FYS3150: Project 5: Diffusion Equation

Henrik Haugerud Carlsen, Martin Moen Carstensen and Øyvind Augdal Fløvig

December 19, 2019

Abstract

The goal of this report is to solve the heat equation in both one and two-dimensions with appropriate boundary conditions. The numerical schemes Forward-Euler, Backward-Euler and Crank-Nicolson were used to solve the problem.
There was no success in determining an optimal numerical scheme as there were no substantial differences between the three methods. For $\Delta t > 1/2\Delta x^2$ the Forward Euler scheme became unstable and diverged to both plus and minus infinity. We were able to solve the two dimensional case, but with the used boundary conditions the result did not resemble the one dimensional solution.

1 Introduction

A key component in understanding physical systems is understanding how they evolve in time. Often the time evolution of these systems can be described mathematically as partial differential equations. Unfortunately these equations can be hard, if not impossible to solve analytically, but with numerical methods the solutions can be approximated. Different numerical schemes yield different numerical errors and some are faster than others. Understanding which methods work best for given problems is key in order to make the best possible model. Applications of partial differential equations range from modeling populations, quantum mechanical systems, the earths mantle or how heat spreads in a metal rod.

In this report the aim is to utilize the numerical schemes Forward-Euler, Backward-Euler and Cranck-Nicolson to solve a partial differential equation, the heat equation(also known as the diffusion equation). The truncation errors and stability properties will be analysed, and an optimal scheme chosen. Lastly the two dimensional case will be solved by translating the schemes to two dimensions. For both the one and two-dimensional case the possibility of close form solutions for the respective boundary conditions were looked at.

2 Method

In order to solve the diffusion equation we have used several numerical methods. For the one dimensional problem we can find an analytical solution for our initial and boundary conditions. The diffusion equation we want to solve is:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} \ . \tag{1}$$

We know that our solution should converge to a steady state when $t \to \infty$ and this gives us:

$$\frac{\partial^2 u(x, t \to \infty)}{\partial x^2} = 0 \Rightarrow u(x, t \to \infty) = ax + b \ ,$$

and the coefficients a and b can be determined by looking at the boundary conditions where we have:

$$u(0) = b = 0 \ ,$$

and

$$u(1) = a = 1 \ ,$$

so we get:

$$u(x, t \to \infty) = x \ .$$

This is a solution to the diffusion equation as well but it does not satisfy the initial conditions of $u(x, 0) = 0$, so we try with a solution of the form

$$u(x, t) = x + f(x, t) \ ,$$

where the result would be that $f(x, 0) = -x$ so $f(x, t)$ is a general solution of the diffusion equation with boundary conditions of 0. The solution is then:

$$f(x, t) = \sum_{n=1}^{\infty} A_n sin(n\pi x) e^{-(n\pi)^2 t} \ .$$

The coefficients A_n can be found by setting $t = 0$ and using the initial condition of $g(x) = -x$. This gives

$$u(x, 0) = g(x) = \sum_{n=1}^{\infty} A_n sin(n\pi x) \ .$$

From this we see that A_n is the Fourier coefficients of $g(x)$, so then A_n is determined by

$$A_n = 2\int_0^1 (-x) sin(n\pi x) dx = (-1)^n \frac{2}{n\pi} \; .$$

With the Fourier coefficients we end up with the complete solution as:

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

2.1 Explicit Euler

When we apply explicit Euler we start off by discretizing the differential equation, in this case the diffusion equation given by

$$u_t = u_{xx}.$$

Discretizing the time derivative gives us

$$u_t = \frac{u(x, t+\Delta t) - u(x,t)}{\Delta t} = \frac{u(x, t_{j+1}) - u(x, t_j)}{\Delta t} \; ,$$

while discretizing the position derivatives gives

$$u_{xx} = \frac{u(x+\Delta x, t) - 2u(x,t) + u(x-\Delta x, t)}{\Delta x^2} \; , \tag{3}$$

$$= \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{\Delta x^2} \; .$$

Setting these expressions equal to each other, defining $\alpha = \Delta t/\Delta x^2$ and solving for $u(x, t_{i+1})$ gives us the equation

$$u(x_i, t_{j+1}) = \alpha u(x_{i-1}, t_j) + (1-2\alpha) u(x_i, t_j) + \alpha u(x_{i+1}, t_j).$$

We can reformulate this as a matrix equation,

$$\mathbf{A}\mathbf{V}_j = \mathbf{V}_{j+1} \; ,$$

where $\mathbf{A}$ is a matrix which looks like

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

and the elements of the vector $\mathbf{V}_j$ consist of all the different $u(x_i)$ values for that particular time step. Applying the initial condition

$$u(x, 0) = 0 \; ,$$

and the boundary conditions

$$u(x, L) = 1 \,,$$

$$u(x, 0) = 0 \,,$$

we can now solve the matrix problem above for t_0 and iterate through all the time steps, obtaining the full solution.

It is however important to keep in mind when using the explicit Euler method that it is not unconditionally stable. The equation

$$\frac{\Delta t}{\Delta x^2} < \frac{1}{2}$$

must be satisfied for the numerical scheme to converge to the correct solution.

2.2 Implicit Euler

In the explicit scheme we looked to express the time with the forward formula for the time derivative. In the implicit scheme however we want to look at the backward formula, which discretized is;

$$u_t \approx \frac{u(x_i, t_j) - u(x_i, t_j - \Delta t)}{\Delta t} = \frac{u_{i,j} - u_{i,j-1}}{\Delta t} \,, \tag{5}$$

where we would still have a truncation error which is of size $O(\Delta t)$. Now we still use the same approximation for the second derivative of the positions which is given in 3

$$u_{xx} \approx \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \,. \tag{6}$$

By putting this together in the diffusion equation and again defining $\alpha = \Delta t / \Delta x^2$ we get:

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

We can now define a matrix A using this equation where we get

$$\mathbf{A} = \begin{bmatrix} 1+2\alpha & -\alpha & 0 & 0 & \cdots \\ -\alpha & 1+2\alpha & -\alpha & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \cdots \\ 0 & \cdots & -\alpha & 1+2\alpha & -\alpha \\ 0 & \cdots & \cdots & -\alpha & 1+2\alpha \end{bmatrix} \,.$$

Using this matrix, our problem now becomes a tridiagonal matrix problem of the form

$$Au = v \, ,$$

where v is the previous time step and u is the next that we want to find. For solving this kind of linear problem we use the numerical recipes from Teukolsky[2] for solving a tridiagonal system. So for each time step we can find u by using this tridiagonal solver.

Now we just need to set up the boundary and initial conditions of the system we want to solve. Here we use the same condition as for the explicit Euler scheme. And with this we can solve for the next time step and then iterate to find the full solution in time

From this we can find a numerical approximation to the solution for any number of given time steps by solving the tridiagonal system for each time step with the given initial and boundary conditions.

2.3 Crank-Nicolson

For the Crank-Nicolson scheme we want to combine the implicit and explicit methods to generate a more general approach to this one dimensional diffusion problem. We now introduce a parameter θ and with that we can set up the diffusion equation as

$$\frac{\theta}{\Delta x^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + \frac{1-\theta}{\Delta x^2}(u_{i+1,j-1} - 2u_{i,j-1} + u_{i-1,j-1}) = \frac{1}{\Delta t}(u_{i,j} - u_{i,j-1}) \, . \tag{7}$$

From this equation we can see that setting $\theta = 0$ we end up with the explicit scheme while $\theta = 1$ gives us the implicit scheme. However, by setting $\theta = 1/2$ we obtain the scheme which is named after its inventors Crank and Nicolson. This scheme is also stable for all possible combinations of Δx and Δt and have a truncation error that goes like $O(\Delta t^2)$. The equation we get then becomes

$$\frac{1}{\Delta x^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + \frac{1}{\Delta x^2}(u_{i+1,j-1} - 2u_{i,j-1} + u_{i-1,j-1}) = \frac{2}{\Delta t}(u_{i,j} - u_{i,j-1}) \, . \tag{8}$$

By again defining $\alpha = \Delta t/\Delta x^2$ and collecting the elements with the same time index we get the final equation of

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

which in matrix form becomes

$$(2\hat{I}+\alpha\hat{B})U_j = (2\hat{I}-\alpha\hat{B})U_{j-1} \ .$$

From this we can solve the right hand side of this equation with the forward scheme where we get

$$\tilde{U}_{j-1} = (2\hat{I}-\alpha\hat{B})U_{j-1} \ .$$

Using this result we end up with a tri-diagonal problem which we solved for the implicit Euler. the equation now becomes

$$(2\hat{I}+\alpha\hat{B})U_j = \tilde{U}_{j-1} \ .$$

This can now be solved using the same tri-diagonal solver as we used in the implicit scheme and solve for time step j and using an iterative method we can find the solution for any given time t

2.4 2D - diffusion

The two dimensional diffusion equation is, for $t > 0$, given as:

$$\frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} = \frac{\partial u(x,y,t)}{\partial t} \ , \tag{9}$$

with $x, y \in [0,1]$. The model we use approximates the function $u(x,y,t)$ on the unit square, which means that $0 \leq x < 1$ and $0 \leq y < 1$. This is done by discretizing the function as $u(x,y,t)$ as $u_{i,j}^{(}n)$. Because we are now working in three dimensions(time included) the timestep is denoted by n, i denotes x and j denotes y. With $x = i\Delta x, y = j\Delta y$ and $t = n\Delta t$ the discretization, when applying finite difference approximation, becomes:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = D[\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{(\Delta y)^2}] \ . \tag{10}$$

Solving this equation for the systems state at $n+1$ with $D = 1$ we get:

$$u_{i,j}^{n+1} = u_{i,j}^{n} + \Delta t[\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{(\Delta y)^2}] \ . \tag{11}$$

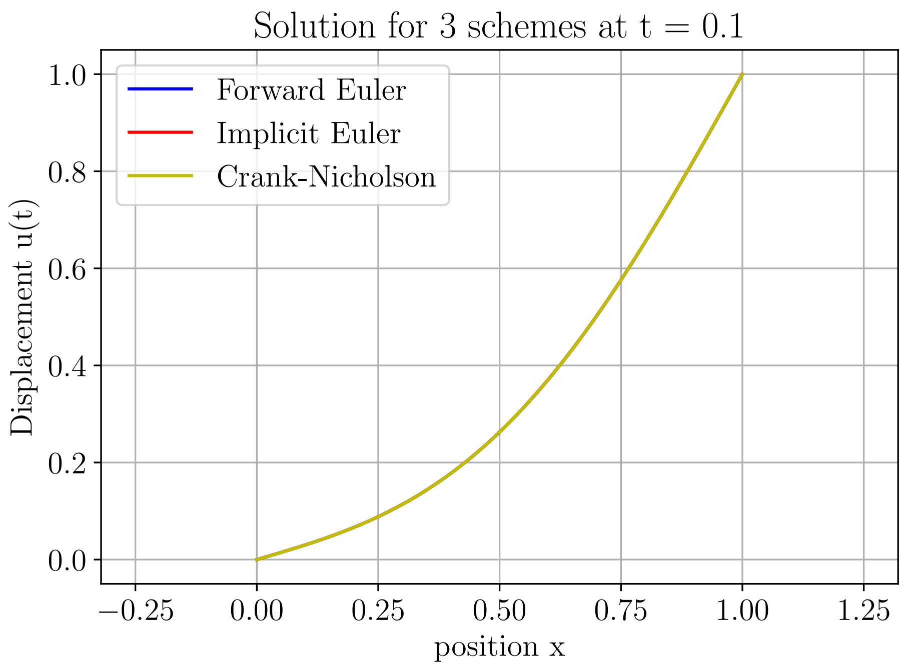
The initial boundary conditions set every point in the plane to zero except at $x = 1$ where $u = 1$, in other words: $u^{n=0}(x = 1) = 1$. The largest time step which can be used before the scheme becomes unstable is given by Hill, C.[1] as:

$$\Delta t = \frac{1}{2D}\frac{(\Delta x \Delta y)^2}{(\Delta x)^2 + (\Delta y)^2} \ . \tag{12}$$

The resolution in x direction is set to be equal to hte resolution of the y direction, so $\Delta x = \Delta y$. This is done because there is no obvious reason to prefer better resolution in one direction than the other. Thus the maximum time step, with $D = 1$, becomes:

$$\Delta t = \frac{1}{4}\frac{\Delta x}{(\Delta x)^2} = \frac{1}{4}\frac{\Delta y}{(\Delta y)^2} \ . \tag{13}$$

3 Results



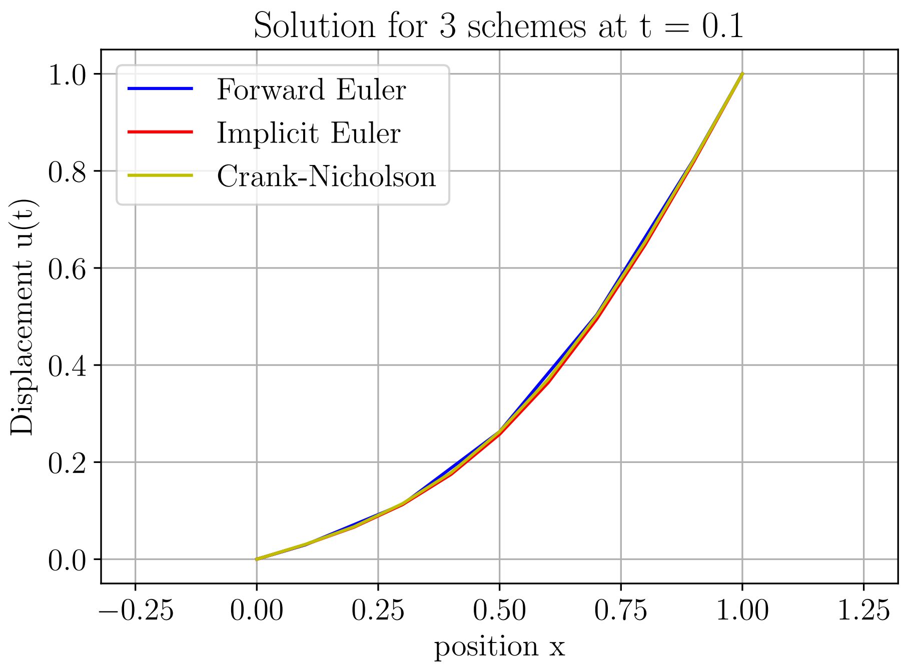
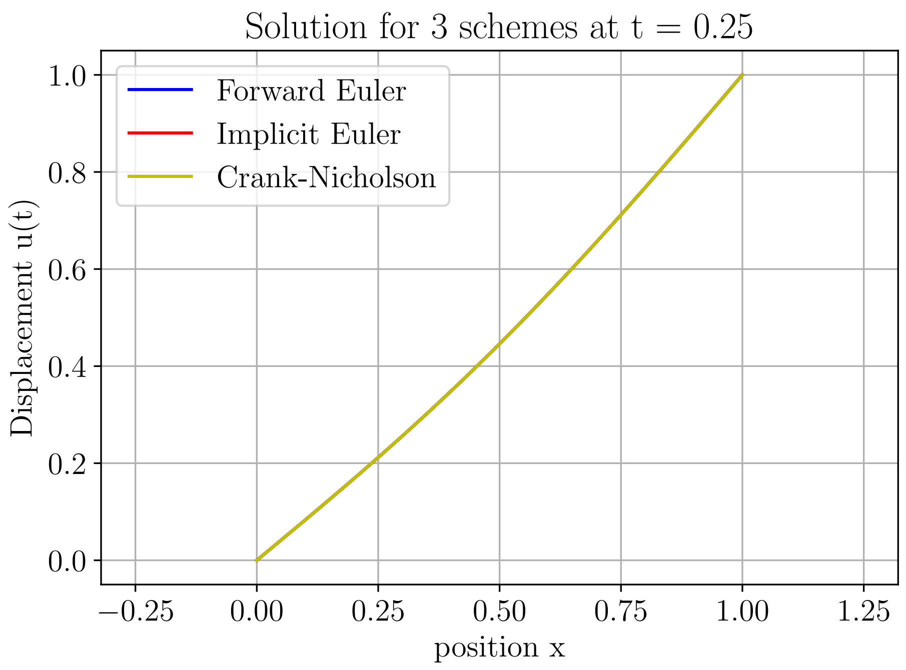


Figure 1: Solution of the differential equation at $t = 0.1$ for step size $\Delta x = 0.01$ and $\Delta x = 0.1$ respectively.



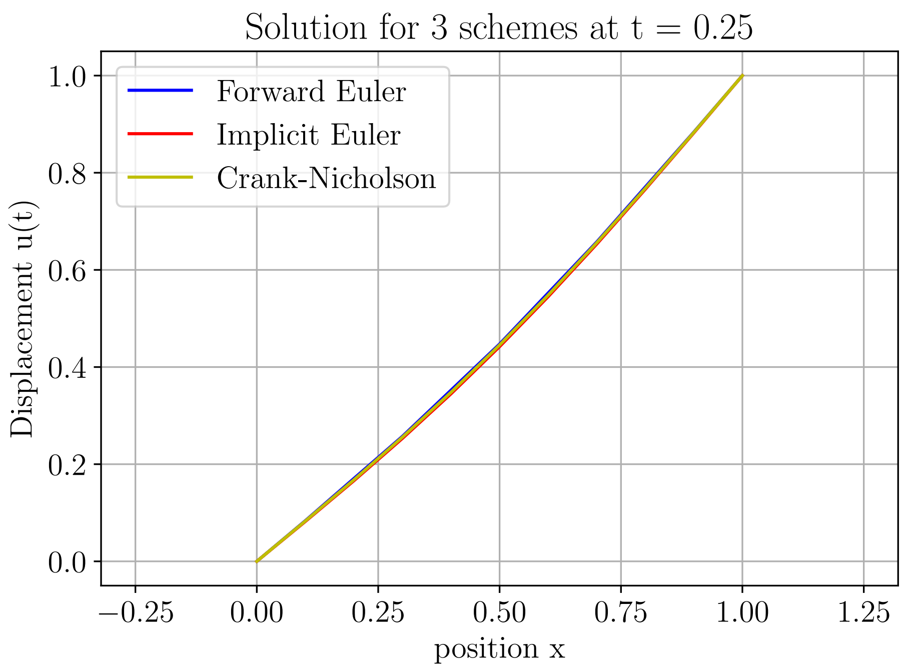
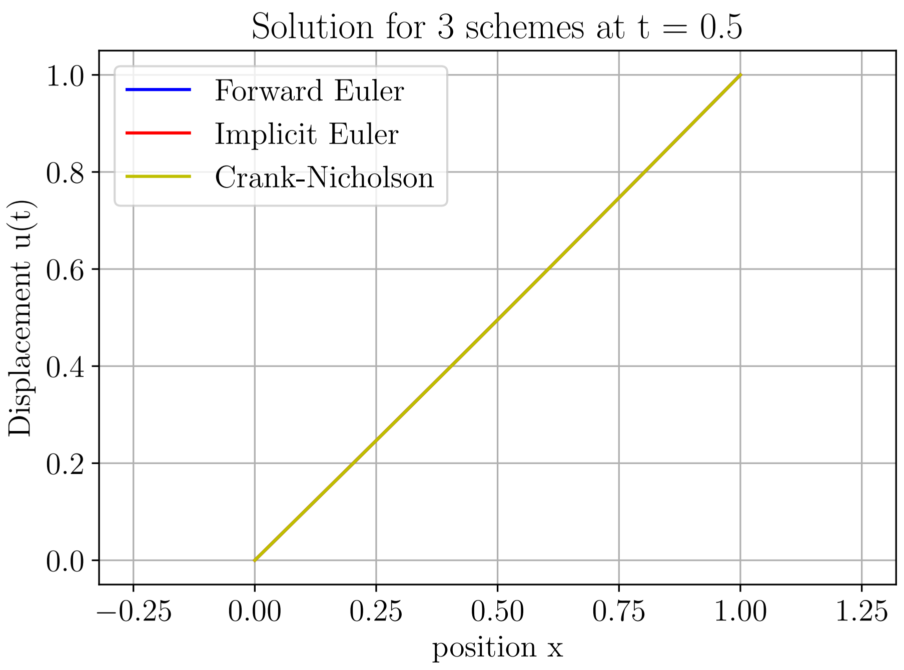


Figure 2: Solution of the differential equation at $t = 0.25$ for step size $\Delta x = 0.01$ and $\Delta x = 0.1$ respectively.



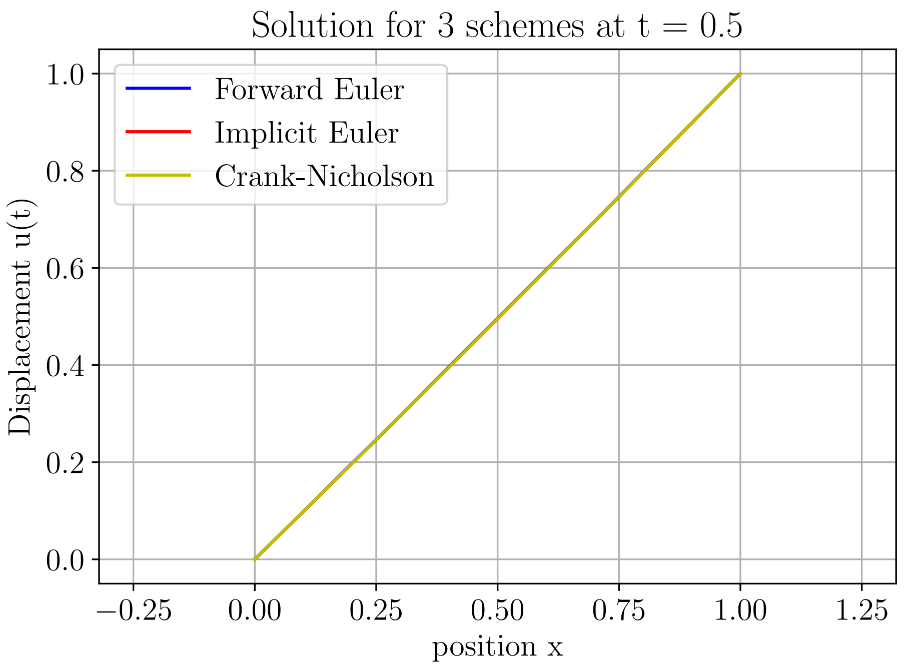


Figure 3: Solution of the differential equation at $t = 0.5$ for step size $\Delta x = 0.01$ and $\Delta x = 0.1$ respectively.

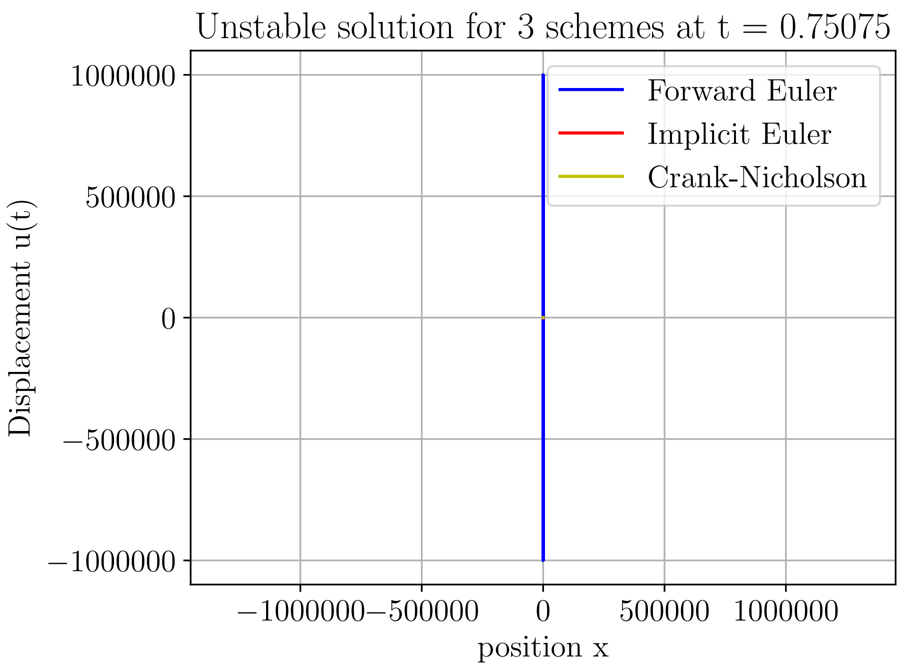


Figure 4: The unstable solution obtained with explicit Euler plotted together with the unconditionally stable solutions of the Crank-Nicholson schemes and the implicit Euler scheme.

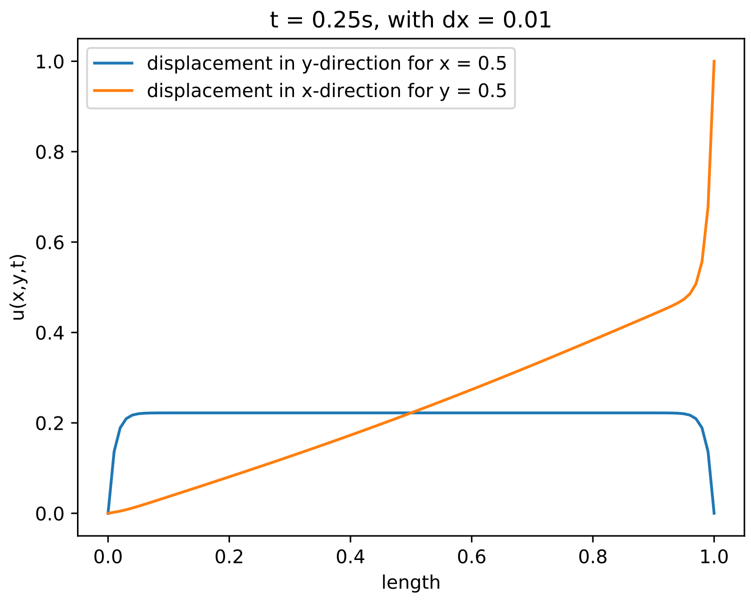


Figure 5: solution for the 2-dimensional diffusion equation shown in both y and x directions for t = 0.25s.

4 Discussion

As we can see from the plots there is very little difference between the numerical schemes when applied to this problem. The only place we can spot any difference at all is the solution at $t = 0.1$ with step size $\Delta x = 0.1$, though this difference vanishes as the system evolves in time. The only case where can detect a substantial difference is when the stability requirement of explicit Euler is violated and the solution blows up completely. We see that the implicit Euler and the Crank-Nicholson schemes behave normally, as we expected since they are unconditionally stable schemes.

For the two dimensional case different boundary conditions than what we chose might be more optimal. The scheme could have yielded better results with periodic boundary conditions in the y direction. This might give us a more identical solution for the x direction as what we got from the one dimensional diffusion equation.

5 Conclusion

There where no substantial differences between the three schemes in the stable region. We were unable to select an optimal scheme for solving the one dimensional heat equation. However for a $\Delta t > 1/2\Delta x^2$ we see that the forward Euler scheme becomes unstable and diverges in both directions. So if we want to calculate the solution for larger time intervals so that more time can pass we should look to implement the implicit scheme or the Crank-Nicolson scheme. Periodic boundary conditions for the two dimensional case would have been beneficial to try out.

6 Bibliography

[1] Hill, C., *The two-dimensional diffusion equation*, read 19.12.2019, https://scipython.com/book/chapter-7-matplotlib/examples/the-two-dimensional-diffusion-equati

[2] Taukolsky, S.A., *Numerical recipes, the art of scientific programming*, pages 56-57, https://e-maxx.ru/bookz/files/numerical_recipes.pdf?fbclid=IwAR0OY52s8XNhsOyazE3WwG1cNUZGDaRMsKl_LKGz_TZ0rfieR2mr6N64Q20

[3] Hjort-Jensen, M.(Fall 2015), Lecture notes, Computational Physics FYS3150, Chapter 10. https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf