

**FYS4150**  
**Project 5, deadline December 10.**



Author: Robin David Kifle, Sander Wågønes Losnedahl, Sigmund Slang, Vemund Stenbekk  
Thorkildsen

## **Abstract**

## Introduction

The aim of this project is to use three different finite difference schemes to simulate temperature variations in Earth's crust and upper mantle. The three schemes are the implicit Euler backward, explicit Euler forward and the Crank-nicolson scheme. This is a combination of the Euler forward and backward schemes. In this report, there will be focus on testing the three algorithms in one dimension, before moving on to two dimensions. The first part will mostly revolve around idealized situations in one dimension, before moving on to an idealized two dimensional problem. This is followed by a three layered model of the earth's crust, where each has different parameters. All the code, runs and figures for the project is accessible [here](#).

## Method

The three different schemes are differential equations that can be rewritten as a set of linear equations. The Euler forward is explicit, the Euler backward is implicit, while the Crank–Nicolson scheme is a combination of the two preceding schemes. These systems can be rewritten as sets of linear equations.

The Euler backwards scheme is implicit, as it uses the current step  $i$ , to derive the previous step  $i - 1$ .

$$u_t \approx \frac{u(x_i, t_j) - u(x_i, t_j - \Delta t)}{\Delta t} \quad (1)$$

$$u_{xx} \approx \frac{u(x_i + \Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j)}{\Delta x^2} \quad (2)$$

It is possible to scale the above equation by  $\alpha = \Delta t / \Delta x^2$ , so the equation only depends on one scaled variable. This leads to:

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

Now the differential equation can be written as a set of linear equations with a matrix  $A$  times a vector  $V_j$  such that  $AV_j = V_{j-1}$ . Where  $A$ , defined from the above differential equations take the form:

$$A = \begin{bmatrix} 1 + 2\alpha & -\alpha & 0 & 0 & \cdots \\ -\alpha & 1 + 2\alpha & -\alpha & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & -\alpha & 1 + 2\alpha \end{bmatrix} \quad (4)$$

It is now possible to find the previous vector  $V_{j-1}$  when  $V_j$  is known. This means that it is essential to have initial conditions to start the calculations. A more generalized equation can be written as:

$$A^{-1}(AV_j) = A^{-1}(V_{j-1}) \quad (5)$$

By continuing to multiply by  $A^{-1}$ , the implicit scheme takes the form:

$$V_j = A^{-j}V_0 \quad (6)$$

A very similar process can be applied to the Euler forward method, but this scheme is explicit:

$$u_t = \frac{u(x_i, t_j + \Delta t) - u(x_i, t_j)}{\Delta t} \quad (7)$$

$$u_{xx} \approx \frac{u(x_i + \Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j)}{\Delta x^2} \quad (8)$$

$$u_{i,j-1} = \alpha u_{i-1,j} + (1 - 2\alpha)u_{i,j} + \alpha u_{i+1,j} \quad (9)$$

$$A = \begin{bmatrix} 1 - 2\alpha & \alpha & 0 & 0 & \cdots \\ \alpha & 1 - 2\alpha & \alpha & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \alpha & 1 - 2\alpha \end{bmatrix} \quad (10)$$

such that:

$$V_{j+1} = AV_j \quad (11)$$

We generalize again and get:

$$V_{j+1} = A^{j+1}V_0 \quad (12)$$

The Crank-Nicolson scheme is a combination of both the implicit Euler backward and explicit Euler forward scheme.

$$\frac{\theta}{\Delta x^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + \frac{1-\theta}{\Delta x^2}(u_{i+1,j-1} - 2u_{i,j-1} + u_{i-1,j-1}) = \frac{1}{\Delta t}(u_{i,j} - u_{i,j-1}) \quad (13)$$

Where  $\theta$  determines whether the scheme is explicit when  $\theta = 0$ , or implicit when  $\theta = 1$ . However, to get the actual Crank-Nicolson scheme, it is required to have  $\theta = 1/2$ . This is stable for all  $\Delta x$  and  $\Delta t$ . The Crank-Nicolson scheme is derived by Taylor expanding the forward Euler method  $u(x, t+\Delta t)$ ,  $u(x+\Delta x, t)$ ,  $u(x-\Delta x, t)$ ,  $u(x+\Delta x, t+\Delta t)$  and  $u(x+\Delta x, t+2\Delta t)$  for  $t+2\Delta t/2$ .

Scaling the equation with  $\alpha = \frac{\Delta t}{\Delta x^2}$  gives the following equation:

$$-\alpha u_{i-1,j} + (2 + 2\alpha)u_{i,j} - \alpha u_{i+1,j} = \alpha u_{i-1,j-1} + (2 - 2\alpha)u_{i,j-1} + \alpha u_{i+1,j-1} \quad (14)$$

which can be rewritten as

$$(2I + \alpha B)V_j = (2I - \alpha B)V_{j-1} \quad (15)$$

$$V_j = (2I + \alpha B)^{-1}(2I - \alpha B)V_{j-1} \quad (16)$$

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

$$B = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & -1 \\ 0 & \cdots & \cdots & -1 & 2 \end{bmatrix} \quad (17)$$

The analytic solution for the one dimensional at equilibrium can solved by using:

$$t \rightarrow \infty \Rightarrow u_t = 0$$

This makes Laplaces' equation  $\nabla^2 u = 0$ . In one dimension this translates to  $u_{xx} = 0$ . By integrating once:  $u_x = c_1$ , integrating twice:  $u_{xx} = c_1x + c_2$ . The boundary conditions:  $u(0) = 0$  and  $u(1) = 1$  makes:

$$\begin{aligned} u(0) &= 0 \rightarrow c_2 = 0 \\ u(1) &= 1 \rightarrow c_1 = 1 \\ \rightarrow u(x, t)_{t \rightarrow \infty} &= x \end{aligned}$$

In other words, a linear graph.

### HERSKALROBINSUGETISS

The analytic solution for the three layer model in one dimension can be solved by considering the heat equation at equilibrium:

$$\nabla(k\nabla T) + Q = \rho c_p \frac{\partial T}{\partial t} \Rightarrow \frac{\partial T}{\partial t} = 0 \quad (18)$$

By integrating two times over z, this equals:

$$a = -\frac{Q_n}{2K} \quad (19)$$

$$\frac{\partial T_n}{\partial z} = 2a_n z + b_n \quad (20)$$

$$T_n = a_n z^2 + b_n z + c_n \quad n = 1, 2, 3 \quad (21)$$

The temperature and heat flow is continuous at the boundaries, which is located at  $z=0$ ,  $z=-20$  and  $z=-40$ :

Heat flow:

$$\frac{\partial T_1(-20)}{\partial z} = \frac{\partial T_2(-20)}{\partial z}, \frac{\partial T_2(-40)}{\partial z} = \frac{\partial T_3(-40)}{\partial z} \quad (22)$$

Temperature:

$$T_1(0) = 8 \rightarrow c_1 = 8 \quad T_1(-20) = T_2(-20) \quad T_2(-40) = T_3(-40) \quad (23)$$

These conditions make it possible to express  $b_1$  and  $b_2$ , by  $b_3$ :

$$b = \begin{bmatrix} -6 + b_3 \\ 2.4 + b_3 \\ b_3 \end{bmatrix} \quad (24)$$

These relations have been used to find  $b-$  and  $c$ -values.

## Implementation

The implicit Euler scheme were first implemented due to its simplicity, not needing any tri-diagonal solvers. However, this simplicity comes on the cost of stability which can be seen in Table 1. Next, the explicit schemes were implemented, as well as the LU triangular solver which is the backbone for the explicit Euler scheme. Crank-Nicolson does not solely rely on the LU solver, but has an implicit component as well.

Only the implicit scheme were extended to an additional dimension. Again, this is for the sake of simplicity, but one has to be aware of the stability requirements.

When the code is extended an additional dimension, the boundary conditions have to be extended accordingly. A problem arises at the edges of the two-dimensional grid where no boundary conditions exists, because the implicit scheme needs four adjacent points to calculate the current point, but at the edges there are only three available points. To resolve this issue, one has to use the point at the opposite edge.

To solve the geological case, the implicit scheme has to be adjusted with the  $1300C^\circ$  at the bottom of the lithosphere and  $8C^\circ$  at the top of the crust. Additionally, there is heat production at certain ranges which were also implemented.

## Results

The truncation errors and stability is calculated in the Taylor expansion and these values are shown in the table below:

Table 1: Truncation errors and stability for the three methods

Method	Truncation error	Stability for
Euler Forward	$\Delta x^2, \Delta t$	$\frac{1}{2}\Delta x^2 \geq \Delta t$
Euler Backward	$\Delta x^2, \Delta t$	$\Delta x^2$ and $\Delta t^2$
Crank-Nicolson	$\Delta x^2, \Delta t^2$	$\Delta x^2$ and $\Delta t^2$

Table 1 shows that the Euler backward and Crank-nicolson schemes have no limitations for  $\Delta x$  and  $\Delta t$ .

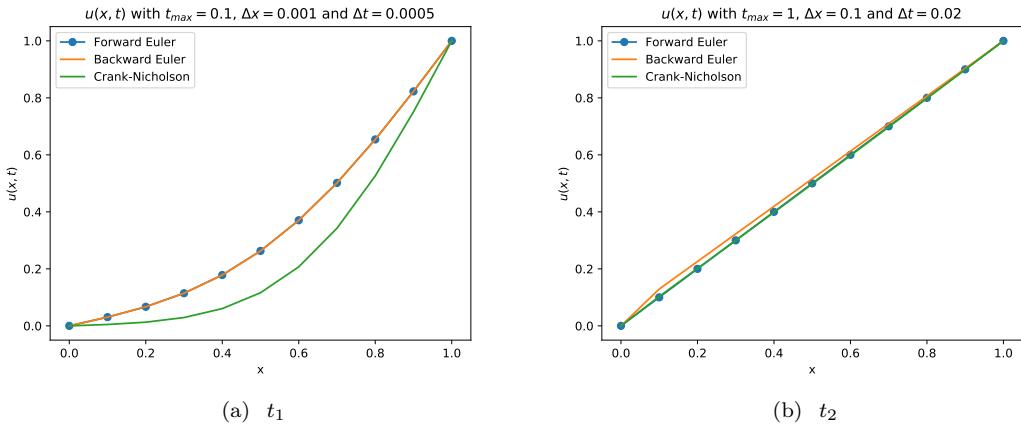


Figure 1: Comparison of  $t_1$  and  $t_2$ .

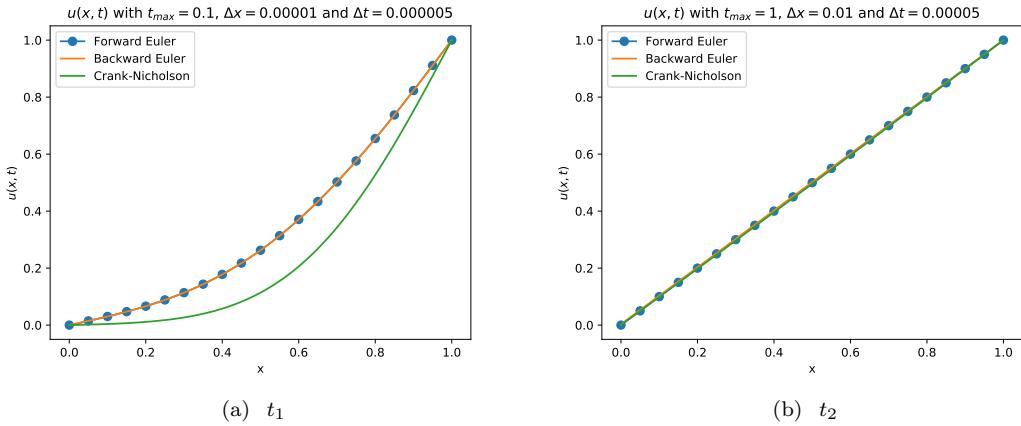


Figure 2: Comparison of  $t_1$  and  $t_2$ .

Figure ?? highlights the distribution of temperature at two different times. At  $t_1$  the different schemes produce different distributions. Forward and backward euler line up closely, in contrast to the Crank-nicolson scheme. At  $t_2$  the temperature distribution is linear and all the schemes line up closely.

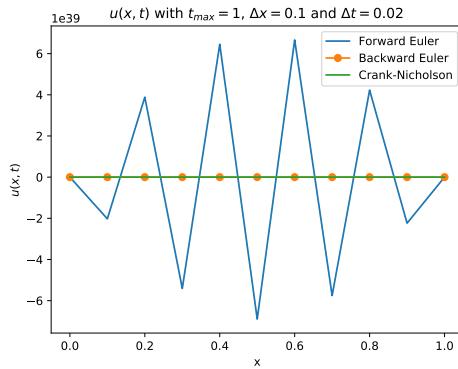


Figure 3: Plot showing the effect of choosing unfavorable  $\Delta x$  and  $\Delta t$

As can be read out of Table 1, the forward Euler scheme has a rigorous stability condition. Figure 3 shows a graphical representation of what happens when this condition is violated.

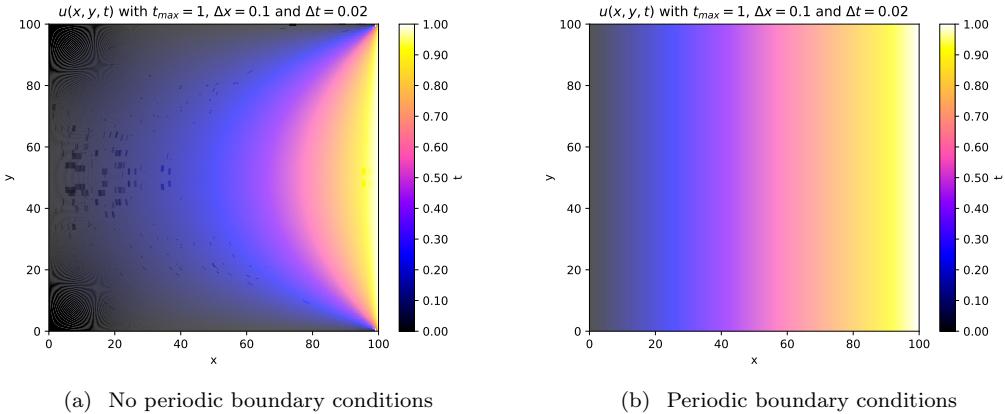


Figure 4: Numerically computed two dimensional simulation with and without periodic boundary conditions

With periodic boundary conditions, the plot is giving the same result as in the one dimensional case (Figure 2B and Figure 4B). In other words, it is showing a uniform increase in temperature with increasing  $x$ . Without periodic boundary conditions, there is heat loss at all the edges except the edge with the heat source.

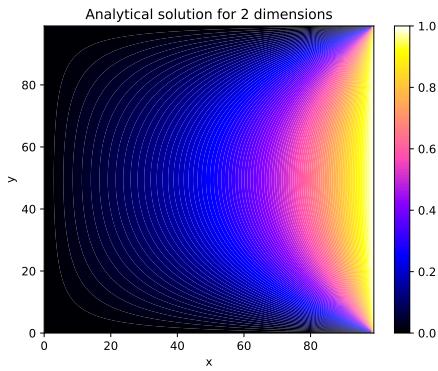


Figure 5: Plot showing the analytic for the two dimensional case

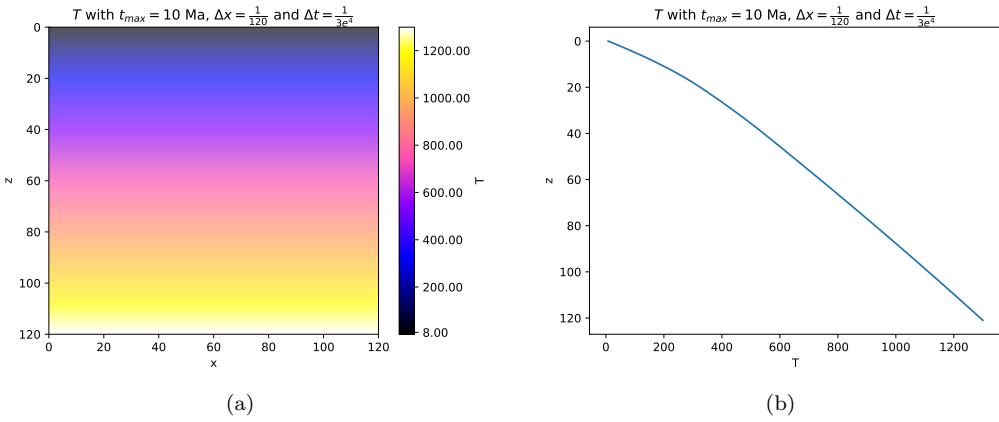


Figure 6: Numerically computed temperature distribution in the three layered lithosphere model after 10 million years, plotted in one and two dimensions without radioactive enrichment.

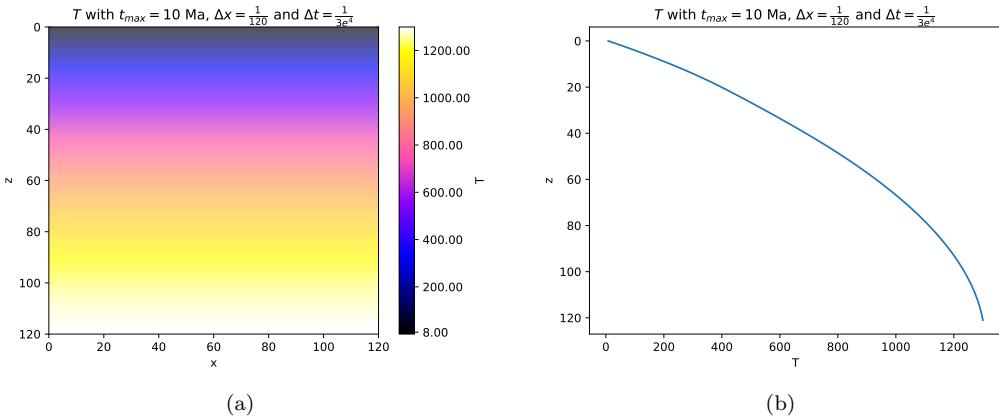


Figure 7: Numerically computed temperature distribution in the three layered lithosphere model after 10 million years, plotted in one and two dimensions with radioactive enrichment.

By using the relations described in equation *robinergay*, the  $c$ -values for the analytic expression can be evaluated to be:

$$c = \begin{bmatrix} 8 \\ 92 \\ 44 \end{bmatrix} \quad (25)$$

The bottom boundary condition is used to evaluate the  $b$ -values. As  $T_3(-120) = 1300$ ,  $c$ -values are known, and  $b_1, b_2$  are expressed with regard to  $b_3$  (equation *robinerfortsattgay*), the  $b$ -values are easy to find.

$$b = \begin{bmatrix} -29 \\ -20.6 \\ -23.8 \end{bmatrix} \quad (26)$$

These values have been used in combination with equation *robinergay2* to plot the analytic temperature distribution for the three layered lithosphere model (8).

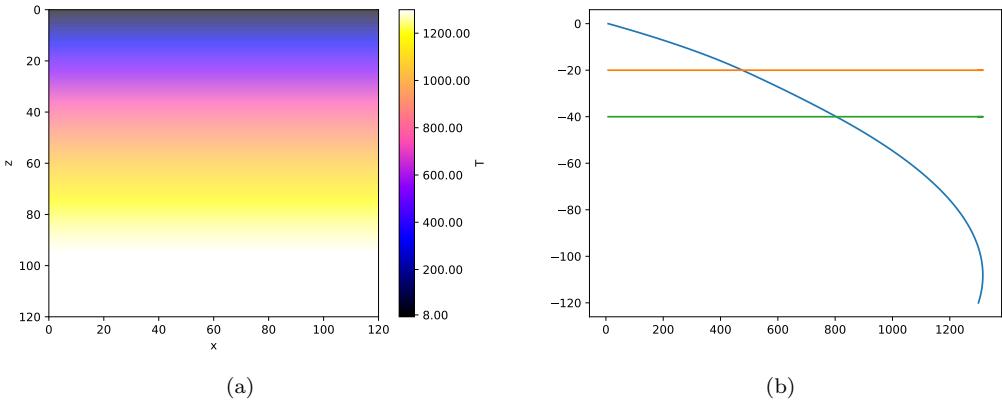


Figure 8: Analytic solution for the three layered lithosphere model after reaching equilibrium, plotted for one and two dimensions with radioactive enrichment.

Table 2: % of different radioactive materials left after 1 Gy

Radioactive material	Half-life	% Before simulation	% after 1 Gy
Potassium ( $K$ )	1,25 Gy	20 %	11,49 %
Uranium ( $U$ )	4,47 Gy	40 %	34,25 %
Thorium ( $Th$ )	14,0 Gy	40 %	38,07 %
Sum		100 %	84,44 %

Table 2 is showing that a lot of the radioactive material is present even one billion years after the enrichment. The  $Q$ -value (heat production) before and after enrichment for the mantle is respectively 0,05 and 0,55. After one billion years  $Q_{1Gy} = 0.46442$ , which is not a big reduction considering the vast time span.

## Discussion

Table 1 shows that, even though the Euler forward method is easy to implement, there are serious limitations for the stability. The Euler backward and Crank-nicolson schemes are stable for all  $\Delta x$  and  $\Delta t$ , but the implementation is more complicated. As the euler forward and Crank-nicolson uses a tridiagonal solver in the computation process, it is expected to be slower than the Euler forward scheme, which uses a more direct approach. Figure 3 shows that the forward Euler fluctuates a lot when the stability condition is violated. The stable schemes actually produce correct results, but are drowned by the fluctuations of the forward Euler.

In the one dimensional case, the system converges to an equilibrium. This is shown graphical in figure 1 and 2. When decreasing the  $\Delta x$  and  $\Delta t$ , the graphs become noticeably smoother. Moreover, the errors decrease in accordance with Table 1. It is hard to compare the three schemes in terms of quality. The explicit scheme is unstable for some values (Table 1, Figure 3), but is easy to implement and computing efficient. The implicit Euler backward scheme is less computing efficient than the explicit scheme, but does not suffer from the same stability limitations. The Crank-Nicolson scheme is stable for all  $\Delta t$  and  $\Delta x$  and has the smallest truncation error when  $\Delta t$  is small, but has its drawback in computing efficiency. On account of the above mentioned properties, it is impossible to unambiguously decide which scheme is best. Each has its advantages and drawbacks, and the most favorable scheme depends heavily on what is to be computed.

Given the boundary conditions  $u(0, t) = 0$  and  $u(1, t) = 1$  for all  $t$ , this can be translated to a one dimensional rod with a heat source at  $x = 1$  and a heat sink at  $x = 0$ . When starting the computation, the only point with a nonzero temperature is at  $x = 1$ . The heat from this point then spreads through the rod. The heat rises quicker close to the heat source, as is shown in figure 1A and 2A, but will with time reach an equilibrium (figure 1B,2B). It is important to note that the behavior of the system is entirely dependent on the boundary conditions. As an example, it would be possible to add another heat source at  $x = 0$  and the temperature would rise from both edges.

The boundary conditions in the two dimensional case depends on what kind of system being simulated. If the system in question is a two dimensional rod with a heat source at one end and heat loss at all the other edges, the temperature distribution would look like figure 4A or figure 5. If the system is for the temperature distribution in the lithosphere, it is not expected to be much heat loss in the lateral direction. This can be simulated with periodic boundary conditions, which in essence removes all lateral heat loss.

The last case looked on is the three layered lithosphere model. This was solved in two dimensions, and with a size of  $120 \times 120$  km. The focus was put on the vertical temperature distribution. The model was simulated using periodic boundary conditions and using the forward euler method. To run the code, and get a reasonable result within the time limit, all the constants had to be rescaled. Without scaling, it would have been impossible to get stable results for larger time spans.

Without any heat production, the temperature distribution would be a linearly dependent on depth. As the three layers have varying heat production, the distribution will not be linear. The larger amount of radioactive material in the crust results in a larger temperature increase in this layer. This can be read directly from the heat production, and is also mirrored in the plots representing temperature distribution before radioactive enrichment (Figure 6).

After radioactive enrichment, the heat production of the mantle is altered. This increases the temperature in the bottom layer, which in turn increase the temperature throughout the entire model (Figure 7,8). The radioactive material consists of three different materials with different half-lives (Table 2). As stated earlier, the reduction of radioactive material is small compared to the time span Table 2). On account of this, it is a good approximation to disregard this effect.

## Concluding remarks

Three algorithms have been used in this project, each having different drawbacks and advantages. The explicit forward Euler scheme is efficient and easy to implement, but has a rigorous stability condition for step size. The implicit Euler backward is stable for all step sizes, but is less efficient, as it uses a tridiagonal solver in the computation process. The Crank-Nicolson scheme is stable for all step sizes in addition to providing the smallest truncation error of the schemes. The drawback of this method is somewhat similar to the Euler backward scheme, as both use a tridiagonal solver.

## References