ME-GY 7153: Computational Fluid Mechanics & Heat Transfer. Semester Project 2 Report

Anand Lalasaheb Jagadale
Student ID: N19577204

Fall 2025

Contents

1 Problem Statement **3**
1.1 Checking if the Flow be modeled as Compressible: 3
1.2 Governing Equations 3
1.2.1 Assumptions: 3
1.2.2 Continuity Equation 3
1.2.3 Momentum Equation (x-direction) 4
1.2.4 Energy Equation 4

2 Numerical Method and Discretization **4**
2.1 Discretization of Momentum Equation 4
2.2 Discretization of Energy Equation 5
2.3 Grid Spacing : 5
2.3.1 Node Density Calculation : 6
2.4 Time Step (CFL condition) : 6
2.5 Boundary Conditions ($t > 0$) 7
2.5.1 Lower Plate Boundary Condition ($h = 0$) 7
2.5.2 Upper Plate Boundary Condition ($h = 0.1$ $inch$) 7

3 Results : **7**
3.1 Flow and Thermal Fields distribution : 7
3.1.1 Velocity distribution 7
3.1.2 Temperature distribution 10
3.2 Grid and Time Step Independence study : 12
3.2.1 Grid Independence : 12
3.2.2 Time Step Independence : 14
3.3 Comparison with Analytical Solution 15
3.3.1 Equation for Analytical Solutions : 15

3.3.2 Validation with Velocity Profile : 15
3.3.3 Error Distribution Analysis 15
3.3.4 Validation with Shear Stress : 17
3.4 Visualization of Momentum Diffusion : 18
3.4.1 Wall Shear Stress Hysteresis : 20

4 Discussion of Physics **20**
4.1 Compressibility Analysis 20
4.2 Velocity Profiles : 21
4.3 Temperature Profiles : 21
4.4 Analysis of Wall Shear Stress Hysteresis : 22
4.5 Momentum Diffusion : 22

5 Conclusion **22**

1 Problem Statement

Consider the flow of a fluid constrained between 2 infinitely long parallel plates. The lower plate is stationary and the upper plate moves at speed U_∞ under the action of an external force. The pressure is uniform throughout the fluid, e.g. the problem of Couette Flow. The gap between the two plates is 0.01 inch. The fluid between the plates is air and the upper plate is oscillating with a maximum velocity of 200 ft/s. The temperature of both plates is $T_w = 519°\text{R}$.

1.1 Checking if the Flow be modeled as Compressible:

To check if we need to model the simulation as compressible or incompressible we need to check the Mach number of the flow. The speed of sound c and Mach number Ma are calculated as:

$$c = \sqrt{\gamma R T} = \sqrt{1.4 \cdot 1716 \cdot 519} \approx 1116 \text{ ft/s} \tag{1}$$

$$Ma = \frac{U_\infty}{c} = \frac{200}{1116} \approx 0.18 \tag{2}$$

Since $Ma < 0.3$, the flow should be modeled as **Incompressible** and thus with constant properties (ρ, μ, k, c_p).

1.2 Governing Equations

Starting from the full Navier-Stokes equations, we apply the assumptions to derive the reduced forms.

1.2.1 Assumptions:

1. 2D Flow imply $\frac{\partial}{\partial z} = 0$.
2. Flow is fully developed ($\frac{\partial(\cdot)}{\partial x} = 0$). Variations occur only in y.
3. Constant Properties: ρ, μ, k do not vary with temperature.
4. No Pressure Gradient: $\frac{\partial P}{\partial x} = 0$.
5. No slip condition.
6. Body forces are negligible.

1.2.2 Continuity Equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) = 0 \tag{3}$$

For incompressible flow ($\rho = \text{const}$) and 1D flow ($v = 0, w = 0$), therefore this simplifies to $\frac{\partial u}{\partial x} = 0$.

While Continuity equation reduces to zero, without this mathematical proof we cannot assume the reductions we are going to perform in the next sections.

1.2.3 Momentum Equation (x-direction)

$$\rho\left(\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}\right)=-\frac{\partial P}{\partial x}+\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right) \tag{4}$$

Applying assumptions ($v=0$, $\frac{\partial P}{\partial x}=0$, $\frac{\partial}{\partial x}=0$), the equation reduces to the diffusion equation:

$$\underbrace{\frac{\partial u}{\partial t}}_{\text{Rate of change of Velocity}} = \underbrace{\nu\frac{\partial^2 u}{\partial y^2}}_{\text{Diffusion Term}} \tag{5}$$

where $\nu=\mu/\rho$ is the kinematic viscosity.

All the components in the y- & z-direction are reduced to zero.

Momentum equation tells us how velocity will change over the gap distance, this is a reduction of the Navier-Stokes equation.

1.2.4 Energy Equation

$$\rho c_p\left(\frac{\partial T}{\partial t}+\vec{V}\cdot\nabla T\right)=k\nabla^2 T+\Phi \tag{6}$$

Reducing terms yields:

$$\underbrace{\rho c_p\frac{\partial T}{\partial t}}_{\text{Rate of Change of Energy}} = \underbrace{k\frac{\partial^2 T}{\partial y^2}}_{\text{Conduction term}} + \underbrace{\mu\left(\frac{\partial u}{\partial y}\right)^2}_{\text{viscous dissipation}} \tag{7}$$

While for small mach number and incompressible model of flow the temperature change over the gap distance is very small. Energy equation tells us how Temperature will change over the gap distance. This equation is also called as energy balance equation, and tells us how energy will change in the fluid over time.

2 Numerical Method and Discretization

The governing equations are solved using a Explicit Finite Difference approach. Superscript '^{n}' denotes current time step & Subscript '$_i$' denotes current grid node. The domain h is divided into N grid points with spacing Δy. A time step of Δt is selected using a stabality criteria. We will discuss more about this in the following sections.

2.1 Discretization of Momentum Equation

Starting with our governing momentum equation ie.

$$\underbrace{\frac{\partial u}{\partial t}}_{\text{Rate of change of Velocity}} = \underbrace{\nu\frac{\partial^2 u}{\partial y^2}}_{\text{Diffusion Term}} \tag{8}$$

We use Forward Difference for time (First Order) and Central Difference for space (Second Order):

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \nu \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta y^2} \tag{9}$$

Rearranging to solve for the next time step:

$$u_i^{n+1} = u_i^n + \frac{\nu \Delta t}{\Delta y^2} \left(u_{i+1}^n - 2u_i^n + u_{i-1}^n \right) \tag{10}$$

2.2 Discretization of Energy Equation

Similarly for the energy equation: We start with the governing energy equation;

$$\underbrace{\rho c_p \frac{\partial T}{\partial t}}_{\text{Rate of Change of Energy}} = \underbrace{k \frac{\partial^2 T}{\partial y^2}}_{\text{Conduction term}} + \underbrace{\mu \left(\frac{\partial u}{\partial y} \right)^2}_{\text{viscous dissipation}} \tag{11}$$

Again, We use Forward Difference for time (First Order) and Central Difference for space (Second Order):

$$\rho c_p \frac{T_i^{n+1} - T_i^n}{\Delta t} = k \frac{(T_{i+1}^n - 2T_i^n + T_{i-1}^n)}{\Delta y^2} + \mu \left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta y} \right)^2 \tag{12}$$

Rearranging to solve for the next time step:

$$T_i^{n+1} = T_i^n + \frac{\alpha \Delta t}{\Delta y^2} (T_{i+1}^n - 2T_i^n + T_{i-1}^n) + \frac{\mu \Delta t}{\rho c_p} \left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta y} \right)^2 \tag{13}$$

Here, 'α' is the Thermal diffusivity ie. $\alpha = \frac{k}{\rho c_p}$

2.3 Grid Spacing :

Mathematically, the simulation should be grid independent as along as we have more than 3 nodes, this is because the minimum required nodes to run the explicit finite difference calculation with central difference is 3, but it is not sufficient to capture the physics of the flow, we are simulating an oscillating viscous flow, where the flow must be defined effectively for the Stokes boundary layer, the thickness of which (δ) is approximated by [3]:

$$\delta \approx \sqrt{\frac{2\nu}{\omega}} \tag{14}$$

where ν is the kinematic viscosity and $\omega = 2\pi f$ is the angular frequency. This is the characteristic length scale of the flow.

For our simulation, where the frequency is (f = 100 Hz), the thickness is calculated to be approximately $\delta \approx 0.00070$ ft. To capture the physics of the oscillating flow, the grid spacing (Δy) must be sufficiently small to place multiple computational nodes within the boundary layer region.

2.3.1 Node Density Calculation :

For our problem the gap between the plates $h = 0.01$ in (≈ 0.00083 ft) and a assumed grid size of $N = 21$, the uniform grid spacing is:

$$\Delta y = \frac{h}{N-1} \approx 0.0000415 \text{ ft} \tag{15}$$

The number of nodes which will be within the Stokes boundary layer (N_{BL}) can be estimated as:

$$N_{BL} = \frac{\delta}{\Delta y} \approx \frac{0.00070}{0.0000415} \approx 16 \text{ nodes} \tag{16}$$

By following recommendations given by [2] for a good simulation we typically require number of nodes in boundary layer $N_{BL} \geq 10$, a resolution of $N_{BL} \approx 16$ is sufficient to capture the curvature of the velocity & Temperature profile and also maintain numerical stability. A significantly coarser grid (e.g., $N = 5$, yielding $N_{BL} \approx 3$) will be computationally cheap but would fail to resolve the curvature, leading to significant numerical error or instability.
For this project to demonstrate a smooth curvature and as there are no constraints on computational resources we select the grid size of $N = 51$.

2.4 Time Step (CFL condition) :

Explicit finite difference schemes are conditionally stable, unlike the grid spacing (Δy), the time step size (Δt) is required to be constrained to prevent numerical instability.

This constraint is derived from von Neumann stability analysis. For the momentum and the energy equations the recommended CFL is 0.5 *or* $\frac{1}{2}$ [2], the stability parameter (diffusion number, d) must satisfy:

$$d = \frac{\nu \Delta t}{(\Delta y)^2} \leq \frac{1}{2} \ ; \ d = \frac{\alpha \Delta t}{(\Delta y)^2} \leq \frac{1}{2} \tag{17}$$

Rearranging Eq. (17), the maximum allowable time step is:

$$\Delta t \leq \frac{1}{2} \frac{(\Delta y)^2}{\max(\nu, \alpha)} \tag{18}$$

This relationship indicates that as the grid spacing is refined, the required time step decreases drastically ($\Delta t \propto \Delta y^2$). To ensure more stability, the time step Δt is restricted by a safety factor:

$$\Delta t \leq \min\left(\frac{\Delta y^2}{2\nu}, \frac{\Delta y^2}{2\alpha}\right) \times \text{Safety Factor } (0.9) \tag{19}$$

2.5 Boundary Conditions $(t > 0)$

2.5.1 Lower Plate Boundary Condition $(h = 0)$

The Lower plate is maintained at a constant Temperature & Velocity :

$$T(0) = 519°\mathrm{R}\ ;\ u(0) = 0ft/s \tag{20}$$

2.5.2 Upper Plate Boundary Condition $(h = 0.1\ inch)$

The Upper plate is also maintained at a constant Temperature

$$T(N) = 519°\mathrm{R} \tag{21}$$

But the Velocity is defined as :

$$u(N) = U_\infty \sin(\omega t) \tag{22}$$

This gives us the required oscillating plate with initial velocity of $U_\infty = 200ft/s$.

3 Results :

3.1 Flow and Thermal Fields distribution :

The problem we tried to simulate is commonly known as stokes second problem and it describes the changes in a flow field due to oscillating plate driven by a sinusoidal velocity. Due to the periodic nature of the boundary condition, the flow field and the resulting thermal field show cyclic behaviors.

3.1.1 Velocity distribution

The motion of the upper plate is governed by:

$$u(t) = U_\infty \sin(\omega t) \tag{23}$$

The maximum velocity magnitude $|u| = U_\infty$ occurs twice per oscillation cycle where the value of the term $\sin(\omega t) = 1$:

- **Positive Peak:** $\omega t = \frac{\pi}{2}$ (or $t = 0.25$ Period)
- **Negative Peak:** $\omega t = \frac{3\pi}{2}$ (or $t = 0.75$ Period

The Velocity distribution shows cyclic behavior between these peaks. The resulting profiles for the velocity distribution over 1 cycle for different Frequency (100, 500, 1000, 2000 Hz) are shown in 1234

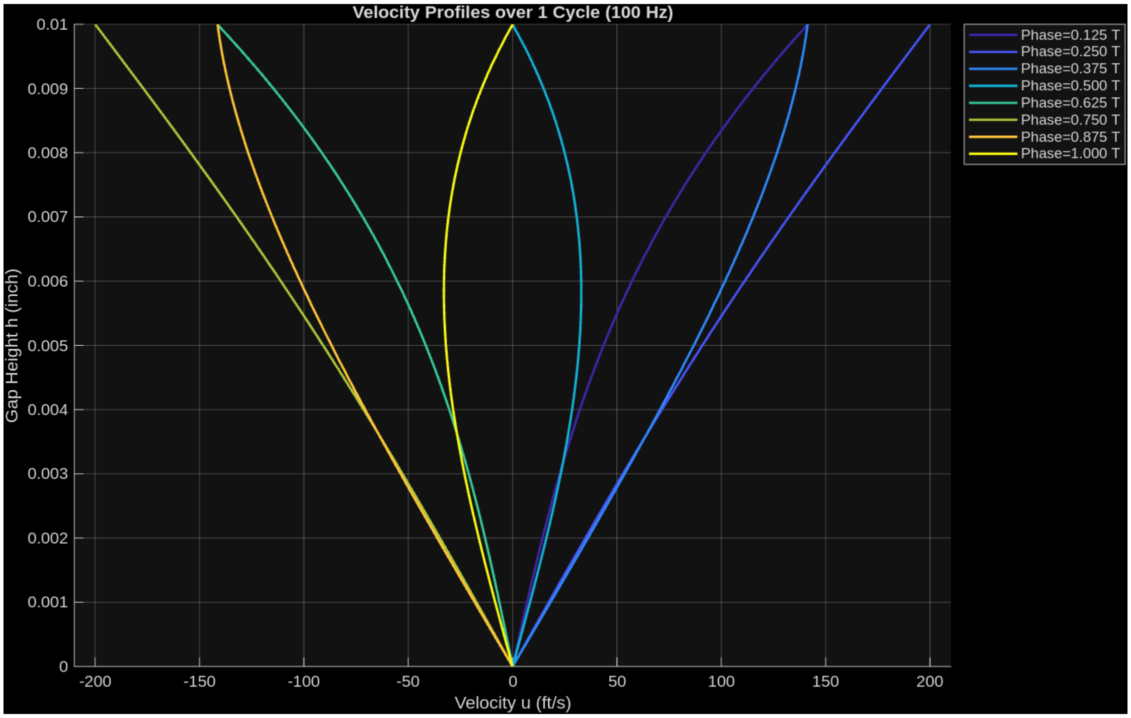

Figure 1: Velocity Distribution over 1st cycle ($U_\infty = 200 ft/s$; $Freq = 100 Hz$)

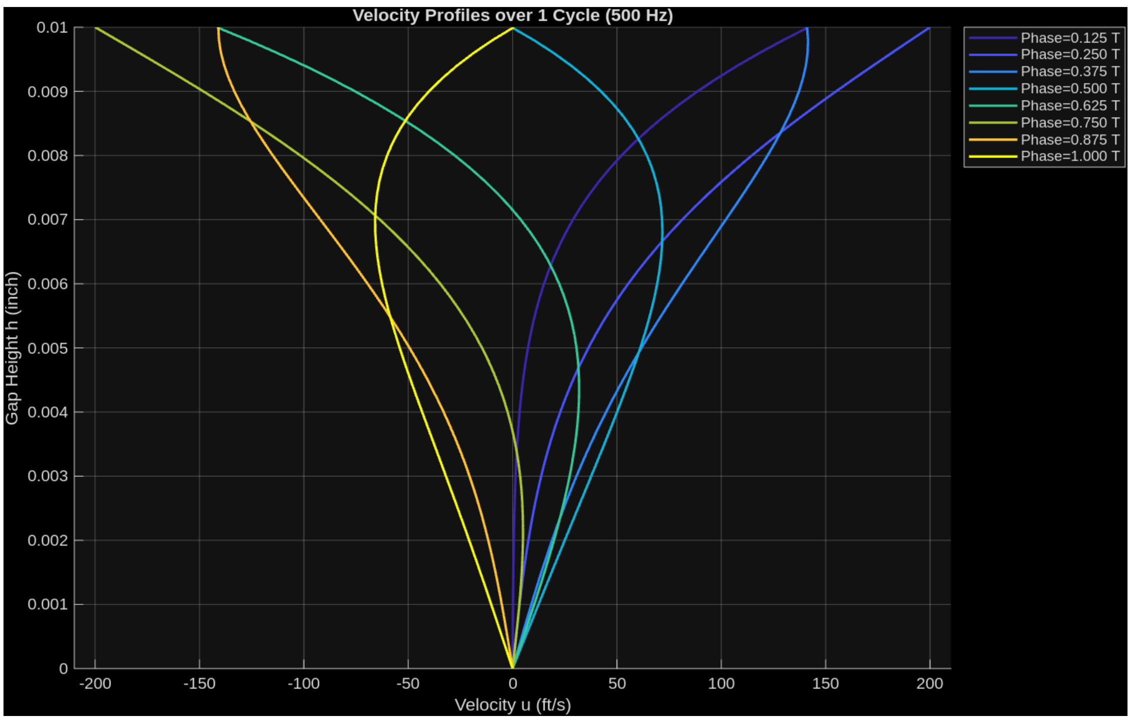

Figure 2: Velocity Distribution over 1st cycle ($U_\infty = 200 ft/s$; $Freq = 500 Hz$)

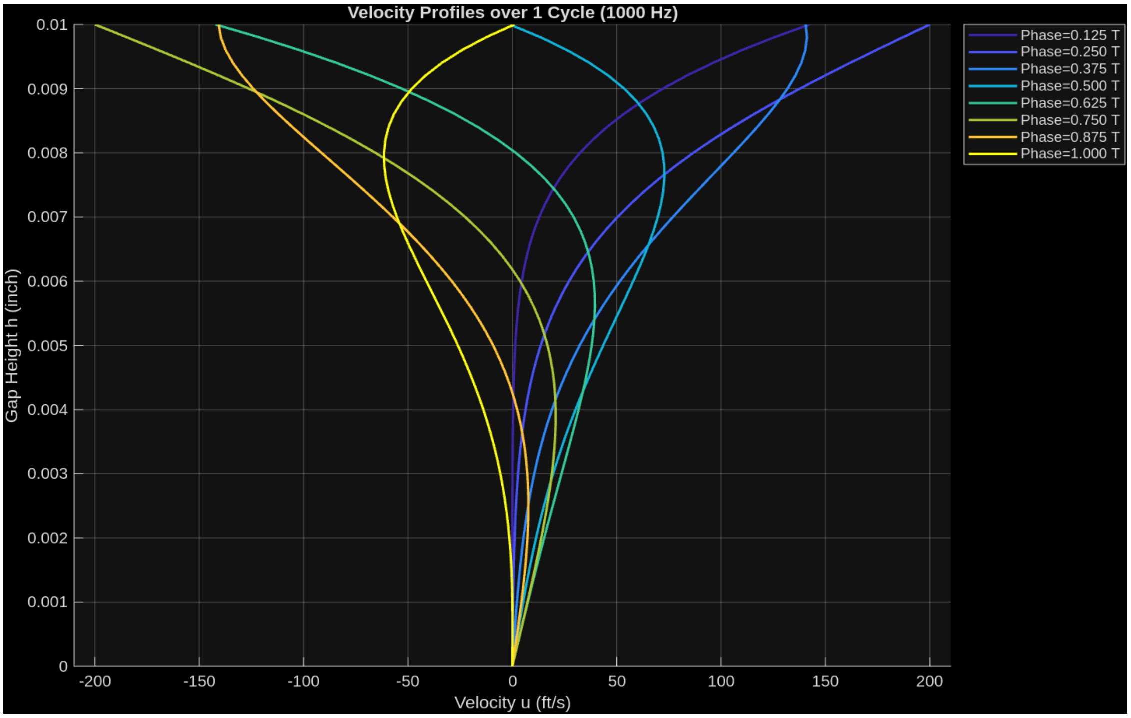

Figure 3: Velocity Distribution over 1st cycle ($U_\infty = 200 ft/s$; $Freq = 1000 Hz$)

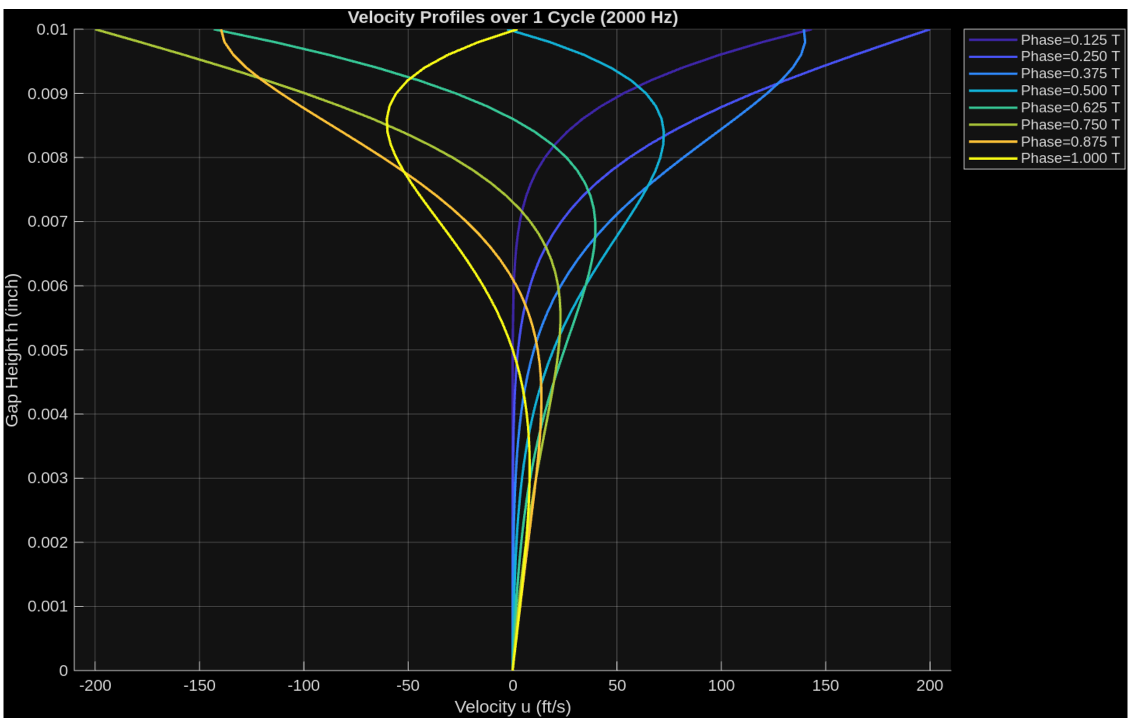

Figure 4: Velocity Distribution over 1st cycle ($U_\infty = 200 ft/s$; $Freq = 2000 Hz$)

3.1.2 Temperature distribution

The temperature rise within the fluid is driven by viscous dissipation (Φ), which acts as the source term in the energy equation. For a 1D flow, we defined this term as:

$$\Phi = \mu \left(\frac{\partial u}{\partial y} \right)^2 \tag{24}$$

This term depends on the *square* of the velocity gradient ie $(\partial u/\partial y)^2$, therefore the heat generation is independent of the flow direction. Consequently, viscous heating occurs at both the positive peak ($+U_\infty$) and the negative peak ($-U_\infty$).
The resulting profiles for the temperature distribution over 1 cycle for different Frequency (100, 500, 1000, 2000 Hz) are shown in 5678

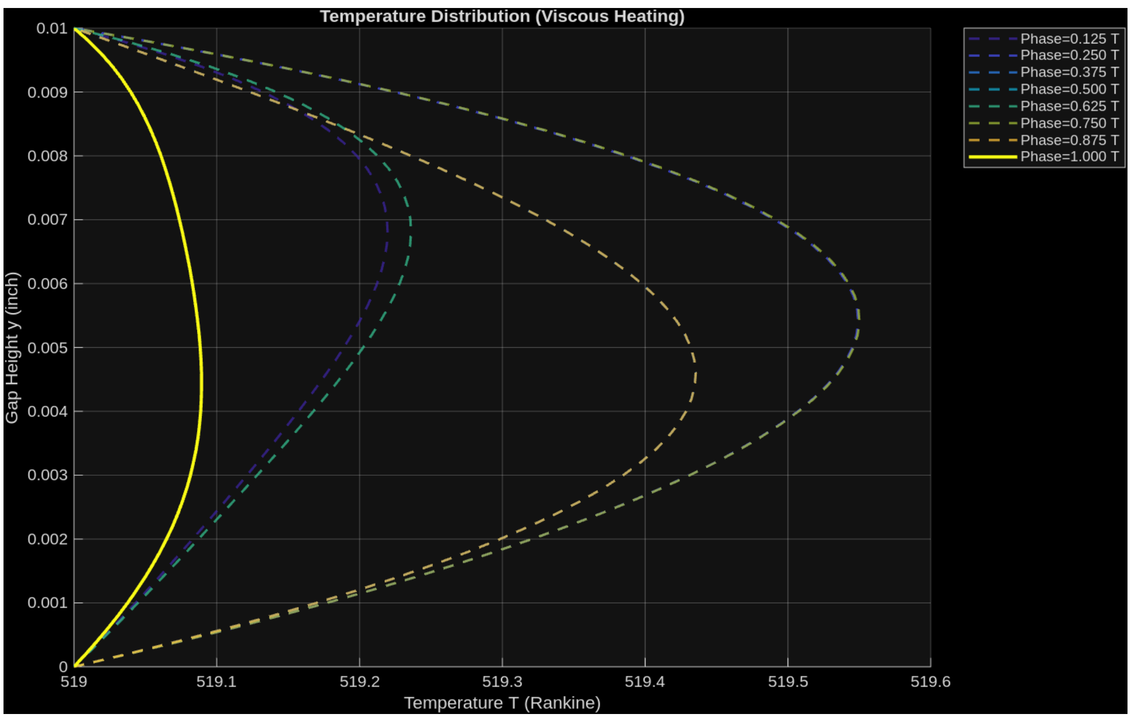

Figure 5: Temperature Distribution over 1st cycle ($U_\infty = 200 ft/s$; $Freq = 100 Hz$)

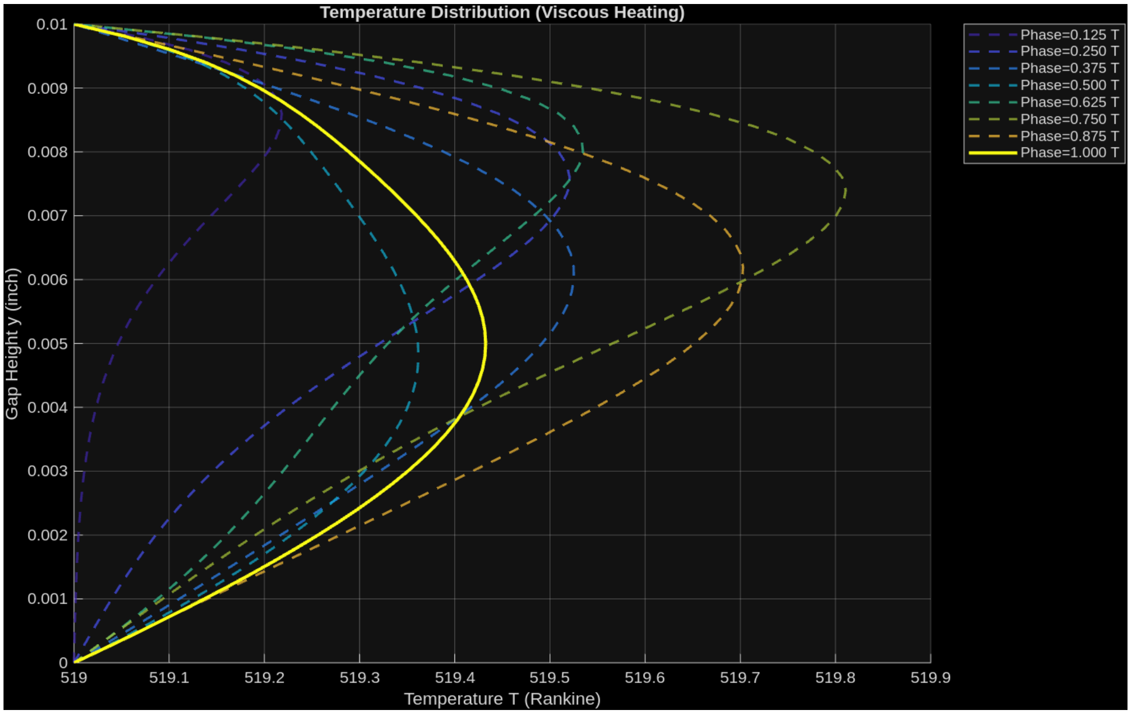

Figure 6: Temperature Distribution over 1st cycle ($U_\infty = 200ft/s$; $Freq = 500Hz$)

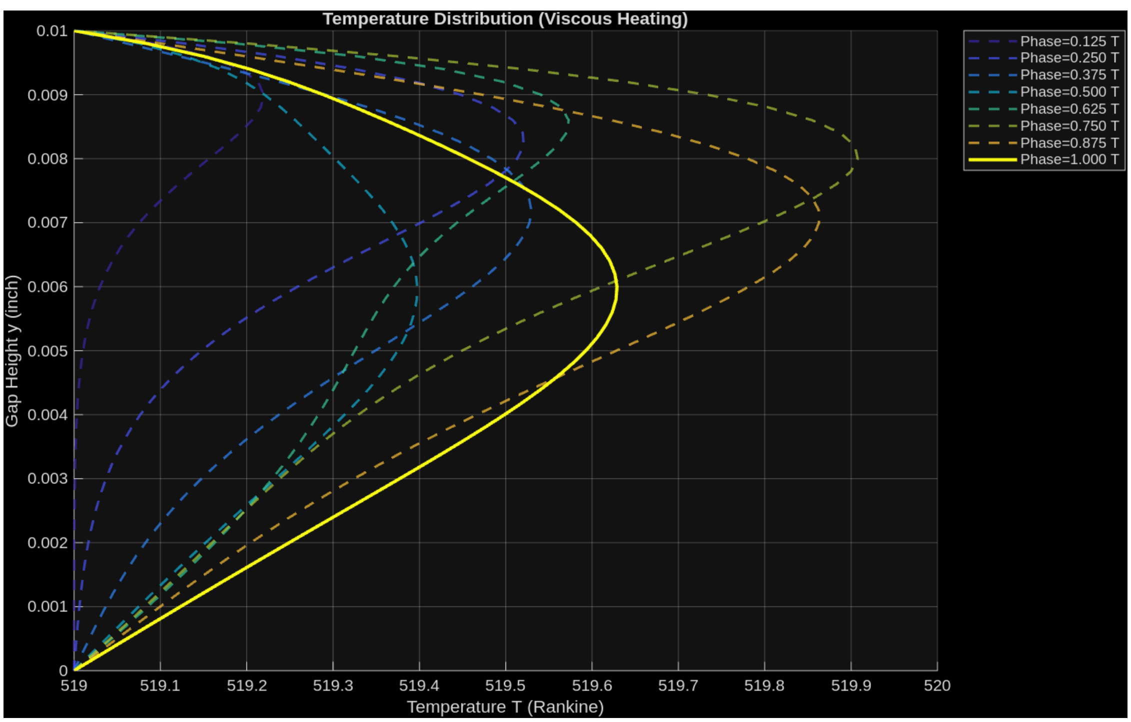

Figure 7: Temperature Distribution over 1st cycle ($U_\infty = 200ft/s$; $Freq = 1000Hz$)

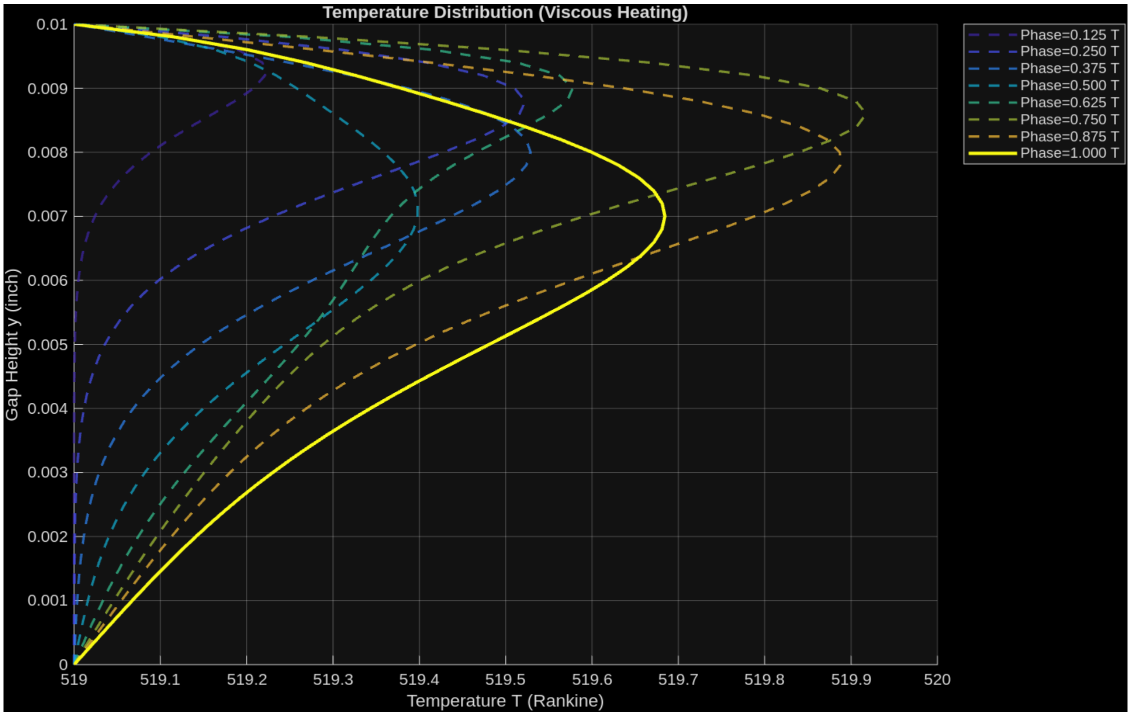

Figure 8: Temperature Distribution over 1st cycle ($U_\infty = 200ft/s$; $Freq = 2000Hz$)

3.2 Grid and Time Step Independence study :

To analyze how different grid size and time step affect the simulation flow field details like Velocity and Temperature are extracted at the positive peak of the oscillation at $t = 1.25 \times Period$.

This specific time step corresponds to the positive velocity peak in the second cycle. We choose the time from the second cycle ($t > 1.0P$) over the first cycle ($t = 0.25P$) to minimize the effects of the initial transient state, in the second cycle ($t > 1.0P$) the solution begins to settle into a quasi-steady periodic state.

3.2.1 Grid Independence :

To analyse the simulations grid independence, at the $t = 1.25 \times Period$ time step, by selecting a Frequency ($Freq = 1000Hz$) and Initial velocity ($U_\infty = 200ft/s$) the velocity distribution and temperature distribution profile was plotted using increasing grid size, the plots for this is given by 910

The grid independence study demonstrates the exact problem that we addressed in the previous section. It can be seen that while a coarse grid $N = 11$ can work but due to the nature of the stokes boundary layer, it is not a good choice, after selecting a sufficiently finer grid $N = 51$ we can see that the changes in grid size do not affect the overall results and the consecetive lines overlap each other.

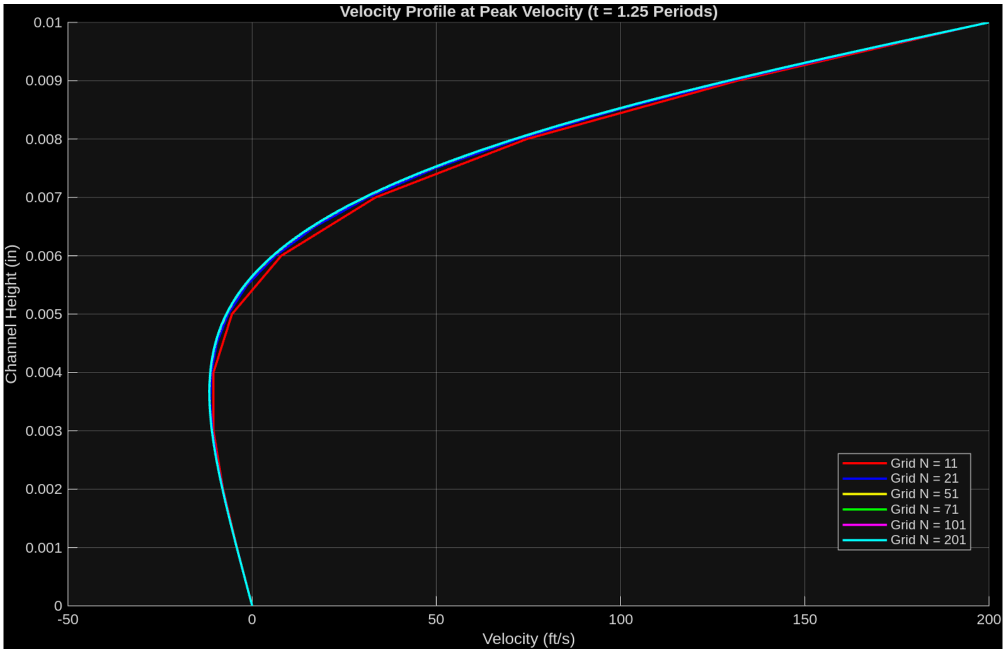

Figure 9: Velocity Distribution at $t = 1.25 \times Period$ for increasing Grid space.

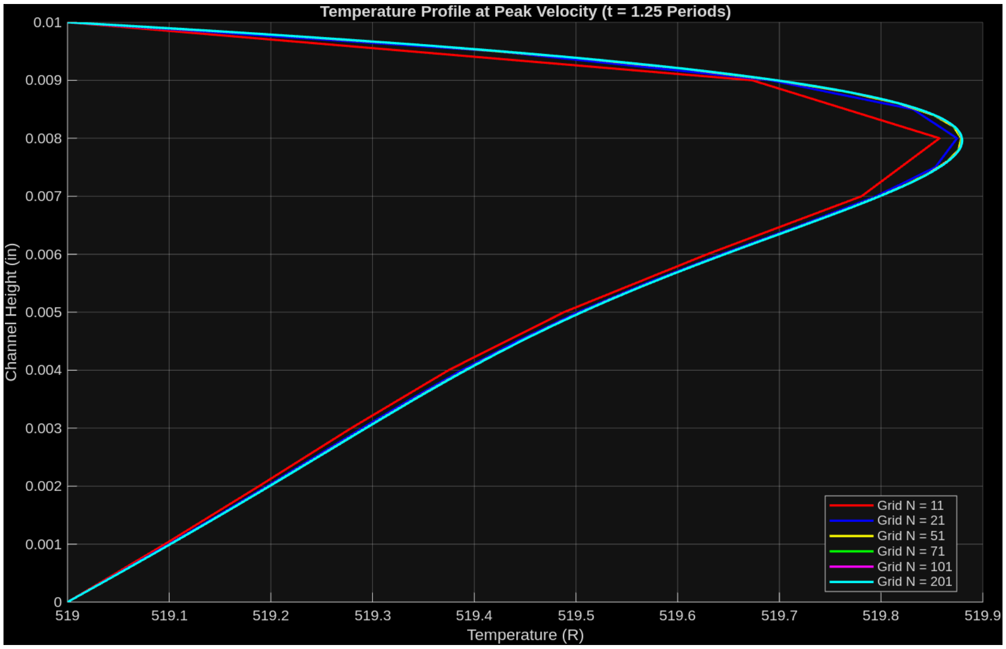

Figure 10: Temperature Distribution at $t = 1.25 \times Period$ for increasing Grid space.

3.2.2 Time Step Independence :

Similarly for the Time step independence study using the same time step, and selecting a Frequency ($Freq = 1000Hz$) with Initial velocity ($U_\infty = 200ft/s$) the velocity distribution and temperature distribution profile was plotted using decreasing safety factor from the CFL condition, the plots for this is given by 1112 . Here the plots perfectly overlap each other showing time step independence of the simulation for the selected CFL condition.

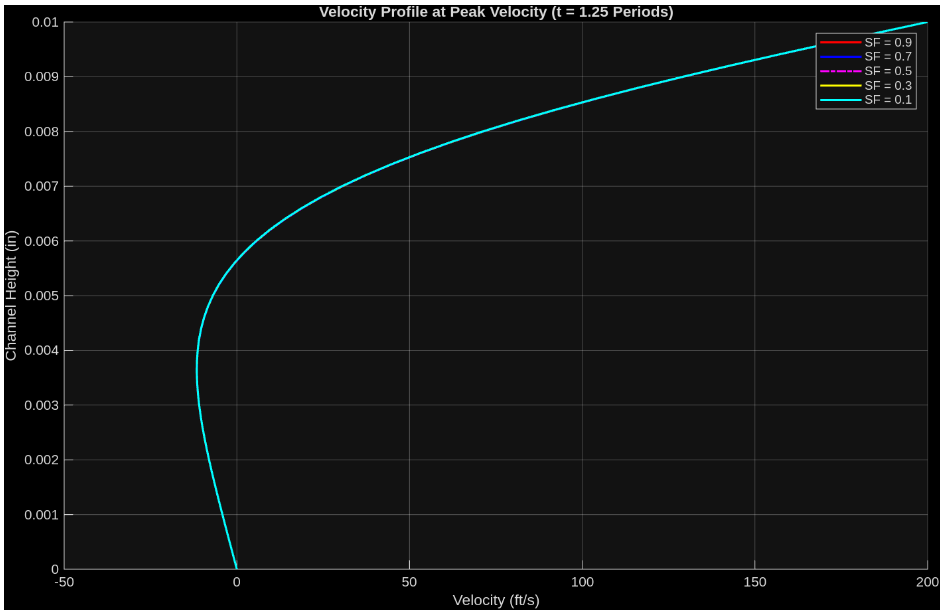

Figure 11: Velocity Distribution at $t = 1.25 \times Period$ for decreasing Safety factor.

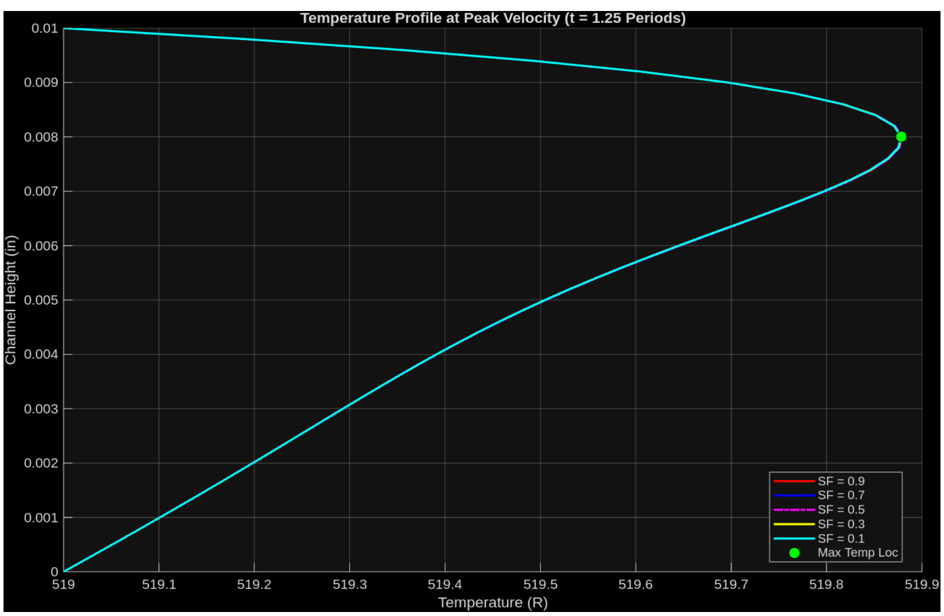

Figure 12: Temperature Distribution at $t = 1.25 \times Period$ for decreasing Safety factor.

3.3 Comparison with Analytical Solution

To validate the accuracy of the finite difference simulation, we compared the numerical results against the exact analytical solution for Stokes Second Problem (modification of the couette flow problem using a oscillating plate at one side). For this validation, the velocity field and the resulting shear stress distribution at a phase of $t = 1.25$ Periods were compared with the analytical solutions.

3.3.1 Equation for Analytical Solutions :

The exact solution for the velocity profile $u(y,t)$ due to a plate oscillating with velocity $U_\infty \sin(\omega t)$ is given by [1]:

$$u_{exact}(y,t) = U_\infty e^{-ky} \sin(\omega t - ky) \tag{25}$$

where k is the decay constant (inverse of the Stokes boundary layer thickness):

$$k = \sqrt{\frac{\omega}{2\nu}} \tag{26}$$

The analytical solution for the shear stress distribution, $\tau_{exact}(y,t)$, is derived by differentiating the velocity profile with respect to the vertical coordinate ($\tau = \mu \frac{\partial u}{\partial y}$) :

$$\tau_{exact}(y,t) = \mu U_\infty k e^{-ky} \left[\sin(\omega t - ky) + \cos(\omega t - ky)\right] \tag{27}$$

3.3.2 Validation with Velocity Profile :

The velocity profile we get from our simulation was compared to Eq. (25) at the peak positive velocity ($t = 1.25T$). The results from the analytical solution and the simulations are plotted on top of each other and are given by 13.

To calculate the accuracy, the Root Mean Square Error (RMSE) was calculated:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(u_{CFD,i} - u_{exact,i})^2} \approx 1.9778 \text{ ft/s} \tag{28}$$

Given the initial velocity $U_\infty = 200$ ft/s, we get a relative error of approximately **0.98%**. Therefore our simulation is **99.02%** accurate, This minimal error confirms that the grid resolution ($N = 51$) and time step condition are sufficient to capture the physics accurately.

3.3.3 Error Distribution Analysis

An analysis of the error distribution 14 reveals that the numerical error is not uniform across the channel. The error magnitude is maximum near the stationary plate ($y = 0$) but decreases as we approach the oscillating plate ($y = h$). This behavior is due to the use of Dirichlet boundary condition at the stationary plate, therefore the simulation fails to capture the changes in velocity at the stationary plate boundary accurately.

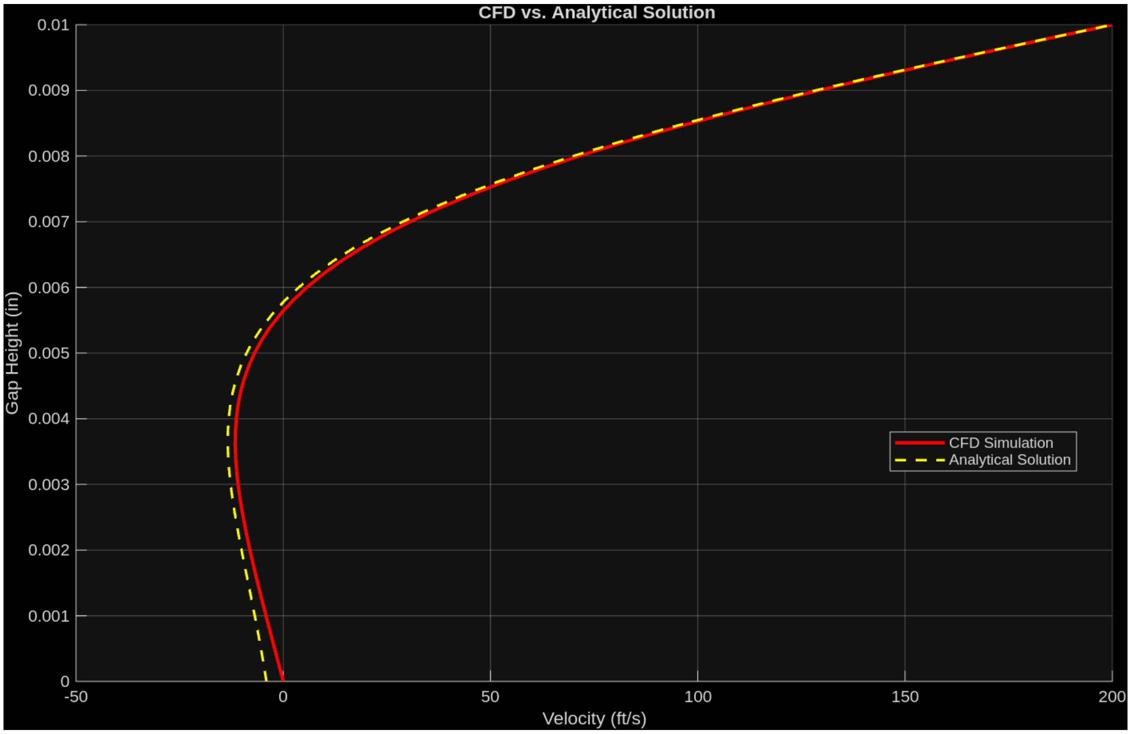

Figure 13: Analytical VS Simulation Velocity Profile at $t = 1.25T$

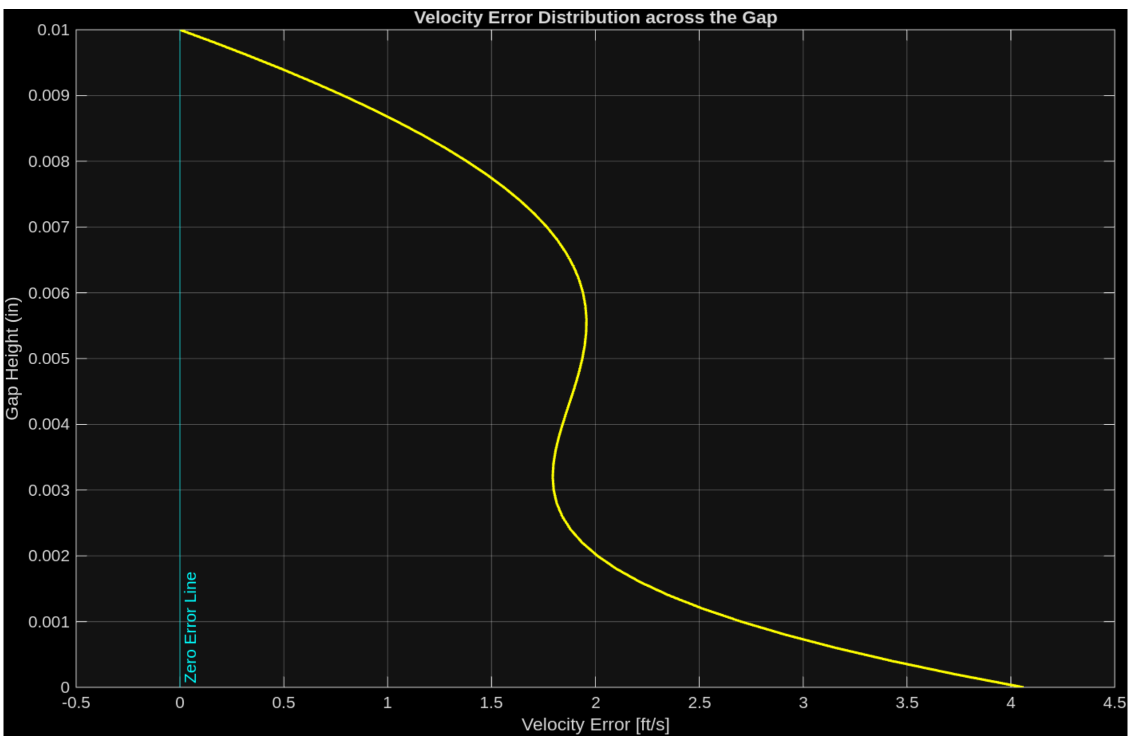

Figure 14: Distribution of error in velocity profile (Analytical VS Simulation)

3.3.4 Validation with Shear Stress :

Shear stress is a more rigorous test of simulation accuracy because it depends on the velocity gradient ($\partial u/\partial y$), this amplifies discretization errors. The shear stress was computed using a central difference approximation:

$$\tau = \mu \left(\frac{u_{i+1} - u_{i-1}}{2\Delta y} \right) \tag{29}$$

As shown in Figure 15, the numerical shear stress profile closely tracks the analytical curve derived from Eq. (27). The maximum discrepancy occurs at the plate boundary ($y = h\&0$), where the gradients are steepest. The average wall shear stress error was found to be approximately **1.33%**, which is within acceptable limits for a Finite Difference Scheme simulation.

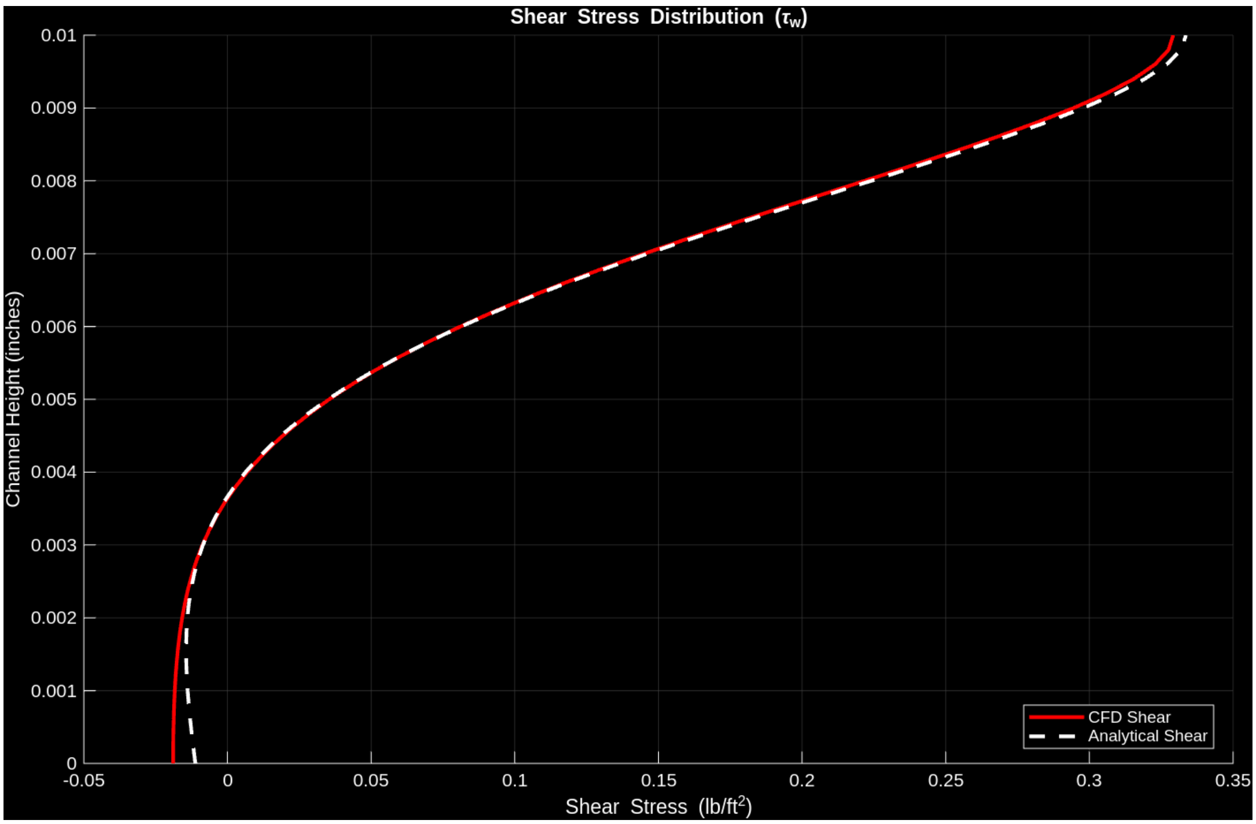

Figure 15: Analytical VS Simulation Shear Stress Profile at $t = 1.25T$

3.4 Visualization of Momentum Diffusion :

To visualize the evolution of the flow field, we plotted spacetime contour plots (Figure 16171819). Theses diagrams map the velocity magnitude $u(y,t)$ as a function of time (x-axis) and Gap height (y-axis), describing how momentum diffuses from the oscillating plate into the fluid field.

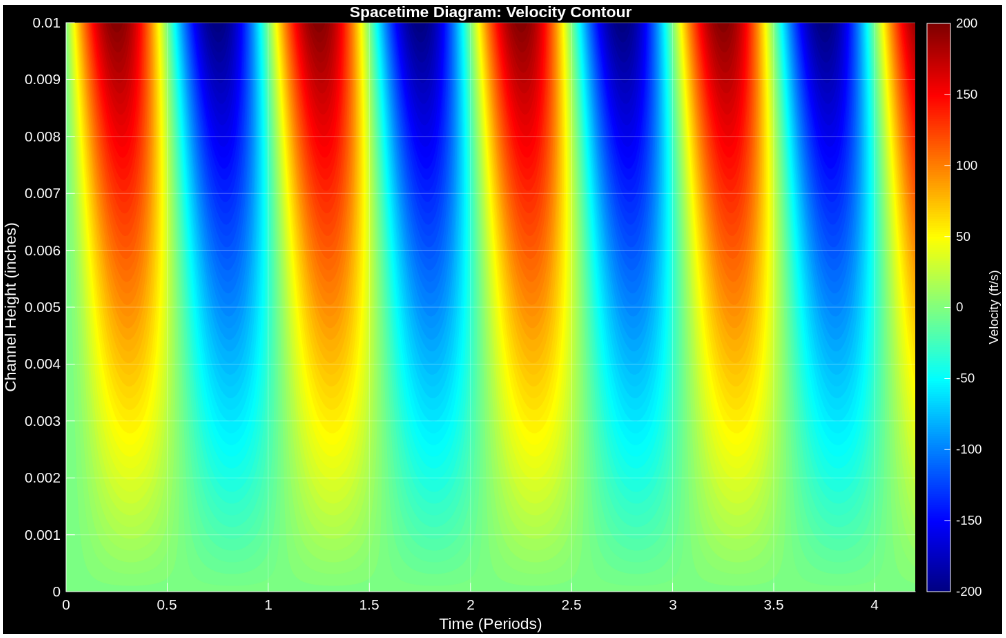

Figure 16: Evolution of the flow field in Space-Time at Frequency = 100Hz

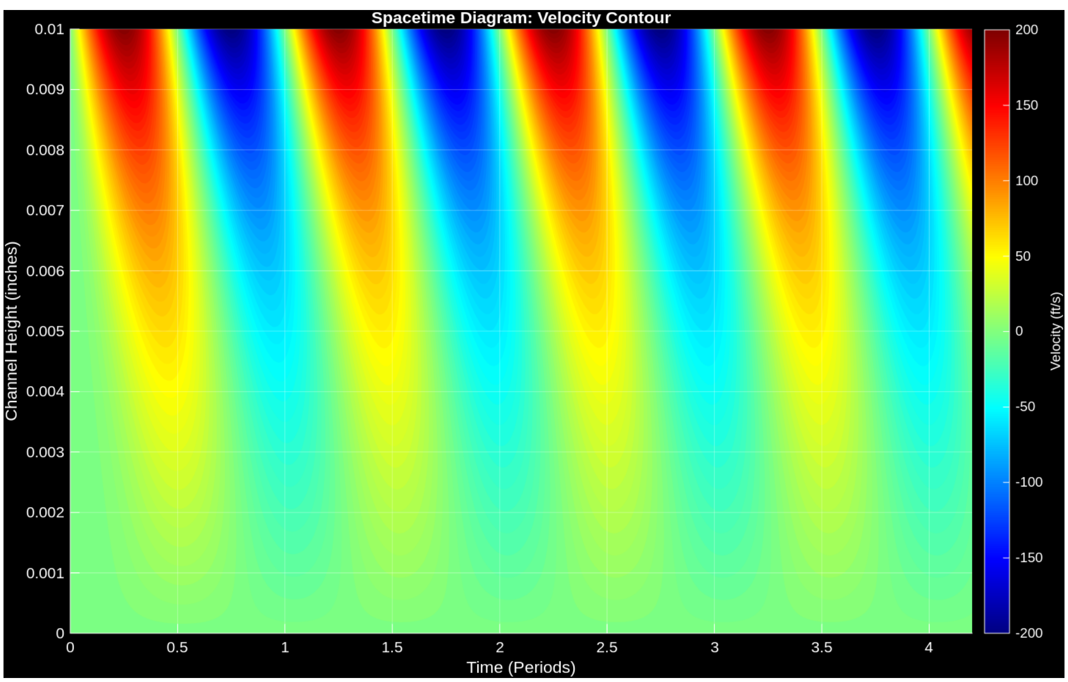

Figure 17: Evolution of the flow field in Space-Time at Frequency = 500Hz

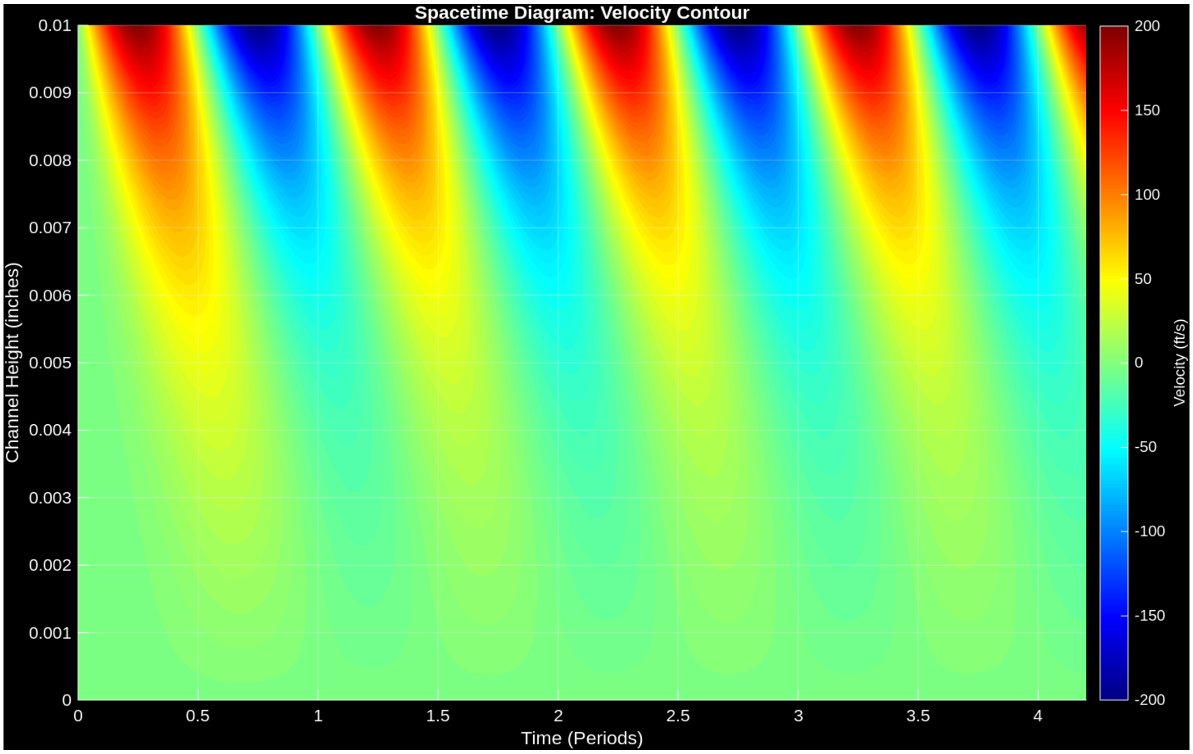

Figure 18: Evolution of the flow field in Space-Time at Frequency = 1000Hz

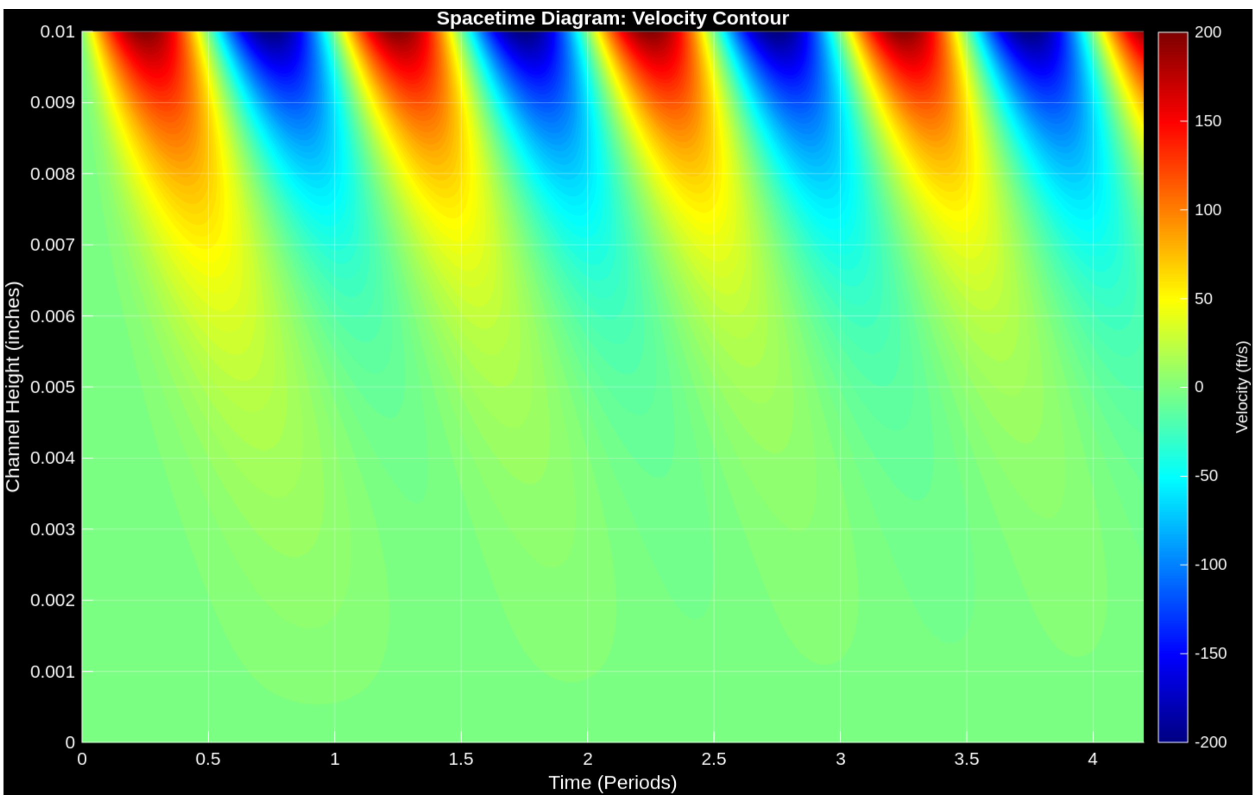

Figure 19: Evolution of the flow field in Space-Time at Frequency = 2000Hz

3.4.1 Wall Shear Stress Hysteresis :

Figure 20 shows the phase plot of the Wall Shear Stress (τ_w) plotted against the Wall Velocity (u_n) for 1 complete oscillation cycle at $f = 1000$ Hz. This plot is also called the Lissajous Plot.

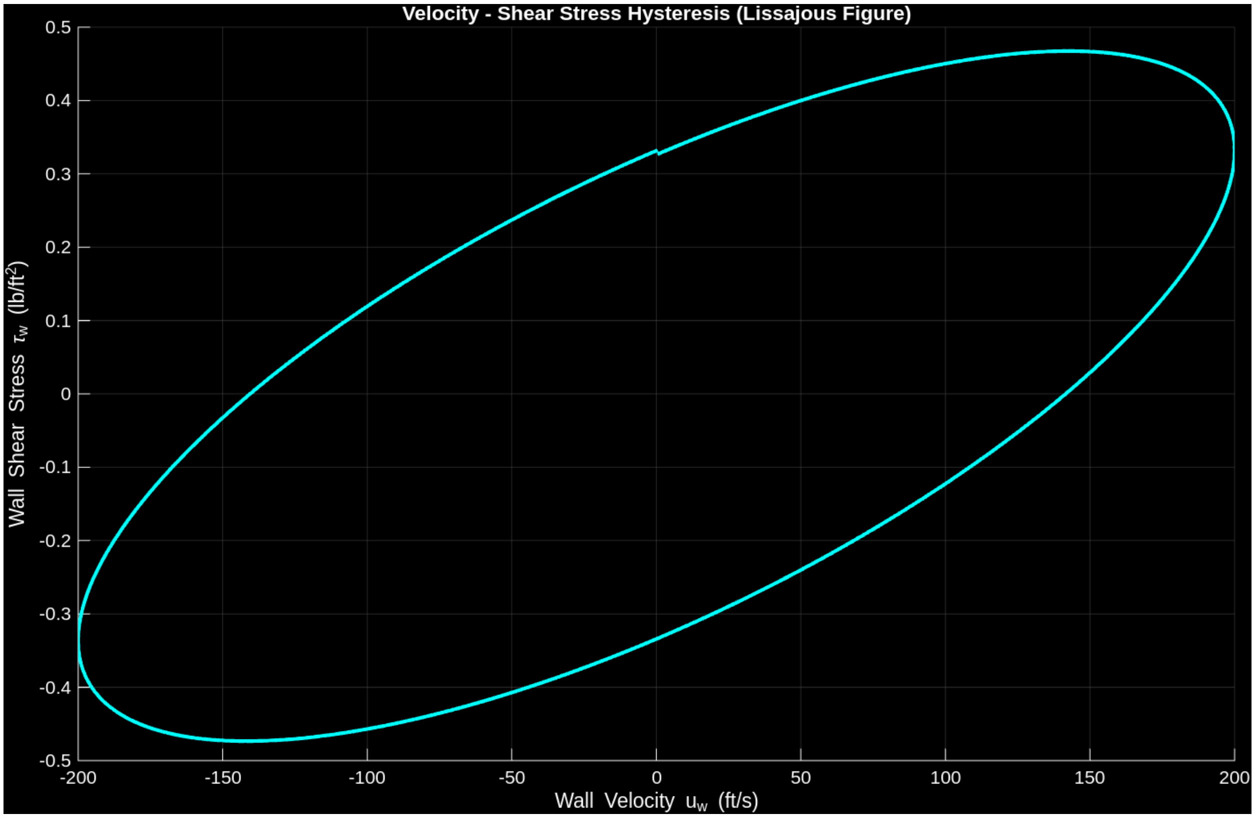

Figure 20: Wall Shear Stress Hysteresis plot.

4 Discussion of Physics

4.1 Compressibility Analysis

Compressibility refers to the extent to which a fluid's density changes with pressure and temperature. The Mach number (Ma), defined as the ratio of the flow velocity to the speed of sound, is the key parameter determining this behavior.

- **AT ($U_\infty = 200$ ft/s):** The calculated Mach number is $Ma \approx 0.18$. Since $Ma < 0.3$, density variations are negligible ($\Delta\rho \approx 0$). The flow is treated as **Incompressible**, which allows the decoupling of the momentum and energy equations (temperature changes do not affect the velocity field).

4.2 Velocity Profiles :

The velocity profile describes how the flow velocity varies across the gap h. As shown in the provided plots, the flow behavior changes drastically as the frequency increases from 100 Hz to 2000 Hz:

- **Low Frequency (100 & 500 Hz):** At $f = 100$ Hz, the penetration depth δ is large relative to the gap height (h). The viscous forces have sufficient time within each cycle to diffuse momentum deep into the flow field. The entire flow field is influenced by the viscous dissipation and the velocity profiles show the same 12 . The influence of the oscillating plate reaches the lower stationary plate, creating a linear profile over time.

- **High Frequency (1000 & 2000 Hz):** As the Frequency increases, the penetration depth decreases significantly ($\delta \propto 1/\sqrt{f}$). The influence of momentum transfer within the flow field keeps decreasing 34. At $f = 2000$ Hz, there is essentially very minimal change in the velocity profile of the flow field and it is reduced to the layers adjacent to the oscillating plate. This is due to the rapid reversal of the plate motion. This prevents the shear wave from propagating into the field.

- **Phase lag :** A distinct feature observed in all plots is the ”Phase Lag” of the velocity profiles. At the instant where the plate velocity is zero (Phase = 0.5T or 1.0T), the fluid velocity inside the domain is non-zero. The fluid layers away from the plate reacts to the plate’s motion with a delay ϕ, which increases linearly with distance: $\phi \approx -y\sqrt{\omega/2\nu}$. Consequently, when the plate changes direction, the inertia inside the fluid maintains its original direction, creating the bulge in the velocity profile.

4.3 Temperature Profiles :

Unlike the velocity field, which is driven by motion, the temperature field is driven by the conversion of kinetic energy into internal energy through viscous dissipation and thermal conduction.

- **Low Frequency :** At lower frequencies Figure 56, the generated heat has sufficient time to diffuse away from the plate and penetrate deep into the fluid field. The temperature profiles show a broader distribution. The peak temperature is lower because the energy is spread over a larger quantity of fluid.

- **High Frequency (Localized Heating):** At higher frequencies Figure 78, the velocity gradients near the plate become extremely steep due to the thin Stokes layer. This results in intense localized heat generation. Because the plate oscillates much faster than the time it takes for thermal diffusion to spread, the heat is trapped in the region near the upper plate. This is can be seen through the sharp temperature rise near $y = 0.1$ and the rapid decay at $y = 0$.

4.4 Analysis of Wall Shear Stress Hysteresis :

The elliptical shape observed in Figure 20 indicates a significant phase shift between the wall velocity and the shear stress. This behavior aligns with the analytical solution for Stokes' second problem, where the shear stress leads the wall velocity by a phase angle of $\pi/4$ (45°). The plot confirms this behavior:

- **At Zero Velocity:** At the instant the plate velocity reaches zero ($u_n \approx 0$) momentarily, the wall shear stress remains significant ($\tau_w \approx 0.33$ lb/ft^2). This indicates that the fluid adjacent to the plate retains momentum even if the plate stops to reverse direction.

- **Peak Wall Stress :** The maximum shear stress occurs before the plate reaches its maximum velocity ($U_{inf} = \pm 200$ ft/s). By the time the plate reaches maximum speed, the wall shear stress has already begun to decay.

- **Viscous Dissipation** The area enclosed by the hysteresis loop represents the irreversible work done by the plate on the fluid field over one cycle. This work is dissipated as heat, The area under the ellipse in Figure 20 confirms high viscous dissipation, which is the energy source driving the temperature rise seen in the temperature profile analysis.

$$W_{cycle} = \oint \tau_w \, du_n \tag{30}$$

4.5 Momentum Diffusion :

The 16171819 contour plots help visualize two distinct physical characteristic of our problem that is :

- **Penetration Depth:** As the momentum diffuses downward from the upper plate into the flow field (decreasing y), the intensity of the velocity contours fades rapidly into the zero velocity region. This visual gradient represents the viscous dissipation of kinetic energy. The transition to this region marks the effective limit of the Stokes Boundary Layer ($\delta \approx \sqrt{2\nu/\omega}$). Beyond this depth, the fluid inertia dominates over viscous diffusion, and the fluid remains effectively undisturbed.

- **Phase Lag:** The contour bands are not vertical, as we increase the frequency they exhibit a distinct tilt. This slope describes the speed at which the shear wave travels. The peak velocity at a depth y occurs later than the peak velocity at the plate. This confirms the phase lag term $(\omega t - ky)$ present in the analytical solution:

$$u(y,t) \propto e^{-ky} \sin(\omega t - ky) \tag{31}$$

5 Conclusion

This project implemented an Explicit Finite Difference Method (FDM) scheme solver to simulate the fluid flow between parallel plates driven by an oscillating boundary (Stokes'

Second Problem). The simulation results highlighted that the velocity profiles confirmed the inverse relationship between frequency and the Stokes penetration depth ($\delta \propto 1/\sqrt{f}$). This validates the theoretical prediction that at high frequency, changes in the flow field is limited to a thin layer close to the oscillating plate boundary. Secondly, the analysis of the wall shear stress versus velocity showed a distinct elliptical hysteresis loop. This shape confirms that the shear stress leads the plate velocity by a phase angle due to inertia inside the fluid. Thirdly, the energy equation captured the conversion of kinetic energy into internal energy, showing that the intense shear rates within the thin Stokes layer result in significant, localized heating near the plate.

From a simulation perspective, the problem was solved using a Forward Time Central Space discretization scheme. To ensure accuracy, the time step was strictly controlled by a CFL criteria for both momentum and thermal diffusion. The results were validated against the analytical solution for Stokes' second problem, analysis showed that the Root Mean Square Error (RMSE) between the analytical solution and the simulation is well within acceptable engineering limits. In conclusion, this simulation demonstrated the Explicit FDM provides a highly accurate solution for Stokes' Second Problem.

References

[1] White, F. M. (2006). *Viscous Fluid Flow* (3rd ed.). New York: McGraw-Hill. (Section 3-5 for the analytical derivation of Stokes' Second Problem).

[2] Anderson, J. D. (1995). *Computational Fluid Dynamics: The Basics with Applications*. New York: McGraw-Hill. (Chapter 4 for Explicit Finite Difference schemes and stability analysis).

[3] Schlichting, H., & Gersten, K. (2016). *Boundary-Layer Theory* (9th ed.). Springer. (Reference for the Stokes boundary layer thickness $\delta \approx \sqrt{2\nu/\omega}$).

Appendix: Code Listing

```
clear all
close all
clc

%%
ADIABATIC_LOWER_PLATE = false;

%% PARAMETERS

freq = 5000;              % Frequency in Hz
omega = 2 * pi * freq;    % Angular velocity
Period = 1 / freq;        % Time for one full cycle

U_inf = 200;              % Max velocity
h_inch = 0.01;            % Plate gap in inches
h = h_inch / 12;
```

```
T0 = 519;                % Wall Temperature (Rankine)

%
rho = 0.00237;
mu = 3.737e-7;

%
J = 778.17;
g_c = 32.174;
cp_btu = 0.24;
cp = cp_btu * g_c * J;
Pr = 0.71;
k = (mu * cp) / Pr;
gamma = 1.4;
R = 1716;

%%
c = sqrt(gamma * R * T0);
Ma = U_inf / c;
fprintf('Mach Number (Max) = %.3f\n', Ma);

%% GRID AND TIME STEP
n = 51;
dy = h / (n - 1);
y = linspace(0, h, n);

% Stability Time Step
alpha = k / (rho * cp);
dt_limit_v = 0.5 * (rho * dy^2) / mu;
dt_limit_t = 0.5 * dy^2 / alpha;
dt = min(dt_limit_v, dt_limit_t) * 0.9;
fprintf('Grid Nodes: %d, Time Step: %.2e s\n', n, dt);

%% INITIAL CONDITIONS
u = zeros(1, n);
T = ones(1, n) * T0;
u(1) = 0;
u(n) = 0;
T(n) = T0;
if ~ADIABATIC_LOWER_PLATE
    T(1) = T0;
end

%% MAIN LOOP
t_now = 0;
t_max = 2.0 * Period;   % Control cycles
iter = 0;

time_targets = [0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1.0] * Period
    ;
N_targets = length(time_targets);
u_save = zeros(N_targets, n);
T_save = zeros(N_targets, n);
saved_flags = zeros(1, N_targets);
```

```
u_new = u;
T_new = T;

fprintf('Starting Oscillating Simulation (Freq: %d Hz)...\n', freq);

while (t_now < t_max)

    % Momentum
    for i = 2:(n-1)
        diffusion = (mu/rho) * (u(i+1) - 2*u(i) + u(i-1)) / dy^2;
        u_new(i) = u(i) + dt * diffusion;
    end

    % Energy
    for i = 2:(n-1)
        diff_T = (k / (rho*cp)) * (T(i+1) - 2*T(i) + T(i-1)) / dy^2;
        du_dy = (u(i+1) - u(i-1)) / (2*dy);
        visc_heat = (mu / (rho*cp)) * (du_dy)^2;
        T_new(i) = T(i) + dt * (diff_T + visc_heat);
    end

    % BC
    u_new(1) = 0;
    u_new(n) = U_inf * sin(omega * (t_now + dt));

    T_new(n) = T0;
    if ADIABATIC_LOWER_PLATE
        T_new(1) = T_new(2);
    else
        T_new(1) = T0;
    end

    % Save Data
    for m = 1:N_targets
        if (t_now >= time_targets(m)) && (saved_flags(m) == 0)
            u_save(m,:) = u_new;
            T_save(m,:) = T_new;
            saved_flags(m) = 1;
        end
    end
    %update data
    u = u_new;
    T = T_new;
    t_now = t_now + dt;
    iter = iter + 1;
end
fprintf('Simulation Complete.\n');

%%
% Shear calculation at the FINAL time step
tau = zeros(1, n);
tau(1) = mu * (u(2) - u(1)) / dy;
for i = 2:n-1
```

```
    tau(i) = mu * (u(i+1) - u(i-1)) / (2*dy);
end
tau(n) = mu * (u(n) - u(n-1)) / dy;

%% PLOTs

% VELOCITY
figure('Name', 'Oscillating Velocity Profile', 'Color', 'black');
hold on;
colors = parula(N_targets);
for m = 1:N_targets
    phase = time_targets(m) / Period;
    plot(u_save(m,:), y*12, 'LineWidth', 1.5, ...
        'Color', colors(m,:), ...
        'DisplayName', sprintf('Phase=%.3f T', phase));
end
xlabel('Velocity u (ft/s)');
ylabel('Gap Height h (inch)');
title(['Velocity Profiles over 1 Cycle (' num2str(freq) ' Hz)']);
legend('Location','bestoutside');
xlim([-U_inf-10, U_inf+10]);
grid on;

% TEMPERATURE
figure('Name', 'Temperature Profile', 'Color', 'black');
hold on;

for m = 1:N_targets
    phase = time_targets(m) / Period;


    if m == N_targets
        current_style = '-';
        current_width = 2;
        current_alpha = 1.0;
    else

        current_style = '--';
        current_width = 1.5;
        current_alpha = 0.7;
    end


    p = plot(T_save(m,:), y*12, ...
             'LineStyle', current_style, ...
             'LineWidth', current_width, ...
             'DisplayName', sprintf('Phase=%.3f T', phase));

    base_color = colors(m, 1:3);
    p.Color = [base_color, current_alpha];
end

xlabel('Temperature T (Rankine)');
ylabel('Gap Height y (inch)');
```

```
title('Temperature Distribution (Viscous Heating)');
legend('Location','bestoutside');
grid on;

% SHEAR STRESS
figure('Name', 'Shear Stress Final Snapshot', 'Color', 'white');
plot(tau, y*12, 'r-', 'LineWidth', 2);
xlabel('Shear Stress \tau (lb/ft^2)');
ylabel('Gap Height y (inch)');
title('Shear Stress Distribution (Final Time Step)');
grid on;

%%
    -----------------------------------------------------------------------

clear all
close all
clc

%% PARAMETERS
freq = 1000;
omega = 2 * pi * freq;
Period = 1 / freq;
U_inf = 200;              % Max Amplitude
h_inch = 0.01;
h = h_inch / 12;          % Gap in FEET
T0 = 519;                 % Wall Temp

rho = 0.00237;
mu = 3.737e-7;
J = 778.17;
g_c = 32.174;
cp_btu = 0.24;
cp = cp_btu * g_c * J;
Pr = 0.71;
k = (mu * cp) / Pr;
gamma = 1.4;
R = 1716;
ADIABATIC_LOWER_PLATE = false;

%% GRID INDEPENDENCE STUDY
Grid_Sizes = [11, 21, 51, 71, 101, 201];
colors = {'r', 'b', 'y','g','m','c'};
line_styles = {'-', '-', '-','-','-','-'};


SF = 0.5;


figure(1); clf; hold on;
title('Velocity Profile at Peak Velocity (t = 1.25 Periods)');
xlabel('Velocity (ft/s)'); ylabel('Gap Height (in)');

figure(2); clf; hold on;
```

```
title('Temperature Profile at Peak Velocity (t = 1.25 Periods)');
xlabel('Temperature (R)'); ylabel('Gap Height (in)');


%% MAIN LOOP
for g_idx = 1:length(Grid_Sizes)

    n = Grid_Sizes(g_idx);


    dy = h / (n - 1);
    y = linspace(0, h_inch, n);


    alpha = k / (rho * cp);
    dt_limit_v = 0.5 * (rho * dy^2) / mu;
    dt_limit_t = 0.5 * dy^2 / alpha;
    dt = min(dt_limit_v, dt_limit_t) * SF;



    u = zeros(1, n);
    T = ones(1, n) * T0;
    u(1) = 0; u(n) = 0; T(n) = T0;
    if ~ADIABATIC_LOWER_PLATE, T(1) = T0; end

    % Loop
    t_now = 0;
    t_max = 2.0 * Period;


    captured_peak = false;
    u_peak = u;
    T_peak = T;

    % Simulation Loop
    while (t_now < t_max)
        u_old = u;
        T_old = T;

        % Momentum
        for i = 2:(n-1)
            diffusion = (mu/rho) * (u_old(i+1) - 2*u_old(i) + u_old(i-1))
 / dy^2;
            u(i) = u_old(i) + dt * diffusion;
        end

        % Energy
        for i = 2:(n-1)
            diff_T = (k / (rho*cp)) * (T_old(i+1) - 2*T_old(i) + T_old(i
-1)) / dy^2;
            du_dy = (u_old(i+1) - u_old(i-1)) / (2*dy);
            visc_heat = (mu / (rho*cp)) * (du_dy)^2;
            T(i) = T_old(i) + dt * (diff_T + visc_heat);
```

```
        end

        % Boundary Conditions
        u(1) = 0;
        u(n) = U_inf * sin(omega * (t_now + dt));
        T(n) = T0;
        if ADIABATIC_LOWER_PLATE
            T(1) = T(2);
        else
            T(1) = T0;
        end

        % at 1.25 Periods
        target_time = 1.25 * Period;
        if (t_now >= target_time) && (captured_peak == false)
            u_peak = u;
            T_peak = T;
            captured_peak = true;
        end

        t_now = t_now + dt;
    end
    fprintf('Complete.\n');

    figure(1);
    plot(u_peak, y, 'Color', colors{g_idx}, 'LineStyle', line_styles{g_idx
    }, 'LineWidth', 1.5, ...
        'DisplayName', sprintf('Grid N = %d', n));

    figure(2);
    plot(T_peak, y, 'Color', colors{g_idx}, 'LineStyle', line_styles{g_idx
    }, 'LineWidth', 1.5, ...
        'DisplayName', sprintf('Grid N = %d', n));
end

%% POST-PROCESSING

figure(1); legend('show'); grid on;
figure(2); legend('show', 'Location', 'Best'); grid on;
%% VALIDATION: ANALYTICAL SOLUTION vs CFD

y_feet = linspace(0, h, n);


y_dist = h - y_feet;


nu = mu / rho;
k_stokes = sqrt(omega / (2 * nu));


t_capture = 1.25 * Period;
u_analytical = U_inf .* exp(-k_stokes .* y_dist) .* ...
               sin(omega * t_capture - k_stokes .* y_dist);
```

```
% 5. Plot Comparison
figure('Name', 'Validation', 'Color', 'black');
hold on;

% Plot CFD Result
plot(u_peak, y_feet * 12, 'r-', 'LineWidth', 2, 'DisplayName', 'CFD
    Simulation');

% Plot Analytical Result
plot(u_analytical, y_feet * 12, 'y--', 'LineWidth', 1.5, 'MarkerSize', 6,
    ...
     'DisplayName', 'Analytical Solution');

title(' CFD vs. Analytical Solution');
xlabel('Velocity (ft/s)');
ylabel('Gap Height (in)');
legend('Location', 'Best');
grid on;

% 6. Calculate Error
error_val = u_peak - u_analytical;
rmse = sqrt(mean(error_val.^2));
fprintf('Root Mean Square Error (RMSE): %.4f ft/s\n', rmse);

%% ERROR DISTRIBUTION ANALYSIS

error_dist = u_peak - u_analytical;
figure('Name', 'Error Distribution', 'Color', 'black');
plot(error_dist, y_feet * 12, 'y-', 'LineWidth', 1.5, 'MarkerSize', 5, ...
     'MarkerFaceColor', 'r');
xline(0, 'c-', 'Zero Error Line', 'LabelVerticalAlignment', 'bottom');

title('Velocity Error Distribution across the Gap');
xlabel('Velocity Error [ft/s]');
ylabel('Gap Height (in)');
grid on;
max_err = max(abs(error_dist));
fprintf('max error: %.4f ft/s\n', max_err);
%% SHEAR STRESS ANALYSIS t = 1.25 Periods

du_dy_num = gradient(u_peak, y_feet);
tau_cfd = mu .* du_dy_num;



theta = omega * 1.25 * Period - k_stokes .* y_dist;
du_dy_exact = k_stokes * U_inf .* exp(-k_stokes .* y_dist) .* (sin(theta)
    + cos(theta));

tau_analytical = mu .* du_dy_exact;


figure('Name', 'Shear Stress Distribution', 'Color', 'black');
```

```
ax = axes;
set(ax, 'Color', 'black', 'XColor', 'white', 'YColor', 'white');
hold on;

% Plot CFD Shear
plot(tau_cfd, y_feet * 12, 'r-', 'LineWidth', 2, 'DisplayName', 'CFD Shear');

% Plot Analytical Shear
plot(tau_analytical, y_feet * 12, 'w--', 'LineWidth', 2, 'DisplayName', 'Analytical Shear');

title('Shear Stress Distribution (\tau_w)', 'Color', 'white');
xlabel('Shear Stress (lb/ft^2)', 'Color', 'white');
ylabel('Channel Height (inches)', 'Color', 'white');
legend('Location', 'Best', 'TextColor', 'white', 'EdgeColor', 'white', 'Color', 'black');
grid on;
ax.GridColor = [0.3, 0.3, 0.3];
ax.GridAlpha = 0.5;


tau_wall_cfd = tau_cfd(end);
tau_wall_exact = tau_analytical(end);

error_tau = abs(tau_wall_cfd - tau_wall_exact);
pct_error_tau = (error_tau / abs(tau_wall_exact)) * 100;

fprintf('SHEAR STRESS ANALYSIS:\n');
fprintf('Wall Shear (CFD):       %.4f lb/ft^2\n', tau_wall_cfd);
fprintf('Wall Shear (Analytical): %.4f lb/ft^2\n', tau_wall_exact);
fprintf('Error at Wall:           %.2f%%\n', pct_error_tau);
%% ---------------------------------------------------------------------------

%% HYSTERESIS LOOP PLOT
figure('Name', 'Hysteresis Loop', 'Color', 'black');
ax = axes;
set(ax, 'Color', 'black', 'XColor', 'white', 'YColor', 'white');
hold on;

% history
plot(hist_u_wall, hist_tau_wall, 'c-', 'LineWidth', 2, 'DisplayName', 'Cycle Data');

% Formatting
title('Velocity - Shear Stress Hysteresis (Lissajous Figure)', 'Color', 'white');
xlabel('Wall Velocity u_w (ft/s)', 'Color', 'white');
ylabel('Wall Shear Stress \tau_w (lb/ft^2)', 'Color', 'white');
grid on;
ax.GridColor = [0.3, 0.3, 0.3];
ax.GridAlpha = 0.5;
%%
```

```
    --------------------------------------------------------------------------------------------
clear all
close all
clc

%% PARAMETERS
freq = 4000;
omega = 2 * pi * freq;
Period = 1 / freq;
U_inf = 200;
h_inch = 0.01;
h = h_inch / 12;

rho = 0.00237;
mu = 3.737e-7;
nu = mu / rho;

%%
n = 51;
dy = h / (n - 1);
y = linspace(0, h_inch, n);

dt = 0.5 * (rho * dy^2) / mu * 0.9;

%% SETUP
n_cycles = 4.2;
t_max = n_cycles * Period;
num_steps = ceil(t_max / dt);
u_spacetime = zeros(n, num_steps);
t_vector = zeros(1, num_steps);

% Initial Conditions
u = zeros(1, n);
u(n) = 0;
t_now = 0;

fprintf('Simulating Spacetime Evolution (%d cycles)...\n', n_cycles);

%% MAIN LOOP
for k = 1:num_steps
    u_old = u;

    % Diffusion Equation
    for i = 2:(n-1)
        diffusion = (nu) * (u_old(i+1) - 2*u_old(i) + u_old(i-1)) / dy^2;
        u(i) = u_old(i) + dt * diffusion;
    end

    % Boundary Conditions
    u(1) = 0;
    u(n) = U_inf * sin(omega * (t_now + dt));

    u_spacetime(:, k) = u';
```

```
    t_vector(k) = t_now;

    t_now = t_now + dt;
end


%% PLOTTING: SPACETIME CONTOUR
figure('Name', 'Spacetime Diagram', 'Color', 'black');
ax = axes;
set(ax, 'Color', 'black', 'XColor', 'white', 'YColor', 'white');
hold on;
cmap = jet(256);
mid_idx = 128;
cmap(mid_idx-5:mid_idx+5, :) = 0;
[C, h_contour] = contourf(t_vector/Period, y, u_spacetime, 100, 'LineColor
    ', 'none');
colormap(jet);
clim([-U_inf, U_inf]);

c = colorbar;
c.Color = 'white';
c.Label.String = 'Velocity (ft/s)';
c.Label.Color = 'white';


title('Spacetime Diagram: Velocity Contour', 'Color', 'white');
xlabel('Time (Periods)', 'Color', 'white');
ylabel('Channel Height (inches)', 'Color', 'white');
ylim([0, h_inch]);
xlim([0, n_cycles]);
ax.Layer = 'top';
grid on;
ax.GridColor = [1, 1, 1];
ax.GridAlpha = 0.3;
```

Listing 1: MATLAB Code for Explicit Scheme