta x = 0.1$ as a function of time. We see that the error is the largest when the solution is curved and rapidly changing while the error is the smallest as the solution converges to a straight line. At first glance it would seem like the three finite difference schemes are performing at about the same level. This is surprising considering that the results obtained from the Forward Euler scheme looked somewhat less visually appealing than the results from Backward Euler and Crank-Nicolson. The three lines corresponding to the respective finite difference methods lies on top of each other and it's hard to distinguish between them. We need to zoom in to see more of what's going on. This is done in figure 3.

In figure 3 it is easier to see the difference between the three finite difference schemes. We see that while the solution is curved and changing rapidly that the Backward Euler scheme performs slightly better than the Forward Euler and Crank-Nicolson schemes. However, later on the Backward Euler scheme seems to be losing the race and performs slightly worse than the two other schemes when the solution is nearly straight. The Forward Euler scheme performs the worst when the solution changes rapidly, but performs at about the same level as the Crank-Nicolson scheme later on. This is very surprising considering that the truncation error of the Crank-Nicolson scheme is second-order accurate in time. One would've thought that the error in the Forward Euler scheme would lie closer to the Backward Euler scheme considering that their truncation errors are both first order accurate in time. Interestingly, the errors in all the three schemes have periodic "bumps" happening at about the same time with about the same shape.

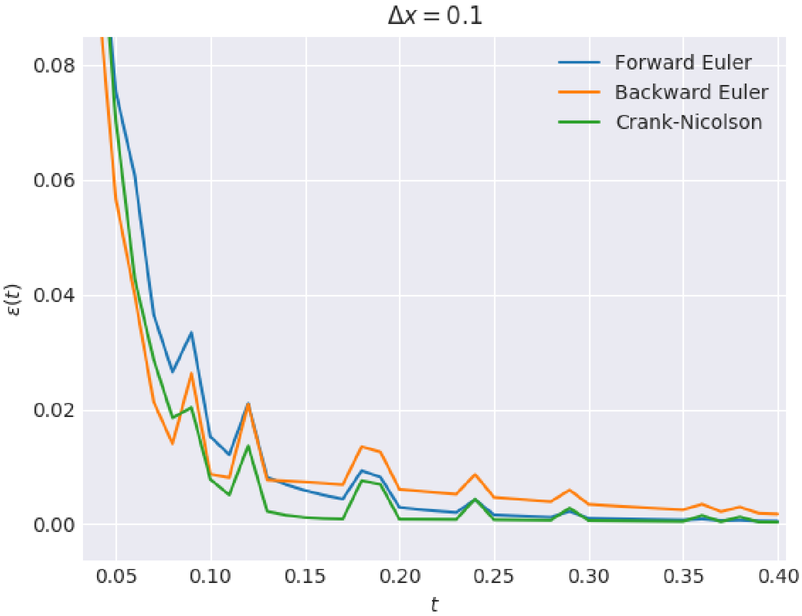


Figure 3: Zoomed in version of Figure 2.

Keep in mind that each line consists of 81 data points (the total time interval is 0.4 and we choose $\Delta t = 5 \cdot 10^{-3}$). The magnitude of the bumps decreases with time. It's difficult to point out anything in the animations in table 1 which correspond to the bumps that we see in figure 3, so this is a surprising result indeed.

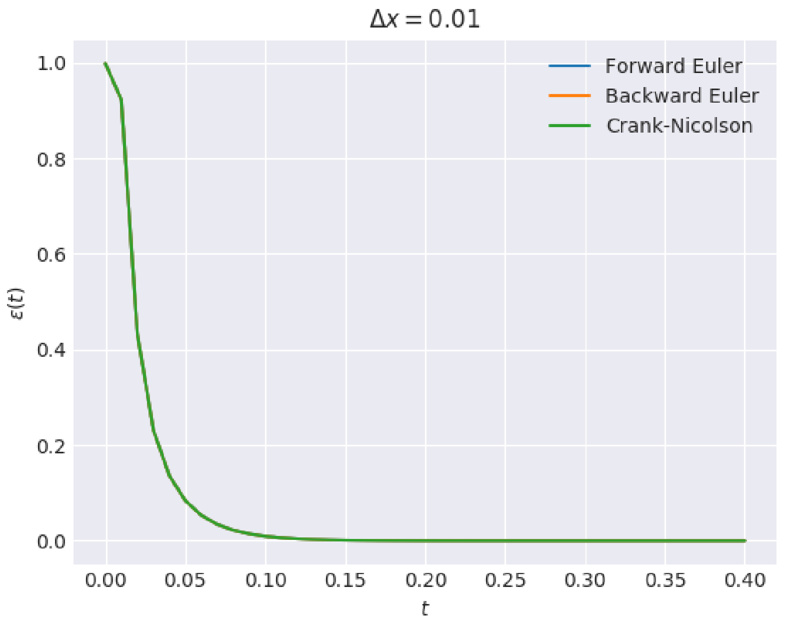


Figure 4: The error in the results obtained from the Forward Euler, Backward Euler and Crank-Nicolson schemes at $\Delta x = 0.01$. The error was calculated according to Eq. (43).

Figure 4 shows the error in the results obtained from the Forward Euler, Backward Euler and Crank-Nicolson schemes at $\Delta x = 0.01$ as a function of time. The three finite difference schemes seem to be performing at an utterly identical level at $\Delta x = 0.01$. This is also very surprising. There is clearly room for improvement in the error at around $t = 0.05$, long past the time where we can blame the truncation at the 20th term in the analytical expression. These are three very different schemes, both explicit and implicit, with different truncation errors as seen in the Taylor expansions. Yet they have decided to perform at nearly the exact same level while there is spare room for improvement in the error. The solution changes somewhat rapidly at $t = 0.05$, so there seems to be some sort of "inertia" that is nearly the same for these three finite difference schemes which prevents the error from becoming so and so small when the solution is changing so and so fast. This seems like a very interesting topic for further studies.

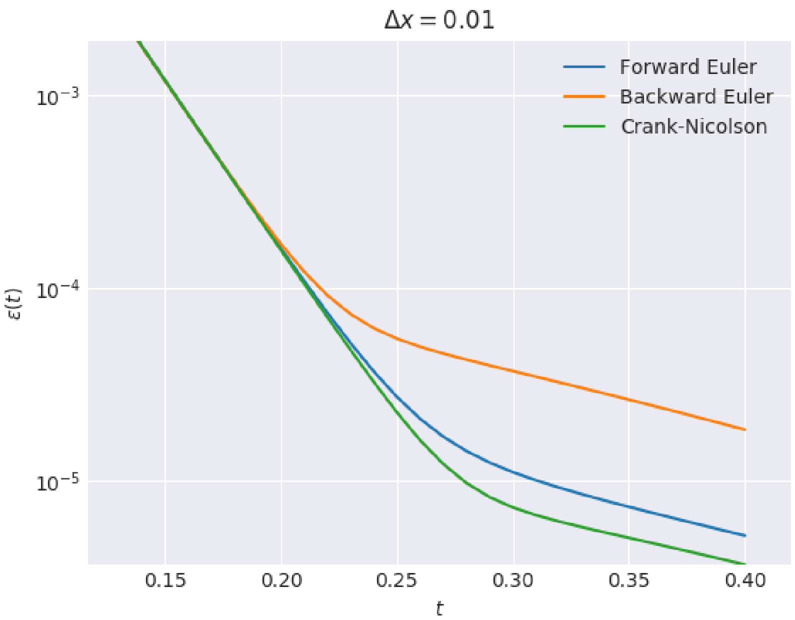


Figure 5: Zoomed in version of Figure 4.

Yes, there are three lines in Figure 4. If we zoom in and scale the y-axis logarithmically we obtain what is seen in Figure 5. The errors do differ from each other, but only marginally. Again we see that the Backward Euler scheme loses the race while the Forward Euler scheme performs at about the same level as the Crank-Nicolson scheme. But judging from figures 4 and 5 there is really no reason why one would choose one finite difference scheme over another at $\Delta x = 0.01$ (as long as the stability criterion of the Forward Euler scheme is satisfied). They give extremely similar results. Crank-Nicolson does seem to be "the best one", as one would've expect from the truncation errors in the Taylor expansion, but only marginally, which is very surprising. As we can see, one might as well use Forward Euler which is much easier and quicker to implement than Crank-Nicolson. The benefit of Crank-Nicolson is that there is no stability criterion to worry about. However, Crank-Nicolson is the only finite difference scheme where we observed the weird shaking behavior, the reason for which should be further investigated.

6.3 Numerical results in two dimensions

An animation of the analytical solution in Eq. (13) is shown here, while an animation of the numerical results obtained from the two-dimensional Forward Euler scheme is shown here and here for $h = 0.1$ and $h = 0.01$ respectively.

While we haven't been able to compare the numerical results with the analytical solution directly as we did with the one-dimensional diffusion equation, we see by eye that there is a good correspondence between the numerical results and the analytical solution. We could (and should) have calculated the error according to a two-dimensional version of Eq. (43), but for now we'll save this for future work.

Finally, the lowest value of $\alpha = \Delta t/h^2$ where the two-dimensional Forward Euler scheme was found to be unstable is $\alpha = 0.267$ corresponding to $\Delta t = 2.67 \cdot 10^{-3}$ and $h = 0.1$. An animation of the result is shown here. We see that the numerical results looks similar to what we've already seen, but that it starts blowing up to infinity after $t = 1.0$ or so. During our process of trial and error we observed that the numerical results would blow up later and later as α was lowered. It would seem reasonably to suggest that as $\alpha \to 1/4$ the time it takes for the numerical results to blow up approaches infinity in which case it is impossible to truly verify the stability criterion with numerical results alone.

7 Conclusion

We have solved the diffusion equation in one dimension using the Forward Euler, Backward Euler and Crank-Nicolson schemes and in two dimensions using the two-dimensional Forward Euler scheme. We obtained a good correspondence between the numerical results and the analytical solution for all of the finite difference schemes. The Forward Euler, Backward Euler and Crank-Nicolson seems to be performing at a nearly equal level in terms of the error between the numerical results and the analytical solution. At $\Delta x = 0.01$ their performance was nearly identical while there was still room for improvement in the error. This suggests that there exists some sort of "inertia" in these finite difference schemes which puts a lower bound on the error when the solution is changing with time. This lower bound increases by how much the solution is changing. Our numerical result with the two-dimensional Forward Euler scheme supports the analytically derived stability criterion. The smallest value of α where the analytical results blow up was found to be $\alpha = 0.267$, in agreement with the stability criterion.

In future work one should investigate this so-called "inertia" in the Forward Euler, Backward Euler and Crank-Nicolson schemes. It is of interest to know why there is a lower bound in the error when the solution is changing with time. In addition, the cause of the periodic bumps in the errors should be investigated along with the cause of the shaking behavior only observed with the results of the Crank-Nicolson scheme. The two-dimensional Forward Euler should be compared against other numerical methods, perhaps the Jacobi iterative method. It would also be of interest to see if we can observe any "inertia" in the two-dimensional methods, as well as any periodic bumps in the error.

8 Code

All code used to obtain the numerical results and to create the figures in this report can be found at erikasan@github.com

References

[1] Dr. Hilary Weller. *Numerics: The analysis and implementation of numerical methods for solving differential equations*. URL: `http://www.met.reading.ac.uk/~sws02hs/teaching/PDEsNumerics/PDEsNumerics_2_student.pdf`.