

# A Numerical Analysis on Partial Differential Equations with Attempts at Applications in Geophysical Processes

Jan Egil Ødegård

(Dated: December 18th, 2019)

The source code files can be found in my [GitHub Repository](#)

For this report, I have looked closer at a general partial differential equation (PDE) and looked at different methods of solving these numerically given a set of boundary and initial conditions. For a PDE in 1 spatial dimension, I have looked at three different schemes for solving them numerically; an explicit scheme, an implicit scheme, and what's known as an 'implicit Crank-Nicolson' scheme. I've looked closer at both the efficiency and relative error when using these methods compared to an analytically calculated value, where the explicit scheme ended up proving both more accurate and efficient than its implicit counterparts, though having a limitation in the ratio between spatially and temporal step sizes. Moving further, I've also extended the explicit scheme to 2 dimensions, taking into account the limitations that this method gives us. For a two-dimensional unitless system, the simulations returned sensible results as well. There has also been an attempt at applying this at the heat equation for the temperature distribution in the lithosphere of the Earth, though this has been without success. This likely stems from either an error in the initial logic of the algorithm, a bug in the program that has been written, or in the subsequent plotting.

## I. INTRODUCTION

Partial differential equations, or PDEs for short, are sets of differential equations in which we need to solve for more than one variable. Sometimes, these are straightforward to solve analytically, with closed form solutions, but more often than not, PDEs tend to be difficult or impossible to solve analytically. Using numerical approximation methods however, it is possible to construct a somewhat good solution regardless of the complexity of the problem at hand.

In this article, we have attempted to solve a set of PDEs using different methods, and we've looked at the limitations and requirements that these methods give us. We've first looked closer at a 1 dimensional-case with (unitless) time and a spatial coordinate as the unknowns. We've used 3 different methods to attack this, an explicit method, primarily using the forward-Euler algorithm, an implicit method, primarily using the backward-Euler algorithm, and a method known as the implicit Crank-Nicolson, which uses a combination of forward and backward Euler approximations.

We've also extended this into a 2-dimensional spatial system, applying the use of the explicit method. We will finish this off with attempting to solve the 2 dimensional heat equation for a geophysical system in the lithosphere numerically. This will also be done using the explicit scheme.

## II. THEORY

We are first looking closer at a general case of a partial differential equation. This is followed by taking a closer look at an application of partial differential equations, namely the heat equation. We finish this with applying the heat equation to a model that we construct of the lithosphere of the Earth, with the goal of looking at its temperature distribution.

### A. Partial Differential Equations

#### 1. General Partial Differential Equations

A general PDE typically has a variable  $u$  which is dependent on multiple other variables  $u(x_1, \dots, x_n)$ , in which we take the derivative of some or all of these variables. For this article, we will in general be focusing on separable PDEs taking the form:

$$\nabla^2 u(\vec{r}, t) = \frac{\partial u(\vec{r}, t)}{\partial t}, \quad (1)$$

where our unknown is a function of a spatial term and a temporal one. More specifically, we will be working in 1 and 2 spatial dimensions in a Cartesian coordinate system, meaning that these equations in general will take the form:

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

$$\frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2} = \frac{\partial u(x, y, t)}{\partial t}. \quad (3)$$

When solving equations 2 or 3 either numerically or analytically, we require that we know both the boundary

conditions of the system ( $T(0, t)$  and  $T(L, t)$  with  $x \in [0, L]$ ), as well as the initial condition of the system ( $T(x, 0)$ ).

We will also look closer at the heat equation, which is a form of PDE with dimensionality included.

## 2. The Heat Equation

The heat equation (in 2 spatial dimensions) is given as [1]:

$$\nabla(k\nabla T) + Q = \rho c_p \frac{\partial T}{\partial t}, \quad (4)$$

Here,  $k$  is the thermal conductivity, given in units of  $\text{W/m}^\circ\text{C}$ , we have  $Q$  as the heat production, given in units of  $\text{W/m}^3$ ,  $\rho$  is the density, given in  $\text{kg/m}^3$  and  $c_p$  is the specific heat capacity, given in units of  $\text{J/kg}^\circ\text{C}^{-1}$ . The temperature  $T$  is a variable of 3 unknowns, which are the two spatial dimensional variables  $x$  and  $y$ , as well as a temporal variable  $t$ . This gives us that  $T \rightarrow T(x, y, t)$ .

## 3. Expected Behaviour: Steady State

Given a regular system like the ones seen in equations 2 and 3, we want to figure out what the expected behaviour of the system will look like. This in general will depend on the initial conditions, as well as the boundary conditions of our system. For simplicity, we will only look at the 1 dimensional case, but it can be generalized to more dimensions. We will also be looking at the case where the initial conditions will be 0 ( $u(x, 0) = 0$  for all  $x$ ), and with boundary conditions set as  $u(0, t) = 0$ , and  $u(L, t) = 1$ .

Looking at this system after some time  $t$  has elapsed, such that it has reached its steady state, we will be interested in how we expect the system to look like. For starters, in a system that has reached its steady state, we would expect that the following holds true:

$$\frac{\partial u}{\partial t} = 0, \quad (5)$$

This is the case because any deviation away from said steady state would imply that we won't be in the steady state after all. This leaves us with the equation

$$\frac{\partial^2 u(x, t \rightarrow \infty)}{\partial x^2} = 0 \quad (6)$$

This is the 1D Laplace operation, which returns us an equation on the form  $u(x, t \rightarrow \infty) = ax + b$ . Assuming that the function  $u(x, t)$  is separable such that  $u(x, t) = V(x)S(t)x + F(x)G(t)$ . Now, solving these for the boundary conditions,  $u(0, t) = 0 \implies F(x)G(t) = 0$ , and  $u(L, t) = 1 \implies V(x)S(t) = \frac{1}{L}$ . Implementing this,

we can see that we in such a system could expect the following general behaviour:

$$u(x, t \rightarrow \infty) = \frac{x}{L}. \quad (7)$$

We therefore simply expect a linear distribution of values for the 1D case. This is perfectly possible to expand to 2D, where the general expression will be linear for  $x$  and  $y$  values each on their own (Given equal boundary conditions for the  $x$ - and  $y$ -axes).

For the heat equation in equation 4, it is more difficult to directly say what the expected steady state behaviour will look like, considering that we have several variables that can both vary spatially and temporally.

## B. The Lithosphere

The lithosphere is, simply described, the solid, outermost sphere of a planet. For the Earth, this includes the crust, as well as the outermost solid part of the mantle.

### 1. Parameters

For our model of the Earth's lithosphere, we will assume that it will have a constant density of  $\rho = 3.5 \cdot 10^{12} \text{ kg/km}^3$ , a constant thermal conductivity  $k = 2500 \text{ W/km}^\circ\text{C}$  and a constant specific heat capacity  $c_p = 1000 \text{ J/kg}^\circ\text{C}^{-1}$ . Since we are working with a constant thermal conductivity  $k$ , we can plug this outside the first gradient in equation 4, which gives us:

$$k\nabla^2 T + Q = \rho c_p \frac{\partial T}{\partial t}. \quad (8)$$

Since we know that this is in 2 dimensions, we can write out the gradient in equation 8, which gives us the following equation:

$$\left[ \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right] + \frac{Q}{k} = \frac{\rho c_p}{k} \frac{\partial T}{\partial t}. \quad (9)$$

As the boundary conditions, we will set the temperature of the top of the lithosphere to  $T(0) = 8^\circ\text{C}$ , and at the bottom of the lithosphere, which is  $\sim 120 \text{ km}$  deep, to  $T(120) = 1300^\circ\text{C}$ . (We define down towards the core as the positive direction). We will model the heat production  $Q$  to be dependent on depth  $y$ . This is because the heat production is directly linked with radioactive decay in the lithosphere, and radioactive isotopes are much more abundantly found closer to the surface as opposed to deeper down.

$$Q(y) = \begin{cases} 1.4 \text{ kW/km}^3 & y \in [0, 20] \text{ km} \\ 0.35 \text{ kW/km}^3 & y \in (20, 40] \text{ km} \\ 0.05 \text{ kW/km}^3 & y \in (40, 120] \text{ km} \end{cases} \quad (10)$$

It is nice to note that we for simplicity use  $km$  as a length unit here. We now rewrite equation 9 in terms of the parameters being variables or constant:

$$\left[ \frac{\partial^2 T(x, y, t)}{\partial x^2} + \frac{\partial^2 T(x, y, t)}{\partial y^2} \right] + \frac{Q(y)}{k} = \frac{\rho c_p}{k} \frac{\partial T(x, y, t)}{\partial t}. \quad (11)$$

The mantle "box" that we will be modelling will be 120 km deep ( $y \in [0, 120]$  km) and be 350 km wide ( $x \in [0, 300]$  km).

## 2. Radioactive Enrichment

We will attempt to model the remains of a geophysical process that occurred roughly 1 billion years ago, or 1 *Gy* (Gigayear) ago, and see to what extent this proposed process has on the temperature distribution in the lithosphere at present day. The process in question creates an enrichment in the lithospheric mantle of radioactive isotopes like uranium (*U*), potassium (*K*) and thorium (*Th*). The enrichment will happen in a 150 km wide area of the mantle, and only enrich the mantle with radioactive elements ( $y \in [40 - 120]$  km).

Initially, the system will be in a steady state (for our numerical simulation coming up, we will have to wait for the system to reach the steady state). and following this, our mantle will be enriched with these radioactive elements. The way this will be modelled is by increasing the heat production in the mantle part of the lithosphere with an additional  $Q_{add} = 0.5 \text{ kW/km}^3$ . Initially, we will model the radioactive content without taking into account the radioactive decay. This means that the heat production  $Q$  in the mantle will stay constant, and only change during the enrichment transition.

The final change we will do to this model, is to include the decay of the radioactive isotopes into the calculations. The half lives and the heat production ratios can be seen in table I.

Element	Half life [ <i>Gy</i> ]	Initial heat production
Uranium [ <i>U</i> ]	4.47	40%
Thorium [ <i>Th</i> ]	14.0	40%
Potassium [ <i>K</i> ]	1.25	20%

Table I. Half lives and the initial heat production percentage of the different enriching elements found in the mantle after enrichment. The percentages indicate only the distribution of the initial heat production, and as time progresses, the ratio of heat production in regards to the others will change depending on the half life values of that given isotope.

Using the values we have in table I, we can construct an expression for the heat production in the enriched part of the mantle as a function of time. In general, the content of a given substance with a half life of  $t_h$

after a given time  $t$  is given as  $N = N_0(1/2)^{t/t_h}$ . Since we initially have a production of  $0.5 \text{ kW/km}^3$ , the heat production will therefore be given as

$$Q_{add}(t) = 0.05 + 0.5 \left[ 0.4 \left( \frac{1}{2} \right)^{t/t_{[U]}} + 0.4 \left( \frac{1}{2} \right)^{t/t_{[Th]}} + 0.2 \left( \frac{1}{2} \right)^{t/t_{[K]}} \right] \quad (12)$$

Where  $t_{[U]}$ ,  $t_{[Th]}$  and  $t_{[K]}$  are the half lives of uranium, thorium and potassium respectively. Equation 12 is simply derived from taking the half lives of the individual substances, together with their initial relative contribution to the heat production at time  $t = 0$ . This gives us the final expression for the heat production as being the following:

$$Q(y) = \begin{cases} 1.4 \text{ kW/km}^3 & y \in [0, 20] \text{ km} \\ 0.35 \text{ kW/km}^3 & y \in (20, 40] \text{ km} \\ Q_{mantle}(x, t) & y \in (40, 120] \text{ km} \end{cases}, \quad (13)$$

With  $Q_{deep}(x, t)$  is defined as:

$$Q_{mantle}(x, t) = \begin{cases} Q_{add}(t) & x \in [100, 250] \text{ km} \\ 0.05 \text{ kW/km}^3 & \text{else} \end{cases}, \quad (14)$$

and  $Q_{add}(t)$  being what we defined in 12. This will be our final expression for the additional heat production in the lithosphere. We are only interested in the additional heat production in a 150 km wide area of the mantle, but we will also model the box as being 350 km wide. This is done so that we can see the effect that this heat production has in a broader region outside the radioactively enriched area.

## 3. Expected Behaviour: Steady State

As for the expected behaviour of the temperature distribution, we will expect that it will after a while reach a steady state. This steady state will be different given the different systems we are looking at, but in general they will follow the same linear relationship  $T(y, t \rightarrow \infty) = ay + b$  as in section II A 3. (Just with  $y$  instead of  $x$ , since depth is given by  $y$ ). The model is split into three different sections, with the upper crust, the lower crust and the solid part of the mantle.

Pre-enrichment, we would expect that the differing heat production in each of the different regions to dictate the slope of the linear function. With a uniform heat production  $Q$ , we would expect the slope to be perfectly linear all the way through the lithosphere. Since we have more heat production in the upper parts, we would expect a larger increase in the temperature as a function of depth for the upper crust as compared to

the lower crust, and a larger increase in the lower crust as compared to the lithospheric mantle. If  $a_U$  is the slope of the temperature as a function of depth in the upper crust,  $a_L$  is for the lower crust, and  $a_M$  is for the lithospheric mantle, we would expect the following to hold true:

$$a_U > a_L > a_M. \quad (15)$$

Mind, this is given the coordinate system where we defined the y-axis to point downwards, so that an increase in depth is given as the positive direction.

Post-enrichment, things will grow more complicated considering that we will have a region of the mantle that will have a larger heat production as compared to another region. Since we also get a dependency in the x-direction as well as the y-direction, we can only reason qualitatively as to the expected behaviour, and not so much get a quantified answer. First of all, we would expect that the additional heat production in the mantle to radiate and diffuse out into both the crust, as well as to the non-enriched parts of the mantle. This, coupled with the fact that the less-heat producing part of the mantle effectively cooling the enriched part, will make this model of the lithosphere a bit more complex.

Finally, when taking into account the half life, we would expect that this will slowly bring our system back to the original steady state as first discussed. Considering that the half lives of the substances are minimal as compared to the time scale of our simulation of roughly 1 billion years (ref table I), our modelled simulation will not get back to the steady state as a whole, but will only gradually get there.

### III. ALGORITHM

In general, we will be looking at the partial differential equation problem on the form,

$$\nabla^2 u(\vec{r}, t) = \frac{\partial u(\vec{r}, t)}{\partial t}, \quad (16)$$

where  $u(\vec{r}, t)$  is the function value at a given point  $\vec{r}$  at a given time  $t$ . For the 1-dimensional case, the left side simplifies, and we get the expression on the following form:

$$\frac{\partial^2 u(\vec{r}, t)}{\partial x^2} = \frac{\partial u(\vec{r}, t)}{\partial t}. \quad (17)$$

For the 1-dimensional case, we will use three different approaches to solving this problem. The derivations of these schemes can be found in Appendix A.

#### A. The Numerical Schemes

##### 1. The Explicit Scheme

The explicit scheme is an implementation based on the usage of the forward-Euler algorithm. The time evolution is thus given by a standard first order forward Euler, on the form:

$$\frac{\partial u}{\partial t} = u_t \approx \frac{u(x_i, t_j + \Delta t) - u(x_i, t_j)}{\Delta t}. \quad (18)$$

To simplify notation, we will write  $u(x_i, t_j) = u_{i,j}$ ,  $u(x_i, t_j + \Delta t) = u_{i,j+1}$  and  $u(x_i + \Delta x, t_j) = u_{i+1,j}$ . Resulting in the following equation:

$$u_t \approx \frac{u_{i,j+1} - u_{i,j}}{\Delta t}. \quad (19)$$

The spatial differential will be given by the second order forward Euler scheme, which is, when using the same notation as above, given as follows:

$$\frac{\partial^2 u}{\partial x^2} = u_{xx} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2}. \quad (20)$$

If we now take the original equation given in equation 17, and plug in equations 19 and 20, we get:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} = \frac{u_{i,j+1} - u_{i,j}}{\Delta t}. \quad (21)$$

Defining a variable  $\alpha = \frac{\Delta t}{(\Delta x)^2}$ , and rearranging the above expression to solve for the next time step  $u_{i,j+1}$ , we get:

$$u_{i,j+1} = \alpha u_{i-1,j} + (1 - 2\alpha)u_{i,j} + \alpha u_{i+1,j}. \quad (22)$$

When we know the boundary conditions ( $u_{0,j}$  and  $u_{L,j}$  where  $i \in [0, L]$ ), as well as the initial values ( $u_{i,0}$ ), we have an explicit expression in equation 22 to calculate  $u_{i,j+1}$  using the known  $u_{i,j}$  and  $u_{i\pm 1,j}$ -values, hence the name "explicit" scheme.

## 2. Implicit Scheme

The implicit scheme is an implementation based on the usage of the backward-Euler algorithm. The time evolution is given by the standard first order backward Euler implementation:

$$\frac{\partial u}{\partial t} = u_t \approx \frac{u_{i,j} - u_{i,j-1}}{\Delta t}. \quad (23)$$

The spatial differential will be given by the second order backward Euler-scheme, which is:

$$\frac{\partial^2 u}{\partial x^2} = u_{xx} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2}. \quad (24)$$

If we now take equations 23 and 24, and plug these into equation 17, we're left with the following expression:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} = \frac{u_{i,j} - u_{i,j-1}}{\Delta t}. \quad (25)$$

Rearranging this expression, with the same defined constant  $\alpha = \frac{\Delta t}{(\Delta x)^2}$  as in the explicit case, we can rearrange equation 25 and solve for  $u_{i,j-1}$ :

$$u_{i,j-1} = -\alpha u_{i-1,j} + (1 - 2\alpha)u_{i,j} - \alpha u_{i+1,j} \quad (26)$$

This expression is not directly solvable, given that we need to know the next time step to calculate the previous one. Using some linear algebra, we can extract the solution from this. If we rewrite the right side of equation 26 in terms of a matrix-vector multiplication, we get:

$$\hat{U}_{j-1} = \begin{bmatrix} (1+2\alpha) & -\alpha & 0 & \cdots & 0 \\ -\alpha & (1+2\alpha) & -\alpha & \ddots & \vdots \\ 0 & -\alpha & (1+2\alpha) & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\alpha \\ 0 & \cdots & 0 & -\alpha & (1+2\alpha) \end{bmatrix} \hat{U}_j. \quad (27)$$

Given  $i \in [0, L]$  and vectors  $\hat{U}_{j-1}^T = [u_{0,j-1}, \dots, u_{L,j-1}]$  and  $\hat{U}_j^T = [u_{0,j}, \dots, u_{L,j}]$ . Renaming the matrix  $\hat{A}$ , we get the following matrix equation:

$$\hat{U}_{j-1} = \hat{A} \hat{U}_j. \quad (28)$$

Multiplying this with the inverse matrix  $A^{-1}$ , we get:

$$\hat{U}_j = \hat{A}^{-1} \hat{U}_{j-1}. \quad (29)$$

As such, to calculate the next time step, you simply need to invert the matrix  $\hat{A}$ . Extrapolating this so that  $\hat{U}_{j-1} = \hat{A}^{-1} \hat{U}_{j-2}$ , we finally get get:

$$\begin{aligned} \hat{U}_j &= \hat{A}^{-1} \hat{U}_{j-1} \\ \hat{U}_j &= \hat{A}^{-1} (\hat{A}^{-1} \hat{U}_{j-2}) \\ &\vdots \\ \hat{U}_j &= \hat{A}^{-j} \hat{U}_0. \end{aligned}$$

Here,  $\hat{U}_0$  is the vector containing the initial condition of the system. Given that  $\alpha$  is a constant that we set, we only need to invert the matrix once, and multiply this with the previous time step vector at any given time  $\hat{U}_{j-1}$  to find the values at the next time step  $\hat{U}_j$ . We can also use the fact that the matrix  $\hat{A}$  is tridiagonal, and when using a tridiagonal solver, we can greatly improve the performance for solving the sets of equation. Running a regular matrix-vector multiplication algorithm for a dense matrix, the standard methods runs as  $\sim O(L^3)$ , while when using a tridiagonal solver, the timing runs as  $\sim O(L)[2]$ , where  $L$  is the amount of spatial steps that we've chosen. This tridiagonal solver solves a matrix-vector multiplication on the form:

$$\hat{A} \hat{x} = \hat{b} \quad (30)$$

where  $\hat{A}$  is a known (symmetric, tridiagonal) matrix and  $\hat{b}$  is a known vector. We have such a matrix-vector multiplication in equation 28, and as such we can use this solver in our problem at hand. This general tridiagonal solver uses the following three calculations to both invert the matrix and solve for the unknown quantities  $\hat{x}[3]$ :

$$\tilde{b}_i = b_i - \frac{e_{i-1}^2}{\tilde{b}_{i-1}}, \quad (31)$$

$$\tilde{g}_i = g_i - \frac{\tilde{g}_{i-1} e_{i-1}}{\tilde{b}_{i-1}}, \quad (32)$$

$$u_i = \frac{\tilde{g}_i - e_i u_{i+1}}{\tilde{b}_i}, \quad (33)$$

where  $b_i$  are the individual diagonal elements,  $e_i$  are the off-diagonal elements,  $g_i$  are the function values at the previous time step,  $\tilde{g}_i$  and  $\tilde{b}_i$  are intermedio calculations used to find  $u_i$ , which is the unknown values we are looking for.

## 3. Implicit Crank-Nicolson Scheme

If we implement the usage of both the forward- and backward-Euler algorithm with a Taylor-expansion around the time  $t' = t + \frac{\Delta t}{2}$ , we get what is known as the Crank-Nicolson scheme. The time-evolution for this scheme is given as:

$$\frac{\partial u}{\partial t} = u_t \approx \frac{u_{i,j+1} - u_{i,j}}{\Delta t}. \quad (34)$$

The spatial derivative when using the Crank-Nicolson scheme is given as ( $\frac{\partial^2 u}{\partial x^2} = u_{xx}$ ):

$$u_{xx} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j} + u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{2(\Delta x)^2}. \quad (35)$$

Combining the approximations we've made in equations 34 and 35, with equation 17, doing some restructuring of the equations and defining the same  $\alpha$  as before ( $\alpha = \frac{\Delta t}{(\Delta x)^2}$ ), we get the expression:

$$-\alpha u_{i-1,j} + (2 + 2\alpha)u_{i,j} - \alpha u_{i+1,j} = \alpha u_{i-1,j-1} + (2 - 2\alpha)u_{i,j-1} + \alpha u_{i+1,j-1}. \quad (36)$$

This expression can further be written down in a matrix form as:

$$(2\hat{I} + \alpha\hat{B})\hat{U}_j = (2\hat{I} - \alpha\hat{B})\hat{U}_{j-1} \quad (37)$$

Where  $\hat{I}$  is the identity matrix,  $\hat{U}_j$  is the vector containing the results for the current time step and  $\hat{U}_{j-1}$  is the vector containing the results from the previous time step. The matrix  $\hat{B}$  is thus defined as:

$$\hat{B} = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots \\ 0 & \cdots & \cdots & -1 & 2 \end{bmatrix}. \quad (38)$$

Multiplying the inverse of  $(2\hat{I} + \alpha\hat{B})$  on both sides of equation 37 gives us an expression for the next time step:

$$\hat{U}_j = (2\hat{I} + \alpha\hat{B})^{-1}(2\hat{I} - \alpha\hat{B})\hat{U}_{j-1}. \quad (39)$$

If we simply define  $\hat{U}'_{j-1} = (2\hat{I} - \alpha\hat{B})\hat{U}_{j-1}$ , we see that we have the same type of problem as with the implicit scheme, and we can also solve this by using the tridiagonal (symmetric) solver method shown in equations 31, 32 and 33.

## B. Numerical Stability & Accuracy

For our calculations to return accurate results, we require that the spectral radius of a matrix representation of the given numerical method  $\hat{A}$  is lower than or equal to 1. The spectral radius  $\rho(\hat{A})$  is defined as:

$$\rho(\hat{A}) = \max\{|\lambda| : \det(\hat{A} - \lambda\hat{I}) = 0\}. \quad (40)$$

More commonly referred to as the maximum (absolute) value of the eigenvalues of the matrix. Mathematically speaking, the requirement that we need to satisfy is the following:

$$\rho(\hat{A}) \leq 1. \quad (41)$$

We will be looking at the explicit and implicit schemes both on their own.

### 1. Explicit Scheme

For the explicit scheme, the algorithm is as stated in equation 22 given as:

$$u_{i,j+1} = \alpha u_{i-1,j} + (1 - 2\alpha)u_{i,j} + \alpha u_{i+1,j}. \quad (42)$$

Having  $\hat{U}_{j+1}^T = [u_{0,j+1}, \dots, u_{L,j+1}]$  and  $\hat{U}_j^T = [u_{0,j}, \dots, u_{L,j}]$ , we can rewrite this on matrix form:

$$\hat{U}_{j+1} = \begin{bmatrix} (1 - 2\alpha) & \alpha & 0 & \cdots & 0 \\ \alpha & (1 - 2\alpha) & \alpha & \ddots & \vdots \\ 0 & \alpha & (1 + 2\alpha) & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \alpha \\ 0 & \cdots & 0 & \alpha & (1 - 2\alpha) \end{bmatrix} \hat{U}_j. \quad (43)$$

Where we rename this matrix  $\hat{A}$ . we can observe that we further can rewrite the matrix as:

$$\hat{A} = \hat{I} - \alpha\hat{B}, \quad (44)$$

where  $\hat{I}$  is the identity matrix, and  $\hat{B}$  is given as the following tridiagonal matrix:

$$\hat{B} = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & 2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{bmatrix}. \quad (45)$$

The eigenvalues for the identity matrix is always equal to 1, meaning the the eigenvalues for the matrix  $\hat{A}$  given in equation 44 is given as  $\lambda_i = 1 - \alpha\mu_i$ , where  $\mu$  are the eigenvalues to matrix  $\hat{B}$  given in equation 45. The eigenvalues to a general  $N \times N$ , symmetric, tridiagonal matrix with diagonal elements  $a$  and off-diagonal elements  $b$  is given as[4]:

$$\mu_i = a + 2b \cos\left(\frac{i\pi}{N+1}\right), \quad i \in \{1, \dots, N\}. \quad (46)$$

Inserting this expression for  $\mu_i$ , we have that the eigenvalues for our matrix  $\hat{A}$  will be:

$$\lambda_i = 1 - 2\alpha \left[1 - \cos\left(\frac{i\pi}{N+1}\right)\right] \quad (47)$$

Since the spectral radius  $\rho(\hat{A}) = \max\{|\lambda_i|\}$ , we need to figure out the requirement on  $\alpha$  so that the spectral radius will remain below 1. Having the iterative variable  $i \in \{1, \dots, N\}$ , gives us that the cosine will take on only negative values with amplitude up to  $\cos(\theta) = -1$ . This means that the maximum (absolute) value of this expression is reached at  $\cos(\theta) = -1$ , which gives us the following limitation on  $\alpha$ :

$$|\lambda_{max}| = |1 - 2\alpha(1 + 1)| = |1 - 4\alpha| \leq 1. \quad (48)$$



The statement in equation 48 is true whenever we have  $\alpha \leq \frac{1}{2}$ . Since  $\alpha = \frac{\Delta t}{(\Delta x)^2}$ , we therefore get the following limitation on our choice of time step  $\Delta t$  coupled with our choice in spatial steps  $\Delta x$ :

$$\frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2} \quad (49)$$

This is an unfortunate numerical limitation, considering that a doubling of the number of positions we would like to analyze would result in having to do 4 times as many time steps to make sure that we have the stability requirement met.

The spatial error of this method runs as  $\mathcal{O}(\Delta x^2)$ , while the temporal error runs as  $\mathcal{O}(\Delta t)$ [5]. This means that the accuracy will increase faster when increasing the amount of spatial points, as opposed to temporal ones. Taking into account the stability requirement from equation 49, we see that we have to strike a balance between the amount of points spatially and temporally when considering accuracy and CPU run time.

## 2. Implicit Scheme

For the implicit scheme, we already have a matrix representation for the system given in equation 27 as:

$$\hat{A} = \begin{bmatrix} (1+2\alpha) & -\alpha & 0 & \cdots & 0 \\ -\alpha & (1+2\alpha) & -\alpha & \ddots & \vdots \\ 0 & -\alpha & (1+2\alpha) & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -\alpha \\ 0 & \cdots & 0 & -\alpha & (1+2\alpha) \end{bmatrix} \quad (50)$$

We can rewrite this on the same form as in the previous section as:

$$\hat{A} = \hat{I} + \alpha \hat{B} \quad (51)$$

where  $\hat{B}$  is the same matrix as in equation 45. This has eigenvalues given as  $\lambda_i = 1 + \alpha \mu_i$ , with  $\mu_i$  given with the same expression as in equation 46. This in turn gives us the full eigenvalues as:

$$\lambda_i = 1 + 2\alpha \left[ 1 + \cos \left( \frac{i\pi}{N+1} \right) \right] \quad (52)$$

This Cosine can still only obtain negative values, which gives us the spectral radius maximum when  $\cos(\theta) = 0$ , giving us the following expression for  $|\lambda_{max}|$ :

$$|\lambda_{max}| = |1 - 2\alpha(1 - 1)| = |1| \leq 1 \quad (53)$$

This statement is true for any choice of  $\alpha$ , and as such we are free to pick any values for  $\Delta t$  and  $\Delta x$  when using this method, and as such we don't have the same limitations as with the explicit scheme.

The spatial and temporal error using this method runs as the same as with the explicit method, with the temporal given as  $\mathcal{O}(\Delta t)$  and the spatial as  $\mathcal{O}(\Delta x^2)$ [6]. Considering that we don't have the same limitations as with the explicit method regarding the ratio between the time step and spatial step, this method is a much more forgiving scheme numerically speaking.

As we can see from equations 37, we have the same sign convention for the Crank-Nicolson scheme as with the implicit scheme. As such, we can deduct the same stability results from the Crank-Nicolson scheme as with the implicit scheme, and that is we are free to choose  $\Delta x$  and  $\Delta t$  as we like. The error using this method runs as  $\mathcal{O}(\Delta t^2)$  for the temporal part and  $\mathcal{O}(\Delta x^2)$  for the spatial part. Meaning that this method is more temporally accurate compared to the implicit and explicit counterparts.

## 3. Error calculations

To calculate the error in our different methods, we will use a finite difference scheme, which is given as:

$$\epsilon(t) = \frac{\sqrt{\sum_{i=1}^{N-1} |u(x_i, t_j) - f(x_i, t_j)|^2}}{\sqrt{\sum_{i=1}^{N-1} f(x_i, t_j)^2}}. \quad (54)$$

Here,  $u(x_i, t_j)$  is the numerically calculated result, while  $f(x_i, t_j)$  is the steady state analytic result, given by equation 7 in 1 dimension. The sums run from  $i = \{1, \dots, N-1\}$ . Given that the boundary values are set, these are ignored. Equation 54 will therefore be our measurement on the error of the 1D-system.

## C. Extension to 2 Dimensions

If we are to expand this problem to two dimensions, we need to look back at equation 16, and write the spatial gradient in two dimensions:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t} \quad (55)$$

Where  $u$  now is a function of two spatial dimensions  $x$  and  $y$ , as well as a temporal dimension  $t$ . When extending our numerical methods into two dimensions, we will be using the explicit scheme due to its overall simplicity of implementing.

### 1. Two-dimensional Explicit Scheme Algorithm

Using the explicit method, we can simply extend our method into two dimensions in a straight forward man-

ner. We first rewrite the variable  $u$  in a discretized manner:  $u(x, y, t) \rightarrow u(x_i, y_j, t_k) = u_{i,j,k}$ . If we now replace the spatial derivatives with the standard double forward Euler method, as given in equation 20, and the temporal derivative with the forward Euler method given in equation 19, we can rewrite equation 55 in the following manner:

$$\begin{aligned} & \frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{(\Delta x)^2} + \frac{u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{(\Delta y)^2} \\ &= \frac{u_{i,j,k+1} - u_{i,j,k}}{\Delta t}. \end{aligned} \quad (56)$$

For simplicity's sake, we will work with a square spatial grid when solving these equations, giving us  $\Delta x = \Delta y = h$ , where we define  $h$  as being the spatial step size. This gives us the equation on the form:

$$\begin{aligned} & \frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k} + u_{i,j+1,k} - 2u_{i,j,k} + u_{i,j-1,k}}{h^2} \\ &= \frac{u_{i,j,k+1} - u_{i,j,k}}{\Delta t}. \end{aligned} \quad (57)$$

Defining the variable  $\alpha = \frac{\Delta t}{h^2}$ , we can solve this for the next time step  $u_{i,j,k+1}$ , which results in a differential equation on the form:

$$\begin{aligned} u_{i,j,k+1} = u_{i,j,k} + \alpha & \left[ u_{i+1,j,k} + u_{i-1,j,k} \right. \\ & \left. + u_{i,j+1,k} + u_{i,j-1,k} - 4u_{i,j,k} \right] \end{aligned} \quad (58)$$

We therefore have an explicit algorithm now given in equation 58. We also need to look closer at the numerical stability requirements of this algorithm given by the spectral radius.

## 2. Two-dimensional Explicit Scheme Limitations

To start off, we can split up all the contributions to the "y-direction matrix" and the "x-direction matrix" each on their own. We can therefore rewrite equation 58 on the following form:

$$\begin{aligned} u_{i,j,k+1} = u_{i,j,k} + \alpha & \left[ u_{i+1,j,k} + u_{i-1,j,k} - 2u_{i,j,k} \right] \\ & + \alpha \left[ u_{i,j+1,k} + u_{i,j-1,k} - 2u_{i,j,k} \right]. \end{aligned} \quad (59)$$

From equation 59, we can rewrite this on matrix form:

$$\hat{U}_{k+1} = \left[ \hat{I} - \alpha \hat{B} - \alpha \hat{B}^T \right] \hat{U}_k. \quad (60)$$

Where  $\hat{U}_{k+1}$  is the matrix containing all spatial elements for time step  $k+1$ ,  $\hat{U}_k$  is the matrix containing all spatial elements for time step  $k$ , and  $\hat{B}$  is the same matrix as

defined in equation 45. Since  $\hat{B}$  is symmetric, we also have that  $\hat{B}^T = \hat{B}$ , so we can further rewrite this as:

$$\hat{U}_{k+1} = \left[ \hat{I} - 2\alpha \hat{B} \right] \hat{U}_k. \quad (61)$$

Renaming this matrix  $\hat{A} = \hat{I} - 2\alpha \hat{B}$ , we can now try to figure out what its eigenvalues are. We know that the identity matrix  $\hat{I}$  has all eigenvalues 1, and from equation 46 we have the eigenvalues of  $\hat{B}$  given as:

$$\lambda_B = 2 - 2 \cos \left( \frac{i\pi}{N+1} \right), \quad i \in \{1, \dots, N\} \quad (62)$$

Inserting this gives us the full eigenvalues as:

$$\lambda_i = 1 - 2\alpha \left[ 2 - 2 \cos \left( \frac{i\pi}{N+1} \right) \right]. \quad (63)$$

As before,  $\cos(\theta) \in (-1, 1)$ , and equation 63 reaches its (absolute) maximum when  $\cos(\theta) = -1$ , we replace the cosine with this value to get:

$$|\lambda_{max}| = |1 - 4\alpha[1 + 1]| = |1 - 8\alpha|. \quad (64)$$

Since we require that  $\rho(\hat{A}) = |\lambda_{max}| < 1$ , we therefore solve this for  $\alpha$ :

$$|1 - 8\alpha| < 1 \implies \alpha < \frac{1}{4}. \quad (65)$$

Since  $\alpha = \frac{\Delta t}{h^2}$ , this means that we impose the following restrictions on the relationship between our temporal and spatial steps:

$$\frac{\Delta t}{h^2} \leq \frac{1}{4}. \quad (66)$$

A nice thing to note is that we could also expand this using the implicit schemes, but this requires us to include an iterative method (like the Jacobi iterative method) for solving these problems[7]. We will however resort to using the explicit method in 2 dimensions, and this will therefore not be further discussed.

## D. Discretizing the Heat Equation

### 1. Finding the algorithm

We have the heat equation given in equation 9 as:

$$\left[ \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right] + \frac{Q}{k} = \frac{\rho c_p}{k} \frac{\partial T}{\partial t}. \quad (67)$$

We have the temperature  $T$  as a function of two spatial and one temporal variable. Discretizing this gives us  $T(x, y, t) \rightarrow T(x_i, y_j, t_k) = T_{i,j,k}$ . Solving the differentials using the explicit method as discussed in section III C, we get the following equation:

$$\begin{aligned} & \frac{T_{i+1,j,k} + T_{i-1,j,k} + T_{i,j+1,k} + T_{i,j-1,k} - 4T_{i,j,k}}{h^2} \\ &= \frac{\rho c_p}{k} \frac{T_{i,j,k+1} - T_{i,j,k}}{\Delta t} - \frac{Q}{k} \end{aligned} \quad (68)$$



Solving this for  $T_{i,j,k+1}$  gives us the following:

$$\begin{aligned} & \frac{k\Delta t [T_{i+1,j,k} + T_{i-1,j,k} + T_{i,j+1,k} + T_{i,j-1,k} - 4T_{i,j,k}]}{\rho c_p h^2} \\ & + \frac{Q\Delta t}{\rho c_p} + T_{i,j,k} = T_{i,j,k+1} \end{aligned} \quad (69)$$

This is the equation that we will have to solve numerically. We have already defined the boundary conditions in the y-direction of the model ( $T(y = 0) = 7^\circ\text{C}$  and  $T(y = L_y) = 1300^\circ\text{C}$  for  $y \in [0, L_y]$ ), as well as starting with the initial condition for the rest of the system as being 0. As for the boundary conditions in the x-direction, we will be using periodic boundary conditions. This is modelled such that the temperature at  $x = 0$  will be adjacent to the temperature at  $x = L$  for  $x \in [0, L_x]$ . Since we don't expect the lithosphere outside our given model to behave any differently to the lithosphere within our model, this is a valid way of simulating the boundary conditions in the x-boundaries.

Considering that we are solving this system using an explicit scheme, we need to figure out the numerical limit to equation 69 given by its spectral radius next.

## 2. Finding the limitations

Just like in section III C, we will have to rewrite our equation 69 on matrix form. Firstly, we define the following relation:

$$\beta = \frac{k\Delta t}{\rho c_p h^2}. \quad (70)$$

With this in hand, we can rewrite the equation as follows:

$$\begin{aligned} & \beta [T_{i+1,j,k} + T_{i-1,j,k} + T_{i,j+1,k} + T_{i,j-1,k} - 4T_{i,j,k}] \\ & + \frac{Q\Delta t}{\rho c_p} + T_{i,j,k} = T_{i,j,k+1}. \end{aligned} \quad (71)$$

Doing a simple rewriting, we turn this equation into:

$$\begin{aligned} T_{i,j,k+1} &= \frac{Q\Delta t}{\rho c_p} + T_{i,j,k} + \beta [T_{i+1,j,k} + T_{i-1,j,k} - 2T_{i,j,k}] \\ &+ \beta [T_{i,j+1,k} + T_{i,j-1,k} - 2T_{i,j,k}]. \end{aligned} \quad (72)$$

We can rewrite this on matrix form as follows (just like in the case in equation 60:

$$\hat{T}_{k+1} = \frac{Q\Delta t}{\rho c_p} + [\hat{I} - \beta\hat{B} - \beta\hat{B}^T]\hat{T}_k. \quad (73)$$

Here, the matrix  $\hat{B}$  is the same as the matrix in equation 45, and as it is symmetric, we can simplify this further into simply being:

$$\hat{T}_{k+1} = \frac{Q\Delta t}{\rho c_p} + [\hat{I} - 2\beta\hat{B}]\hat{T}_k \quad (74)$$

Renaming  $\hat{A} = \hat{I} - 2\beta\hat{B}$ , we need to find its spectral radius. We have already discussed a similar system in equations 61-65, meaning that we know the limitation on our parameter  $\beta$  to be the following:

$$\beta \leq \frac{1}{4}. \quad (75)$$

Inserting for the full expression of  $\beta$ , we get the full limitation on the spatial and temporal steps together to be:

$$\frac{k\Delta t}{\rho c_p h^2} \leq \frac{1}{4} \implies \frac{\Delta t}{h^2} \leq \frac{\rho c_p}{4k}. \quad (76)$$

Which is the limit we have to work with to make sure that we have receive stable, numerical results. Solving equation 76 for  $\Delta t$ , we get:

$$\Delta t \leq \frac{\rho c_p h^2}{4k}. \quad (77)$$

Inserting for  $\rho = 3.5 \cdot 10^{12} \frac{\text{kg}}{\text{km}^3}$ ,  $c_p = 1000 \frac{\text{W s}}{\text{kg}^\circ\text{C}}$ ,  $h = 1 \text{ km}$  and  $k = 2500 \frac{\text{W}^\circ\text{C}}{\text{km}}$ , this gives us:

$$\Delta t \leq 3.5 \cdot 10^{11} \text{ s} \approx 11.091 \text{ ky}, \quad (78)$$

Where "ky" indicates "Kiloyears". We will here model the step size  $h$  as being  $h = 1 \text{ km}$ , and by modelling this as a 120 km deep and 350 km wide "box", this results in a  $[351 \times 121]$ -grid. We will for simplicity set  $\Delta t = 10 \text{ ky}$ , meaning that we will for a full gigayear-simulation do  $10^5$  time steps. Finally, we will plan to do three separate simulations:

- One where we don't have radioactive enrichment of the mantle.
- One with radioactive enrichment of the mantle, disregarding the radioactive decay.
- One with radioactive decay in the picture.

## IV. RESULTS

The results will be split up into 3 parts. First we will present the results for the 1D runs, using the explicit, implicit and Crank-Nicolson schemes respectively. We will then be presenting the two-dimensional run using the explicit scheme. We will finally be presenting the results for the model of the temperature distribution of the lithosphere.

### A. Unitless 1 Dimensional

#### 1. Explicit Scheme

In figure 1, we can see a visualization of the results we have gotten by running the explicit algorithm. The difference of

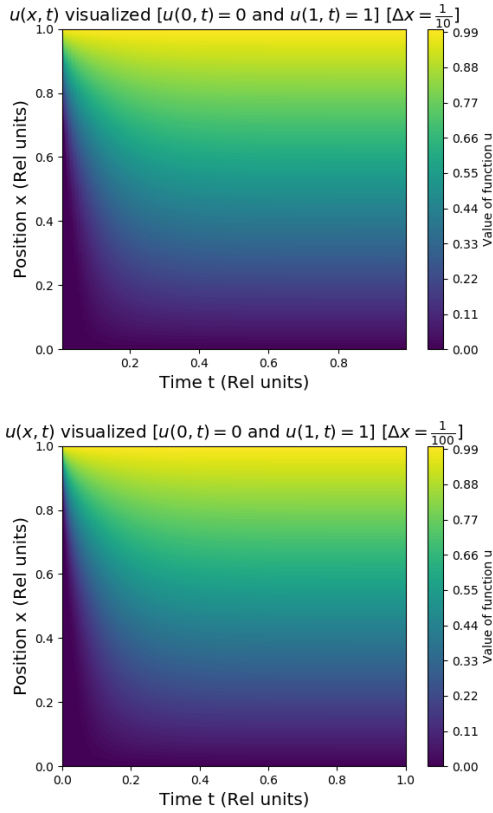


Figure 1. The results for running the explicit scheme with boundary conditions  $u(x = 0, t) = 0$  and  $u(x = 1, t) = 1$ . Time on the x-axis, position on the y axis, and amplitude / function value represented by the color bar. The top figure represents a 11-point simulation with  $\Delta x = 0.1$ , while the bottom has 101 points and  $\Delta x = 0.01$ . As we can see, the plots on their own look pretty similar. The bottom figure has, as a result of having a smaller  $\Delta x$ , a greater resolution than the top figure. We also observe that after a given relaxation time, that the plots seem to form a linear relationship between the two boundaries.

To further illustrate the linearity of the steady state, as well as the "movement" of the function towards the steady state, I have created a GIF. This GIF can be found by clicking on the following URLs:

- For  $\Delta x = 0.1$ : [Click here](#). (Links to GitHub)
- For  $\Delta x = 0.01$ : [Click here](#). (Links to GitHub)

In general, we can see that these both (qualitatively) converge towards a linear state as expected.

#### 2. Implicit Scheme

In figure 2, we have the visualization of the results after running the implicit algorithm, with both time, position and function value represented.

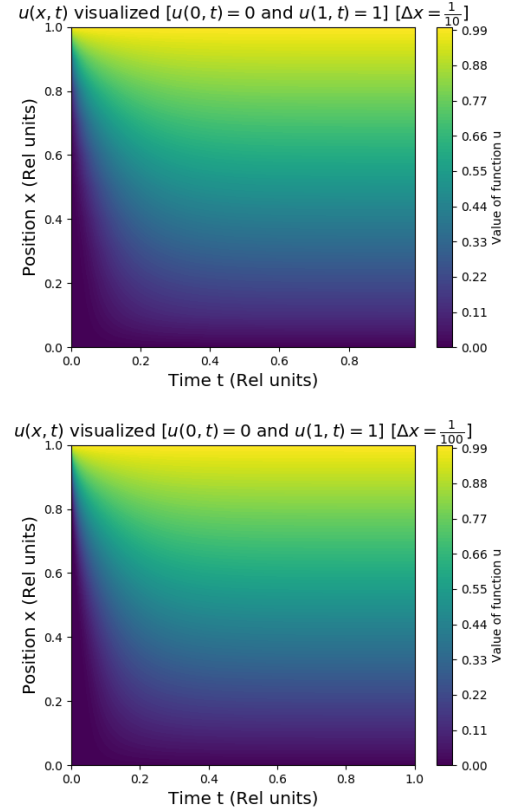


Figure 2. The results for running the implicit scheme with boundary conditions  $u(x = 0, t) = 0$  and  $u(x = 1, t) = 1$ . Time is on the x-axis, position on the y-axis and amplitude / function value is represented by the color bar. The top figure represents a 11-point simulation with  $\Delta x = 0.1$ , while the bottom has 101 points and  $\Delta x = 0.01$ . As we can see, the plots look pretty similar, whereas the bottom plot will have a greater resolution compared to the top plot due to the smaller  $\Delta x$ .

To further illustrate the linearity of the steady state for the implicit scheme, as well as the "movement" towards the steady state, the following GIFs are linked:

- For  $\Delta x = 0.1$ : [Click here](#). (Links to GitHub)
- For  $\Delta x = 0.01$ : [Click here](#). (Links to GitHub)

A more in-depth look to the steady state of this system can be seen in figure 3. Here, we can visually see that the steady state of the system for  $\Delta x = 0.1$  is slightly perturbed away from being perfectly linear. We can't see a similar inaccuracy for the  $\Delta x = 0.01$ , but we can expect there exists a similar, yet smaller inaccuracy.

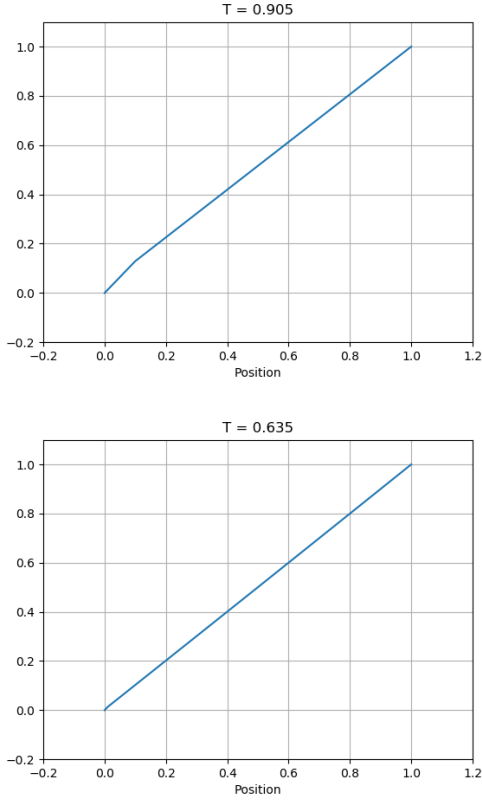


Figure 3. The steady state of the implicit schemes.  $\Delta x = 0.1$  for the top plot,  $\Delta x = 0.01$  for the bottom plot. We can visually see a slight "dent" in the top plot, indicating a deviation, and as such, an inaccuracy in this method.

### 3. Implicit Crank-Nicolson Scheme

In figure 4, we have the visualization of the results after running the implicit Crank-Nicolson scheme, with both time, position and function value represented.

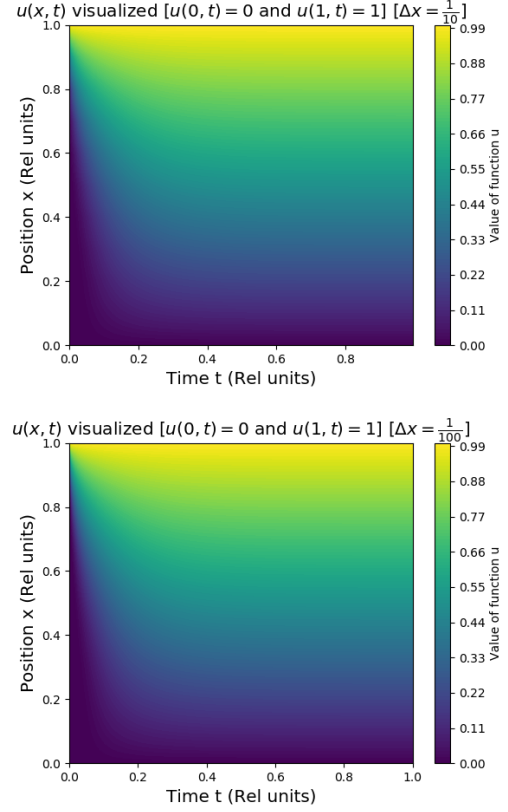


Figure 4. The results from running the implicit Crank-Nicolson scheme with similar boundary conditions as the other ones ( $u(x=0, t) = 0$  and  $u(x=1, t) = 1$ ). Time on the x-axis, position on the y-axis, and amplitude / function value represented by the color bar. The top figure represents a 11-point simulation with  $\Delta x = 0.1$ , while the bottom is a 101-point simulation with  $\Delta x = 0.01$ . As with the other plots, there are more or less no difference between these plots.

To further illustrate the linear steady state, as well as the time evolution for the Crank Nicolson scheme, see the following GIFs:

- For  $\Delta x = 0.1$ : [Click here](#). (Links to GitHub)
- For  $\Delta x = 0.01$ : [Click here](#). (Links to GitHub)

#### 4. Comparing Efficiency and Accuracy

In table II, we see the run times of the different schemes for  $\Delta x = 0.1$  and  $\Delta x = 0.01$  respectively. For these runs, we've chosen to have just as many time steps as the stability criteria of the explicit scheme allows us (given by  $\frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$  for the 1-dimensional case).

Method	$\Delta x = 0.1$	$\Delta x = 0.01$
Explicit	$\approx 2$ ms	838.4 ms
Implicit	$\approx 3$ ms	930.8 ms
Crank-Nicolson	$\approx 3.5$ ms	1127.82 ms

Table II. Comparing the time spent doing all three algorithms for  $\Delta x = 0.1$  and  $\Delta x = 0.01$  respectively. The results are comparable since we've also put the same limitation to the  $\Delta t$  values for both simulations, giving us just as many integration points for both methods. For the runs with ' $\approx$ ', the run times are approximate, given that the run times varied significantly around said value.

From table II, we can see that the explicit scheme is slightly more efficient than its implicit counterparts. Further, we can see in figure 5, we can see the (logarithm) of the error in our methods using the finite difference scheme shown in equation 54.

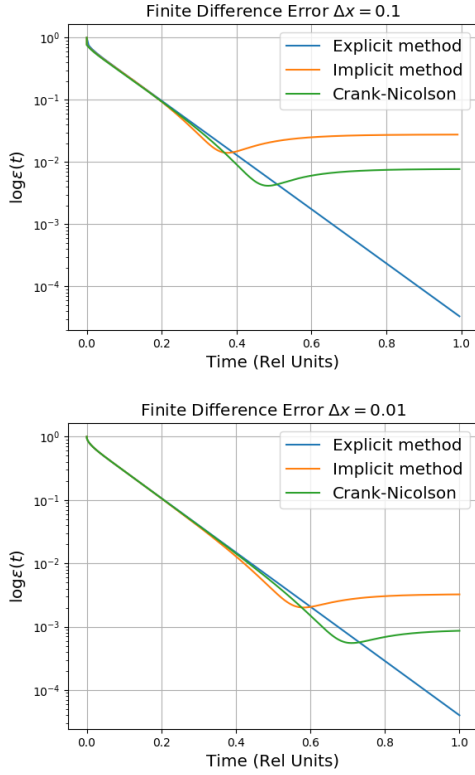


Figure 5. Error calculations using the finite difference scheme. Top plot for the system with  $\Delta x = 0.1$ , bottom plot for the system with  $\Delta x = 0.01$ .

#### B. Unitless 2 Dimensional

For the runs of the 2-dimensional PDE-solver, we've used the boundary conditions where the top and bottom have constant boundary conditions set at 1. ( $u(x, y = 0, t) = u(x, y = 1, t) = 1$ ). The right and left boundaries are set to 0. ( $u(x = 0, y, t) = u(x = 1, y, t) = 0$ ). The corners where these boundaries intersect are also set to 1. For the 2 dimensional case, the explicit scheme has been solely used to calculate the values  $u(x, y, t)$ .

##### 1. Step Size $\Delta x = \Delta y = 0.1$

In figure 6, we can see the results of the diffusion taking place in the first few instances after the simulation has started. We can see that the higher values are starting to distribute across the canvas.

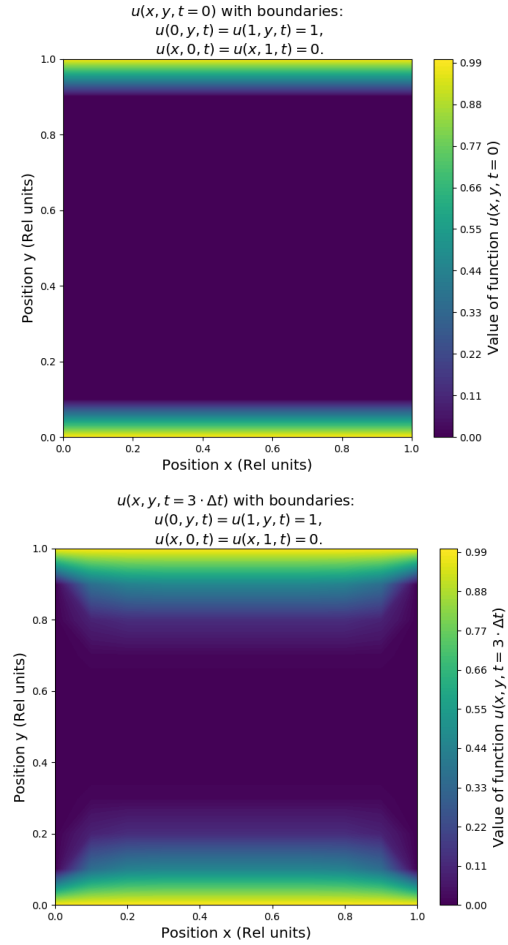


Figure 6. A selection of time steps of results from running the explicit scheme for the 2 dimensional simulation. The top plot gives us the values of  $u(x, y, t)$  at the initial time  $t = 0$ , while the bottom plot gives us the distribution at a time  $t = 3\Delta t$  a short while later.

In figure 7, we see the results of the diffusion taking

place at some later time when the system is approaching the steady state.

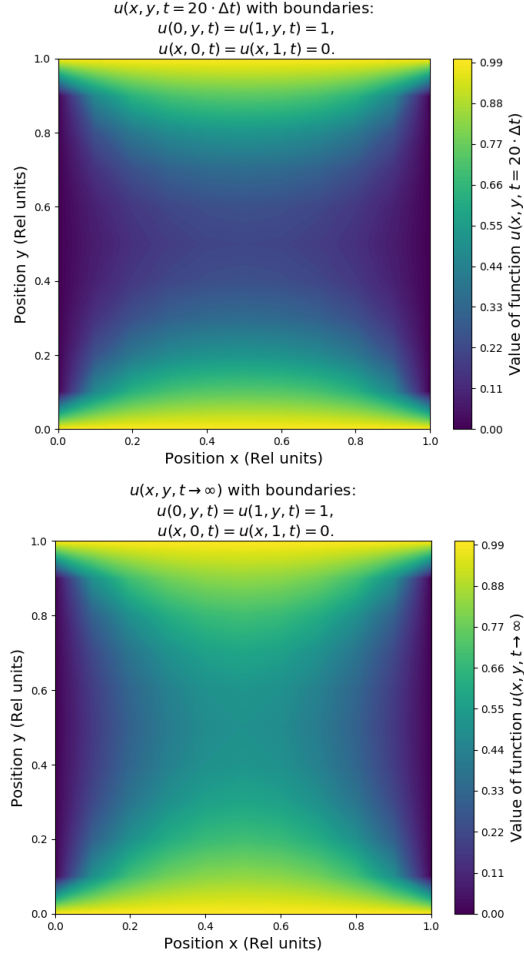


Figure 7. A selection of time steps of results from running the explicit scheme for the 2 dimensional simulation. The top plot gives us the values of  $u(x, y, t)$  at time  $t = 20\Delta t$ , when the system is closing in on getting into a steady state. The bottom plot shows us at a sufficiently late time step  $t \rightarrow \infty$ , where the system has hit a steady state.

Purely qualitatively, we can say that the system in question for  $\Delta x = \Delta y = 0.1$  reaches the steady state after  $\approx 200\Delta t$ .

For a GIF simulating this system, [Click here](#). (Links to GitHub)

## 2. Step Size $\Delta x = \Delta y = 0.01$

In figure 8, we see a visualization of the results of running the explicit algorithm for the PDE with  $\Delta x = \Delta y = 0.01$ . We can see that the higher values are starting to distribute across the canvas here as well, but at a slower rate as compared to the case in figure 6.

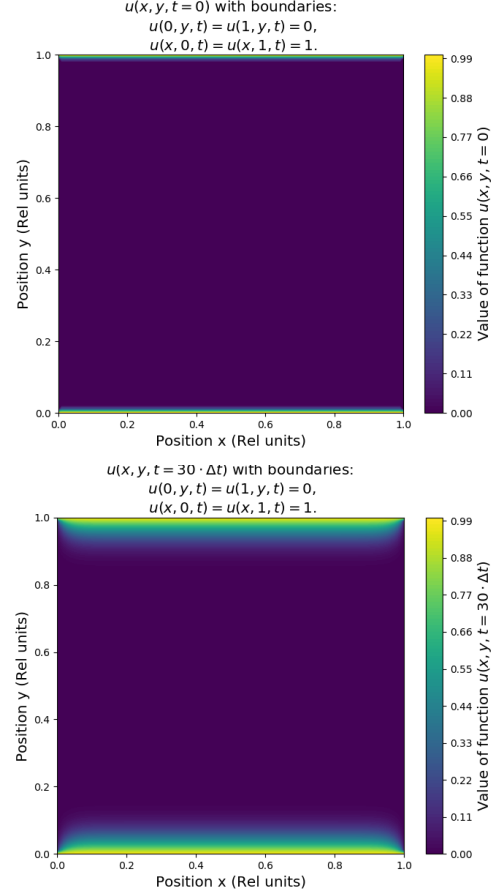


Figure 8. A selection of time steps of results from running the explicit scheme for the 2 dimensional simulation. The top plot gives us the values of  $u(x, y, t)$  at the initial time  $t = 0$ , while the bottom plot gives us the distribution at a time  $t = 30\Delta t$  a short while later.

In figure 9, we see the same results as in figure 8, but even later into the simulation when it is closing into the steady state.

Purely qualitatively, we can say that this system with  $\Delta x = \Delta y = 0.01$  reaches the steady state after  $\approx 4000\Delta t$ . Considering that this system is notoriously slow to make into a GIF, as well as this only being a higher-resolution run of the previous simulation, I will not be making this into a GIF.

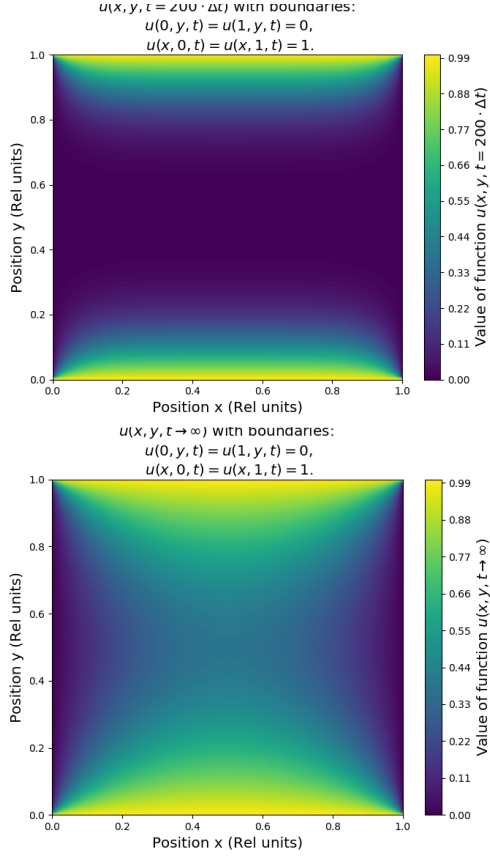


Figure 9. A selection of time steps of results from running the explicit scheme for the 2 dimensional simulation. The top plot gives us the values of  $u(x, y, t)$  at time  $t = 200\Delta t$ , when the system is closer to reaching the steady state. The bottom plot gives us the distribution at a time  $t \rightarrow \infty$ , at a time when we most definitely has reached the steady state.

### C. Lithosphere Results

The results of running the heat equation simulation as already discussed can be seen in figure 10

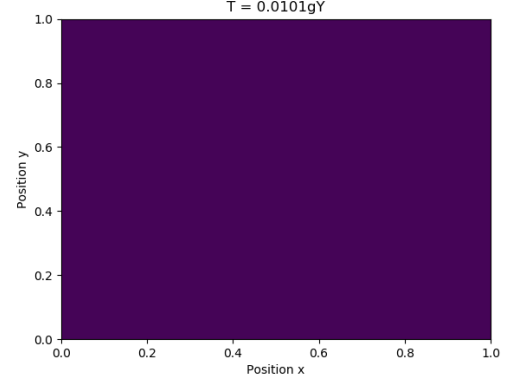


Figure 10. Results for running the lithosphere simulation. These results are inconclusive, and the temperature values of the lithosphere never grew any particularly. (Within the lithosphere, the values stayed at roughly  $\sim 10^{-15}$  at most as a steady state, indicating a flaw in our calculations.

In figure 10, we see temperature as the colors within a canvas with the x-values being scaled to  $\tilde{x} = x/L_x$  and the y-values scaled to  $\tilde{y} = y/L_y$ , with  $L_x = 350$  km and  $L_y = 120$  km. What we see here is a pretty homogeneous distribution of temperature, with all of the temperatures lying at around  $\sim 10^{-15}$  °C. (The color bar is not visible, due to bug in plotting program denying the usage of color bars). Due to the lithosphere simulation never converging to a steady state, Any other work on trying to model the radioactive enrichment of the upper mantle was postponed due to not getting accurate results.



## V. DISCUSSION

This discussion will be promptly split between the unitless numerical solution to a PDE, and the lithosphere heat equation system.

### A. Dimensionless, General PDE

The dimensionless discussion will be split between the 1-dimensional and 2-dimensional runs.

#### 1. 1 Spatial Dimension PDE

For the unitless, 1 dimensional simulation, we have the results as seen in figures 1-4, as well as in table II and figure 5.

The general expected behaviour of a 1-dimensional function as it reached a steady state is a linear relation between the two boundary conditions, as stated in equation 7. Looking at the steady state of our simulation (either in the GIFs, or purely qualitatively in the color plots of figures 1, 2 or 4, that we in all cases get a steady state linear relation. In figure 3 for the implicit method, it is apparent that the steady state is linear, as expected. This can in turn be a sign of our simulation returning sensible results.

One way of determining the validity of our results is to have a comparative look between the different methods. Looking at figures 1, 2 and 4, we see that for any choice of method and any choice of  $\Delta x$  (with  $\Delta t$  dictated by the explicit algorithm's limitation), the evolution towards the steady state, which we can call the systems' "relaxation time", looks more or less perfectly similar. The fact that we get similar results when we use different independent methods on identical systems can further confirm the fact that our results are sensible and as expected.

Looking at steady state for the implicit scheme in figure 3, we can clearly see a "dent" in the  $\Delta x = 0.1$  run, where the steady state that has been reached isn't perfectly linear, but slightly curved near the bottom values. Looking at the corresponding GIF, it is apparent that this method "overshoots" the correct analytic value, and doesn't stabilize perfectly. This is not directly apparent in the  $\Delta x = 0.01$  run, and is likely a global error which shrinks quickly. However, a more thorough error analysis of this system should be done to quantify the observed deviation.

Looking at table II, we see that in a simulation given equal amount of integration points and time steps, the explicit scheme is the most efficient among the schemes we've been using. Where a simulation using

the explicit scheme takes about  $\sim 850$  ms, using the implicit will yield us a run of about  $\sim 950$  ms and the Crank-Nicolson scheme will have a run of  $\sim 1150$  ms. Even though the explicit scheme is more efficient, we also need to consider the fact that the implicit schemes has a greater flexibility regarding choice of  $\Delta x$  and  $\Delta t$ -values. This, coupled with the error running as  $\mathcal{O}(\Delta t^2)$  for the Crank-Nicolson scheme, making it even more accurate, means that whatever application these methods are used for, need to strike a balance between efficiency, flexibility and accuracy.

Looking at figure 5, we see the error in our calculations using the different methods as well. Here, we can see that as time progresses, the explicit method grows more accurate (the error drops) compared to both the implicit and Crank-Nicolson methods. This is the case both for the  $\Delta x = 0.1$  and  $\Delta x = 0.01$  case. This might stem from the fact that the implicit methods make use of the tridiagonal solver as already discussed, eventually leading to loss of numerical precision at a quicker rate than the explicit method. As such, it actually seems like the explicit method is the method in which we can achieve the greatest accuracy given our spatial and temporal choice of span.

#### 2. 2 Spatial Dimensions PDE

For the unitless, 2 Dimensional box simulation we have the results as seen in figures 6-9.

Looking at the initial conditions on the top plot in both figures 6 and 8, we see that the boundary conditions both at the top and bottom are a tad bit different. This simply stems from the difference in the resolution of the system, and how the model is projected. The color-plots consist of a grid of different values, and the color value between the individual points are set to linearly change between the nodes. For this reason, the plots only give a model on how the distribution is between any given point, where the more points we have, the more accurate of a model we have. We can however also use this initial time plot to tell us something about the resolution of the plot. Given that we know the initial values to be 0 everywhere and 1 at the top and bottom, we can get a qualitative measure on how accurate we can expect the given model to be.

Considering that we set up identical boundary conditions on opposite sides of the "box", we can't really expect the same behaviour of a linear drop from one end to the other as predicted in equation 7. We would however always expect that the "flow" of the  $u(x, y, t)$ -values after it has reached steady state to go from the higher values to the lower values, which is something we somewhat see in the steady state plots at the bottom of figure 7 and 9.

## VI. CONCLUSION

We can do a thought experiment where the box is made up of a homogeneous material with a constant heat capacity and no heat production. The top and bottom walls of this box has a constant temperature 1 while the rightmost and leftmost walls has a constant temperature 0. If this is the case, we would expect that the temperature distribution within the box would be low close to the low-temperature walls, and high close to the high-temperature walls. We would also expect that the middle would behave as a saddle point. Since the middle point is as far away from any walls as possible, we would expect the gradient at this point to be 0. Purely qualitatively, this seems to be the case for the steady state, and as such, the results seem sensible.

Doing these simulations using an implicit method could also be of use, given the less restrictive this is to our choice in the time steps  $\Delta t$  and spatial steps  $h$ . Purely speculatively, given that we were to write an implicit algorithm using an iterative method that works, we could expect, if anything, a lot more freedom in simulating the systems both in terms of time steps  $\Delta t$  and spatial steps  $h$ . The overall efficiency of such an algorithm as compared to the explicit scheme is however difficult to say anything about without having said algorithm. Comparing to the 1-dimensional analysis in regards to speed, as seen in table II, we see that the implicit methods in general are slower than the explicit method. As such, we would expect that the implicit methods for two dimensions should be slower than the explicit method. If this is the case, we would have to strike a balance between the efficiency of the explicit methods together with the freedom that the implicit methods yield us.

Taking this into consideration, we can deduct that this simulation behaves as expected, as it converges into a steady state which physically makes sense. As such, it has been attempted to expand this into a real-world example, that being the lithosphere temperature distribution using the heat equation.

### B. Lithosphere and the Heat Equation

Our results from running the model in the lithosphere can be found in figure 10. These results are inconclusive, and the reason for this is unknown. I have decided to retain the results that I do have in this report, so that an eventual reader can figure out where this thought process might have gone wrong.

It is difficult to see what has gone wrong, whether it is a flaw in the logic in the algorithm, a bug in the program code, or a fault in the plotting program. This will be "left as an exercise to the reader".

To conclude this, we need to look at the different simulations each on their own.

For the 1 dimensional analysis, we got sensible and logical results when both using the explicit scheme, implicit scheme, and Crank-Nicolson scheme. We got that the steady state of the system was as expected, as well as having a similar "relaxation time" for all 3 schemes. We also got that the explicit scheme was the most efficient among the three, but the least flexible, while the Crank-Nicolson, being the least efficient, also being the most accurate and flexible. The balance therefore has to be struck between efficiency, accuracy and flexibility when choosing which scheme to apply.

For the 2 dimensional analysis using the explicit scheme, we got somewhat expected results qualitatively speaking. As we didn't have a direct analytic expression to compare it to however, we could only see if the system behaves somewhat as expected, and see if the behaviour is something that can be justified. This doesn't mean that our results are correct, but given a thought experiment with temperature distribution, the behaviour seem to align with our intuition at the very least.

Our attempt at modelling the Lithosphere unfortunately failed, and without any sensible information extracted from those simulations, we can't really tell if our numerical method actually works, or whether or not there is a logical flaw in our derivations. As such, our simulation regarding the lithosphere are inconclusive, with results that are non-realistic for unknown reasons.

In hindsight, an attempt at doing the 2 dimensional analysis using an implicit method could have proven useful, especially given the lack of restrictions on the time step  $\Delta t$  and spatial step  $\Delta x/\Delta y$ . This is therefore one of the main points of improvement that can be made in the short term to the results that my simulations have given us. We should also have given an attempt at parallelizing the algorithms, such that we could utilize more processing power. This was also unfortunately left out, and is another factor in which this could be improved.

## Appendix A: Deriving the Different Schemes

We will here derive the numerical relationships that we've used for the explicit, implicit and Crank-Nicolsons scheme in this report. These methods are derived where we have the double derivative of the spatial terms, and a single derivative on the temporal. In other words, an equation on the form:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t}. \quad (\text{A1})$$

### 1. Explicit Scheme

In general with the explicit scheme, we will use the forward Euler numerical approach to a Taylor-expansion.

#### a. Temporal Derivative

With the temporal derivative, we simply do a first-order Taylor expansion of the function  $u(x, t = t + \Delta t)$  at a point around  $t$ , giving us:

$$u(x, t + \Delta t) = u(x, t) + \frac{\partial u(x, t)}{\partial t} (t - t + \Delta t) + \mathcal{O}(\Delta t). \quad (\text{A2})$$

Solving this equation for the derivative gives us:

$$\frac{\partial u(x, t)}{\partial t} = \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} - \mathcal{O}(\Delta t) \quad (\text{A3})$$

Switching notation to the one used in equation 19, with  $u(x, t) \rightarrow u(x_i, t_j) = u_{i,j}$ , and removing the error  $\mathcal{O}(\Delta t)$ , we get:

$$u_t \approx \frac{u_{i,j+1} - u_{i,j}}{\Delta t}. \quad (\text{A4})$$

Which was the temporal term I was going to show.

#### b. Spatial (double) Derivative

For the spatial double derivative, we first do the same Taylor approximation as above, just an approximation around  $x$  for the function around  $x - \Delta x$ :

$$u(x - \Delta x, t) = u(x, t) + \frac{\partial u(x, t)}{\partial x} (x - x - \Delta x) + \mathcal{O}(\Delta x) \quad (\text{A5})$$

Solving this for the derivative gives us:

$$\frac{\partial u(x, t)}{\partial x} = -\frac{u(x - \Delta x, t) - u(x, t)}{\Delta x} - \mathcal{O}(\Delta x). \quad (\text{A6})$$

We'll leave it at this for now. We now do a new similar Taylor expansion, but this time to the second order:

$$u(x + \Delta x, t) = u(x, t) + \frac{\partial u(x, t)}{\partial x} \Delta x + \frac{\partial^2 u(x, t)}{\partial x^2} (\Delta x)^2 - \mathcal{O}(\Delta x^2) \quad (\text{A7})$$

Now we can implement our expression for the first derivative from equation A6 into equation A7, and if we disregard the errors, we get:

$$u(x + \Delta x, t) \approx u(x, t) + u(x - \Delta x, t) + u(x, t) + \frac{\partial^2 u(x, t)}{\partial x^2} (\Delta x)^2 \quad (\text{A8})$$

Solving this equation for the double derivative gives us:

$$\frac{\partial^2 u(x, t)}{\partial x^2} = \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{(\Delta x)^2}. \quad (\text{A9})$$

Changing up the notation as seen in equation 20, this equals:

$$u_{xx} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} \quad (\text{A10})$$

Which was the spatial term that I was going to show.

### 2. Implicit Scheme

For the implicit scheme, we do much of the same as with the explicit scheme, just that we this time primarily use the backward Euler numerical approach to a Taylor expansion

#### a. Temporal Derivative

For the temporal derivative, we do the first-order Taylor expansion of the function  $u(x, t - \Delta t)$  at a point around  $t$ , giving us:

$$u(x, t - \Delta t) = u(x, t) + \frac{\partial u(x, t)}{\partial t} (t + t - \Delta t) + \mathcal{O}(\Delta t). \quad (\text{A11})$$

Solving this for the derivative gives us:

$$\frac{\partial u(x, t)}{\partial t} = \frac{u(x, t - \Delta t) - u(x, t)}{-\Delta t} - \mathcal{O}(\Delta t). \quad (\text{A12})$$

Redefining our equation on the form seen in equation 23, and neglecting the error, we end up with:

$$u_t \approx \frac{u_{i,j} - u_{i,j-1}}{\Delta t}. \quad (\text{A13})$$

Which was the temporal derivative I was going to show.

#### b. Spatial (double) Derivative

For the spatial derivative, we first do a forward, first order Taylor approximation of the function  $u(x + \Delta x, t)$  around the point  $x$  as follows:

$$u(x + \Delta x, t) = u(x, t) + \frac{\partial u(x, t)}{\partial x} \Delta x + \mathcal{O}(\Delta x). \quad (\text{A14})$$

Solving this for the derivative, we have:

$$\frac{\partial u(x, t)}{\partial x} = \frac{u(x + \Delta x, t) - u(x, t)}{\Delta x}. \quad (\text{A15})$$

And we'll leave this at that for now. We now do a second order Taylor approximation of the function  $u(x - \Delta x, t)$  around the point  $x$ , disregarding the error, giving us:

$$u(x - \Delta x, t) \approx u(x, t) - \frac{\partial u(x, t)}{\partial x} \Delta x + \frac{\partial^2 u(x, t)}{\partial x^2} (\Delta x)^2. \quad (\text{A16})$$

If we now take the expression in equation A15, disregarding the error, and insert for the derivative in equation A16, we get:

$$u(x - \Delta x, t) \approx u(x, t) + u(x, t) - u(x + \Delta x, t) + \frac{\partial^2 u(x, t)}{\partial x^2} (\Delta x)^2. \quad (\text{A17})$$

Finally, we can solve this for the double derivative:

$$\frac{\partial^2 u(x, t)}{\partial x^2} = \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{(\Delta x)^2}. \quad (\text{A18})$$

Rewriting this on the form seen in equation 24, we finally get:

$$u_{xx} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2}. \quad (\text{A19})$$

Which was the spatial term we were going to show.

### 3. Crank-Nicolson Scheme

For the Crank-Nicolson scheme, we include a parameter  $\theta$ , and combine the explicit and implicit schemes as follows:

$$\begin{aligned} & \frac{\theta}{(\Delta x)^2} (u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + \frac{1-\theta}{(\Delta x)^2} (u_{i+1,j-1} \\ & - 2u_{i,j-1} + u_{i-1,j-1}) = \frac{1}{\Delta t} (u_{i,j} - u_{i,j-1}). \end{aligned} \quad (\text{A20})$$

Here, if we pick  $\theta = 0$ , this will give us the explicit scheme, and if we pick  $\theta = 1$ , this will give us the implicit scheme. Picking  $\theta = \frac{1}{2}$ , this will give us the Crank-Nicolson scheme.[8]

## Appendix B: Stencil Representation of the Methods

There are charts in which we can represent what the new values on a point  $u_{i,j}$  depend on when calculating the new values. These dependency-charts are known as stencils, and here I've listed up the stencil representation for the three methods I've used (explicit, implicit, and Crank-Nicolson schemes).

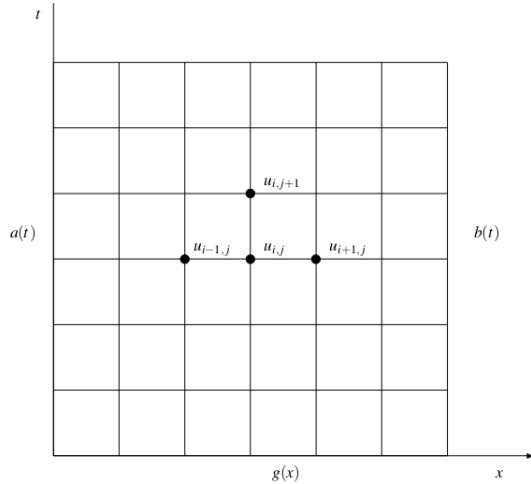


Figure 11. The stencil representation for the explicit scheme. To calculate any given point using this scheme, we would require to know a value on any of the others. Typically, we require to know anything at the time step  $t$  to calculate for a time step  $t + \Delta t$ .<sup>a</sup>

<sup>a</sup> Image Source: M. Hjorth-Jensen, Lecture notes in Computational Physics. p. 306, figure 10.1. Course URL: <https://www.uio.no/studier/emner/matnat/fys/FYS3150/index-eng.html>

In figure 11, we see the stencil representation of the explicit scheme.

In figure 12, we see the stencil representation of the implicit scheme.

In figure 13, we see the stencil representation of the Crank-Nicolson scheme.

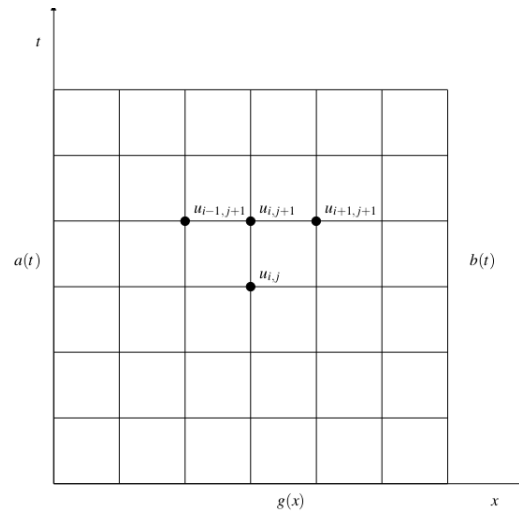


Figure 12. The stencil representation for the implicit scheme. To calculate any given point using this scheme, we would require to know a value on any of the others. Given that we usually don't know all the values for the next time step, we can't directly calculate the next time step. It is possible however, hence the name "implicit".<sup>a</sup>

<sup>a</sup> Image Source: M. Hjorth-Jensen, Lecture notes in Computational Physics. p. 310, figure 10.2. Course URL: <https://www.uio.no/studier/emner/matnat/fys/FYS3150/index-eng.html>

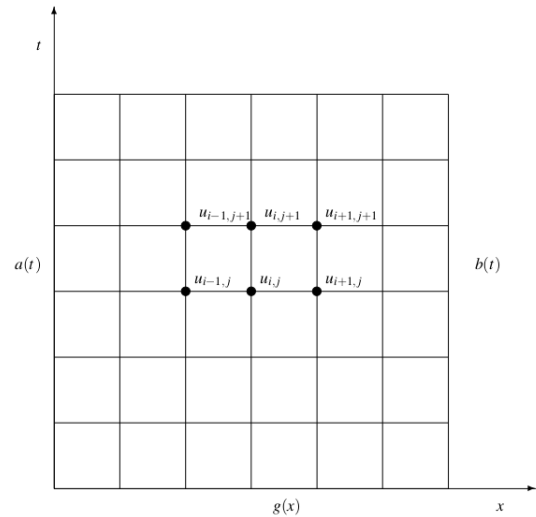


Figure 13. The stencil representation for the Crank-Nicolson scheme. To calculate any given point using this scheme, we would require to know a value on any of the others. Just like with the implicit scheme, we don't necessarily know all of the values to calculate the next time step, but it is possible, making this another implicit scheme.<sup>a</sup>

<sup>a</sup> Image Source: M. Hjorth-Jensen, Lecture notes in Computational Physics. p. 313, figure 10.3. Course URL: <https://www.uio.no/studier/emner/matnat/fys/FYS3150/index-eng.html>

## ACKNOWLEDGEMENTS

I have found most of the numerical methods discussed in this report in the lecture notes of M. Hjorth-Jensen for the course in Computational Physics at the University of Oslo, which can be found at the following URL: <https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf>. Most of the relevant stuff regarding PDEs are found in chapter 10 (p. 301).

The lecture notes are so organized and easy to follow as to I felt like it needed a general acknowledgement. Cheers!

## HYPERLINK LOCATIONS

The GIFs that are linked to can in general be found at the GitHub Repository in the following URL: <https://github.com/Jan-Egil/FYS3150/tree/master/Project5/Project5/GIF>

The full GitHub repository containing all the code and other files can be found at the following URL: <https://github.com/Jan-Egil/FYS3150>

The reason I will explicitly link it here is in case this report is printed out, rendering the hyperlinks useless. This is simply an attempt at making the results as well documented as possible.

- 
- [1] Matthews, P.C. Vector Calculus. London: Springer, 1998. Print. Springer Undergraduate Mathematics Ser. - p. 132, - equations (8.2) and (8.3).
  - [2] The tridiagonal matrix solver that we are using is discussed further in the article 'A Numerical Analysis on Approximating a Second-Order Differential Equation': [https://github.com/adnvre/Fys3150/blob/master/Project1/Rapport\\_Project1.pdf](https://github.com/adnvre/Fys3150/blob/master/Project1/Rapport_Project1.pdf) (A. Vrevic, K. Sæbjørnsen & J.E. Ødegård).
  - [3] These equations are listed on page 3 in the following article: 'A Numerical Analysis on Approximating a Second-Order Differential Equation': [https://github.com/adnvre/Fys3150/blob/master/Project1/Rapport\\_Project1.pdf](https://github.com/adnvre/Fys3150/blob/master/Project1/Rapport_Project1.pdf) (A. Vrevic, K. Sæbjørnsen J.E. Ødegård).
  - [4] This equation is found in Project 2 in the course 'FYS3150 - Computational Physics' at the University of Oslo, at the following URL: <http://compphysics.github.io/ComputationalPhysics/doc/Projects/2019/Project2/pdf/Project2.pdf> (Morten Hjorth-Jensen, 2019).
  - [5] M. Hjorth-Jensen, Lecture notes in Computational Physics, University of Oslo (2013), p. 305. Course URL: <https://www.uio.no/studier/emner/matnat/fys/FYS3150/index-eng.html>.
  - [6] M. Hjorth-Jensen, Lecture notes in Computational Physics, University of Oslo (2013), p. 312, table 10.1. Course URL: <https://www.uio.no/studier/emner/matnat/fys/FYS3150/index-eng.html>.
  - [7] M. Hjorth-Jensen, Lecture notes in Computational Physics, University of Oslo (2013), p. 66, section 6.6. Course URL: <https://www.uio.no/studier/emner/matnat/fys/FYS3150/index-eng.html>.
  - [8] M. Hjorth-Jensen, Lecture notes in Computational Physics, University of Oslo (2013). p. 311, , equations 10.10 and onwards for a more thorough derivation of this scheme. Course URL: <https://www.uio.no/studier/emner/matnat/fys/FYS3150/index-eng.html>.