THE DIFFUSION EQUATION IN ONE AND TWO DIMENSIONS

SHAFAQ NAZ SHEIKH
HTTPS://GITHUB.UIO.NO/SHAFAQNS/FYS4150/TREE/MASTER/PROJECT5

Abstract. In this article we want to study numerical schemes to solve the diffusion equation in one and two dimensions. We are going to implement the explicit scheme, the implicit scheme, and the Crank Nicholson scheme, to solve the diffusion equation in one dimension. Then we will implement the explicit scheme to solve the diffusion equation in two dimensions. We will look at the properties of the various numerical methods. In addition we will compare our numerical solution with the analytical solution, this means we can confirm our numerical results with the analytical results. We find that the Crank Nicolson scheme is best for the 1D equation, as it gives the lowest error and is stable for all values of Δt and Δx. The explicit scheme is best for the 2D equation, as long as we conform to the stability requirement.

Date: 16th December 2020.

1. Introduction

In the branch of natural science, we often encounter problems with many variables constrained by boundary conditions and initial values. Many of these problems can be modelled as partial differential equations(PDE). PDE's play an important role in the modelling of physical processes, from the diffusion of heat, to our understanding of tsunamis. The diffusion equation is a parabolic PDE. In typical applications, it describes the evolution in time of the density of a quantity such as, the particle density, energy density, temperature gradient, or chemical concentrations. In physics, it describes the macroscopic behaviour of particles in Brownian motion, resulting from the random movements and collisions of particles. In mathematics, it is related to Markov processes, such as random walks, and has application in many other fields, such as materials science, information theory, and biophysics.

In this article, we start by presenting the diffusion equation in one and two dimensions, in this case we are able to find an analytical solution to our problem, so we will also present that. After that we present the numerical methods we will implement to solve the equation. For the one dimensional case, we will be using the explicit forward Euler scheme, the implicit backward Euler scheme, and the implicit Crank-Nicoloson scheme. For the two dimensional case, we will be using an explicit scheme. We will also look at the stability properties of these methods. Finally we present our results and discuss our findings before giving a conclusion.

2. Theory

2.1. **The diffusion equation in one dimension.**

For the one dimensional problem, we want to solve the diffusion equation,

$$\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{\partial u(x,t)}{\partial t} \qquad t > 0 \qquad x \in [0, L] \tag{1}$$

or

$$u_{xx} = u_t,$$

with the initial condition

$$u(x,0) = f(x) = 0 \qquad 0 < x < L \tag{2}$$

and the boundary conditions,

$$u(0,t) = T_1 = 0 \qquad t \geq 0 \tag{3}$$

$$u(L,t) = T_2 = 1 \qquad t \geq 0 \tag{4}$$

Here $u(x,t)$ is the unknown function to be solved for, x is a coordinate in space, and t is time. We wish to study this problem with $L = 1$.

2.1.1. *The analytical solution.* Since we will be implementing numerical methods to solve this PDE, it can be useful to have an analytical solution to compare our simulated results with. This way we will be able to figure out if our numerical solution is correct or not. Generally we are not able to find analytical solutions, but in this case, an analytical solution can be found. We can solve this problem by considering the fact that after a long time, that is, as $t \to \infty$, a steady temperature distribution $v(x)$ will be reached. $v(x)$ is independent of the time t, and the initial conditions, $v(x)$ is then given by,

$$v(x) = (T_2 - T_1)\frac{x}{L} + T_1 = (1-0)\frac{x}{L} = \frac{x}{L} \tag{5}$$

We want to express $u(x,t)$ as the sum of the steady state temperature distribution $v(x)$ and another transient temperature distribution $w(x,t)$, thus we get,

$$u(x,t) = v(x) + w(x,t) \tag{6}$$

The problem will be solved if we can determine $w(x,t)$. The boundary value problem for $w(x,t)$ is then found to be,

$$(v+w)_{xx} = (v+w)_t \Rightarrow w_{xx} = w_t$$

This is due to the fact that $v_{xx} = 0$ and $v_t = 0$. Inserting the boundary and initial conditions,

$$w(0,t) = u(0,t) - v(0) = T_1 - T_1 = 0 - 0 = 0 \tag{7}$$
$$w(L,t) = u(L,t) - v(L) = T_2 - T_2 = 1 - 1 = 0 \tag{8}$$
$$w(x,0) = u(x,0) - v(x) = f(x) - v(x) = 0 - \frac{x}{L} = -\frac{x}{L} \tag{9}$$

Thus we now wish to find the solution to this problem, that is, we want to find $w(x,t)$. This problem can be solved by using the separation of variables technique. We make the ansatz that,

$$w(x,t) = X(x)T(t)$$

By inserting it into the equation $w_{xx} = w_t$, we get $w_{xx} = X^{''}T$ and $w_t = XT'$, which gives us,

$$X^{''}T = XT^{'}$$

Dividing both sides by XT, we obtain

$$\frac{X^{''}}{X} = \frac{T}{T'} = -\lambda^2$$

The left hand side only depends on X and the right hand side only depends on T, hence both equations must be equal to a common constant, we call this constant $-\lambda^2$. This gives us the two ordinary differential equations,

$$X^{''} + \lambda^2 = 0$$
$$T^{'} + \lambda^2 T = 0$$

with general solutions,

$$X(x) = Asin(\lambda x) + Bcos(\lambda x)$$
$$T(t) = Ce^{-\lambda^2 t}$$

To satisfy the boundary conditions $X(0) = X(L) = 0$, we must have that $B = 0$, and $\lambda = k\pi/L$. One solution is thus,

$$w(x,t) = A_n\, sin(k\pi x/L)\, e^{-(k\pi/L)^2 t}$$

Since the diffusion equation is linear we know that a superposition of solutions will also be a solution of the equation, thus we write,

$$w(x,t) = \sum_{k=1}^{\infty} A_k \, sin(k\pi x/L) \, e^{-(k\pi/L)^2 t} \tag{10}$$

The coefficient A_k, is in turn determined from the initial condition $w(x,0) = -x/L$, A_k are the Fourier coefficients for the function $-x/L$.

$$A_k = \frac{2}{L} \int_0^L \frac{-x}{L} sin(k\pi x/L) dx \tag{11}$$

In this case we have $L = 1$, this gives the value for A_k to be,

$$A_k = 2 \int_0^1 -x \, sin(k\pi x) \, dx = \frac{2}{k\pi}(-1)^k$$

Thus $w(x,t)$ with the value of $L = 1$ is given by,

$$w(x,t) = \frac{2}{\pi} \sum_{k=1}^{\infty} \frac{(-1)^k}{k} sin(k\pi x) \, e^{-(k\pi)^2 t}$$

Thus we are now able to solve the original problem $u(x,t)$, as it has the solution $u(x,t) = v(x) + w(x,t) = x + w(x,t)$.

The solution $u(x,t)$ is given by,

$$u(x,t) = x + \frac{2}{\pi} \sum_{k=1}^{\infty} \frac{(-1)^k}{k} \; sin(k\pi x) \; e^{-(k\pi)^2 t} \tag{12}$$

2.2. **The diffusion equation in two dimensions.** For the two dimensional problem, we want to solve the diffusion equation,

$$\frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} = \frac{\partial u(x,y,t)}{\partial t} \quad t > 0 \;\; x, y \in [0,1] \tag{13}$$

or

$$u_{xx} + u_{yy} = u_t$$

In this case, I am free to chose the initial and boundary conditions by myself. Therefore I have chosen to impose homogenous dirichlet boundary conditions. I have chosen the initial condition to be sine functions, as this will help in calculating the analytical solution for the problem. Therefore, we wish to solve the PDE (13) with the initial condition

$$u(x,y,0) = f(x,y) = sin(\pi x) sin(\pi y) \qquad x, y \in R = [0,1] \times [0,1] \tag{14}$$

and the boundary conditions

$$u(0,y,t) = u(1,y,t) = 0 \qquad 0 \le y \le 1, t \ge 0 \tag{15}$$

$$u(x,0,t) = u(x,1,t) = 0 \qquad 0 \le x \le 1, t \ge 0 \tag{16}$$

Here $u(x,y,t)$ is the unknown function to be solved for, x and y are the spacial coordinates, and t is time.

2.2.1. *The analytical solution.* Just as in the one dimensional case, we wish to find an analytical expression for the solution of the two dimensional problem. Fortunately in this case, an analytical solution is possible. To find the analytical solution we again use the separation of variables technique. We make the ansatz that,

$$u(x,y,t) = X(x)\,Y(y)\,T(t) \tag{17}$$

Inserting this into the diffusion equation (13) and using the boundary conditions, we get the three ordinary differential equations,

$$X^{''} - BX = 0, \qquad X(0) = 0, X(1) = 0 \tag{18}$$

$$Y^{''} - CY = 0, \qquad Y(0) = 0, Y(1) = 0 \tag{19}$$

$$T^{'} - (B + C)T = 0 \tag{20}$$

The solutions to (18) and (19) are given by,

$$X_m(x) = sin(m\pi x), \qquad B = -(m\pi)^2$$

$$Y_n(y) = sin(n\pi y), \qquad C = -(n\pi)^2$$

for $m, n \in \mathbb{N}$. Inserting these values into equation (20) gives the solution,

$$T_{mn} = e^{-(\lambda_{mn})^2 t}$$

where $\lambda_{mn} = \sqrt{(m\pi)^2 + (n\pi)^2}$. From the principle of superposition, we find that the general solution is of the form,

$$u(x,y,t) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} A_{mn}\, sin(m\pi x)\, sin(n\pi y)\, e^{-t[(m\pi)^2 + (n\pi)^2]} \tag{21}$$

The coefficients A_{mn} are determined by the initial condition. A_{mn} are the Fourier coefficients of the double Fourier series for $f(x, y)$. Thus the A_{mn} are given by,

$$\begin{aligned} A_{mn} &= 4\int_0^1\int_0^1 f(x,y)\, sin(m\pi x)\, sin(n\pi y)\, dy\, dx \\ &= 4\int_0^1\int_0^1 sin(m\pi x)\, sin(n\pi y)\, sin(m\pi x)\, sin(n\pi y)\, dy\, dx \end{aligned}$$

In this case, the initial condition chosen is exactly the terms in the integral with $m = n = 1$. Thus the solution to the integral becomes,

$$A_{mn} = \begin{cases} 1, & if\ m = n = 1 \\ 0, & otherwise \end{cases}$$

Therefore the solution $u(x, y, t)$ is given by,

$$u(x,y,t) = sin(\pi x)\, sin(\pi y)\, e^{-2\pi^2 t} \tag{22}$$

3. Numerical methods

In this article we are going to be looking at three numerical schemes for solving the PDE in one dimension, and one numerical scheme for solving the PDE in two dimensions. We will be looking at explicit and implicit finite difference schemes. A scheme is called explicit if the solution at one time step can be computed directly from the solution at the previous time step. A scheme is called implicit, if the solution on the next time step is obtained by solving a system of equations. The basic idea is to replace the derivatives involved in the diffusion equation, by finite differences.

3.1. **Discretization of the PDE.** Before we can look at any numerical scheme, we have to first discretize the PDE.

3.1.1. *One dimension case.* For the one dimensional diffusion equation, we have to replace the domain $[0, L] \times [0, T]$ by a set of mesh points. Let $n \geq 1$ be a given integer. Then we define the grid spacing in the x-direction by $\Delta x = 1/(n+1)$. The grid points in the x-direction are given by $x_j = j\Delta x$ for $j = 0, 1, 2, ..., n+1$. Similarly we define $t_m = m\Delta t$ for integers $m \geq 0$, where Δt denotes the time step. Finally, we let v_j^m denote the approximation of $u(x_j, t_m)$. We discretize the boundary conditions by

$$v_0^m = T_1 = 0 \text{ and } v_{n+1}^m = T_2 = 1$$

We dicretize the initial condition by,

$$v_j^0 = f(x_j) = 0$$

3.1.2. *Two dimensional case.* For the two dimensional diffusion equation, we define equal grid spacing in the x and y directions, such that $\Delta x = \Delta y = h$. The grid spacing is defined by $h = 1/(n + 1)$. The grid points in the x-direction are given by $x_l = l\ h$, and in the y-direction $y_j = j\ h$, for $l, j = 0, 1, 2, ..., n + 1$. We define $t_m = m\Delta t$, for integers $m \geq 0$, where Δt denotes the time step. Finally, we let v_{lj}^m denote the approximation of $u(x_l, y_j, t_m)$. The boundary conditions are discretized by,

$$v_{0,j}^m = v_{n+1,j}^m = 0$$
$$v_{l,0}^m = v_{l,n+1}^m = 0$$

The initial conditions are discretized by,

$$v_{lj}^0 = f(x_l, y_j) = sin(\pi x_l)\ sin(\pi y_j)$$

3.2. **Numerical methods in one dimension.**

3.2.1. *Explicit scheme.* This is the explicit forward Euler scheme. We approximate the derivatives by using a forward difference in time, and a central difference in space. The approximations are,

$$u_t(x, t) = \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} + O(\Delta t) \tag{23}$$

$$u_{xx}(x, t) = \frac{u(x - \Delta x, t) - 2u(x, t) + u(x + \Delta x, t)}{\Delta x^2} + O(\Delta x^2) \tag{24}$$

These approximations motivate the following scheme

$$\frac{v_j^{m+1} - v_j^m}{\Delta t} = \frac{v_{j-1}^m - 2v_j^m + v_{j+1}^m}{\Delta x^2} \tag{25}$$

We can rewrite this scheme in a more convenient form given by

$$v_j^{m+1} = rv_{j-1}^m + (1-2r)v_j^m + rv_{j+1}^m \tag{26}$$

where we define the variable r to be $r = \Delta t/(\Delta x)^2$. The truncation error for this scheme is of $O(\Delta x^2)$ and $O(\Delta t)$.

3.2.2. *Implicit scheme.* This is the implicit backward Euler scheme. We will apply the backward Euler formula in time. We will expand forwards and backwards in space, at a time $t + \Delta t$. This gives us the approximations

$$u_t(x, t+\Delta t) = \frac{u(x, t+\Delta t) - u(x,t)}{\Delta t} + O(\Delta t) \tag{27}$$

$$u_{xx}(x, t+\Delta t) = \frac{u(x-\Delta x, t+\Delta t) - 2u(x, t+\Delta t) + u(x+\Delta x, t+\Delta t)}{\Delta x^2} \tag{28}$$

$$+ O(\Delta x^2) \tag{29}$$

This leads to the following scheme,

$$\frac{v_j^{m+1} - v_j^m}{\Delta t} = \frac{v_{j-1}^{m+1} - 2v_j^{m+1} + v_{j+1}^{m+1}}{\Delta x^2} \tag{30}$$

In order to write this in a more convenient form, we introduce the vector $v^m \in \mathbb{R}^n$ with components $v^m = (v_1^m, ..., v_n^m)^T$. The scheme can be rewritten as,

$$v_j^{m+1} + r(-v_{j-1}^{m+1} + 2v_j^{m+1} - v_{j+1}^{m+1}) = v_j^m \tag{31}$$

Here r is the same as before, $r = \Delta t/(\Delta x)^2$. We can write this in matrix-vector notation as

$$\begin{bmatrix} 1+2r & -r & 0 & \dots & 0 \\ -r & 1+2r & -r & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -r \\ 0 & \dots & \dots & -r & 1+2r \end{bmatrix} \begin{bmatrix} v_1^{m+1} \\ v_2^{m+1} \\ \vdots \\ \vdots \\ v_n^{m+1} \end{bmatrix} = \begin{bmatrix} v_1^m \\ v_2^m \\ \vdots \\ \vdots \\ v_n^m \end{bmatrix} \tag{32}$$

This can be written with the compact notation as

$$(I + rA)\mathbf{v}^{m+1} = \mathbf{v}^m \tag{33}$$

where I is the identity matrix, and A is the tridiagonal matrix,

$$A = \begin{bmatrix} -2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \vdots & 0 & -1 & 2 \end{bmatrix} \tag{34}$$

To compute numerical solutions with this scheme we have to solve linear systems of the form (33), the matrix in this system is tridiagonal. This means we can solve this system using the Thomas algorithm for tridiagonal matrices. The truncation error for this scheme if of $O(\Delta x^2)$ and $O(\Delta t)$.

3.2.3. *Crank Nicholson scheme.* This scheme is a combination of the explicit and implicit methods. The idea is to apply centered differences in space and time, combined with an average in time. With centered differences in space and time, we get an approximation,

$$\frac{1}{\Delta x^2}(v_{j-1}^{m+\frac{1}{2}} - 2v_j^{m+\frac{1}{2}} + v_{j+1}^{m+\frac{1}{2}}) \tag{35}$$

This expression contains $v_j^{m+\frac{1}{2}}$ which is unknown, therefore we choose to replace $v_j^{m+\frac{1}{2}}$ by an arithmetic average,

$$v_j^{m+\frac{1}{2}} \approx \frac{1}{2}(v_j^m + v_j^{m+1}) \tag{36}$$

After using this average and finding the differences, the Crank Nicholson scheme can be written as shown below. Note that the time derivative approximation is the same as we had in the explicit scheme.

$$\frac{v_j^{m+1} - v_j^m}{\Delta t} = \frac{1}{2}\Big(\frac{v_{j-1}^{m+1} - 2v_j^{m+1} + v_{j+1}^{m+1}}{\Delta x^2} + \frac{v_{j-1}^m - 2v_j^m + v_{j+1}^m}{\Delta x^2}\Big) \tag{37}$$

This can be rewritten as

$$v_j^{m+1} - \rho(v_{j-1}^{m+1} - 2v_j^{m+1} + v_{j+1}^{m+1}) = v_j^m + \rho(v_{j-1}^m - 2v_j^m + v_{j+1}^m) \tag{38}$$

Here the variable $\rho = \frac{1}{2}r = \frac{\Delta t}{2(\Delta x^2)}$. This scheme also leads to a matrix system, we can write the system in matrix-vector notation as

$$\begin{bmatrix} 1+2\rho & -\rho & 0 & \dots & 0 \\ -\rho & 1+2\rho & -\rho & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -\rho \\ 0 & \dots & \dots & -\rho & 1+2\rho \end{bmatrix} \begin{bmatrix} v_1^{m+1} \\ v_2^{m+1} \\ \vdots \\ \vdots \\ v_n^{m+1} \end{bmatrix} = \begin{bmatrix} 1-2\rho & \rho & 0 & \dots & 0 \\ \rho & 1-2\rho & \rho & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \rho \\ 0 & \dots & \dots & \rho & 1-2\rho \end{bmatrix} \begin{bmatrix} v_1^m \\ v_2^m \\ \vdots \\ \vdots \\ v_n^m \end{bmatrix}$$

Introducing the vectors $\mathbf{v}^m$ and $\mathbf{v}^{m+1}$, we can write this equation in a more compact form as

$$(I + \rho A)\mathbf{v}^{m+1} = (I - \rho A)\mathbf{v}^m = \mathbf{b}^m \tag{39}$$

Here I is the identity matrix, and A is the tridiagonal matrix given in (34). To solve this set of equations, we can first compute the right hand side $\mathbf{b}^m$, and then solve the resulting set of equations using the Thomas algorithm, just like the implicit method. This means we will solve the system of equations

$$(I + \rho A)\mathbf{v}^{m+1} = \mathbf{b}^m \tag{40}$$

Where the $i-th$ element of $\mathbf{b}$ is given by,

$$b_i^m = v_i^m + \rho(v_{i+1}^m - 2v_i^m + v_{i-1}^m) \tag{41}$$

The truncation error for this scheme if of $O(\Delta x^2)$ and $O(\Delta t^2)$.

3.3. **Numerical methods in two dimensions.**

3.3.1. *Explicit scheme.* The two dimensional diffusion equation was given by

$$u_t = u_{xx} + u_{yy}$$

We want to discretize the position and time. In this case we had equal spacing in the x and y directions, that is, $\Delta x = \Delta y = h$. We get the approximations in the spacial coordinates,

$$u_{xx} \approx \frac{v^m_{l+1,j} - 2v^m_{l,j} + v^m_{l-1,j}}{h^2} \tag{42}$$

$$u_{yy} \approx \frac{v^m_{l,j+1} - 2v^m_{l,j} + v^m_{l,j-1}}{h^2} \tag{43}$$

We use again the forward Euler formula for the first derivative in time. The discretized form for time is then given by

$$u_t \approx \frac{v^{m+1}_{l,j} - 2v^m_{l,j}}{\Delta t} \tag{44}$$

Inserting these approximations into the diffusion equation, we get the scheme

$$\frac{v^{m+1}_{l,j} - 2v^m_{l,j}}{\Delta t} = \frac{v^m_{l+1,j} - 2v^m_{l,j} + v^m_{l-1,j}}{h^2} + \frac{v^m_{l,j+1} - 2v^m_{l,j} + v^m_{l,j-1}}{h^2} \tag{45}$$

rearranging the terms, we get the following explicit scheme,

$$v^{m+1}_{l,j} = v^m_{l,j} + r(v^m_{l+1,j} + v^m_{l-1,j} + v^m_{l,j+1} + v^m_{l,j-1} - 4v^m_{l,j}) \tag{46}$$

Here the left hand side, with the solution at the new time step, is the only unknown term. Here the variable r is defined as $r = \Delta t/h^2$. The truncation error for this scheme is of $O(h^2)$ and $O(\Delta t)$.

4. Stability analysis

We will look at the Von Neumann stability analysis for the PDE. The basic idea of Von Neumann's method is to compare the growth of the analytical and discrete particular solutions. In addition, we want to find conditions on the mesh parameters Δx and Δt, such that the growth of the discrete solutions, are bounded by the analytical solutions.

4.1. **Explicit scheme in one dimension.**

Consider the diffusion equation $u_t = u_{xx}$. This has a solution of the form $u(x,t) = e^{-k^2 t}\, e^{ikx}$. Next we define a function $G(k) = e^{-k^2 t}$, this is the growth factor. The numerical scheme is considered stable if

$$|G(k)| \leq 1 \text{ for all } k$$

The explicit scheme is given by

$$\frac{v^{m+1}_j - v^m_j}{\Delta t} = \frac{v^m_{j-1} - 2v^m_j + v^m_{j+1}}{\Delta x^2} \tag{47}$$

Inserting $u(x,t_m) = e^{ikx}$ into this scheme, we get

$$\begin{aligned} v_j^{m+1} &= e^{ik\Delta xj} + r(e^{ik\Delta x(j+1)} - 2e^{ik\Delta xj} + e^{ik\Delta x(j-1)}) \\ &= (1 + r(e^{ik\Delta x} + e^{-ik\Delta x} - 2))e^{ik\Delta xj} \\ &= G(k)e^{ik\Delta xj} \end{aligned}$$

We get $G(k) = 1 - 2r(1 - cos(k\Delta x))$. In the worst case we would have $k\Delta x = \pi$. This means that we would get $G(k) = 1 - 4r$. In order for the scheme to be stable we must have $|G(k)| \leq 1$. The scheme is stable if

$$r \leq \frac{1}{2} \iff \frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}$$

Hence this scheme is conditionally stable.

4.2. **Implicit scheme in one dimension.** Just like before, we insert the solution $u(x,t_m) = e^{ikx}$ into the implicit scheme (30) . We get that,

$$\begin{aligned} \frac{G-1}{\Delta t} &= G\frac{e^{-ik\Delta x} - 2 + e^{ik\Delta x}}{\Delta x^2} \\ &= \frac{G}{\Delta x^2}(e^{-ik\Delta x} - 2 + e^{ik\Delta x}) \\ &= \frac{2G}{\Delta x^2}(cos(k\Delta x) - 1) \end{aligned}$$

We get that,

$$G(k) = \frac{1}{1 + 2r(1 - cos(k\Delta x))}$$

In this case we have that $|G(k)| \leq 1$, therefore this requirement is fulfilled for all mesh parameters. Hence this scheme is unconditionally stable.

4.3. **Crank Nicholson in one dimension.** We insert the solution $u(x,t_m) = e^{ikx}$ into the Crank Nicolson scheme given by (37). We get that

$$\frac{G-1}{\Delta t} = \frac{1}{2}(G+1)\frac{e^{-ik\Delta x} - 2 + e^{ik\Delta x}}{\Delta x^2}$$

Which gives the expression for $G(k)$ to be,

$$G = \frac{1 - r(1 - cos(k\Delta x))}{1 + r(1 - cos(k\Delta x))}$$

Here also we see that $|G(k)| \leq 1$, therefore this requirement is fulfilled for all mesh parameters. Hence this scheme is unconditionally stable.

4.4. **Explicit scheme in two dimensions.** In this case we define

$$u(x,y,t_m) = e^{ikx}\, e^{ily}$$

Inserting this into equation (45), for the more general case where we have different spacing in the x and y directions, we get

$$\frac{G-1}{\Delta t} = \frac{e^{ik\Delta x} - 2 + e^{-ik\Delta x}}{\Delta x^2} + \frac{e^{il\Delta y} - 2 + e^{-il\Delta y}}{\Delta y^2}$$

Summary of the schemes			
Scheme	Truncation error	Stability requirements	r
Explicit 1D	$O(\Delta x^2)$ and $O(\Delta t)$	$\frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$	$r = \Delta t/(\Delta x)^2$
Implicit 1D	$O(\Delta x^2)$ and $O(\Delta t)$	stable or all Δt and Δx	$r = \Delta t/(\Delta x)^2$
Crank Nicolson 1D	$O(\Delta x^2)$ and $O(\Delta t^2)$	stable or all Δt and Δx	$r = \Delta t/(\Delta x)^2$
Explicit 2D	$O(\Delta h^2)$ and $O(\Delta t)$	$\frac{\Delta t}{h^2} \leq \frac{1}{4}$	$r = \Delta t/h^2$

Table 1. Summary of the basic properties of the numerical methods used to solve the diffusion equation in one and two dimensions.

This leads to the function G being,

$$G = 1 - \frac{2\Delta t}{\Delta x^2}(1 - cos(k\Delta x)) - \frac{2\Delta t}{\Delta y^2}(1 - cos(l\Delta y))$$

In the worst case, we would have $k\Delta x = \pi = l\Delta y$. This means that

$$G = 1 - \frac{4\Delta t}{\Delta x^2} - \frac{4\Delta t}{\Delta y^2}$$

The stability condition gives that

$$\frac{\Delta t}{\Delta x^2} + \frac{\Delta t}{\Delta y^2} \leq \frac{1}{2} \iff \Delta t \leq \frac{1}{2}\Big(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\Big)^{-1}$$

In this case, we have $\Delta x = \Delta y = h$, this means the stability condition becomes

$$\Delta t \leq \frac{1}{2}\Big(\frac{2}{h^2}\Big)^{-1} = \frac{1}{2}\Big(\frac{h^2}{2}\Big) = \frac{h^2}{4} \tag{48}$$

The scheme is stable if

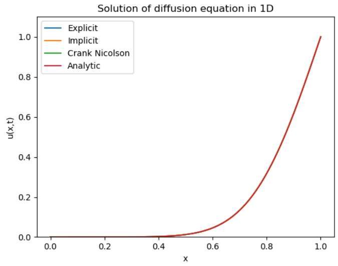
$$\frac{\Delta t}{h^2} \leq \frac{1}{4} \tag{49}$$

Hence this scheme is conditionally stable.

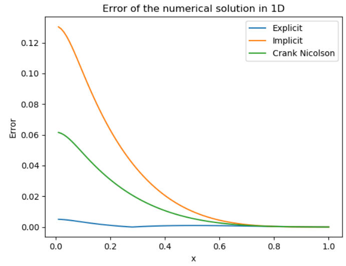
Before presenting the results, it is useful to take a look at the main properties of the numerical schemes. This is given in table 1.

5. Results and discussion

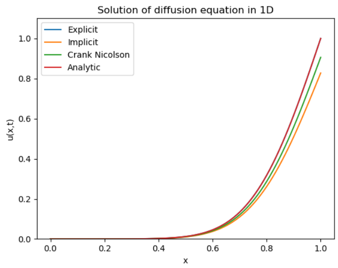
5.1. **One dimensional diffusion equation.** The results for the one dimensional diffusion equation are shown by figures 1-6. Firstly I would have to say something important about my code for the implementation of the three numerical methods. For the implicit scheme and the Crank Nicholson scheme, there is something wrong in my implementation of the tridiagonal solver. As seen in the numerical methods section, the implicit and CR scheme require the solution of a linear system of equations to go from one time step to the next. The matrix for this system is a tridiagonal matrix, that can be solved using the Thomas algorithm. However, I was unable to get my Thomas algorithm functioning correctly to solve the system of equations. Therefore I have used numpy's built in linear algebra function to solve the linear system of equations. For the sake of the stability analysis of the numerical schemes, I decided to use the numpy linear algebra



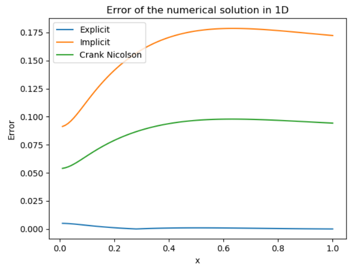
(A) Solution using numpy's linear algebra function to solve the linear system of equations.



(B) Relative error of the solution curves shown in figure 1a for the three numerical schemes.



(C) Solution using my own code for solving the linear system of equations.



(D) Relative error of the solution curves shown in figure 1c for the three numerical schemes.

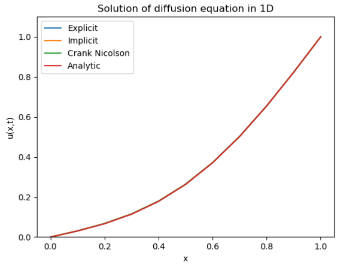
FIGURE 1. Solution of the diffusion equation in 1D from the three numerical schemes and the actual solution. This result is obtained by Nx=100, Nt=1000, and T=0.02. The value for $\Delta x = 0.01$, $\Delta t = 2 \times 10^{-5}$ and $r = 0.20$.

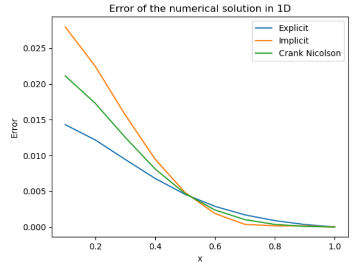
function so I can see that even if my function worked correctly, how would this affect the numerical schemes. This is a problem for the implicit and the CN scheme, the explicit scheme is working fine. Thus I have presented my results using my own tridiagonal solver, and using the numpy function to solve the equations in the implicit and the CN scheme.

Figure 1a shows the numerical solution of the diffusion equation for $\Delta x = 0.01$, using Nx=100, Nt=1000 at time 0.02. Due to the stability condition for the explicit scheme, we must have $\Delta t \leq \Delta x^2/2$. This means that we must have

$$\Delta t \leq \frac{\Delta x^2}{2} = \frac{(0.01)^2}{2} = \frac{1}{20000}$$

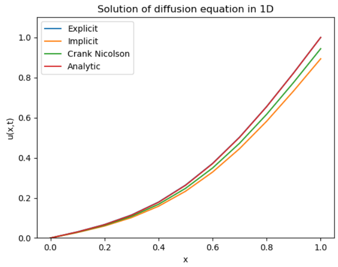
Here we use $\Delta t = 2 \times 10^{-5}$, this fulfills the stability requirement and the explicit scheme performs just as well as the other schemes. The plot shows that the solution is quite curved, the numerical schemes all match the analytical solution. This shows that the numerical scheme is correct. Figure 1c shows the results with my own tridiagonal solver for the implicit and CN scheme. I can see that my results are not far away from the actual solution, something odd starts happening as x approaches 1, the three schemes begin to go away from each other. I have also looked at the relative error for the

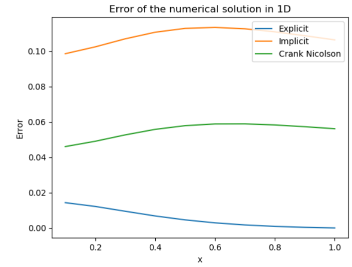




(A) Solution using numpy's linear algebra function to solve the linear system of equations.

(B) Relative error of the solution curves shown in figure 2a for the three numerical schemes.

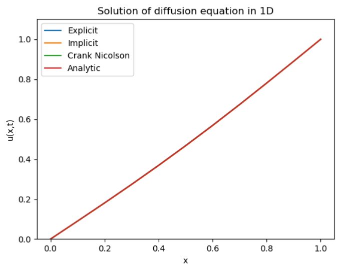


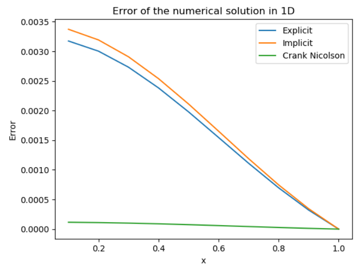


(C) Solution using my own code for solving the linear system of equations.

(D) Relative error of the solution curves shown in figure 2c for the three numerical schemes.

FIGURE 2. Solution of the diffusion equation in 1D from the three numerical schemes and the actual solution. This result is obtained by Nx=10, Nt=100, and T=0.1. The value for $\Delta x = 0.1$, $\Delta t = 0.001$ and $r = 0.101$.

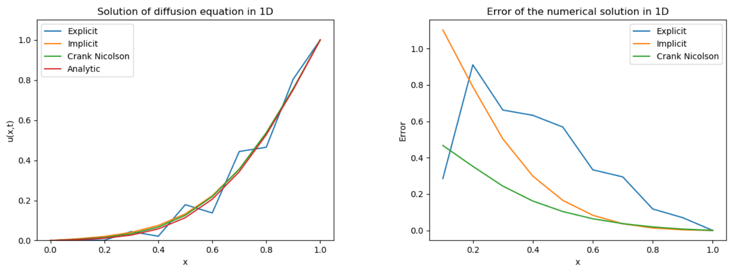




(A) Solution curves with the three numerical schemes for the 1D diffusion equation.

(B) Relative error of the solution curves shown in figure 3a for the three numerical schemes.

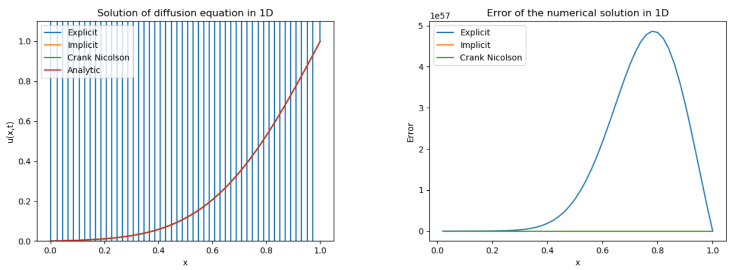
FIGURE 3. The solution is a straight line for Nx=10, Nt=100, and T=0.3. The value for $\Delta x = 0.1$, $\Delta t = 0.003$ and $r = 0.30$. This is the steady state solution for the 1D equation. Note however that this results was obtained by using using numpy's linear algebra function to solve the linear system of equations.



(A) Solution curves for the three numerical schemes

(B) The corresponding relative error for figure 4a

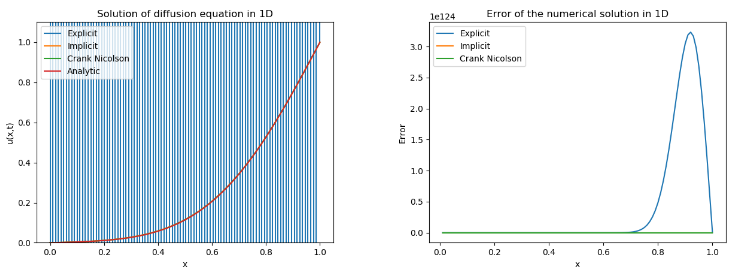
FIGURE 4. Results for Nx=10, Nt=10, and T=0.05. The value for $\Delta x = 0.1$, $\Delta t = 0.0055$ and $r = 0.55$. This result is obtained from using numpy's function to solve the matrix system.



(A) Solution curves for the three numerical schemes

(B) The corresponding relative error for figure 5a

FIGURE 5. Results for Nx=50, Nt=100, and T=0.05. The value for $\Delta x = 0.2$, $\Delta t = 0.0005$ and $r = 1.26$. This result is obtained from using numpy's function to solve the matrix system.



(A) Solution curves for the three numerical schemes

(B) The corresponding relative error for figure 6a

FIGURE 6. Results for Nx=100, Nt=100, and T=0.05. The value for $\Delta x = 0.01$, $\Delta t = 0.005$ and $r = 5.05$. This result is obtained from using numpy's function to solve the matrix system.

solutions, This is shown in figure 1 b and c. The relative error is defined as

$$\text{relative error} = \frac{|\hat{u}(x,t) - u(x,t)|}{|\hat{u}(x,t)|}$$

where $\hat{u}(x,t)$ is the numerical solution, and $u(x,t)$ is the analytic solution. Looking at figure 1b, I can see that in the beginning, the error is almost quite high, and as x goes towards 1, the error starts to decrease. In addition, the implicit scheme has the highest error of 0.12, while the CR has an error of 0.08, and the explicit scheme has 0.01. At about $x = 0.06$, the errors are almost 0. The explicit scheme has the lowest error in the start, this could be sue to the fact that the explicit scheme is stable as $r < 1/2$. As the error is decreasing as x goes to 1, the numerical solution is getting closer to the actual solution. Looking at figure 1d, I can see that the error in my own implementation is not decreasing as x goes to 1. In fact the error seems to remain the same as x increases. This is what I would expect as the numerical solution in figure 1c is not completely correct. However the implicit scheme still has a higher error compared to the other two schemes.

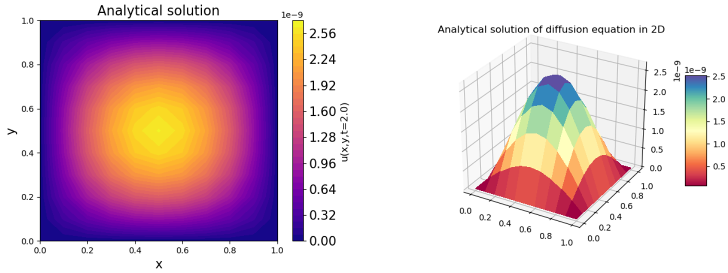
Figure 2a shows shows the numerical solution for $\Delta x = 0.1$, using Nx=10, Nt=100 at time 0.1. In order for that explicit scheme to be stable, we must have $\Delta t \leq \Delta x^2/2$. This means that we must have

$$\Delta t \leq \frac{\Delta x^2}{2} = \frac{(0.1)^2}{2} = \frac{1}{200}$$

Here we use $\Delta t = 0.001$, this means the explicit scheme works fine, just as well as the others. The plot shown is a bit less curved than the one in figure 1. Figure 2b shows the error in the numerical schemes in figure 2a. The curves start off high, but go towards zero as x goes to 1. Again figure 2c shows the same results but with my own tridiagonal solver for the implicit and CN schemes. I can see that it is quite close to the actual solution. The error cures in figure 2d show similar behaviour to the curved in figure 1d. Seems like the error is not necessarily decreasing as x is increasing towards 1.

Figure 3 shows the *stationary solution* of the diffusion equation. The diffusion equation has the property that it converges to a stationary solution as $t \to \infty$. The steady state occurs when $u(x, t \to \infty) = ax + b$, the linear function shows that the solution is in steady state. This solution is obtained at $T = 0.3$, as we saw in figure 1a at $T = 0.02$, and figure 2a and $T = 0.1$, the solution is approaching the straight line. The relative error in figure 3b is also going to zero in a straight line. However we see that the CN scheme has the lowest error here. In figure 3 the $r = 0.31$, so the explicit scheme is still stable.

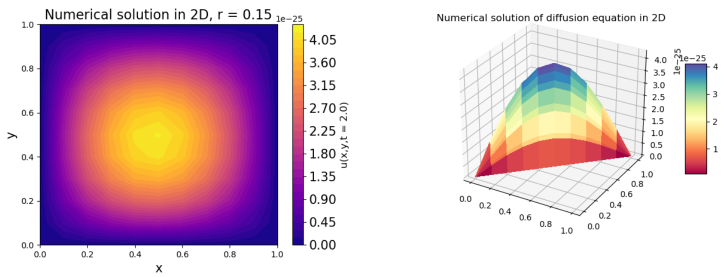
Figure 4, 5 and 6 show the solutions for different values of r, as well as their corresponding relative error curves. Figure 4 shows the numerical solution when $r = 0.55$, figure 5 with $r = 1.26$, and figure 6 with $r = 5.05$. The common thing in these three plots is the fact that r breaks the stability condition for the explicit scheme. This is why we see the odd behaviour of the explicit scheme curves in the three figures. In figure 4a, the explicit method starts to diverge from the other two methods, and in figure 5a and 6a, the solution is completely off the analytical solution. We can see that



(A) Visualization of the analytical solution for the 2D PDE.

(B) Shape of the surface for the visualization in figure 7a.

FIGURE 7. The analytical solution of the 2D diffusion equation. This is obtained using h=0.1, r=0.15, and T=2. The value for $\Delta t = 0.0015$.



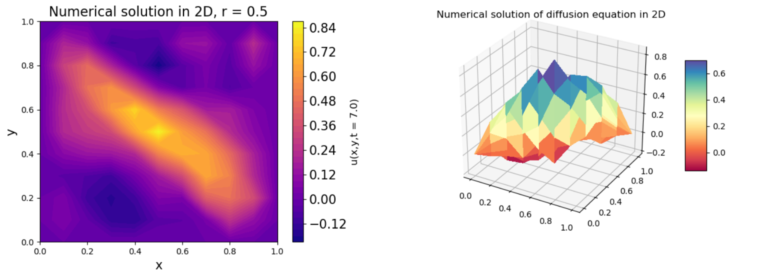
(A) Visualization of the solution

(B) Shape of the surface for the visualization in figure 8a.

FIGURE 8. Numerical solution of the 2D diffusion equation. This is obtained using h=0.1, r=0.15, and T=2. The value for $\Delta t = 0.0015$.

the implicit and CN schemes are still stable even though $r \geq 1/2$, this is what we would expect, as these methods are stable for all values of Δx and Δt. The relative error in figure 4b shows that the CN scheme has the lowest error, and the error is close to 0 in figure 5b and 6b for the implicit and CN methods. Theoretically, we would expect the CN scheme to have the lowest error, and the explicit scheme to have the worst error, if the stability condition is not full filled.

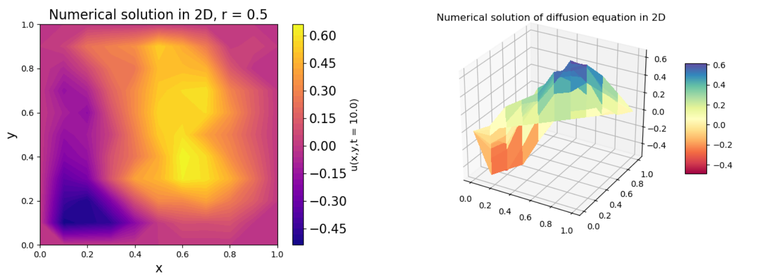
5.2. **Two dimensional diffusion equation.** The results for the two dimensional diffusion equation are shown by figures 7-10. The numerical scheme that I have implemented to solve the diffusion equation in two dimensions, is the explicit scheme in 2D. Thus there is only one numerical scheme in the 2D case. However, just like the 1D case, this scheme also has a stability condition that must be fulfilled to get stable results. Figure 7a shows the analytical solution for the 2D diffusion equation, figure 7b shows the shape of the surface. This solution is obtained by $h = 0.1, r = 0.15, T = 2$, and $\Delta t = 0.0015$. The surface is quite smooth with the highest value being



(A) Visualization of the solution

(B) Shape of the surface for the visualization in figure 9a.

FIGURE 9. Numerical solution of the 2D diffusion equation. This is obtained using h=0.1, r=0.5, and T=7. The value for $\Delta t = 0.005$.



(A) Visualization of the solution

(B) Shape of the surface for the visualization in figure 9a.

FIGURE 10. Numerical solution of the 2D diffusion equation. This is obtained using h=0.1, r=0.5, and T=10. The value for $\Delta t = 0.005$.

$u(x, t = 2) = 2.56$. This is what we will compare our numerical solutions with.

Figure 8 shows the numerical solution with $h = 0.1, r = 0.15, T = 2$, and $\Delta t = 0.0015$. The analytical solution with these values is shown in figure 7. The numerical solution is quite close to the analytical solution. If we compare the two surface plots, the numerical solution has a highest value being $u(x, t = 2) = 4.05$, which is not the same as the one for figure 7. The shape of the surface in figure 8b is not as well defined as the one in 7b. However we can see that the general shape in correct. Here I have used $h = 0.1$, so the mesh grid is not well defined. Perhaps I could have gotten a more accurate solution if I used much smaller spacing, in the x and y directions. The solution in figure 8 is stable, as $r = 0.15 \leq 0.25$, we just need to have more mesh points to get closer to the analytical solution.

Figure 9 and 10 show the numerical solution when the stability condition is not full filled, the explicit scheme is unstable and the results are completely off the analytical solutions. Figure 9 shows the numerical solution with $r = 0.5$, this value of r breaks the stability condition of the explicit scheme, leading to innaccurate numerical results. Figure 9 shows the results of $r = 0.5, h = 0.1$, and $T = 7$. It is easy to see from figure 9b that the shape of the surface curve is not like the one in figure 7b, the surface is not smooth, it is quite rocky. Figure 10 shows the results of the numerical solution with $r = 0.5, h = 0.1$, and $T = 10$. The values are the same as the ones in figure 9, except now we are at a later time, at $T = 10$. From figure 10b we can see that the shape of the surface plot is more uneven, it looks nothing like the one in figure 7a. Figure 9 and 10 show the divergence of the scheme if the stability condition is not full filled. Therefore is stability condition is violated, we cannot expect our results to be similar to the analytical solution. In that case, we would have to use a different scheme to solve our numerical problem.

6. Conclusion

The main task in this article was to solve the diffusion equation in one and two dimensions, using numerical schemes. We have definitely achieved this. Based on my results for the one dimensional diffusion equation, I can say that the best method to implement would be the Crank Nicholson method. This method is stable for all values of Δt and Δx, and it had the lowest relative error which we saw in figure 4. However if the stability condition for the explicit scheme, which is $\frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}$, is full filled, then the explicit scheme had the lowest relative error, compared to the other two. In addition the explicit scheme is simple to implement, and it does not require solving a system of linear equations, as the implicit and CN methods require. Although we can use any value of Δt in the implicit and CN schemes, increasing Δt leads to a loss of accuracy in the numerical solution. For the two dimensional diffusion equation, the explicit method works quite well, if we are able to satisfy the stability condition, of $\frac{\Delta t}{h^2} \leq \frac{1}{4}$. This method is easy to implement, and gives accurate results, as long as we satisfy the stability condition. In order to improve my results, I would try to fix my code for the tridiagonal solver so that I can get accurate results.

7. References

- All codes can be found in the github repository `https://github.uio.no/shafaqns/FYS4150/tree/master/project5`
- Morten Hjorth-Jensen. Computational Physics, Lecture notes, Fall 2015, chapter 10. 2015
- Ocw.mit.edu. 2009. Von Neumann Stability Analysis. [online] Available at: `https://ocw.mit.edu/courses/mathematics/18-336-numerical-methods-f` `lecture-notes/MIT18_336S09_lec14.pdf` [Accessed 18 December 2020].
- C. Daileda, R., 2012. The Two Dimensional Heat Equation. [online] Ramanujan.math.trinity.edu. Available at: `http://ramanujan.math.trinity.edu/rdaileda/teach/s12/m3357/lectures/lecture_3_6_short.pdf` [Accessed 18 December 2020].
- Tveito, A., 2009. Introduction To Partial Differential Equations. New York: Springer, pp.119-120, 140-143.