

# Numerical Simulation of Heat Diffusion in the Lithosphere

Steinn Hauser and Simen Håpnes

(Dated: December 14, 2018)

Research regarding the diffusion equation and a simulation of the heat flow in the lithosphere is presented. One dimensional cases with known analytic solutions were simulated with the explicit and implicit Euler schemes, as well as the Crank-Nicolson method. Although the explicit scheme was limited to the Courant-Friedrichs-Lewy condition, it was found to be the most accurate. A two dimensional simulation with closed form analytic solutions was conducted using the implicit Euler scheme solved iteratively using Jacobi's method. The results were found to exhibit errors which were closely related to the spatial and temporal step sizes. The two dimensional model was then used to simulate a subducted  $120km$  deep and  $150km$  wide slice of the lithosphere segmented in three layers: the upper crust ( $0-20km$ ), the lower crust ( $20-40km$ ) and the mantle ( $40-120km$ ). Four simulation variants of heat diffusion over  $1Ga$  ( $10^9$  years) are conducted. Two such steady state simulations (without subduction) had results consistent with their corresponding analytic solutions. Two subduction simulations had additional heat production from radioactive isotopes in the mantle ( $^{238}U$ ,  $^{232}Th$  and  $^{40}K$ ). The two models were simulated with and without decay of the radioactive isotopes to research the impact of the decay on the temperature of the lithosphere. The simulation with decay was found to reach a lower temperature in relation to the non-decaying simulation. The maximum temperature difference was found to reach  $34.6^\circ C$  at a depth of  $68.4km$  after  $1Ga$ .

## I. INTRODUCTION

The diffusion phenomena in physics is widely applicable to modernized research. The effects of diffusion presented in this report are exhibited (among other things) by various thermodynamic and biological systems. The diffusion equations presented in this report have for example been used to conduct invasion biology research models (see work done by Lionel Roques et. al regarding invasive insect modelling [4]).

The research presented regards the transport of heat in medium connected to heating and cooling sources. The medium is modelled by a grid of points which can interact with its neighbors, spreading the heat throughout the medium. The degree of the heat spread through the medium is dictated by parameters characteristic of the medium simulated. These parameters are all included in the general *diffusion equation*. Three algorithms which model the diffusion equation are presented as simulation variants: The *Explicit* and *Implicit* Euler schemes, and the *Crank Nicolson* method. These three models are first presented in one spatial dimension (1D) before subsequent implementation of the implicit scheme into two dimensions (2D). The diffusion equation presented has analytic solutions in both the 1D and 2D cases. The numerical data gathered from the simulations will be compared with these analytic expressions for assessment and affirmation of the validity of the model.

Finally, this research will be implemented for a more specific study with regards to a relevant geological research case: The study of heat diffusion in the lithosphere. The lithosphere will be modeled as three separate layers which have different natural heat productions. All layers are assumed to have the same density, thermal conductivity, and heat capacity. The research will additionally include the heat produced by radioactive material within the lowest layer of the lithosphere, taking into account the half-lives of the elements. Geological evidence suggest radioactive enrichment in the mantle caused by a subducted slab  $1.02-1.06$  ago [2]. The research of additional heat production caused by these isotopes is a central objective of the report.

This report is designed to brief the reader on the 1D and 2D analytic solutions to the problem at hand, before explaining in detail the algorithms utilized for simulation. This is all presented in the *Theory* section. After a briefing of the algorithms and research theory, the methods and their objectives are described in detail in the *Method* section. The results of the methods and any appropriate discussions which arise are subsequently presented in the *Results* and *Discussion* sections. Finally, a comprehensive conclusion of the study is stated to summarize.

The *Method*, *Results* and *Discussion* sections are split into subsections to cover the more specific areas of research which the study goes into. These subsections have as titles: One Dimension, Two Dimensions, and Lithosphere. The simulations presented are coded in C++ and plotted (and animated) in Python. The research presented is a cooperative effort between Steinn Hauser and Simen Håpnes, and the simulation scripts and their results can be found in the following public GitHub repository: <https://github.com/steinnhauser/FYS3150>.

## II. THEORY

### A. Analytic solutions of the Diffusion Equation

When researching a subject numerically it is exceptionally useful to have benchmark analytic expressions for comparison. Without these, there is no way to assess the degree of precision of the numerical results. The reliability of the simulation model and confidence in the execution method is greatly improved if there is little to no difference between the results and the analytic predictions. The analytic expressions of the one dimensional and two dimensional cases will therefore be presented.

The one dimensional diffusion model studied regards the following relation:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} \quad (1)$$

This is known as the diffusion partial differential equation (PDE) and can be solved analytically when given the appropriate boundary conditions. The one dimensional boundary conditions of the research regard a source (of heat or other phenomena) at  $x = L$  and a drain ( $u = 0$ ) at  $x = 0$ . These conditions are constant and are not influenced by their environment:

$$u(0, t) = 0, \quad u(L, t) = 1, \quad t \geq 0. \quad (2)$$

To solve the PDE, the equation  $u$  is split up into the sum of it's steady state solution  $u_{SS}$  and the transient solution  $u_{TR}$ . The expression for  $u$  is rewritten to:

$$u(x, t) = u_{SS}(x, t) + u_{TR}(x, t) \quad (3)$$

These two are the key to solving the PDE analytically. The steady state solution (by its definition  $\frac{\partial}{\partial t} u_{SS} = 0$ ) is simply given by:

$$\frac{\partial^2}{\partial x^2} u_{SS} = 0 \quad (4)$$

$$\frac{\partial}{\partial x} u_{SS} = C_1 \quad (5)$$

$$u_{SS} = C_1 x + C_2 \quad (6)$$

The integration constants  $C_1$  and  $C_2$  can be calculated by way of the boundary conditions:

$$u_{SS}(0, t) = 0 \Rightarrow C_2 = 0 \quad (7)$$

$$u_{SS}(L, t) = 1 \Rightarrow C_1 = 1/L \quad (8)$$

$$u_{SS}(x, t) = x/L \quad (9)$$

The transient solution  $u_{TR}$  can then be found by similar reasoning. However, the boundary conditions of the transient solution are different due to the steady state solution ensuring that  $u(0, t) = 0$  and  $u(L, t) = 1$ . The boundary conditions of the transient solution is then given by:

$$u_{TR}(0, t) = 0, \quad u_{TR}(L, t) = 0, \quad t \geq 0. \quad (10)$$

These boundary conditions are both zero due to the fact that the steady state solution  $u_{SS}$  ensures that  $u$  has the

proper boundaries. The transient solution can then be found by the following:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} \quad (11)$$

$$\frac{\partial^2}{\partial x^2} (u_{SS} + u_{TR}) = \frac{\partial}{\partial t} (u_{SS} + u_{TR}) \quad (12)$$

Inserting that the steady state is independent of time ( $\frac{\partial}{\partial t} u_{SS} = 0$ ) and has a second derivative of zero returns:

$$\frac{\partial^2}{\partial x^2} u_{TR}(x, t) = \frac{\partial}{\partial t} u_{TR}(x, t) \quad (13)$$

This PDE has the characteristic of being *separable*, meaning that the function  $u_{TR}(x, t)$  can be rewritten as two independent  $x$  and  $t$  functions  $u_{TR}(x, t) = F(x)G(t)$ . The PDE is then given as:

$$G \frac{\partial^2 F}{\partial x^2} = F \frac{\partial G}{\partial t} \quad (14)$$

$$\Rightarrow \frac{1}{F} \frac{\partial^2 F}{\partial x^2} = \frac{1}{G} \frac{\partial G}{\partial t} = \text{constant}. \quad (15)$$

Since the right and left sides of the equation are equal and independent of each other (one varies with  $x$  and the other with  $t$ ), then these must both be constant. This constant will be expressed by the factor  $-\lambda^2$ . The two equations to solve for  $u_{TR}$  are then:

$$\frac{1}{F} \frac{\partial^2 F}{\partial x^2} = -\lambda^2 \quad (16)$$

$$\frac{1}{G} \frac{\partial G}{\partial t} = -\lambda^2, \quad (17)$$

where  $u_{TR} = F \cdot G$ . Beginning with equation 16: The general solution of this equation is given by:

$$\frac{\partial^2 F}{\partial x^2} = -\lambda^2 F \quad (18)$$

$$\Rightarrow F = A \sin(\lambda x) + B \cos(\lambda x), \quad (19)$$

where  $A$  and  $B$  are arbitrary scalars. Inserting the boundary conditions for the transient solution described in equation 10 returns:

$$F(0) = 0 \Rightarrow B = 0 \quad (20)$$

$$F(L) = 0 \Rightarrow A \sin(\lambda L) = 0 \quad (21)$$

$$\Rightarrow \lambda_n = \frac{n\pi}{L}, \quad (22)$$

where  $n$  is a positive integer. This gives us the general spatial relation:

$$F_n(x) = A_n \sin(\lambda_n x) \quad (23)$$

Following is the expression for the temporal  $u_{TR}$  function  $G(t)$  (using  $\lambda \rightarrow \lambda_n$ ):

$$\frac{\partial}{\partial t} G = -\lambda_n^2 G \quad (24)$$

$$\Rightarrow G = C e^{-\lambda_n^2 t} \quad (25)$$

Combining equations 23 and 25 returns the transient solution:

$$u_{TR,n}(x, t) = a_n \sin(\lambda_n x) e^{-\lambda_n^2 t}, \quad (26)$$

where  $\lambda_n = n\pi/L$ . Since any positive integer value of  $n$  satisfies the PDE, there is an infinite amount of solutions to the equation. Since all the PDE solutions are linear and homogeneous, the final general solution can be expressed as the following linear combination of infinite individual solutions:

$$u_{TR}(x, t) = \sum_{n=1}^{\infty} a_n \sin(\lambda_n x) e^{-\lambda_n^2 t} \quad (27)$$

Since the PDE solutions are linear and homogeneous, the Sturm-Liouville Equation can be applied to calculate the coefficient  $a_n$ . The general solution of  $u = u_{SS} + u_{TR}$  is then given by (see Chapter 7 of Mary L. Boas: Mathematical Methods in The Physical Sciences [1])

$$u(x, t) = \frac{1}{L} x + \sum_{n=1}^{\infty} a_n e^{-\lambda_n^2 t} \sin(\lambda_n x). \quad (28)$$

The Fourier coefficient  $a_n$  is given by:

$$a_n = \frac{-2}{L} \int_0^L x \sin(\lambda_n x) dx \quad (29)$$

$$= \frac{-2}{L} \left[ \frac{\sin(\lambda_n x) - \lambda_n x \cos(\lambda_n x)}{\lambda_n^2} \right] \Big|_0^L \quad (30)$$

$$= \frac{-2}{L} \left[ \frac{\sin(\lambda_n L) - \lambda_n L \cos(\lambda_n L)}{\lambda_n^2} \right]. \quad (31)$$

Notice that the transient solution  $u_{TR} \rightarrow 0$  in the limit  $t \rightarrow \infty$ , such that the equilibrium state  $u(x, \infty)$  is only expressed by the steady state  $u_{SS}$  contribution.

In two dimensions, the analytic solution is calculated in a similar fashion. The two dimensional PDE is given by:

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

Using simple boundary conditions is very useful in reducing the complexity of the solution, so following boundary conditions will be studied in the two-dimensional wave equation:

$$u(0, y, t) = u(L, y, t) = 0, \quad y \in (0, L), \quad t \geq 0 \quad (33)$$

$$u(x, 0, t) = u(x, L, t) = 0, \quad x \in (0, L), \quad t \geq 0 \quad (34)$$

This simplified boundary condition problem does not require calculations of the Fourier coefficients and is therefore not CPU-heavy when calculating. Separation of  $u(x, y, t)$  such that  $u(x, y, t) = F(x)G(y)H(t)$  allows for the solution to be written as (assuming that  $L = 1$ )[3]:

$$u(x, y, 0) = \sin(\pi x) \sin(\pi y) \quad x \vee y \in [0, 1] \quad (35)$$

$$\Rightarrow u(x, y, t) = e^{-2\pi^2 t} \sin(\pi x) \sin(\pi y) \quad t \geq 0 \quad (36)$$

This solution resembles a circular paraboloid which is centered in  $(x, y) = (1/2, 1/2)$  and equals zero at the boundaries. Notice that the entire solution equals zero in the limit  $t \rightarrow \infty$ , meaning that the heat from the middle of the medium is dissipated by the boundaries.

## B. Numerical Implementation of the Diffusion Equation

The diffusion model implemented regards the following general relation:

$$\vec{\nabla} \cdot (k \vec{\nabla} u) = a \frac{\partial u}{\partial t} \quad (37)$$

Assuming that  $k$  and  $a$  are constant scalars, produces the general scaled relation:

$$\nabla^2 u = \frac{\partial u}{\partial t}. \quad (38)$$

In one dimension, the scaled diffusion equation is given by:

$$\frac{\partial^2 u(x, t)}{\partial x^2} = \frac{\partial u(x, t)}{\partial t} \quad (39)$$

To study the partial differential equations numerically, a discretization of the spatial and time dimensions is implemented. This involves mapping the parameters  $x$  and  $t$  such that they are described by an initial value and step size  $h$ :  $x \rightarrow x_i = x_0 + i \cdot h_x$  and  $t \rightarrow t_j = t_0 + j \cdot h_t$ . This discretization allows for an approximation of the derivative by way of Taylor expansions of  $u$ . The Taylor expansion for  $u(x_i, t_j)$  with a small step size  $h_t$  (utilizing discretized notation  $u(x_i, t_j + h_t) = u_{i,j+1}$ ) along the  $t$  axis is as follows:

$$u_{i,j\pm 1} = u_{i,j} \pm h_t u_{i,j}^{(1t)} + \mathcal{O}(h_t^2) \quad (40)$$

The notation  $u^{(nt)}$  indicates the  $n$ -th derivative of the function  $u$  with respect to  $t$ . This notation will be used throughout the report. The last term  $\mathcal{O}(h_t^2)$  represents the truncation error of the approximation, proportional to  $\mathcal{O}(h_t^2) \sim h_t^2$ . Rewriting this equation in terms of the first derivative returns:

$$\frac{\partial u}{\partial t} \approx u_{i,j}^{(1t)} = \pm \frac{u_{i,j\pm 1} - u_{i,j}}{h_t} + \mathcal{O}(h_t) \quad (41)$$

The difference between the explicit and implicit Euler schemes presented depends on the  $\pm$  sign in this equation. Notice that the truncation error  $\mathcal{O}$  is only proportional to  $\sim h_t$  in both cases. These are the two temporal approximations of  $\frac{\partial}{\partial t} u(x_i, t_j)$  which will be used for the simulation.

Similarly, the Taylor expansion for  $u(x_i, t_j)$  with a small step size  $h_x$  along the  $x$  axis is as follows:

$$u_{i\pm 1,j} = u_{i,j} \pm h_x u_{i,j}^{(1x)} + \frac{h_x^2}{2!} u_{i,j}^{(2x)} + \frac{h_x^3}{3!} u_{i,j}^{(3x)} + \mathcal{O}(h_x^4) \quad (42)$$

Here  $u^{(nx)}$  indicates the  $n$ -th derivative of the function  $u$  with respect to  $x$ . In this case the error is proportional to the step size  $h_x$  to the fourth power  $\mathcal{O}(h_x^4) \sim h_x^4$ . A combination of two such expansions (varying the sign) results in the following:

$$u_{i+1,j} + u_{i-1,j} = 2u_{i,j} + \frac{2}{2!} h_x^2 u_{i,j}^{(2x)} + \mathcal{O}(h_x^4) \quad (43)$$

Rewriting this expression produces the following approximation to the second spatial derivative:

$$\frac{\partial^2 u}{\partial x^2} \approx u_{i,j}^{(2x)} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} + \mathcal{O}(h_x^2). \quad (44)$$

This is the general approximation of  $\frac{\partial}{\partial x}u(x_i, t_j) \approx u_{i,j}^{(2x)}$  used in the study. Three algorithms are presented to solve the general differential equation numerically. The *implicit* and *explicit* Euler schemes are the basis of simulation. A combination of the two, namely the Crank-Nicolson method, is also presented for comparison. All these methods utilize the partial derivative approximations presented, with slight variants to each of them.

### C. Simulation Variants

#### 1. Explicit Scheme

The explicit solution is given by inserting  $\pm = +$  into equation 40, and is as follows (implementing discretized notation  $u(x_i, t_j) \rightarrow u_{i,j}$ ):

$$u_{i,j+1} = u_{i,j} + h_t u_{i,j}^{(1t)} + \mathcal{O}(h_t^2) \quad (45)$$

$$\Rightarrow u_{i,j}^{(1t)} = \frac{u_{i,j+1} - u_{i,j}}{h_t} + \mathcal{O}(h_t), \quad (46)$$

Combining the derivative approximations by insertion of equations 45 and 44 into equation 39 returns:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} \quad (47)$$

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(h_x)^2} = \frac{u_{i,j+1} - u_{i,j}}{h_t} \quad (48)$$

$$\Rightarrow u_{i,j+1} = \alpha u_{i-1,j} + (1 - 2\alpha)u_{i,j} + \alpha u_{i+1,j}, \quad (49)$$

where  $\alpha = h_t/h_x^2$ . Recall that the spatial approximation has a truncation error of  $\mathcal{O}(h_x^2)$  and the temporal approximation has a truncation error  $\mathcal{O}(h_t)$ .

This system of equations can be rewritten in terms of a matrix-vector multiplication  $\hat{A}x = b$  given by (example of a  $3 \times 3$  grid):

$$\begin{bmatrix} (1-2\alpha) & \alpha & 0 \\ \alpha & (1-2\alpha) & \alpha \\ 0 & \alpha & (1-2\alpha) \end{bmatrix} \cdot \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \end{bmatrix} = \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \end{bmatrix} \quad (50)$$

For a simulation of the system over  $N$  time steps, the matrix  $\hat{A}$  is applied  $N$  times, formally written as:

$$\begin{bmatrix} (1-2\alpha) & \alpha & 0 \\ \alpha & (1-2\alpha) & \alpha \\ 0 & \alpha & (1-2\alpha) \end{bmatrix}^N \cdot \begin{bmatrix} u_{1,0} \\ u_{2,0} \\ u_{3,0} \end{bmatrix} = \begin{bmatrix} u_{1,N} \\ u_{2,N} \\ u_{3,N} \end{bmatrix} \quad (51)$$

This system will either converge or diverge towards a steady state solution upon a large number  $N$  matrix multiplications. Whether or not the solution  $u(x_i, t_j)$  converges in the limit  $t_j \rightarrow \infty$  depends on the eigenvalues of the matrix  $\hat{A}$ . This condition for an explicit scheme convergence is known as the *Courant-Friedrichs-Lewy* condition, and depends on the spectral radius  $\rho_{max}$  of the matrix  $\hat{A}$ . The spectral radius of the matrix is defined as the largest absolute eigenvalue of the matrix:

$$\rho(A) \equiv \max \left\{ |\lambda| : \det(\hat{A} - \lambda \hat{I}) = 0 \right\} \quad (52)$$

If the spectral radius of  $\hat{A}$  is larger than one  $\lambda_{max} > 1$ , then there will be a corresponding eigenvector  $\vec{v}_0$  of  $A$  which is scaled upwards by  $\vec{v}_N = \lambda_{max}^N \vec{v}_0$ , diverging to infinity in the limit of  $N \rightarrow \infty$ . The condition for convergence is therefore given by  $\rho(\hat{A}) < 1$ .

This condition can be formally shown to be given by [3]:

$$\alpha < \frac{1}{2} \quad (53)$$

$$\frac{h_t}{h_x^2} \leq \frac{1}{2} \quad (54)$$

#### 2. Implicit Scheme

The implicit solution is given by inserting  $\pm = -$  into equation 40, and is as follows:

$$u(x_i, t_j - h_t) = u(x_i, t_j) - h_t u^{(1t)}(x_i, t_j) + \mathcal{O}(h_t^2) \quad (55)$$

$$\Rightarrow \frac{\partial u}{\partial t} \approx u_{i,j}^{(1t)} = \frac{u_{i,j} - u_{i,j-1}}{h_t} + \mathcal{O}(h_t), \quad (56)$$

Combining the derivative approximations by insertion of equations 56 and 44 into equation 39 returns:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} \quad (57)$$

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(h_x)^2} = \frac{u_{i,j} - u_{i,j-1}}{h_t} \quad (58)$$

$$\Rightarrow u_{i,j-1} = -\alpha u_{i-1,j} + (1 + 2\alpha)u_{i,j} - \alpha u_{i+1,j} \quad (59)$$

This can be rewritten to the following matrix-vector multiplication by the same reasoning used in the explicit scheme. The implicit scheme expressed by matrix-vector multiplication is then given by:

$$\begin{bmatrix} (1+2\alpha) & -\alpha & 0 \\ -\alpha & (1+2\alpha) & -\alpha \\ 0 & -\alpha & (1+2\alpha) \end{bmatrix} \cdot \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \end{bmatrix} = \begin{bmatrix} u_{1,j-1} \\ u_{2,j-1} \\ u_{3,j-1} \end{bmatrix} \quad (60)$$

$N$  repeated matrix multiplications are now expressed as:

$$\begin{bmatrix} (1+2\alpha) & -\alpha & 0 \\ -\alpha & (1+2\alpha) & -\alpha \\ 0 & -\alpha & (1+2\alpha) \end{bmatrix}^N \cdot \begin{bmatrix} u_{1,N} \\ u_{2,N} \\ u_{3,N} \end{bmatrix} = \begin{bmatrix} u_{1,0} \\ u_{2,0} \\ u_{3,0} \end{bmatrix} \quad (61)$$

Multiplying with the inverse of the matrix on both sides returns:

$$\begin{bmatrix} (1+2\alpha) & -\alpha & 0 \\ -\alpha & (1+2\alpha) & -\alpha \\ 0 & -\alpha & (1+2\alpha) \end{bmatrix}^{-N} \cdot \begin{bmatrix} u_{1,0} \\ u_{2,0} \\ u_{3,0} \end{bmatrix} = \begin{bmatrix} u_{1,N} \\ u_{2,N} \\ u_{3,N} \end{bmatrix} \quad (62)$$

It can be shown that this matrix has eigenvalues which always satisfy  $\lambda_i > 1$  [3]. This means that the implicit method converges to the solution for any value of  $\alpha$ . This is due to the inverse of the matrix being applied, such that the eigenvectors are instead scaled by the factor  $\lambda^{-N} \vec{v}_0 \rightarrow 0$  in the limit  $N \rightarrow \infty$ .

### 3. Crank-Nicolson scheme

The one dimensional Crank-Nicolson scheme is found by combination of the two Implicit and Explicit (in one dimension) schemes. There are several ways to derive this scheme, though can be found by simply inserting  $\theta = 1/2$  in the  $\theta$ -rule [3], returning:

$$-\alpha u_{i-1,j} + (2 + 2\alpha)u_{i,j} - \alpha u_{i+1,j} = \alpha u_{i-1,j-1} + (2 - 2\alpha)u_{i,j-1} + \alpha u_{i+1,j-1} \quad (63)$$

In matrix-vector form it can be expressed as:

$$(2\hat{I} + \alpha\hat{B})V_j = (2\hat{I} - \alpha\hat{B})V_{j-1}, \quad (64)$$

where the vector  $V_j = [u_{0,j}, u_{1,j}, \dots]$ ,  $\alpha = h_t/h_x^2$ , and the matrix  $\hat{B}$  is given by:

$$\hat{B} = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ 0 & -1 & 2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -1 \\ 0 & 0 & \dots & -1 & 2 \end{bmatrix}. \quad (65)$$

This system can also be shown to be stable for all variants of  $\alpha$ [3]. The Crank Nicolson algorithm is derived more fundamentally in appendix A, where both temporal and spatial truncation errors are included and shown to be  $\mathcal{O}(h_x^2)$  and  $\mathcal{O}(h_t^2)$ . In summary, the three one-dimensional simulation variants have the following characteristics:

Scheme	Truncation errors	Stability Condition
Explicit	$\mathcal{O}(h_x^2), \mathcal{O}(h_t)$	$\alpha \leq 1/2$
Implicit	$\mathcal{O}(h_x^2), \mathcal{O}(h_t)$	None
Crank-Nicolson	$\mathcal{O}(h_x^2), \mathcal{O}(h_t^2)$	None

TABLE I. A summary of the three simulation variants researched in in one dimension. The parameter  $\alpha$  is given by  $\alpha = h_t/h_x^2$ .

## D. Two Dimensions

### 1. Implicit Scheme

Generally in two dimensions, the following scaled partial differential equation will be solved:

$$\frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2} = \frac{\partial u(x, y, t)}{\partial t}, \quad (66)$$

Discretized notation of these variables will be implemented such that  $u(x_i, y_i, t_l) = u_{i,j}^l$ . Taylor expansion of the temporal derivative and both second degree spatial derivatives (in the same fashion as before) produces the following implicit two dimensional approximations:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1,j}^l - 2u_{i,j}^l + u_{i-1,j}^l}{(h_x)^2} + \mathcal{O}(h_x^2) \quad (67)$$

$$\frac{\partial^2 u}{\partial y^2} \approx \frac{u_{i,j+1}^l - 2u_{i,j}^l + u_{i,j-1}^l}{(h_y)^2} + \mathcal{O}(h_y^2) \quad (68)$$

$$\frac{\partial u}{\partial t} \approx \frac{u_{i,j}^l - u_{i,j}^{l-1}}{h_t} + \mathcal{O}(h_t) \quad (69)$$

These describe any general time step  $l$ . Combining these equations produces the following relation for equally spaced ( $h_x = h_y$ ) grid points  $u_{i,j}^l$ :

$$u_{i,j}^l = \frac{\alpha}{1 + 4\alpha} \Delta_{i,j}^l + \frac{1}{4\alpha} u_{i,j}^{l-1}, \quad (70)$$

where  $\Delta_{i,j}^l = (u_{i+1,j}^l + u_{i-1,j}^l + u_{i,j+1}^l + u_{i,j-1}^l)$  and  $\alpha = h_t/h_x^2$ . Solving the equation with respect to the grid point  $u_{i,j}^l$  is not possible, as the factor  $\Delta_{i,j}^l$  describes the current state of it's neighbouring points. The only known factors in the time step  $l$  are the boundary conditions and the previous point  $u_{i,j}^{l-1}$ . This equation must therefore be solved in a different fashion.

Simplifying this for the first time step  $t = t_0$  (and setting  $h_x = h_y = h$ ) allows us to examine how this system of equations can be solved (ignoring the time dependency indices and setting  $(\frac{\partial u}{\partial t})_{l_0} = \rho_{i,j}$ ), simplifying the PDE to:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho_{i,j}. \quad (71)$$

Inserting the numerical approximations of the partial derivatives returns the following solution to  $u_{i,j}$ :

$$u_{i,j} = \frac{1}{4}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) + \frac{h_x^2}{4}\rho_{i,j} \quad (72)$$

Assuming a  $4 \times 4$  mesh grid and inserting the indices  $i \in [0, 3]$  and  $j \in [0, 3]$  (and writing in terms of  $\tilde{\rho}_{i,j} = h_x^2 \rho_{i,j}$ ) produces the following system of equations:

$$\mathbf{i=1,j=1:} \quad 4u_{1,1} - u_{2,1} - u_{1,2} - u_{1,0} - u_{0,1} = \tilde{\rho}_{1,1}$$

$$\mathbf{i=1,j=2:} \quad 4u_{1,2} - u_{2,2} - u_{1,1} - u_{0,2} - u_{1,3} = \tilde{\rho}_{1,2}$$

$$\mathbf{i=2,j=1:} \quad 4u_{2,1} - u_{2,2} - u_{1,1} - u_{2,0} - u_{3,1} = \tilde{\rho}_{2,1}$$

$$\mathbf{i=2,j=2:} \quad 4u_{2,2} - u_{2,1} - u_{1,2} - u_{2,3} - u_{3,2} = \tilde{\rho}_{2,2}$$

Establishing boundary conditions in the end points of the grid such that all points with an  $x$  or  $y$  index equal to 0 or 3 allows for rewriting the equation in terms of the known and unknown variables:

$$\mathbf{i=1,j=1:} \quad 4u_{1,1} - u_{2,1} - u_{1,2} = \tilde{\rho}_{1,1} + u_{1,0} + u_{0,1}$$

$$\mathbf{i=1,j=2:} \quad -u_{1,1} + 4u_{1,2} - u_{2,2} = \tilde{\rho}_{1,2} + u_{0,2} + u_{1,3}$$

$$\mathbf{i=2,j=1:} \quad -u_{1,1} + 4u_{2,1} - u_{2,2} = \tilde{\rho}_{2,1} + u_{2,0} + u_{3,1}$$

$$\mathbf{i=2,j=2:} \quad -u_{2,1} - u_{1,2} + 4u_{2,2} = \tilde{\rho}_{2,2} + u_{2,3} + u_{3,2}$$

This system of equations can then be expressed as the following matrix-vector multiplication  $\hat{A}x = b$ :

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix} \cdot \begin{bmatrix} u_{1,1} \\ u_{1,2} \\ u_{2,1} \\ u_{2,2} \end{bmatrix} = \begin{bmatrix} \tilde{\rho}_{1,1} + u_{1,0} + u_{0,1} \\ \tilde{\rho}_{1,2} + u_{0,2} + u_{1,3} \\ \tilde{\rho}_{2,1} + u_{2,0} + u_{3,1} \\ \tilde{\rho}_{2,2} + u_{2,3} + u_{3,2} \end{bmatrix} \quad (73)$$

This matrix-vector multiplication is of course specific for a  $4 \times 4$  grid, though it behaves in the same fashion for larger grids. This matrix  $\hat{A}$  is a positive definite and diagonally dominant matrix, and becomes a sparse matrix when enlarged. The standard solver for the vector  $x = [u_{1,1}, u_{1,2}, u_{2,1}, u_{2,2}]$  (in the case of positive definite and diagonally dominant matrices  $\hat{A}$ ) is to use an iterative solver.

## 2. Jacobi method

The Jacobi method is designed to solve a set of linear equations described by the matrix-vector multiplication

$$\hat{A}x = b \quad (74)$$

with respect to the unknown vector  $x$ . There are two requirements for the method: the matrix  $\hat{A}$ 's spectral radius  $\rho(\hat{A})$  must be strictly less than 1, and it must be diagonally dominant. If both conditions are fulfilled, then the iterative method is guaranteed to converge to the solution  $x$ .

The method works as follows: An initial candidate  $x_0$  of the solution is presented. Any randomly generated vector  $x_0$  is eligible for the Jacobi method as long as the matrix  $\hat{A}$  meets the requirements mentioned. The Jacobi method takes this input  $x_0$  and iterates  $k$  times until a sufficient solution  $\hat{A}x_k \approx b$  is reached.

In the case of the two dimensional diffusion equation, Jacobi's method is applied to the relation described in equation 70. The guess  $x_0$  is utilized in the neighbouring point factor  $\Delta_{i,j}^l = x_0$ . The equation is used to produce a new guess candidate  $u_{i,j}^l = x_1$ , and represents a single iteration of Jacobi's method from  $x_0$  to  $x_1$ . Once the vector  $x_1$  is calculated, Jacobi's method assesses whether this is a sufficient solution to  $\hat{A}x = b$  by the condition  $|x_1 - x_0| < \epsilon$ . The tolerance  $\epsilon$  is typically a very small number (depending on the desired precision) which dictates when Jacobi's method should stop iterating. If this condition is not met, the  $\Delta_{i,j}^l$  vector is set to equal  $\Delta_{i,j}^l = x_1$  and the process is repeated for a new vector  $x_2$ .

This method is only valid for use for a linear set of equations with a positive definite and diagonally dominant matrix. Jacobi's method by linear algebra is expressed by separating the matrix  $\hat{A}$  into the sum of two matrices  $\hat{D}$  and  $\hat{R}$ . The matrix  $\hat{D}$  consists only of the diagonal elements of  $\hat{A}$  and the matrix  $\hat{R}$  includes the non-diagonals. The equation  $\hat{A}x = b$  is then given by:

$$(\hat{D} + \hat{R})x = b, \quad (75)$$

$$\hat{D}x + \hat{R}x = b, \quad (76)$$

$$\hat{D}x = b - \hat{R}x \quad (77)$$

$$x = \hat{D}^{-1}(b - \hat{R}x) \quad (78)$$

The matrix form of Jacobi's iterative method is then given by:

$$x_{k+1} = \hat{D}^{-1}(b - \hat{R}x_k), \quad k \in 0, 1, 2, \dots \quad (79)$$

## E. Lithosphere

### 1. Heat Diffusion

The study of the Lithosphere will involve separating the study into three main areas of research: the upper crust (0 – 20km depth), lower crust (20 – 40km depth), and the mantle (40 – 120km depth). The temperature boundaries

at the surface and at the depth 120km are set to be 8°C and 1300°C respectively throughout the lithosphere study. These areas all have their own designated heat production systems, and can all be described by the general temperature  $T$  diffusion equation:

$$\vec{\nabla}(k\vec{\nabla}T) + Q_i = \rho c_p \frac{\partial T}{\partial t}. \quad (80)$$

$\rho$  is the density in  $[kg/m^3]$ ,  $k$  is the thermal conductivity in  $[W/m/^\circ C]$ ,  $c_p$  is the specific heat capacity in  $J/kg/^\circ C$ , and  $Q_i$  is the heat production of the system  $i \in [1, 2, 3]$  (for upper, lower, and mantle respectively) in  $\mu W/m^3$ . The heat productions of the areas are given by (before radioactive enrichment):

Heat production	Upper Crust $Q_1$	Lower Crust $Q_2$	Mantle $Q_3$
$Q_i, [\mu W/m^3]$	1.4	0.35	0.05

TABLE II. The heat productions of the different depth systems. The upper crust ranges between the depth of 0 – 20km, the lower crust has depth 20 – 40km and the mantle is in the depth 40 – 120km.

This equation has analytic steady state solutions similar to the 1D and 2D cases presented previously. These solutions can be compared with the numeric data for assessment of the simulation. The steady state solutions are derived in Appendix C.

### 2. Radioactive Enrichment

When studying the heat productions of these materials over long periods of time, the half-lives must also be included for a more accurate depiction of reality. The half-lives of these materials will be implemented such that the heat productions are continuously updated over time. To implement this, the radioactive half-life formula is introduced:

$$\ln\left(\frac{m_t}{m_0}\right) = -kt, \quad (81)$$

where  $m_t$  is the isotope mass at time  $t$ ,  $m_0$  is the original isotope mass at  $t = 0$ ,  $k$  is the material's characteristic decay constant in  $[yrs^{-1}]$ , and  $t$  is the time in  $yrs$ . The characteristic decay constant  $k$  for a material can be found by inserting  $m_t/m_0 = 1/2$  and the half-life time  $t = t_{1/2}$ :

$$k = \frac{\ln(2)}{t_{1/2}} \quad (82)$$

The half-lives, concentrations, and heat productions each material are listed in table III. This model assumes that each material has an equal contribution to heat production, such that the heat production of each material is dictated by the total sum  $Q_{tot,0}$  and the corresponding material concentration:

Isotopes	$U$	$Th$	$K$
Concentration [%]	40	40	20
Half-life [1Ga]	4.47	14.0	1.25
Heat production [ $\mu W/m^3$ ]	0.2	0.2	0.1

TABLE III. The distribution and heat production contributions of the radioactive materials  $U$ ,  $Th$ , and  $K$  in the mantle. The half-lives of each radioactive material in 1Ga is listed as well.

The half-lives of each isotope allows for calculation of the decay constant for each isotope  $k_U$ ,  $k_{Th}$  and  $k_K$ . Table IV lists these in units of  $[yr s^{-1}]$ :

Isotopes	Uranium	Thorium	Potassium
Decay constant $k$ , $[10^{-9} yr s^{-1}]$	0.155	0.0495	0.555

TABLE IV. Characteristic decay constant of the Uranium, Thorium and Potassium isotopes studied. These are given in units of  $[10^{-9} yr s^{-1}]$  and are used to describe the evolution of the isotopes.

This produces the following continuous equation which describes the concentration (and heat production) of a material  $i \in [U, Th, K]$  as a function of the time:

$$m_i(t) = m_0 e^{-k_i t} \quad (83)$$

The additional radioactive enrichment heat production of all the isotopes can then be expressed by:

$$Q_R(t) = \underbrace{0.2e^{-0.155t}}_{Uranium} + \underbrace{0.2e^{-0.0495t}}_{Thorium} + \underbrace{0.1e^{-0.555t}}_{Potassium} \quad (84)$$

This is the expression for the additional heat production caused by the radioactive enrichment. The heat production  $Q_R$  is given in units of  $\mu W/m^3$  and the time  $t$  is given in units  $[10^9 yrs]$ .

### 3. Numerical Lithosphere Implementation

Implementing this into a numerical simulation involves rewriting equation 80 using the same spatial and temporal approximations as presented previously:

$$\frac{\partial^2 T}{\partial x^2} \approx T^{(2x)}(x, y, t) \quad (85)$$

$$\frac{\partial^2 T}{\partial y^2} \approx T^{(2y)}(x, y, t) \quad (86)$$

$$\frac{\partial T}{\partial t} \approx T^{(1t)}(x, y, t) \quad (87)$$

The expressions of the derivative approximations are identical to the two dimensional case presented in equations 67, 68 and 69. Equation 80 is then written as (utilizing discretized notation  $T(x_i, y_j, t_l) \rightarrow T_{i,j}^l$ ):

$$k \left( T_{i,j}^{l(2x)} + T_{i,j}^{l(2y)} \right) + Q = \rho c_p T_{i,j}^{l(1t)} \quad (88)$$

Inserting the numerical expressions for the implicit scheme and setting the spatial steps  $h_x = h_y$  returns:

$$\frac{k}{h_x^2} (\Delta_{i,j}^l - 4T_{i,j}^l) + Q = \rho c_p \frac{T_{i,j}^l - T_{i,j}^{l-1}}{h_t}, \quad (89)$$

where  $\Delta_{i,j}^l$  is given by  $\Delta_{i,j}^l = (u_{i+1,j}^l + u_{i-1,j}^l + u_{i,j+1}^l + u_{i,j-1}^l)$ . Rewriting the equation returns:

$$\frac{h_t k}{\rho c_p h_x^2} (\Delta_{i,j}^l - 4T_{i,j}^l) + \frac{h_t}{\rho c_p} Q = T_{i,j}^l - T_{i,j}^{l-1} \quad (90)$$

Rewriting in terms of  $T_{i,j}^l$  returns:

$$T_{i,j}^l = \frac{1}{1 + 4 \frac{h_t k}{\rho c_p h_x^2}} \left[ \frac{h_t k}{\rho c_p h_x^2} \Delta_{i,j}^l + \frac{h_t}{\rho c_p} Q + T_{i,j}^{l-1} \right] \quad (91)$$

Introducing the scaling parameter  $T_c$  such that the temperature  $T$  is expressed within the interval  $[0, 1]$  returns:

$$T_{i,j}^l = \frac{1}{1 + 4 \frac{h_t k}{\rho c_p h_x^2}} \left[ \frac{h_t k}{\rho c_p h_x^2} \Delta_{i,j}^l + \frac{h_t}{\rho c_p T_c} Q + T_{i,j}^{l-1} \right], \quad (92)$$

where  $T_{i,j}^l$ ,  $T_{i,j}^{l-1}$  and each factor in  $\Delta_{i,j}^l$  is now scaled to be in the interval  $[0, 1]$ . Introducing the spacial and temporal scaling parameters  $x_c$ ,  $y_c$  and  $t_c$  such that the step sizes  $h_x$  and  $h_t$  are mapped to  $h_x x_c$  and  $h_t t_c$  (using that  $h_{new} = h_{old}/scale$ ) respectively, and introducing the parameters

$$\alpha = \frac{h_t}{h_x^2}, \quad \beta = \frac{t_c k}{\rho c_p x_c^2}, \quad \text{and} \quad Q_c = \frac{t_c h_t}{\rho c_p T_c} \quad (93)$$

returns the following expression for the implicit solution:

$$T_{i,j}^l = \frac{1}{1 + 4\alpha\beta} [\alpha\beta\Delta_{i,j}^l + T_{i,j}^{l-1} + Q_c Q] \quad (94)$$

Recalling the truncation errors from the two dimensional case, the errors of this expression are proportional to  $\mathcal{O}(h_x^2)$  and  $\mathcal{O}(h_t)$ . The 2D lithosphere PDE can be shown to produce a matrix-vector multiplication system, where the matrix is diagonally dominant and has a spectral radius is strictly less than one. This system of equations can therefore be solved iteratively in the same fashion as in the two-dimensional case presented previously.

## III. METHOD

### A. One dimension

The one dimensional analysis will regard the stability properties of the three different methods presented. This will be done by comparing the *explicit forward Euler scheme*, *implicit backward Euler scheme* and the *Crank-Nicolson scheme* with the analytic solution from equation 28. The four situations which will be compared have different parameter variants which are listed in table V.

Situation no.	$h_x$	$n_x$	$h_t$	$n_t$	$t$
1	$10^{-1}$	11	$10^{-3}$	$10^3 + 1$	0.1
2	$10^{-1}$	11	$10^{-3}$	$10^3 + 1$	0.2
3	$10^{-2}$	101	$10^{-5}$	$10^5 + 1$	0.1
4	$10^{-2}$	101	$10^{-5}$	$10^5 + 1$	0.2

TABLE V. The four different 1D situations that will be simulated and compared with the analytic solution.  $h_x$  is the step length and  $n_x$  is the number of points. These are calibrated such that  $x$  is scaled to the domain  $x \in [0, 1]$ .  $h_t$  is the temporal step length and  $n_t$  is the number of time steps, also set such that  $t \in [0, 1]$ . The times  $t \in [0.1, 0.2]$  are chosen as the two comparison states.  $t_1 = 0.1$  exhibits a non-equilibrium state, and  $t_2 = 0.2$  exhibits an approximate steady-state.

Two step sizes  $h_x = 10^{-1}$  and  $h_x = 10^{-2}$  and two times  $t_1 = 0.1$  and  $t_2 = 0.2$  are researched. The two times chosen  $t_1 = 0.1$  and  $t_2 = 0.2$  are dictated by the form of the solution: at  $t_1$  the temperature gradient was somewhat curved, and at  $t_2$  the system was close to equilibrium (an approximately linear temperature increase from  $x = 0$  to  $x = 1$ ). The step size in time is dictated by the Courant-Friedrichs-Lewy condition of the explicit scheme presented in equation 54: The temporal step size  $h_t$  is set to  $\frac{1}{10}h_x^2$  to ensure stability for the explicit forward Euler scheme.

The analytic solution was evaluated numerically by equation 28. The Fourier coefficient  $a_n$  was calculated using equation 31, setting  $\infty = 1000$ . This produces an analytic solution which is given by a linear combination of 1000 different sine functions (sufficient accuracy for the research at hand). The aim of the one dimensional study was to reveal which numerical scheme had best correspondence with the analytic solution. This was done by analyses of the numerical error for a variation of step sizes. The absolute error with respect to the analytic results was found by simply subtracting the two solutions for each variant.

### 1. Algorithm for the numerical schemes

All simulations variants had the initial conditions

$$u(x, 0) = 0, \quad 0 \leq x < L, \quad (95)$$

$$u(0, t) = 0, \quad t \geq 0, \quad (96)$$

$$u(L, t) = 1, \quad t \geq 0. \quad (97)$$

The step length  $h_x$  and number of points  $n_x$  were chosen such that  $x \in [0, 1]$ . The explicit scheme simply solved the simulation using equation 49.

The implicit and Crank-Nicolson schemes were expressed by tridiagonal matrix-vector multiplications,  $\hat{A}x = b$ , where  $\hat{A}$  was the tridiagonal matrix presented in equation 62. Both of these schemes were solved using the *Thomas algorithm* (see appendix B for explanation and derivation of the Thomas algorithm). For the implicit scheme, the diagonal elements were  $(1 + 2\alpha)$  and the off-diagonal elements were  $-\alpha$ .

For the Crank-Nicolson scheme, equation 64 was solved by a combination of the two previous methods. The right hand side was first solved iteratively:  $b = (2\hat{I} - \alpha\hat{B})u_{i,j-1}$ . The Thomas algorithm was then used to evaluate the left hand side, where the diagonal elements were  $(2 + 2\alpha)$  and the off-diagonal elements were  $\alpha$ .

### B. Two dimensions

The implicit scheme was chosen for simulation of the two dimensional case. At each time step, all the inner position points must be evaluated using equation 70, which is solved using the Jacobi method presented previously. The algorithm flow chart is illustrated in figure 1:

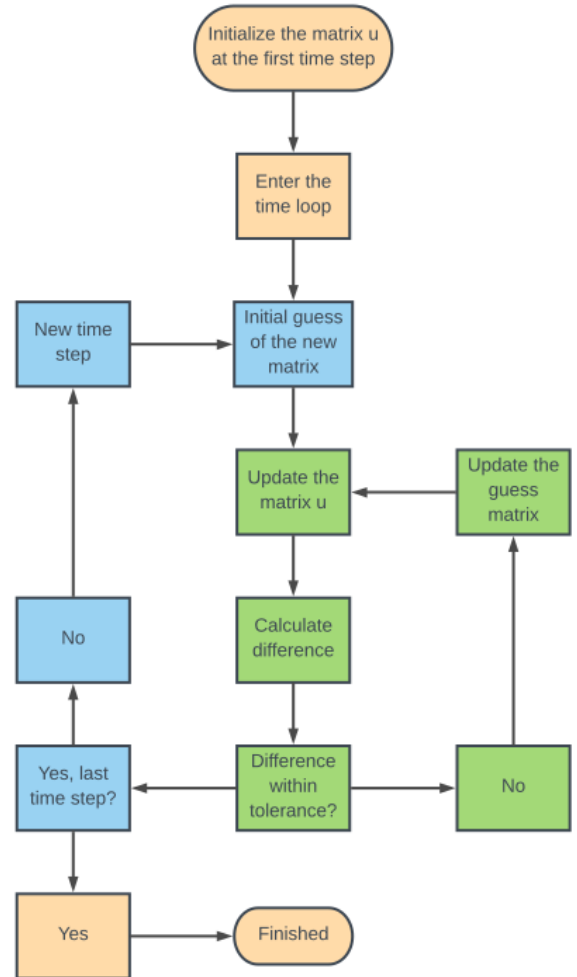


FIG. 1. A flow chart illustrating the Jacobi method applied to solve the two dimensional diffusion equation. The green boxes illustrate the iteration loop, where the solution will always converge towards the exact solution. The blue boxes illustrate the time loop, where one time step is finished once the difference is within a specified tolerance.

The guess matrix can be any matrix since the solution will converge towards the exact solution regardless of the initial guess. The matrix  $u^l$  is updated at the time step  $l$  according to Jacobi's method and equation 70. The convergence requires a certain number of iterations, and



the iteration loop is stopped if either the pre-specified maximum number of iterations is reached or if the difference between the matrices  $u^l$  and  $u_{guess}$  is within a certain tolerance (the maximum number of iterations and tolerance are typically set to  $10^3$  and  $10^{-8}$  respectively).

The initial matrix  $u^0$  is chosen according to the analytic solution:

$$u(x, y, 0) = \sin(\pi x) \sin(\pi y), \quad x, y \in [0, 1], \quad (98)$$

$$u(0, y, t) = u(1, y, t) = 0, \quad t \geq 0, \quad (99)$$

$$u(x, 0, t) = u(x, 1, t) = 0, \quad t \geq 0. \quad (100)$$

Recall that the analytic time evolution of the initial condition is given by:

$$u(x, y, t) = u(x, y, 0)e^{-2\pi^2 t}. \quad (101)$$

This matrix was simulated numerically using the parameters  $h_x = h_y = 0.01$ , number of position points  $n_x = n_y = 101$ , time step  $h_t = 0.0001s$  and number of time points  $n_t = 2001$ . This yields a total time frame  $t \in [0, 0.2s]$ .

The matrix  $u^l$  was saved as a binary file for each time step  $l$  and the results were analyzed in Python in several ways:

- The numerical and analytic solutions were plotted as a 3D surface plot at the time  $t = 0.1s$ . This time was chosen such that the system had not yet reached its steady state.
- The numerical solution was animated using the *pyplot* package from Python's *matplotlib* library. This animation was then saved as a Graphics Interchange Format (GIF) file.
- The absolute error between the numerical and analytic solutions was calculated as the average difference per point between the matrices. This was done for various spatial and temporal step sizes  $h_x$  and  $h_t$  and plotted vs. time (from  $t = 0$  to  $t = 0.2s$ ). The aim was to evaluate how the error depended on the step sizes  $h_x$  and  $h_t$ , and which combinations of  $h_x$  and  $h_t$  yielded the most accurate results.

### C. Lithosphere

The lithosphere simulations involved implementing the implicit Euler's method solution with the Jacobi method as an iterative solver. The algorithm was the same as for the two dimensional algorithm presented in figure 1. The difference between the two was the equation for updating the matrix presented in equation 94. The equation for heat diffusion is properly scaled in the theory section. The heat diffusion in a total area of  $120km$  deep and  $150km$  wide was be simulated with various initial and boundary conditions:

1. No heat production:  $Q = 0$  over the entire area. The horizontal boundary conditions were  $T = 8^\circ$  at the surface and  $T = 1300^\circ$  at the bottom of the mantle. These variables were scaled using a parameter

$T_c = 1292^\circ C$  such that  $T_{scale} \in [0, 1]$ . The vertical boundary conditions were set to have a linear increase in temperature from the surface to the bottom of the mantle.

2. Natural heat production was included. The heat productions are different in the three different layers and are listed in table II. The boundary conditions and the initial state of the matrix will be a temperature as a function of depth given by equations 119, 120, and 121. These are also scaled for  $T_{scale} \in [0, 1]$ .
3. An additional heat source (natural heat production is still included) in the mantle due to subduction was added. The boundary conditions and the initial state matrix were given by equations 119, 120, and 121, which were then scaled such that  $T_{scale} \in [0, 1]$ . There were two different simulations:
  - Without decay, which means that the radioactive materials had the same heat production throughout the entire simulation.
  - With decay, which means that the concentration of radioactive isotopes decreased over time (according to their half-life periods), decreasing their total heat production.

The shape of the lithosphere matrix was  $n_y \times n_x$ , where  $n_y = 101$  represents depth and  $n_x = 126$  represents width. The step lengths were  $h_x = h_y = 1.2km$  (Note that this is not the scaled step length), which yields size of the lithosphere being  $120km \times 150km$ . The sections of the lithosphere were included, and (using the values described) had the indexing:

- Zone 1 ( $0 - 20km$ ) had indices  $j \in [0, 16]$ ,
- Zone 2 ( $20 - 40km$ ) had indices  $j \in [17, 33]$
- Zone 3 ( $40 - 120km$ ) had indices  $j \in [34, 101]$

The position matrix was updated a number of times according to equations 93 and 94. The maximum number of iterations (of Jacobi's method) per time step was set to 20,000 and the tolerance was set to  $10^{-10}$ . The difference was dictated by the average difference per point between the matrices  $T^l$  and  $T_{guess}^l$ . There were typically several thousand iterations at each time step before the Jacobi method converged. The step sizes  $h_x$  and  $h_y$  and the time step  $h_t$  were all set to be  $h_x = h_y = h_t = 0.01$  for the simulation. This means that the scaling unit for distance was  $x_c = 120000$  such that  $x \in [0, 150km]$  and  $y \in [0, 120km]$ . The number of time steps were  $n_t = 100$  such that  $t \in [0, 1]$ , such that the temporal scaling unit was  $t_c = 3.15 \cdot 10^{16}$  (seconds per  $10^9$  years). The scaling unit for temperature was  $T_c = 1292^\circ C$  (an additional  $+8^\circ C$  was added in post analysis. The scaling parameters which relate to the constants  $\alpha$   $\beta$  and  $Q_c$  (as presented in equation 93) were therefore set to be  $t_c = 1Gy$ ,  $x_c = 120km$ , and  $T_c = 1292^\circ C$ .

The end state of the four different simulations were plotted in python. A slice in the middle of the matrix was used to plot temperature vs. depth. The following three analytic solutions were also plotted for comparison:

1. Solution to the situation with no heat production (linear increase as a function of depth).
2. Solution to the natural heat production (given by equations 119, 120 and 121).
3. Solution to the steady state with subduction (given by equations 122, 123 and 124).

These analytic results were compared to the corresponding numerical simulation data. Note that the last analytic solution did not include isotope decay. The numerical results including isotope decay will therefore be compared with the numerical results without decay (and in turn the analytic results without decay). The relative errors as a function of depth were plotted for the three situations:

(1): No heat production

(2): Natural heat production

(3): Natural and enriched heat production without decay

As noted previously, the last and most realistic simulation of radioactive enrichment (with isotope decay included) does not have an analytic solution. The results of these were therefore compared with the simulation of the enriched mantle without decay (constant additional heat source). Instead of comparing a single temperature state in time, the evolution of the temperature difference between the two situations was animated as a GIF. This was done by simulating both cases using the same parameters, and subtracting the matrix  $T_{decay}^l$  from the matrix  $T_{nodecay}^l$ , for all time steps  $l$ . This produced an animation which displayed the additional heat production of the less realistic (excluding isotope decay) simulation, illustrating the impact of the decay implementation in the time frame of  $1Gy$ . Finally, a plot was also produced illustrating the final time step of the difference (temperature difference after  $1Gy$ ).

## IV. RESULTS

### A. One dimension

In the four following figures, the top subplots show the solution from the three schemes (explicit forward Euler, implicit backward Euler and Crank-Nicolson) and the analytic solution. In the bottom subplots the relative error is shown. Figure 2 shows the simulation using  $h_x = 0.1$  and  $h_t = 10^{-5}$  at  $t_1 = 0.1$ .

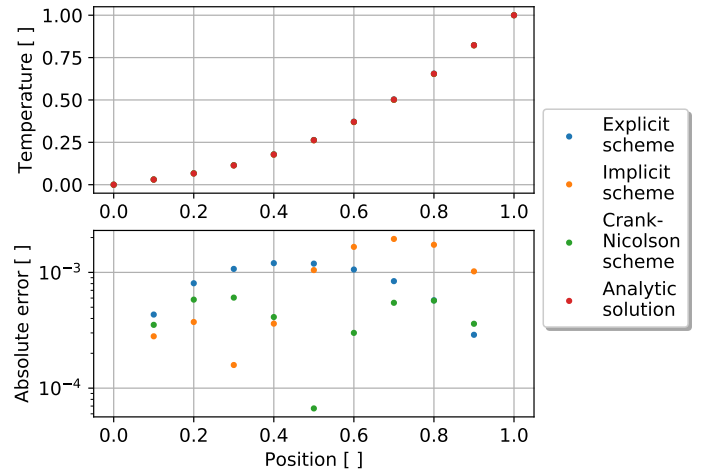


FIG. 2. Comparison of the three schemes and analytic solution at time  $t = 0.1s$ , with a spatial step  $h_x = 0.1$  and temporal step  $h_t = 10^{-3}$ . The top plot illustrates the temperatures vs. position for all schemes (graphs overlap), and the bottom plot illustrates the absolute errors vs. position for all schemes.

Figure 3 shows the simulation using  $h_x = 0.1$  and  $h_t = 10^{-5}$  at  $t_2 = 0.2$ .

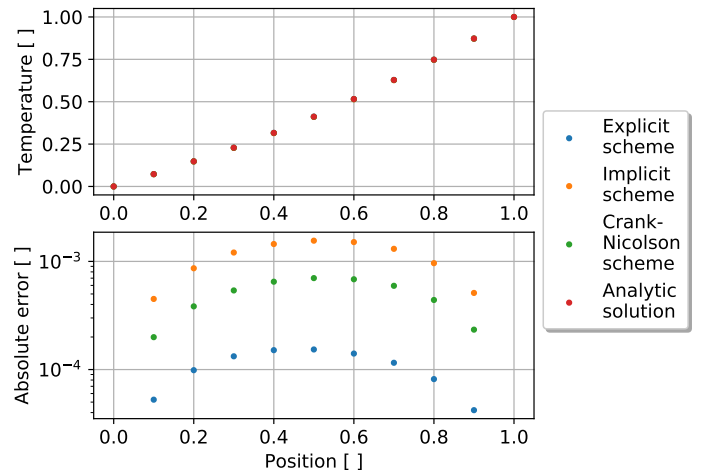


FIG. 3. Comparison of the three schemes and analytic solution at time  $t = 0.2s$ , with a step length  $h_x = 0.1$  and temporal step  $h_t = 10^{-3}$ . The top plot illustrates the temperatures vs. position for all schemes (graphs overlap), and the bottom plot illustrates the absolute errors vs. position for all schemes.

Figure 4 shows the simulation using  $h_x = 0.01$  and  $h_t = 10^{-5}$  at  $t_1 = 0.1$ .

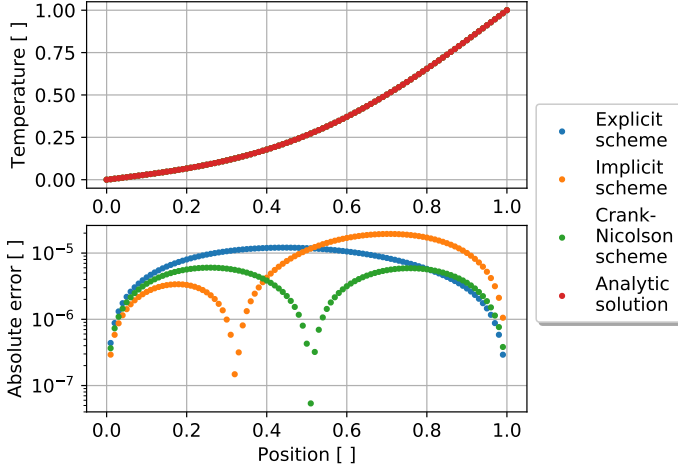


FIG. 4. Comparison of the three schemes and analytic solution at time  $t = 0.1s$ , with a step length  $h_x = 0.01$  and temporal step  $h_t = 10^{-5}$ . The top plot illustrates the temperatures vs. position for all schemes (graphs overlap), and the bottom plot illustrates the absolute errors vs. position for all schemes.

Figure 5 shows the simulation using  $h_x = 0.01$  and  $h_t = 10^{-5}$  at  $t_2 = 0.2$ .

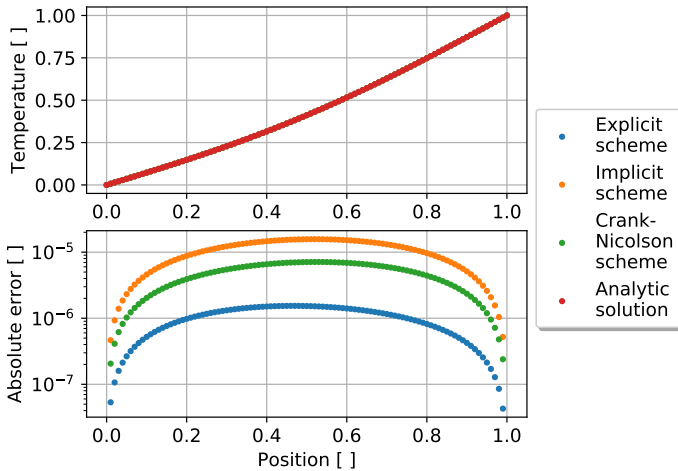


FIG. 5. Comparison of the three schemes and analytic solution at time  $t = 0.2s$ , with a step length  $h_x = 0.01$  and temporal step  $h_t = 10^{-5}$ . The top plot illustrates the temperatures vs. position for all schemes (graphs overlap), and the bottom plot illustrates the absolute errors vs. position for all schemes.

### B. Two dimension

The time evolution of the numerical solution is visualized as an animated GIF (with  $h_x = 0.01$ ,  $n_x = n_y = 101$ ,  $h_t = 0.0001$  and  $t \in [0, 0.2]$ ) at the following link <https://github.com/steinhauser/FYS3150/blob/master/Project5/twodim.gif>.

The two-dimensional numerical solution at time  $t = 0.1s$  is

illustrated in figure 6.

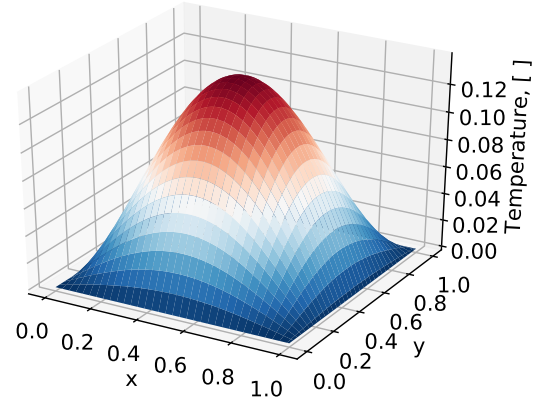


FIG. 6. The numerical solution at time  $t = 0.1s$ . The z-axis, temperature is scaled (solution is always in the domain of  $[0, 1]$ ). The simulation has step length  $h_x = 0.01$ , number of spatial points  $n_x = n_y = 101$ , time step  $h_t = 0.0001$  and number of temporal points  $n_t = 1001$ .

The two-dimensional analytic solution at time  $t = 0.1s$  is illustrated in figure 7.

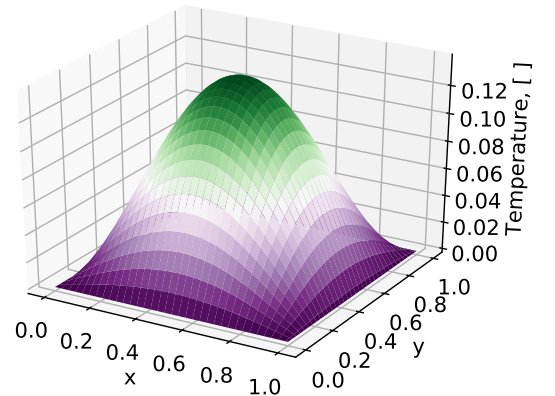


FIG. 7. The analytic solution at time  $t = 0.1s$ , given by  $u(x, y, t = 0.1) = \sin(\pi x) \sin(\pi y) e^{-2\pi^2 \cdot 0.1}$ . The z-axis is scaled such that the solution is always in the domain of  $[0, 1]$ . The step length is  $h_x = 0.01$ , number of spatial points is  $n_x = n_y = 101$ , time step is  $h_t = 0.0001$ , and number of temporal points is  $n_t = 1001$ .

In the following figures, the time evolution of the numerical error is illustrated. The error at each time step is evaluated as the average error per point (in relation to the analytic solution) of the matrix. Figure 8 keeps a fixed  $h_x = 0.01$  and shows three variants of the temporal step size  $h_t$ .

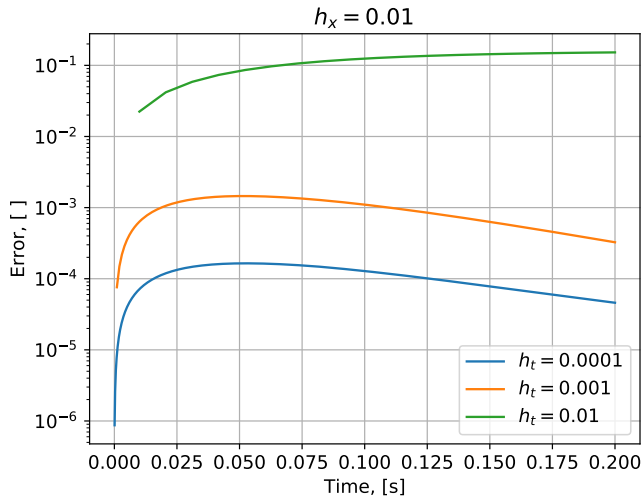


FIG. 8. The average absolute error per point vs. time. The time scale here is in the range  $[0, 0.2s]$ . The error is evaluated as the average difference between the numerical and analytic results for the entire  $n_x \times n_y$  grid. The unit is that of the temperature, which is scaled to the interval  $[0, 1]$ .

Figure 9 keeps a fixed  $h_t = 0.01$  and shows three variants of spatial step size  $h_x$ .

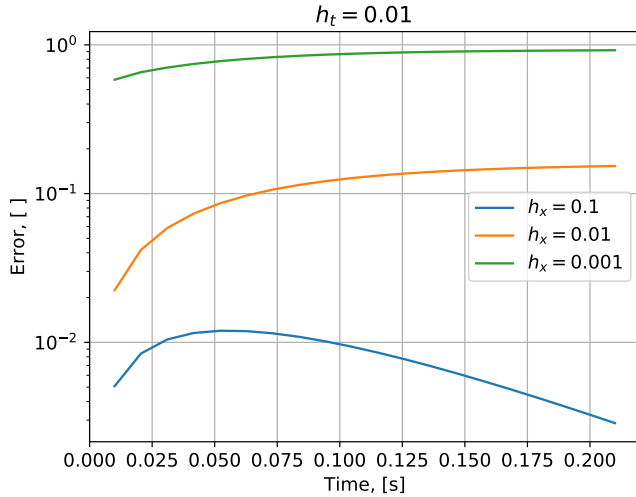


FIG. 9. The average absolute error per point vs. time. The time scale here is in the range  $[0, 0.2s]$ . The error is evaluated as the average difference between the numerical and analytic results for the entire  $n_x \times n_y$  grid. The unit is that of the temperature, which is scaled to the interval  $[0, 1]$ .

Figure 10 keeps a fixed  $h_t = 0.001$  and shows three variants of the spatial step size  $h_x$ .

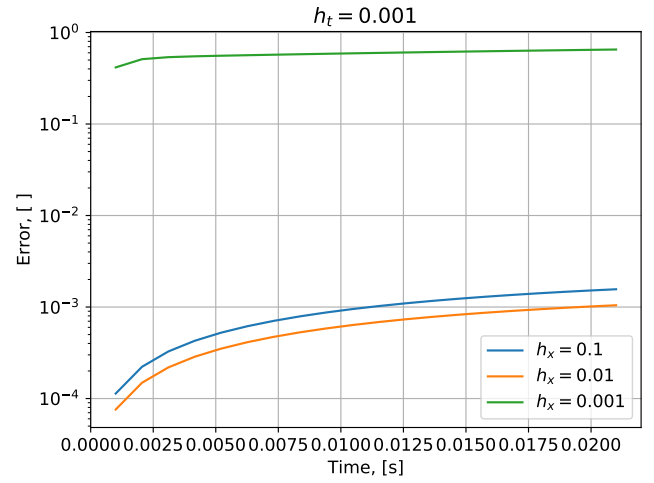


FIG. 10. The average absolute error per point vs. time. The time scale here is in the range  $[0, 0.02s]$ . The error is evaluated as the average difference between the numerical and analytic results for the entire  $n_x \times n_y$  grid. The unit is that of the temperature, which is scaled to the interval  $[0, 1]$ .

### C. Lithosphere

Figure 11 shows the temperature vs. depth for 4 different simulations over  $1Ga$  and analytic solutions for three steady state solutions.

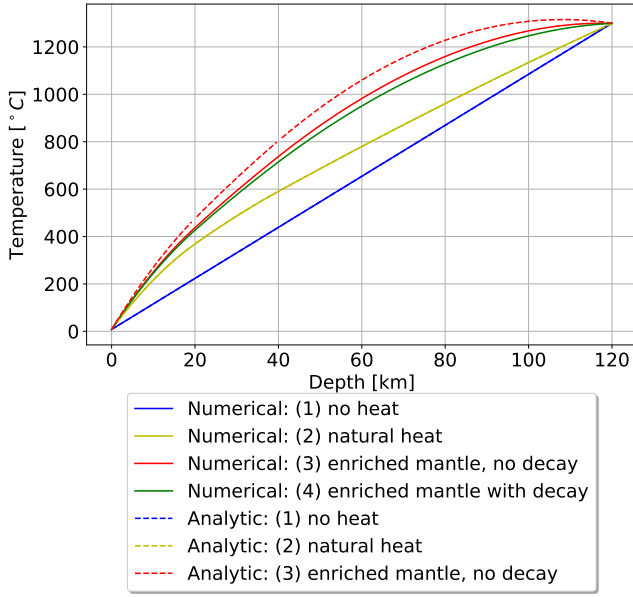


FIG. 11. Numerical solution for four simulations over  $1Ga$ . The one dimensional graphs are taken as the midpoint  $x$  value (75 km from both side boundaries) and all  $y$  values. Situation (1) shows the solution for no heat sources (numerical and analytic, almost overlap). Situation (2) shows the solution for natural heat sources separated in three layers (numerical and analytic, almost overlap). Situation (3) shows the steady state solution for natural heat sources and an additional enrichment of radioactive materials (without decay) in the mantle, 40-120 km (numerical and analytic). Lastly, the numerical solution for the enrichment situation with decay is shown.

Figure 12 shows the error between analytic and numerical solutions for the three steady state situations (1), (2) and (3) from figure 11.

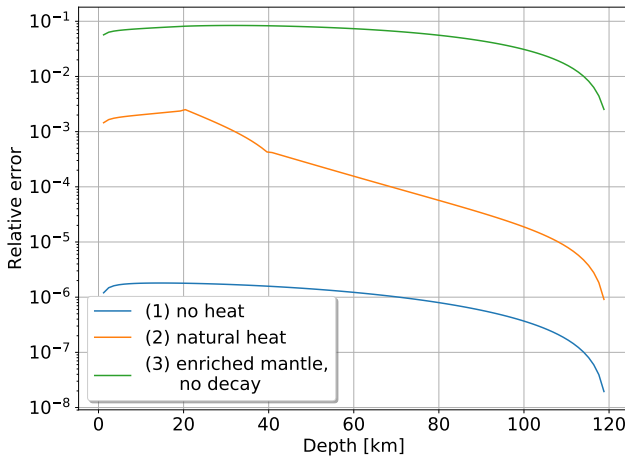


FIG. 12. The relative error vs. depth between the analytic and numerical solution for three steady state situations: (1) no heat sources, (2) natural heat sources separated in three layers and (3) natural heat sources with an additional enrichment of radioactive materials (without decay) in the mantle, 40-120 km.

The time evolution of the impact radioac-

tive decay has on the temperature is visualized as an animated GIF at the following link: [https://github.com/steinnhauser/FYS3150/blob/master/Project5/lithosphere\\_decay\\_impact.gif](https://github.com/steinnhauser/FYS3150/blob/master/Project5/lithosphere_decay_impact.gif)

Figure 13 illustrates the temperature difference caused by radioactive decay. The figure illustrates the difference at the end of the simulation  $t = 1Gy$ .

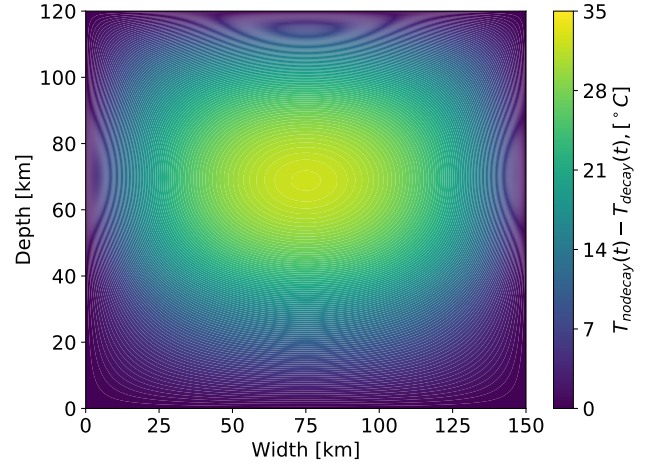


FIG. 13. An illustration of the difference between two simulations. One simulation held the additional heat production (due to subduction) constant, and the other included decay of the isotopes. Only the final time step  $t = 1Gy$  is illustrated. The largest temperature difference is  $T_{diff} = 32.6^\circ C$  and is located at  $68.4km$  depth and  $75km$  width.

## V. DISCUSSION

### A. One dimension

As stated by table I, the truncation error of the explicit and implicit schemes are proportional to  $\mathcal{O}(h_x^2)$  and  $\mathcal{O}(h_t)$ , and the Crank-Nicolson method had truncation errors of  $\mathcal{O}(h_x^2)$  and  $\mathcal{O}(h_t^2)$ . In all of the four situations studied, the time step  $h_t$  was set such that  $h_t \ll h_x$ , causing the spatial error to be the decisive factor. The numerical error should be no more than  $10^{-2}$  when  $h_x = 0.1$  and no more than  $10^{-4}$  when  $h_x = 10^{-2}$ . The error of the calculations should therefore be proportional to  $\mathcal{O}(h_x^2)$  for all schemes.

The data also illustrates this. Figures 2 and 3 illustrate the results of the simulations with  $h_x = 0.1$  at  $t_1$  and  $t_2$  respectively. In both cases the implicit scheme exhibits an error which is slightly larger than  $10^{-3}$ , which is well within the largest error dictated by the truncation factors ( $10^{-2}$ ). At  $t_1$ , the Crank-Nicolson scheme exhibits the smallest error, while at  $t_2$  the explicit scheme provides the most accurate solution.

Figures 4 and 5 illustrate the results of the simulations with  $h_x = 0.01$  at  $t_1$  and  $t_2$  respectively. The time step  $h_t = 10^{-5}$  is chosen more or less the same as for  $h_x = 0.1$  in that the numerical error should be dictated by

the spacial step (below  $h_x^2 = 10^{-4}$ ). This is the case for all simulation schemes. The implicit scheme exhibits the largest error around the position  $x = 0.7$ , resulting in an error is slightly larger than  $10^{-5}$ .

The numerical error of all schemes at time  $t_2 = 0.2$  is the largest at  $x = 0.5$ . This is illustrated by both the  $h_x = 0.1$  and the  $h_x = 0.01$  graphs. An interesting pattern is illustrated by the implicit and Crank-Nicolson scheme errors at  $t_1 = 0.1$ . Figures 2 and 4 show the results at  $t_1$  and the error seems to smallest at  $x \approx 1/2$  for the Crank-Nicolson scheme and at  $x \approx 1/3$  for the implicit scheme. The cause of this effect is unknown.

The best scheme for the one dimensional diffusion equation is either the explicit scheme or the Crank-Nicolson scheme. The implicit scheme exhibited the largest absolute error in all cases illustrated. Although the explicit scheme has a limiting stability condition which forces  $h_t$  to be very small, the error produced by the scheme is shown to be the least at equilibrium ( $t_2$ ), and most stable (the explicit scheme does not exhibit the strange humps which the others do) at  $t_1$ . The explicit scheme achieved the most accurate result at time  $t_2$ , and is also an easier code implementation and has a far shorter run-time (fewer floating point operations). However, the Crank-Nicolson scheme has both no such limiting condition (the explicit scheme is limited by the Courant-Friedrichs-Lewy condition) as well as a smaller error contribution in the non-equilibrium case  $t_1$ .

The best scheme between these two depends on the research case and what time step is desired. while the explicit scheme might be better for the smallest time steps, the Crank-Nicolson scheme is likely preferable if the objective is simulation over a long period of time. This is due to the Crank-Nicolson scheme being adaptable to more customizable temporal/spatial steps.

## B. Two dimensions

As illustrated by figures 6 and 7, the numerical solution of the two-dimensional diffusion equation is very consistent with the closed form analytic solution. Further analysis of the accuracy is done by plotting the errors for a variety of step sizes  $h_x$  and  $h_t$ . The aim is to reveal which step sizes  $h_x$  and  $h_t$  produce the the optimal implicit 2D scheme results.

Figure 8 illustrates the time evolution of the numerical error for various time steps  $h_t$  (keeping  $h_x = 0.01$  fixed). The most accurate result of all three time steps is at the first time step  $t_1 = h_t$ . This is intuitive as the initial condition  $t_0$  for the simulation was dictated by the analytic solution. Some of the time steps varied farther from the analytic solution after a single time step than others. Despite this variance from the analytic solution, all simulation results produced by the implicit 2D scheme are stable.

The errors of the simulations reach a maximum value after about  $t = 0.05s$  in several variants. This is the case for the temporal step variants ( $h_t = 0.001, h_x = 0.01$ )

and ( $h_t = 0.0001, h_x = 0.01$ ). The largest time step  $h_t = h_x = 0.01$  has an error which increases towards  $10^0$  continuously. Whether or not it equals  $10^0$  in the limit requires a longer simulation run-time to study.

The difference between the temporal steps  $h_t = 0.0001$  and  $h_t = 0.001$  illustrated by figure 8 is about a factor 10. However, the difference between  $h_t = 0.001$  and  $h_t = 0.01$  is about a factor 100 and increases to almost a factor 1000. This is a good illustration of how the error is related to the step sizes. The largest time step was shown to produce the worst error. This might indicate that  $h_t$  should ideally be smaller than  $h_x$ .

Figures 9 and 10 illustrate the numerical error for various spatial step lengths  $h_x$  at fixed time steps  $h_t = 0.01$  and  $h_t = 0.001$  respectively. Figure 9 (with  $h_t = 0.01$ ) shows a surprising result: the largest step length  $h_x = 0.1$  produces the smallest error in relation to the analytic prediction. This step length decreases after a time of about  $0.05s$ . However, the error from the other step lengths ( $h_x = 0.01$  and  $h_x = 0.001$ ) continue to increase over time. The fact that a smaller step produces larger errors is counter intuitive, and reinforces the idea that  $h_t$  should ideally be smaller than  $h_x$  as the errors behave similarly to the truncation error  $\mathcal{O}$  predictions.

Figure 10 has a time step of  $h_t = 0.001$ . The smallest step length  $h_x = h_t = 0.001$  has the worst error, which is remarkably bad. This solution is not consistent with the analytic solution, and it is unknown why (the only parameters changed from the other simulations is  $h_x$  and  $n_x$ ). The step length  $h_x = 0.01$ , which is a factor of 10 larger than  $h_t$ , achieves the smallest error. In all three cases the error increases over time, but these plots only show the time range of  $t \in [0, 0.02]$ . The time where the other errors started decreasing was at about  $t = 0.05$ , so it is likely that these would also start decreasing after a similar run-time. The overall best error is achieved by  $h_x = 0.01$  and  $h_t = 0.0001$  which is a factor of 100 smaller than  $h_x$ .

## C. Lithosphere

Figure 11 illustrates the three analytic and the four numerical simulation results (additional numerical case including isotope decay). Graphs analytic(1) and numerical(1) have little to no difference to be seen by-eye. The same can be said for analytic(2) and numerical(2). The difference between graphs numerical(3) and analytic(3) is more noticeable. All of these three situations are steady state solutions, which means that after a certain amount of simulation time the numerical results should approach the analytic solutions.

The errors between all the cases (1), (2) and (3) are illustrated in figure 12. The accuracy differences of cases (1) and (2) are more visible now. The enriched mantle (3) numerical results are the largest and the analytic(1) and numerical(1) graphs are the most similar. Given that the simulations ran long enough for the systems to reach equilibrium, the error should relate to the truncation error

in the implicit scheme. The error is also likely to be closely related to the tolerance ( $10^{-10}$ ) chosen for the Jacobi method. The maximum number of iterations was never reached, so the simulation results were certainly related to this tolerance.

Figure 12 shows that the case with no heat production (1) is by far the most precise result. This is likely due to the linearity of the results. For the natural heat production, the error seems to be discontinuous (of the derivative) at the boundaries between the layers at depths  $20\text{km}$  and  $40\text{km}$ . The largest error for this case is at depth  $20\text{km}$ , and it exhibits a consistent decrease at larger depths.

The most noticeable difference of the simulation is that of the enriched mantle without decay. As shown in figure 12, the relative error for the enriched mantle (3) is less than  $10^{-1}$  and largely greater than  $10^{-2}$ . This is a fairly significant error in comparison with the other results. This might be due to the simulation having not yet reached a steady state after  $1\text{Ga}$ , though the likely explanation is the vertical boundary conditions.

While the analytic results are derived as a one dimensional temperature vs. depth function, the simulation is a model of a two dimensional slice with neighboring material. This neighboring material (represented by the vertical boundary conditions) is not above a subducted area, and exhibits no additional heat production as a result of radioactive enrichment. Since the enrichment causes additional heat production, there will be a continuous heat flow from the subducted area to its vertical boundaries. This heat loss to the surrounding material is not taken into consideration in the analytic expression, explaining the slightly lesser total temperature of the numerical results illustrated by graphs analytic(3) and numerical(3) in figure 11.

Figure 13 illustrates the temperature difference caused by the addition of radioactive decay at the end of the simulation (after  $1\text{Ga}$ ). The full time evolution is animated (see the **github repository**). In the beginning the two cases numerical(3) and numerical(4) are similar because the heat productions from the radioactive materials are the same. With decay however, the heat production decreases over time, and ends up at only 84.4% of the original heat production after  $1\text{Ga}$ . The temperature difference

is greatest in the mantle, and decreases towards zero in the boundaries. The maximum temperature difference is  $T_{\text{nodecay}} - T_{\text{decay}} = 32.6^\circ\text{C}$  at depth  $68.4\text{km}$  (about 35.5% deep into the mantle) in the middle of the subduction zone ( $75\text{km}$  width). This is an indication that the decay of the radioactive isotopes has a noticeable impact on the system when simulated over such a long time. The most significant area of the heat difference is in the mantle, as this is where the additional enrichment factors are included. Any heat difference outside of the mantle ( $0\text{--}40\text{km}$  depth) is a result of diffusion.

## VI. CONCLUSION

In one dimension, the three schemes simulated were found to not be so different in terms of numerical errors, with the spatial step length being the dominant error factor. The explicit scheme seemed to be the most reasonable scheme in the 1D case, despite the temporal and spatial steps being limited by the Courant-Friedrichs-Lewy condition.

The two dimensional simulation using the implicit scheme and the Jacobi method was found to be consistent with the closed-form analytic solutions presented. The errors of the simulation were however shown to be very dependant on the ratio of the step sizes  $h_x$  and  $h_t$ . The results indicated that the condition  $h_t \geq h_x$  seemed to be a determining factor for minimizing the error.

The lithosphere model presented was successfully implemented numerically using the implicit scheme and the Jacobi method. The radioactive decay of the isotopes was ultimately found to have a significant impact on the temperature in the lithosphere over long periods of time.

Further improvements within this study would be to simulate with less simplification. A few suggestions are splitting up the lithosphere in more than three zones in terms of natural heat production, and having the additional heat in the mantle be a gradient dictated by the degree of subduction. Another improvement would be to vary heat capacity  $C_p$ , density  $\rho$  and thermal conductivities  $k$  as a function of depth. Finally, the simulation can also be extended to three dimensions and include a time transition from natural heat production to additional heat due to subduction.

## REFERENCES

- 
- [1] Boas, M. (2005). *Mathematical Methods in the Physical Sciences*. Wiley.
  - [2] Möller, C. and Andersson, J. (2018). Metamorphic zoning and behaviour of an underthrusting continental plate. *Journal of Metamorphic Geology*, 36(5):567–589.
  - [3] Morten, H.-J. (2015). Lecture notes fall 2015.
  - [4] Roques, L., Auger-Rozenberg, M.-A., and Roques, A. (2007). Modelling the impact of an invasive insect via reaction-diffusion. working paper or preprint.

### APPENDIX A: CRANK NICOLSON DERIVATION

The Crank-Nicolson scheme regards a combination of the explicit and implicit schemes. Following are the three Taylor expansions which lead to the explicit scheme (utilizing the notation  $\frac{\partial^n}{\partial m^n}u = u^{(nm)}$ ):

$$u(x + h_x, t) = u(x, t) + u^{(1x)}(x, t)h_x + u^{(2x)}(x, t)\frac{h_x^2}{2} + \mathcal{O}(h_x^3) \quad (102)$$

$$u(x - h_x, t) = u(x, t) - u^{(1x)}(x, t)h_x + u^{(2x)}(x, t)\frac{h_x^2}{2} + \mathcal{O}(h_x^3) \quad (103)$$

$$u^{(2x)}(x, t) = \frac{u(x + h_x, t) - 2u(x, t) + u(x - h_x, t)}{h_x^2} + \mathcal{O}(h_x^2) \quad (104)$$

The Crank-Nicolson algorithm combines the implicit and explicit methods and takes an average of the two:

$$\begin{aligned} \frac{\partial u(x, t)}{\partial t} \Big|_{t+1/2} = & \underbrace{\frac{u(x + h_x, t) - 2u(x, t) + u(x - h_x, t)}{2h_x^2}}_{\text{Explicit}} + \mathcal{O}(h_x^2) + \\ & \underbrace{\frac{u(x + h_x, t + h_t) - 2u(x, t + h_t) + u(x - h_x, t + h_t)}{2h_x^2}}_{\text{Implicit}} \end{aligned}$$

The two  $\mathcal{O}(h_x^2)$  factors were added together. This is the general conceptualization of the Crank Nicolson scheme. Rewriting this by utilizing the notation  $u(x \pm h_x, t \pm h_t) \rightarrow u_{i \pm 1, j \pm 1}$  produces:

$$\begin{aligned} \frac{\partial u(x, t)}{\partial t} \Big|_{t+1/2} = & \frac{1}{2h_x^2} \left[ u_{i+1, j} - 2u_{i, j} + u_{i-1, j} \right. \\ & \left. + \mathcal{O}(h_x^4) + u_{i+1, j+1} - 2u_{i, j+1} + u_{i-1, j+1} \right] \end{aligned} \quad (105)$$

The partial derivative on the left hand side can be approximated by the two following Taylor expansions about the parameter  $\tilde{t} = t + h_t/2$ :

$$\begin{aligned} u(x, t + h_t) = & u(x, \tilde{t}) + u^{(1t)}(x, \tilde{t})\frac{h_t}{2} \\ & + u^{(2t)}(x, \tilde{t})\frac{h_t^2}{2} + \mathcal{O}(h_t^3), \end{aligned} \quad (106)$$

and

$$\begin{aligned} u(x, t) = & u(x, \tilde{t}) - u^{(1t)}(x, \tilde{t})\frac{h_t}{2} \\ & + u^{(2t)}(x, \tilde{t})\frac{h_t^2}{2} + \mathcal{O}(h_t^3). \end{aligned} \quad (107)$$

Inserting these into the left hand side of equation 105 (and realizing that  $u^{(1t)}(x, \tilde{t}) = (u(x, t + h_t) - u(x, t))/h_t$ ) results in:

$$\begin{aligned} u^{(1t)}(x, \tilde{t}) + \mathcal{O}(h_t^2) = & \frac{1}{2h_x^2} \left[ u_{i+1, j} - 2u_{i, j} + u_{i-1, j} \right. \\ & \left. + u_{i+1, j+1} - 2u_{i, j+1} + u_{i-1, j+1} \right] + \mathcal{O}(h_x^2) \end{aligned} \quad (108)$$

This is the general Crank-Nicolson scheme, with truncation errors  $\mathcal{O}(h_t^2) \sim h_t^2$  and  $\mathcal{O}(h_x^2) \sim h_x^2$ .

### APPENDIX B: THOMAS ALGORITHM DESCRIPTION

The Thomas algorithm involves solving a system of equations through Gaussian elimination. This involves utilizing row matrix operations to solve a matrix-vector multiplication  $\hat{A}x = b$  (given that  $\hat{A}$  is a tridiagonal matrix) with regards the vector  $x$ .

The Thomas algorithm consists of two steps: Forward substitution to eliminate the lower diagonal, and then backward substitution to eliminate the upper diagonal. This transforms the matrix and vectors such that the matrix only has diagonal elements, leading to a trivial calculation of the vector  $x$ .

Assume a general tridiagonal matrix  $\hat{A}$  given by:

$$\hat{A} = \begin{bmatrix} d_1 & c_1 & 0 & \dots & 0 \\ a_2 & d_2 & c_2 & & \vdots \\ 0 & a_2 & \ddots & & 0 \\ \vdots & & & d_{n-1} & c_{n-1} \\ 0 & \dots & 0 & a_n & d_n \end{bmatrix} \quad (109)$$

involved in the matrix-vector multiplication

$$\hat{A} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (110)$$

The forward substitution step of the Thomas algorithm modifies the coefficients (denoting the modified coefficients with a tilde  $a \rightarrow \tilde{a}$ ) of the  $\hat{A}$  matrix in the following fashion:

- **i=1:**  $\tilde{c}_1 = c_1/d_1$
- **i=2,3,...,n:**  $\tilde{c}_i = c_i/(d_i - a_i\tilde{c}_{i-1})$ .

The coefficients of the vector  $b$  are also modified (using the same modified parameter indexing) by the following relations:

- **i=1:**  $\tilde{b}_1 = b_1/d_1$



- $\mathbf{i=2,3,\dots,n}$ :  $\tilde{b}_i = (b_i - a_i\tilde{b}_{i-1})/(d_i - a_i\tilde{c}_{i-1})$

The matrix-vector multiplication now has the following form:

$$\begin{bmatrix} 1 & \tilde{c}_1 & 0 & \dots & 0 \\ 0 & 1 & \tilde{c}_2 & & \vdots \\ 0 & 0 & \ddots & & 0 \\ \vdots & & & 1 & \tilde{c}_{n-1} \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_{n-1} \\ \tilde{b}_n \end{bmatrix} \quad (111)$$

After these calculations are completed, the vector  $x$  is calculated by backwards substitution:

- $\mathbf{i=n}$ :  $x_n = \tilde{b}_n$
- $\mathbf{i=n-1,n-2,\dots,1}$ :  $x_i = \tilde{b}_i - \tilde{c}_i x_{i+1}$

This is the Thomas algorithm used to solve the matrix-vector multiplication  $\hat{A}x = b$  with regards to  $x$  given a tridiagonal matrix  $\hat{A}$ .

### APPENDIX C: ANALYTIC SOLUTIONS TO THE LITHOSPHERE

The analytic lithosphere steady state ( $\frac{\partial T}{\partial t} = 0$ ) solution is given by:

$$\frac{\partial^2 T_i}{\partial z^2} = -\frac{Q_i}{k} \quad (112)$$

$$\frac{\partial T_i}{\partial z} = -2a_i z + b_i \quad (113)$$

$$T_i(z) = a_i z^2 + b_i z + c_i, \quad (114)$$

where  $a_i$  is given by  $a_i = -Q_i/2k$ ,  $b_i$  and  $c_i$  are integration constants for the respective system  $i \in [1, 2, 3]$ , and  $z$  is the depth in  $km$ . The solutions in all three cases are solved using the following boundary conditions (requiring continuous derivatives and temperatures in the transitions between the

systems):

$$\frac{\partial}{\partial z} T_1(z = -20km) = \frac{\partial}{\partial z} T_2(z = -20km), \quad (115)$$

$$\frac{\partial}{\partial z} T_2(z = -40km) = \frac{\partial}{\partial z} T_3(z = -40km) \quad (116)$$

$$T_1(z = -20km) = T_2(z = -20km) \quad (117)$$

$$T_2(z = -40km) = T_3(z = -40km) \quad (118)$$

These boundary conditions between the systems return the following coefficients for the functions  $T_1$ ,  $T_2$  and  $T_3$  (without enrichment):

$$T_1 = -0.28z^2 - 23.66z + 8 \quad (119)$$

$$T_2 = -0.07z^2 - 15.26z + 92 \quad (120)$$

$$T_3 = -0.01z^2 - 10.46z + 188 \quad (121)$$

An addition to the Lithospheric study will be to include the heat production of radioactive materials in the ground. The Lithosphere has abundant amounts of Uranium ( $U$ ), Thorium ( $Th$ ) and Potassium ( $K$ ) which contribute to the heat production in mantle only (40 – 120km depth). The total heat production contribution of these radioactive materials is  $Q_{tot,0} = 0.5\mu W/m^3$  [3]. This additional heat production in the mantle leads to different coefficients  $a, b, c$  in the analytic solutions for  $T_1$ ,  $T_2$  and  $T_3$ . When including the extra heat production of radioactive enrichment, the heat distribution functions are given by (using the same boundary conditions as previously):

$$T_1 = -0.28z^2 - 29z + 8 \quad (122)$$

$$T_2 = -0.07z^2 - 20.6z + 92 \quad (123)$$

$$T_3 = -0.11z^2 - 23.8z + 28 \quad (124)$$

These analytic expressions set the positive  $z$  direction downwards, such that  $z = 30km$  is the input which expresses a depth of 30km. These are excellent comparisons for the numerical simulations.