

MT5846: Advanced Computational Techniques

Project One

Name: Cameron Sander
Matriculation number: 210005973

Date Due: 4th March 2024

Abstract

This paper analyses different numerical schemes to simulate fluid flow between parallel plates. The primary focus is examining the characteristics and stability conditions for four methods; including the Forward Time Central Space (FTCS), DuFort-Frankel, Laasonen, and Crank-Nicolson methods used to solve the partial differential equations governing the flow. The paper evaluates the efficacy of these methods in capturing the fluid's behavior accurately. Investigation reveals the conditional stability of the FTCS method and the unconditional stability of the DuFort-Frankel, Laasonen, and Crank-Nicolson methods.

Through the implementation of the schemes to solve the PDE, it is demonstrated how the numerical methods influence the velocity profiles of the fluid, with particular focus on varying time step sizes. It is observed that the solutions found using the FTCS method diverge significantly from the true solution when its stability condition is not met. In addition to this, the DuFort-Frankel method, displays oscillatory behaviour causing the scheme to be less accurate than expected due to the explicit nature of the method, making it prone to numerical diffusion.

In addition to this investigations were conducted to show how the errors of the Laasonen method varied as different values of time steps for the simulations were used. It was shown that the method exhibited behaviour predicted based on the order of the method being one in time and two in space.

The Crank-Nicolson method is selected for its superior accuracy and stability, albeit at a higher computational cost. This paper provides insights into selecting the most appropriate numerical scheme for the context of the problem, balancing the trade-offs between computational demands and the precision of results.

Contents

1	Introduction	3
2	Model	5
2.1	The Problem	5
2.2	Approach	5
3	Numerical approach	7
3.1	Problem Setup	7
3.2	Forward Time Central Space Method	7
3.2.1	Stability Analysis	8
3.3	DuFort-Frankel Method	9
3.3.1	Stability Analysis	9
3.4	Laasonen Method	11
3.4.1	Stability Analysis	11
3.5	Crank-Nicolson Method	12
3.5.1	Stability Analysis	12
3.6	Stability Analysis Results	13
4	Results	14
4.1	Simulations	14
4.1.1	Forward Time Central Space	14
4.1.2	DuFort-Frankel Method	15
4.1.3	Laasonen Method	15
4.1.4	Crank-Nicolson Method	16
4.2	Errors	17
4.2.1	Forward Time Central Space	18
4.2.2	DuFort-Frankel Method	19
4.2.3	Laasonen Method	21
4.2.4	Crank-Nicolson Method	22
4.3	Comparing Error Profiles	23
4.4	Time step in the Laasonen method	24
5	Discussion and Conclusion	25

1 Introduction

The problem being investigated in this paper is governed by an equation:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2}$$

this is a variation of a very well known equation called the heat equation which models heat conduction as shown here:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

where:

- $u(x, t)$ represents the temperature at position x and time t ,
- α is the thermal diffusivity of the material, which characterizes the rate at which heat diffuses through the material,
- $\frac{\partial u}{\partial t}$ is the partial derivative of the temperature with respect to time, indicating how the temperature changes over time,
- $\frac{\partial^2 u}{\partial x^2}$ is the second partial derivative of the temperature with respect to space, representing the curvature of the temperature profile in space.

This is an example of a Partial Differential Equation commonly abbreviated to PDE. These are mathematical equations that involve functions of several variables and their partial derivatives. These differ from Ordinary Differential Equation (ODEs), which involve derivatives with respect to a single variable. PDEs involve derivatives with respect to multiple spatial and/or temporal variables. In the case of the heat equation, the PDE contains one second order spatial derivative being the $\frac{\partial^2 u}{\partial x^2}$ and one first order temporal derivative being $\frac{\partial u}{\partial t}$. PDEs are used in expressing and solving the dynamics of systems in physics, engineering, and many other disciplines. They allow us to model a wide range of phenomena, including heat conduction (seen above), sound propagation, fluid flow, electromagnetism, and the behavior of financial instruments.

”PDEs are the language of dynamic phenomena. They allow us to articulate the laws of physics, the patterns of engineering, and the rhythms of nature. Their solutions bridge the gap between abstract mathematical conjectures and the tangible physical universe, providing insights into the complexities of the world around us.” [1]

As seen from this quote, the importance of PDEs cannot be understated and their relevance in the functioning of a modern society. There are several classifications of PDEs based on the nature of their partial derivatives. The system we will be investigating is classified as a linear, second-order, parabolic PDE. Parabolic PDEs describe the evolution over time, They exhibit a smoothing effect, where conditions that appear sharp and abrupt become smoother over time. It is second-order since it involves second partial derivatives. The order of a PDE is determined by the highest derivative of the unknown function that appears in the equation. Finally it is linear meaning that any linear combination of solutions is also a solution.

To classify the heat equation as a parabolic PDE it must be expressed as a general second-order PDE:

$$Au_{xx} + 2Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu + G = 0$$

where u is the unknown function of x and y , and A, B, C, D, E, F , and G are coefficients that can depend on x, y , and u .

Having done this the type depends on the discriminant of the quadratic form of the highest-order derivatives, given by $B^2 - AC$:

- **Elliptic** if $B^2 - AC < 0$
- **Parabolic** if $B^2 - AC = 0$
- **Hyperbolic** if $B^2 - AC > 0$

In our case:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Comparing this with the general form:

$$-\alpha u_{xx} + u_t = 0$$

where $A = -\alpha$, $B = 0$, and $C = 0$. Thus, the discriminant $B^2 - AC = 0$, classifying the heat equation as **parabolic**.

To solve this PDE, a variety of Finite Difference Methods (FDM) were employed. These methods work by the process of democratization of the continuous domain (space and time) into a grid of discrete points. Having done this a variety of combinations of finite difference methods are used on both the temporal and spatial points to approximate the partial derivatives. These methods can be broken up into two classes of implicit and explicit. These methods differ in how they approximate the derivatives and handle time stepping, leading to distinct characteristics in terms of stability, accuracy, and computational effort. The characteristics of these methods will be outlined. With explicit methods, the next time step of the system is calculated directly from the known values meaning it is relatively simple and computationally less demanding. The major drawback of these methods is that their stability is conditional on the time step size and the spatial grid size used. This culminates in a stability criterion (such as the Courant-Friedrichs-Lowy (CFL) condition for the heat equation), to prevent numerical instabilities as will be discussed later in this paper. On the other hand with implicit methods the future state is computed in a way that involves the future values themselves, resulting in a system of algebraic equations that must be solved at each time step, increasing computational cost. A key advantage of implicit methods is their unconditional stability for linear problems, allowing for much larger time steps, compared to explicit methods.

A note here a method investigated further later on the Crank-Nicolson method offers a compromise between explicit and implicit schemes. It averages the explicit and implicit forms, providing a method that is second-order accurate in both time and space, and unconditionally stable for linear problems.

Although they are not discussed in this paper, it is worth to mention that there are alternative numerical methods to solving PDEs such as Finite Element Methods, Finite Volume Methods, Spectral Methods or Meshless Methods. Each class having different benefits and drawbacks to their methodology.

2 Model

In this section, we will be describing the problem and its corresponding PDE (Partial Differential Equations) which we will be investigating for the remainder of this research paper.

2.1 The Problem

We consider a viscous fluid, between two parallel plates, which are a distance of y apart. These plates are infinitely long, hence the ends of them do not have any effect of the physical process of the fluid between them. Initially both plates are stationary. When the experiment is begun, the lower plate, at $y = 0$, is instantly accelerated to a velocity of U_0 in the x direction, whilst the other plate remains stationary. The movement causes the velocity of the fluid to change as time progresses. This velocity u can be modelled using the equation:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2}, \quad (1)$$

where ν is the kinematic viscosity of the fluid. The velocity profile $u(t, y)$ is required. To illustrate the setup of this problem the Figure 1 shows the computational grid and the plates for this simulation.

2.2 Approach

The PDE described in the above problem is known as the Heat Equation. It is usually used to describe the distribution of heat (or variation in temperature) in a given region over time. However, in this problem it is being used as a way to model the velocity profile of the fluid being investigated. In addition to this, the commonly used α which describes the thermal diffusivity constant has been replaced in this setting with ν (constant) representing the kinematic viscosity of the fluid.

Note: Although the spatial axis used in the diagram to represent the distance of the plates is using the letter j , for the remainder of this paper the notation for this has been changed to i . The initial conditions have been changed to reflect this change.

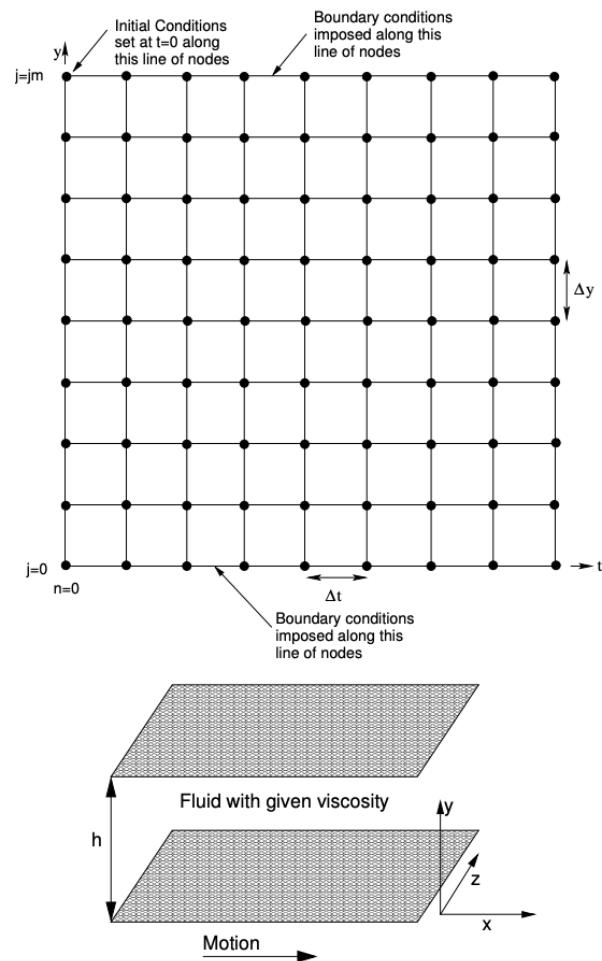


Figure 1: Parallel Plate Simulation

3 Numerical approach

In this section, a description of the numerical approaches being used will be given as well as a description of the initial and boundary conditions.

3.1 Problem Setup

- Defined Values

$$\begin{aligned} h &= 0.04\text{m} & im &= 40 \\ \nu &= 0.000217 \text{ m}^2/\text{s} & \Delta y &= 0.001\text{m} \\ U_0 &= 40 \text{ m/s} & t &= 1.08 \text{ s} \end{aligned}$$

- Initial Condition

$$\begin{aligned} t = 0 && u = U_0 && \text{for } y = 0 \\ t = 0 && u = 0 && \text{for } 0 < y \leq h \end{aligned}$$

- Boundary Conditions

$$\begin{aligned} t \geq 0 && u = U_0 && \text{for } y = 0 \\ t \geq 0 && u = 0 && \text{for } y = h \end{aligned}$$

3.2 Forward Time Central Space Method

The first method being used to analytically solve this PDE is the FTCS method, this is an example of a type of finite difference method, where derivatives in the continuous equations are approximated using differences between discrete points in a grid. In particular the time derivative is approximated using forward difference and the space derivative is approximated using the central difference approximation. Substituting these two finite difference methods into the PDE (1):

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \nu \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2}, \quad 0 < i < im \quad (2)$$

Rearranging the terms and using the substitution $d = \frac{\nu \Delta t}{(\Delta x)^2}$ we arrive at the equation

$$u_i^{n+1} = u_i^n + d(u_{i+1}^n - 2u_i^n + u_{i-1}^n), \quad 0 < i < im \quad (3)$$

As we can see from the equation above, since all the terms on the RHS of the equation are dependent on the current time step n to calculate the next time step $n + 1$ on the LHS, this is an example of an explicit method. This has impacts on the stability of the method which will be discussed further in the Stability Analysis section. The order of this scheme can be deduced from the fact that the temporal approximation (forward difference) was made using a 1st order scheme i.e $O(\Delta t)$ and the spatial approximation (centered difference) was made using a 2nd order scheme i.e $O((\Delta x)^2)$. Combining these two together we can note the order of the FTCS is $O(\Delta t, (\Delta x)^2)$ meaning that the method is order one in time and order two in space.

To visualise how this method uses the surrounding spacial and temporal points, the figure 2 shows a stencil of the **FTCS** scheme. The three filled black dots represent the three spatial points at time n being used to approximate the value of the not filled dot representing the approximation at time $n + 1$. This stencil also provides an understanding of why points on the boundary cannot be approximated, since they would rely on points that lay outside of the region specified in the problem.

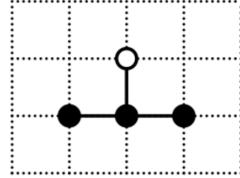


Figure 2: FTCS Stencil

3.2.1 Stability Analysis

It can be shown that the round-off error ϵ^n evolves in time like the finite difference method. We set $u_i^n \rightarrow \epsilon_i^n := \xi^n e^{I\theta i}$ where I is the imaginary unit and $\theta \in [-\pi, \pi]$. Substituting these values into Equation 3 we obtain:

$$\xi^{n+1} e^{I\theta i} = \xi^n e^{I\theta i} + d(\xi^n e^{I\theta(i+1)} - 2\xi^n e^{I\theta i} + \xi^n e^{I\theta(i-1)})$$

Simplifying this equation by dividing through by $e^{I\theta i}$:

$$\xi^{n+1} = \xi^n [1 + d(e^{I\theta} + e^{-I\theta} - 2)]$$

Using the relation $\cos \theta = \frac{e^{I\theta} + e^{-I\theta}}{2}$ to obtain:

$$G = \frac{\xi^{n+1}}{\xi^n} = 1 - 2d(1 - \cos \theta)$$

G : Amplification Factor, ξ^n is not growing when $G \leq 1 \iff |1 - 2d(1 - \cos \theta)| \leq 1$ resulting in the cases:

$$1 - 2d(1 - \cos \theta) \leq 1, \quad 1 - 2d(1 - \cos \theta) \geq -1$$

The first of which is always true since $1 - \cos \theta \geq 0$ and $d = \frac{\nu \Delta t}{(\Delta x)^2} > 0$, simplifying the second inequality we obtain:

$$d(1 - \cos \theta) \leq 1$$

since $\cos \theta \leq 1$ we can finally deduce:

$$d \leq \frac{1}{2} \quad \text{or} \quad \frac{\nu \Delta t}{(\Delta x)^2} \leq \frac{1}{2} \tag{4}$$

As a result of this analysis the FTCS method is **conditionally stable** in the Von-Neumann sense on equation (4).

3.3 DuFort-Frankel Method

The next method that we will discuss is another explicit method, this method again uses the central space approximation similar to the **FTCS** however, for the time approximation instead of a forward method as before, a centered approximation is used. In addition this, in the approximation of the second order spatial derivative, u_i^n is replaced by the average of u_i^{n+1} and u_i^{n-1} .

Note: This final adjustment is done since without it this method would be the Richardson Method, which can be shown to be unconditionally unstable and hence have no practical use in solving the PDE.

$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \nu \left(\frac{u_{i+1}^n - 2\frac{u_i^{n+1} + u_i^{n-1}}{2} + u_{i-1}^n}{(\Delta x)^2} \right) , 0 < i < im \quad (5)$$

Solve with respect to u_i^{n+1} :

$$u_i^{n+1} = \frac{2d(u_{i+1}^n + u_{i-1}^n) + (1 - 2d)u_i^{n-1}}{1 + 2d} , 0 < i < im \quad (6)$$

This important differences between this method and the **FTCS** are the fact that this method is of order $O((\Delta t)^2, (\Delta x)^2)$ because of the fact that a central time approximation was used for this method. In addition, it is a two-step method in time. i.e., we require u_i^1 in addition to u_i^0 to start computing u_i^n . The **FTCS** method (one step in time) was used to compute u_i^1 since its local error is of second order, meaning taking only one step using this method does not compromise the 2nd order characteristics of the DuFort-Frankel Method. The stencil shown in figure 3 for this scheme shows how points from previous time steps and spatial positions are used to calculate the next approximation.

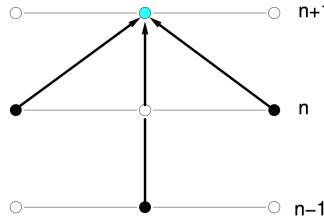


Figure 3: DuFort-Frankel Stencil

3.3.1 Stability Analysis

Making similar substitutions for those made for the stability analysis for FTCS we can obtain:

$$(1 + 2d)\xi^{n+1}e^{I\theta i} = (1 - 2d)\xi^{n-1}e^{I\theta i} + 2d(e^{I\theta(i+1)} + e^{I\theta(i-1)})\xi^n$$

Dividing through by $\xi^{n-1}e^{I\theta i}$ we obtain:

$$(1 + 2d)\xi^2 - 2d(e^{I\theta} + e^{-I\theta})\xi - (1 - 2d) = 0$$

Solving this as a quadratic in ξ and using $\cos \theta = \frac{e^{I\theta} + e^{-I\theta}}{2}$ we obtain:

$$\xi = \frac{4d \cos \theta \pm \sqrt{16d^2 \cos^2 \theta + 4(1 - 4d^2)}}{2(1 + 2d)}$$

Simplifying and using the relation $\sin^2 \theta + \cos^2 \theta = 1$:

$$\xi = \frac{2d \cos \theta \pm \sqrt{1 - 4d^2 \sin^2 \theta}}{(1 + 2d)}$$

Case 1 ($1 - 4d^2 \sin^2 \theta < 0$):

In this case the value of $G = \xi$ would be imaginary as a result to fulfill the condition $|G| \leq 1$ this would be $G^*G \leq 1$ where G^* is the complex conjugate of G .

$$\begin{aligned} G^*G &= \frac{4d^2 \cos^2 \theta + 1 - 4d^2 \sin^2 \theta}{(1 + 2d)^2} = \frac{1 + 4d^2(1 - 2\sin^2 \theta)}{(1 + 2d)^2} \leq 1 \\ \implies 1 + 4d^2(1 - 2\sin^2 \theta) &\leq 1 + 4d + 4d^2 \implies -2d \sin^2 \theta \leq 1 \end{aligned}$$

This last equality hold for $\forall d > 0$.

Case 2 ($1 - 4d^2 \sin^2 \theta \geq 0$):

In this case the value of $G = \xi$ would be real and the condition for stability would be:

$$\left| \frac{2d \cos \theta \pm \sqrt{1 - 4d^2 \sin^2 \theta}}{(1 + 2d)} \right| \leq 1$$

This results in four cases to verify noting that since we are in the case where the discriminant is real, the maximum value of the discriminant is 1 when $4d^2 \sin^2 \theta = 0$ and hence:

$$2d \cos \theta + 1 \leq 1 + 2d \implies \cos \theta \leq 1$$

$$2d \cos \theta - 1 \leq 1 + 2d \implies \cos \theta \leq 1 + \frac{1}{d}$$

and for the remaining two case:

$$2d \cos \theta - 1 \geq -1 - 2d \implies \cos \theta \geq -1$$

$$2d \cos \theta + 1 \geq -1 - 2d \implies \cos \theta \geq -1 - \frac{1}{d}$$

All of these inequalities hold for $\forall d > 0$, as a result the DuFort-Frankel Method is **unconditionally stable** in the Von-Neumann sense.

Theorem 1. If an explicit finite difference approximation method of a parabolic initial-value problem is convergent, the time and space steps must satisfy. $\frac{\Delta t}{\Delta x} \rightarrow 0$ as $\Delta t \rightarrow 0$.

This restriction arises due to the consistency error, as a result even though the method is unconditionally stable in the Von-Neumann sense it convergence is still conditional.

3.4 Laasonen Method

The next method to be discussed is an implicit method commonly also referred to as the Backward Time Central Space (**BTCS**) method. The difference between implicit and explicit methods are that implicit methods use spatial points on the current time step being calculated when calculating new points. The result of this is that they are more computationally expensive as a linear system needs to be solved at every time-step. The benefit of using them is that they generally have better stability properties as will be discussed further in the stability analysis section.

$$\frac{u_i^{n+1} - u_i^{n-1}}{\Delta t} = \nu \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta x)^2} , 0 < i < im \quad (7)$$

Since this is an implicit method it is not possible to rearrange for u_i^{n+1} however instead the substitution for $d = \frac{\nu \Delta t}{(\Delta x)^2}$ as before to obtain:

$$u_i^{n+1} = u_i^{n-1} + d(u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}) , 0 < i < im \quad (8)$$

The figure 4 stencil shows which points will be used to calculate the approximation of each point u_i^{n+1} . From this stencil we can also see the implicit nature of this method by the fact that 2 of the points being used to calculate the new point lie on the same time level as it is being calculated on.

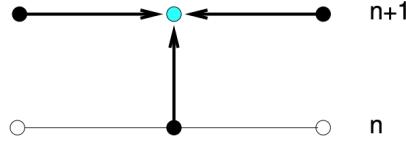


Figure 4: Laasonen Stencil

Since this method has used forward difference (a first order method) for its temporal approximation and a centered approximation (second order method) for its spatial approximation. The order of the the method is $O(\Delta t, (\Delta x)^2)$.

3.4.1 Stability Analysis

Making similar substitutions for those made for the stability analysis for FTCS we can obtain:

$$\xi^{n+1} e^{I\theta i} = \xi^n e^{I\theta i} + d(\xi^{n+1} e^{I\theta(i+1)} - 2\xi^{n+1} e^{I\theta i} + \xi^{n+1} e^{I\theta(i-1)})$$

dividing by $e^{I\theta i}$ using $2 \cos \theta = e^{I\theta} + e^{-I\theta}$ this simplifies to:

$$\xi^{n+1} = \xi^n + 2\xi^{n+1} d(\cos \theta - 1) \implies G = \frac{\xi^{n+1}}{\xi^n} = [1 + 2d(1 - \cos \theta)]^{-1}$$

So the condition for stability is:

$$|G| = |[1 + 2d(1 - \cos \theta)]^{-1}| \leq 1$$

Case 1:

$$1 \leq 1 + 2d(1 - \cos \theta)$$

This is true $\forall d > 0$ since $1 - \cos \theta \geq 0$

Case 2:

$$1 \geq -1 - 2d(1 - \cos \theta) \iff 1 \geq -d(1 - \cos \theta)$$

This is true $\forall d > 0$ since $-d < 0$ and $1 - \cos \theta \geq 0$ hence the RHS of the equation is always negative. This means that the Laasonen Method is **Unconditionally Stable** in the Von-Neumann sense.

3.5 Crank-Nicolson Method

The final method in our discussions in this paper approximates the $\frac{\partial^2 u}{\partial^2 x}$ using the average of the central differences at t^n and t^{n+1} .

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\nu}{2} \left[\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{n-1}^{n+1}}{(\Delta x)^2} + \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} \right], 0 < i < im \quad (9)$$

Rearanging:

$$u_i^{n+1} = u_i^n + \frac{d}{2} [u_{i+1}^{n+1} - 2u_i^{n+1} + u_{n-1}^{n+1} + u_{i+1}^n - 2u_i^n + u_{i-1}^n], 0 < i < im \quad (10)$$

This method can be seen as taking two steps at each time level, similar to how a Runge-Kutta method with two intermediate stages. The first step uses the explicit formulation (FTCS) to calculate $u_i^{n+\frac{1}{2}}$ and then subsequently using the implicit formulation (BTCS) to calculate u_i^{n+1} . This is well displayed when looking at the figure 5. The order of this method is $O((\Delta t)^2, (\Delta x)^2)$ since, the use of a central difference for the spatial second order derivative and an effective centered difference has been used for the temporal derivative if we consider each time step to be $\frac{\Delta t}{2}$ in size.

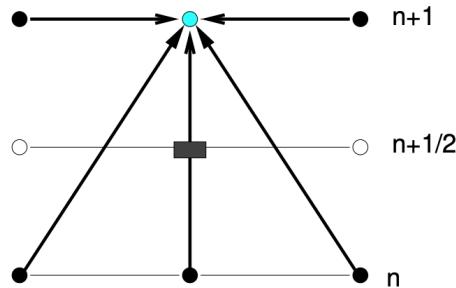


Figure 5: Crank-Nicolson Stencil

3.5.1 Stability Analysis

Making similar substitutions for those made for the stability analysis for previous methods we can obtain:

$$\xi^{n+1} e^{I\theta i} - \xi^n e^{I\theta i} = \frac{d}{2} (\xi^{n+1} e^{I\theta(i+1)} - 2\xi^{n+1} e^{I\theta i} + \xi^{n+1} e^{I\theta(i-1)} + \xi^n e^{I\theta(i+1)} - 2\xi^n e^{I\theta i} + \xi^n e^{I\theta(i-1)})$$

Dividing by $e^{I\theta i}$ and using $2 \cos \theta = e^{I\theta} + e^{-I\theta}$

$$\xi^{n+1} = \xi^n + d(\xi^{n+1} + \xi^n)(\cos \theta - 1)$$

Rearranging this the condition for stability becomes:

$$\left| \frac{\xi^{n+1}}{\xi^n} \right| = |G| = \left| \frac{1 - d(1 - \cos \theta)}{1 + d(1 - \cos \theta)} \right| \leq 1$$

Case 1:

$$1 - d(1 - \cos \theta) \leq 1 + d(1 - \cos \theta)$$

This is always true since $d(1 - \cos \theta) \geq 0$

Case 2:

$$1 - d(1 - \cos \theta) \geq -1 - d(1 - \cos \theta) \iff 1 \geq -1$$

Again this is true $\forall d > 0$. This means that this method is **Unconditionally Stable** in the Von-Neumann sense.

3.6 Stability Analysis Results

A summary of the stability analysis results can be found below:

- **Forward Time Central Space** is **Conditionally Stable** in the Von-Neumann sense with the condition that $d \leq \frac{1}{2}$
- **DuFort-Frankel Method** is **Unconditionally Stable** in the Von-Neuman sense, however does still have the condition $\frac{\Delta t}{\Delta x} \rightarrow 0$ as $\Delta t \rightarrow 0$ from Theorem (1).
- **Laasonen Method** is **Unconditionally Stable** in the Von-Neuman sense
- **Crank-Nicolson Method** is **Unconditionally Stable** in the Von-Neuman sense

Note: All 4 methods are consistent since as, $(\Delta x \rightarrow 0, \Delta t \rightarrow 0)$ the FDE tends to the PDE. Using the **Lax Equivalence Theorem** we can then deduce that since all four methods are consistent when they are stable they are also convergent. In this case the stability of the methods will determine whether or not the FDE converges to the PDE or not.

4 Results

This section will present the findings of the analysis of the simulations specified in the project. For each method plots will be shown to show the time evolution of how the calculated values of the velocity profiles changed based on which method was selected as well as which Δt was used. In conjunction to this, error plots will then be used to show how the error of each method changed.

4.1 Simulations

Below we will show plots to show the time evolution for each simulation.

4.1.1 Forward Time Central Space

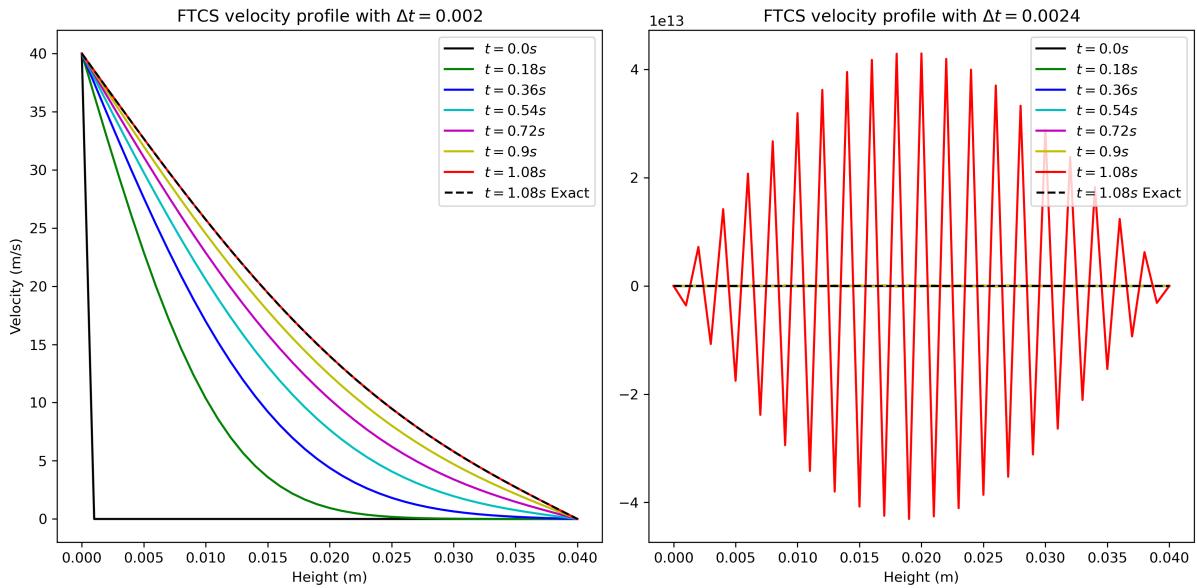


Figure 6: Forward Time Central Space Simulation

The Figure 6 shows how the velocity profile calculated values change as time progresses in the simulation. On the left we see what we would expect, being that at $t = 0$ the velocity profile matches the initial condition and then as t increases the velocity profile of the fluid at all other heights being incrementally affected. To verify that the solution being displayed at $t = 1.08s$ is what we would expect an additional line has been added to both plots to indicate the analytical solution at this time. As we can see from the graph on the left the two lines of our numerical line overlaps with the analytical solution, supporting our hypothesis that the FTCS method has converged to the PDE solution.

In contrast to this we can see on the figure on the right that the graph looks completely different to the figure on the left. This indicates that the FTCS method has become unstable and has not converged to the solution of the PDE, we can see that the scale of the y axis has changed and is now in the order of magnitude of 10^{13} , this is obviously incorrect and this is verified by the fact that the analytical solution does not match the numerical solution at $t = 1.08s$. In particular in this graph since the velocity profile at the end of the simulation is so large in magnitude compared to the other time points, this has caused the graph to the constant function 0 until the last time step. This is not the case, it is simply that the graph is unable to show the smaller changes at earlier time steps as well as the larger changes at the final time step. The reason this the convergence of this method has failed is due to that fact that Δt has changed

value between the two simulations. In the first one, $\Delta t = 0.002$ and hence $d = \frac{\nu\Delta t}{(\Delta x)^2} = 0.434 \leq 0.5$. This is significant since as we deduced in the stability analysis section for the FTCS method if $d \leq \frac{1}{2}$ then the method is stable (shown is consistent). This is seen as the graph on the left converges to the analytical solution. In contrast in the second simulation $\Delta t = 0.0024 \implies d = 0.5208 \not\leq 0.5$. This means the condition for stability for the FTCS method has been violated, hence as a result the method is not stable and not convergent.

4.1.2 DuFort-Frankel Method

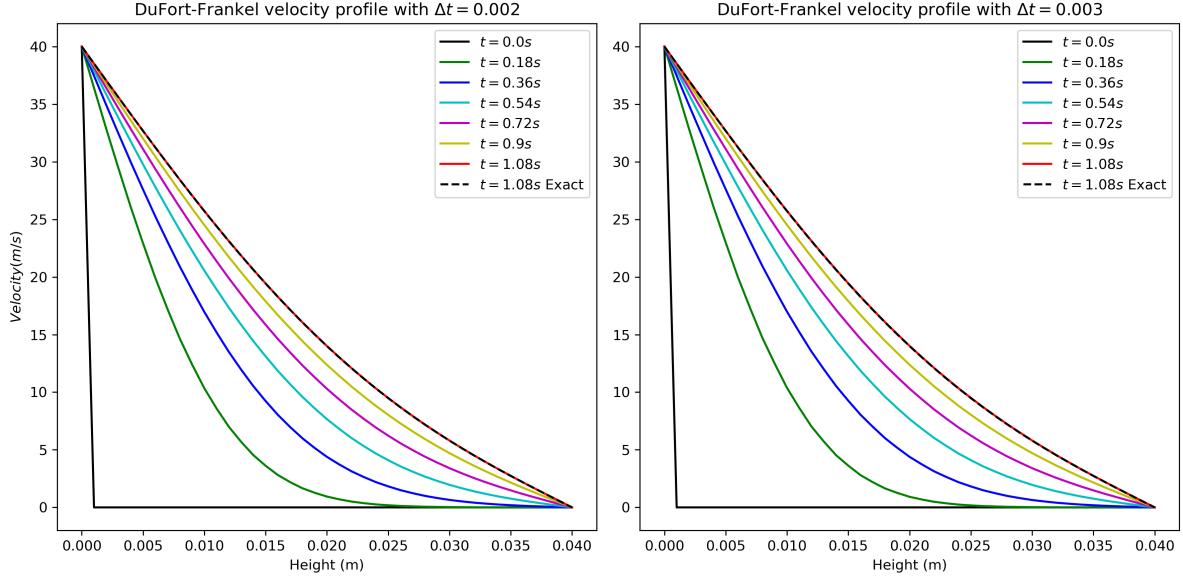


Figure 7: DuFort-Frankel Simulation

The figure 7 demonstrate that the DuFort-Frankel Method remains stable and hence convergent for both of the Δt values used. However it is important to note here that although the DuFort-Frankel Method is unconditionally stable large time steps can cause a loss of accuracy, which will be investigated further in the Errors section. Additionally, since this method required two initial conditions and only one was provided. The Laasonen method was used to calculate the first time step. The reason for this was because it is unconditionally stable and in the second simulation, the condition for stability for the FTCS was violated.

4.1.3 Laasonen Method

The figure 8 demonstrates that the Laasonen Method remains stable (and consistent) and hence convergent for both of the Δt values used. Interestingly here, although the $\Delta t = 0.01$ for the second simulation was set much greater than $\Delta t = 0.003$ like in the Du-Fort Frankel method. It appears as though, which will be investigated further in the error section, that the method remains accurate even with this larger time step. This is due to the fact that this is an implicit method whereas the previous two methods were explicit. Implicit methods can make larger time steps without becoming unstable, one of their key advantages over explicit methods, this is exhibited here since the curves still appear smooth even with the larger time step value.

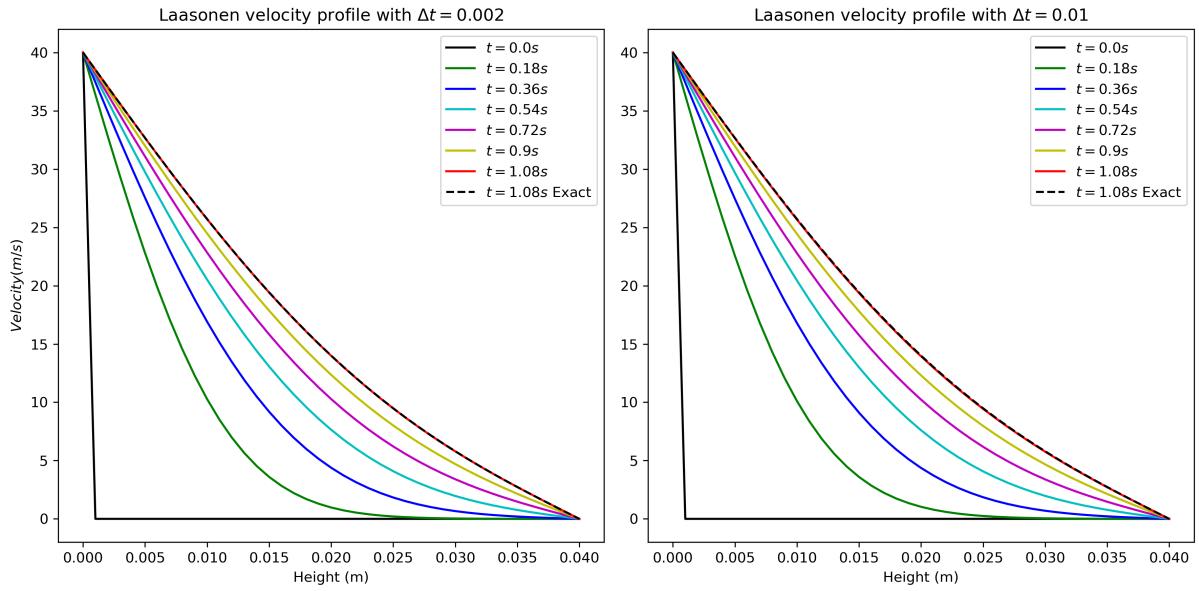


Figure 8: Laasonen Simulation

4.1.4 Crank-Nicolson Method

The figure 9 demonstrates that the Crank-Nicolson Method remains stable (and consistent) and hence convergent for both of the Δt values used. Again this exhibits similar behaviour as that observed with the other Laasonen method, this being due to the fact that both of these methods are implicit and hence have similar characteristics. The difference here being that this method is second order in time and so should have better accuracy, this cannot be observed in these plots and will be investigated further in the error section more.

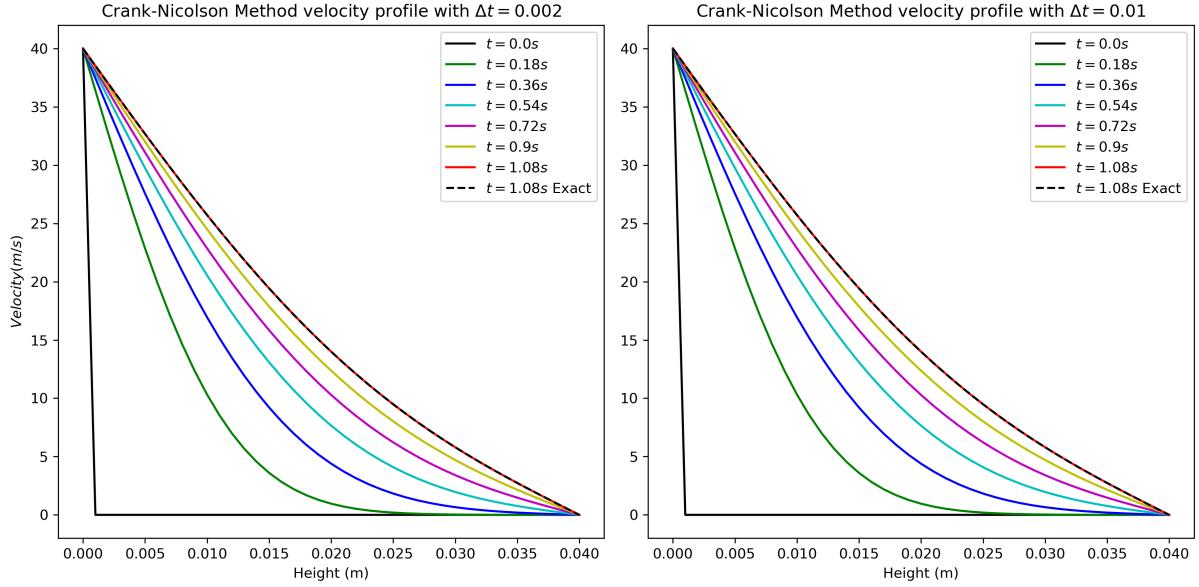


Figure 9: Crank-Nicolson Simulation

4.2 Errors

In this section, investigations will be conducted to explore how the errors of these methods changes, before doing so, we must define the error which we will be using:

$$\text{Error} = \text{Analytical} - \text{Numerical} \quad (11)$$

Numerical being the numerical solution calculated using one of the methods being investigated and *Analytical* being the approximation of the exact solution of the PDE.

When implementing this solution, the solution contained two infinite series. Since it is not possible to compute the exact sum of this series an approximation had to be made. This was done by creating a function for each of the infinite sums to closely approximate their values by continually adding values to the series until one of two conditions was met. First if the magnitude of the term being added was less than 10^{-12} or if the number of terms in the series had reached 10,000 terms. After investigation it was found that the first criteria was reached within only adding a few terms, meaning even though the tolerance was set very low it did not incur much computational cost to reach it. The reason this value was chosen was because it was small enough to not impact the results in the forthcoming errors investigation but was not too small to be affected by machine precision error.

In the analysis of the errors of the methods the infinity norm was used, so it is defined here for completeness. L_∞ of a vector \mathbf{x} is defined as follows:

$$L_\infty = \max_i |x_i| \quad (12)$$

where x_i are the components of \mathbf{x} .

4.2.1 Forward Time Central Space

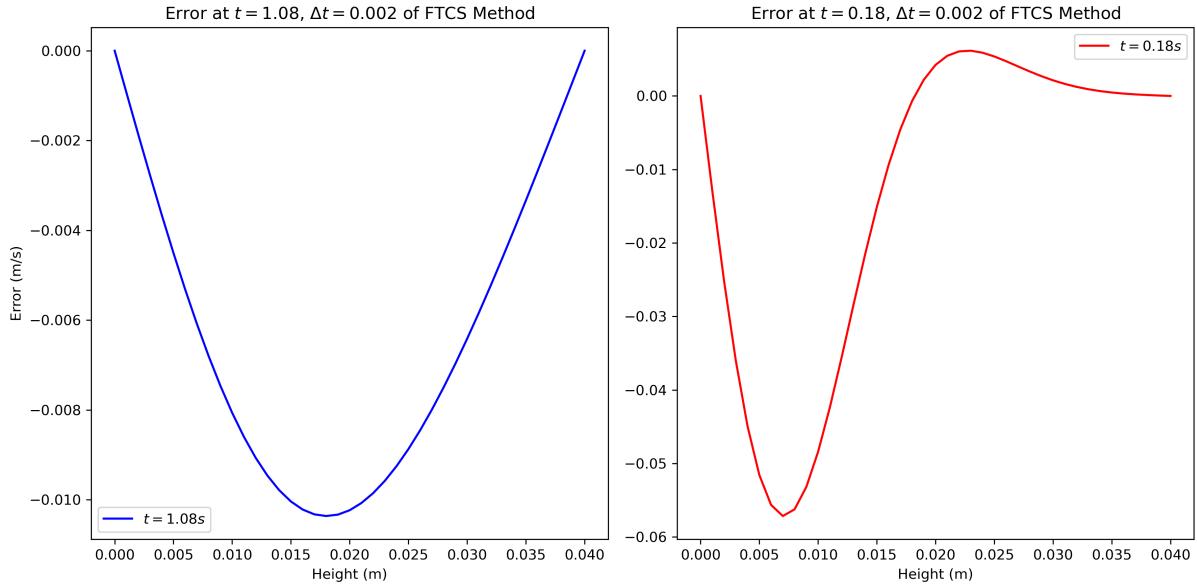


Figure 10: Forward Time Central Space error with $\Delta t = 0.002$

In figure 10 we can see how the error defined by (11) changes throughout the height between the parallel plates. As we saw earlier using $\Delta t = 0.002$ for this method resulted in $d \leq \frac{1}{2}$ meaning that the method was stable (and consistent) and hence convergent. We see that the error follows a parabolic shape curve at $t = 1.08$ this is to be predicted since the boundary values fix the error at the endpoints to be zeros and hence the largest error occurring furthest from these points would be predicted. The maximum error for this method was calculated using the L_∞ (infinity norm) to be: $L_\infty = 0.01036$. The graph on the right shows the error at $t = 0.18s$ this being earlier in the simulation. We can see a different shape emerging here with $L_\infty = 0.05714$ being much larger than at the end of the experiment. This can be attributed to near the beginning of the simulation the rate at which the velocity changes is much greater than close to the end. This rapid change of velocity profile means that the numerical scheme is more likely to make bigger errors nearer the beginning than at the end when the rate of change slows down. In addition to this due to the fact that an initial condition was set of $40m/s$ at one of the plates and 0 at all other heights on the grid this caused a very steep gradient at this point in the profile. When the simulation begins the biggest change will occur near the bottom of the plate, since this is the plate that is moving, this is again why the error would be larger nearer this plate as it exhibits more volatile behaviour.

In figure 11 we see how the error changes for the unstable method (not convergent) since in this case $d > \frac{1}{2}$. Although the errors are massive with an order of magnitude of 10^{13} with $L_\infty \approx 4.3 \times 10^{13}$ at $t = 1.08s$. We can see that the errors rise in magnitude as they move away from the defined boundary points, which is the same trend seen in the stable (convergent) case. In addition we also see that on the graph on the right similar to the stable case the larger errors are concentrated nearer the moving plate since this is where more change of velocity is occurring. We note that the reason why the error near the stationary plate has remained small at $t = 0.18s$ could be due to the fact that the FDE method has had fewer iterations of change on this side to deviate from the solution, since it is earlier on in the experiment. The oscillatory behaviour of the error is interesting as it is observed even some of the stable cases to be seen, explanation to this will be discussed but, it would be incorrect to analyse the error here since the method is not convergent and so the result cannot be interpreted as correct results.

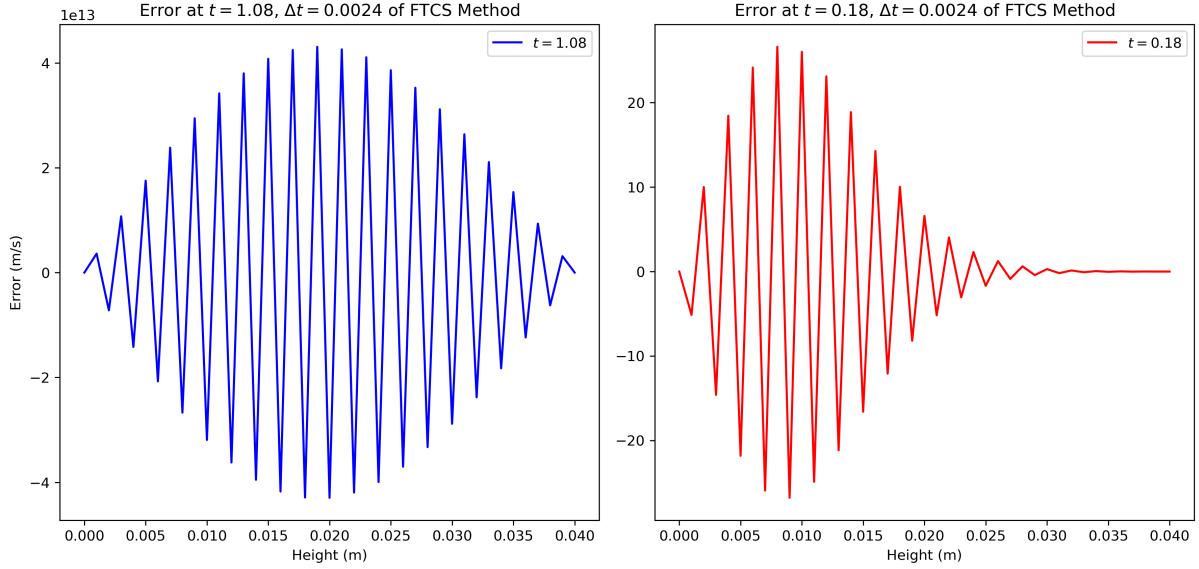


Figure 11: Forward Time Central Space error with $\Delta t = 0.0024$

4.2.2 DuFort-Frankel Method

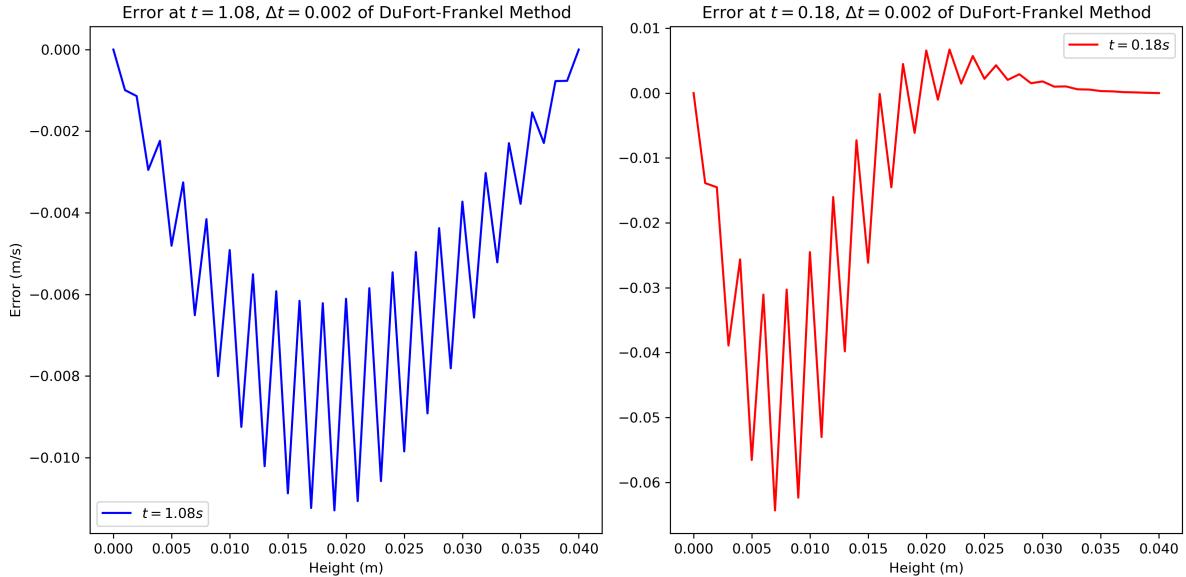


Figure 12: Forward Time Central Space error with $\Delta t = 0.002$

The figure 12 shows how the error varies when $\Delta t = 0.002$. A similarity between this error and the one for the FTCS method is that the maximum error occurs at similar parts in the graph, for each graph respectively. The first graph has its largest error $L_\infty = 0.0113$ furthest from the boundaries and for when $t = 0.18$ the $L_\infty = 0.06432$ occurred nearer the moving plate for reasons discussed previously. The biggest difference being that the errors for this method exhibit an oscillatory behaviour. This can be explained by the fact that this is an explicit method. A characteristic of explicit methods is numerical diffusion. This being the fact the method artificially smooths out sharp gradients, potentially causing non-physical oscillations around steep fronts or discontinuities. In this case since a sharp gradient was

introduced by the initial condition being $40m/s$ at one point and $0m/s$ everywhere else. Another explanation could be that the time step being taken is too large, this meaning that the information in the solution may not propagate accurately between distant grid points within a single time step. Another observation about this error profile is that even though the method being used is second order in time and space, and for FTCS it was first order in time and second order in space. The maximum error for both at $t = 1.08$ is relatively similar. The cause of this may be due to the fact that the solution is oscillating and so even though the DuFort-Frankel method is greater order in time this behaviour is not visible due to these oscillations.

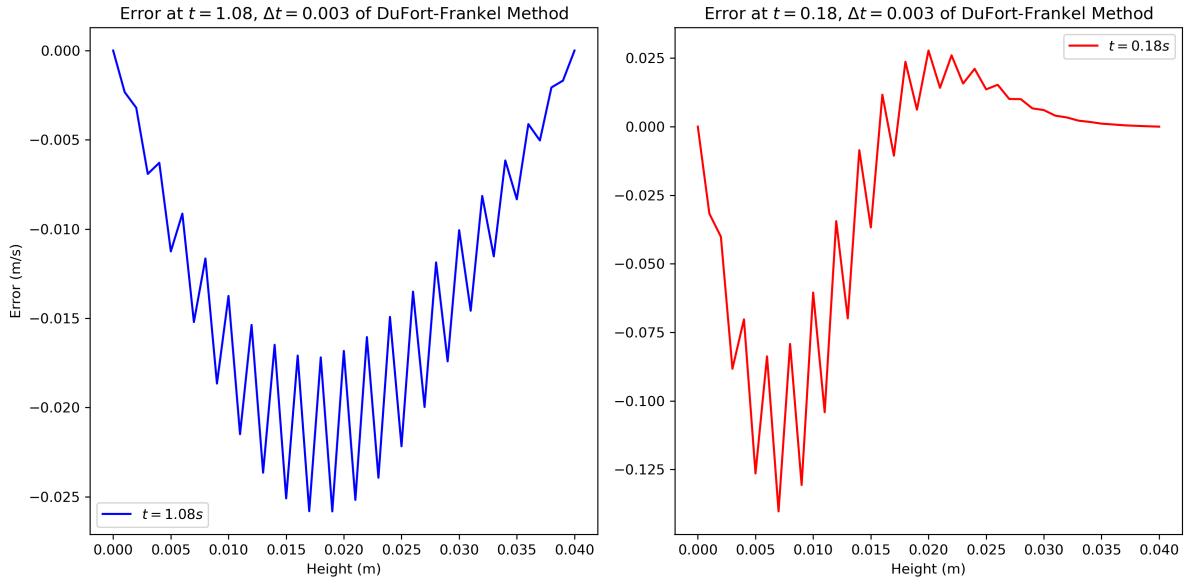


Figure 13: Forward Time Central Space error with $\Delta t = 0.003$

The figure 13 shows how the error changes when $\Delta t = 0.003$ here we see a very similar error trend with the key difference being the magnitude of the errors. The $L_\infty = 0.02582$ at $t = 1.08$ and $L_\infty = 0.14025$ at $t = 0.18$. These are both larger than those observed previously and this is to be expected since a larger Δt was chosen. Interestingly the errors have more than doubled even though the Δt was only only multiplied by 1.5. This again is likely due to the oscillatory behaviour of the solution affected the maximum error observed.

4.2.3 Laasonen Method

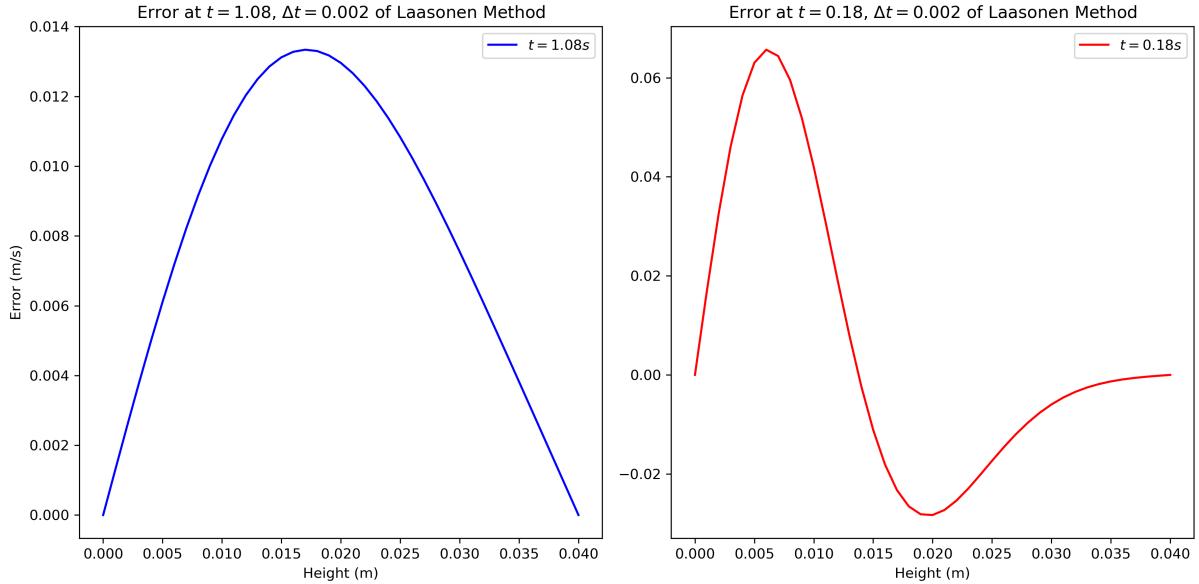


Figure 14: Laasonen Method error with $\Delta t = 0.002$

The figure 14 shows how the error changes when $\Delta t = 0.002$ for the Laasonen Method. The $L_\infty = 0.01334$ at $t = 1.08$ and $L_\infty = 0.06566$ at $t = 0.18$. We observed similar error profiles to the FTCS method. This is because this method is implicit and is unconditionally stable, and similar to the FTCS method is order one in space and order two and time. This explains why the values of the errors are similar to those found for the FTCS method. The reasons for the shapes of the graph is the same as that explained for the FTCS error graphs.

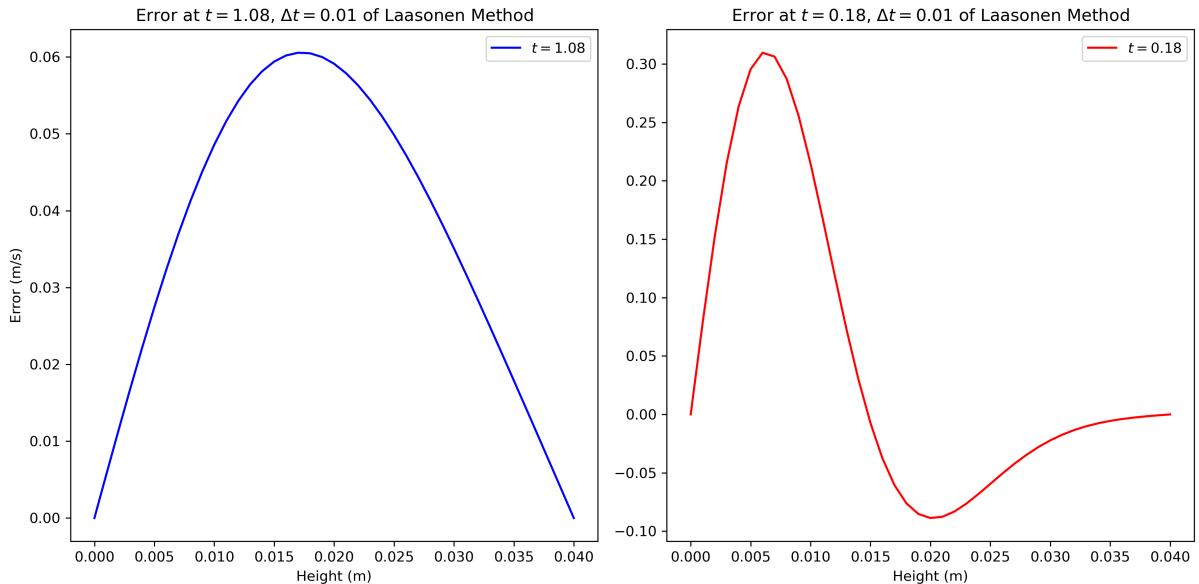


Figure 15: Laasonen Method error with $\Delta t = 0.01$

The figure 15 shows how the error changes with $\Delta t = 0.01$. The graphs look very similar but again the key difference is the magnitude of the errors. The $L_\infty = 0.06055$ at $t = 1.08$ and $L_\infty = 0.30956$

at $t = 0.18$. This is more in line with what we would expect since the value for Δt has been multiplied by 5 and the error has also increased by roughly 5 times too. This also supports our conclusion that this method is first order in time.

4.2.4 Crank-Nicolson Method

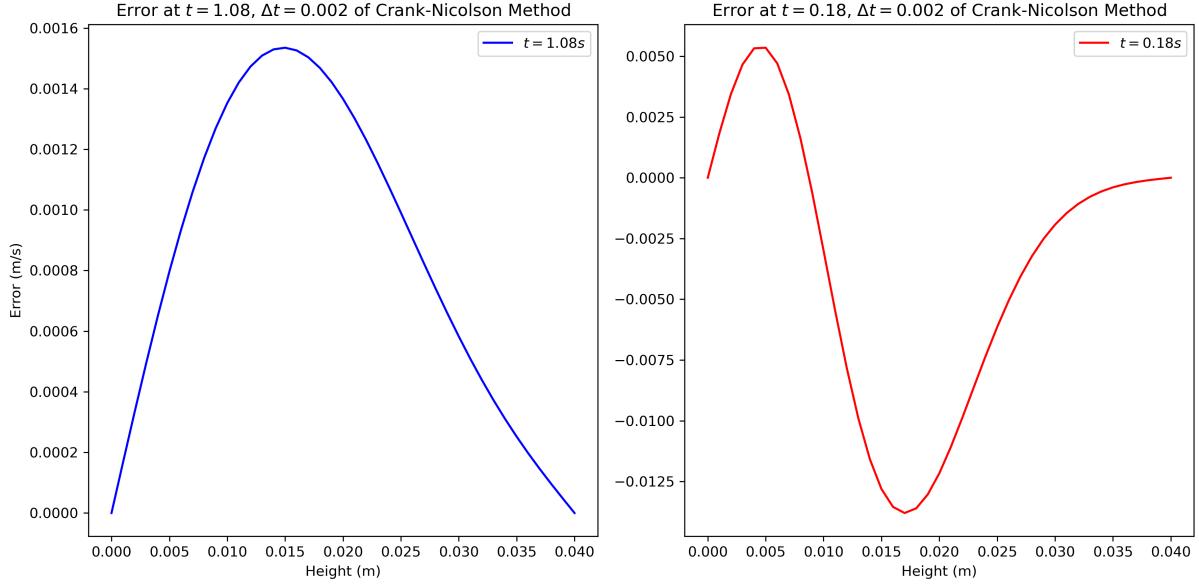


Figure 16: Crank-Nicolson Method error with $\Delta t = 0.002$

The figure 16 shows how the error changes with $\Delta t = 0.002$. This again is an example of an implicit method, the profile of the errors follows a similar pattern to the BTCS method. The $L_\infty = 0.00154$ at $t = 1.08$ and $L_\infty = 0.0138$ at $t = 0.18$. The key difference here is that fact that the magnitude of the errors is much smaller than any of the methods previously analysed, this is due to the fact that this method is both second order in time and space, and unlike the DuFort-Frankel method is implicit and does not exhibit the oscillatory behaviour masking the accuracy of the method.

The figure 17 shows how the error changes with $\Delta t = 0.01$. The graph of the error at $t = 1.08$ is what we would expect since, the size of the time step has increased and therefore the accuracy of the method has decreased. The interesting graph is how the error appears at $t = 0.18$ this is significantly different to the when $t = 0.002$ and this is because the time step is so much larger and this error was taken earlier on in the simulation the method has still not been able to smooth out the steep gradient of the initial condition, hence the small oscillation near the top left of the graph, this shows that even though the scheme is stable when a large enough time step is taken the accuracy of the method does still suffer.

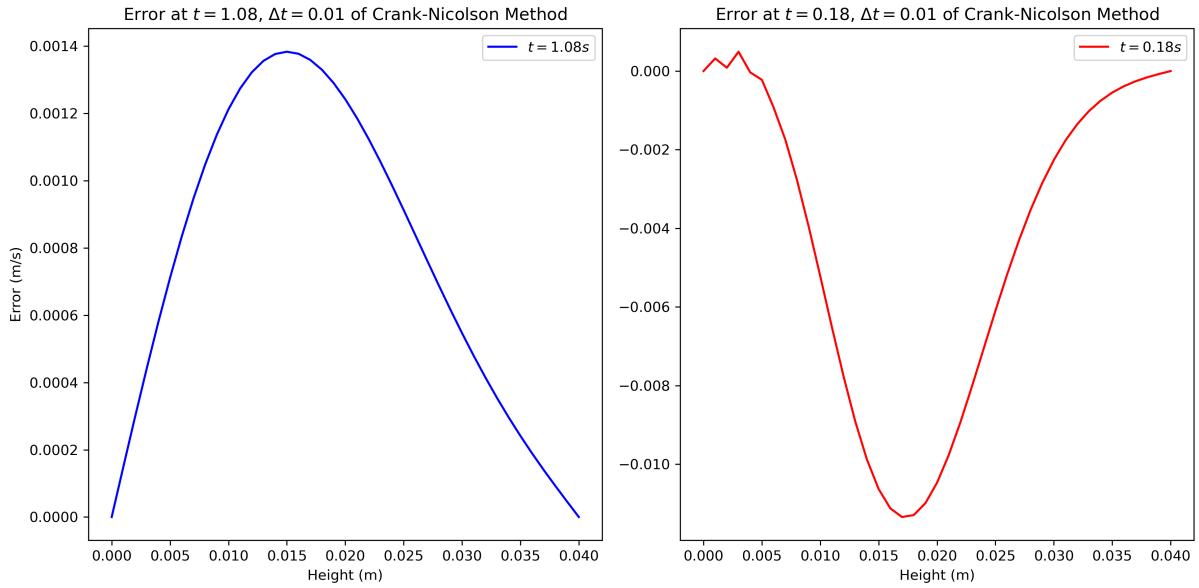


Figure 17: Crank-Nicolson Method error with $\Delta t = 0.01$

4.3 Comparing Error Profiles

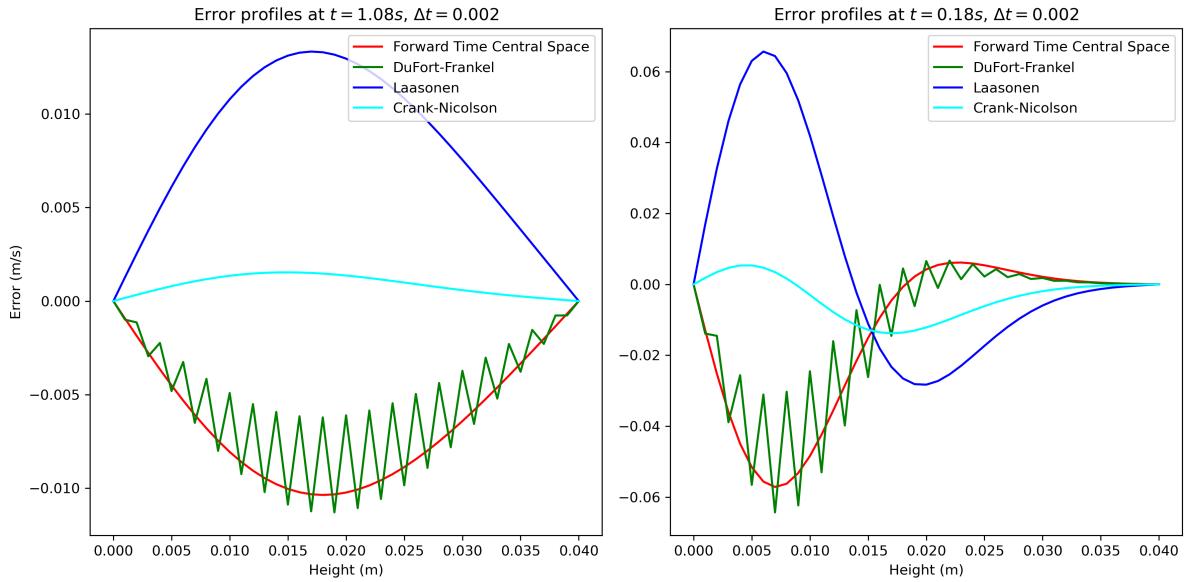


Figure 18: Errors of methods

The figure 18 summaries the results of the analysis above and demonstrates a comparison of the methods at both of the time values. We can see from this that it appears, as confirmed by the analysis above that the Crank-Nicolson method performed the best, at both evaluation times.

4.4 Time step in the Laasonen method

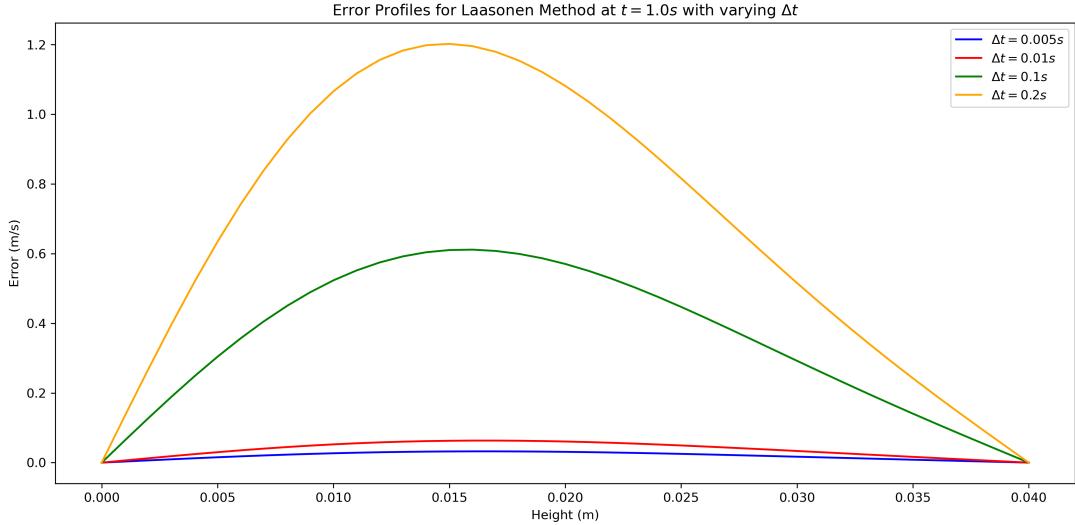


Figure 19: Laasonen Method changing Δt

The figure 19 shows the impact of the accuracy of the Laasonen method when varying the value of Δt . The $L_\infty = 0.03236$ at $\Delta t = 0.005$. The $L_\infty = 0.06327$ at $\Delta t = 0.01$. The $L_\infty = 0.61152$ at $\Delta t = 0.1$. The $L_\infty = 1.20233$ at $\Delta t = 0.2$. As discussed previously the Laasonen method is an implicit method which is order one in time and order two in space. We can see here that as the value of Δt decreases the overall error of the solution also decreases. Specifically, we can see from the maximum errors it decreases roughly at the rate of Δt . The maximum error Doubles from $\Delta t = 0.005$ to $\Delta t = 0.01$ and then multiplies by 10 to $\Delta t = 0.1$ and again doubles when $\Delta t = 0.2$. As well as the maximum error following the correct trend we can see from the graph that it also appears the the remainder of the points also appear to following the same trend seen in the infinity norm.

It is important to note here that as Δt was reduced although the error went down too, the number of time steps needed increased, and as a result the overall computation time for each one increased as seen in figure 20. This displays the trade off of computation vs accuracy that must be considered when picking which grid size to pick, in this case the computation times were very small, so this didn't have a big impact. But what we can see here is that the magnitude of decreases in the errors decreasing gets progressively smaller. For example the gain from moving from $\Delta t = 0.2$ to $\Delta t = 0.01$ was massive, but comparatively the change to $\Delta t = 0.005$ had a minimal magnitude of difference change. If the absolute lowest error is needed then picking the lowest grid size is appropriate but if the accuracy of the solution does not need to be extremely accurate then choosing a slightly larger grid size would speed up computation, and still return a relatively similar answer.

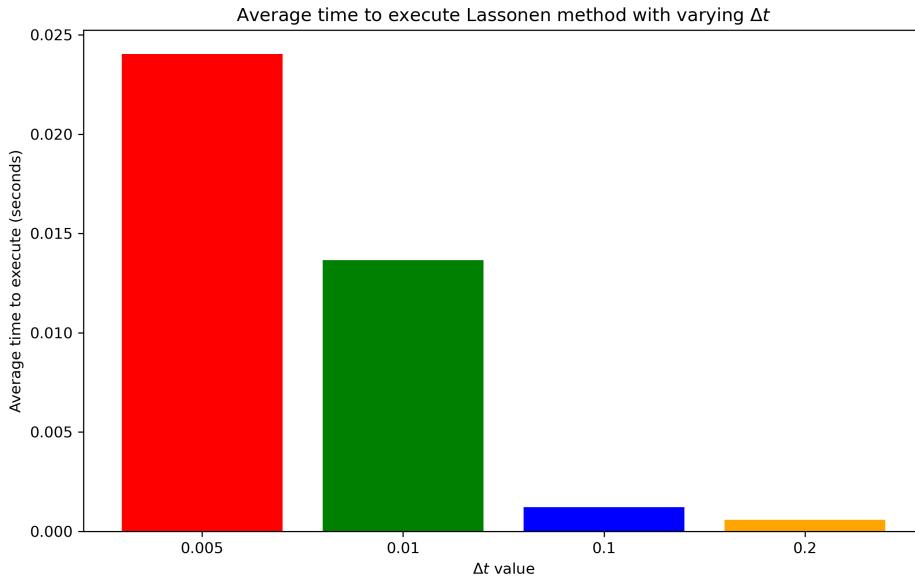


Figure 20: Average computation time of Laasonen Method changing Δt

5 Discussion and Conclusion

In this section, a discussion of the strengths and weaknesses of each of the four methods will be a discussed and a conclusion will be drawn to show which method should be selected in this situation.

Forward Time Central Space: This method has the benefit of being simple to implement and understand in addition since it is an explicit method, it can be computed directly from the previous time step without solving a linear system, leading to lower computational cost per step. The downsides to this method is has the condition of stability of $d \leq \frac{1}{2}$, so requires small time steps to ensure stability, this could lead to high computational cost for simulations which last over long periods of time. In addition it is only first order in time, so the accuracy of the scheme could be improved.

DuFort-Frankel: This method had the benefit of being unconditionally stable, unlike FTCS, meaning that larger timesteps could be taken, whilst still having the benefit of low computational cost per step. Another benefit of this method is that it is second order in space and time, meaning it should be more accurate than FTCS, however as we have seen in this simulation, since it is an explicit method it can be prone to issues such as numerical diffusion, causing oscillatory behaviour due to the the initial conditions chosen, meaning that this additional order of accuracy cannot be seen in the results. Since it is a three level scheme, this means that the implementation is less straightforward than FTCS. In addition it still also had a condition on convergence as mentioned in theorem (1).

Laasonen: This method again had the benefit of again being unconditionally, meaning that regardless of how large the time step size was it remained stable. Being an explicit method it suffered less from numerical diffusion so the initial conditions did not cause the oscillatory behaviour as seen in DuFort Frankel. The downside to this method was that it requires solving a linear system for each time step, increasing computational cost and complexity. In addition, as a result it was more complex to implement than the explicit methods. Fianlly, it also was only first order accurate in time, meaning that accuracy of the scheme could have been improved as seen in the Crank-Nicolson method.

Crank-Nicolson: This method had the benefit of being unconditionally stable, regardless of the time step taken, the method remained stable. Again since it was an implicit method, it suffered less from numerical dispersion and diffusion compared to explicit methods. This method was also second order in space and time, meaning that the accuracy of the scheme was better than the BTCS method. The main drawback of this method is that again it required solving a system of equations at each time step, which can be computationally intensive. Also since it required two starting points to begin, an approximation had to be made for the second starting point using the FTCS method. This introduced an inherent error in the solution since its starting conditions now contained one approximated value. This error was very small however since, only one time step was taken. Finally, being an implicit method, with a multi-level scheme the implementation of this scheme was more complex and harder to interpret.

Overall, all methods have situations they are best suited for. FTCS method being good for problems where simplicity and low computational cost is a priority, over having to take smaller time steps for stability. DuFort-Frankel method being a higher order explicit scheme, which is unconditionally stable but facing issues of accuracy due to numerical diffusion, which causes oscillatory behaviour. Laasonen method, being an implicit unconditionally stable, but with the drawback of being computationally more costly. Finally, the Crank-Nicolson method being of higher order than the Laasonen method, but still requiring the solving of a linear system at each time step.

In the context of this problem, the best method appears to be the Crank-Nicolson method. Despite, the increased computational cost, its characteristics of unconditionally stability and being second order in time and space, meaning it is more accurate in this context than the other 3 methods (same order as DuFort-Frankel, but more accurate since is not affected by numerical diffusion), shows that it is the best choice of the four methods in this situation. The comparison of these methods does highlight, how the specific requirements of the problem, affect which method is more well suited.

References

- [1] R. Courant and D. Hilbert, *Methods of Mathematical Physics, Vol. II: Partial Differential Equations*, Wiley-VCH, 1953.