UiO : Department of Physics
University of Oslo

Project 5: The diffusion equation

FYS3150 - Computational physics

Kristine Baluka Hein
Anders Johansson

Contents

Abstract

Hei.

1 Introduction

2 Mathematical theory

We will now take a look into how we can numerically approximate the given problem. As there exist many different approaches, we have chosen to look at three different methods, namely Forward Euler, Backward Euler and Crank-Nicolson.

2.1 Discretisation

For the diffusion equation to be numerically solvable, it must be discretised. This is done in the standard way: Discretise x as $x_0, x_1, \ldots, x_n$ with $h = (x_n - x_0)/n$, and t as $t_0, t_1, \ldots, t_m$ with $\Delta t = (t_m - t_0)/m$. The function itself is also discretised, with the notation $u_{i,j} = u(x_i, t_j)$.

2.2 Forward Euler

2.2.1 Derivation and error analysis

The Forward Euler scheme is an explicit scheme based on Taylor polynomials. To find an approximation of the time derivative of u at point (x_i, t_j), a first order Taylor polynomial around x_i, t_j is used to calculate $u(x_i, t_{j+1})$:

$$u_{i,j+1} = u_{i,j} + \Delta t \frac{\partial u_{i,j}}{\partial t} + \frac{1}{2}(\Delta t)^2 \frac{\partial^2 u(x_i, \tilde{t})}{\partial t^2} \implies \frac{\partial u_{i,j}}{\partial t} = \frac{u_{i,j+1} - u_{i,j}}{\Delta t} + \frac{1}{2}\Delta t \frac{\partial^2 u(x_i, \tilde{t})}{\partial t^2} \tag{1}$$

where $\tilde{t} \in (t_j, t_{j+1})$ and the last term is the truncation error, which is proportional to Δt.

Similarly, the three point approximation to the second derivative (derived in [1]) with its error is used to approximate the second derivative of u at point (x_i, t_j) with respect to position:

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i+1,j} + u_{i-1,j} - 2u_{i,j}}{h^2} - \frac{1}{12} h^2 \frac{\partial^4 u(\tilde{x}, t_j)}{\partial x^4} \tag{2}$$

where the last term is the truncation error, which for this approximation is proportional to h^2. The observation $\tilde{x} \in (x_{i-1,j} + x_{i+1,j})$ is shown in the referenced report.

Inserting these two expressions into the diffusion equation gives

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} + \frac{1}{2}\Delta t \frac{\partial^2 u(x_i, \tilde{t})}{\partial t^2} = \frac{u_{i+1,j} + u_{i-1,j} - 2u_{i,j}}{h^2} - \frac{1}{12} h^2 \frac{\partial^4 u(\tilde{x}, t_j)}{\partial x^4}$$

The goal is to find the value of u at the next time step, i.e. $u_{i,j+1}$. Multiplying by Δt on both sides of the equation and moving one term to the right, we get

$$u_{i,j+1} = u_{i,j} + \frac{\Delta t}{h^2}\left(u_{i+1,j} + u_{i-1,j} - 2u_{i,j}\right) + \frac{1}{2}(\Delta t)^2 \frac{\partial^2 u(x_i, \tilde{t})}{\partial t^2} - \Delta t \cdot \frac{1}{12} h^2 \frac{\partial^4 u(\tilde{x}, t_j)}{\partial x^4}$$

The quantity $\Delta t/h^2$ can be defined as α, which, with a slight reorganisation yields the final expression

$$u_{i,j+1} = (1-2\alpha)u_{i,j} + \alpha\big(u_{i+1,j} + u_{i-1,j}\big) + \frac{1}{2}(\Delta t)^2\frac{\partial^2 u(x_i,\tilde{t})}{\partial t^2} - \Delta t\cdot\frac{1}{12}h^2\frac{\partial^4 u(\tilde{x},t_j)}{\partial x^4}$$

The two error terms are proportional to $(\Delta t)^2$ and $\Delta t\cdot h^2$. As per usual, the global error is one order lower, as the error is accumulated. This gives one error term proportional to Δt and one propotional to h^2. The order of h is not reduced, as this error is not accumulated.

2.2.2 Stability analysis

2.3 Backward Euler

The idea of backward Euler is similar to the forward Euler. Actually, the only difference lies in how we approximate the time derivative of u.

2.3.1 Derivation and error analysis

We can also use Taylor polynomials of $u(x_i,t_{j-1})$to write an approximation of the time derivative of u around x_i,t_j:

$$u_{i,j-1} = u_{i,j} - \Delta t\frac{\partial u_{i,j}}{\partial t} + \frac{1}{2}(\Delta t)^2\frac{\partial^2 u(x_i,\tilde{t})}{\partial t^2} \implies \frac{\partial u_{i,j}}{\partial t} = \frac{u_{i,j}-u_{i,j-1}}{\Delta t} + \frac{1}{2}\Delta t\frac{\partial^2 u(x_i,\tilde{t})}{\partial t^2}$$

where $\tilde{t}$ and the second term are the same as defined in section 2.2.1 on the preceding page. Equation (2) on the previous page can be reused as an approximation to the second derivative.

With these approximations, the diffusion equation can be written as

$$\frac{u_{i,j}-u_{i,j-1}}{\Delta t} + \frac{1}{2}\Delta t\frac{\partial^2 u(x_j,\tilde{t})}{\partial t^2} = \frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2} + \frac{1}{12}h^2\frac{\partial^4 u(\tilde{x},t_j)}{\partial x^4}$$

$$u_{i,j}-u_{i,j-1} = \alpha(u_{i+1,j}-2u_{i,j}+u_{i-1,j}) + \Delta t\cdot\frac{1}{12}h^2\frac{\partial^4 u(\tilde{x},t_j)}{\partial x^4} - \frac{1}{2}(\Delta t)^2\frac{\partial^2 u(x_j,\tilde{t})}{\partial t^2}$$

$$-\alpha u_{i-1,j} + (1+2\alpha)u_{i,j} - \alpha u_{i+1,j} = u_{i,j-1} + \Delta t\cdot\frac{1}{12}h^2\frac{\partial^4 u(\tilde{x},t_j)}{\partial x^4} - \frac{1}{2}(\Delta t)^2\frac{\partial^2 u(x_j,\tilde{t})}{\partial t^2}$$

In this equation, only the u on the right hand side is known, while the u's on the left side are the ones we are interested in. As it is not possible to find an explicit expression for the quantities of interest, this leads to an implicit scheme. Observe that the error is the same as for the Forward Euler scheme.

To solve the equation above, note that it must hold for all $i\in[1,n]\cap\mathbb{N}$, giving the following set

of equations:

$$\begin{array}{ccccccc}
-\alpha\overbrace{u_{0,j}}^{0} & + & (1+2\alpha)u_{1,j} & - & \alpha u_{2,j} & = & u_{1,j-1} \\
-\alpha u_{1,j} & + & (1+2\alpha)u_{2,j} & - & \alpha u_{3,j} & = & u_{2,j-1} \\
-\alpha u_{2,j} & + & (1+2\alpha)u_{3,j} & - & \alpha u_{4,j} & = & u_{3,j-1} \\
\vdots & & \vdots & & \vdots & & \vdots \\
-\alpha u_{n-3,j} & + & (1+2\alpha)u_{n-2,j} & - & \alpha u_{n-1,j} & = & u_{n-2,j-1} \\
-\alpha u_{n-2,j} & + & (1+2\alpha)u_{n-1,j} & - & \alpha\underbrace{u_{n,j}}_{1} & = & u_{n-1,j-1}
\end{array}$$

This can then be written on a matrix form:

$$\begin{bmatrix}
1+2\alpha & -\alpha & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
-\alpha & 1+2\alpha & -\alpha & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\
0 & -\alpha & 1+2\alpha & -\alpha & 0 & \dots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \dots & -\alpha & 1+2\alpha & -\alpha & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \dots & -\alpha & 1+2\alpha & -\alpha \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & -\alpha & 1+2\alpha
\end{bmatrix}
\begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \\ \vdots \\ u_{n-3,j} \\ u_{n-2,j} \\ u_{n-1,j} \end{bmatrix}
=
\begin{bmatrix} u_{1,j-1} \\ u_{2,j-1} \\ u_{3,j-1} \\ \vdots \\ u_{n-3,j-1} \\ u_{n-2,j-1} \\ u_{n-1,j-1}+\alpha \end{bmatrix}$$

This is a simple, linear system with a tridiagonal matrix, for which an efficient solving algorithm was developed and implemented in project 1[1].

2.3.2 Stability analysis

2.4 Crank-Nicolson

2.4.1 Derivation and error analysis

2.4.2 Stability analysis

References

[1] Anders Johansson. “Project 1”. In: *FYS3150, Computational Physics* (Sept. 2016), pp. 4–6. URL: https://github.com/anjohan/Offentlig/blob/master/FYS3150/Oblig1/Johansson_Anders_FYS3150_Oblig1.pdf.