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 Earths 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 earths 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 $i-1$.

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

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

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:

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

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:

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

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:

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

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

$$V_j = A^{-j} V_0 \tag{6}$$

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

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

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

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

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

such that:

$$V_{j+1} = AVj \tag{11}$$

We generalize again and get:

$$V_{j+1} = A^{j+1}V_0 \tag{12}$$

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

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

Where θ 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 Δx and Δt. 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 $u(x+\Delta x, t+\Delta t)$ for $t+\Delta t/2$.

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

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

which can be rewritten as

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

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

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

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

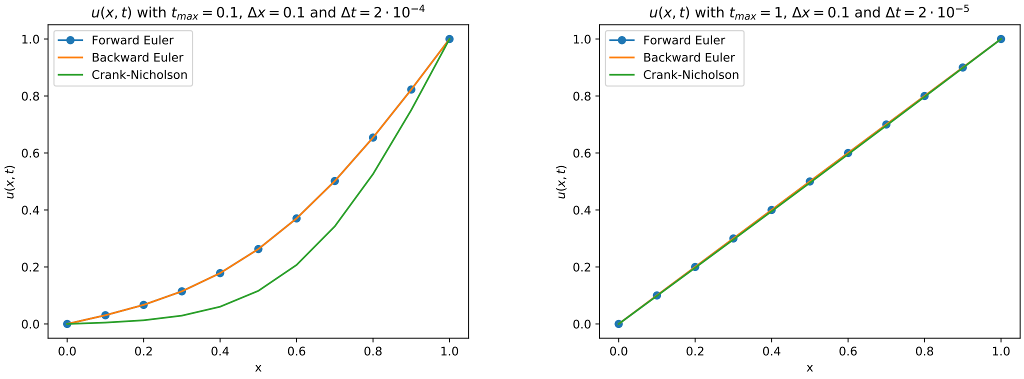
Results

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

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

Table 1: Truncation errors and stability for the three methods

Figure 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 Δx^2 and Δt^2, 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



(a) t_1

(b) t_2

Figure 1: Comparison of t_1 and t_2.

Figure 1 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.

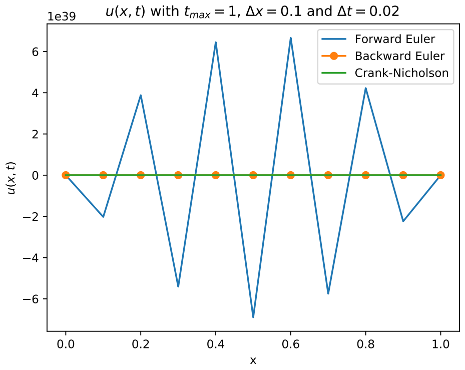


Figure 2: Plot showing the effect of choosing unfavorable Δx and Δt

As can be read out of Table 1, the forward Euler scheme has a rigorous stability condition. Figure 2 shows a graphical representation of what happens when this condition is violated.

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 Δx^2 and Δt^2, 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 2 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 1A and 1B. 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, but will with time reach an equilibrium (figure 1B). 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 fast in the edges and slower in the middle.
The boundary conditions in the two dimensional case is depending 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 $enfigurejegikkehar$. If the system is the temperature distribution in the lithosphere, it is not expected to be much heat loss in the lateral direction.

Concluding remarks

References