

Diffusion

Sebastian Amundsen, Marcus Berget and Andreas Wetzel

December 19, 2020

We found that the heat equation can be solved using the explicit forward method, the implicit backward Euler method and the Crank- Nicholson scheme. The stability of the explicit method is decided by the condition $\Delta t \leq \Delta x^2/2$. We found that the implicit methods were stable regardless of this condition. The Forward Euler method was the fastest numerical solver, being 76% and 75 % faster than Crank Nicholson and Backward Euler respectively.

1 Introduction

In this report we want to test three algorithms for solving partial differentiation equations using finite difference schemes. Specifically we want to solve the heat equation in one dimension using three different methods, where we compare our numerical results to an analytical solution. This is done by calculating the root mean square error and by looking at the deviation between numerical and analytical values for each of the algorithms.

We will also be looking at two different stability conditions, which will impact the stability of our explicit solver. We then expand the most suited numerical method for solving the heat equation to two dimensions. Afterwards we wish to simulate the temperature distribution in the lithosphere. This will showcase how an algorithm can be implemented numerically to solve a physical problem.

2 Theory

2.1 Heat equation

The general heat equation can be written as,

$$\frac{\partial T(\mathbf{x}, t)}{\partial t} = \frac{k}{C\rho} \nabla^2 T(\mathbf{x}, t). \quad (1)$$

where \mathbf{x} is the spatial vector, t is time, c_p is the specific heat capacity, ρ is the density and k is the thermal conductivity. We can then gather all the constants in the diffusion constant, $D = \frac{k}{C\rho}$. For the first part of

the project we will just set the diffusion constant equal to one. The heat equation in one dimension then becomes,

$$\frac{\partial T(x, t)}{\partial t} = \frac{\partial^2 T(x, t)}{\partial x^2}, \quad (2)$$

or

$$T_{xx} = T_t. \quad (3)$$

2.2 Numerical methods for solving the heat equation

We set the initial conditions of equation (2) at $t = 0$ to,

$$T(x, 0) = 0, \quad 0 < x < L \quad (4)$$

where $L = 1$ is the length of the x-region of interest. We set the boundary conditions to

$$T(0, t) = 0, \quad t \geq 0, \quad (5)$$

and

$$T(L, t) = 1, \quad t \geq 0. \quad (6)$$

Equation (2) with the mentioned initial conditions and boundary conditions can be solved numerically using the forward Euler method, the backward Euler method and the implicit Crank-Nicholson scheme.

2.2.1 Explicit forward Euler method

We proceed with equation (2) and the mentioned initial/boundary conditions. We define the step length for the spatial variable x ,

$$\Delta x = \frac{1}{n+1}. \quad (7)$$

The position after i steps and time after j steps are then given by,

$$\begin{aligned} t_j &= j\Delta t, \quad j \geq 0, \\ x_i &= i\Delta x, \quad 0 \leq i \leq n+1. \end{aligned}$$

By using the forward formula to approximate the derivatives we obtain

$$T_t = \frac{T_{i,j+1} - T_{i,j}}{\Delta t} \quad (8)$$

and

$$T_{xx} = \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2}. \quad (9)$$

Defining the value $\alpha = \Delta t / \Delta x^2$, the one-dimensional heat equation can be rewritten as

$$T_{i,j+1} = \alpha T_{i-1,j} + (1 - 2\alpha)T_{i,j} + \alpha T_{i+1,j}. \quad (10)$$

We can then see that since the initial conditions are known, one could use equation (10) to find the temperature in the next time step, which one could use to find the temperature after two time steps, and so on. This algorithm is an explicit scheme, since the temperature in the next time step is explicitly given.

2.2.2 Implicit backward Euler method

Here, we do just as for the forward Euler method, but instead of using the forward formula to approximate the first derivative, we use the backward formula,

$$T_t = \frac{T_{i,j} - T_{i,j-1}}{\Delta t}. \quad (11)$$

The spatial second derivative becomes just as for the explicit scheme,

$$T_{xx} = \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2}. \quad (12)$$

Again, by defining $\alpha = \Delta t / \Delta x^2$ we obtain

$$T_{i,j-1} = -\alpha T_{i-1,j} + (1 + 2\alpha)T_{i,j} - \alpha T_{i+1,j}. \quad (13)$$

The only unknown quantity in equation 13 is $T_{i,j-1}$, we can then define the matrix A ,

$$A = \begin{bmatrix} 1+2\alpha & -\alpha & 0 & \cdots & 0 \\ -\alpha & 1+2\alpha & -\alpha & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\alpha \\ 0 & 0 & \cdots & -\alpha & 1+2\alpha \end{bmatrix} \quad (14)$$

so that we can reformulate the problem as a matrix vector multiplication equation:

$$AV_j = V_{j-1} \quad (15)$$

We can then write the problem as:

$$\begin{aligned} V_j &= A^{-1}V_{j-1} = A^{-1}(A^{-1}V_{j-2}) \\ &= A^{-1}(A^{-1}(A^{-1}V_{j-3})) = A^{-j}V_0 \end{aligned} \quad (16)$$

we can then use the tridiagonal algorithm discussed in Appendix B.1 to solve this set of linear equations for every time step.

2.2.3 Crank-Nicholson scheme

The Crank-Nicholson scheme a bit more tedious to derive, so we will go ahead just give the approximations it uses for the time and spatial derivative:

$$T_t = \frac{T(x_i, t_j + \Delta t) - T(x_i, t_j)}{\Delta t} \quad (17)$$

and

$$\begin{aligned} T_{xx} &= \frac{1}{2} \left(\frac{T(x_i + \Delta x, t_j) - 2T(x_i, t_j) + T(x_i - \Delta x, t_j)}{\Delta x^2} \right. \\ &\quad \left. \frac{T(x_i + \Delta x, t_j + \Delta t) - 2T(x_i, t_j + \Delta t) + T(x_i - \Delta x, t_j + \Delta t)}{\Delta x^2} \right) \end{aligned} \quad (18)$$

We then combine equation 17 and 18, where the left side is given by $t_j + \Delta t$ and the right by t_j :

$$\begin{aligned} &-\alpha T(x_i - \Delta x, t_j + \Delta t) + (2 + 2\alpha)T(x_i, t_j + \Delta t) \\ &-\alpha T(x_i + \Delta x, t_j + \Delta t) = \alpha T(x_i - \Delta x, t_j) + (2 - 2\alpha)T(x_i, t_j) \\ &+ \alpha T(x_i + \Delta x, t_j) \end{aligned} \quad (19)$$

Where we have used that $\alpha = \frac{\Delta t}{\Delta x^2}$. We can write this as a tridiagonal matrix system:

$$(2I + \alpha B)V_j = (2I - \alpha B)V_{j-1} \quad (20)$$

We can rewrite equation 20 as:

$$V_j = (2I + \alpha B)^{-1}(2I - \alpha B)V_{j-1} \quad (21)$$

Where I is the identity matrix and B is given by:

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

The truncation error for the Crank Nicholson scheme goes as $O(\Delta t^2)$ and is stable for all combinations of Δx and Δy .

2.3 Analytical solution for one dimensional heat equation

It is beneficial to have a analytical solution which we can compare our numerical methods to. For the one dimensional diffusion equation we have the analytical expression:

$$u(x, t) = x/L + \sum_{n=1}^{\infty} \frac{2(\pi n \cos(\pi n) - \sin(\pi n))}{\pi^2 n^2} \sin(n\pi x/L) e^{-n^2 \pi^2 t/L} \quad (22)$$

Where $g(x)$ is a function of position. The calculations are given in appendix B.

2.4 Heating of the Fennoscandian lithosphere

The lithosphere is the solid upper-most part of the earth, consisting of the mantle, lower crust and upper crust. We will focus on an area of the earth's lithosphere located in Scandinavia, which we will call the fennoscandian lithosphere. About 1 Gy ago the oceanic lithosphere on the west coast of Norway was subducted under the fennoscandian lithosphere. This process released radioactive elements trapped in, and heating, the fennoscandian lithosphere.

2.4.1 Parameters of the model

The boundary conditions are 8°C at the surface and 1300°C at the bottom at a depth of 120 km. We will assume a constant density of $\rho = 3500 \text{ kg/m}^3$, a constant thermal conductivity of $\kappa = 2.5 \text{ W/m/}^\circ \text{C}$ and a constant specific heat capacity of $c_p = 1000 \text{ J/kg/}^\circ \text{C}$. Before the subduction of the oceanic lithosphere we

will assume that the fennoscandian lithosphere already contained radioactive elements causing the following heat production:

$$P(z, t) = \begin{cases} 1.4 \mu\text{W/m}^3 & 0 \text{ km} < z < 20 \text{ km} \\ 0.35 \mu\text{W/m}^3 & 20 \text{ km} < z < 40 \text{ km} \\ 0.05 \mu\text{W/m}^3 & z > 40 \text{ km} \end{cases}$$

for $t \in [0, 1]$ Gy. **It is worth noting that we here give the heat production in units of power, and not change in temperature per time unit.** We will assume that the additional heat production caused by the enrichment of radioactive elements at 1 Gy is $P_0 = 0.5 \mu\text{W/m}^3$. We assume that this additional heat is produced is at 40% by U, 40% by Th, and 20% by K, which have halflives of 4.47 Gy, 14.0 Gy, 1.25 Gy respectively. The reduction of radioactive sources over time will cause the additional heat production to decrease as function of time, which can be described as follows:

$$P_R(t) = P_0 \left(0.4 \left(\frac{1}{2} \right)^{t/4.47} + 0.4 \left(\frac{1}{2} \right)^{t/14.0} + 0.2 \left(\frac{1}{2} \right)^{t/1.25} \right) \quad (23)$$

3 Method

3.1 One dimensional model

We start by implementing our numerical methods in one dimension. We begin by initializing our boundary and initial conditions. We will be testing our algorithms with different spatial steps given $\Delta x = 1/10$ and $\Delta x = 1/100$, where Δt will be dictated by the stability limit of the explicit scheme. This limit is given $\alpha < 0.5$. We therefore need to adjust the duration of each time step and the number of time steps we are using to fulfill this explicit limit. We wish to study the algorithms during two different occasions. There will be an equilibrium time before our solution becomes stable. It is relevant to study the system during this equilibrium time t_1 and when the system is close to stationary t_2 . This will be done for each spatial step Δx . The truncation error should go according to Table 1.

3.1.1 Comparison between numerical and analytical model

We can test the stability and precision of our numerical algorithms by comparing our numerical approximations to our analytical solution given equation 22. We need to set N to something less than ∞ due to numerical limitations. We set $N=1000$. We compare our numerical and analytical results both visually by using plots and by studying the relative root mean square error (RRMSE). This will give us insight to how well our numerical approximations represent the one dimensional heat equation. We can then choose which of the three methods (Forward Euler, Backward Euler and Crank-Nicholson) best fit our analytical solution. The relative root mean square error is given by:

$$RRMSE = \sqrt{\frac{\frac{1}{n} \sum_{i=1}^n (u_{ana} - u_{num})^2}{\sum_{i=1}^n u_{ana}^2}} \quad (24)$$

Where u_{ana} is the analytical solution and u_{num} is the numerical approximation [4]. We will also be plotting the difference between the analytical solution and the values given the algorithms. This is to be done for both $\alpha > 0.5$ and $\alpha < 0.5$, which hopefully will showcase the stability of the implicit methods. To obtain the different values for α we adjust the duration of each time step in our simulation. The limit given time steps Δt and spatial steps Δx for the explicit method is:

$$\Delta t \leq \Delta x^2 / 2 \quad (25)$$

We wish to study the case where this limit is fulfilled and when it is unsatisfied.

3.2 Heating of the Fennoscandian lithosphere

We will assume that the fennoscandian lithosphere will be affected equally across the whole area surface, so that the depth is the only spatial dimension affecting the time evolution of the temperature. We also have to take in to account the heat production caused by radioactive elements. This translates to the heat equation as:

$$\frac{\partial T}{\partial t} = \frac{P}{\rho c_p} + \frac{\kappa}{\rho c_p} \frac{\partial^2 T}{\partial z^2} \quad (26)$$

where z is the depth. We set $D = \frac{\kappa}{\rho c_p}$ and introduce $z = a\hat{z}$, so that the heat equation becomes

$$\frac{\partial T}{\partial t} = \frac{P}{\rho c_p} + \frac{1}{Da^2} \frac{\partial^2 T}{\partial \hat{z}^2}. \quad (27)$$

We then choose $Da^2 = 1$ so that the final heat equation becomes,

$$\frac{\partial T}{\partial t} = \frac{P}{\rho c_p} + \frac{\partial^2 T}{\partial \hat{z}^2}. \quad (28)$$

By using the exact same approximations as for the forward Euler method, we have an explicit expression for temperature after in the next time step:

$$T_{i,j+1} = \frac{P}{\rho c_p} \Delta t + \alpha T_{i-1,j} + (1 - 2\alpha) T_{i,j} + \alpha T_{i+1,j} \quad (29)$$

where $\alpha = \frac{\Delta t}{\Delta x^2}$. When initializing the temperature we will consider the state without radioactive elements, but in thermal equilibrium. The temperature distribution must therefore be linear:

$$T(z) = 10.77z + 8.0 \quad (30)$$

when z is in units km. We then start time evolving the temperature as explained above, with the relevant parameters.

4 Implementation

All algorithms are implemented in the C++ language. All our data are written to textfiles and read by python scripts. These scripts are used for analyzation and visualization of our numerical data. Our codes are available at the GitHub repository: https://github.com/Sebamun/FYS3150_Projekter/tree/master/Project_5.

5 Results

5.1 One dimensional model

The analytical solution for different time occasions (t_1 and t_2) and spatial steps are given in Figure 1. The simulation time for each of the different algorithms are given in Table 2.

The numerical solution for different step lengths and time periods are given in Figure 2.

The deviation from the analytical solution for the three methods are given in Figure 3 and 5 for $\Delta x = 1/10$. The deviation for $\Delta x = 1/100$ is given in Figure 4 and 6. The RRMSE values are given in Table 3.

6 Discussion

6.1 One dimensional model

There is little variance between the three algorithms when looking at the one dimensional case (see Figure 2). All of the curves seem to evolve similarly. We have a period where the solution moves toward an equilibrium state and a solution given this equilibrium state, which seems to be linear. This development also corresponds with the analytical solutions given in Figure 1, which indicates that we have implemented the algorithms correctly.

The RRMSE values for $t_1 = 0.01$ were smallest for the Backward Euler method and highest for the Forward Euler method, when looking at $\Delta x = 1/10$. For $t_1 = 0.01$ and $\Delta x = 1/10$ the RRMSE values were highest for the Crank Nicholson scheme and smallest for the Backward Euler scheme. This is surprising given the fact that the Crank Nicholson algorithm has an higher order of truncation error than both the backward and forward method (Table 1). One explanation could be the amount of FLOPS for each method. The Crank Nicholson method has more calculations than both the other algorithms, which makes room for round off errors. It is also possible that either the algorithms, the analytical solution or the RRMSE method are not implemented correctly.

The differences in RRMSE were not detectable for t_2 (4). This is to be expected given that all three algorithms should move towards the same final solution (as long as the stability condition for the explicit method is satisfied). We can see that the increased spatial resolution (decreasing Δx) leads to better precision for all the methods, due to the error becoming smaller for $\Delta x = 1/100$ than for $\Delta x = 1/10$. This is true for both t_1 and t_2 .

The forward Euler method has the shortest simulation time by far, which make it time efficient when solving large simulations (Table 2). When $\alpha < 0.5$ the deviation from the analytical solution is indistinguishable for the three methods when we are looking at Figure 3 and 4. Based on these factors, it seems that Forward Euler method is the best algorithm for solv-

ing the one dimensional problem as long as the stability condition is fulfilled.

However, when $\alpha > 0.5$ the error for the explicit method (Forward Euler) "skyrockets" (Figure 5 and Figure 6). For the implicit schemes we don't have the same problem. The implicit solvers maintain stable at all values for α . It is therefore necessary to choose the numerical method based on the problem you wish to solve. The Forward Euler method might not be suitable when we are required to use larger time-steps in our simulation.

6.2 Heating of the Fennoscandian lithosphere

Figure (BLAH) tells us a lot about how the temperature is affected by the radioactive elements, depending on the depth. We can see that in the very beginning the temperature increases the most for the depths in the middle, before the temperature reaches equilibrium. For the depth of 50 km the temperature increase is significantly bigger than for the depth of 10 km. This is in spite of the heat production being 28 times bigger at $z = 10$ km, compared to the heat production at $z = 90$ km to the surface. However, this does make sense. Even though the heat production close to the surface is high, it doesn't have time to increase the temperature before the heat disappears into the cold surface.

We can then see that after 1 Gy, the oceanic lithosphere gets subducted and the additional radioactive elements start kicking in. As we would expect, the temperature has a sudden spike at $t = 1$ Gy before it reaches equilibrium and starts declining. The reason for the decline is of course the radioactive elements start decaying. We would then expect that after a very long time that the temperature would go back to same as before the additional radioactive elements started kicking in.

In figure (HEAT) we can see the temperature distribution across the depth for different times. for $t = 0$ Gy we can see that the temperature is linear, as expected. We can then see that after $t = 0.25$ Gy the temperature has reached close to equilibrium, given that there is at least a graphically small difference between the temperature distribution at $t = 0.25$ Gy and $t = 1.0$ Gy. The temperature distributions at the other times only further confirm what we discussed for figure (BLAH). Though it is arguably visually pleasing, figure (BLAH) doesn't seem to add any information

other from what is already discussed.

7 Concluding remarks

We have implemented all three algorithms numerically and compared them with the analytical solution. We found that the implicit Backward Euler method

had the least relative root mean square error and that the explicit Forward Euler method had the fastest simulation time. We studied the explicit method at the stability boundary and found that the assumption of instability at $\Delta t \geq \Delta x^2/2$ was correct. Simulation time, simplicity and relatively insignificant truncation error led us to choose the Forward Euler method for solving the 2D case.

A Appendix

A.1 Figures

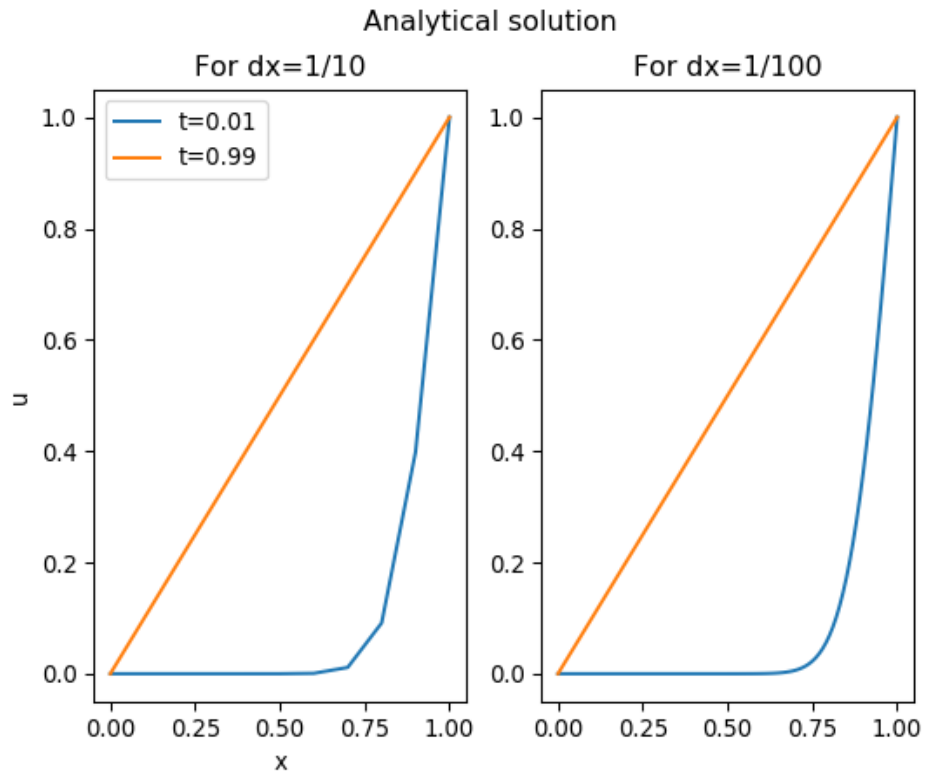


Figure 1: Analytical solution for $\Delta x = 1/10$ and $\Delta x = 1/100$.

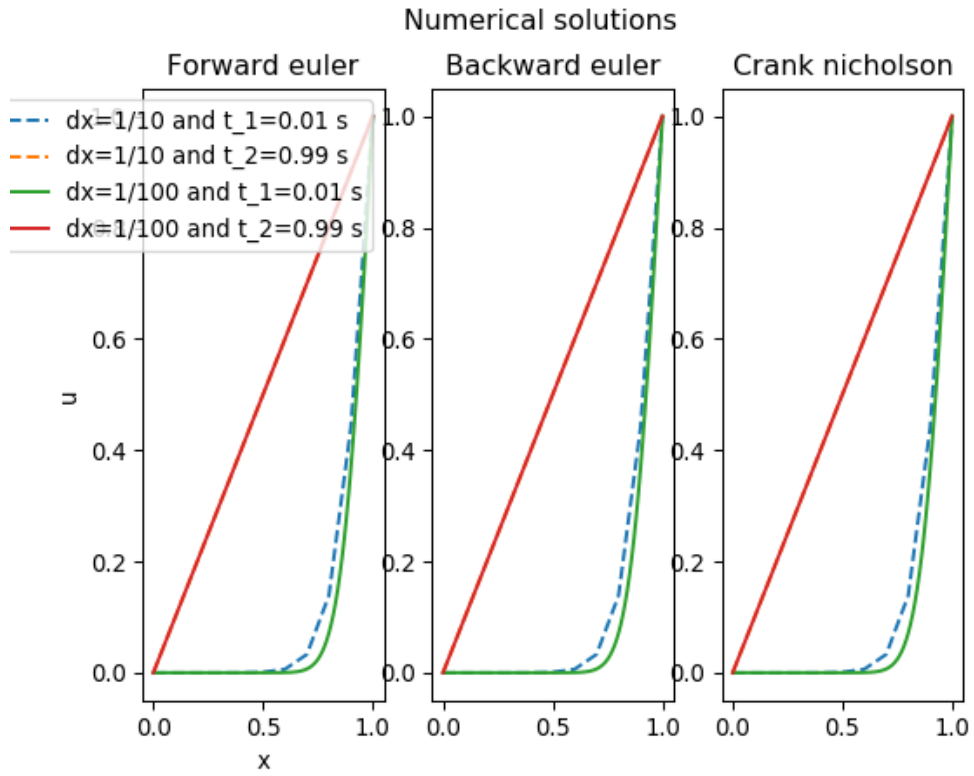


Figure 2: The numerical solution given different times t_1 and t_2 and spatial steps $\Delta x = 1/10$ and $\Delta x = 1/100$.

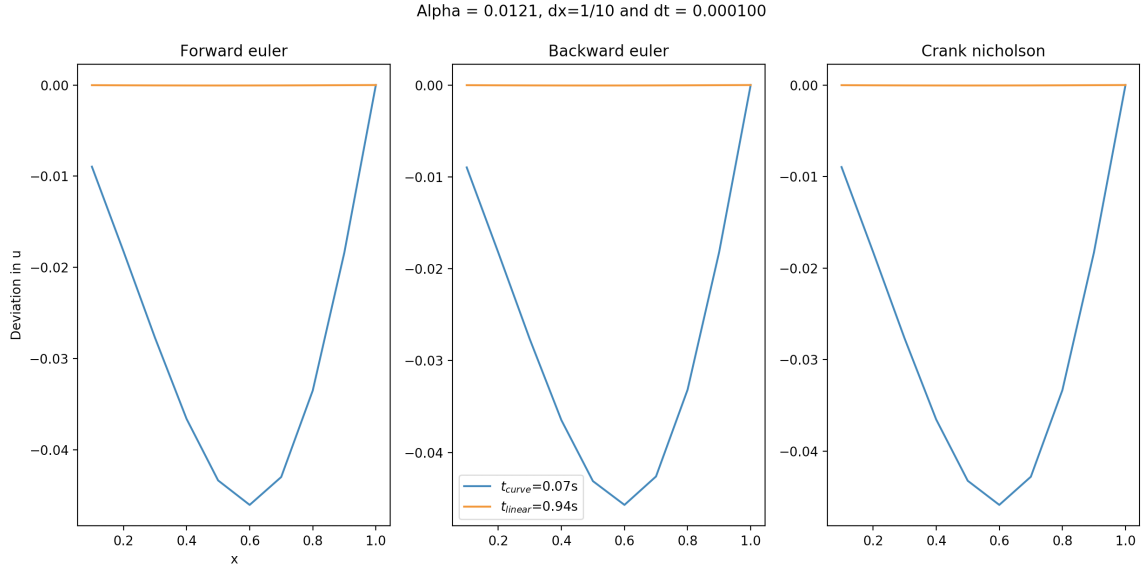


Figure 3: Deviation from analytical solution for $\alpha = 0.0121$, $\Delta x = 1/10$ and $\Delta t = 0.0001$. The blue line $t_{curve} = 0.07$ is the error after a short time period, where the solution is curved. The orange line $t_{linear} = 0.94$ is the error after a longer time period, where the solution is linear.

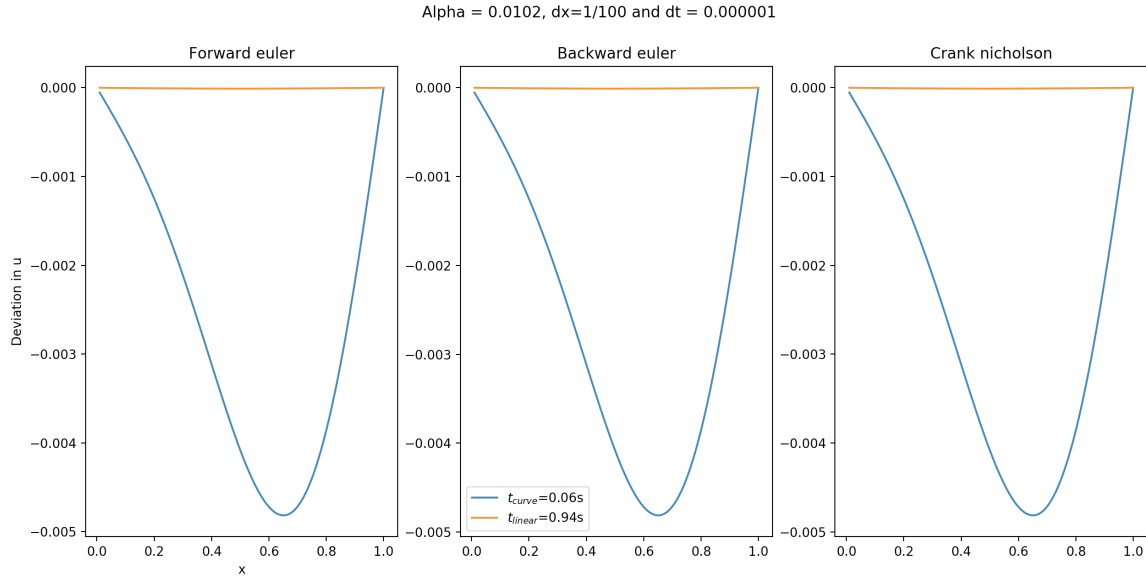


Figure 4: Deviation from analytical solution for $\alpha = 0.0102$, $\Delta x = 1/100$ and $\Delta t = 0.000001$. The blue line $t_{curve} = 0.06$ is the error after a short time period, where the solution is curved. The orange line $t_{linear} = 0.94$ is the error after a longer time period, where the solution is linear.

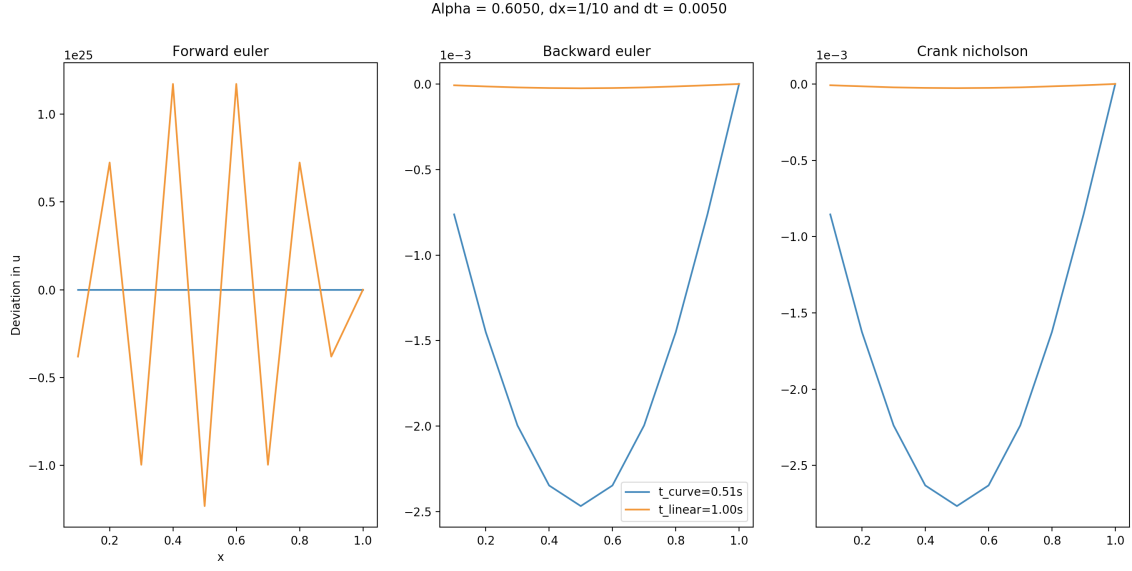


Figure 5: Deviation from analytical solution for $\alpha = 0.6050$, $\Delta x = 1/10$ and $\Delta t = 0.005$. The blue line $t_{curve} = 0.51$ is the error after a short time period, where the solution is curved. The orange line $t_{linear} = 1.0$ is the error after a longer time period, where the solution is linear.

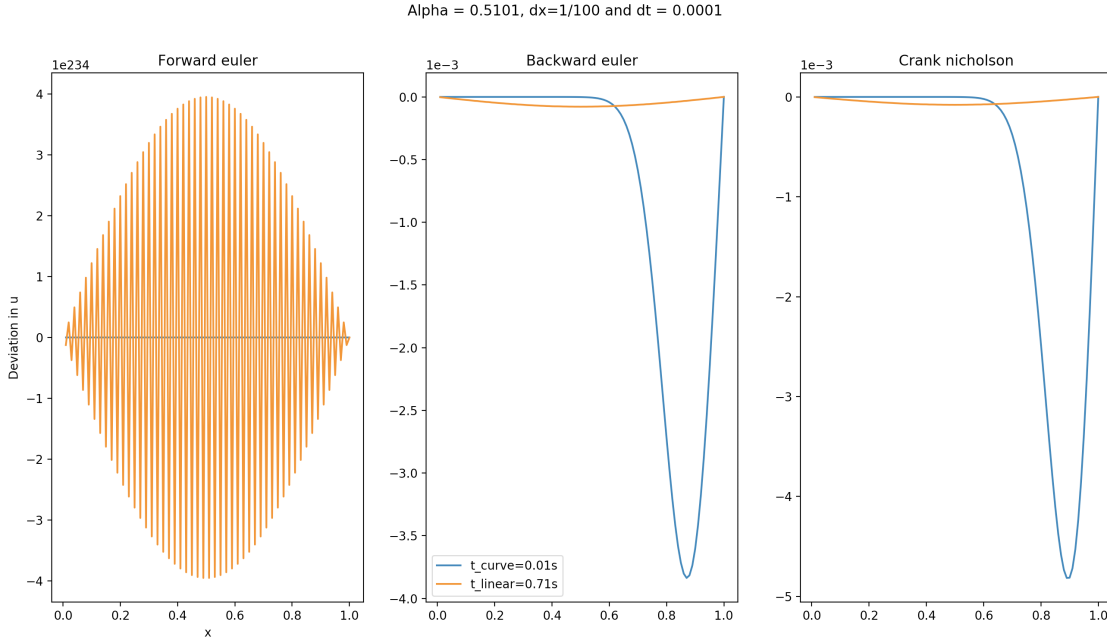


Figure 6: Deviation from analytical solution for $\alpha = 0.5101$, $\Delta x = 1/100$ and $\Delta t = 0.0001$. The blue line $t_{curve} = 0.01$ is the error after a short time period, where the solution is curved. The orange line $t_{linear} = 0.71$ is the error after a longer time period, where the solution is linear.

A.2 Tables

Table 1: Theoretical truncation errors.

Scheme:	Truncation error	Stability requirments
Crank Nicholson	$O(\Delta x^2)$ and $O(\Delta t^2)$	Stable for all
Backward Euler	$O(\Delta x^2)$ and $O(\Delta t)$	Stable for all
Froward Euler	$O(\Delta x^2)$ and $O(\Delta t)$	$\Delta t \leq 1/2\Delta x^2$

Table 2: Simulation time for algorithms (1D case).

Scheme:	$\Delta x = 1/10$	$\Delta x = 1/100$
Crank Nicholson	0.019732s	14.9372s
Backward Euler	0.021026s	14.5824s
Froward Euler	0.00865s	3.58414s

Table 3: RRMSE for the different algorithms for $t_1 = 0.01$.

Scheme:	$\Delta x = 1/10$	$\Delta x = 1/100$
Crank Nicholson	0.034560	0.0038095
Backward Euler	0.034521	0.0038079
Froward Euler	0.034599	0.0038086

Table 4: RRMSE for the different algorithms for $t_2 = 0.99$.

Scheme:	$\Delta x = 1/10$	$\Delta x = 1/100$
Crank Nicholson	9.5847294×10^{-5}	6.7524668×10^{-5}
Backward Euler	9.5847294×10^{-5}	6.7524668×10^{-5}
Froward Euler	9.5847294×10^{-5}	6.7524668×10^{-5}

B Appendix

B.1 General algorithm for tri-diagonal matrix

There exists an algorithm for solving generic sets of linear equations, but in the case for equation (??) where we have a tri-diagonal matrix we can use a different algorithm which decreases the amount of floating point operations needed. We denote the elements in the leading diagonal of A as b_1, b_2, \dots, b_n , the elements above the leading diagonal as a_2, a_3, \dots, a_n , and the elements below the leading diagonal as c_1, c_2, \dots, c_{n-1} . The algorithm uses a forward substitution to replace the leading diagonal with elements denoted by \tilde{b}_i , and replaces the righthand side in with the elements \tilde{f}_i as shown below[REFERENCE TIL PROSJEKT 1]:

$$\begin{aligned}\tilde{b}_i &= b_i - \frac{a_i c_{i-1}}{\tilde{b}_{i-1}} \\ \tilde{f}_i &= f_i - \frac{a_i \tilde{f}_{i-1}}{\tilde{b}_{i-1}}\end{aligned}$$

where $\tilde{b}_1 = b_1$ and $\tilde{f}_i = f_i$. The algorithm then continues with a backward substitution which gives the solution:

$$u_{i-1} = \frac{\tilde{f}_{i-1} - c_{i-1} u_i}{\tilde{b}_{i-1}} \quad (31)$$

B.2 Analytical expression for 1D diffusion equation

The one dimensional diffusion equation has an analytical expression for the continuous problem.

$$\nabla^2 u(x, t) = \frac{\partial u(x, t)}{\partial t} \quad (32)$$

The initial condition is $u(x, 0) = g(x)$, when $0 < x < L$. We also have $u(0, t) = 0$ and $u(L, t) = 0$, when $t \geq 0$. We use separation of variables when solving 32:

$$u(x, t) = F(x)G(t) \quad (33)$$

This gives us one function only depending on position x and one only depending on time t. We get the set of equations:

$$\frac{\partial u}{\partial t} = F \frac{\partial G}{\partial t}$$

and

$$\frac{\partial^2 u}{\partial x^2} = G \frac{\partial^2 F}{\partial x^2}$$

We can now use the relation given equation 2:

$$\begin{aligned}F \frac{\partial G}{\partial t} &= G \frac{\partial^2 F}{\partial x^2} \\ \frac{1}{G} \frac{\partial G}{\partial t} &= \frac{1}{F} \frac{\partial^2 F}{\partial x^2}\end{aligned} \quad (34)$$

We set the equations equal to a negative constant $-\lambda^2$. This gives us the solutions:

$$\begin{aligned} G &= Ae^{-\lambda^2 t} \\ F &= B \cos(\lambda x) + C \sin(\lambda x) \end{aligned}$$

We can now use equation 33 to find the general solution u :

$$\begin{aligned} u &= GF = Ae^{-\lambda^2 t} [B \cos(\lambda x) + C \sin(\lambda x)] \\ &= e^{-\lambda^2 t} [C_1 \cos(\lambda x) + C_2 \sin(\lambda x)] \end{aligned} \quad (35)$$

The boundary conditions gives us the solution:

$$u(x, t) = A_n \sin(n\pi x/L) e^{-n^2 \pi^2 t/L} \quad (36)$$

We have infinitely many solutions to this equation, due to the n factor. We can use a superposition of these solutions since the diffusion equation is linear:

$$u(x, t) = \sum_{n=1}^{\infty} A_n \sin(n\pi x/L) e^{-n^2 \pi^2 t/L} \quad (37)$$

We decide the coefficient A_n by using the initial condition:

$$u(x, 0) = g(x) = \sum_{n=1}^{\infty} A_n \sin(n\pi x/L) \quad (38)$$

Which gives us that:

$$A_n = \frac{2}{L} \int_0^L g(x) \sin(n\pi x/L) dx$$

We need to find the function $g(x)$. We should obtain an equilibrium temperature when the time goes to infinity. This can be written as:

$$\lim_{t \rightarrow \infty} u(x, t) = u_E(x)$$

Where $u_E(x)$ is the equilibrium temperature. We can look at how this function behaves given the boundary conditions:

$$\frac{d^2 u_E}{dx^2} = 0 \quad u_E(0) = 0 \quad u_E(L) = 1$$

This gives us the general solution:

$$u_E(x) = k_1 x + k_2$$

Where k_1 and k_2 are constants. We get the function $u_E(x) = x/L$. Now we define a new function $v(x, t) = u(x, t) - u_E(x)$, where $u(x, t)$ is the solution to equation 37, which we wish to solve for:

$$u(x, t) = v(x, t) + u_E(x) \quad (39)$$

We need to know the boundary and initial conditions for $v(x, t)$:

$$\begin{aligned}
v(x,0) &= u(x,0) - u_E(x) = 0 - x/L = -x/L \\
v(0,t) &= u(0,t) - u_E(0) = 0 - 0 = 0 \\
v(L,t) &= u(L,t) - u_E(L) = 1 - 1 = 0
\end{aligned}$$

We use the same method as earlier and find the solution for $v(x,t)$:

$$v(x,t) = \sum_{n=1}^{\infty} B_n \sin(n\pi x/L) e^{-n^2\pi^2 t/L} \quad (40)$$

Where the coefficient B_n is given by:

$$B_n = \frac{2}{L} \int_0^L g(x) \sin(n\pi x/L) dx \quad (41)$$

Now we can use the fact that $g(x)=v(x,0)=-x/L$ and equation 39:

$$u(x,t) = x/L + \sum_{n=1}^{\infty} \frac{2}{L} \int_0^L (-x/L) \sin(n\pi x/L) dx \sin(n\pi x/L) e^{-n^2\pi^2 t/L} \quad (42)$$

We solve the integral and find the final solution:

$$\begin{aligned}
\int_0^L (-x/L) \sin(n\pi x/L) dx &= \frac{L(\pi n \cos(\pi n) - \sin(\pi n))}{\pi^2 n^2} \\
\Rightarrow u(x,t) &= x/L + \sum_{n=1}^{\infty} \frac{2(\pi n \cos(\pi n) - \sin(\pi n))}{\pi^2 n^2} \sin(n\pi x/L) e^{-n^2\pi^2 t/L}
\end{aligned}$$

We now have a analytical expression for heat diffusion in one dimension [1][3].

References

- [1] Jensen, M.H., 2015, Computational Physics Lecture Notes Fall 2015
- [2] Jensen, M.H., 2017, Computational Physics Lectures: Statistical physics and the Ising Model
- [3] Dawkins, Paul., 2018, Heat Equation With Non-Zero Temperature Boundaries
- [4] Despotovic, M, et al., 2016, Evaluation of empirical models for predicting monthly mean horizontal diffuse solar radiation
- [5] Daileida, R., 2012, The two dimensional heat equation