Investigating Finite Difference Schemes of the Heat Equation

Aaby I.,Rashid S., Steinnes L. *

Department of physics, University of Oslo, P.O. Box 1048 Blindern, N-0316 Oslo, Norway

Abstract

In this paper, the diffusion equation is numerically solved in one and two dimensions with finite difference methods, with a focus on numerical stability and accuracy. The Forward Euler, Implicit Backward Euler and Crank-Nicolson methods are applied to the 1D case, whereas a Backward Euler scheme with Jacobi iterative method solves the 2D case. In 1D, all methods approaches the stationary state of the diffusion system, as long as the stability criteria is being met. Forward Euler perform best on runtime and error estimates in this case, but experience non-physical oscillations and amplitude growth when the stability criteria is not met. Yet, it provides a less computationally costly algorithm. In 2D, both a product of sines and a Gauss function is used as initial conditions, and the numerical solution handles both cases well, as it reproduces the qualitatively expected solution behaviour. The 2D implicit scheme is numerically stable, but for high Δt resolution, it has a high runtime due to the computational cost of Jacobi iterations.

I Introduction

It is said that mathematics is the language of the natural world, and it could not be more true in the case of partial differentials equations (PDEs). The equations of partial derivatives first arose within physics, to describe models of of continuous media, and was then expanded upon to include everything from elasticity, vibrations, heat conduction, diffusion, electricity and potential fields [1].

French mathematician Jean le Rond d'Alembert first described the one dimensional wave equation in 1752, and a description of the two and three dimensional equations followed shortly thereafter from the works of Leonhard Euler and Daniel Bernoulli [1]. Another famous equation, the heat equation, was introduced in 1822 by Jean-Baptiste

*isakaab@student.matnat.uio.no, sidrar@student.matnat.uio.no, lasse.steinnes@fys.uio.no [December 19, 2020]

Joseph Fourier [1]. In the next century, a comprehensive and rich theoretical foundation was created, which set the physical models in a rigorous theoretical framework.

Although analytical methods have been developed to solve PDEs, there are many equations which are not solvable with a closed form solution, or there might not (yet) exist a solution at all. Thus, there is a demand for numerical algorithms which both give accurate and stable discrete solutions of the problem at hand.

The most common numerical methods to solve PDE's are the finite element method (FEM) and finite difference method (FDM) [2]. Whereas the finite element method approximates the solution as a finite sum of basis functions, and solves a linear system to obtain the unknown coefficients, in FDM one discretizes the differential equations by approximating the derivative [2]. Thus, a numerical scheme can be implemented, and a solution can be obtained for a given dimensionality. For explicit schemes, one can obtain the numerical solution directly. However, in implicit schemes, a linear system must be solved.

The aim of this paper is to apply FDMs to solve the heat equation numerically in one and 2 dimensions. There is a particular emphasis on verification and numerical analysis of a method's accuracy and stability. The methods applied are the explicit Forward Euler method (FE), and the implicit schemes Backward Euler (BE) and Crank-Nicolson (CN).

The paper is laid out by first introducing the heat equation and explaining the theory of relevant finite difference methods, including accuracy and stability considerations regarding the numerical solution of the PDE. In addition, a short description is given of the choice of algorithm to solve the linear system, which is needed for implicit methods. Thereafter, the results are presented with graphs and benchmarks, followed by a discussion and critical evaluation of the FDM methods and their application to solving PDEs, and the heat equation in particular. The paper is ended with some brief concluding remarks.

All programmes of relevance to solving the heat equation with FDMs are available at `https://github.com/lasse-steinnes/FYS4150-Project5`.

II Theory and Methods

The theory section on FDM schemes is mainly based upon the book *Finite Difference Computing with PDEs* by Langtangen [2]. A description of the linear methods needed to solve the implicit FDMs can be found in Morten Hjorth-Jensen's lecture notes in the Computational Physics course (FYS4150) at the University of Oslo. The relevant themes are the Thomas algorithm to solve a tridiagonal matrix [3], and the Jacobi iterative method

applied to PDEs [4].

II.I The Heat Equation

The heat equation, also known as the diffusion equation, presents itself as

$$\alpha \nabla^2 u = \frac{\partial u}{\partial t}, \tag{1}$$

or in one dimension

$$\alpha \frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t}, \quad t > 0, \quad x \in [0, L]. \tag{2}$$

where $u = u(x,t)$ is the temperature as a function of spatial coordinate (x) and time (t). α is the thermal diffusivity and has units m^2/s. A large α implies a fast transfer of heat between regions of inverse temperature relation.

Note that eq. 2 can be interpreted as a scaled version of the heat equation, if $x \in [0,1]$ so that x/L is the spatial coordinate and $\alpha t/L^2$ is the new time coordinate. Thus the heat equation can be rewritten in a dimensionless form

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t}, \quad t > 0, \quad x \in [0, 1]. \tag{3}$$

To solve this problem, one initial condition (IC) is needed for the linear time dependence and two boundary conditions (BCs) are required for the second order derivative in space. In this paper, the IC considered is

$$u(x,0) = 0, \;\; 0 < x < 1, \tag{4}$$

whereas the BCs are

$$U(0,t) = 0 \quad \text{and} \quad u(1,t) = 1, \quad t \geq 0. \tag{5}$$

To solve eq. 3 analytically, the method of separation of variables is applied together with Fourier analysis. The steps is explained in detail in sec.VI.I. The final solution to eq. 3, with eqs. 4 and 5 as conditions, reads

$$u(x,t) = x + \frac{2}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^n}{n} \sin(n\pi x) e^{-(\pi n)^2 t}. \tag{6}$$

One can interpret eq. 6 as representing a stationary state for the system as $t \to \infty$. Then $u(x) = x$ is the solution of the Laplace equation $u_{xx} = 0$, because in this limit $\partial u/\partial t = 0$. A similar derivation is performed for the 2D case, which is shown in sec. VI.II, with the analytical solution

$$u(x,y,t) = sin(\pi x)sin(\pi y)e^{-2\pi^2 t}. \tag{7}$$

II.II Finite Difference Methods

FDMs are excellent tools to explore the dynamics of the heat equation away from the stationary state, since the analytical solution (eq. 6) then presents itself as an infinite sum. As mentioned in the introduction, one explicit method (FE) and two implicit methods (BE and CN) are utilized for this purpose. At the core of each algorithm is the choice of approximating the derivative when transforming a continuous equation to a finite mesh.

II.II.1 Forward Euler in 1D

The FE method uses a forward difference in time and a centered difference in space to approximate eq. 3. The heat equation is to be fullfilled at mesh points $t_n = n\Delta t \; \forall \; n = \{0,1,...,N_t\}$ in time and $x_i = i\Delta x \; \forall \; i = \{0,1,...,N_x\}$ in space. Here $\Delta x = 1/N_x$ and $\Delta t = T/N_t$, in which T is the chosen final time. The discretized version of eq. 3 then becomes

$$\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} = \frac{u_i^{n+1} - u_i^n}{\Delta t}. \tag{8}$$

Assuming the state of u being known for the previous time step, the scheme for the unknown u_i^{n+1} becomes

$$u_i^{n+1} = u_i^n + F\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right). \tag{9}$$

In the above finite difference scheme, $F = \Delta t/\Delta x^2$ is the Fourier number for FDMs approximating the heat equation. As elaborated upon in sec. II.III, it is essential in the analysis of the reliability of numerical method. Note that eq. 9 is solvable as long as an initial condition u_i^0 is known for all i. It is also important that BCs are enforced for $i = \{0, Nx\}$.

II.II.2 Backward Euler in 1D

Applying a backward difference in time and a central difference in space leads to the discrete approximation of the heat equation

$$\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} = \frac{u_i^n - u_i^{n-1}}{\Delta t}. \tag{10}$$

Eq. 10 is fullfilled at t_n as in the FE approximation. However, there are now three unknowns; u_{i-1}^n, u_i^n and u_{i+1}^n, and a linear system must be solved to achieve the next time step. Now, let $n \to n+1$ and $n-1 \to n$. The above equation can be rewritten to

$$u_i^{n+1}(1+2F) - F\left(u_{i+1}^{n+1} + u_{i-1}^{n+1}\right) = u_i^n. \tag{11}$$

The above scheme can be represented as a linear system $Au = b$ for each time iteration, where $A \in \mathbb{R}^{Nx-2 \times N_x-2}$ is a tridiagonal matrix and $u, b \in \mathbb{R}^{N_x-2}$. For simplicity, an equation with 2 boundary points and 3 internal points is shown below.

$$\begin{bmatrix} 1+2F & -F & 0 \\ -F & 1+2F & -F \\ 0 & -F & 1+2F \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \end{bmatrix} = \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \end{bmatrix} + F \begin{bmatrix} u_0^{n+1} \\ 0 \\ u_4^n + 1 \end{bmatrix}. \tag{12}$$

Remember that $F = \Delta t / \Delta x^2$.

II.II.3 Crank-Nicolson in 1D

A CN approximation to the derivatives in eq. 3 entails a centered difference in both space and time. In this case, the derivative is evaluated at $t_{n+1/2} = (n+1/2)\Delta t$, so it is a two-sided difference, compared to a one-sided difference (such as in FE and BE). However, $u(t_{n+1/2})$ must be approximated at mesh points t_n, t_{n+1} and so on. For this purpose, an arithmetic mean is chosen, i.e.

$$u_i(t_{n+1/2}) \approx \frac{1}{2}(u_i^n + u_i^{n+1}). \tag{13}$$

As a consequence, the discretized heat equation becomes

$$\frac{1}{2\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n + u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right) = \frac{u_i^{n+1} - u_i^n}{\Delta t}. \tag{14}$$

Rewriting with respect to the unknowns u_{i-1}^n, u_i^n and u_{i+1}^n and letting $\gamma = \frac{1}{2}F$ yields

$$-\gamma u_{i-1}^{n+1} + (1+2\gamma)u_i^{n+1} - \gamma u_{i+1}^{n+1} = \gamma u_{i-1}^n + (1-2\gamma)u_i^n + \gamma u_{i+1}^n. \tag{15}$$

A linear system $Au = b$ describes the equation. For simplicity, if only two boundary points and three internal mesh points is included in the spatial mesh, the system becomes

$$\begin{bmatrix} 1+2\gamma & -\gamma & 0 \\ -\gamma & 1+2\gamma & -\gamma \\ 0 & -\gamma & 1+2\gamma \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \end{bmatrix} = \begin{bmatrix} \gamma u_2^n + (1-2\gamma)u_1^n \\ \gamma u_3^n + (1-2\gamma)u_2^n + \gamma u_1^n \\ (1-2\gamma)u_3^n + \gamma u_2^n \end{bmatrix} + 2\gamma \begin{bmatrix} u_0^{n+1} \\ 0 \\ u_4^{n+1} \end{bmatrix}. \tag{16}$$

The last term comes from $u_0^n = u_0^{n+1}$ and $u_4^n = u_4^{n+1}$ at the boundaries.

II.II.4 2D Finite Differences

A Backward Euler method is chosen to approximate the time derivative for the 2D scheme. From this choice, one avoids the complications from having a two-sided difference (CN), but also gets a looser stability criteria compared to that in FE.

The discretized version of the 2D diffusion equation becomes

$$\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{\Delta x^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{\Delta y^2} = \frac{u_{i,j}^n - u_{i,j}^{n-1}}{\Delta t}, \tag{17}$$

where x_i and y_j are the spatial coordinates, and t_n is the time iteration. These, and the Δx and Δt are as described in the 1D case (sec. II.II.1). However, a simplification is made in the spatial mesh, so that $h = \Delta x = \Delta y$. Let $\alpha = dt/h^2$ From rewriting eq.17

$$(1+4\alpha)\, u_{i,j}^n - \alpha \left(u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n \right) = u_{i,j}^{n-1}. \tag{18}$$

Collecting known values at the right hand side, this becomes a linear system $Au = b$. Let $n \to n+1$ and $n-1 \to n$, to be concise in the notation. For simplicity, the linear system from a 4×4 mesh with 12 boundary nodes (mesh points) and 4 internal nodes is shown below

$$\begin{bmatrix} 1+4\alpha & -\alpha & -\alpha & 0 \\ -\alpha & 1+4\alpha & 0 & -\alpha \\ -\alpha & 0 & 1+4\alpha & -\alpha \\ 0 & -\alpha & -\alpha & 1+4\alpha \end{bmatrix} \begin{bmatrix} u_{11}^{n+1} \\ u_{12}^{n+1} \\ u_{21}^{n+1} \\ u_{22}^{n+1} \end{bmatrix} = \begin{bmatrix} \alpha \left(u_{01}^{n+1} + u_{10}^{n+1} \right) \\ \alpha \left(u_{13}^{n+1} + u_{02}^{n+1} \right) \\ \alpha \left(u_{31}^{n+1} + u_{20}^{n+1} \right) \\ \alpha \left(u_{32}^{n+1} + u_{23}^{n+1} \right) \end{bmatrix} + \begin{bmatrix} u_{11}^n \\ u_{12}^n \\ u_{21}^n \\ u_{22}^n \end{bmatrix}. \tag{19}$$

We Here, the solution of u_{ij}^{n+1} depends on unknown values $u_{i-1,j}, u_{i+1,j}, u_{i,j-1}, u_{i,j+1}$. Evident from eq.19, the system is not easily solved by Gauss elimination. Fortunately, the linear system is strictly diagonally dominant, i.e. $|A_{ii}| > \sum_{j \neq i} A_{ij}\ \forall\ i$ [4]. Thus, convergence of Jacobi's iterative method is ensured. Take an initial guess at $u^{n+1,0} = u^n$. Letting $u^{n+1,-} = u^-$ be the current guess, $u^{n+1} = u$ and $u^n = u^{(1)}$, eq. 18 becomes

$$u_{i,j} \approx \frac{1}{1+4\alpha} \left[\alpha \left(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} \right)^- + u_{i,j}^{(1)} \right]. \tag{20}$$

One then updates $u^- = u$ until a convergence criteria is met. The full algorithm is presented in sec. VI.III.

II.III Accuracy and Stability

II.III.1 Stability

Let

$$u_k = e^{-\alpha k^2 t} e^{ikx} = A e^{ikx}, \tag{21}$$

be a wave component, with A as a damping factor and $i = \sqrt{-1}$. Then the Fourier representation of a general solution of the heat equation is the linear combination

$$u(x,t) = \sum_{k \in K} b_k u_k, \tag{22}$$

where b_k is the coefficient of wave component k, and can be determined from the initial condition by Fourier analysis. Thus, a general scheme can be described by

$$u_q^n = A^n e^i kq\Delta x = A^n e^{ikx}, \tag{23}$$

where A is the amplification factor. Analytically it can be expressed as

$$A_e = e^{-\alpha k^2 \Delta t}. \tag{24}$$

However, for each numerical scheme, the amplification factor, A, will differ from A_e, and as can be deduced from eq. 23, $|A| \leq 1$ for the numerical scheme to remain within an upper or lower bound. As a consequence, each method has a stability criteria which must be met. The amplification factor and stability criteria is presented in tab. 1.

Table 1: **Amplification factor and stability for FDMs**: Amplification factors and stability criteria for the finite difference schemes Forward Euler (FE) (eq. 9), Backward Euler (BE) (eq. 11) and Crank-Nicolson (CN) (eq. 13). Stability implies the scheme has an upper or lower bound. Oscillations arise as a result of $A < 0$, so that the numerical scheme has non-physical oscillations. To get a stable, and non-oscillating numerical scheme, the criteria must be met. Parameters: A - amplification factor, $F = \Delta t/\Delta x^2$, $\rho = k\Delta x/2$, criteria - stability criteria.

	FE	BE	CN
A	$1 - 4F\sin^2\rho$	$(1 + 4F\sin^2\rho)^{-1}$	$(1 - 2F\sin^2\rho)/(1 + 2F\sin^2\rho)$
Criteria	F $\leq \frac{1}{2}$	Stable and non-oscillating	$F \leq \frac{1}{2}$

One can interprete the amplification factor as a function of $\rho = k\Delta x/2$. As presented in fig. 1, FE schemes which doesn't fulfill $F \leq 1/2$, will become particularly unstable if $|A| > 1$.

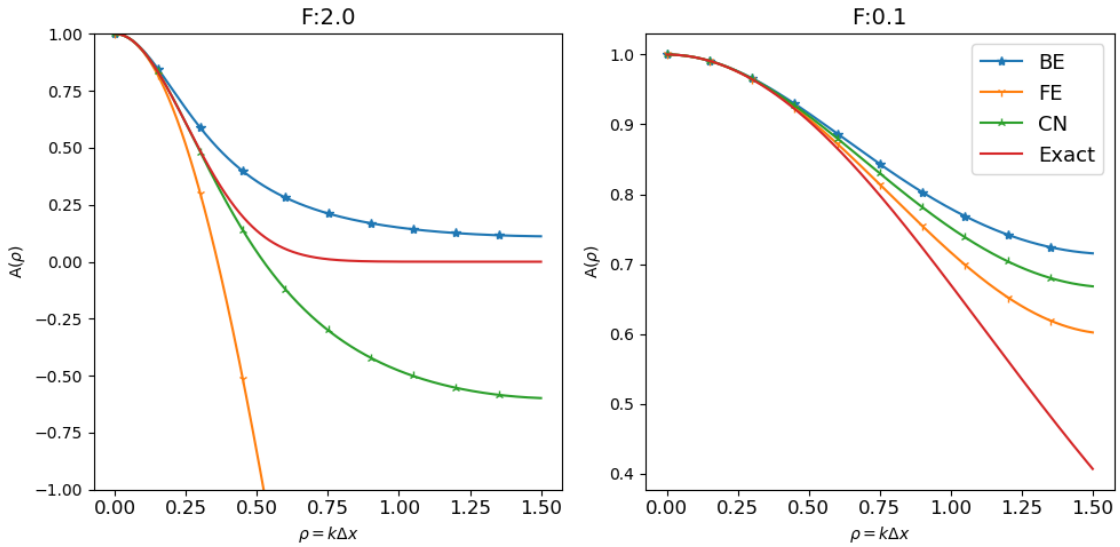


Figure 1: **Amplification factor analysis**: Amplification factor as a function of $\rho = k\Delta x$ for three 1D FDM schemes at $F = 2.0$ and $F = 0.1$. **Left)** When the Fourier number (F) doesn't fullfill $F \leq 1/2$, $A(\rho)$ for CN and FE becomes negative. In the FE scheme $|A| > 1$, and hence it will have a growing amplitude, whereas CN is only prone to non-physical oscillations ($-1 \leq A < 0$). **Right)** When F fullfills the stability criteria (tab. 1) $A(\rho) > 0$, all methods are stable. Scheme - colour: Exact -red, Forward Euler (FE) - orange, Backward Euler (BE) - blue, Crank-Nicolson (CN) - green.

II.III.2 Accuracy

The one-sided differences FE and BE has a truncation error $O(\Delta t)$ in time, and the centered difference in space has a truncation error $O(\Delta x^2)$. If the error is defined as $e_i^n = u_e(x_i, t_n) - u_i^n$, let the error norm at time t_n be defined as

$$E_{l_2} = \sqrt{h \sum_{i=1}^{Nx} (e_i^n)^2}, \tag{25}$$

where $h = \Delta x = C\Delta t$, for a constant C. For the FE and BE schemes, it follows from the truncation error that

$$E_{l_2} = C_t \Delta t + C_x \Delta x^2, \tag{26}$$

for some constants C_t and C_x. By defining $h = \Delta t$ and $\Delta x = K h^{1/2}$, $K \leq \sqrt{2}$ is defined so that the stability criteria is upheld (see tab. 1). Then $E \propto h$ for FE and BE, and is expected to converge with $O(h)$ accuracy. CN has a two-sided difference in time, hence its truncation error goes as $O(\Delta t^2)$ and $O(\Delta x^2)$, or

$$E_{l_2} = C_t \Delta t^2 + C_x \Delta x^2. \tag{27}$$

Thus if $h = \Delta x = C\Delta t$, it follows that CN converges with O(h^2) - one order higher than BE and FE.

II.IV Gauss Elimination and Iterative Strategies

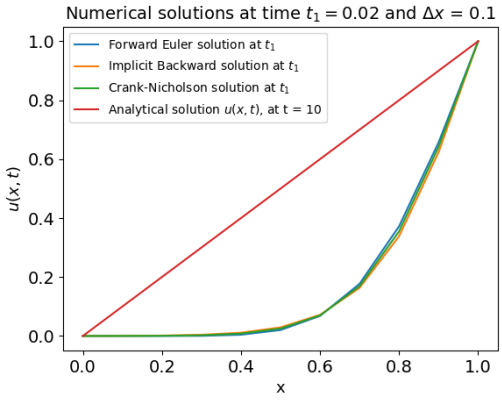
The Thomas algorithm is applied to solve the 1D implicit FDM schemes. It works for a tridiagonal matrix, reducing number of operations from $O(n^3)$ in Gauss elimination to $O(n)$ [3]. For a 2D BE FDM scheme, Jacobi's iterative method [4] is used to solve the linear system (algo. 1).

III Results

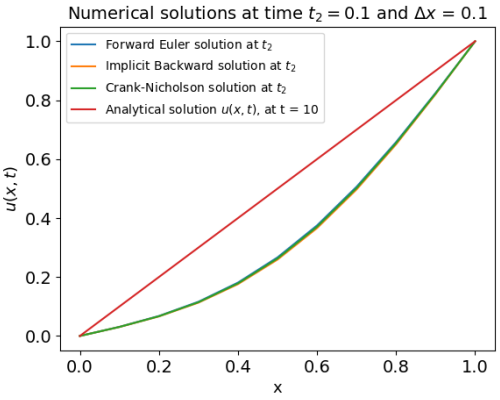
The results from applying Forward Euler (FE), Backward Euler (BE) and Crank-Nicolson (CN) time discretization to the diffusion equation are presented below. For the 1D case in sec. III.I, all methods are applied with different resolution in time (Δt) and spatial (Δx) step sizes. First an investigation of the impact of low resolution (fig. 2) and high resolution (fig. 3) on how the numerical solution approaches the stationary state $t \to \infty$. Thereafter, consequences of not satisfying the stability criteria are explored in fig. 4. Benchmarks of runtime (fig. 7) and error (fig. 8) can be found in the appendix (sec. VI.IV). FE has faster runtime and perform better than the two other methods for low resolution runs.

Numerical solution of the 2D diffusion equation from applying BE time discretization and the Jacobi iterative method, is presented in sec. III.II. Fig. 5 shows how the numerical solution approaches the analytical solution for different time and spatial resolutions, when the IC is a product of sine functions. Thereafter, a time series is explored for a problem with no-closed form solution (fig. 6). In this instance a Gaussian IC is used, with steeper slope than what is the case for the sine IC presented in fig. 5.

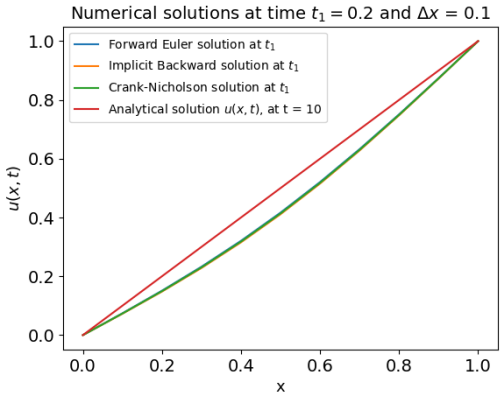
III.I 1D solutions for the various methods compared with analytic solution



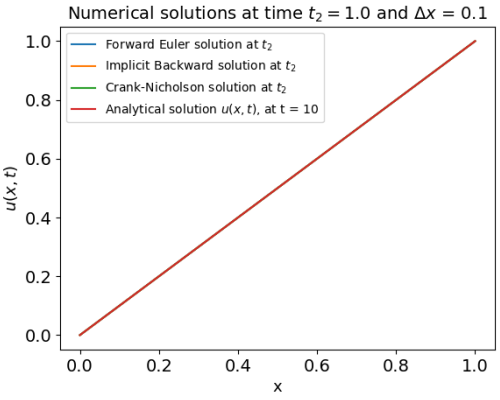
(a) $t = 0.02$, $\Delta x = 0.1$



(b) $t = 0.1$, $\Delta x = 0.1$

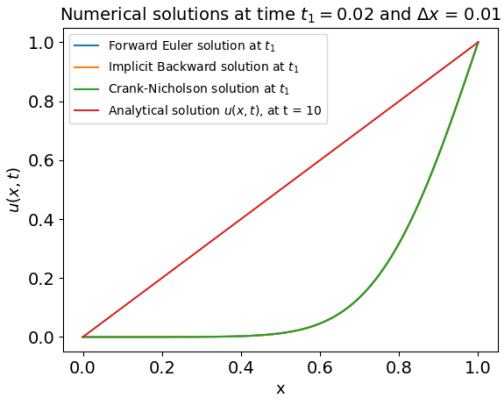


(c) $t = 0.2$, $\Delta x = 0.1$

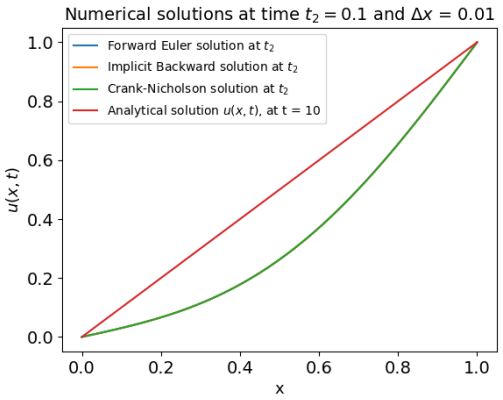


(d) $t = 1.0$, $\Delta x = 0.1$

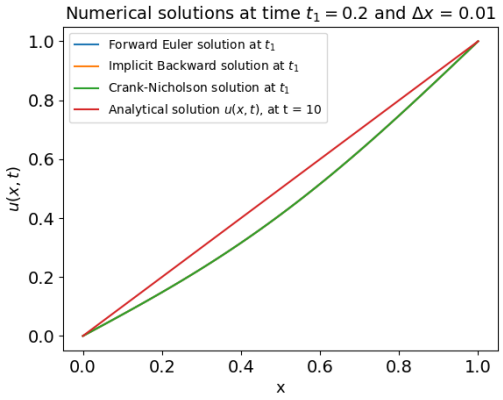
Figure 2: **Comparing 1D solutions for numerical methods with analytic solution using** $\Delta x = 0.1$: **Above**: The different figures shows how Forward Euler, implicit backward and Crank-Nicholson behave compared to the analytical 1D stationary state 6. For lower simulation times the numerical solutions differs from the stationary state and is curved. Due to lower resolution with $\Delta x = 0.1$, the curves appear less smooth and more angular. **a)** Shows the numerical solutions at $t = 0.02$ together with analytic solution $u(x,t)$ at $t = 10$. **b)** Numerical solutions at $t = 0.1$. **c)** Numerical solutions at $t = 0.2$. **d)** Again shows numerical solutions, now at the final time of the simulation with $t = 1.0$. We now see all numerical solutions in good agreement with the stationary state. Parameters: Total simulation time $T = 1.0$, steplength $\Delta x = 0.1$, timestep used $\Delta t = \frac{\Delta x^2}{3} \approx 0.0033$. Analytical stationary state $u(x,t)$ calculated for $t = 10$.



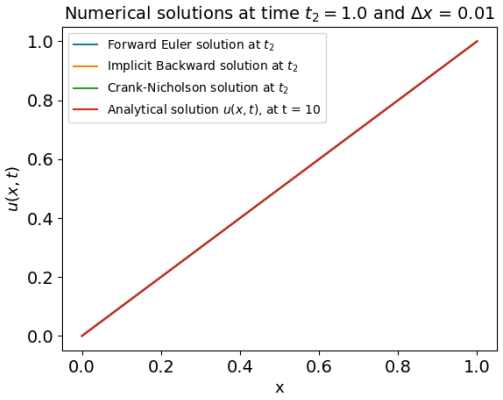
(a) $t = 0.02$, $\Delta x = 0.01$



(b) $t = 0.1$, $\Delta x = 0.01$

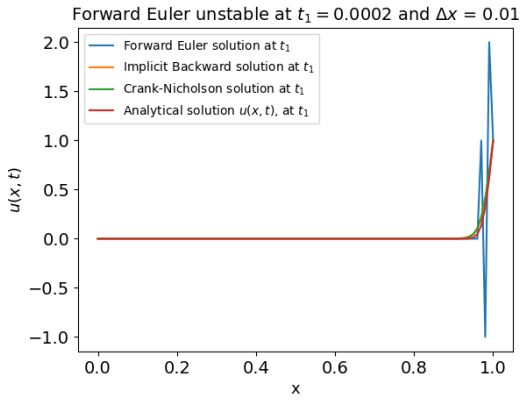


(c) $t = 0.2$, $\Delta x = 0.01$

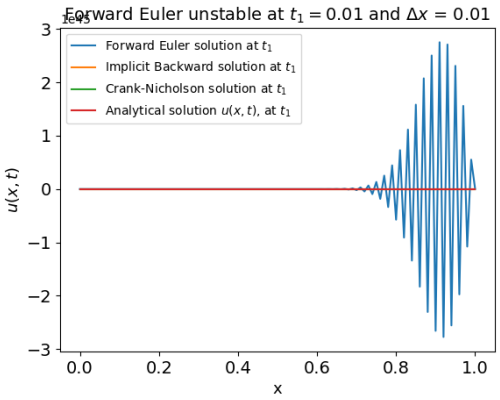


(d) $t = 1.0$, $\Delta x = 0.01$

Figure 3: **Comparing 1D solutions for numerical methods with analytic solution using** $\Delta x = 0.01$: **Above**: The different figures shows how Forward Euler, implicit backward and Crank-Nicholson behave compared to the analytical 1D stationary state 6. For lower simulation times the numerical solutions differs from the stationary state and appears more curved. The higher resolution for Δx makes the curves look more smooth overall. **a)** Shows the numerical solutions at $t = 0.02$ together with analytic solution $u(x,t)$ at $t = 10$. **b)** Numerical solutions at $t = 0.1$. **c)** Numerical solutions at $t = 0.2$. **d)** Again shows numerical solutions, now at the final time of the simulation with $t = 1.0$. We now see all numerical solutions in good agreement with the stationary state. Parameters: Total simulation time $T = 1.0$, steplength $\Delta x = 0.01$, timestep used $\Delta t = \frac{\Delta x^2}{3} \approx 0.00003$. Analytical stationary state $u(x,t)$ calculated for $t = 10$.



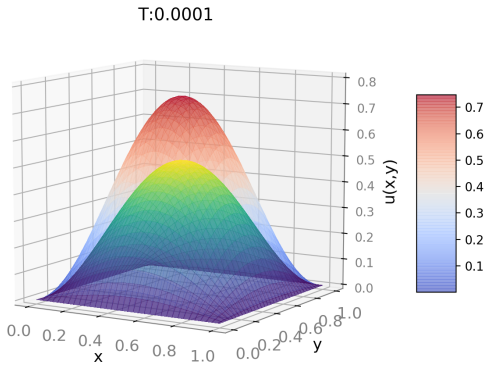
(a) $t = 0.0002$, $\Delta x = 0.01$



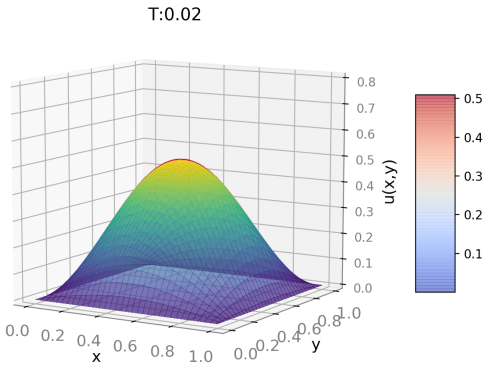
(b) $t = 0.01$, $\Delta x = 0.01$

Figure 4: **Unstability of Forward Euler when the stability criteria is not satisfied**: Both figures shows the behaviour of all three numerical methods when we use $\Delta t = \Delta x^2 > \frac{\Delta x^2}{2}$. This makes the Forward Euler method become unstable compared to Implicit backward and Crank-Nicholson. **a)** In the early stages of simulation for $t = 0.0002$. The Forward Euler solution starts to oscillate. Analytical solution calculated at $t = 0.0002$. **b)** At a later time $t = 0.01$, we have much bigger oscillations and forward euler is very unstable. Analytical solution calculated at $t = 0.01$. Parameters: Total simulation time $T = 0.1$, steplength $\Delta x = 0.01$, timestep $\Delta t = \Delta x^2 = 0.0001$.

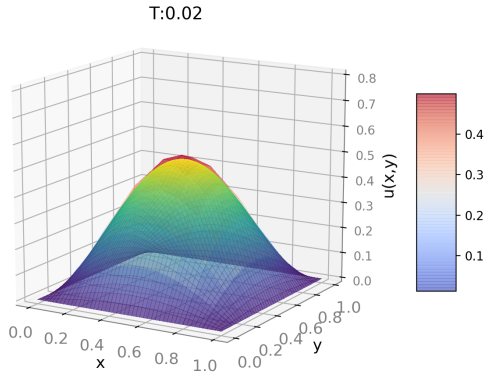
III.II 2D solutions from Jacobi's Iteration applied on BE



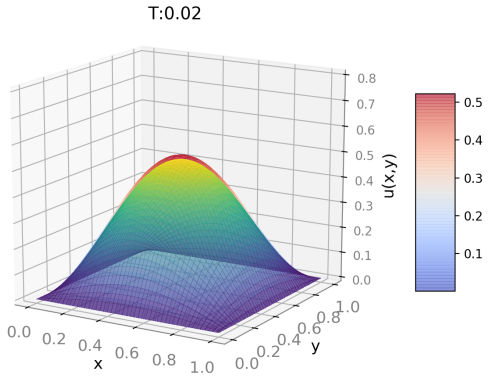
(a) $\Delta t = 0.0001$, $h = 0.01$



(b) $\Delta t = 0.0025$, $h = 0.01$

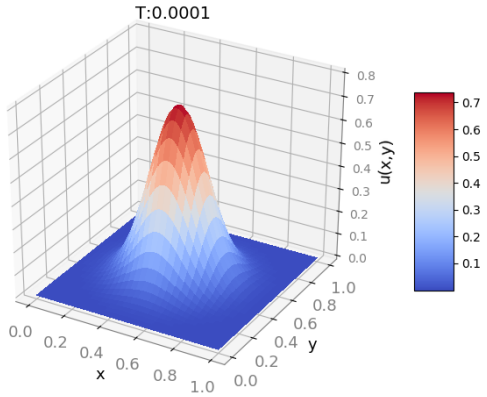


(c) Large h: $\Delta t = 0.01$, $h = 0.1$

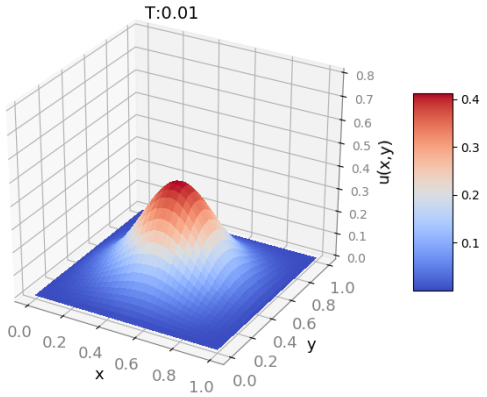


(d) Large Δt: $\Delta t = 0.01$, $h = 0.01$

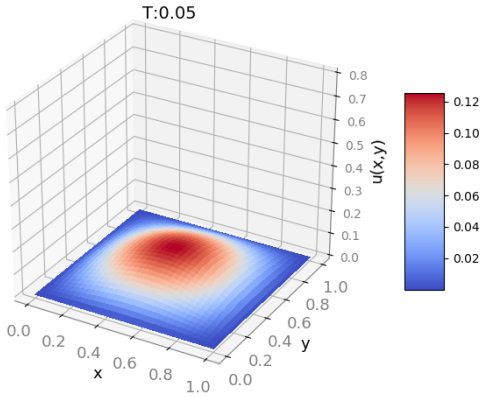
Figure 5: **Time series of the 2D numerical solution with sine initial condition**: The time development for a 2 dimensional diffusion equation are shown above. The blue and green surface is the analytical solution at $t = 0.02$. A 2D numerical BE sheme was solved iteratively at each time discrete level by applying Jacobi's iterative method. **a)** At first, the solution takes shape from the sine initial condition. **b)** The amplitude decreases quickly with time and approaches the shape and amplitude of the analytical solution at t = 0.02. The method is insensitive for the $\Delta t : h$ ratio, and the numerical solution still approximates the analytical solution at $t = 0.02$, as seen in **c)** (large h compares to Δt) and **d)** (large Δt - same magnitude as h). However, due to low resolution some accuracy is lost. With time, the numerical solution goes to 0, as expected (not shown here for the shake of brevity). Parameters: Δt - time step, $h = \Delta x = \Delta y$ - spatial step size/resolution, T - total simulation time. Number of Jacobi iterations: 10^3.



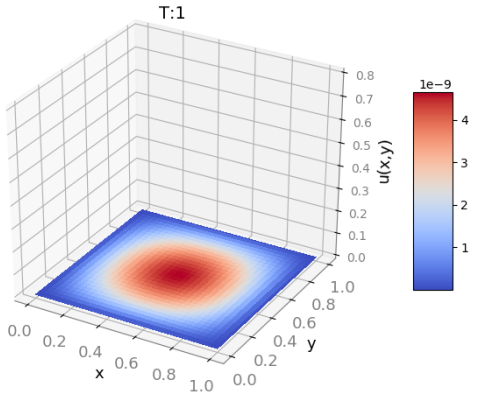
(a) $\Delta t = 0.0001$, $h = 0.01$



(b) $\Delta t = 0.005$, $h = 0.01$



(c) $\Delta t = 0.005$, $h = 0.01$



(d) $\Delta t = 0.01$, $h = 0.01$

Figure 6: **Time series of the 2D numerical solution with Gauss initial condition**: The time development for a 2 dimensional diffusion equation are shown above. A 2D numerical BE sheme was solved iteratively at each time discrete level by applying Jacobi's iterative method. **a)** At first, the solution takes shape as a Gauss curve, which is set as the initial condition. **b)**The amplitude decreases quickly with time and at $T = 0.01$ it is approximately halved. As seen in **c)** and **d)**, as heat is dissipated at the boundaries, after T = 0.05 the numerical solution approaches 0. Parameters: Δt - time step, $h = \Delta x = \Delta y$ - spatial step size/resolution, T - total simulation time. Number of Jacobi iterations: 10^3.

IV Discussion

As mentioned in the beginning of this article, FDMs are effective tools to investigate dynamical systems, especially in cases where no analytical solution exists. However, it is important to know of the numerical attributes, both the stability and accuracy, of the method at hand.

The results above (sec. III) are therefore discussed in the context of the expected behaviour of the diffusion equation, and the numerical artefacts of the applied discrete method is elaborated upon. In addition, the 2D solution with Gauss IC is discussed briefly as an example of a problem which is difficult to handle analytically, and how the IC affects propagation of numerical errors.

In the 1D-case, when the stability criteria is met (tab. 1), the numerical solutions (figs. 2 - 3) approaches the the stationary, linear state of the diffusion equation. However, with a low resolution in spatial step size ($\Delta x = 0.1$) and time $\Delta t = \Delta x^2/3$, FE in general lies above the other methods in amplitude ($u(x,t)$), followed by CN and BE (fig. 2). The spatial resolution contributes a less smooth solution compared to that in fig. 3 with a higher spatial and temporal resolution. However, the discrepancy in amplitude ($u(x,t)$) between FDMs in fig. 2 most likely is an effect of low temporal resolution. The FE discretization has a tendency to overestimate the time derivative, which is in concordance with Langtangen [2]. A surprising consequence is that the FE scheme has smaller supremum norm (error estimate) than the two other schemes (fig. 8), when the stability criteria is upheld. From the theory, one would expect CN to converge faster towards the analytical solution (O(h^2) [2]. This might be a combined consequence of error-accumulation in solving the linear system, and the zero IC and steep BCs.

With $F > 1/2$, the numerical stability criteria is not met (fig 4). In this case $F = 1$. The FE scheme has clear oscillations, and the amplitude is growing with time, which is exactly what would be expected when $A < -1$ (see tab. 1) and fig. 1) [2]. Some non-physical oscillations were also expected for the CN, scheme, however they are not severe enough to be visible in fig. 4. Despite being unstable for large $\Delta t : \Delta x$ ratio, the FE scheme provides certain benefits, as having a lower computational cost than both the BE and CN methods [2][4], and is thus faster than the latter methods (fig. 7) .

In the 2D-case, the BE scheme correctly reproduces the analytical solution (sine IC) at time $t = 0.02$ (fig.5). The accuracy increases with higher spatial (Δx) and temporal (Δt) resolution. Moreover, the method is stable for lower resolutions, even when the stability criteria is not met. This an important feature of the BE-scheme. However, it is worth noting that lower resolution comes at the cost of reduces accuracy, as seen in fig. 5. Yet stable, the BE-scheme is computationally costly, and time-inefficient for large time-spans

and small Δt's, since the scheme is solved at every time-discrete level with the Jacobi-algorithm. Thus, if a high resolution in Δt is desired, FE is the preferred method, because it is safe to use for low $\Delta t : \Delta x$ ratios, and more efficient.

It is evident from fig. 6 that the 2D BE scheme can easily deal with problem's which is difficult to handle analytically. Here a Gaussian IC is taken as input, with a sharper curve compared to that in fig. 5. Visually, the numerical solution displays the expected behaviour, namely heat loss (or diffusion) to and along the boundaries, and $u(x,y,t) \to 0$ at $t \to \infty$. This at the core of the diffusion process. However, it is dangerous to believe all initial conditions will give a numerical solution with physical properties. This is not the case with many sharp and ICs with non-continuous derivatives, where numerical errors might accumulate for each time propagation, resulting in non-physical attributes in the discrete solution [2].

V Conclusion

In this paper, the machinery and numerical artefacts of finite difference methods is explored and discussed in the context of the diffusion equation. In 1D, the Forward Euler, Backward Euler and Crank-Nicolson schemes display different numerical attributes. The methods approximate the time derivatives differently, and as a result, the FE scheme seem to approximate the analytical solution at lower numerical error and faster runtimes. However, if the the numerical stability criteria is not satisfied (tab. 1), only BE and CN approach the staionary state with time. If not met, the FE numerical method begins to oscillate, with a growing amplitude in time.

The BE scheme with Jacobi's iterative method is applied to the 2D equation. The BE scheme reproduces the qualitative behaviour of a known analytical solution, and it is always stable. Nevertheless, some accuracy is lost when a low temporal and spatial resolution is applied. The 2D BE iterative solver also handles sharper IC curves well, in this case a Gaussian function.

We have seen that the FE method is the most unstable, and the numerical solution it provides may have non-physical attributes. Thus numerical methods should always be bench-marked with known analytical solutions, before venturing into solving problems with unknown solutions. It is worth noting that the simpler FE method is less computationally costly, and thus provides lower runtime. If a high resolution in Δt compared to the spatial resolution (eg. Δx, Δy) is needed, then an efficient FDM scheme might be the optimal choice, compared to methods relying on solving large linear systems. Hence, in choice of method, the trade-offs between stability, accuracy and computational cost should always be taken into consideration.

References

[1] Haïm Brezis and Felix Browder. Partial differential equations in the 20th century. *Advances in Mathematics*, pages 76–144, 1998.

[2] Hans Petter Langtangen; Svein Linge. *Finite Difference Computing with PDEs: A Modern Software Approach*. SpringerOpen, 2017.

[3] Morten Hjorth-Jensen. Computational Physics Lectures: Linear Algebra methods, 2020. Accessed: November 2020.

[4] Morten Hjorth-Jensen. Computational Physics Lectures: Partial Differential Equations, 2020. Accessed: December 2020.

VI Appendix

VI.I Analytical Solution of the 1D Scaled Heat Equation

The one-dimensional diffusion is

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} \tag{28}$$

which we will solve analytically on the interval x ∈ [0,1] for t > 0 with the boundary conditions u(0,t) = 0, u(1,t) = 1 and the initial condition u(x,0) = 0. We can think of u(x, t) as the temperature distribution along a rod of length 1 with a constant heat source at one of the end points. The temperature distribution will reach a steady state after some amount of time and we try finding a solution on the form of a sum of two solutions

$$u(x,t) = x + f(x,t) \tag{29}$$

where $f(x,t)$ has to satisfy $f(0,t) = f(1,t) = 0$ and $f(x) \equiv f(x,0) = -x$. We can find $f(x,t)$ by using the method of separation of variables and assume that it can be written on the form $f(x,t) = F(x)G(t)$. Inserting it into eq.(28) we obtain

$$F\frac{dG}{dT} = G\frac{d^2F}{dx^2}$$

rewritten as

$$\frac{1}{G}\frac{dG}{dT} = \frac{1}{F}\frac{d^2F}{dx^2} \tag{30}$$

Since the left side is a function of t alone and the right side is a function of x alone, both sides of eq.(30) must be a constant. It's convenient to call this constant $-k^2$. We can then split eq.(30) into two ordinary differential equations and obtain the respective general solutions

$$\frac{dG}{dT} = -k^2 G \rightarrow G(t) = e^{-k^2 t}$$

$$\frac{d^2F}{dx^2} = -k^2 F \rightarrow F(x) = A\sin(kx) + B\cos(kx)$$

Choosing a constant on the form $-k^2$ ensures that there is no time dependence as $t \rightarrow \infty$. To find the solution $f(x,t) = F(x)G(t)$ which satisfies $f(0,t) = f(1,t) = 0$ it is convenient to set $t = 0$ so that $G = 1$ and $F(0) = F(1) = 0$. From the first boundary condition we get

$$Asin(0) + Bcos(0) = A0 + B1 = B = 0$$

And from $F(1) = 0$ we get

$$Asin(k) = 0 \rightarrow k = n\pi$$

A solution $f(x,t)$ which satisfies $f(0,t) = f(1,t) = 0$ is thus on the form

$$f(x,t) = Asin(n\pi x)e^{-t(n\pi)^2}$$

but it does not satisfy $f(x,0) = f(x) = -x$ for any n. We use the linearity of the diffusion equation and write $f(x,t)$ as

$$f(x,t) = \sum_{n=1}^{\infty} A_n sin(n\pi x)e^{-t(n\pi)^2} \tag{31}$$

which is a Fourier series. The functions $sin(n\pi x)$ are orthogonal with respect to the inner product

$$\delta_{nm} = 2\int_0^1 sin(m\pi x)sin(n\pi x)dx$$

If we set t = 0 in eq.(31) and "multiply" both sides by $2\int_0^1 sin(m\pi x)dx$ we obtain

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

The analytical solution can thus be written as

$$u(x,t) = x + \frac{2}{\pi}\sum_{n=1}^{\infty} \frac{(-1)^n}{n} sin(n\pi x)e^{-t(n\pi)^2} \tag{32}$$

VI.II Analytical Solution of the 2D Scaled Heat Equation

The two-dimensional diffusion equation is

$$\frac{\partial u(x,y,t)}{\partial t} = \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} \tag{33}$$

which for simplicity will be solved on the area x,y $\in$ [0,1] with the boundary conditions u | x=0 = u | y=0 = u | x=1 = u | y=1 = 0 and with initial condition $f(x,y) = u(x,y,0)$. Again

we use the method of separation of variables and assume a solution on the form $u(x,y,t) = X(x)Y(y)T(t)$. Substituting this into eq.(33) we obtain

$$XY\frac{dT}{dt} = YT\frac{d^2X}{dx^2} + XT\frac{d^2Y}{dy^2}$$

Divide both sides by XYT

$$\frac{1}{T}\frac{dT}{dt} = \frac{1}{X}\frac{d^2X}{dx^2} + \frac{1}{Y}\frac{d^2Y}{dy^2} \tag{34}$$

where the left side is a function of t alone and the right side is a function of position alone. So both sides are equal to some constant $-k^2$. But we could move the t-term over to the right and either the x-or y-term over to the left, in which case the left side would be a function of either x or y alone. So all terms in eq.(34) must be constants.

Let the x-term be $-p^2$ and the y-term be $-q^2$ where $p^2 + q^2 = k^2$. Then we get the three ordinary differential equations

$$\frac{d^2X}{dx^2} = -p^2X \rightarrow X(x) = Asin(px) + Bcos(px)$$
$$\frac{d^2Y}{dy^2} = -q^2Y \rightarrow Y(y) = Csin(qy) + Dcos(qy)$$
$$\frac{dT}{dt} = -k^2T \rightarrow T(t) = e^{-k^2t}$$

where the coefficient in front of the exponential in $T(t)$ has been absorbed into $X(x)Y(y)$.

Analogous to the one-dimensional case, the boundary conditions are satisfied by choosing B = D = 0 and p = mπ, q = nπ for some positive integers m and n. Again by exploiting the linearity of the diffusion equation we get

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

Setting t = 0 and multiplying both sides by $4\int_0^1\int_0^1 sin(n'\pi x)sin(m'\pi x)dxdy$ we get

$$A_{nm} = 4\int_0^1\int_0^1 f(x,y)sin(n\pi x)sin(m\pi x)dxdy$$

In particular, if f(x,y) = sin(πx) sin(πy) only $A_{1,1} = 1$ is nonzero. In this case the analytical solution is

$$u(x,y,t) = sin(\pi x)sin(\pi y)e^{-2\pi^2 t} \tag{35}$$

where the factor of 2 in the exponential comes from $(m^2 + n^2 = 1^2 + 1^2 = 2)$.

VI.III Jacobi's Iterative Method

Algorithm 1 Jacobi Iterative Method: The basic outline of the Jacobi Iterative Method for solving the 2D discrete version of the heat equation with Backward Euler time discretization. The system diagonally dominant, and thus converges to the actual solution [4]. Nx, Ny: Number of intervals between spatial mesh points. Note that u_{temp} is the guess, and u is the solution at time t_n, u_n at t_{n-1}. They are here written as flattened matrices

iterations = 0
while iterations <= max iterations **do**
  for int i= 1; c < Nx; i++ **do** ▷ For each x inner mesh point
    for int j=1; i < Ny ; j++ **do** ▷ For each y inner mesh point
      $delta_{ij} = u_{temp}((j+1) \times m_k + i) +$
      $u_{temp}((j-1)m_k + i) + u_{temp}(j \times m_k + (i-1)) + u_{temp}(j \times m_k + (i+1));$
      $u(j \times m_k + i) = 1/(1 + 4 \times m_s) \times (m_s \times delta_{ij} + u_n(j \times m_k + i));$
    end for
  end for
  iterations ++
end while ▷ Update time scheme

VI.IV Error and time performance

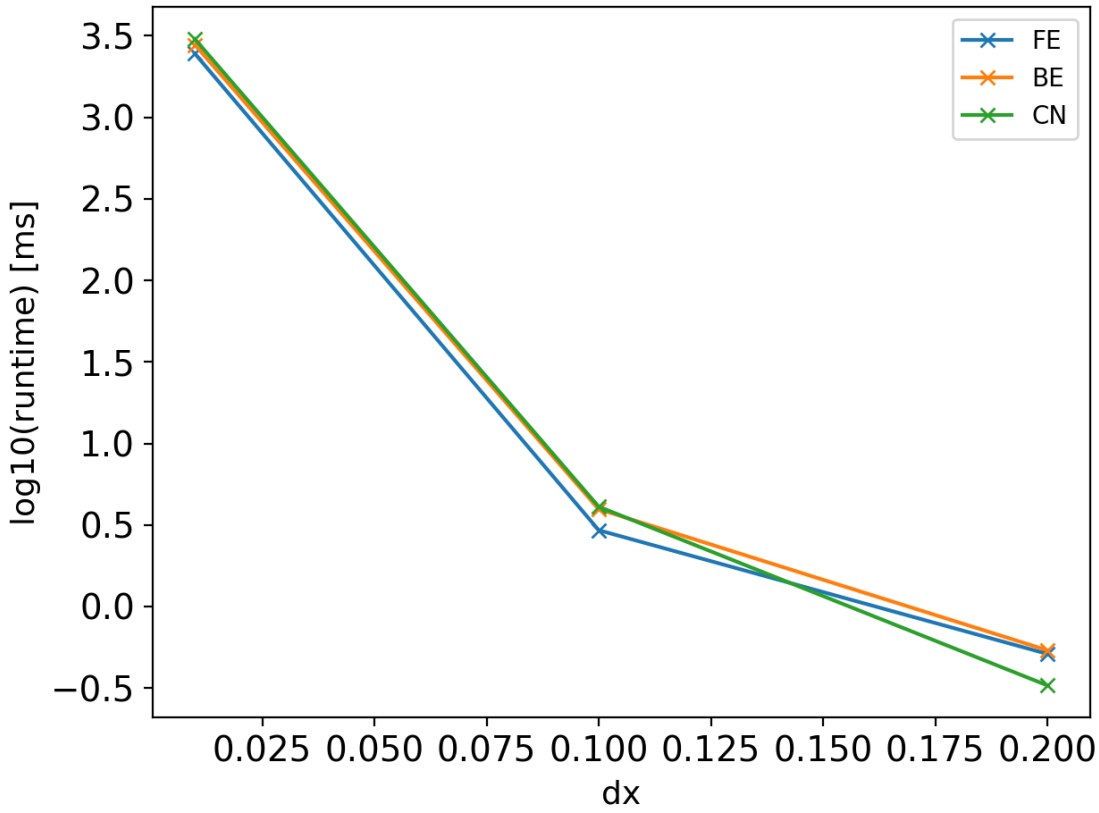


Figure 7: **Time performance of three FDMs in 1D case:** Time performance in milliseconds of the Forward Euler (FE), Backward Euler (BE) and Crank-Nicolson (CN) schemes, as a function of Δx. $\Delta t = \Delta x^2/3$. Total simulation time $T = 1$. The resolution affects CN the most, followed by BE and FE respectively. High resolution (small dx) takes longer time to finish.

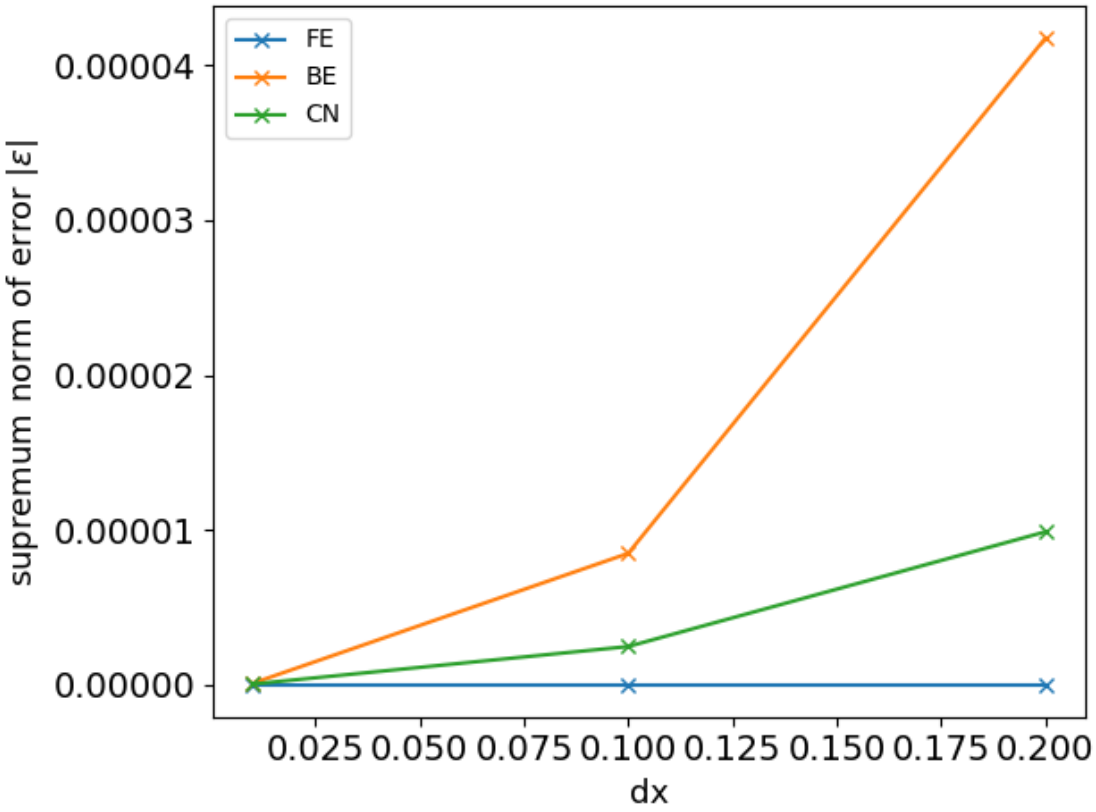


Figure 8: **Error analysis of three FDMs in 1D case:** Supremum norm ($\varepsilon = max|u_{num} - u_{exact}|$) of the Forward Euler (FE), Backward Euler (BE) and Crank-Nicolson (CN) schemes, as a function of Δx. $\Delta t = \Delta x^2/3$. Total simulation time $T = 1$, which is the time which the analytical solution is evaluated at. FE proves to be the most accurate method, whereas CN and BE seem to approach the analytical solution more slowly.