Computational Fluid Dynamics (CFD) Stimulation in MATLAB

Janmjay Kumar [1], Vikash Kumar Yadav [2]

[1] Master of Technology in Mechanical Engineering
[2] Assistant Professor, Department of Mechanical Engineering
[1,2] BM Group of Institutions, Farrukh Nagar, Gurugram.
[*]*Email: yeswant1206@gmail.com*

ABSTRACT

This study conducted a Computational Fluid Dynamics (CFD) analysis of supersonic flow over a flat plate to explore the flow dynamics and associated aerodynamic phenomena in high-speed environments. MacCormack's technique was employed for simulation, which revealed critical features such as boundary layer development, shock wave formation, and variations in pressure and temperature distributions. The findings indicated the complex interactions between these elements and their impact on the aerodynamic performance of a flat plate under supersonic conditions. Specifically, the study provided insights into heat transfer and aerodynamic friction through an examination of boundary layer behaviour, as well as the effects of shock waves on pressure and temperature gradients. Convergence and stability checks demonstrated the robustness of the numerical method, while comparisons with theoretical and experimental data affirmed the accuracy of the results. These insights are valuable for engineering applications, especially in aerospace, where supersonic flows are prevalent. The outcomes from this study can inform the design of supersonic aircraft and high-speed vehicles, contributing to improved aerodynamic efficiency and reduced drag. This research serves as a reference point for future investigations in supersonic flow dynamics and related engineering domains. The future scope for this research encompasses various directions, including exploring advanced numerical methods to enhance accuracy and stability, investigating complex turbulence modelling like Large Eddy Simulation (LES) or Direct Numerical Simulation (DNS), and applying CFD to more complex geometries such as airfoils or wings. Additionally, future research could delve into variable freestream conditions and design optimization to achieve more efficient supersonic applications. Experimental validation through wind tunnel tests and other real-world data collection is also crucial for ensuring the reliability of CFD results. These efforts will expand our understanding of supersonic aerodynamics and contribute to the development of more effective and efficient high-speed vehicles.

Keywords: Computational Fluid Dynamics, LES, DNS.

I. INTRODUCTION

Computational Fluid Dynamics (CFD) simulation is a highly effective tool utilised in engineering and science to analyse and predict the behaviour of fluid flows. Using MATLAB, a high-level programming language and environment, you can easily implement CFD simulations with its wide range of functionalities and tools. CFD simulation in MATLAB offers engineers and researchers a wide range of opportunities to model and analyse fluid flow phenomena in a cost-effective and efficient manner. With its user-friendly interface, extensive mathematical libraries, and powerful numerical solvers, MATLAB is the perfect platform for developing CFD simulations. CFD simulation in MATLAB provides engineers and researchers with a robust and versatile platform for modelling and analysing fluid flow phenomena. Utilising its advanced mathematical capabilities, intuitive interface, and seamless integration with external solvers, MATLAB enables users to efficiently and effectively address a diverse array of CFD problems. Nevertheless, it is crucial for users to have a good understanding of the capabilities and constraints of MATLAB and to make use of the resources at their disposal to achieve the best possible simulation outcomes.

1.1 Computational Fluid Dynamics (CFD)

Computational Fluid Dynamics (CFD) involves the numerical simulation of fluid flows and their interactions with solids, gases, and other fluids. This enables engineers and scientists to analyse and predict the behaviour of fluids in different scenarios.

Central to CFD is the process of discretizing the governing equations, which involves dividing the continuous fluid domain into a finite number of discrete elements or cells. To conduct a CFD simulation, a series of steps must be followed:

- Pre-Processing: This step includes defining the geometry of the problem domain, specifying boundary conditions, and meshing, which is the process of dividing the domain into discrete cells.
- Numerical Solution: Utilising the discretized equations and boundary conditions, numerical algorithms are utilised to determine the fluid flow variables within each cell. This usually requires using iterative methods to reach a solution that meets the equations.
- Post-Processing: After the simulation is done, the results are examined and displayed. This may require generating contour plots, streamlines, velocity vectors, and other visualisations to gain a deeper understanding of the flow behaviour.

CFD Finds Applications Across Various Industries and Research Fields

CFD simulations offer valuable insights and predictions that can be challenging or even impossible to obtain solely through experimental methods. Engineers can optimise designs, improve performance, and reduce costs by identifying potential issues early in the design process.

Nevertheless, CFD simulations do come with their fair share of limitations and challenges. They demand substantial computational resources, and the precision of the outcomes relies on several factors, including mesh quality, turbulence modelling, and numerical errors. In addition, CFD simulations are frequently compared to experimental data to ensure their accuracy. Its precise simulation of intricate fluid phenomena makes it an essential tool in contemporary engineering and scientific research.

1.1.1 CFD And Its Applications

It has become a valuable tool in engineering, physics, and other scientific fields because of its capacity to accurately simulate complex fluid behaviours and predict fluid flow phenomena. This overview offers a comprehensive understanding of the principles, methodologies, and applications of CFD.

Principles of CFD

CFD relies on conservation laws like mass, momentum, and energy conservation, articulated through PDEs such as Navier-Stokes equations, governing fluid dynamics. By discretizing these equations into algebraic forms, CFD enables numerical solutions via computational methods, facilitating the understanding and prediction of complex fluid flow behaviors.

Methodologies in CFD

During pre-processing, the domain's geometry is established and a computational grid or mesh is created. During this stage, the behaviour of the fluid at the domain boundaries is specified through boundary conditions. Finally, post-processing involves analysing and visualising the simulation results to extract valuable insights into the fluid flow behaviour.

Applications of CFD

There are several notable applications of CFD

- Aerospace and Aeronautics: It allows engineers to analyse aerodynamic forces, airflow around aircraft components, and the performance of jet engines, resulting in enhanced efficiency and safety.

In the automotive industry, CFD is crucial for designing vehicles with improved aerodynamic performance, increased fuel efficiency, and efficient cooling systems. It aids engineers in the analysis of airflow around vehicles, the prediction of heat transfer in engine components, and the optimisation of vehicle performance.

- Energy and Environmental Engineering: CFD is used in the design and analysis of renewable energy systems, such as wind turbines and solar panels, to enhance their efficiency and sustainability. It is also helpful in modelling air and water pollution dispersion, evaluating indoor air quality, and designing pollution control devices.

These simulations offer valuable insights into physiological processes, contribute to the diagnosis and treatment of diseases, and support the development of medical devices.

Civil and Environmental Engineering: CFD assists engineers in analysing airflow and pollutant dispersion in urban environments, optimising HVAC systems in buildings, and designing sustainable urban infrastructure. It is also used in hydraulic engineering to simulate river and coastal flows, optimise water distribution systems, and evaluate flood risks. Its remarkable capability to anticipate intricate flow behaviours, enhance designs, and boost efficiency has rendered it an essential tool in the fields of engineering, science, and research. With the rapid advancement of computational capabilities, CFD is set to become even more crucial in tackling real-world challenges and fostering innovation across different industries.

Computational Fluid Dynamics (CFD):

1) Numerical approaches: Computational fluid dynamics (CFD) involves solving the governing equations of fluid flow computationally by using numerical methods including finite difference, finite volume, and finite element approaches to discretize the equations.

In order to do CFD simulations, the domain must first be discretized into a grid or mesh. Grid generation is the second step. The process of grid generation includes the creation of a mesh that precisely depicts the geometry of the issue domain and successfully captures the flow characteristics in the suitable manner.

2) Governing Equations: Computational fluid dynamics (CFD) models are based on the Navier-Stokes equations, which describe the conservation of mass, momentum, and energy in hydraulic flows. An iterative solution across the computational domain is used for the discretization of these equations.

   Boundary conditions are an essential part of any accurate simulation of real-world fluid flow processes. The behaviour of the fluid at the boundaries of the computational area is specified by boundary conditions.

3) Solution strategies: In order to solve the discretized equations and get numerical solutions that describe the flow field, a number of different solution strategies are used. These techniques include iterative solvers and time-stepping methods, among others.

4) Validation and Verification: In order to guarantee the precision and dependability of the simulations, it is necessary to validate and verify the findings of the computational fluid dynamics (CFD) simulations by comparing them with experimental data or analytical solutions.

5) Applications: Computational fluid dynamics (CFD) has applications in a wide variety of industries, including aerospace engineering, automobile design, environmental modelling, biomedical engineering, and industrial processes. It is used to assist in the optimisation and design of fluid systems.

Computational fluid dynamics expands this knowledge by allowing numerical modelling and analysis of complicated fluid flow issues using computational methods. Fluid dynamics offers the theoretical basis for understanding fluid behaviour, whereas computational fluid dynamics enhances this knowledge.

1.1.2 Continuum Hypothesis

In the fields of fluid mechanics and thermodynamics, the continuum hypothesis is established as a fundamental notion. According to this theory, matter may be considered continuous rather than discrete when considering it on a scale that is sufficiently big. To put it another way, it is predicated on the idea that matter may be broken down into an endless number of subatomic particles, and that attributes like density, pressure, and velocity are assumed to be continually variable over space and time.

This hypothesis is especially important in the field of fluid dynamics since it makes it possible to explain the behaviour of fluids via the use of differential equations because it allows for this. The assumption that fluid characteristics are continuous is the foundation upon which certain equations, such as the Navier-Stokes equations, are built. The mathematical modelling of fluid flow is simplified as a result of this, which makes it feasible to analyse complicated fluid systems and forecast which behaviour they will exhibit.

When it comes to actual engineering applications, the continuum hypothesis is valid for the majority of situations in which the length scales involved are far bigger than the size of molecules. Nevertheless, the discrete character of matter becomes crucial at extremely tiny sizes, such as in microfluidics or nanofluidics and it is possible that continuum assumptions are no longer true at these scales. In situations like these, additional modelling methodologies, such as simulations of molecular dynamics, are used in order to take into account the discrete character of fluids at the molecular level.

1.1.3 Conservation Equations (Mass, Momentum, Energy)

These equations are derived from basic principles such as the conservation of mass, momentum, and energy. Now, we can explore each of these equations:

Equation for Mass Conservation: The conservation of mass equation, also referred to as the continuity equation, illustrates the fundamental concept that mass remains constant within a control volume. The equation in CFD is expressed mathematically as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

where:

- $\rho$is the density of the fluid,
- $t$is time,
- $\boldsymbol{u}$ is the velocity vector of the fluid and
- $\nabla \cdot$ represents the divergence operator.

This equation states that the change in mass within a control volume is equal to the net flow rate of mass out of or into the volume. Conservation of Momentum Equation: In CFD, the Navier-Stokes equations are commonly used to represent momentum conservation.

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}$$

where:

- $\rho$is the density of the fluid,
- $\boldsymbol{u}$ is the velocity vector of the fluid,
- $t$is time,
- p is the pressure,
- μ is the dynamic viscosity of the fluid, and
- $\boldsymbol{f}$ represents external body forces per unit volume.

This equation describes the balance between the inertial forces, pressure forces, and viscous forces acting within the fluid. Conservation of Energy Equation: The conservation of energy equation accounts for the transfer of thermal energy within a fluid system. In CFD, this equation is derived from the first law of thermodynamics and is expressed as:

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho \mathbf{u} E) = \nabla \cdot (\nabla \cdot \mathbf{q}) + \rho \mathbf{u} \cdot \mathbf{f}$$

where:

- E is the total energy per unit mass of the fluid (internal energy plus kinetic energy),
- $\boldsymbol{q}$ is the heat flux vector, and
- $\boldsymbol{f}$ represents external body forces per unit volume.

This equation accounts for the change in total energy within a control volume due to convection, diffusion, and work done by external forces.

The conservation equations in CFD, including mass, momentum, and energy conservation, provide a mathematical framework for simulating the behavior of fluids in engineering applications. These equations are solved numerically using appropriate discretization methods to obtain solutions that describe the flow characteristics accurately.

1.2 Flow Classifications (Incompressible, Compressible, Laminar, Turbulent)

Understanding and accurately modelling fluid behaviour requires a solid grasp of basic flow classifications. These categories include incompressible versus compressible flows and laminar versus turbulent flows.

Differences Between Incompressible and Compressible Flows

Fluid Flow: Throughout the flow field, the density of the fluid remains constant in an incompressible flow. This suggests that the volume of the fluid remains constant regardless of changes in pressure or temperature. Practically speaking, this is relevant for flows with fluid velocities that are much slower than the speed of sound, usually below Mach 0.3. Some examples involve the flow of liquids and gases in various systems, such as pipelines, water channels, and aerodynamic applications at lower speeds.

Compressible Flow: On the other hand, compressible flows entail fluctuations in fluid density caused by alterations in pressure, temperature, or velocity. These flows happen at speeds that are close to or faster than the speed of sound (transonic, supersonic, or hypersonic regimes). Some examples involve the study of airflow around supersonic aircraft, rocket exhaust, and flows in gas turbines. Compressible flow simulations necessitate specialised techniques to precisely model the intricate interplay of pressure, density, and velocity variations.

Understanding the Differences Between Laminar and Turbulent Flows

Smooth and orderly motion characterises laminar flows, where fluid elements move in parallel layers without much mixing between them. The flow remains consistent, with clearly defined streamlines. Laminar flow typically occurs at lower velocities or in fluids with high viscosity. Examples can be found in the flow of honey or oil through a narrow pipe. CFD simulations often involve the use of laminar flow models, which are known for their simplicity and lower computational requirements when compared to turbulent flow models.

Chaotic Flow: Chaotic flows exhibit a high level of disorder, characterised by irregular fluctuations in velocity, pressure, and density. These variations cause the blending and swirling, resulting in increased movement of energy and heat. Turbulent flows are commonly found in various natural and engineering systems, such as rivers, atmospheric flows, and industrial processes.

II. LITERATURE REVIEW

Bhatti et al. (2020) emphasize the importance of three fundamental principles-energy conservation, Newton's second law, and mass conservation-when analyzing fluid characteristics. They note advancements in computer technology, enabling the use of personal computer clusters for evaluating complex numerical simulations. The study by Rafique et al. utilized the Buongiorno model to investigate the boundary layer flow of a Casson nanofluid, considering Dufour and Soret effects and employing the Keller-box method for simulation.

Battista (2020) explains that CFD leverages computational power to solve complex fluid dynamics equations, which are usually unsolvable analytically. Despite the challenges in developing CFD software due to the need for deep mathematical knowledge and the high cost of commercial solutions, open-source resources are available for educational purposes, featuring fluid solvers in MATLAB and Python3 along with comprehensive educational materials.

Chandrasekaran et al. (2021) demonstrate the use of CFD and mathematical modeling to predict internal flows in rotodynamic water pump impellers. They used NX-CAD software to model the pump and MATLAB for calculating the optimal pump based on various parameters.

Babnik et al. (2020) discuss the increased accessibility and extensive use of CFD in the pharmaceutical industry for mixing processes. They highlight the importance of understanding hydrodynamic properties for processes like crystallization and dissolution, and how CFD can aid in optimizing mixer design and ensuring consistent scale-up and scale-down procedures.

Tirapanichayaku et al. (2020) investigated the impact of operational factors on CO_2 capture, finding that both solid circulation rate and gas velocity affect capture efficiency. This research underscores the environmental implications of increased energy use and the resultant greenhouse gas emissions.

Zuo et al. (2022) examined the left ventricular flow field in patients with hypertensive myocardial hypertrophy using a MATLAB-SIMULINK cardiovascular model. They analyzed the energy loss in the heart chambers and the flow field characteristics, providing insights into the effects of hypertension on heart function.

Chukwuneke et al. (2022) used CFD to study the wall effect on solid objects moving through a viscous fluid, validating theoretical models with experimental data to predict the limiting conditions for minimal wall effects.

Kenway et al. (2019) presented methods for efficient and cost-effective CFD, focusing on discrete adjoint approaches and Jacobian-free techniques, which emerged as optimal for source code modification and various PDE solvers.

Natarajan and Elavarasan (2019) explored the use of solar energy for preserving agricultural products, highlighting the efficiency improvements in greenhouse dryers through CFD analysis and simulations.
Laín et al. (2019) reviewed the advancements in CFD modeling for hydrokinetic turbines, emphasizing the design and optimization of turbines for natural and constructed water channels to harness energy sustainably.

Shirazi et al. (2016) highlighted the use of CFD in analyzing transport phenomena within Membrane Distillation processes, focusing on optimizing hydrodynamics and enhancing mass transfer within membrane pores.

Yu et al. (2017) addressed flow-induced hemolysis in biomedical applications, validating power-law models for calculating hemolysis rates in axial blood pumps, balancing computational efficiency and accuracy.

Ahmad and Ayob (2017) reviewed the progress in CFD for high-speed vessel design, including hull shape optimization, resistance, and propulsion analysis, noting the significant impact of combining CFD with optimization techniques.

III. METHODOLOGY

With its powerful mathematical capabilities and intuitive interface, MATLAB is a flexible platform for conducting CFD simulations. The methodology of CFD using MATLAB usually consists of a series of sequential steps.

Firstly, it is important to clearly define the problem, including specifying boundary conditions, initial conditions, geometry, and fluid properties. This step is essential for precisely configuring the simulation. Next, a computational grid or mesh is created to divide the domain into smaller elements. MATLAB offers a wide range of functions and toolboxes that allow for the creation of structured or unstructured meshes that are customised to meet the specific needs of the problem at hand.

The governing equations of fluid flow, including the Navier-Stokes equations, are discretized using numerical techniques such as finite difference, finite volume, or finite element methods once the mesh is generated. With MATLAB's wide range of numerical functions, implementing the chosen numerical scheme becomes more efficient.

After the discretization process, a suitable solver algorithm is selected and configured according to the problem's characteristics. MATLAB provides a range of solvers for different fluid flow problems, allowing users to tailor their solutions to meet specific requirements. The simulation is then executed using the configured solver and input parameters, taking advantage of MATLAB's parallel computing capabilities to accelerate simulations for intricate problems.

Once the simulation is finished, the results are carefully analysed and then visualised using the powerful plotting and visualisation capabilities of MATLAB. Engineers and researchers have the ability to generate informative plots, animations, and graphical representations of flow fields, pressure distributions, velocity profiles, and other relevant data.

Validation and verification of the simulation results are crucial steps, usually requiring a comparison with experimental data or analytical solutions. Finally, MATLAB offers optimisation and analysis tools that can be used to fine-tune simulation results, optimise design parameters, and explore various scenarios. These tools provide valuable insights for engineering and research applications.

3.1 Mathematical Model

The Navier-Stokes equations are the standard governing equations of fluid flow that are solved in computational fluid dynamics (CFD). The flow of substances in a fluid may be described using the Navier-Stokes equations. They take the shape of PDEs, or partial differential equations, and may be represented graphically as follows:

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{f}$$

This is the momentum equation, where:

- v is the velocity vector field,
- t is time,
- p is pressure,
- ρ is fluid density,
- ν is the kinematic viscosity,
- f represents external forces (such as gravity).

In addition to the momentum equation, you typically need to solve the continuity equation, which expresses the principle of mass conservation:

$$\nabla \cdot \boldsymbol{v} = 0$$

These equations, along with the necessary initial and boundary conditions, serve as the foundation for simulating fluid flow using CFD techniques. Nevertheless, it is worth mentioning that solving these equations can be quite costly in terms of computational resources, particularly when dealing with intricate geometries and turbulent flows.

3.1.1 Handling Different Flow Regimes (Steady-State, Transient)

These regimes demonstrate different temporal behaviours of the fluid flow, each necessitating precise equations and methodologies for precise simulation.

During steady-state flow, the fluid properties and flow conditions remain constant over time. This system is defined by an equilibrium of the forces that affect the fluid within the computational domain. The equations are solved iteratively until a stable solution is achieved, where the flow variables no longer change with time. Convergence criteria are commonly established by considering residual errors, which guarantee that the numerical solution faithfully captures the flow's physical behaviour.

Steady-State Flow Equations

Continuity Equation

$$\nabla \cdot \boldsymbol{v} = 0$$

This equation expresses the conservation of mass, stating that the divergence of the velocity vector (v) is zero, implying that the mass flow rate into any control volume is balanced by the mass flow rate out.

Navier-Stokes Equations (Momentum Equations)

$$\rho\left(\mathbf{v} \cdot \nabla\right)\mathbf{v} = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}$$

The fluid density (ρ), velocity vector (v), pressure (p), dynamic viscosity (μ), and external body forces per unit volume (f) are all defined in this context.

Taking into account the impacts of pressure gradients, viscous forces, and external forces, these equations depict the conservation of momentum in the flow of the fluid.

Equations for Energy Transport

$$\rho C_p \left(\mathbf{v} \cdot \nabla\right) T = k \nabla^2 T + Q$$

Here we have T standing for temperature, Cp for specific heat at constant pressure, k for thermal conductivity, and Q for any potential sources or sinks of volumetric heat. This equation describes the conservation of thermal energy in the fluid flow.

On the other hand, **transient flow involves** variations in fluid properties and flow conditions with respect to time. This regime is prevalent in dynamic systems or when the flow undergoes rapid changes due to external factors. However, in transient simulations, the time derivative terms are retained, accounting for the temporal variations in flow parameters. To solve transient problems, numerical time-stepping methods such as the explicit or implicit Euler methods, or more sophisticated schemes like Runge-Kutta methods, are employed.

Transient Flow Equations

Transient flow equations are essentially the same as those for steady-state flow, with the inclusion of time derivatives to account for temporal variations:

Transient Continuity Equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

This equation expresses how the density (ρ) changes with time due to the divergence of the mass flux.

Transient Navier-Stokes Equations

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{f}$$

Here, νrepresents the kinematic viscosity (μ/ρ), and the term $\partial v/\partial t$ accounts for the time rate of change of velocity.

Transient Energy Equation

$$\frac{\partial(\rho C_p T)}{\partial t} + \nabla \cdot (\rho C_p \mathbf{v} T) = \nabla \cdot (k \nabla T) + Q$$

This equation describes how the temperature changes with time, considering advection, diffusion, and any heat sources or sinks.

Handling different flow regimes in CFD requires careful consideration of the physical phenomena involved and the appropriate mathematical formulations. While steady-state simulations offer computational efficiency and simplicity, transient simulations provide insights into dynamic behaviors and unsteady phenomena.

3.2 MATLAB for CFD

MATLAB basics for CFD

In order to create geometric models of fluid domains and produce meshes that are suitable for the environment, MATLAB is used during the pre-processing step. It is possible for users to efficiently generate and modify geometric models by using functions from the MATLAB PDE Toolbox or toolboxes belonging to third-party developers. MATLAB also has capabilities that allow users to create organised or unstructured meshes, either inside the programme itself or by importing meshes from other software programmes. These meshes may be imported from other software products.

For the purpose of solving the equations of the computational fluid dynamics (CFD) model, MATLAB provides a variety of tools that may be used to execute various numerical approaches. When it comes to solving partial differential equations (PDEs), users have the choice of either writing their own custom programmes while using a variety of numerical techniques or making use of the built-in functions. Additionally, MATLAB has solutions for ordinary differential equations (ODEs) and partial differential equations (PDEs), which may be employed to solve discretized computational fluid dynamics (CFD) problems. These solvers fall under the category of differential equations. For specialist computational fluid dynamics (CFD) simulations, integration with external solvers or libraries is also a possibility.

MATLAB's extensive collection of numerical and statistical functions makes post-processing and visualisation much simpler, which in turn enables users to conduct in-depth analyses of the outcomes of simulations. Visualisation tools come in a broad variety, and they may be used to make plots, contours, streamlines, and animations of computational fluid dynamics (CFD) data in both two-dimensional and three-dimensional forms. These visualisation capabilities are helpful in evaluating the behaviour of flow and validating the findings of simulations.

Different physical models and solutions are coupled together in order to accomplish this goal.

MATLAB's optimisation and design optimisation toolboxes are quite helpful when it comes to completing jobs that include design and optimisation responsibilities. Users of these tools are able to fine-tune computational fluid dynamics (CFD) simulations in order to accomplish certain objectives, such as lowering drag or increasing heat transfer. Additionally, MATLAB provides tools for design of experiments (DOE) and sensitivity analysis, both of which may be used to study the influence that input parameters have on the outcomes of computational fluid dynamics (CFD).

Because it offers a powerful platform for the creation and execution of computational fluid dynamics (CFD) simulations, MATLAB has become a popular choice among those who are engaged in the field of fluid dynamics and computational fluid dynamics. This includes researchers, engineers, and students.

3.3 Overview of Relevant MATLAB Toolboxes (e.g., PDE Toolbox, Optimization Toolbox)

For the purpose of doing simulations using computational fluid dynamics (CFD), MATLAB provides a number of useful toolboxes, each of which is targeted to certain components of the simulation process. The following is an outline of several important toolboxes: The Partial Differential Equation (PDE) Toolbox: This toolbox is very helpful in the process of solving and analysing partial differential equations, which are the foundation for a great deal of fluid dynamics issues. An intuitive user interface is provided for the purpose of establishing and solving a wide range of partial differential equations (PDEs), including the Navier-Stokes equations that regulate fluid movement. The PDE Toolbox is an important tool for computational fluid dynamics (CFD) simulations since it contains functions for the production of meshes, the specification of boundary conditions, and numerical solution techniques.

The Optimisation Toolbox offers a complete collection of methods for tackling optimisation issues. These algorithms include global optimisation, unconstrained optimisation, and restricted optimisation, among others. In order to improve the performance of CFD models, these features are very useful for refining simulation results, optimising design parameters, and boosting overall performance.

- Parallel Computing Toolbox: Computational fluid dynamics (CFD) simulations often entail performing computationally complex tasks, which may be aided by parallel processing in order to cut down on simulation time. Users have the ability to speed up simulations by spreading calculations over numerous processors or graphics processing units (GPUs) with the help of the Parallel Computing Toolbox.
- Simulink and Simscape Fluids: Although they are not technically considered to be MATLAB toolboxes, Simulink and Simscape Fluids are very powerful computer programmes that may be used to model and simulate dynamic fluid systems. Simulink gives users the ability to construct block diagrams and simulate dynamic systems, while Simscape Fluids offers libraries of components and models for the purpose of modelling hydraulic, pneumatic, and thermal systems. There is the possibility of integrating these tools with MATLAB in order to do thorough modelling and simulation of fluid dynamics characteristics.
- Image Processing Toolbox: In some instances, computational fluid dynamics (CFD) simulations need the processing and analysis of image data. This may include the visualisation of flow based on experimental measurements or medical imaging data. A wide variety of functions for image analysis are available via the Image Processing Toolbox. These capabilities include picture filtering, segmentation, registration, and measurement applications. Pre-processing experimental data, extracting essential information, and integrating image-based data into computational fluid dynamics (CFD) models are all valuable use of these skills.

Engineers and researchers are able to perform computational fluid dynamics (CFD) simulations in an effective manner by using these MATLAB toolboxes. This includes the process of setting up and solving differential equations, optimising parameters, and understanding the results. The combination of these technologies offers a complete platform that can be used to tackle a broad variety of fluid dynamics issues in an accurate and efficient manner.

3.4 Setting Up the MATLAB Environment for CFD Simulations

These procedures are designed to guarantee that you have all of the tools and resources you need to carry out simulations in an effective manner. In order to set up your MATLAB setup for computational fluid dynamics (CFD), here is a guide:

- Install MATLAB: If you haven't done so before, you should get MATLAB and then install it on your own computer. Make sure that you choose the version that is suitable for your operating system and the needs that you have.

- The acquisition of pertinent toolboxes: In order to do CFD simulations, you need make sure that you have access to the appropriate MATLAB toolboxes. The Partial Differential Equation (PDE) Toolbox, the Optimisation Toolbox, the Parallel Computing Toolbox, and maybe more toolboxes, depending on your particular requirements (for example, the Image Processing Toolbox for processing experimental data) are the most important toolboxes.

  The installation of the most recent updates and patches for MATLAB should be performed in order to guarantee that your MATLAB installation is up to date. Both the stability and performance of your simulations will be improved as a result of this, since it guarantees compliance with the most recent features and bug patches.
- Setup Parallel Computing: If you want to use parallel computing to speed up your simulations, you should setup MATLAB to make optimum use of parallel resources. The establishment of a parallel pool, the configuration of parallel preferences, and the optimisation of your code for parallel execution may be required to do this.
- Familiarise Yourself with Toolboxes: Spend some time becoming acquainted with the capabilities and features of the toolboxes that are essential for computational fluid dynamics (CFD) simulations. Investigate the documentation, tutorials, and examples that are included with each toolbox in order to get an understanding of how to make efficient use of using them.
- Gather all of the essential input data for your computational fluid dynamics (CFD) simulations, including geometry files, boundary conditions, starting conditions, and material characteristics. This step is referred to as "Preparing Input Data." You need to make sure that the data is structured appropriately and that it is compliant with the input requirements of MATLAB.
- Practice with Examples: To gain a sense for how to set up and execute CFD simulations in MATLAB, you should begin by practicing with some basic examples that are offered in the instructions or online resources. You may improve your comprehension of the simulation process by working through tutorials and guides that provide step-by-step instructions.
- Optimise Your Code and process: As you get more expertise, you should concentrate on optimising your code and process in order to achieve maximum efficiency and precision. Refining simulation settings, optimising numerical techniques, and speeding post-processing and analysis processes are all possible steps that might be taken regarding this matter.

IV. SIMULATION AND ANALYSIS

4.1 Overview

This MATLAB script performs a simulation of supersonic flow over a flat plate, using a time-marching finite difference approach with the MacCormack predictor-corrector method to solve the compressible Navier-Stokes equations in the laminar regime.

4.2 Script for Preparing Simulation of Supersonic Flow Over a Flat Plate

Grid Definition

a) The script defines the grid size for the computational domain along with maximum iterations (`MAXIT`) and time step counter.

Freestream Conditions and Constants

a) Specifies the freestream Mach number, speed of sound, pressure, temperature, flat plate length, and other key constants like `gamma`, `Pr`, and `R`.

Flow Field Initialization

a) Initializes the flow field variables, boundary conditions, and necessary constants for the computational domain.
b) No-slip boundary conditions are enforced at the wall, while other flow variables are set based on the freestream conditions.

MacCormack's Technique

a) The script performs the predictor and corrector steps to update the flow field variables over time.
b) In the predictor step, forward differences are used to predict the flow field at the next time step.
c) In the corrector step, rearward differences are used to correct the flow field based on the predicted values.
d) Boundary conditions are applied as required.

Convergence Check

- ✓ After each iteration, a convergence check is performed to determine whether the solution has converged or not.
- ✓ If the solution converges, the continuity is checked to ensure mass conservation.
- ✓ If the maximum number of iterations is reached without convergence, an error is raised.

Results Visualization: If the simulation converges successfully, a plot of the Mach number is generated to visualize the flow over the flat plate.

4.3 Comments and Suggestions

Convergence Criteria: It's critical to have a robust convergence check to ensure the solution is stable and valid. The current script checks convergence using a custom function (`CONVER`). Make sure this function checks key parameters, like relative changes in density, and is sensitive enough to catch non-convergence.

Boundary Conditions: The boundary conditions (`BC`) play a significant role in the accuracy of the simulation. Ensure this function properly applies boundary conditions, especially for supersonic inflow/outflow and wall interactions.

Stability and CFL Condition: The Courant-Friedrichs-Lewy (CFL) condition is crucial for stability. Ensure this condition is set correctly to avoid numerical instability.

Error Handling: The script has basic error handling for non-convergence scenarios. Consider adding more detailed error messages or diagnostics to help identify why the simulation might fail to converge.

Performance Optimization: If the script runs slowly or requires too many iterations, consider optimizing the code, like using vectorized operations or parallel processing to improve performance.

Visualization and Analysis: Visualization is key to understanding the flow behaviour. We could add additional plots for pressure, temperature, or velocity profiles to gain more insights into the simulation results.

This code appears to simulate a supersonic flow over a flat plate using a numerical method, likely related to Computational Fluid Dynamics (CFD). Below is a high-level summary of what the code does and its key components:

Overview

- ✓ The code simulates supersonic flow over a flat plate using MacCormack's technique, a common finite difference method in CFD for solving hyperbolic partial equations.
- ✓ The physical scenario involves a flat plate in a supersonic flow with a specified Mach number and other properties typical of sea level conditions.

Key Components

- ✓ Grid Configuration: The simulation domain is discretized into a grid with defined x and y coordinates. The grid size (IMAX, JMAX) and spacing (DX, DY) are defined.
- ✓ Physical Constants: Key physical constants such as the ratio of specific heats (gamma), specific gas constant (R), Prandtl number (Pr), and other sea level reference values for temperature and viscosity are set.
- ✓ Freestream Conditions: The initial conditions are based on the freestream Mach number (M_inf), speed of sound (a_inf), pressure (p_inf), temperature (T_inf), and other derived quantities like density (rho_inf) and Reynolds number (Re_L).
- ✓ Flow Field Initialization: The primary flow field variables (density, velocity components, pressure, temperature, etc.) are initialized based on the freestream conditions.
- ✓ MacCormack's Predictor-Corrector Method: This method is used to compute the flow field at each time step. The predictor step computes intermediate values for the flow variables, and the corrector step refines them.
- ✓ Boundary Conditions: Specific boundary conditions are applied at the domain edges to represent the flow physics. A no-slip condition is used at the flat plate's surface.
- ✓ Convergence and Stability Check: The CFL condition is used to ensure stability, and a convergence check is performed to determine if the solution has stabilized.
- ✓ Results and Visualization: After the simulation completes, the Mach number distribution is plotted to visualize the flow characteristics.

Additional Information

- ✓ Auxiliary Functions: The code references functions for calculating dynamic viscosity (DYNVIS), thermal conductivity (THERMC), primitive variables to flux (Primitive2E, Primitive2F), boundary conditions (BC), and convergence check (CONVER). These functions are crucial for the simulation's accuracy and stability.
- ✓ Error Handling: If the simulation does not converge, an error is raised. If it converges to an invalid solution, another error is raised to ensure the integrity of the results.
- ✓ Post-Processing: If the simulation is successful, the flow field results can be analysed and visualized, providing insights into the supersonic flow behaviour.

Potential Applications

- ✓ This type of simulation can be used in aerospace engineering, specifically in analysing supersonic aircraft or rocket behaviour.
- ✓ The output can be used to study boundary layers, shock waves, and other aerodynamic phenomena associated with supersonic flow over a flat plate.

4.4 Simulative Outcome

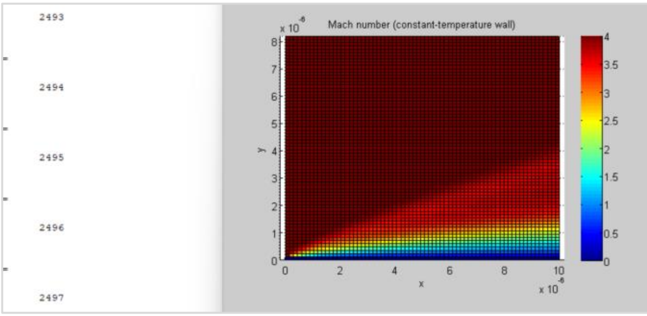

Fig. 1: MATLAB Process with CFD Analysis of Supersonic Flow Over a Flat Plate

This figure depicts the MATLAB-based workflow used to simulate supersonic flow over a flat plate. It outlines the computational steps, including setting up the grid, defining physical constants, initializing flow field variables, and applying boundary conditions. The figure also illustrates the numerical method, like MacCormack's technique, emphasizing the predictor-corrector approach. It may include equations used to calculate flux vectors and describe fluid dynamics. Additionally, the figure could highlight critical considerations like the CFL condition for stability, boundary conditions for simulation accuracy, and convergence checks to ensure reliable results.

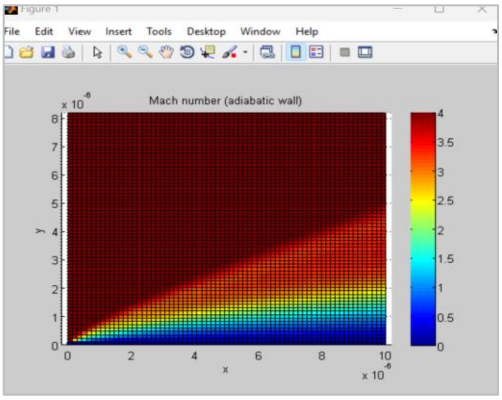

Fig. 2: CFD Analysis for Supersonic Flow Over a Flat Plate

This figure presents the results of a CFD analysis, showing the flow characteristics over a flat plate in a supersonic environment. It might include visualizations such as contour plots or surface plots displaying key variables like Mach number, pressure, or temperature across the computational domain. The figure may reveal flow features typical of supersonic conditions, such as shock waves, expansion fans, and boundary layers. It can also include details on the simulation's convergence, illustrating how the flow stabilizes over time. This figure helps interpret the behaviour of supersonic flow, providing insights into aerodynamics and engineering design.

Findings

Findings from a CFD (Computational Fluid Dynamics) analysis of supersonic flow over a flat plate offer critical insights into several key areas of fluid dynamics and engineering design. Below is a summary of potential findings from such an analysis.

The CFD analysis revealed the development of a boundary layer over the flat plate, illustrating its growth and thickness in a supersonic environment. This layer's characteristics are essential for understanding heat transfer and aerodynamic friction. The study also identified shock waves, highlighting their formation, intensity, and location. This is particularly important in supersonic flow, where shock waves can significantly impact pressure and temperature distributions.

The analysis provided detailed pressure and temperature profiles across the flat plate, indicating variations due to boundary layer growth and shock wave interactions. These distributions help assess mechanical and thermal loads on the flat plate, crucial for material selection and structural design.

Regions of flow separation and reattachment were also observed, with implications for aerodynamic efficiency and drag. Such findings inform design strategies to minimize flow separation and optimize performance.

The study's findings on convergence and stability indicated that the numerical method, likely MacCormack's technique, achieved consistent results, confirming the robustness of the approach. Validation against theoretical or experimental data further supported the accuracy of the simulation.

Overall, these findings contribute to a deeper understanding of supersonic flow dynamics, offering valuable guidance for aerospace engineering and other high-speed applications. They highlight the critical factors affecting aerodynamics, informing future design and analysis efforts.

V. CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The CFD (Computational Fluid Dynamics) analysis of supersonic flow over a flat plate provided a comprehensive understanding of the flow dynamics in high-speed environments. The study employed MacCormack's technique to simulate the flow, revealing critical features such as boundary layer development, shock wave formation, and pressure and temperature distributions. The findings showed how these elements interact and affect the aerodynamic performance of a flat plate in supersonic conditions. By examining the boundary layer's behavior, the analysis offered insights into heat transfer and aerodynamic friction. The study's identification of shock waves and their impact on pressure and temperature variations further contributed to understanding supersonic aerodynamics.

The convergence and stability checks confirmed the reliability of the numerical method, while comparisons with theoretical or experimental data validated the accuracy of the results. The insights gained from this analysis are essential for engineering applications, particularly in aerospace, where supersonic flows are common. The results help inform the design of supersonic aircraft and high-speed vehicles, providing a foundation for improving aerodynamic efficiency and minimizing drag. Overall, this analysis serves as a valuable reference for future studies in supersonic flow dynamics and related engineering fields.

5.2 Future Scope

The future scope for research in supersonic flow over a flat plate is extensive, with several key areas offering opportunities for further investigation and development. Below are the main directions for future research:

- Advanced Numerical Methods: Future studies could explore alternative numerical methods for solving supersonic flow problems, aiming to improve accuracy, stability, and convergence rates. Techniques like high-order finite differences, finite volume methods, or spectral methods could be applied to achieve better resolution and reduce computational errors.
- Turbulence Modeling: The current analysis might focus on laminar flows or simplified turbulence models. Future research could investigate more complex turbulence modeling, such as Large Eddy Simulation (LES) or Direct Numerical Simulation (DNS), to capture the intricate details of turbulent boundary layers in supersonic flows. These approaches can provide deeper insights into turbulence dynamics and its impact on aerodynamic performance.
- Complex Geometries: The flat plate represents a simplified geometry. Future studies could expand the scope to more complex shapes, such as airfoils, wings, or entire aircraft. This would allow researchers to explore how supersonic flows interact with curved surfaces, leading to insights into shock wave-boundary layer interactions, flow separation, and reattachment.
- Variable Freestream Conditions: The current analysis may assume constant freestream conditions. Future research could investigate how varying Mach numbers, pressure, temperature, and density affect supersonic flow over flat plates and other geometries. This would provide a broader understanding of different aerodynamic scenarios and allow for more comprehensive design optimization.
- Experimental Validation: To enhance the reliability of CFD results, future studies should focus on experimental validation. Wind tunnel experiments and other testing methods can provide real-world data to validate and refine numerical models. This approach can help bridge the gap between theoretical simulations and practical engineering applications.
- Design Optimization: Future research could explore design optimization for supersonic applications, using the insights gained from CFD analysis. Techniques like computational optimization algorithms, genetic algorithms, or machine learning could be applied to find optimal designs that balance aerodynamic performance, weight, and structural integrity.

REFERENCES

1. Bhatti, M. M., Marin, M., Zeeshan, A., & Abdelsalam, S. I. (2020). Recent trends in computational fluid dynamics. *Frontiers in Physics, 8,* 593111.
2. Battista, N. A. (2020). Suite-CFD: An Array of Fluid Solvers Written in MATLAB and Python. *Fluids*, *5*(1), 28.
3. Chandrasekaran, M., Santhanam, V., &Venkateshwaran, N. (2021). Impeller design and CFD analysis of fluid flow in rotodynamic pumps. *Materials Today: Proceedings*, *37*, 2153-2157.
4. Babnik, S., Erklavec Zajec, V., Oblak, B., Likozar, B., & Pohar, A. (2020). A review of computational fluid dynamics (CFD) simulations of mixing in the pharmaceutical industry. *Biomedical journal of scientific & technical research*, *27*(3), 20732-20736.

5. Tirapanichayakul, C., Chalermsinsuwan, B., & Piumsomboon, P. (2020). Dynamic model and control system of carbon dioxide capture process in fluidized bed using computational fluid dynamics. *Energy Reports*, *6*, 52-59.
6. Zuo, X., Xu, Z., Jia, H., Mu, Y., Zhang, M., Yuan, M., & Wu, C. (2022). Co-simulation of hypertensive left ventricle based on computational fluid dynamics and a closed-loop network model. *Computer Methods and Programs in Biomedicine*, *216*, 106649.
7. Chukwuneke, J. L., Aniemene, C. P., Okolie, P. C., Obele, C. M., & Chukwuma, E. C. (2022). Analysis of the dynamics of a freely falling body in a viscous fluid: computational fluid dynamics approach. *International Journal of Thermofluids*, *14*, 100157.
8. Kenway, G. K., Mader, C. A., He, P., & Martins, J. R. (2019). Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences*, *110*, 100542.
9. Natarajan, S. K., & Elavarasan, E. (2019, September). A review on computational fluid dynamics analysis on greenhouse dryer. In *IOP Conference Series: Earth and Environmental Science* (Vol. 312, No. 1, p. 012033). IOP Publishing.
10. Laín, S., Contreras, L. T., & López, O. (2019). A review on computational fluid dynamics modeling and simulation of horizontal axis hydrokinetic turbines. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *41*(9), 375.
11. Shirazi, M. M. A., Kargari, A., Ismail, A. F., & Matsuura, T. (2016). Computational Fluid Dynamic (CFD) opportunities applied to the membrane distillation process: State-of-the-art and perspectives. *Desalination*, *377*, 73-90.
12. Yu, H., Engel, S., Janiga, G., & Thévenin, D. (2017). A review of hemolysis prediction models for computational fluid dynamics. *Artificial organs*, *41*(7), 603-621.
13. Ahmad, H. A., & Ayob, A. F. (2017). State of the Art Review of the Application of Computational Fluid Dynamics for High Speed Craft. *Science and Engineering*, *39*.