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

Partial differential equations is a subject of many fields. These equations can describe the behavior of many natural phenomenas such as waves of sound or sea, the flow of a liquid, growth of a population or diffusion of different processes.
These equations are often solved numerically since an analytical solution is difficult, or perhaps impossible to find. An example is the Navier-Stokes equation to model the flow of a viscous flow when not certain assumptions to simply the equations has been made.

skrive noe kult om diffusjon?

In this report we will focus on how we can numerically solve the diffusion equation, which is given by

$$\begin{aligned}
\frac{\partial^2 u(x,t)}{\partial x^2} &= \frac{\partial u(x,t)}{\partial t} & t > 0,\ x \in [0,L] \\
u(x,0) &= 0 & 0 < x < L \\
u(0,t) &= 0 & t > 0 \\
u(L,t) &= 1 & t > 0
\end{aligned} \tag{1}$$

noen forklaringer paa boundaries

2 Mathematical theory

Before we can begin our analysis on how it is possible to numerically solve equation (1), we will first take a look at the analytical solution. When having an analytical solution at hand, we can use it further to find which conditions the numerical methods must hold to give us stable results.

2.1 Analytical solution of the diffusion equation

Consider the diffusion equation with the given boundaries:

$$\begin{aligned}
\frac{\partial^2 u(x,t)}{\partial x^2} &= \frac{\partial u(x,t)}{\partial t} & t > 0,\ x \in [0,L] \\
u(x,0) &= g(x) & 0 < x < L \\
u(0,t) &= a & a \in \mathbb{R}, t > 0 \\
u(L,t) &= b & b \in \mathbb{R}, t > 0
\end{aligned} \tag{2}$$

First, we notice the boundaries at $x = 0$ and $x = L$. As we will see, having these boundaries equal to 0 would make the problem much easier to solve. Therefore, we introduce a new function

$$v(x,t) = u(x,t) + w(x,t)$$

where we want the following to hold:

$$\begin{aligned} \frac{\partial^2 v(x,t)}{\partial x^2} &= \frac{\partial v(x,t)}{\partial t} & t>0,\ x\in[0,L] \\ v(x,0) &= f(x) & 0<x<L \\ v(0,t) &= 0 & t>0 \\ v(L,t) &= 0 & t>0 \end{aligned} \tag{3}$$

By introducing this new function $v(x,t)$, we have also introduced a new problem; what does $w(x,t)$ actually look like? We note that

$$\begin{aligned} \frac{\partial^2 v(x,t)}{\partial x^2} &= \frac{\partial v(x,t)}{\partial t} \\ \frac{\partial^2 u(x,t)}{\partial x^2} + \frac{\partial^2 w(x,t)}{\partial x^2} &= \frac{\partial u(x,t)}{\partial t} + \frac{\partial w(x,t)}{\partial t} \end{aligned}$$

We know from equation (2) that

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

This means that if we set

$$\frac{\partial^2 w(x,t)}{\partial x^2} = 0 \tag{4}$$

and

$$\frac{\partial w(x,t)}{\partial t} = 0 \tag{5}$$

the first condition for our "dummy" function in equation (3) will hold. We can now find a general solution for w. From equation (4), we have that

$$\begin{aligned} &\frac{\partial^2 w(x,t)}{\partial x^2} = 0 \\ \implies &\frac{\partial w(x,t)}{\partial x} = A \quad A\in\mathbb{R} \\ \implies &w(x,t) = Ax + B + h_1(t) \quad B\in\mathbb{R} \end{aligned} \tag{6}$$

From equation (5), we get that

$$\begin{aligned} &\frac{\partial w(x,t)}{\partial t} = 0 \\ &w(x,t) = C + h_2(x), \quad C\in\mathbb{R} \end{aligned} \tag{7}$$

From equation (6) and equation (7), we see that

$$w(x,t) = w(x) = Ax + D, \quad D = B + C$$

because there is no dependence of t in equation (7). We have now found how w looks like in general.
However, we need to decide the values of A and D. This can be done by looking at the boundaries for equation (3) at $x=0$ and $x=L$:

$$v(0,t) = 0$$

$$
\begin{aligned}
0 &= u(0,t) + w(x=0) \\
0 &= a + D
\end{aligned}
$$

which implies $D = -a$. Same goes for the boundary at $x = L$, which gives

$$
\begin{aligned}
v(L,t) &= 0 \\
0 &= u(L,t) + w(x=L) \\
0 &= b + AL - a \\
\frac{a-b}{L} &= A
\end{aligned}
$$

In conclusion, we have

$$
w(x) = \frac{a-b}{L}x - a
$$

As the expression of w has been found, we have also found $f(x)$ in equation (3) on the preceding page:

$$
\begin{aligned}
v(x,0) &= u(x,0) + w(x) \\
&= g(x) + w(x) \\
&= f(x)
\end{aligned}
$$

Having set up the problem properly, we are now ready to solve for $v(x,t)$. To do so, we make an ansatz that

$$
v(x,t) = F(x)G(t)
$$

which gives

$$
\begin{aligned}
\frac{\partial^2 v(x,t)}{\partial x^2} &= \frac{\partial v(x,t)}{\partial t} \\
F''(x)G(t) &= F(x)G'(t) \\
\frac{F''(x)}{F(x)} &= \frac{G'(t)}{G(t)} \qquad (8)
\end{aligned}
$$

Since the ratios of functions of different variables equals each other, they must be equal to a constant, thus giving

$$
\frac{F''(x)}{F(x)} = \frac{G'(t)}{G(t)} = -\lambda
$$

Solving for F we get

$$
F'' = -\lambda F
$$

with a general solution

$$
F(x) = c_1 \cos\left(x\sqrt{\lambda}\right) + c_2 \sin\left(x\sqrt{\lambda}\right)
$$

To find the constants c_1, c_2, we will use the boundaries for $v(x,t)$ at $x = 0$ and $x = L$. It is at this point we will see that it is advantageous for us to have the boundaries equal to zero.
Since

$$
v(x,t) = F(x)G(t)
$$

We have

$$
0 = F(0)G(t) \implies F(0) = 0
$$

and

$$0 = F(L)G(t) \implies F(L) = 0$$

At $x = 0$, the general solution becomes

$$0 = c_1$$

and at $x = L$:

$$0 = c_2 \sin\left(L\sqrt{\lambda}\right)$$

Of course, we could set $c_2 = 0$ too, but this will only give us a trivial solution which is of no interest. Therefore, we must have

$$0 = \sin\left(L\sqrt{\lambda}\right)$$

Since sine is zero for every multiple of π, we have that

$$L\sqrt{\lambda} = k\pi \quad k \in \mathbb{N}$$

$$\lambda = \left(\frac{k\pi}{L}\right)^2$$

We notice that we have infinite solutions since $k \in \mathbb{N}$. Therefore, we might have a different scaling of sine for every k, meaning $c_2 = c_k$.
For every k we have the particular solution

$$F_k(x) = c_k \sin\left(x\frac{k\pi}{L}\right)$$

Similar calculations goes for finding $G(t)$, which also will vary with k:
From equation (8) on the previous page we have

$$G_k'(t) = -\lambda_k G_k(t)$$

this gives

$$G_k(t) = e^{-\left(\frac{k\pi}{L}\right)^2 t}$$

and so we have found the kth particular solution of $G(t)$.
Therefore, we have that

$$v_k(x,t) = c_k e^{-\left(\frac{k\pi}{L}\right)^2 t} \sin\left(x\frac{k\pi}{L}\right)$$

So, what remains now is to find c_k. We have not yet taken into account the boundary of $v(x,t)$ when $t = 0$. At $t = 0$, we have that

$$f(x) = c_k \sin\left(x\frac{k\pi}{L}\right)$$

Multiplying the equation above bysin$\left(x\frac{k\pi}{L}\right)$

2.2 Discretization

For the diffusion equation to be numerically solvable, it must be discretized. This is done in the standard way: Discretize 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 discretized, with the notation $u_{i,j} = u(x_i, t_j)$.

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.

2.3 Forward Euler

2.3.1 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 $u(x_i, t_{j+1})$:

$$u_{i,j+1} = u_{i,j} + \Delta t \frac{\partial u_{i,j}}{\partial t} + \frac{1}{2}(\Delta t)^2 \frac{\partial^2 u(x_i, \tilde{t})}{\partial t^2} \implies \frac{\partial u_{i,j}}{\partial t} = \frac{u_{i,j+1} - u_{i,j}}{\Delta t} + \frac{1}{2}\Delta t \frac{\partial^2 u(x_i, \tilde{t})}{\partial t^2} \tag{9}$$

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 [**oblig1**]) with its error is used to approximate the second derivative of u at point (x_i, t_j) with respect to position:

$$\frac{\partial^2 u_{i,j}}{\partial x^2} = \frac{u_{i+1,j} + u_{i-1,j} - 2u_{i,j}}{h^2} - \frac{1}{12} h^2 \frac{\partial^4 u(\tilde{x}, t_j)}{\partial x^4} \tag{10}$$

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

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

The 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.

2.3.2 Stability analysis

Sette inn plot av stabil vs ikke stabil og spoerre oss selv hvorfor dette skjer?

As we now have set up the algorithm using Forward Euler to approximate the derivatives, we might ask ourselves whether this method gives us stable results. We

2.4 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.

2.4.1 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 2.3.1 on the preceding page. Equation (10) 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[**oblig1**].

2.4.2 Stability analysis

2.5 Crank-Nicolson

2.5.1 Derivation and error analysis

2.5.2 Stability analysis