Eng. Sci., 2022, **20**, 218-235

Engineered Science

DOI: https://dx.doi.org/10.30919/es8d709

# Implementation of Higher-Order PIMPLE Algorithm for Time Marching Analysis of Transonic Wing Compressibility Effects with High Mach Pre-Conditioning

Aravind Karthik M,[1] Srinivas G[1,*] and Nithesh Naik[2]

## Abstract

Computational accuracy and time to perform complex turbulence research remain the two critical metrics for the success of a simulation. This is more challenging for highly-turbulent and high-speed compressible flows attributing to their shock-capturing uncertainty and complex fluid-surface interactions. Recent developments in numerical techniques for high-speed compressible flows have made significant progress but often struggle to strike a balance between speed and accuracy in predicting shock-induced separations. A promising hybrid-central solver based on Pressure IMplicit with splitting of operator for Pressure-Linked Equations (PIMPLE) algorithm and Kurganov-Tadmor (K.T.) scheme addresses this issue. This paper aims to leverage modified implementation of the PIMPLE algorithm to understand the effects of transonic compressibility and shock-capturing over an aircraft wing with effective pre-conditioning and flux stabilization. Juxtaposition of flux stabilized PIMPLE algorithm with Pressure Implicit with Splitting of Operator (PISO), and Large-Eddy Simulations (LES) methods showcases that modified PIMPLE algorithm with corrector terms shows 9.75% faster convergence than PISO and 128% faster convergence than LES, with little to no oscillations. PIMPLE has 94% agreement with LES results for this transonic wing implementation, which can have even better improvements by altering inner and outer corrector parameters. Modified PIMPLE algorithm strikes perfect balance between accuracy and computational time. This study corroborates it as an ideal choice of solver algorithm for high-speed compressible flows over other higher-order turbulence models.

*Keywords:* Transonic flow; PIMPLE algorithm; Flux stabilization; Shock capturing; Computational Fluid Dynamics.
Received: 28 April 2022; Revised: 07 May 2022; Accepted: 09 May 2022.
Article type: Research article.

## 1. Introduction

The dynamics of the seemingly chaotic motion of the fluid have constantly challenged humankind towards the pursuit of understanding fluid physics. Kim *et al*.[1] and Patterson *et al*.[2] showcased computing systems' potential to tackle complex fluid physics, many studies sprouted over the decades, which solidly paved the path for Computational Fluid Dynamics (CFD). CFD in recent decades has become an indispensable utility to understanding fundamental topics in turbulence research and solving flows of industrial interest. Despite its successful implementation in diverse engineering and research domains, a few of the significant limiting factors in tapping the true potential of CFD remain the trade-off between accuracy and time taken for computation. Current research is driven towards developing accurate and hybrid schemes, specifically addressing the compressible and high-speed ionized flow domains. Spectral methods,[3] high-order Finite-Difference (F.D.) methods,[4] discretely energy-preserving schemes,[5,6] and accurate, explicit time integration[7,8] are common features of many academic flow solvers. In addition, recent improvements in essentially non-oscillatory schemes and their hybrid counterparts tend to effectively unwind the issues associated with CFD scalability and applicability.[9] On the flip side, most commercial flow solvers widely stick to higher-order unstructured finite volume schemes, where the non-linear terms are stabilized by outdated yet effective methods like upwind algorithms, implicit segregated algorithms, and total variation diminishing schemes, to name a few. The employment of severely diffusive numerical methods is a frequent feature of most commercial flow solvers,

[1] *Department of Aeronautical and Automobile Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education(MAHE), Manipal, Karnataka, 576104, India.*
[2] *Department of Mechanical and Industrial Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education(MAHE), Manipal, Karnataka, 576104, India.*
**Email:* srinivas.g@manipal.edu (G. Srinivas)

which can have a significant impact on the prediction of unsteady turbulent flows, especially in Large-Eddy Simulations (LES).[9] Schemes developed in academic settings have shown significant improvement in the way recent challenges in CFD can be addressed. However, their widespread utilization is not yet achieved due to the scarcity of literature on their implementation in various use cases and the lack of a comprehensive study. The present study comprehensively addresses the effects of transonic compressibility over a finite wing by implementing a modified Pressure IMplicit with splitting of operator for Pressure-Linked Equations (PIMPLE) algorithm, accounting for effective pre-conditioning and flux stabilization.

Transonic flow is essentially the complex interaction of fluids between the subsonic flow regime and its locally accelerated early supersonic counterpart. This nascent intermediate region of flow, due to its local acceleration over a curved geometry such as in a wing, creates supersonic pockets over the surface, causing discontinuity and a steep shift in the flow properties. In the case of an aircraft, drag reduction is a great challenge, but there is certainly room for improvements. Contrary to most regimes of the flow, usually stretched over the entire geometry, this transonic regime is of prime interest among the research community owing to its flow non-linearity. It is not just arising from the discontinuity contributed by spatially "unpredictable" supersonic pockets but also due to complex transonic wake flutters, turbulence and flow separation during its transition from one flow regime to another.[10] Thus, a model aimed at predicting the compressibility effects of such a flow domain should be able to accurately pinpoint the flow discontinuities, their locations and the complex interactions, all without much compromise in the computational time. Several mathematical and experimentally driven methodologies such as data driven Evolutionary Algorithms (EAs) and Response Surface Methodology (RSM) were developed to address the complex fluid interactions and compressibility effects in transonic regime.[11,12] A greater inclination is also noticed towards advanced techniques like Jameson's rotated difference mathematical modelling, which proved to have substantial impact on wing and airfoil design and thus used as the precursor for several design optimization studies.[13] While the incompressible regime sheds a great deal of light on the recirculation zones and the variation of flow behaviour within them, the compressible regime broadens the understanding of the shock wave phenomena, its structure and other parametric changes across the flow inside the geometry, making it an ideal candidate to explore it as a use case for promising compressible flow algorithms.

## 2. Methodology

Compressible solvers in CFD utilize fluxes both in mass and volumetric scale to calculate the derivative terms over the cell surface.[14] However, their dependency on "how" and "where" density and velocity fields are calculated or updated determines critical flow parameters' computation.

Highlighting the prime difference between academic and commercial solvers is the numerical diffusion, where recent efforts are polarized towards altering existing Field Operation and Manipulation (FOAM) algorithms to dampen the excessive diffusion effects. For instance, Vuorinen *et al*.[15] have introduced a scale-selective mixed central/upwind discretization coupled with diffusion damped Runge Kutta transient scheme, which proved effective specifically in LES. Despite proving to serve diffusion reduction,[16] explicit dampening efforts do not always guarantee the discrete conservation of total kinetic energy within the inviscid limits, thus limiting its versatility.[17]

Numerical algorithms based on approximate Riemann solvers or flux splitting schemes[18] and polyhedral unstructured collocated meshes find their widespread utilization in compressible flow simulations. A few noteworthy algorithms are available, such as Advection Upstream of Splitting Method (AUSM) scheme,[19] AUSM+ scheme,[20,21] Rusanov (Kurganov-Tadmor or K.T.), HLL (Kurganov-Tadmor (KT) / Kurganov-Noelle-Petrova (KNP) or KNP2), monotone Upstream-Centred Schemes for Conservation Laws (MUSCL),[22] Essentially Non-Oscillatory (ENO) schemes,[23] Weighted ENO (WENO) schemes,[23] Piecewise Parabolic Method (PPM)[24] and the Runge-Kutta-Galerkin (RKDG) method.[25] But not many offer significant benefits in better robustness and scalability of the computations, resulting in less oscillatory and faster converging solutions. However, their effective integration in a collocated polyhedral framework has been challenged, due to incorporation of characteristic decomposition and Reimann solvers, adding on to the complexity in computation.

Nessyahu-Tadmor suggested a workaround[26], where they tried the generalization of the Lax-Friedrichs scheme by superseding it with a second-order central difference scheme. One of the notable differences between research-level solvers and commercial solvers is the spike in diffusion and excessive numerical dissipation on the commercial solver side. This second-order implementation aided in partial resolvement of this spike in diffusion, thus rendering less smeared outflows,[17] but took its toll on computational cost and time when using smaller time steps.[26]

Previous implementations of the KT/KNP scheme highlight a few of the flaws that limit its utilization to flows with smaller pressure and temperature gradients, restricting it to flows adhering to ideal gas law and those with low Mach numbers only. This limitation can be attributed to the inability to comply with the Courant number criteria. A workaround for this issue was proposed by Kraposhin *et al*.,[27] who suggested a hybrid implementation of the KT/KNP scheme with the Pressure Implicit with Splitting of Operator (PISO) method, where the former handles the advection fluxes and the latter controls the coupling of pressure, energy and velocity equations. Juxtaposing it with the second-order implementation by Nessyahu-Tadmor, this novel algorithm

did prove effective in minor time step considerations. This method also allows steady-state simulations using a local time-stepping technique for high-speed flows, reducing the computational time needed to obtain a steady-state solution for transonic and supersonic velocities. However, testing with various use cases pointed to a flaw with PISO method-based linking, where for large Courant-Friedrichs-Lewy (CFL) values (>0.1), underestimation of the location of flow discontinuities (namely, shock front, rarefaction waves) was noticed. A fresh set of suggestions was proposed to overcome this drawback, where one was to add multiple iterations per block using the Semi-Implicit Method for Pressure Linked Equations (SIMPLE) algorithm, and the other way was to adopt a new hybrid scheme, coupling the best of PISO and SIMPLE algorithm. The latter was coined the term PIMPLE algorithm and found its broad applicability in tackling several compressible flow engineering problems.

User cases for the PIMPLE algorithm are constantly being explored, and one such implementation was by Cerminara *et al.*[28] who developed a multi-phase solver for OpenFOAM based on the PIMPLE algorithm for the simulation of volcanic ash plumes. This implementation was considerably less diffusive than other algorithm counterparts and showcased greater convergence with higher scalability and faster computation. Few other test cases did prove the effectiveness of the PIMPLE algorithm in being adept for complex geometries and flows, which also made them compatible with robust parallel simulations.

Ultimately, the simplicity of these approximations lies in the expressions for the inviscid fluxes, and it allows its usage for different problems without noteworthy changes in the numerical implementation.

### 2.1 Computational Equations

For a compressible and viscous fluid flowing in a domain with pronounced heat conduction effects, the conservation equations for the mass, momentum and energy[1] in partial differential form can be described from Equations (1), (2) and (3),

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\vec{U}\rho\right) = 0 \tag{1}$$

$$\frac{\partial \rho \vec{U}}{\partial t} + \nabla\left(\vec{U} \times p\vec{U}\right) = \nabla \cdot \hat{\Pi} \tag{2}$$

$$\frac{\partial pe}{\partial t} + \nabla \cdot \left(\vec{U}\rho e\right) = \nabla \cdot (\lambda \nabla T) + \nabla \cdot \left(\hat{\Pi}\vec{U}\right) \tag{3}$$

where, $t$ is the time taken, $\rho$ is the density, $v$ is the specific volume, $\vec{U}$ is the vector field, and $T$ is the temperature. With these parameters, the tension tensor $\hat{\Pi}$ can be formulated and computed as specified in Equations (4a) and (4b),

$$\hat{\Pi} = -p\hat{I} + \hat{\sigma} \tag{4a}$$

$$\hat{\sigma} = -\hat{I}\frac{2}{3}\eta^{\nabla} \cdot \vec{U} + \eta\left(\nabla\vec{U} + \left(\nabla\vec{U}\right)^{T}\right) \tag{4b}$$

Considering $u$ as the internal energy, the total energy $E$ can be defined as (Equation 5):

$$E = u(P,T) + 0.5 \cdot \vec{U} \cdot \vec{U} \tag{5}$$

Using the equation of state in its standard form, the heat conductivity $\lambda$ and the dynamic viscosity $\eta$ are used in their general formulations as mentioned in Equation (6),

$$p = p(\rho,T)\ \lambda = \lambda(p,T)\ and\ \eta = \eta(p,T) \tag{6}$$

Stability being the focus of the simulation, from a numerical standpoint, the energy equation can be expressed in terms of the total enthalpy Equation (7),

$$h_{tot} = h + 0.5 \times \vec{U} \cdot \vec{U} = u + \frac{p}{\rho} + 0.5 \times \vec{U} \cdot \vec{U} \tag{7}$$

Instead of the total energy expression. Thus, the energy equation previously expressed becomes Equation (8),

$$\frac{\partial \rho h_{tot}}{\partial t} + \nabla \cdot \left(\vec{U}\rho h_{tot}\right) - \frac{\partial p}{\partial t} = \nabla \cdot (\lambda \nabla T) + \nabla \cdot \left(\hat{\sigma} \cdot \vec{U}\right) \tag{8}$$

It is customary in many commercial-grade solvers to use total energy expression in its entirety to account for better capture of energy separation phenomena. However, utilization of total enthalpy expression outweighs the benefits of the total energy equation, as the mean kinetic energy and turbulent kinetic energy are treated explicitly, based on previous calculations of the velocity field from momentum equation and modelled equation for the kinetic energy of turbulence.

Another improvement to enhance the stability of the solution would be to express the temperature diffusion term as the enthalpy diffusion term, which by using thermodynamic relations can be described as in Equation (9).

$$dh = T\,ds + v\,dp = C_p\,dT + \left(\frac{\partial h}{\partial p}\right)_T dp \tag{9}$$

The heat flux $\bar{q}$ can hence be represented as Equation (10):

$$\bar{q} = -\lambda \nabla T = \frac{-\lambda}{C_p}\left(\nabla h - \left(\frac{\partial h}{dp}\right)_T \nabla_p\right) \tag{10}$$

Keeping a constant temperature, the derivative of enthalpy with respect to pressure is Equation (11),

$$\left(\frac{dh}{\partial p}\right)_T = v - T\left(\frac{\partial v}{\partial T}\right)_P \tag{11}$$

Pulling the thermodynamic relations corresponding to entropy ($s$), compressibility ($\psi$) and thermal expansion coefficient ($\beta_v$) from Equation (12),[29]

$$\psi = \left(\frac{\partial p}{\partial p}\right)_T; \kappa = -\frac{1}{v}\left(\frac{\partial v}{\partial p}\right)_T = \frac{\psi}{\rho}; \beta_v = \frac{1}{v}\left(\frac{dv}{dT}\right)_p;$$

$$C_P - C_v = v\frac{T\beta_v^2}{\kappa} \tag{12}$$

The heat flux could thus be formulated in Equation (13),

$$\bar{q} = \frac{-\lambda}{Cp}\left(\nabla h - \frac{1}{\rho}\left(1 - T\sqrt{\frac{Cp - C_v}{T}\psi}\right)\nabla p\right) \tag{13}$$

A PIMPLE approach is used over PISO algorithm to support partial convergence of intermediate iterations, which can be altered to its parent SIMPLE or PISO algorithm by changing the inner and outer corrector parameters (loops). Merged PISO-SIMPLE algorithm, *i.e.*, PIMPLE, accounts for large courant numbers and functions seamlessly in the case of smaller and larger time steps, contrary to PISO and SIMPLE. One significant merit of PIMPLE over PISO is the option to specify more than one sub-iteration, which is better, rather

 

than being locked at just one in the case of PISO. Having more sub-iterations naturally opens a larger margin pressure and momentum convergence, thus improving accuracy. The ability algorithm block (Fig. 1), the density is re-calculated with the equation of state as stated by Vallet *et al.*[30] It helps in ensuring the correct relation between Y1 and $\rho$. Existing or new turbulence models more suitable for large density fluctuations[31] could be implemented to compute the turbulent viscosity. Reynolds-Averaged Navier-Stokes (RANS) and LES approach both are suitable.

Finally, the compressibility $\Psi$ of the mixture has been taken as a volumetric average of the compressibility of the phases, as formulated in Equations (14a) and (14b),

$$\frac{\partial(\psi \cdot p)}{\partial t} + \nabla \cdot (\phi_d \cdot p) - \nabla \cdot \left(\frac{\rho}{ap} \cdot \nabla p\right) = 0 \quad (14a)$$

$$\frac{\partial(\psi \cdot p)}{\partial t} + \nabla \cdot (\phi) - \nabla \cdot \left(\frac{\rho}{ap} \cdot \nabla p\right) = 0 \quad (14b)$$

The mass flux, a magnitude normal to cell faces ($\varphi$), is calculated as the inner product of the velocity times the density, as in Equations (15a) and (15b). This mass flux approximation, although not common, is found in compressible flow solvers due to its better stabilization and convergence merits, especially in cases where the fields have nascent shock formation but do not have strong shocks.

$$\phi = (\rho \cdot U)_f \approx (\rho)_f \cdot U_f \quad (15a)$$

$$\phi = \rho \cdot \phi_U \quad (15b)$$

Provided that the mesh and cell sizes are fixed, it is noticed that flux changes with the density and/or the velocity parameters. Therefore, in those regions where fluxes are already updated in a conservative manner, the pressure equation could be skipped, and the corresponding fluxes could be updated either after the mass fraction equation (based on density updated data from the previous time step) or by using the continuity equation or after the velocity equation. And also control the under-relaxation factor in the PIMPLE algorithm makes it a desirable algorithm to explore.

Post solving the turbulence equations, within the PISO nonphysical oscillations of the numerical solution near discontinuities are damped using the Kurganov–Tadmor/KurganovNoelle-Petrova (KT/KNP) scheme for convective fluxes. Even though updating flux changes via changes in the velocity parameter seems like an optimized option, it is to be noted that the mass conservation principle is not exactly enforced based on the velocity parameter. Instead, it is based on the $\Psi$ parameter. Hence, there is an acceptable level of probability for the occurrence of conservation errors. The fact that FOAM based solvers are developed in C++ has huge merit over those solvers written in procedural programming languages like C and FORTRAN, owing to its object-oriented approach in programming involves abstraction, inheritance and polymorphism. These capabilities aid the solver in leveraging the potential of complex mathematical solutions and implementing them in physical models for higher-level mathematical solutions, which otherwise would be an impeding factor in computing numerically.

## 2.2 Case geometry generation

The modified CFD algorithm is used over an aircraft wing to capture the compressibility effects, whose geometry is an approximate representation of the Boeing 737-800 commercial airliner. 3-D geometry of the wing of this Boeing commercial aeroplane was sketched and lofted in Solidworks, with its root airfoil being B737A-BAC449 and the midspan airfoil being B737B-BAC451 as shown in Fig. 2(b). The chord length of aerofoil used is 6 m for profile (root) and 1.4 m for wing-tip. The total span length was 14.907 m with a dihedral angle of 5.4 deg and a washout angle of 2.3 deg (Fig. 2(a)). Owing to capturing the near accurate transonic fluid

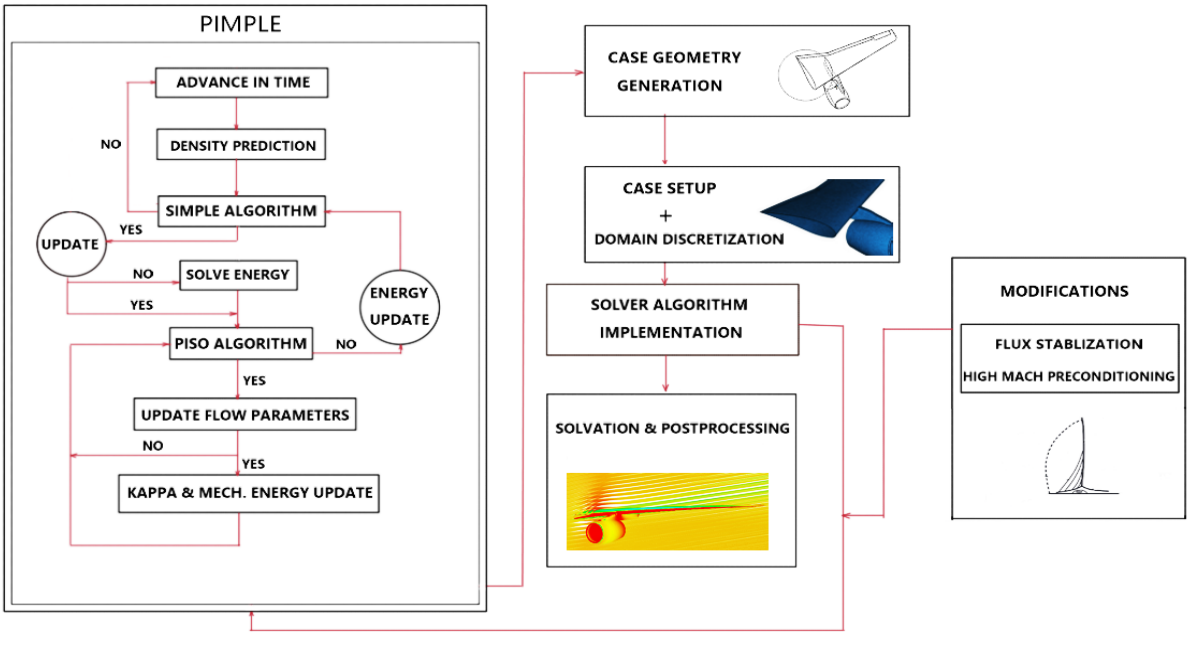


**Fig. 1** PIMPLE Algorithm workflow for transonic shock capturing analysis - flux stabilized.

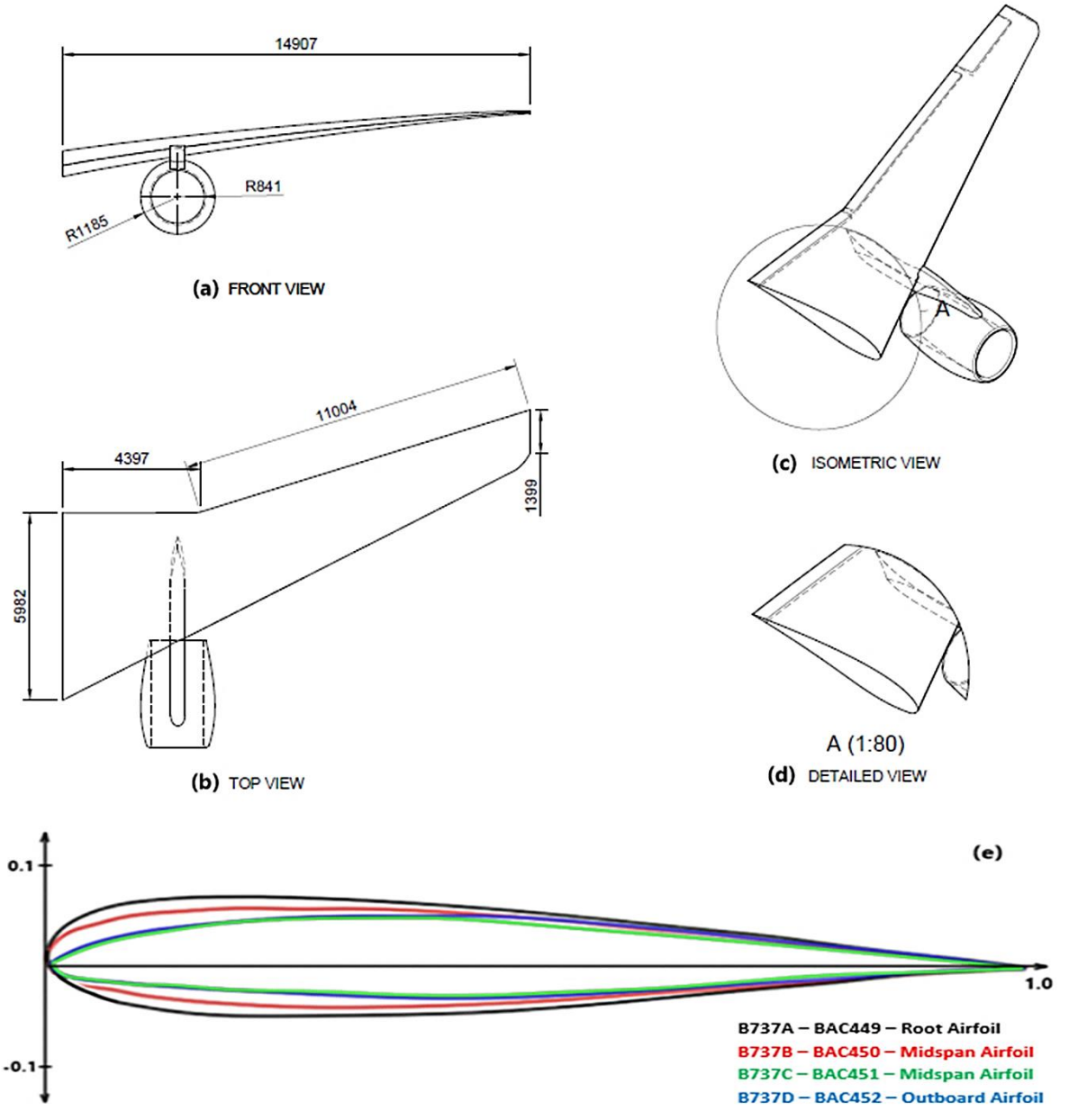


**Fig. 2** Aircraft Wing 3 View Diagram (a) Front View (b) Top View (c) Isometric View (c) Detailed View; (e) Boeing 737-800 Airfoil sections.

interactions and understanding the PIMPLE algorithm's capability better to capture the transonic wake and turbulence amidst high-speed operations and flow discontinuities, a jet engine shroud mounted under the wing with a streamlined pylon was also modelled.

The virtual cuboid-shaped enclosure mimicking a virtual wind tunnel was created using the computer-aided design tool. In any CFD study, choosing the right domain dimension and shape is critical. It should be large enough to account for backflow caused by turbulent dissipation, wake, and the creation of wing-tip vortices. With the variable H=10 m as the multiplier, a computational enclosure with the height of the domain being 30H, the width of the domain as 10H and the length of the domain as 30H was modelled. The wing structure was positioned at a distance of 10H from the inlet of the enclosure, which, from a fluid dynamic standpoint, allows sufficient zone for the incoming flow to completely develop, leaving behind 20H at the rear for effective capturing of the post surface interaction wake (Fig. 3(a)). Fluid wake zone is of prime importance in the analysis of the critical aerodynamic characteristics and is a key factor in understanding the diffusive effects of flow post interaction with the surface while accounting for minute gradients in fluid properties is required to be accurately captured. In this regard, two bodies of influence, as shown in Fig. 3(b), is created in the vicinity for better discretization. Domain A captures the fluid wake zone, while the Domain B captures the near geometry flow discontinuities and accounts for geometry preservation by aiding in small surface mesh refinement. The dimensions of the geometric entities are summarized in Table 1.

 

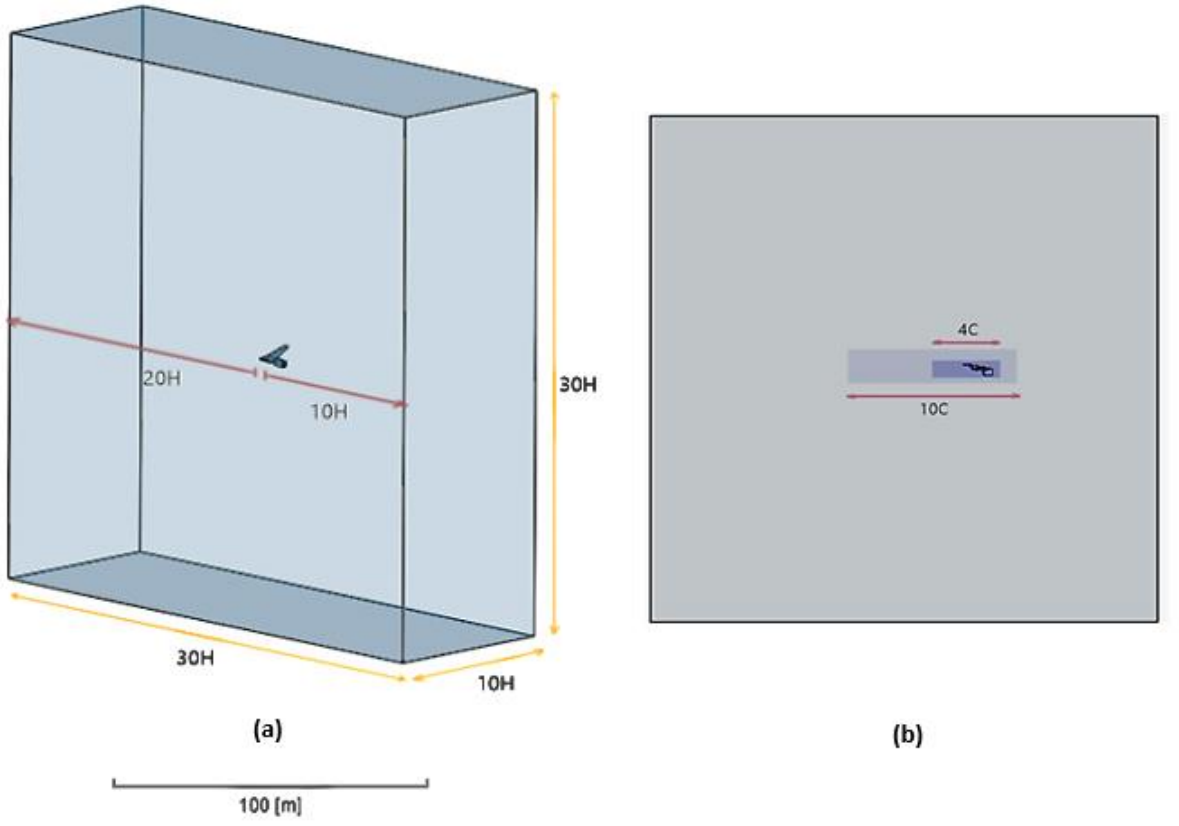


**Fig. 3** Computational flow domain (a) Dimensions (b) Refinement zone dimensions (C - Root chord length).

**Table 1.** Aircraft Wing key geometry information.

| Geometric Entity | Length | Width | Height |
|---|---|---|---|
| Enclosure | 300 m | 100 m | 300 m |
| Domain A (Refinement) | 10C = 10*6 = 60 m | 150 m | 15 m |
| Domain B (Refinement) | 4C = 4*6 = 24 m | 150 m | 9 m |

### 2.3 Simulation definition - Transonic study

In the formulation of finite element or volume, the equations of motion are solved within the small grid elements constituting the entire spatial flow domain with matching conditions between elements, leading to a systematic algebraic equation that can be solved numerically. The choice of discretization scheme depends on the prevailing flow conditions

The flow domain within which the aircraft wing is assigned a material property of air, whose flow interaction is governed by the sensible internal energy flow scheme. This energy scheme aids in using the internal energy form of the equation without the heat of formation and incorporates energy change due to reactions. Considering the steep gradients in the flow properties across the discontinuities, this is an ideal option to go ahead with. For this transonic flow, the Prandtl number, which is a relative measure of the momentum transfer to that of energy transfer, was fixed at 0.713. Prandtl number signifies the thickness of the thermal boundary layer and the thickness of the hydrodynamic boundary layer. If it is equal to one, it signifies that the thickness of the thermal boundary layer is equal to that of the velocity boundary layer. This parameter is of significance in the simulation test case setup, as it sheds light on how fast the thermal diffusion takes place in comparison to momentum diffusion and conveys critical information on the thickness of the thermal boundary layer in relation to the momentum boundary layer.[32]

The transport equation was not chosen to be based on the Sutherland equation, contrary to the common consideration in predominant literature on high-speed compressible flows. This choice was driven by the motive to simplify the computation whilst ensuring adequate accuracy for the flow regime under consideration. Scientific sources point out that viscous effects play a vital role in the highly temperature-dependent flows and could be fairly altered with a constant viscous term for those flows where temperature gradients are not steep. Transonic flow being under the latter category can safely be approximated with constant dynamic viscosity.

The incoming flow from the inlet, considered as a velocity inlet (Fig. 4), was assigned a fixed value velocity of 288.85 m/s (Mach 0.85), and ambient pressure and temperature at the inlet were used. For better flow development and stabilization in due course of the transient study, a linear variation in the velocity (flow ramping), starting from 28.885 m/s to 288.85 m/s, was considered till about 30% of the total iterations (Fig. 5). Such gradual ramping during the initial calculations have

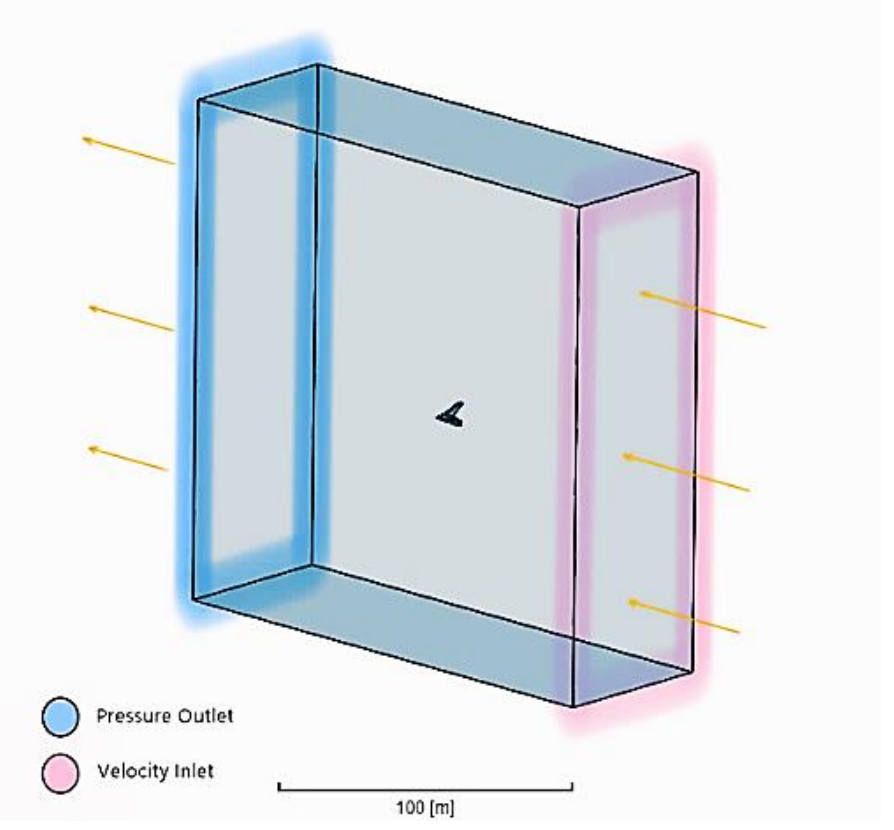


**Fig. 4** Inlet and Boundary conditions of the flow domain under consideration.

also proven effective in attaining a relatively faster convergence. The pressure at the outlet was set at zero Pascal to avoid the reverse flow of the outflowing turbulent air. For the initial steady simulation run, the turbulent intensity was set at 2.77%, based on the assumed Reynolds number and size of the inlet frontier. Using the relationship ($I = 0.16 \times Re^{-18}$), the corresponding *Re* number equals $1.239 \times 10^6$. Boundaries of the bounding box at the far end were considered to be walls with slip conditions, and those on the symmetric end of the wing were assigned the symmetry boundary condition.

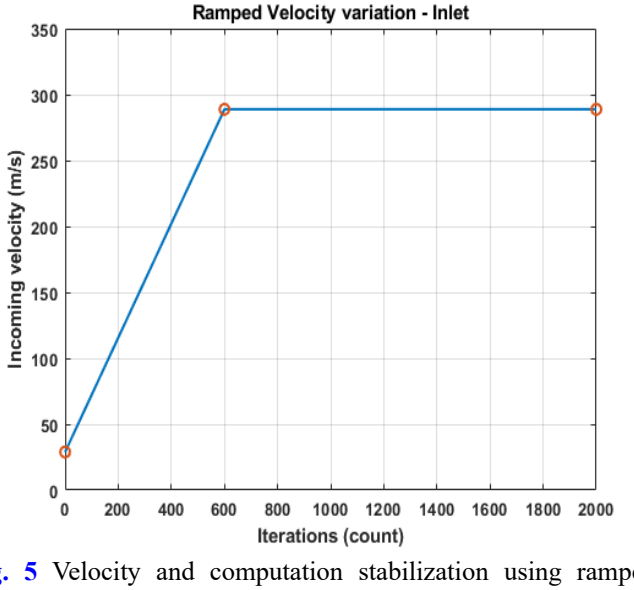


**Fig. 5** Velocity and computation stabilization using ramped velocity input.

### 2.4 Spatial Discretization and Mesh Sensitivity Study

Previous studies corroborate that the element size directly influences the accuracy of the computational results. This depends on the kind of algorithm adopted to decompose the domain. While a highly fine mesh yields reliable numerical predictions with minimal error in contrast to its coarser counterpart, a finer mesh definitely takes its toll on the computational cost, time and effort. Traditionally, striking the perfect balance between these criteria was based on the engineer's experience in setting up the case. But recent research-grade automated domain discretizers make physics informed and parameter controlled meshes possible.[33] In the present study, the entire domain was discretized using the Standard FOAM meshing algorithm, where the geometric and the physical nuances of the model are taken into account for generating a highly automated and unstructured mesh. The essential advantage of using unstructured mesh over a structured one is to account for high flow gradients and thus allow point intersections. The mesh automation, which makes use of the Delaunay algorithm, remains the prime choice over other explicit meshing algorithms, as no grid points intersect with each other at the boundary surface and the ability to connect points and recover the boundary layer is possible, especially in unstructured surface and volumetric meshed setting. Considering the body of influence, the element size is gradually telescoped to more refined sizes in the geometry vicinity. The resulting finalized discretized geometry contained a total of 2.5 million cells with a maximum precision of entities of about 0.02544 m. Body of influence

 

Domain A was manually refined, limited to a maximum edge length of about 0.8 m and Domain B with a maximum edge length of 0.5 m. To better capture the shear effects, boundary layer and viscous shock layer to enhance the prediction of surface dependent functions such as skin friction coefficient, pressure coefficient, and others in the wing structure vicinity, an inflation layer is additionally considered with 8 layers and a growth rate of 1.46 to achieve a $y^+$ value of 1 (Fig. 6), as governed by Equation (16).

$$y^+ = \frac{u_\tau \Delta y_1 \rho}{\mu} \tag{16}$$

where, $u_\tau$ is the frictional velocity and $\Delta y_1$ is the desired first layer height.

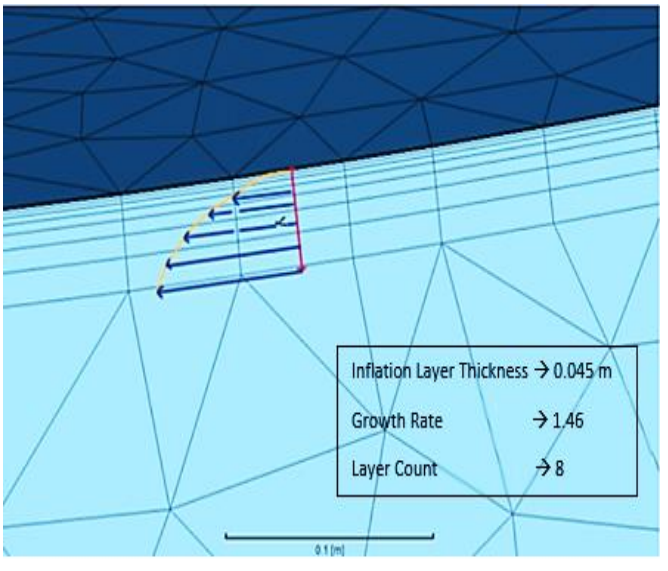


**Fig. 6** Inflation layer first layer thickness computation using boundary layer thickness.

Capturing the viscous boundary sublayer in a turbulent flow is critical in determining the tugging action of the flow in the wing vicinity. Although a rule of thumb is to maintain $y^+$ value at 1 for viscous dominant turbulent flows with relatively moderate Reynolds number of operation, the baselining of the $y^+$ non-dimensional parameter depends on surrounding flow density, skin friction, boundary layer thickness and the governing flow physics. Researchers are guided to a comprehensive study by Sijal Ahmed to understand the selection of $y^+$ parameter specific to the dominant physical traits in regime of operation.[34] Moreover, the entire aircraft wing zone was meshed with a hybrid hex element core scheme with an element size of 0.07 m so that wherever possible, the algorithm combines the tetrahedral/pyramid elements with hexahedral elements, ensuring better mesh quality in critical zones (Fig. 7(b)).

The specified finalized mesh shown in Fig. 7(a) is actually an outcome of a grid independence study, selected post comparison with critical computational parameters from different possible grid refinements. A grid independence study is carried out to ensure that the discretization technique is reliable enough to no longer rely on the element sizes. To understand this, the results of lift and drag coefficients were compared for the same case setup and solver setting (addressed in 4.4). The absolute convergence criteria were set at $1\times10^{-6}$ to allow the solution to converge. From this study, it could be inferred from Figs. 8(a) and 8(b) that the discretization method attained unconditional stability after 2.5 million elements for the 3D geometry.

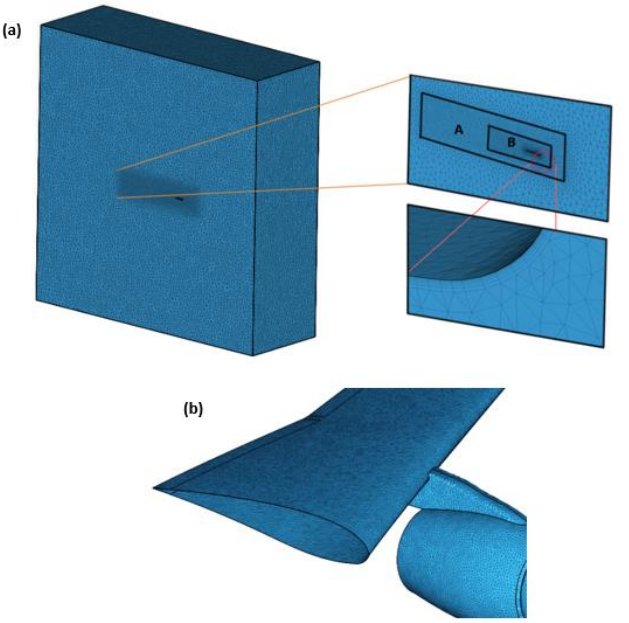


**Fig. 7** Computational Domain with (a) Discretized domain with two levels of refinement and near boundary inflation layer (b) Local refinement over wing surface with a proliferation of nodal points at zones with higher curvature

A major issue in employing FOAM based mesh generation schemes compared to established commercial solver mesh generation utility is to account for the cell volume and quality aberrations and carry on with the simulation with adaptive solver control. A comparative study between OpenFOAM and FLUENT by Prasanna *et al.*[35] indicated that ANSYS FLUENT was able to simulate the case for those discretized geometries even with maximum skewness of 0.93. However, OpenFOAM struggled to simulate the case for such a high skewness value and successfully simulated the study only for those with maximum skewness of around 0.88. Considering the merits of opting for FOAM based solvers over commercial solvers, where tailoring of the solvers is possible, it was decided to resort to the former while ensuring a better quality of the mesh.

The body and edge sizes were changed during meshing to keep the element skewness below the specified limit (0.880). The average element quality and skewness obtained here are 0.68 and 0.13, respectively. For nesting the mesh at proximities, the curvature angle was lowered to 2 deg. In addition to this, the small feature suppression was considered to be $1\times10^{-5}$ and the global gradation rate was taken as 1.22 to better account for the transition from the small segments of the grid to the larger counterparts. A summary of the critical mesh

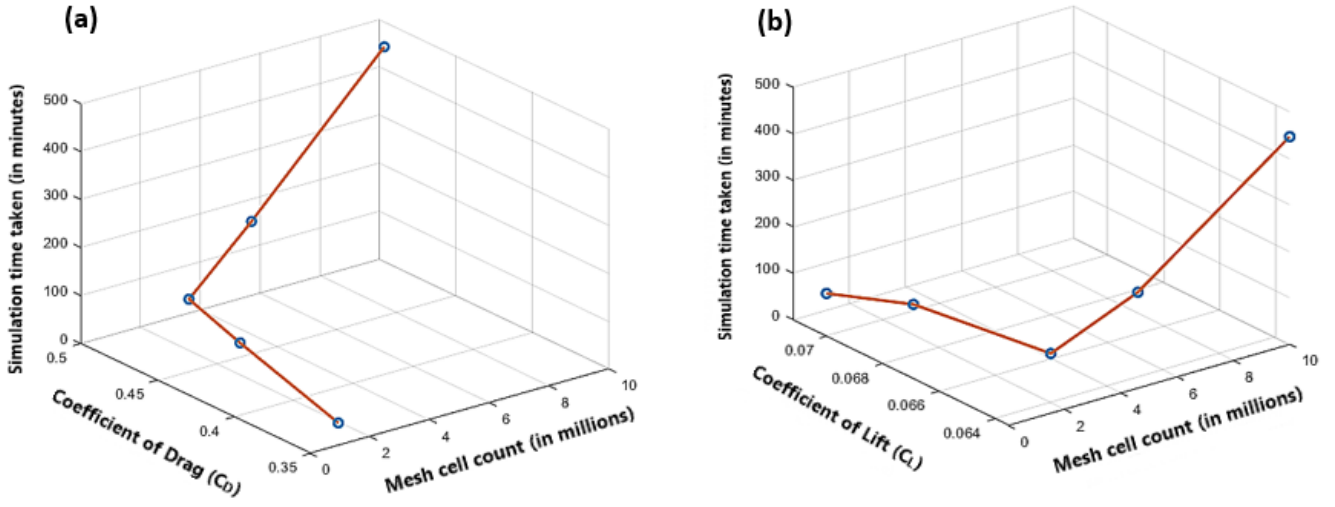


**Fig. 8** Grid independence study (a) $C_L$ variation comparison for varying levels of grid refinement (b) $C_D$ variation comparison for varying levels of grid refinement.

**Table 2.** Mesh Quality metric.

| Property | Minimum | Maximum | Average | Zone of concern |
|---|---|---|---|---|
| Non-Orthogonality | 0 | 86.815 | 12.85 | Inflation Layer near larger curvatures |
| Skewness | 0 | 5.526 | 0.13 | Cowling and Pylon |
| Aspect Ratio | 0.006 | 16.06 | 1.1 | Leading edge |
| Edge Ratio | 0.000085 | 3.4434 | 1.7217425 | Inflation layer |
| Overall mesh quality | | 0.679 | | |
| Number of cells | | 2549859 | | |

quality metrics is tabulated in Table 2 and detailed in Fig. 9. It can be inferred that mesh defeaturing, although implemented in the mesh vicinity, is clustered mostly along surfaces of higher curvature and refinement zone (Fig. 9). These zones of concerns (Table 2) however lie within the threshold limit to qualify as a good mesh and thus can be baselined for the current study.

### 2.5 Solver setup

Evaluation of the governing equations to render output for the PIMPLE algorithm was done by using a pressure-based solver, accounting for the advection fluxes and small-time step considerations. The main advantage of a pressure-based solver is that it requires less computational resources than a density-based solver due to the segregation between the governing equations.

The governing equations are solved sequentially through the solution algorithm. A preliminary steady-state solution was carried out using the k-omega SST (Shear Stress Transport) turbulence model as it provides a better prediction of flow separation and adverse pressure gradients than most Reynolds-Averaged Navier-Stokes (RANS) models. It can account for the transport of the principal shear stress in adverse pressure gradient boundary layers. Its ability to consider principal shear stress in negative gradients of pressure within the boundary layers makes it an ideal candidate for several turbulence modelling applications.

K-omega SST is a modified form of k-omega, one of the most common turbulence models. It belongs to the category of RANS set of models. In addition to the solvation of the conventional Navier-Stokes equation, this model solves two additional partial differential transport equations, which aids in accounting for the diffusion dynamics of the turbulent energy. By consideration of two transport variables, namely, turbulent kinetic energy ($k$) responsible for capturing the energy in the turbulence and the specific turbulent dissipation rate (omega), which determines the rate of dissipation in relation to the '$k$' parameter, this model effectively predicts the influences of turbulence in the flow. While k-omega has an excellent performance in the near-wall flow predictions, this model tends to overshoot flow separation predictions and shear effects due to its ingrained low Reynolds number formulation. K-omega SST fills the gap created by standard k-omega by toggling to the k-epsilon behaviours to avoid the problem of sensitivity to inlet freestream turbulence effects. The turbulent energy $k$ is given in Equation (17),

$$k = \frac{3}{2}(UI)^2 \tag{17}$$

where, $U$ is the mean flow velocity, and $I$ is the turbulence intensity

The specific turbulent dissipation rate can be calculated using the following Equation (18),

 

$$\omega = C_{\mu}^{\frac{3}{4}}\left(\frac{k^{\frac{1}{2}}}{l}\right) \quad (18)$$

where, $C_{\mu}$ is the turbulence model constant, which usually takes the value 0.09, $k$ is the turbulent energy, $l$ is the turbulent length scale which describes the size of large energy-containing eddies in a turbulent flow.

The numerical prediction loop is iterated to get a converged solution since the governing equations are coupled and nonlinearly linked. Relaxation factor (α), which is a technique used for better stability of calculations in the steady-state analysis, was considered to be FOAM automated. This factor limits the amount by which flow properties transition from one iteration to the next iteration and controls the under-relaxation of the solution. Since transonic flow predictions coupled with turbulence considerations are non-linear, it is important to control the variable's change, probing the solution's divergence.

- α < 1 denotes under-relaxation, which slows down the convergence rate but increases the stability.
- α = 1 denotes no relaxation at all and thus uses directly computed values without any cap
- α > 1 signifies over-relaxation, which may be used to accelerate the convergence rate but at the cost of stability.

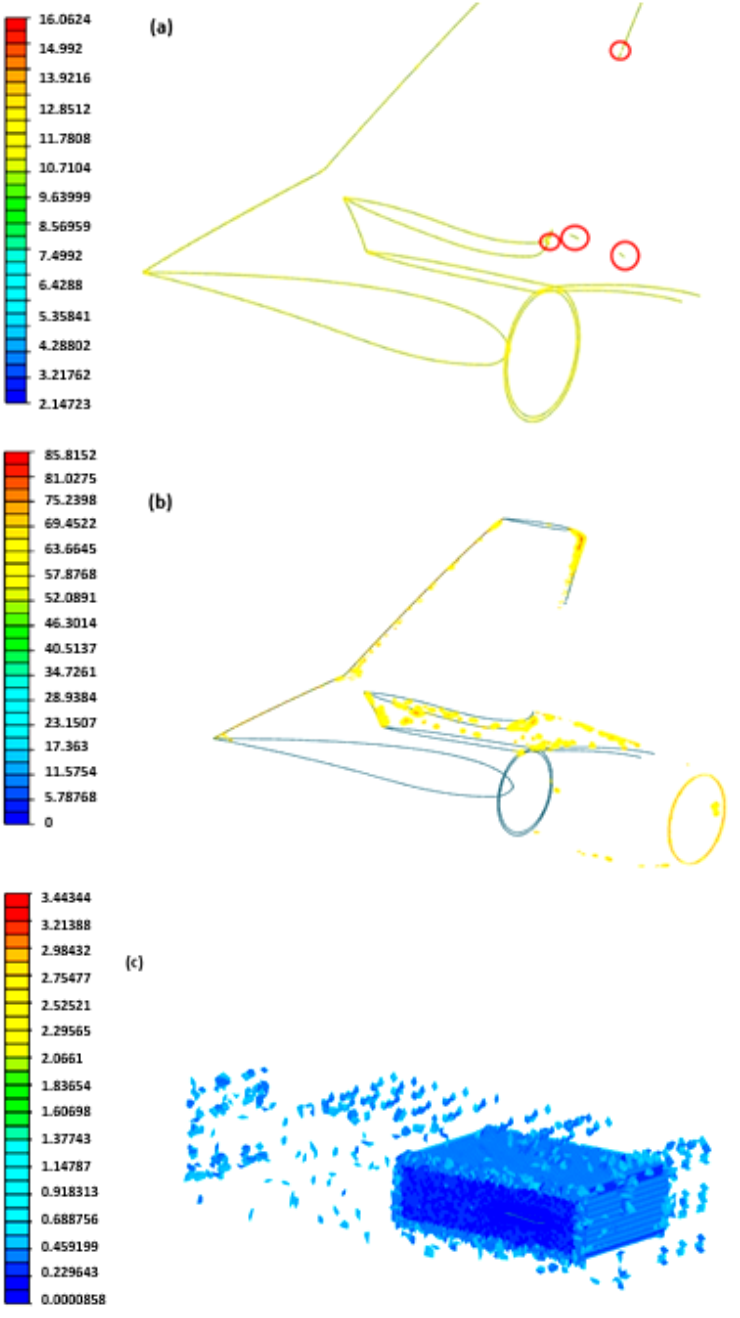


**Fig. 9** Scoping into elements over the discretized domain with (a) Critical Aspect ratio (b) Critical Non-orthogonality (c) Critical mesh edge length ratio.

While the auto-relaxation factor proves effective, if divergence is noticed in due course of computation, it is always advisable to resort to manual relaxation factor assignment, where values between 0.3 and 0.7 are recommended.

With the PIMPLE algorithm, the stability of the computation can further be improved by fiddling with the inner and outer corrector parameters. The number of inner correctors is the number of times the pressure is corrected within an iteration, and the outer connectors denote the number of times each time step/iteration is to be recomputed. In this study, an inner correction factor of 2 and an outer correction factor of 2 are considered for better treatment of the shock prevalent flow with steep variation in the flow properties. An outer corrector of 1 is synonymous with the PISO algorithm, from which PIMPLE is derived and hence will not be chosen. The stabilization settings for the PIMPLE algorithm are summarized in Table 3.

**Table 3.** PIMPLE Algorithm specific stabilization settings.

| Relaxation type | Automatic |
|---|---|
| Relaxation factor | 0.55 |
| Inner corrector | 2 |
| Outer corrector | 2 |
| Absolute tolerance | $1.001\times10^{-5}$ |

FOAM based CFD solvers offer the capability to assign individual solver control for each of the critical flow parameters. The most commonly used CFD solvers for these flow parameters are Pre-conditioned Bi-Conjugate Gradient (PBiCG), Pre-conditioned Conjugate Gadient (PCG), and Geometric Agglomerated Algebraic Multigrid (GAMG), and Smooth solver, to name a few. These are multi-grid solvers, and the basic idea behind them is to use a coarse grid for hiking the solution computation time in local regions of interest, thus aiding in flattening out the high-frequency errors and generating a good starting solution. In GAMG, the mesh is coarsed in steps where the coarsening algorithm is an algebraic pair. PCG aids in symmetric matrix solvation with better parallel framework scalability. But for the current study, PBiCG and Smooth solvers are of prime interest. These two solvers, which will be used for the solver control of transonic flow properties in this study, are highly influenced by the kind of pre-conditioning they undergo. This is where the third modification, high Mach pre-conditioning comes into play. The idea behind high Mach pre-conditioning is to switch between pressure and density-based solvers with adaptations to the PDEs at the discretization level, ultimately to produce a hyperbolic PDE that will converge to the same steady-state with wave speeds that are closer in value, thus having much better convergence characteristics. Discussion of the mathematical nuances of the pre-conditioning is beyond the scope of the current study. The readers are directed to Xue Song *et al*.[36] to further explore the pre-conditioning. In short, PBiCG is fast but not stable, and Smooth solver is stable but slow. For this study, each flow properties are assigned specific solvers in agreement with the physical and computational requirements, as tabulated in Table 4.

**Table 4.** Critical flow parameter specific Solver settings.

| Parameter | Solver |
|---|---|
| Velocity (U) | Smooth Solver |
| Pressure (P) | GAMG |
| Internal energy (e) | Smooth Solver |
| Enthalpy (h) | Smooth Solver |
| Trub. K.E. (k) | Smooth Solver |
| Specific Dissipation rate (omega) | Smooth Solver |

Finally, for the convective terms of the governing equations, a Green-Gauss cell-based spatial discretization gradient is chosen, whereas second-order accuracy is chosen for the viscous terms. This pressure correction works as a constraint on the velocity field to satisfy the continuity equation. Regulation control schemes are specifically to be assigned for the gradient, divergence, Laplacian parameters, interpolation and surface normal gradient. The assigned schemes for each of the above-specified parameters are tabulated in Table 5.

**Table 5.** Regulation control schemes for solver stabilization and optimization properties.

| Property | Scheme | Limiter coefficient |
|---|---|---|
| Time Differentiation | Steady-state | N/A |
| Gradient | Cell-limited least squares | 1 |
| Divergence | Gauss Linear | N/A |
| Laplacian | Gauss Linear Limited | 0.5 |
| Interpolation | Linear | N/A |
| Surface normal gradient | Limited | 0.5 |

From the initial simulation the residuals were found to converge after 1,300 iterations with a convergence criterion of $10^{-6}$. The simulation was hosted in a remote cloud system having a 30 vcore hyperthreaded, overclocked Intel-based processor operating at 2.6 GHz and a RAM of 16 GB. For setting up the job in the parallel framework, the Scotch decomposition algorithm was utilized. Decomposition algorithms are methods used for parallel computing in FOAM based solvers, which conventionally utilize the public domain OpenMPI implementation of the standard Message Passing Interface (MPI). Compared to Simple and Hierarchical algorithm counterparts, which require manual specification of the domain split and decomposition, the Scotch algorithm, depending on the processor loads and available computing power, self-assigns the task of domain decomposition to specific cores and GPUs to accelerate the computation whilst

 

ensuring minimum resource utilization. In scenarios where different processors have different loads, manual override feature is available by using the 'processorWeights' keyword in FOAM Linux based implementation.

## 3. Results and discussions

For the above-specified case, the simulation, with the available computation resource, took about 150 minutes to complete in the cloud. The current model being an approximate representation of the Boeing 737-800 aircraft wing, data validation with experimental data would not be an accurate and justifiable representation to validate the modified PIMPLE algorithm-based solution. In addition, most available scientific pieces of literature that seem to validate their wing models with that of Boeing's wing tend to be questionable, as the data restrictions followed within this corporate firm make availability of accurate geometric and experimental data of specific components of the aircraft void. Hence, the obtained results from the flux stabilized modified PIMPLE algorithm are juxtaposed with LES Smagorinsky transient results for turbulent flow and with the PISO algorithm with fewer inner correctors. Direct Numerical Simulation (DNS) and RANS are on either end of the spectrum where the Navier-Stokes equation is directly solved for the former, whereas in the latter, the turbulent fluctuation (Reynold's tensor) is modelled for all the scales at one node by means of averaging of the turbulent properties and critical flow parameters. LES is an intermediary turbulence modelling where only the turbulent fluctuation (Leonard tensor) below the filter size is modelled. All the scales larger than this size are considered as `perfectly' (or directly in the sense of DNS) solved. Basically, RANS equations are derived by taking a time average of the N.S. equations. The effect of turbulence is simulated through modelling the Reynolds stresses. In comparison, LES is not a time average. The NS equations are solved on a large scale to a small scale to resolve the eddies in the turbulent flow. Only very small eddies are 'averaged' on a sub-grid scale (smaller than 4 elements). An LES simulation is always unsteady. Hence the purpose of LES simulation in the present study is to use it as pseudo-accurate/near correct experimental results owing to its simulation accuracy and physics representations.

Interpretation of the obtained velocity streamlines in the mid-chord of the wing from the PIMPLE algorithm clearly depicts the effects of shear in the near boundary zone, thereby inducing drag (Fig. 10). The larger the cavitation form, the greater the drag force experienced by the aircraft. Critical flow features such as flow separation initiation, boundary layer flow, and swirl/stagnation points are quite conspicuous in the generated contour, indicating a local acceleration to 376.3 m/s velocity over the suction surface of the wing and a zone of perturbed air in the vicinity of the wing surface.

In addition to velocity prediction, pressure variables are also obtained (Fig. 11). High static pressure at lower parts of the wing is paramount to maintaining the effective lift of the aircraft. As the effects of viscosity dons from the wing's surface towards the free stream flow, the shear between the travelling air and the airfoil surface is well noticed from the obtained contours. The choice of an algorithm that accurately captures near-boundary effects is critical in aircraft design since the higher the shear rate, the higher the induced drag and thus the higher the fuel consumption. The small zone of airflow recirculation at the trailing edge might induce unbalanced static pressure distribution, which, if not probed and controlled, creates a constant action of pitching up motion of the aircraft. The penalty associated with this phenomenon is the creation of disturbance in the airflow that trails behind the flight path. Furthermore, the airflow is interrupted by the engine structure, resulting in a mixed flow of air at the trailing part of the engine, according to an inspection of the flow streamlines for the entire wing. The development of wakes and

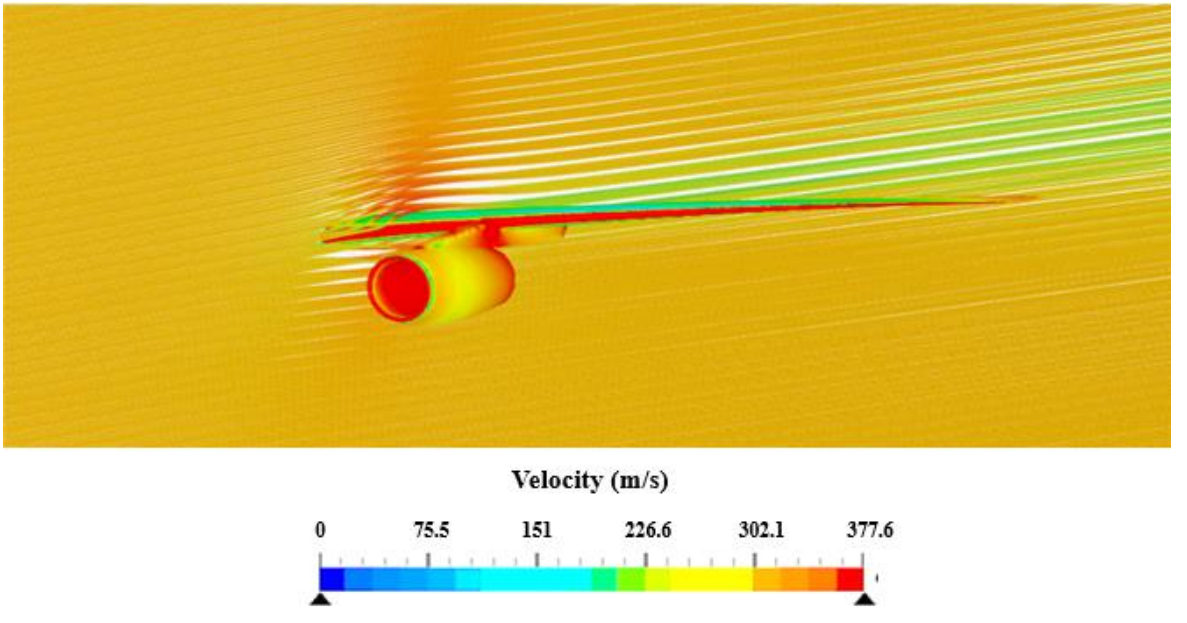


**Fig. 10** Velocity Streamline plot over the aircraft wing.

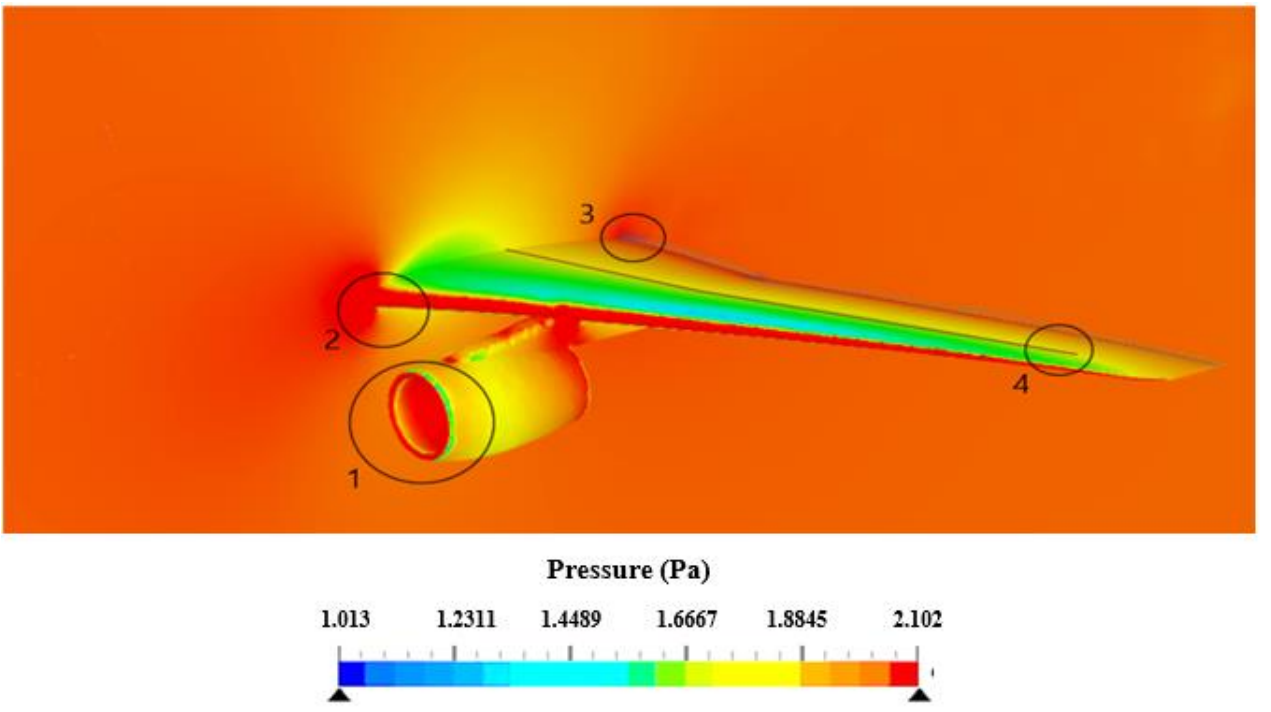


**Fig. 11** Static Pressure contour over the aircraft wing. Key inferences (1, 2) Localized stagnation of flow (3) Nascent flow recirculation zone attributed to engine nacelle induced disturbance (4) Solid line (Brown line) indicating the development of nascent weak shock near the mid-span of the aircraft wing.

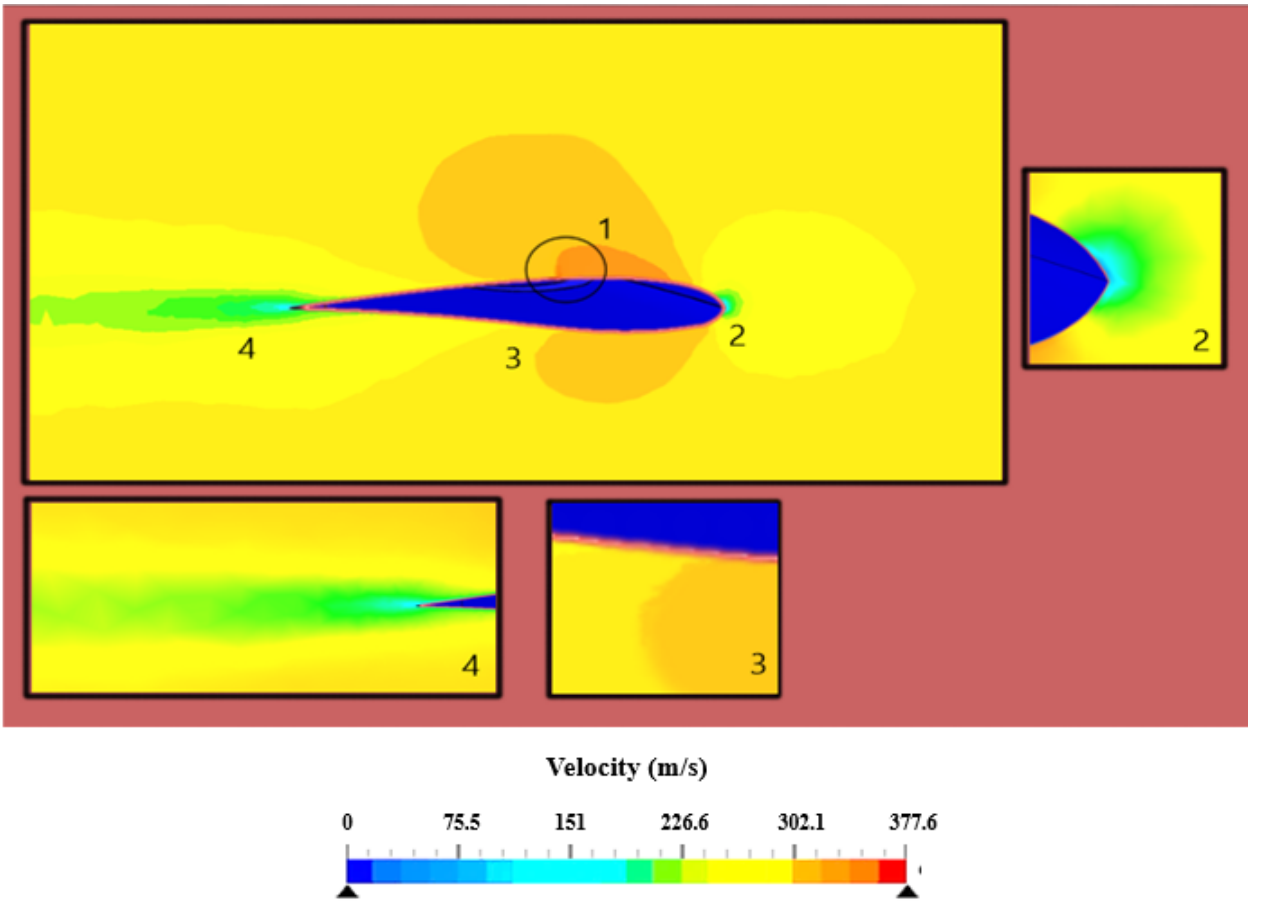


**Fig. 12** Banded velocity contour plot. key inferences (1) Formation of nascent shock over the suction surface of the aircraft wing. Presence of washout angle could be attributed to the difference in the localized velocity magnitude attained. (2) Flow stagnation zone (3) Smearing out of velocity contours due to partially prevalent diffusive effects when using the PIMPLE algorithm.

 

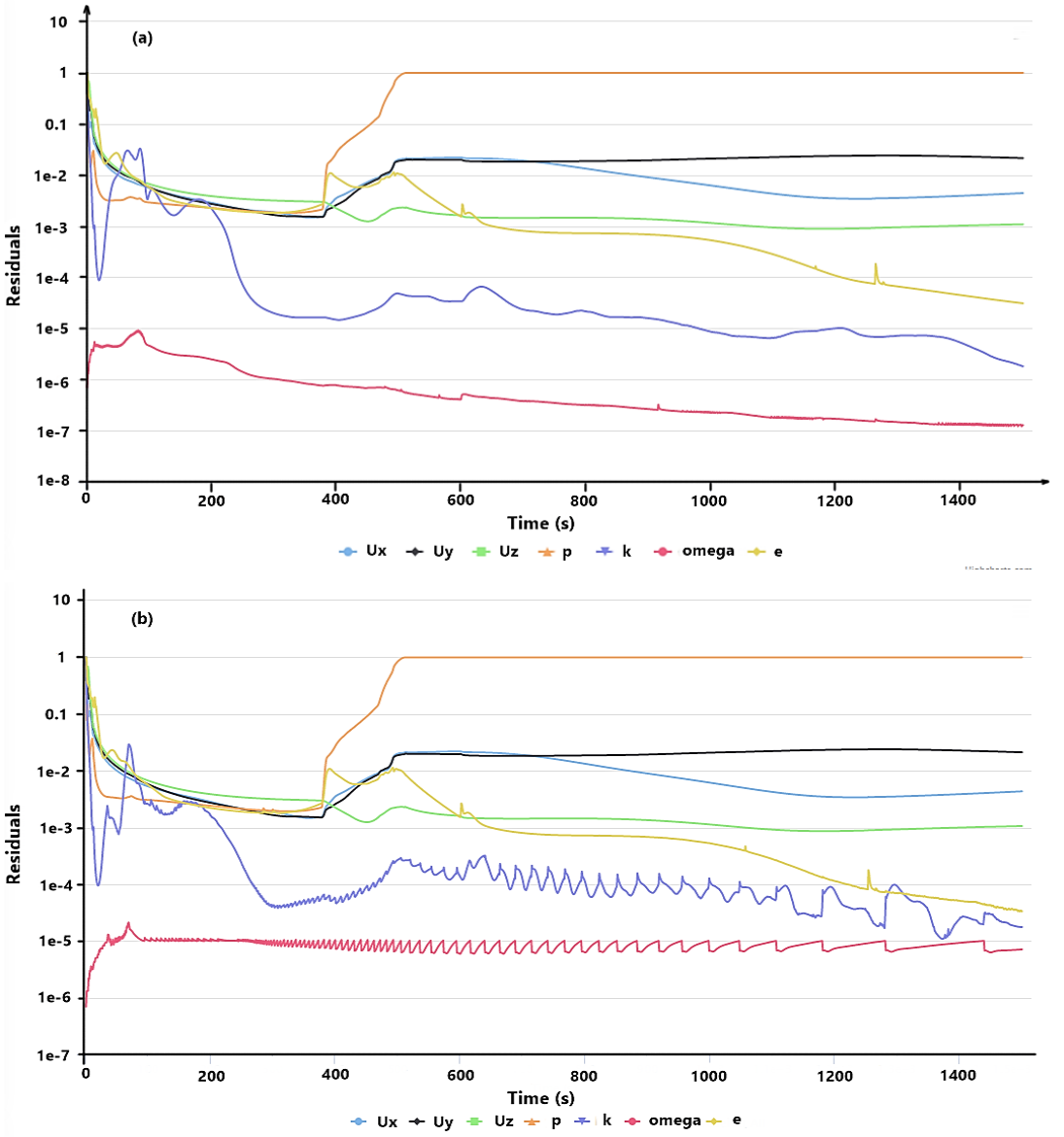


**Fig. 13** Convergence of Scaled residuals (a) for PIMPLE Algorithm. Inference: Solution attained uniform convergence after 600 iterations (attributed to the velocity ramping during the initial run) (b) for PISO Algorithm. Inference: Residuals tend to have undamped oscillations beyond 300 iterations with more non-uniform turbulence model metrics (k and omega). It is to be noted that convergence is not attained with PISO algorithm utilization

noise is the subsequent effect. The wakes created behind a moving aircraft, as well as the engine position, are thus explained.

The presence of a nascent weak shock wave over the mid-span of the aircraft wing is noticed (Fig. 12), which is a testament to the current algorithm implementation in capturing the relevant and critical features of the flow. Coalescing of the thin membrane of the air in the flow adjacent to the shock wave is smeared out to some extent which could be attributed to the still prevalent diffusive effect that the PIMPLE algorithm attempted to damp to a certain extent.

Changes in fluid density, airspeed, and the number of vortices formed in the airflow are all taken into account when calculating the time variation of the wing model's aerodynamic coefficients. The findings are shown in Table 6. The simulation ran for 100 seconds with a 0.1 second step size. Five iterations were performed for each flow time. A comparison of the residual convergence between LES, PISO and PIMPLE shows the uniform and non-oscillatory convergence in PIMPLE in contrast to the PISO algorithm, where the simulation after certain runs tends to have uncontrolled and growing oscillations. Accuracy and computational time being of priority in comparing different algorithms, PIMPLE also proves to be a better choice over LES and PISO due to its computational time being 9.75% better than PISO, in addition to PISO's non-convergence and 128% faster than LES. As proposed earlier, owing to the higher degree of freedom of LES calculations, they are considered baseline values and PIMPLE and PISO results are compared accordingly. In comparison to the classical PISO method, PIMPLE again performs better in terms of computing near accurate results for the case under consideration. Table 6

summarises the obtained Cl and Cd values for different algorithms and the aggregate accuracy in relation to LES results.

**Table 6.** Comparison of aerodynamic coefficients and critical computational metrics for baseline and testing solvers.

| | LES | PIMPLE | PISO |
|---|---|---|---|
| Time | 187 min | 82 min | 90 min |
| Iterations | 2750 | 1500 | 1500 |
| Convergence | Yes | Yes | No |
| CL | 0.514 | 0.478 | 0.422 |
| CD | 0.063 | 0.0645 | 0.091 |
| Accuracy | 100% (baseline) | 94.58% | N/A |

The average error is roughly 6%, indicating that the LES solution is well accepted. Nonetheless, update two increases the number of iterations per time step, owing to the fact that the flux employed in the continuity equation differs from that used in the remainder of the transport equations, necessitating the employment of more PIMPLE loops to achieve convergence.

The plots of drag, lift and moment coefficients against convergence iterations are presented in Fig. 13 and Fig. 14, respectively. These figures provide good information on the convergence effects of critical transonic flow parameters for different solver schemes implemented.

We observe that as the unsteady behaviour of the airflow continues to prevail, the wing losses lift and cause the aircraft to stall earlier than the design condition, which is at a much lower wind speed. This may be due to the energy cascade path, which starts with main flow energy to large eddies all the way down to small eddies energy to smallest eddies and finally to internal energy. FOAM based solver, as a result of this, tends to spike the turbulent kinetic energy converged value for the domain and hence will have a direct influence over the vortex generated and hence the associated unsteady behaviour.

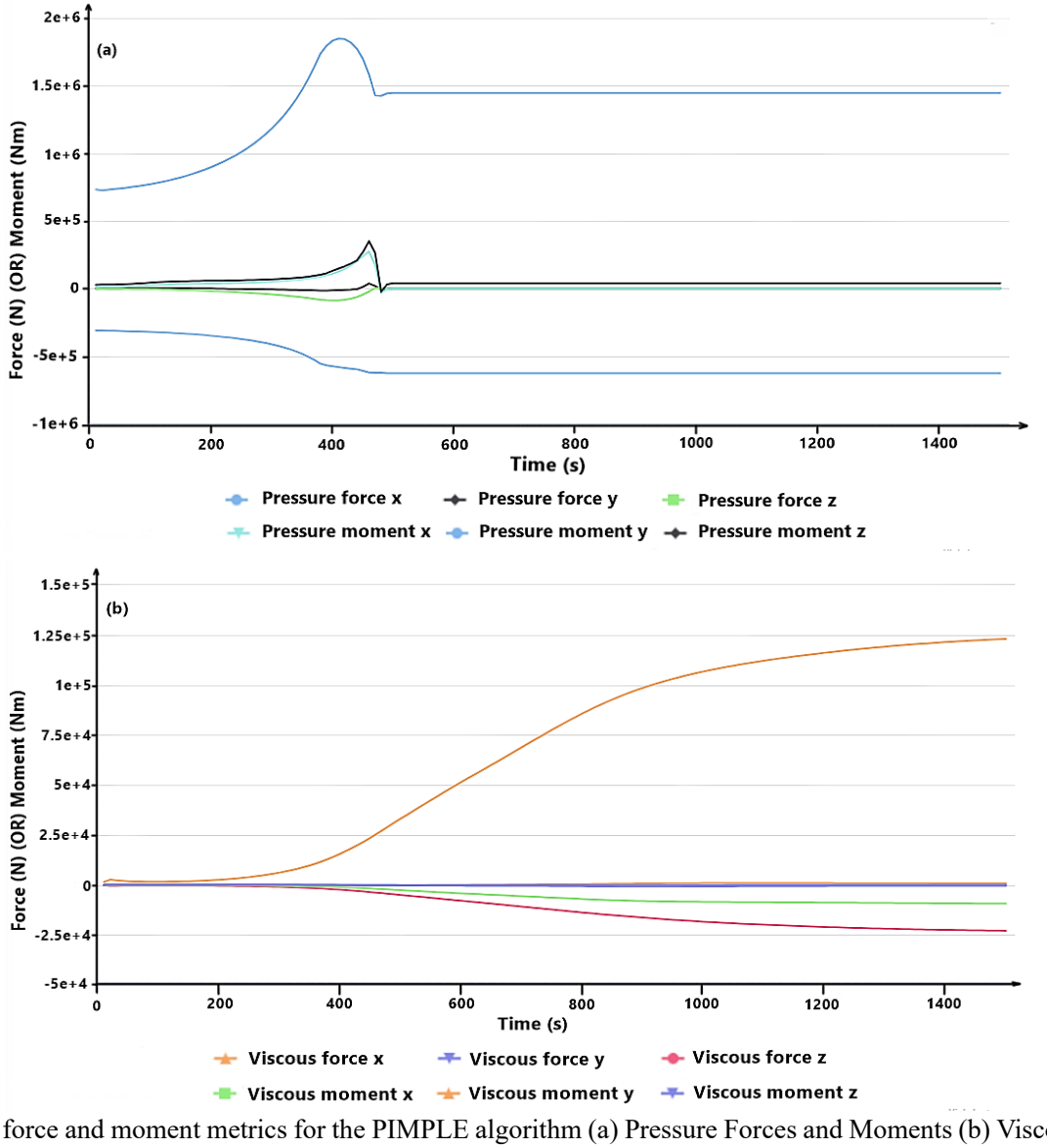


**Fig. 14** Critical force and moment metrics for the PIMPLE algorithm (a) Pressure Forces and Moments (b) Viscous Forces and Moments.

## 4. Conclusions

From the obtained CFD results, a quick juxtaposition of modified and flux stabilized PIMPLE algorithm with classical PISO and LES methods showcased that the modified PIMPLE algorithm, despite using averaged fluxes and high Mach pre-conditioning with suitable corrector terms, showed 9.75% faster convergence than PISO and 128% faster convergence than LES, all with little to no oscillations. For the particular transonic wing flow implementation, PIMPLE had 94% agreement with LES results, which have even better prospects of further improvement by altering the inner and outer corrector parameters. The implemented algorithm had accurate predictions of the location of nascent shock formation in the transonic regime, which gives promising scope for its extension to higher dimensional and high-speed applications. However, it is to be noted that implementation of the PIMPLE algorithm over LES and PISO showed a spike in the predicted turbulent kinetic energy and partial retention of the diffusive properties noticed in other solver counterparts. Utilization of FOAM based solver requires higher quality discretization due to its limitations in adapting for high skewness limits compared to FLUENT. This acts as a bottleneck in the utilization of open-source solvers to incorporate the PIMPLE algorithm. One of the main problems of the operator splitting approach is the poor global conservation of energy during a time step because the energy equation is solved before the pressure equation. Thus, an update of the pressure leads to a change in the mechanical energy and, consequently, changes in all other variables. In many cases, several PIMPLE iterations (about 3) allow for a decrease in the energy imbalance to values less than 1%. But in some cases, it becomes substantial to preserve the energy balance precisely. In those cases, manual control over the inner and outer connecter variables is necessary. Overall, the modified PIMPLE algorithm strikes perfect balance between accuracy and computational time and is an ideal choice of solver algorithm for high-speed compressible flow applications over other higher-order turbulence models, with greater prospects to extend its application to other regimes of the flow.

## 5. Future work

Inferences from the implementation of the PIMPLE algorithm for transonic regime showcases promising capabilities, which could further be explored for hypersonic and supersonic reactive flow cases, where its implementation and accuracy challenge the scientific community to date. The flux stabilization and pre-conditioning modification might be viable options to arrive at less oscillatory results in these cases. In addition, for the same transonic case, our current and future work is devoted to computations using full Reynold's stress closure models. Finally, physics informed data analysis using Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (PMD) could be combined with the capabilities of the PIMPLE solver to scrutinize and make a more tough comparison of unsteady flow computations by standard FOAM solvers for distributed architectures.

## Conflict of Interest

The authors declare no conflict of interest.

## Supporting information

Applicable.

## Reference

[1] J. Kim, P. Moin, R. Moser, Turbulence statistics in fully developed channel flow at low reynolds number, *Journal of Fluid Mechanics*, 1987, **177**, 133-166, doi: 10.1017/S0022112087000892.

[2] S. A. Orszag, G. S. Patterson, Numerical simulation of three-dimensional homogeneous isotropic turbulence, *Physical Review Letters*, 1972, **28**, 76-79, doi: 10.1103/PhysRevLett.28.76.

[3] J. W. Eastwood, Book Review: Springer series in computational physics, *Computer Physics Communications*, 1984, **32**, 439-439, doi: 10.1016/0010-4655(84)90059-6.

[4] S. K. Lele, Compact finite difference schemes with spectral-like resolution, *Journal of Computational Physics*, 1992, **103**, 16-42, doi: 10.1016/0021-9991(92)90324-R.

[5] F. H. Harlow, J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *The Physics of Fluids*, 1965, **8**, 2182-2189, doi: 10.1063/1.1761178.

[6] P. Orlandi, Fluid flow phenomena: a numerical toolkit, *Springer Science & Business Media*, 2000, **55**, 1-356, doi:10.1007/978-94-011-4281-6.

[7] A. Suresh, H. T. Huynh, Accurate monotonicity-preserving schemes with Runge-Kutta time stepping, *Journal of Computational Physics*, 1997, **136**, 83-99, doi: 10.1006/jcph.1997.5745.

[8] A. Jameson, W. Schmidt, E. Turkel, Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes, *14th Fluid and Plasma Dynamics Conference*, 1981, **81**, 1-19, doi: 10.2514/6.1981-1259.

[9] Y. T. Zhang, C. W. Shu, ENO and WENO schemes, *Handbook of Numerical Analysis*, 2016, **17**, 103-122, doi: 10.1016/bs.hna.2016.09.009.

[10] G. Srinivas, R. Kurkal, S. Shenoy, Flow blockage in a transonic axial flow compressor: simulation analysis under distorted conditions, *International Journal of Engineering and Technology*, 2018, **7**, 43-49, doi: 10.14419/ijet.v7i2.21.11833.

[11] V. Potty, S. Angelo, P. Srinivasa Rao, G. Srinivas, Recent developments of an aircraft fuselage along theoretical, experimental and numerical approach-A review, *Universal Journal of Mechanical Engineering*, 2019, **7**, 21-28, doi: 10.13189/ujme.2019.071403.

[12] S. Angelo, V. Potty, P. Srinivasa Rao, G. Srinivas, Aircraft fuselage recent developments-A review, *Universal Journal of Mechanical Engineering*, 2019, **7**, 12-20, doi: 10.13189/ujme.2019.071402.

[13] G. Srinivas, R. Kurkal, S. Shenoy, Recent developments of axial flow compressors under transonic flow conditions, *IOP Conference Series: Materials Science and Engineering*, 2017,

**197**, 012078, doi: 10.1088/1757-899X/197/1/012078.
[14] K. Jiang, Q. Zhang, Y. Kong, C. Xu, X. Ju, X. Du, Numerical study of parameters effect on thermo-flow characteristics of up bubbling fluidized bed particle solar receiver, *ES Energy & Environment*, 2020, **11**, 46-57, doi: 10.30919/esee8c932.
[15] V. Vuorinen, J. P. Keskinen, C. Duwig, B. J. Boersma, On the implementation of low-dissipative Runge-Kutta projection methods for time dependent flows using OpenFOAM®, *Computers & Fluids*, 2014, **93**, 153-163, doi: 10.1016/j.compfluid.2014.01.026.
[16] S. Mao, A. Kan, N. Wang, Numerical analysis and experimental investigation on thermal bridge effect of vacuum insulation panel, *Applied Thermal Engineering*, 2020, **169**, 114980, doi: 10.1016/j.applthermaleng.2020.114980.
[17] H. S. Arunkumar, S. Kumar, K. V. Karanth, A numerical study on the performance of different shaped perforation hole on the absorber duct insert in a solar air heater, *Engineered Science*, 2022, **18**, 1-9, doi: 10.30919/es8d644.
[18] N. Madhwesh K. V. Karanth, Effects of innovative suction slots on the performance of a radial bladed impeller of a centrifugal blower-A numerical transient analysis, *Engineered Science*, 2021, **17**, 91-100, doi: 10.30919/es8d587.
[19] M. Hajzman, J. Vimmr, O. Bublik, On the modelling of compressible inviscid flow problems using AUSM schemes, *Applied and Computational Mechanics*, 2007, **1**, 469-478.
[20] M. S. Liou, A sequel to AUSM, *Journal of Computational Physics*, 1996, **129**, 364-382, doi: 10.1006/jcph.1996.0256.
[21] D. Modesti, S. Pirozzoli, A high-fidelity solver for turbulent compressible flows on unstructured meshes, *Fluid Dynamics*, 2016, 1-17, doi: 10.48550/arXiv.1612.05223.
[22] S. I. Sohn, A new TVD-MUSCL scheme for hyperbolic conservation laws, *Computers & Mathematics with Applications*, 2005, **50**, 231-248, doi: 10.1016/j.camwa.2004.10.047.
[23] S. Pirozzoli, Conservative hybrid compact-WENO schemes for shock-turbulence interaction, *Journal of Computational Physics*, 2002, **178**, 81-117, doi: 10.1006/jcph.2002.7021.
[24] P. Colella, P. R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, *Journal of Computational Physics*, 1984, **54**, 174-201, doi: 10.1016/0021-9991(84)90143-8.
[25] B. Cockburn, C. W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, *Journal of Scientific Computing*, 2001, **16**, 173-261, doi: 10.1023/A:1012873910884.
[26] P. Arminjon, M. C. Viallon, A. Madrane, A Finite volume extension of the lax-friedrichs and nessyahu-tadmor schemes for conservation laws on unstructured grids, *International Journal of Computational Fluid Dynamics*, 1998, **9**, 1-22, doi: 10.1080/10618569808940837.
[27] M. Kraposhin, A. Bovtrikova, S. Strijhak, Adaptation of Kurganov-Tadmor numerical scheme for applying in combination with the PISO method in numerical simulation of flows in a wide range of Mach numbers, *Procedia Computer Science*, 2015, **66,** 43-52, doi: 10.1016/j.procs.2015.11.007.
[28] M. Cerminara, T. E. Ongaro, L. C. Berselli, ASHEE-1.0: a compressible, equilibrium-Eulerian model for volcanic ash plumes, *Geoscientific Model Development*, 2016, **9**, 697-730, doi: 10.5194/gmd-9-697-2016.
[29] M. Kraposhin, M. Banholzer, M. Pfitzner, I. Marchevsky, A hybrid pressure-based solver for non-ideal single-phase fluid flows at all speeds: Non-ideal single-phase fluid flow solver, *International Journal for Numerical Methods in Fluids*, 2018, **88**, 79-99, doi: 10.1002/fld.4512.
[30] A. Vallet, A. A. Burluka, R. Borghi, Development of a Eulerian model for the" Atomization" of a liquid jet, *Atomization and Sprays*, 2001, **11**, 619-642, doi: 10.1615/atomizspr.v11.i6.20.
[31] M. Zhao, D. Wan, Y. Gao, Comparative study of different turbulence models for cavitational flows around NACA0012 hydrofoil, *Journal of Marine Science and Engineering*, 2021, **9**, 742-762, doi: 10.3390/jmse9070742.
[32] R. Verzicco, R. Camussi, Prandtl number effects in convective turbulence, *Journal of Fluid Mechanics*, 1999, **383**, 55-73, doi: 10.1017/S0022112098003619.
[33] M.O. Mendez, M. Matteucci, S. Cerutti, A. M. Bianchi, J. Kortelainen, Automatic detection of sleep macrostructure based on bed sensors, *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, 5555-5558, doi: 10.1109/IEMBS.2009.5333734.
[34] S. Ahmed, 2019, **1**, 1-3, https://www.linkedin.com/pulse/understanding-y-cfd-simulation-sijal-ahmed/.
[35] P. Welahettige, K. Vaagsaether, Comparison of openFOAM and ANSYS fluent, *Proceedings of the 9th EUROSIM congress on modelling and simulation, EUROSIM 2016, The 57th SIMS Conference on Simulation and Modelling SIMS 2016*, 2018, **142**, 1005-1012, doi: 10.3384/ecp171421005.
[36] X. song Li, C. wei Gu, J. zhong Xu, Development of Roe-type scheme for all-speed flows based on preconditioning method, *Computers & Fluids*, 2009, **38**, 810-817, doi: 10.1016/j.compfluid.2008.08.002.

## Author Information

***Aravind Karthik*** *is a senior year aeronautics Honours student pursuing his Bachelors program at Manipal Institute of Technology, India. His primary field of research revolves around developing Novel numerical schemes, Reduced Order Models and AI accelerated solvers for application in Hypersonic Aerothermodynamics, Plasma ionization and combustion flowfield computation. His research inclination is complimented by his diverse experiences with leading labs like Defence research & Development Laboratory, Ministry of Defence, India and Space Transportation and Systems Engineering Laboratory, Kyushu University, Japan. He holds an Indian patent on Plasma actuated Vertical Axis Wind turbine concept. He is a student affiliate in Royal Aeronautical Society, Space and Satellite Professionals International and an active contributor to the DIY community. He is currently*

 

*exploring GPU accelerated Modelling techniques for CFD optimization and Augmented Reality for CFD.*

***Dr. Srinivas G*** *is a faculty in the Department of Aeronautical and Automobile engineering at Manipal Institute of Technology - Manipal Academy of Higher Education (MAHE). He is a doctorate from Manipal Academy of Higher Education -MAHE (Institute of Eminence)- Manipal, India. He is experienced in the field of Aircraft aerodynamics, Aircraft Propulsion, Rocket and Missiles Aerodynamics, Air Transportation system, computational fluid mechanics and has published more than 40+ articles in leading Aerospace and Mechanical Science related journals. He has mentored several UG and Graduate students and is the Faculty Advisor of MIT's collegiate and one of the world's leading student sounding rocketry design and Research club "thrustMIT". He is an institute coordinator and member of prestigious Institution of Power Engineers (IPowerE). His research interest includes Aircraft Propulsion, Rocket Propulsion, Aerodynamics of Rockets and Missiles, Aircraft Design, Computational Fluid Dynamics, Mechanics of Fluids and Aerodynamics, and is currently exploring topics related to Transonic flow and Hypersonic flow regimes applications.*

***Nithesh Naik*** *is a faculty in the Department of Mechanical and Manufacturing Engineering, Manipal Institute of Technology. He has four years of industry experience in the field of planning and design of HVAC systems. He is a master graduate in Design Engineering from prestigious university Manipal Academy of Higher Education (Institute of Eminence). His research interest includes the development of FE analysis of dental sciences, artificial intelligence, composite materials and design and product development techniques. He has applied 4 patents at Indian Patent Office. He has published several research publications in international journals of repute and has keen interests in medical innovations. He received the award of South India's most exciting young teacher. His areas of expertise include FE analysis of dental and medical sciences, composite materials, design and product development, medical devices and innovations, artificial intelligence, medical web and app development.*