Numerical simulation of fluid dynamics between two parallel plates using FTCS explicit, DuFort-Frankel explicit, Laasonen implicit, and Crank-Nicolson implicit

James Portier

Date: March 2024

Abstract

This report investigates the numerical simulation of fluid dynamics between two parallel plates, focusing on the comparison of four numerical methods: FTCS explicit, DuFort-Frankel explicit, Laasonen implicit, and Crank-Nicolson implicit. Through initial stability analysis using the Von Neumann technique, it was found that the FTCS method is conditionally stable, requiring careful parameter selection, whereas the other three methods are unconditionally stable, offering greater flexibility. Each method was also simulated using Python code, verifying the analytically derived stability properties, comparing the errors of the methods, and investigating the impact of step size on the error magnitude of the Laasonen method. The Crank-Nicolson method, in particular, provides a superior balance of accuracy and stability, making it highly recommended for high-precision studies. The findings guide researchers towards selecting the most suitable numerical method based on their specific accuracy requirements, computational resources, and stability needs, offering a comprehensive understanding of their applicability and efficacy in fluid dynamics research.

Keywords: Fluid dynamics, FTCS explicit, DuFort-Frankel explicit, Laasonen implicit, Crank-Nicolson implicit, Von Neumann stability analysis, Python simulation.

Contents

1 Introduction

The study of fluid dynamics plays a pivotal role in understanding various physical and biological systems. By describing the behavior of liquids and gases in motion, fluid dynamics facilitates the prediction of flow patterns, which is vital for numerous applications ranging from weather forecasting to industrial process design. This project focuses on a specific case of a viscous fluid constrained between two parallel plates separated by a distance y. Such configurations are common in engineering and physics, representing simplified models for more complex fluid flow scenarios. The dynamics of this system are particularly interesting as they can be indicative of the transitional behaviors from laminar to turbulent flow, which are critical in optimizing systems for aerodynamic efficiency and in the design of various engineering apparatuses.

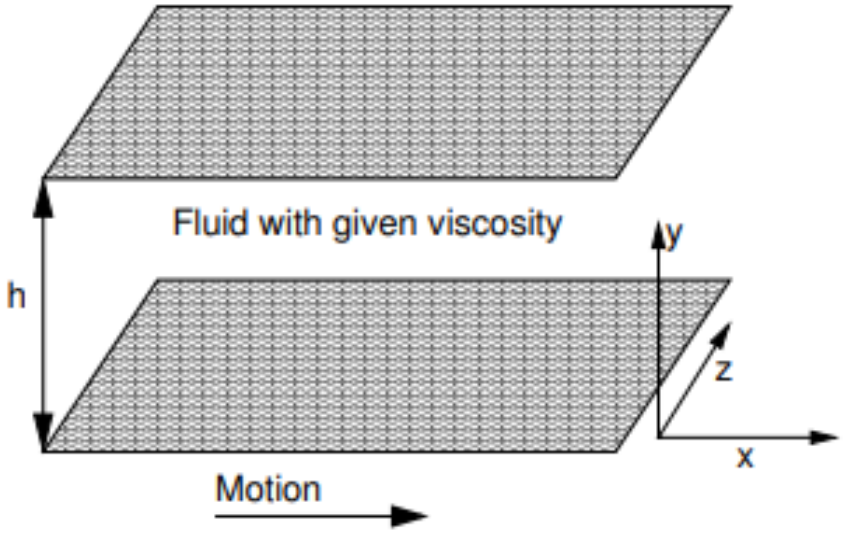

Figure 1: **Parallel plate simulation**

We consider the situation where the fluid and plates are initially at rest, and an instantaneous velocity U_0 is applied to the lower plate in the x-direction, keeping the upper plate stationary. This setup leads to a time-evolving velocity profile of the fluid between the plates, governed by a particular parabolic Partial Differential Equation (PDE). A PDE is a mathematical equation that involves rates of change with respect to continuous variables. In the context of fluid dynamics, PDEs enable the modeling of the flow and behavior of fluids over time and space. The PDE we will use, alongside appropriate initial and boundary conditions, describes the evolution of the fluid's velocity profile, which is crucial for various design purposes in engineering applications.

The objective of this study is to analyze the fluid's velocity profile using numerical methods due to the analytical challenges posed by complex boundary conditions. Our approach encompasses a comprehensive evaluation of several numerical schemes: the FTCS explicit method, known for its simplicity but restricted by a stringent stability condition; the DuFort-Frankel explicit method, which offers better stability but can introduce numerical dispersion errors; the Laasonen implicit method, valued for its unconditional stability at the cost of increased computational effort for solving linear systems; and the Crank-Nicolson implicit method, which strikes an optimal balance between accuracy and stability but also requires solving linear systems at each time step. These methods will be simulated and analyzed, addressing their unique challenges such as stability, convergence, and accuracy. Stability analysis,

including the use of the von Neumann technique, will play an important role in assessing method suitability, particularly in the context of the fluid's kinematic viscosity and the configuration of the parallel plates. Convergence and accuracy will be evaluated through a comparison against known analytical solutions where possible, and by investigating the error behavior as a function of grid resolution and time step size. The trade-offs inherent in each method will be explored, specifically the computational cost versus the accuracy and stability, to guide the selection of the most appropriate method for different simulation scenarios.

2 Model

2.1 PDE

A simple second order parabolic PDE used for modeling the velocity of a viscous fluid between two parallel plates is given by the equation

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2} \tag{1}$$

where

- $\frac{\partial u}{\partial t}$ represents the partial derivative of the velocity field u with respect to time t, indicating how the velocity changes over time.
- $\nu \frac{\partial^2 u}{\partial y^2}$, is the kinematic viscosity ν multiplied by the second partial derivative of the velocity field u with respect to the spatial coordinate y. This term represents the diffusion of velocity, which is the rate at which momentum is spread out in the fluid.

The equation itself is a form of the diffusion or heat equation applied to fluid flow, specifically describing how the velocity of a fluid diffuses due to viscosity.

2.2 Formulation of the Finite Difference Model

By discretising equation (1), we can obtain the Finite Difference models to be used in this investigation. Applying a first forward difference of order $O(\Delta t)$ for the time derivative and a second order central difference $O((\Delta y)^2)$ for the spatial second derivative $\frac{\partial^2 u}{\partial y^2}$ to equation (1), we obtain the finite difference equation (FDE)

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

The *implicit form* for this equation would take the form

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \nu \left(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta y)^2} \right). \tag{3}$$

To illustrate the link between the finite different methods we're going to use, we present the *beta scheme* ([2], p25)of the parabolic model equation, which allows us to obtain the different finite difference schemes by changing the value of β:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \nu \left[\beta \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\Delta y)^2} + (1-\beta) \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta y)^2} \right] + O(\Delta t, (\Delta y)^2)$$

where

$$\begin{aligned} \beta = 0 &\quad \rightarrow \quad \text{FTCS} \\ \beta = 1 &\quad \rightarrow \quad \text{BTCS} \\ \beta = \frac{1}{2} &\quad \rightarrow \quad \text{Crank-Nicolson} \end{aligned}$$

Note that since the DuFort-Frankel method involves terms from **two** earlier times, it cannot be obtained from this particular β scheme. It can however be derived from equation (2), which we will see in the section following.

3 Numerical approach

3.1 Set-up

3.1.1 Conditions, parameters, and experiments

For the particular problem under investigation the parallel plates are at a distance apart of $0.04\,\text{m}$. The fluid between the plates is oil with a kinematic viscosity of $0.000217\,\text{m}^2/\text{s}$. The spatial discretisation of the computational grid should have $\Delta y = 0.001\,\text{m}$, therefore our $j_m = 40$ with $(j_m + 1) = 41$ points in y. The initial velocity U_0 is given as $40\,\text{m/s}$. The solution $u(t, y)$ is generated up to time point $t = 1.08$.

The initial and boundary conditions, along with diagrams visualising the simulation and the grid spacing, are shown in Figure 2 below:

(a) Initial condition $t = 0$, $u = U_0$ for $y = 0$

$u = 0$ for $0 < y \leq h$

(b) Boundary conditions $t \geq 0$, $u = U_0$ for $y = 0$

$u = 0$ for $y = h$

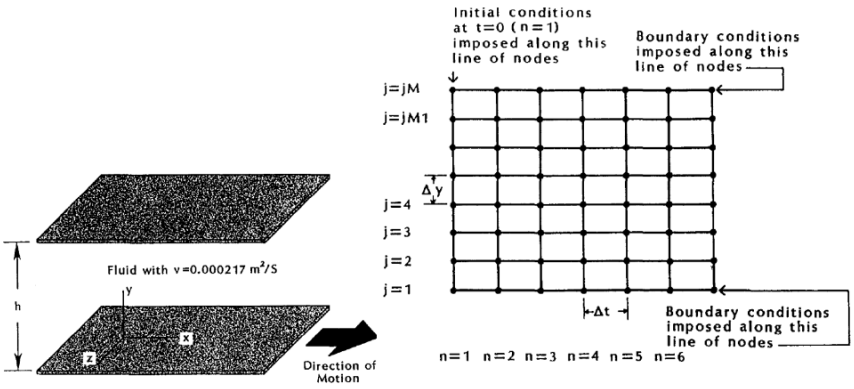

Figure 2: **Simulation set-up, taken from ([2], p20)**

To investigate some different numerical schemes, we provide simulations for the following numerical methods. NOTE that $nm + 1$ is the number of temporal points so the number of time steps would be nm. For each method there are two experiments, detailed as follows:

- FTCS explicit method.
  1. $\Delta t = 0.002$ and $nm = 540$
  2. $\Delta t = 0.0024$ and $nm = 450$
- DuFort-Frankel explicit method.
  1. $\Delta t = 0.002$ and $nm = 540$
  2. $\Delta t = 0.003$ and $nm = 360$
- Laasonen implicit method.
  1. $\Delta t = 0.002$ and $nm = 540$

2. $\Delta t = 0.01$ and $nm = 108$

- Crank-Nicolson implicit method.
  1. $\Delta t = 0.002$ and $nm = 540$
  2. $\Delta t = 0.01$ and $nm = 108$

3.1.2 Exact/analytical solution

The partial differential equation (1) along with constant Dirichlet boundary conditions, as given, can be solved analytically. Below, we formulate the solution as a series of complimentary error functions.

$$\begin{aligned} u &= U_0 \left\{ \sum_{n=0}^{\infty} \text{erfc}\left[2n\eta_1 + \eta\right] - \sum_{n=0}^{\infty} \text{erfc}\left[2(n+1)\eta_1 - \eta\right] \right\} \\ &= U_0 \left\{ \text{erfc}(\eta) - \text{erfc}(2\eta_1 - \eta) + \text{erfc}(2\eta_1 + \eta) - \text{erfc}(4\eta_1 - \eta) + \text{erfc}(4\eta_1 + \eta) - \ldots \right\}, \end{aligned}$$

where

$$\eta = \frac{y}{2\sqrt{\nu t}}, \eta_1 = \frac{h}{2\sqrt{\nu t}}.$$

We implement this using Python code, and use it to determine the error of each of the Finite Difference methods (given by Analytical - Numerical). Notice that in the above formulation we have an infinite series. Including more terms in the series generally increases the accuracy of the approximation because it accounts for more of the series' behavior. However, after a certain point, additional terms contribute negligibly to the final result due to the properties of the erfc function, which rapidly approaches zero for large arguments. We therefore choose to **truncate the series** to a suitable number of terms.

In our code we create a function that determines the number of terms we need to keep in the summation such that the coded function is correct to within a tolerance of 1×10^{-14}. This methodology is encapsulated in the function `calculate_N_required`, which iteratively evaluates the contribution of successive terms in the series until the absolute value of the latest addition falls below the specified tolerance threshold.

Specifically, for each iteration, the function computes the difference between two successive terms of the complementary error function, erfc, based on the variables η and η_1. The iterative process is as follows:

1. Calculate the contribution of the Nth term as the difference $\text{erfc}(2N\eta_1+\eta) - \text{erfc}(2(N+1)\eta_1-\eta)$.
2. If the maximum absolute value of this contribution across all y points is less than the tolerance, the process is terminated, and the current value of N indicates the number of terms required to meet the accuracy criterion.
3. Otherwise, increment N and repeat the process.

After running this function, it turns out that we need to truncate to **three terms**.

3.2 Finite Difference Methods

3.2.1 Forward Time Central Space (FTCS)

The FTCS (Forward Time, Centered Space) method is a numerical scheme used to solve partial differential equations, characterized by its forward differencing in time and central differencing in space. For our model equation (2) this gives

$$\boxed{u_i^{n+1} = u_i^n + \frac{\nu(\Delta t)}{(\Delta y)^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right).} \tag{4}$$

The method is of order $O\left((\Delta t), (\Delta y)^2\right)$, i.e. is of first order in time and second order in space, and the stencil for this method is shown in Figure 3

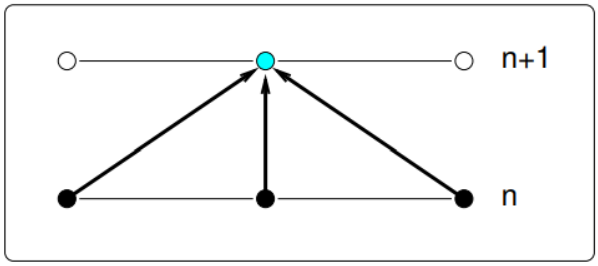

Figure 3: **Stencil for FTCS method**

3.2.2 DuFort-Frankel

The DuFort-Frankel explicit method uses a three-time-level approach. Central differencing is applied both temporally and spatially. For the second-order spatial derivative, the dependent variable u_i^n is substituted by the mean of u_i^{n+1} and u_i^{n-1} to enhance stability. For our model equation (2) this gives the following FDE:

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

Which gives the recurrence relation

$$\boxed{u_i^{n+1} = \frac{\left[1 - \frac{2\nu(\Delta t)}{(\Delta y)^2}\right]u_i^{n-1} + \frac{2\nu(\Delta t)}{(\Delta y)^2}\left[u_{i+1}^n + u_{i-1}^n\right]}{\left[1 + \frac{2\nu(\Delta t)}{(\Delta y)^2}\right]}} \tag{5}$$

The method is of order $O\left((\Delta t)^2, (\Delta x)^2, \left(\frac{\Delta t}{\Delta y}\right)^2\right)$. The stencil for this method is shown in Figure 4.

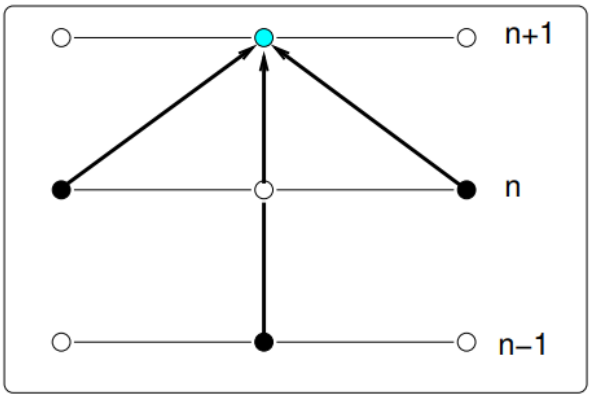

Figure 4: **Stencil for DuFort-Frankel Method**

Notice from the formulation and the method's stencil in Figure 4 that information from two earlier times t^n and t^{n-1} is required to complete the calculation of u^{n+1} at time t^{n+1}. This poses a problem for the first time step, as we don't have information from the step before it. To deal with this, we have the choice between either:

- Using two initial conditions to get started
- Using a one-step method for the first time-step (note that the overall accuracy of the method could be affected by the choice of the method used for the first time step).

In our implementation of the DuFort-Frankel method, the challenge of missing information for the first time step is addressed by employing the Laasonen simple implicit method for the initial time-step calculation. This choice is motivated by the unconditional stability of the Laasonen method, making it particularly suitable for our needs, especially in scenarios where the Forward Time Centered Space (FTCS) method proves inappropriate due to its conditional stability. Specifically, in the second experiment of the DuFort-Frankel method, $d = \frac{\nu \Delta t}{\Delta y^2}$ exceeds the critical value of $\frac{1}{2}$ (8), rendering the FTCS method unstable for that time-step size. This instability makes the FTCS method unsuitable for initiating the DuFort-Frankel method in this context.

By opting for the Laasonen method for the first step, we ensure a stable and accurate foundation for subsequent iterations using the DuFort-Frankel method. This approach significantly reduces the Mean Absolute Error (MAE) and the error under the infinity norm when compared to initiating with the FTCS or Crank-Nicolson methods. The Laasonen method, despite being implicitly computed and thus requiring more computational effort for the first step, offers a robust start to the DuFort-Frankel iterations. This strategic choice ensures that the overall error characteristics of the numerical solution are optimized, allowing for the effective application of the DuFort-Frankel method while maintaining the numerical scheme's reliability from the outset.

We will now use the implicit form of the discretised equation shown in equation (3) to present the final two Finite Difference methods.

3.2.3 Laasonen

The first of the two implcit methods we're investigating is the Laasonen Implicit Method, also known as Backwards Time Central Space, formulated as follows:

$$\boxed{\frac{u_i^{n+1} - u_i^n}{\Delta t} = \nu \left(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta y)^2} \right)} \tag{6}$$

The above formulation is of order $O\left((\Delta t), (\Delta y)^2\right)$. Implicit methods produce more than one unknown in the FDE - therefore a linear system of equations needs to be solved at each timestep.

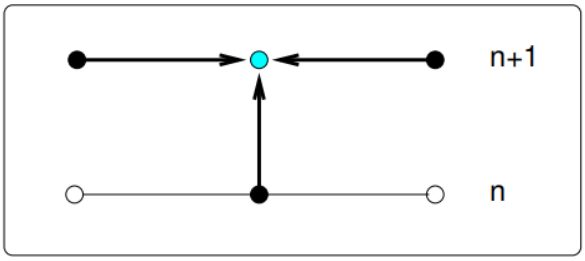

Figure 5: **Stencil for Laasonen Method**

3.2.4 Crank-Nicolson

In the Crank-Nicolson method, the differential $\frac{\partial^2 u}{\partial y^2}$ is replaced by the average of central differences at time t^n and t^{n+1}. This gives us the formulation

$$\boxed{\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\alpha}{2} \left[\frac{u_{i+1}^{n+1} + u_{i-1}^{n+1}}{(\Delta x)^2} + \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} \right].} \tag{7}$$

The stencil for the Crank-Nicolson method is given below in Figure 6

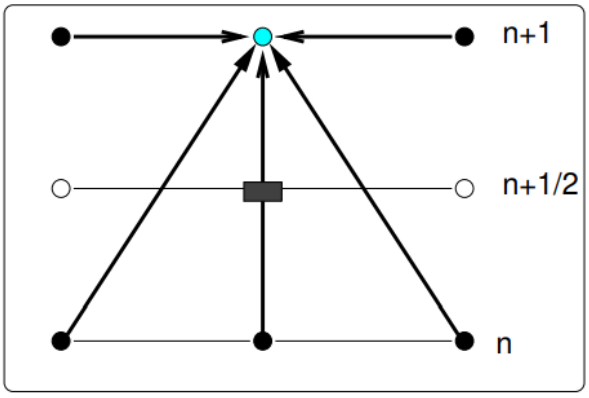

Figure 6: **Stencil for Crank-Nicolson Method**

3.3 Stability analysis

3.3.1 The Von Neumann Method

In *Von Neumann Stability Analysis*, the solution of the finite difference equation is expanded in a Fourier series. The decay or growth of the *amplification factor* obtained from this expansion indicates whether

or not the numerical algorithm is stable. Following, we apply to Von Neumann method to each of the finite difference equations in order to derive constraints with respect to stability.

FTCS

Recall the FTCS:

$$U_i^{n+1} = U_i^n + d(U_{i+1}^n - 2U_i^n + U_{i-1}^n), \quad 0 < i < i_m,$$

where $d = \frac{\nu \Delta t}{(\Delta y)^2}$ is the diffusion number. We next set $U_i^n \to e_i^n := \xi^n e^{l\theta i}$, where l is the imaginary unit and $\theta \in [-\pi, \pi]$. Hence for the FTCS scheme:

$$U_i^{n+1} \to \xi^{n+1} e^{l\theta i}, \quad U_{i-1}^n \to \xi^n e^{l\theta(i-1)}, \quad U_{i+1}^n \to \xi^n e^{l\theta(i+1)}$$

Plugging the above expressions into the scheme gives:

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

Use the relation $\cos(\theta) = \frac{e^{l\theta} + e^{-l\theta}}{2}$ to obtain:

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

The above relation is of the form:

$$\xi^{n+1} = G\xi^n, \quad \text{where} \quad G = 1 + 2d(\cos(\theta) - 1)$$

where G: *amplification factor*.

When is ξ^n not growing? When $|G| \leq 1$. Hence for stability in the Von Neumann sense we must have $1 - 2d(1 - \cos(\theta)) \leq 1, \quad 1 - 2d(1 - \cos(\theta)) \geq -1$ Therefore, for stability in the Von Neumann sense for FTCS we have:

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

The first is always true. The second gives $d(1 - \cos(\theta)) \leq 1$ Since $\cos(\theta) \leq 1$, the second relation gives the Von Neumann criterion for the FTCS method, which is:

$$\boxed{d \leq \frac{1}{2} \quad \text{or} \quad \frac{\nu \Delta t}{(\Delta y)^2} \leq \frac{1}{2}} \tag{8}$$

DuFort-Frankel

To start the Von Neumann stability analysis of the DuFort-Frankel method, let's first rewrite equation (5) as:

$$U_i^{n+1}(1 + 2d) = 2d(U_{i+1}^n + U_{i-1}^n) + U_i^{n-1}(1 - 2d)$$

where $d = \frac{\nu \Delta t}{(\Delta y)^2}$.

We will now examine a typical Fourier mode:

$$e_i^n = \xi^n e^{l\theta i}$$

Substituting the typical Fourier mode, e_i^n, for U_i^n in the finite difference equation, we get:

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

Dividing through by the common factor $\xi^n e^{l\theta i}$ gives:

$$(1 + 2d)\xi = 2d\left(e^{l\theta} + e^{-l\theta}\right) + (1 - 2d)\xi^{-1}$$

We now use the relationship for complex exponentials that $e^{l\theta} + e^{-l\theta} = 2\cos(\theta)$ and the definition of the growth factor, G, as ξ to rewrite as follows:

$$(1+2d)G = 4d\cos(\theta) + (1-2d)\frac{1}{G}$$

$$\Leftrightarrow (1+2d)G^2 - 4d\cos(\theta)G - (1-2d) = 0$$

Solving for G from the quadratic equation gives:

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

We simplify the term under the square root:

$$\begin{aligned} &16d^2\cos^2(\theta) + 4(1+2d)(1-2d) \\ &\qquad = 16d^2\cos^2(\theta) + 4(1-4d^2) \\ &\qquad = 4 - 16d^2 + 16d^2\cos^2(\theta) \\ &\qquad = 4 - 16d^2\sin^2(\theta) \end{aligned}$$

This leads to the solution for G:

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

G may be real or complex depending on if the sign of the square root argument is positive or negative. We will get a negative argument if

$$\begin{aligned} &1 - 4d^2\sin^2(\theta) < 0 \\ \Leftrightarrow \quad &d > \frac{1}{2\sin(\theta)} \end{aligned}$$

Stability depends on having $|G| \leq 1$. If G is complex, we must have $G^*G \leq 1$. Using the equation just derived for G, we obtain the following stability requirement for complex G.

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

The $\sin^2(\theta)$ term ranges from zero to one. Below we show that the stability condition is satisfied in both these limits; thus, it is satisfied for all values of d that produce a complex growth factor.

$$\textit{When } \sin^2(\theta) = 1, G^*G = \frac{1 - 4d^2}{(1+2d)^2} \leq 1$$

$$\textit{When } \sin^2(\theta) = 0, G^*G = \frac{1 + 4d^2}{(1+2d)^2} \leq 1$$

When d is less than or equal to $\frac{1}{2\sin(\theta)}$, the growth factor is real. In these cases, the argument of the square root is one minus a positive number. This positive number must be less than one to maintain a

real solution. Thus, the square root in the real case will always be less than or equal to one. The limiting condition will be when the square root term is +1. In this case, the growth factor inequality becomes:

$$G = \frac{1 + 2d\cos(\theta)}{1 + 2d} \leq 1$$

We see that this will be satisfied for any values of d since $\cos(\theta)$ is always less than or equal to one. Thus, the stability analysis shows that any value of d produces a growth factor whose magnitude is less than or equal to one, regardless of whether the growth factor is real or complex. We therefore conclude that the DuFort-Frankel method is **unconditionally stable**.

Laasonen

Recall from equation (6) the Laasonen discretised equation:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \nu \left(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta y)^2} \right)$$

Now set $_i^n$ to $e_i^n := \xi^n e^{l\theta i}$, and substitute to get:

$$\xi^{n+1} - \xi^n = \frac{\nu \Delta t}{(\Delta y)^2} \left[\xi^{n+1} e^{l\theta i} - 2\xi^{n+1} + \xi^{n+1} e^{-l\theta i} \right]$$

This leads to the amplification factor G:

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

where $d = \frac{\nu \Delta t}{(\Delta y)^2}$. Since d is always positive, the denominator will always be greater than or equal to 1, and so G will always have a magnitude less than or equal to 1. This proves that the Laasonen Implicit method is **unconditionally stable**.

Crank-Nicolson

Finally, we perform Von Neumann stability analysis on the Crank-Nicolson method. Using the given discretization of equation (7), we apply a Fourier transformation to the discretized equation, leading to a characteristic equation of the form:

$$-du_{i-1}^{n+1} + (2 + 2d)u_i^{n+1} - du_{i+1}^{n+1} = du_{i-1}^n + (2 - 2d)u_i^n + du_{i+1}^n$$

where $d = \frac{\nu \Delta t}{\Delta y^2}$.

We introduce a Fourier mode $u_i = \xi^n e^{l\theta i}$ into the equation, which following some simplifications, gives us the following amplification factor:

$$G = \frac{1 - d(1 - \cos\theta)}{1 + d(1 - \cos\theta)}$$

The magnitude of G must satisfy $|G| \leq 1$ for stability for any $\Delta t, \Delta y$. Since the nominator here is always less than or equal to the denominator, this condition is indeed met for all $\Delta t, \Delta y$, thus proving the **unconditional stability** of the scheme.

3.3.2 Summary

The von Neumann stability analysis applied to the numerical methods for simulating the flow of a viscous fluid between two parallel plates yields critical insights into their applicability. The FTCS explicit method's conditional stability for $d \leq \frac{1}{2}$ necessitates precise parameter selection to avoid numerical instability, particularly given the problem's specifics—like the fluid's kinematic viscosity and the plates' separation distance. This condition imposes limitations on time step size and spatial resolution, crucial for accurately capturing the fluid's velocity profile evolution.

Conversely, the unconditional stability of the DuFort-Frankel explicit, Laasonen implicit, and Crank-Nicolson implicit methods grants them a robust advantage, allowing for larger time steps without compromising stability. This attribute is especially valuable in simulations demanding high temporal resolution over extended periods.

Given the project's requirements—modeling fluid flow with a specified kinematic viscosity over a set distance between plates—the theoretical unconditional stability of the latter three methods provides a significant strategic advantage. It enables the exploration of a broader range of time steps and spatial discretizations without the risk of numerical instability, thereby facilitating a more flexible and potentially accurate investigation of the fluid's behavior under the given experimental conditions.

4 Results

In this results section, we present and discuss the findings from our simulations. This includes a look at how well each method performs against the analytical solution, an examination of the methods' stability, and a comparison of the errors and computational times with varying time steps using the Laasonen method. We aim to uncover the strengths and weaknesses of each numerical method, offering a clear view of which method works best under various conditions.

4.1 Simulations

We start the discussion of our results by presenting the velocity profiles produced by each of the methods for their respective experiments, as well as the "exact" velocity profile produced using the analytical solution. The below plots show the analytical velocity profile of the fluid between two parallel plates over time. The x-axis represents velocity in meters per second (m/s), and the y-axis represents height in meters (m). Each curve represents the velocity profile at a different time increment, starting from t=0.00s up to t=1.08s. Note that in order to ensure the plots aren't "overcrowded", we only show 10 of the velocity curves in each of the plots:

Analytical/Exact solution

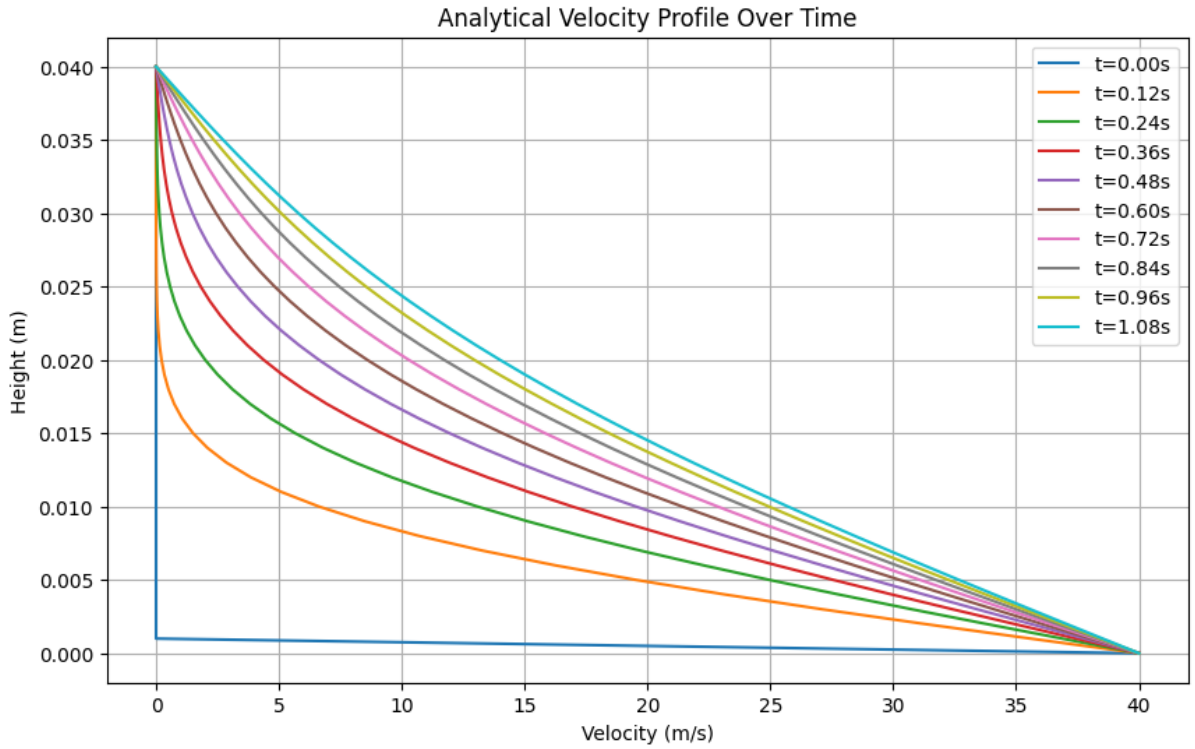

Figure 7: **Plot of the velocity profile using the Exact/Analytical solution**

FTCS

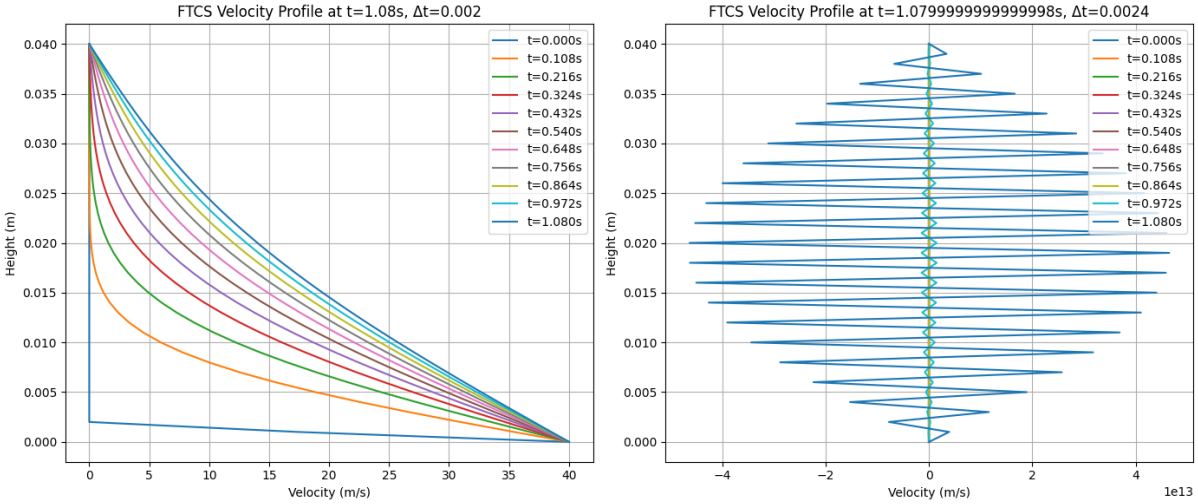

Figure 8: **Plot of Velocity profile using FTCS for each experiment. Left plot shows Experiment 1, Right plot shows Experiment 2**

Note that the d values (as presented in equation (8)) here are 0.434 and 0.520 for the 1st and 2nd experiments respectively.

DuFort-Frankel

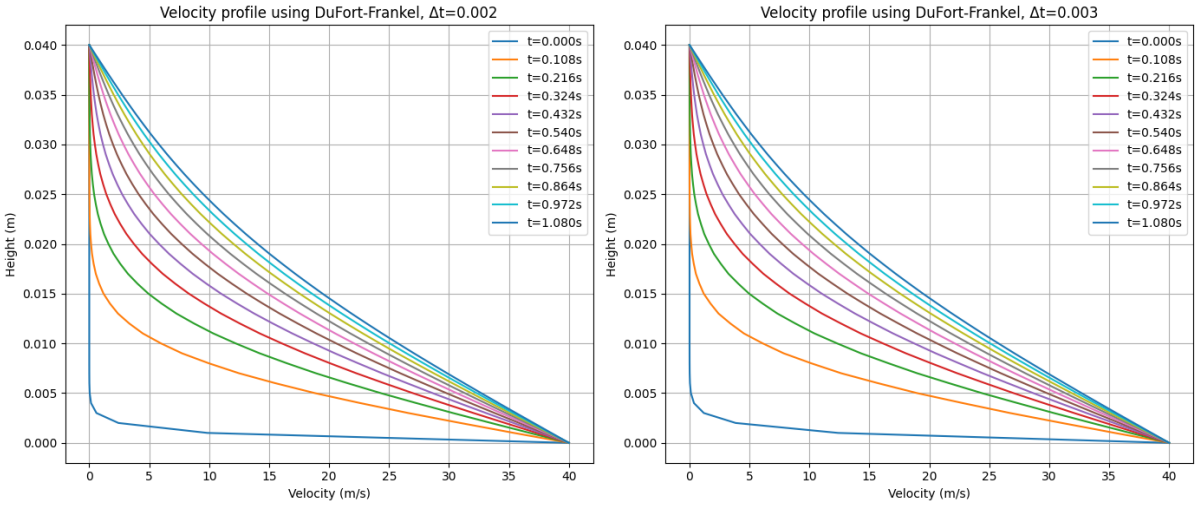

Figure 9: **Plot of Velocity profile using DuFort-Frankel for each experiment**

Laasonen

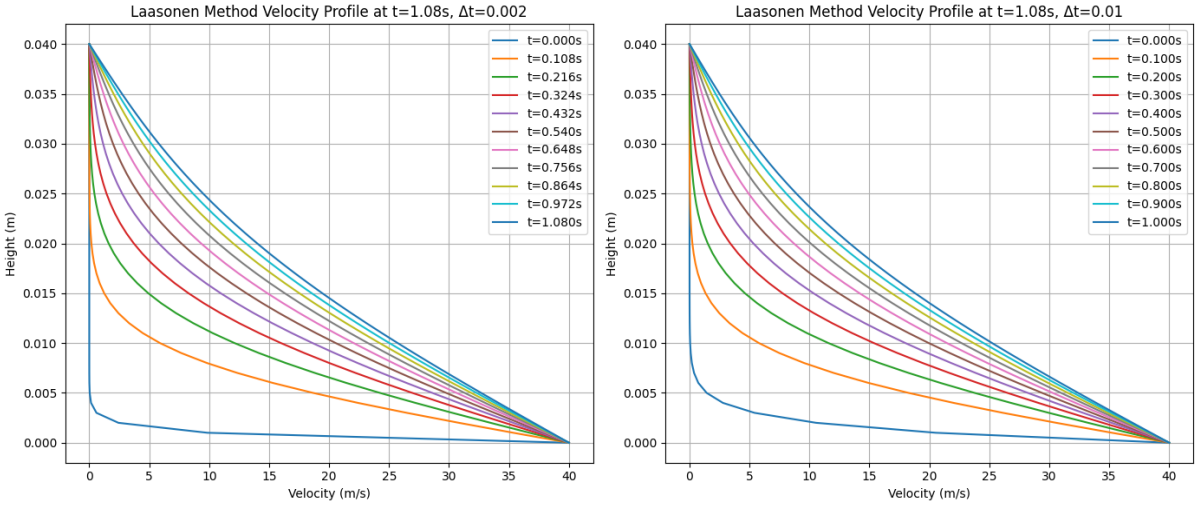

Figure 10: **Plot of Velocity profile using Laasonen for each experiment**

Crank-Nicolson

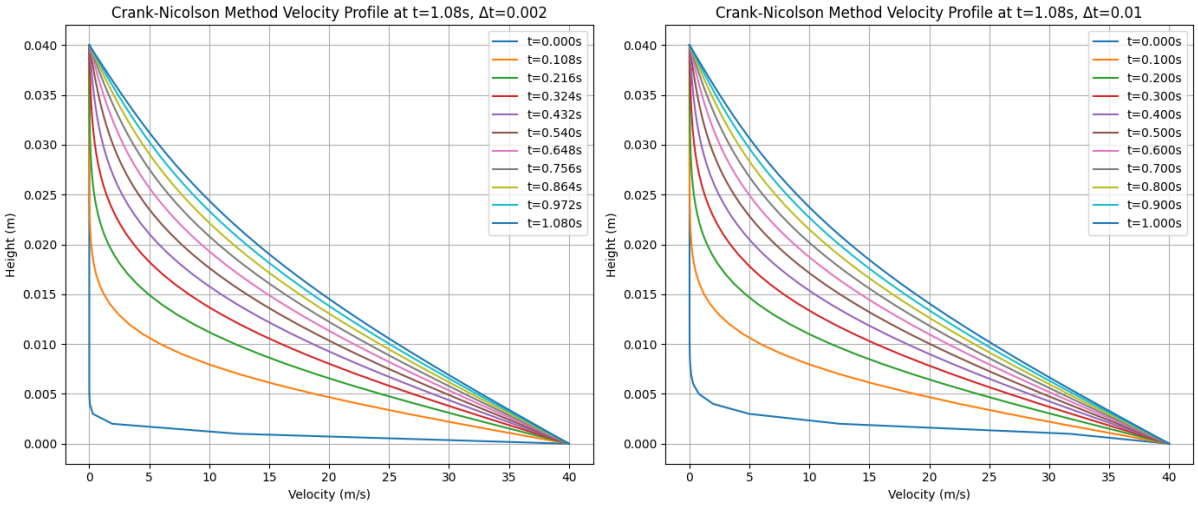

Figure 11: **Plot of Velocity profile using Crank-Nicolson for each experiment**

It can be observed that at t=0.00s, the velocity is zero across (almost) the entire height between the plates, indicating the fluid is initially at rest. At the very lowest height, the velocity of the fluid matches the plate velocity almost instantaneously because the fluid in direct contact with the moving plate is dragged along with it. As time progresses, the velocity increases, especially near the moving plate, showing the shear effect where fluid layers closer to the moving plate move faster due to less viscous resistance compared to layers further away. The velocity profiles are parabolic in nature, which is characteristic of laminar flow between plates. Over time, the profiles flatten out towards the top, indicating the development of the velocity boundary layer.

As expected in a fluid dynamics scenario involving sudden acceleration of one plate, the velocity is highest near the moving plate and decreases with height. The flattening of the curve over time suggests the fluid is approaching a *steady state* as it continues to be influenced by the moving plate.

Comments on stability From the above plots, we observe that almost all the velocity profiles produced by each of the methods/experiments seem to match that produced by the exact solution, and are also clearly stable. The one exception is experiment 2 of the FTCS method. From figure (8) we see that whilst experiment 1 looks stable, experiment 2 is clearly unstable as the velocity profile has become erratic/nonsensical. Given the d values used are 0.434 and 0.520 for experiment 1 and 2 respectively, this aligns with the theoretical constraints on stability derived using Von Neumann stability analysis in equation (8), since experiment 2 uses a d value that has exceeded the stability threshold. Each of the experiments for the other methods being stable also aligns with the stability analysis performed in section 3.3.1 - they were all expected to be unconditionally stable.

4.2 Errors

We now take a closer look at the accuracy of the methods by looking at the errors. The error term in terms of the solutions is given by:

$$\text{Analytical} - \text{Numerical}$$

Following we produce plots comparing the error terms for each of the simulations at times $t = 0.18$ and $t = 1.08$, using a timestep of 0.002:

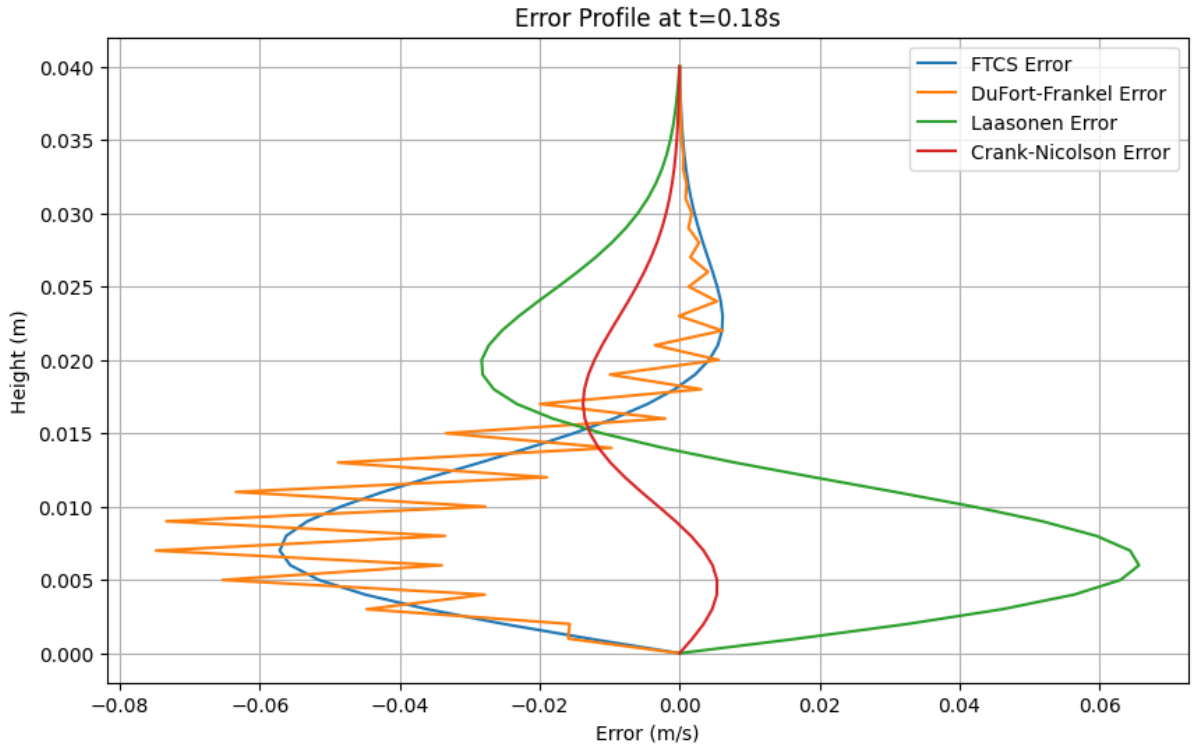

Figure 12: **Plot of the error profile of each method at t = 0.18**

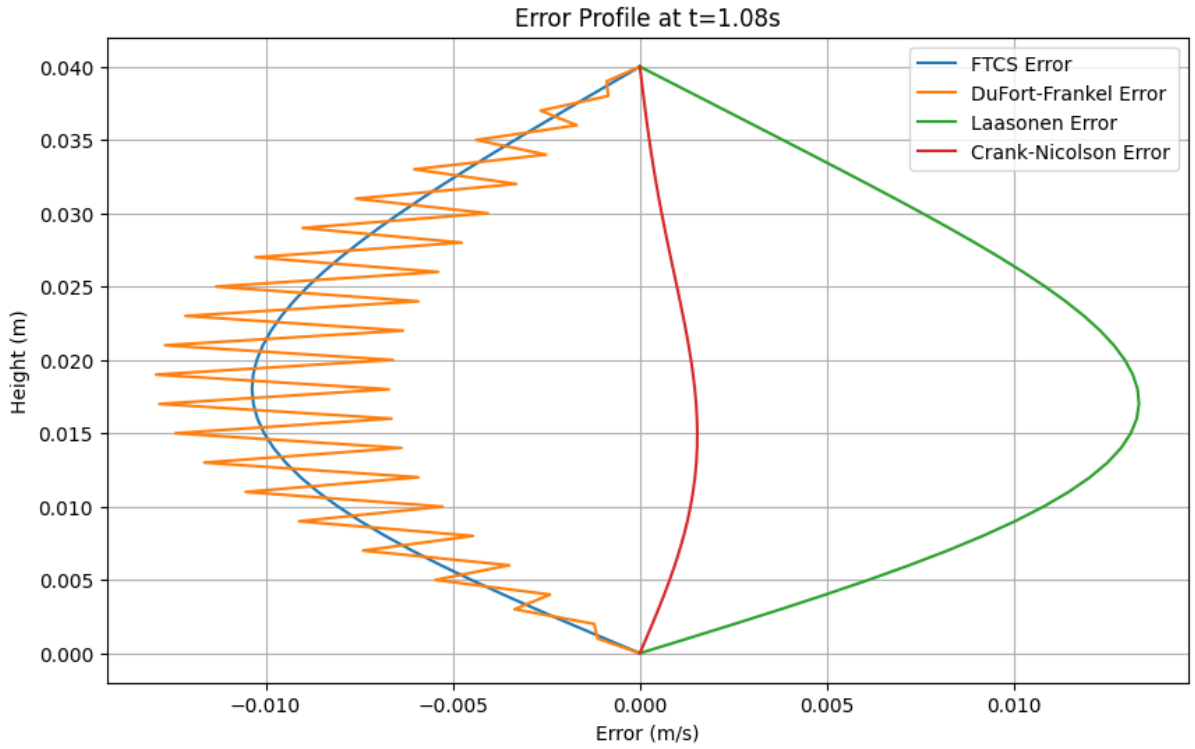

Figure 13: **Plot of the error profile of each method at t = 1.08**

Examining the first plot, it's evident that the error associated with the Crank-Nicolson method is notably minimal, in contrast, the DuFort-Frankel method displays the largest error (at every other height). The errors produced by both the FTCS and Laasonen methods fall in between, offering moderate accuracy.

Interestingly, we see an oscillatory looking pattern for the error of the DuFort-Frankel method. This can be attributed to its intrinsic calculation approach that utilizes information from two preceding time points (t^n and t^{n-1}) to estimate the subsequent point (u^{n+1}). This unique aspect of the DuFort-Frankel method, coupled with the steep gradient imposed by the initial condition at the start of the simulation, leads to an alternation between overestimation and underestimation of the calculated velocity at each step. This phenomenon also contributes to dispersion errors—where different frequency components travel at varying numerical speeds.

The error profile observed at t=1.08s reveals a coherent trend, albeit with certain distinctions. The profiles exhibit a more uniform pattern; the FTCS and DuFort-Frankel methods consistently overestimate the velocity across all heights, whereas the Laasonen and Crank-Nicolson methods consistently underestimate it. Additionally, the DuFort-Frankel method displays a recurring oscillatory behavior in its error profile. Notably, the peak error for each method seems to be reduced at this time step - we will verify this observation in the following section.

One final observation pertains to the variance in the oscillatory pattern exhibited by the DuFort-Frankel method at different time points. Specifically, at $t = 0.18$, the oscillations manifest more prominently at lower heights, whereas at $t = 1.08$, the oscillations appear more evenly distributed across the entire height range. This phenomenon can be attributed to the temporal evolution of errors within the numerical solution. At the earlier time point ($t = 0.18$), errors are more pronounced at lower heights due to the immediate impact of the initial condition and the method's inherent calculation dynamics. As the simulation progresses to $t = 1.08$, the errors have had more time to propagate throughout the domain, leading to a more uniform distribution of oscillations across different heights. This pattern underscores the in-

fluence of both the initial condition's steep gradient and the DuFort-Frankel method's unique approach to error propagation over time and across the spatial domain.

4.2.1 Quantifying the errors

In order to more concisely compare the method errors, we now quantitatively calculate the error using two error metrics: Mean Absolute Error (MAE) and the Infinity Norm. The MAE provides an average of the absolute errors across the domain, offering a general sense of the overall error magnitude, while the Infinity Norm error identifies the maximum deviation at any point in the domain, highlighting the worst-case scenario. The errors for each of the methods at the two time points, are summarised in the table below:

Method	Mean Absolute Error	Error under Infinity Norm
	at $t = 0.18$s	
FTCS	0.0160256935842377	0.057144253819650714
DuFort-Frankel	0.016084212629322595	0.07482460520977696
Laasonen	0.02129257509523537	0.06556606609411695
Crank-Nicolson	0.00517902431339108	0.013797202327673563
	at $t = 1.08$s	
FTCS	0.06355241780450687	0.010364048437978058
DuFort-Frankel	0.006040537086271375	0.01292817454989681
Laasonen	0.00804984553201206	0.01336337215208444
Crank-Nicolson	0.00804903015637779	0.0015350116698531716

Table 1: Summary of Errors at the two time points

From this table, we see mixed results for the Mean Absolute Error, with the error decreasing slightly at the later time-step for two of the methods yet increasing slightly for the other two. Notably though, the error under the infinity norm is in fact lower for each of the methods at the later time-step. This is to be expected - over time, as the system approaches a steady state or a more stable profile, the error measured by the Infinity Norm decreases because the sharp gradients that cause high initial errors are smoothed out. This decrease in error over time also confirms that the methods are stable for the given time steps here, since if they were unstable, the error would accumulate over time.

4.3 Time step in the Laasonen method

We will now take a look at the effects of varying the time step in the Laasonen method, displayed in the two figures below:

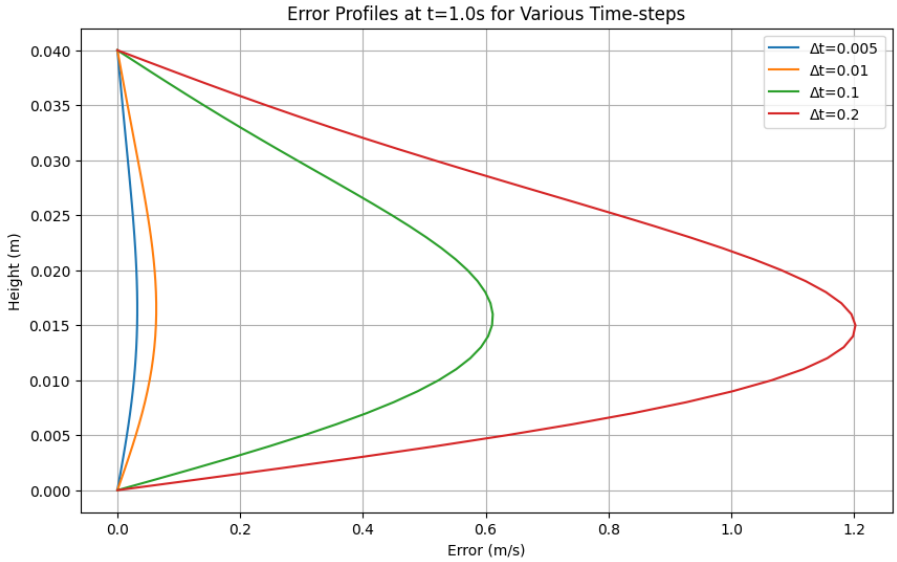

Figure 14: **Plot of the error profile for varying step sizes using the Laasonen method**

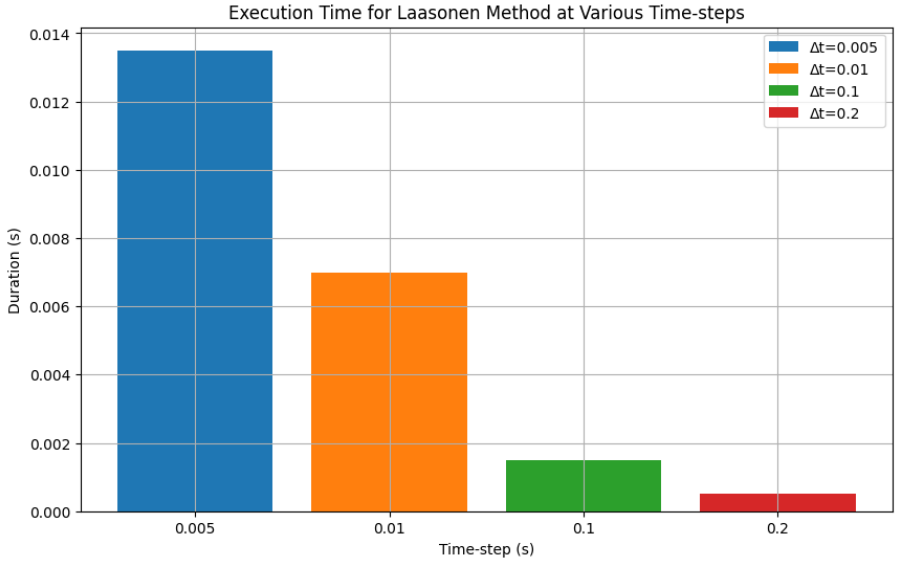

Figure 15: **Plot of the execution time for the Laasonen method using various time-steps**

From Figure 14 above, we see that as the step size is increased, the error of the Laasonen method increases significantly. This is to be expected - the step-size in numerical methods, such as the Laasonen method, dictates the temporal resolution of the solution. Smaller step-sizes, like $\Delta t = 0.005$, offer higher resolution and typically result in more accurate solutions, evidenced by smaller errors. Conversely, larger step-sizes, such as $\Delta t = 0.2$, yield less accurate results with higher errors, as the plot clearly shows. This occurs because a larger step-size may overlook subtle changes in the fluid's behavior, inadequately capturing the true dynamics of the system.

From Figure 15, we observe that the computational duration required by the Laasonen method diminishes markedly with increasing step size. Specifically, a smaller step size of $\Delta t = 0.005$ seconds results in the longest execution time, indicating a higher computational cost due to finer temporal resolution. As the step size escalates to $\Delta t = 0.2$ seconds, the execution time decreases significantly.

The two plots thus demonstrate the general principle that a finer step-size (smaller Δt) enhances computational accuracy but requires more computational effort.

5 Discussion and Conclusion

Our investigation into the numerical simulation of fluid dynamics between parallel plates has yielded insightful observations regarding the applicability, stability, and accuracy of the four studied methods. The FTCS method, while conditionally stable for $d \leq \frac{1}{2}$, exhibits limitations in flexibility due to its stringent stability criterion. In contrast, the unconditional stability of the DuFort-Frankel, Laasonen, and Crank-Nicolson methods enhances their applicability across a broader range of simulation parameters without compromising numerical stability.

The Crank-Nicolson method stands out for its minimal error, making it the preferred choice for scenarios where high accuracy is paramount. However, this method, along with the Laasonen implicit method, requires solving linear systems of equations at each time step, which can be computationally intensive. The DuFort-Frankel method, while unconditionally stable, introduces an oscillatory error that can affect accuracy, particularly in simulations with significant temporal or spatial variation.

In our detailed exploration of step-size variations within the Laasonen method, a discernible trade-off between computational time and error magnitude was observed. Specifically, larger step-sizes significantly reduced computation times but at the cost of increased errors, indicating a potential oversight of critical dynamics within the fluid's behavior. Conversely, smaller step-sizes, while computationally more demanding, yielded greater accuracy in the simulations. This nuanced interplay between computational time and precision underscores the importance of tailored step-size selection in alignment with the desired fidelity of the simulation outcomes. Such insights are invaluable when optimizing simulations for efficiency without unduly sacrificing accuracy.

Essentially, the choice of numerical method hinges on the specific requirements of the simulation task at hand. For high-accuracy needs and when computational resources are not a limiting factor, the Crank-Nicolson method is highly recommended. For scenarios prioritizing computational efficiency over extreme accuracy, the Laasonen implicit method offers a viable alternative. The FTCS method, with its conditional stability, may still be useful in controlled settings where its stability criteria can be satisfied. The DuFort-Frankel method's unconditional stability makes it suitable for a wide range of applications, although its potential for oscillatory errors warrants caution.

If I were researching this topic, I would likely lean towards using the Crank-Nicolson method for its notable balance between accuracy, stability, and computational efficiency. This method offers second-order accuracy in both time and space, which is advantageous for capturing the detailed dynamics of fluid behavior accurately. Although it might require more computational resources due to the necessity of solving linear systems at each timestep, its benefits in terms of stability and accuracy are substantial, making it a robust choice for comprehensive and precise fluid dynamics research.

References

[1] Irene Kyaza *MT5846 Lecture Notes*. University of St. Andrew's School of Mathematics and Statistics

[2] Ahmed N. Elmekawy https://drahmednagib.com/CFD$_2$018/$CFD_Lecture_4.pdf$