

INTERNSHIP REPORT

Thomas Abraham (22JE1018)

CFD Simulation of a Supersonic Cold Jet from a Nine-Nozzle Rocket Cluster

“A Study on Mesh Refinement, Boundary Conditions, and Solver Control for Accurate Jet Characterization”



Certificate

This is to certify that the project work entitled
"CFD Simulation of a Supersonic Cold Jet from a Nine-Nozzle Rocket Cluster"
is a bona fide record of the work carried out by **Thomas Abraham**,
a student of **Bachelor of Technology in Mechanical Engineering, IIT (ISM), Dhanbad**,
during his internship at the **High Altitude Testing Facility (HATF), Satish Dhawan Space
Centre, Sriharikota**,
as a part of the partial fulfilment of academic requirements.

This work was carried out under the supervision of **Mr. K. Kaliprasad**, Deputy Manager,
Solid Motor Performance and Environmental Testing Facilities. The contents of this report are
the original work of the candidate and have not been submitted to any other institution or
university for the award of any degree or diploma.

We hereby certify that the work reported herein meets the standards expected of academic project
work and reflects the intern's sincere effort, technical understanding, and dedication to the subject.



K. Kaliprasad

Dy. Manager,

Solid Motor Performance and

Environmental Test Facilities,

SDSC-SHAR

के वीएसएन कालीप्रसाद / K Kaliprasad
उप प्रबंधक, जीटीएफ & आरडीएफ, टीआ
Dy. Manager, GTF & RDF, TO
एसएमपीवईटीएफ / SMP&ETF
एसडीएससी शार SDSC SHAR

Acknowledgements

I would like to express my sincere gratitude to **Shri A. Rajarajan**, Director, for providing me with the opportunity and environment to undertake this research project.

I am deeply thankful to **Dr. S. Sankaran**, Deputy Director, Vehicle Assembly and Launch Facilities (VALF) and Chief General Manager, Solid Motor Performance and Environmental Testing Facilities, and to **Dr. T. Srinivas Reddy**, General Manager, for their constant support and encouragement.

I gratefully acknowledge the guidance of **Mr. G. Venkatesh** and **Mr. K. Kaliprasad**, Deputy Managers, for their technical advice and support throughout the project.

I am especially thankful to my mentor and guide, **Mr. K. Kaliprasad**, for his insightful guidance, constructive suggestions, and constant encouragement.

I also wish to thank **Ms. M. Deepthi** and **Mr. B. N. V. S. Aditya**, scientists at the High Altitude Testing Facility (HATF), for their patient guidance, valuable inputs, and continued support during the course of this work.

My sincere thanks to **Prof. S. K. Das**, Department of Mechanical Engineering, IIT (ISM) Dhanbad, for supporting my academic journey in CFD, and to **Prof. Satyabrata Sahoo**, Associate Professor, and **Prof. Somnath Chattopadhyaya**, Head of the Department of Mechanical Engineering, for their academic encouragement and support.

Finally, I would like to acknowledge all the scientists and staff at the facility whose support helped make this work possible.

Table of Contents

1. Abstract
2. Computational Fluid Dynamics: The Pre-Requisites.
 - i. An Introduction
 - ii. Numerical Methodology of CFD
 - iii. A Basic Introduction to Turbulence Modelling
 - iv. Why CFD?
3. Analytical Solution
 - i. MATLAB Code
 - ii. Plots
4. Potential Challenges
5. Explaining the Work-Flow
6. Coarse Mesh Trial Runs
 - i. Meshing Strategy
 - ii. Basic Solver Set-Up
 - iii. Boundary Conditions
7. Processing the Converged Case
8. Validation (Comparing Plots from CFD-POST with Plots from MATLAB)
9. Learnings from the Coarse Mesh Run
10. Dealing with the Numerical Oscillations
11. Full Run 1
 - i. Meshing Strategy
 - ii. Solver Set-Up
 - iii. Results
12. An Improved Mesh
13. Conclusion
14. References

Abstract

The nine-nozzle configuration under investigation in this study is a proposed layout for the next generation of ISRO's heavy-lift launch vehicles. While such clustered nozzle arrangements offer significant thrust advantages and modularity, they also introduce complex aerodynamic interactions that pose serious design challenges. These include asymmetric jet behaviour, shock-shock interactions, flow separation, and potential performance losses due to plume interference. Accurate Computational Fluid Dynamics (CFD) simulations are therefore critical for understanding these phenomena in detail, enabling informed design choices that mitigate adverse effects and improve overall engine efficiency. However, setting up a reliable 3D CFD simulation of this configuration is inherently challenging due to the high mesh resolution required to resolve shock structures while maintaining computational economy over the vast domain, the need for precise boundary condition definition, and the computational cost associated with simulating supersonic, multi-jet interactions in three dimensions. This work presents a cold-flow simulation of the clustered nozzle setup and highlights key flow features that influence future design directions.



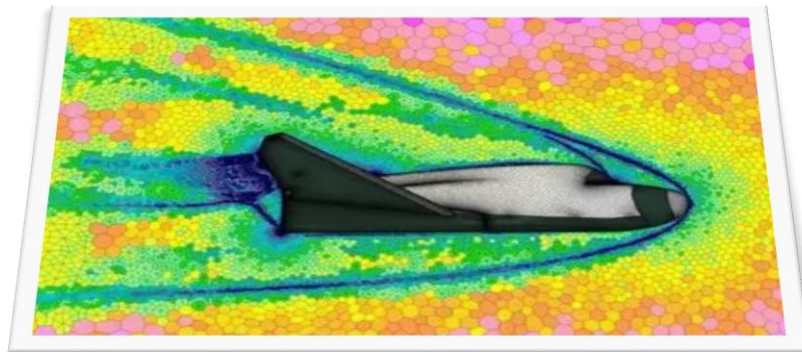
(The Space-X Falcon 9 uses a 9-cluster configuration of the Merlin engines.)

Computational Fluid Dynamics: The Pre-Requisites

An Introduction:

Computational fluid dynamics modelling is a technique built on principles of fluid mechanics in which equations governing fluid motions are applied to provide insights and qualitative predictions of the behaviour of physical systems. Simply, this involves applying numerical methods and algorithm to analyse fluid flow problems. To proceed, the equations governing the fluid flow are replaced by discrete approximations at grid points, that is, the fluid flow is discretized into smaller chunks, small enough that solutions obtained at the grid points are unaffected by the grid point spacing. Techniques such as; finite volume method (FVM), finite element analysis (FEA), or finite difference method (FVM) are then applied for further processing to derive discrete equations which links all the grid points together, thereby providing relevant information needed to describe the model.

The aerospace industry heavily relies on Computational Fluid Dynamics (CFD) as a powerful tool for analysing and optimizing fluid flow around complex geometries, such as aircraft bodies, rocket nozzles, and propulsion systems. CFD enables engineers to simulate real-world aerodynamic conditions virtually, reducing the need for costly wind tunnel tests and full-scale prototypes in early design stages. It helps predict key performance parameters like lift, drag, thrust, pressure distribution, and thermal loads with high fidelity. By providing detailed insight into flow behaviour, CFD accelerates design cycles, enhances safety, and supports innovation in developing more efficient and reliable aerospace systems.



Numerical Methodology

The governing equations relating to the transport of the mass of the fluid and the heat in it are strongly coupled, non-linear partial differential equations. Except for cases where simplifying assumptions are made, (like reducing the space complexity to 1 dimension), an analytical solution is impossible.

$$\partial(\rho \cdot \phi)/\partial t + \nabla \cdot (\rho \cdot u \cdot \phi) - \nabla \cdot (\Gamma(\nabla \phi)) = S$$

The above equation is the basic form of the transport equation of an intensive quantity ' ϕ '. On multiplication with ' ρ ', it becomes a volume specific property. Thus, on integrating over a control volume, it becomes its conservation equation over that control volume.

The first term is the unsteady term, variation of the quantity ' ϕ ' with time. The second term represents the transport of ' ϕ ' due the convection velocity ' u '. The third term represents the transport of ' ϕ ' by a gradient based diffusion method (' Γ ') being the diffusion coefficient. The term on the RHS is the "Source/Sink" term that accounts for the addition/subtraction of ' ϕ ' from the control volume.

Most governing equations more or less can take this form. This includes the transport equation for momentum and energy. The "Navier Stokes" equations combine all many individual transport equations and can thus describe the viscous, compressible, transient and thermal effects in any flow.

The convective and diffusive term induces (second and third terms on the LHS) are responsible for inducing non-linearity. Thus, an alternate way involving numerical methods is a must for being able to solve the governing equations of fluid flows for practical engineering applications.

A "numerical solution" of a differential equation consists of a set of numbers from which the distribution of the dependent variable ' ϕ ' can be constructed. In this sense, a numerical method is akin to a laboratory experiment, in which a set of instrument readings enables us to establish the distribution of the measured quantity in the domain under investigation. The numerical analyst and the laboratory experimenter both must remain content with only a *finite* number of numerical values as the outcome, although this number can, at least in principle, be made large enough for practical purposes.

Let us suppose that we decide to represent the variation of ϕ by a polynomial in ' x ',

$$\phi(x) = a_0 + a_1 * x + a_2 * x^2 \dots a_m x^m$$

and employ a numerical method to find the finite number of coefficients a_0, a_1 , etc.. This will enable us to evaluate ' ϕ ' at any location x by substituting the value of ' x ' and the values of the a 's into the above equation. This procedure is, however, somewhat inconvenient if our ultimate interest is to obtain the values of ' ϕ ' at various locations. The values of the a 's are, by themselves, not particularly meaningful, and the substitution operation must be carried out to arrive at the required values of ' ϕ '. This leads us to the following thought: Why not construct a method that employs the values of ' ϕ ' at a number of given points as the primary unknowns

Thus, a numerical method treats as its basic unknowns the values of the dependent variable at a finite number of locations (called the *grid points*) in the calculation domain. The method includes the tasks of providing a set of algebraic equations for these unknowns and of prescribing an algorithm for solving the equations.

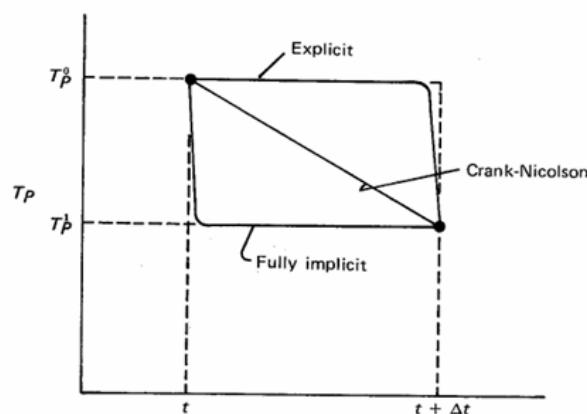
Therefore, the core of the discretisation schemes like FVM, FDM and FEM involve:

converting a continuous partial differential equation into a system of algebraic equations defined at discrete points in the domain, which can then be solved using numerical linear algebra techniques.

The two most commonly used discretisation schemes in the numerical analysis of fluid flows are the Finite Difference Method (FDM) and the Finite Volume Method (FVM).

In FDM, the governing partial differential equations are discretised by approximating derivatives using Taylor series expansions. The accuracy of the method depends heavily on the truncation error introduced by these approximations. For example, the central differencing scheme has a second-order truncation error, making it more accurate than the forward and backward differencing schemes, which are first-order accurate.

Temporal derivatives in FDM can be discretised using explicit, implicit, or Crank–Nicolson schemes, allowing control over stability and accuracy in time-dependent problems.



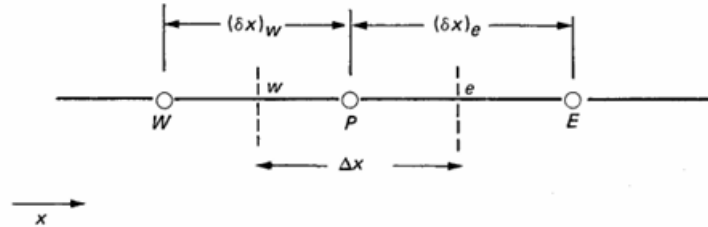
The Finite Volume Method (FVM), on the other hand, is a special case of the Weighted Residual Method — a family of methods where the governing equations are approximated in a way that minimizes the residual (error) when weighted by specific functions. In FVM, this concept is applied by integrating the conservation equations over small control volumes, ensuring that the net flux of each conserved quantity (mass, momentum, energy) is balanced within each cell.

FVM is especially robust for fluid flow problems, as it guarantees local and global conservation of physical quantities. It is well-suited to complex geometries and compressible flows, which is why it is the most widely used approach in modern CFD solvers.

A basic illustrative example:

Consider a steady 1-dimensional heat conduction governed by:

$$\frac{d}{dx} \left(K \frac{dT}{dx} \right) + S = 0$$

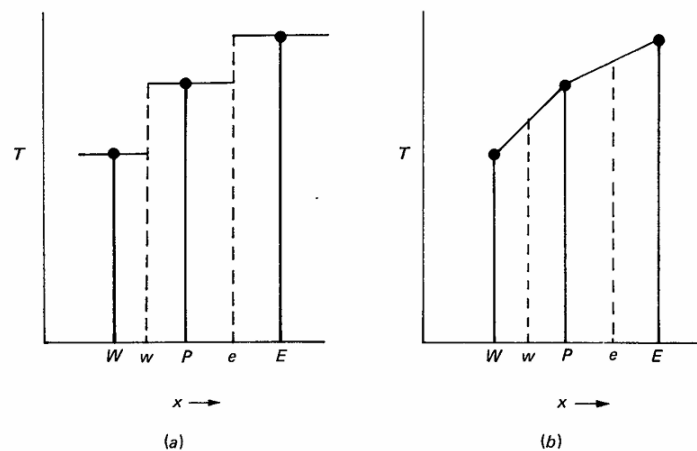


Here, 'K' is the thermal conductivity, 'T' is the temperature and 'S' is the source/sink term.

To derive the discretisation equation, we shall employ the grid-point cluster shown in the above figure. We focus attention on the grid point P, which has the grid points 'E' and 'W' as its neighbours. (E denotes the east side, i.e., the positive x direction, while W stands for west or the negative x direction.) The dashed lines show the faces of the control volume; their exact locations are unimportant for the time being. The letters 'e' and 'w' denote these faces. For the one-dimensional problem under consideration, we shall assume a unit thickness in the y and z directions. Thus, the volume of the control volume shown is ' $\Delta x \times 1 \times 1$ '. If we integrate the above equation over the control volume, we get

$$k \left(\frac{dT}{dx} \right)_e - k \left(\frac{dT}{dx} \right)_w + \int_w^e S \cdot dx$$

To make further progress, we need a profile assumption or an interpolation formula. Two simple profile assumptions are shown in the figure below. The simplest possibility is to assume that the value of 'T' at a grid point prevails over the control volume surrounding it. This gives the stepwise profile sketched in Fig. 3.3a. For this profile, the slope ' dT/dx ' is not defined at the control-volume faces (i.e., at 'w' or 'e'). A profile that does not suffer from this difficulty is the piecewise-linear profile. Here, linear interpolation functions are used between the grid points.



((a) Stepwise Profile, (b) Linear Profile)

If we evaluate the derivatives 'dT/dx' from the piecewise-linear profile, the resulting equation will be:

$$\frac{k_e(T_E - T_P)}{\delta x_e} - \frac{k_w(T_P - T_W)}{\delta x_w} + \bar{S}\Delta x$$

Casting the above equation into the form:

$$a_P * T_P = a_E * T_E + a_W * T_W + b$$

Where,

$$a_E = \frac{k_e}{\delta x_e}$$

$$a_W = \frac{k_w}{\delta x_w}$$

$$a_P = a_E + a_W$$

$$b = \bar{S}\Delta x$$

(\bar{S} is the average of the source term over the control volume)

Now, four basic rules must be satisfied:

Rule 1: Consistency at control-volume faces

When a face is common to two adjacent control volumes, the flux across it must be represented by the same expression in the discretisation equations for the two control volumes.

Rule 2: The positive coefficients rule

In discretised equations for convection and diffusion, the value at a grid point should change in the same direction as changes at neighbouring points, under the influence of convection and diffusion alone. To ensure this physically consistent behaviour and avoid oscillations, the coefficients in the discretised equations (central and neighbouring coefficients) must all have the same sign. For stability and simplicity, they are usually chosen to be positive.

Rule 3: Negative slope linearisation of the source term

Often, the source term is a function of the dependent variable 'T'

'T' itself, and it is then desirable to acknowledge this dependence in constructing the discretization equation. We can, however, formally account for only a linear dependence because, as often, the discretisation equations will be solved by the techniques for linear algebraic equations.

The source term is usually linearised in this fashion:

$$\bar{S} = S_c + S_p * T_p$$

Here, ' S_c ' stands for the constant part of the source term while ' S_p ' is the corresponding coefficient at for ' T_p '.

The term ' S_p ' becomes part of ' a_p ' when casting the discretised equation into the general form:

$$a_p * T_p = a_e * T_e + a_w * T_w + b$$

Where,

$$a_p = (a'_p - S_p)$$

(Here, ' a'_p ' denotes the terms other than ' S_p ' that form a part of ' a_p ')

If the magnitude of ' S_p ' is large enough, it can turn ' a_p ' negative, thus violating the positive coefficients rule.

This rule is not as arbitrary as it sounds. Oftentimes, the source term has a negative sloped relationship with the dependant variable. For example, in the conduction equation for heat (Fourier's Heat Conduction Law):

$$\dot{Q} = -K\nabla T$$

('K' is the heat conduction coefficient, $K > 0$)

Rule 4: Sum of the neighbour coefficients

Often, the governing equations consists of only the derivatives of the dependant variable. For example, both ' T ' and ' $T+c$ ' would satisfy the governing equations for heat transfer. This should also be reflected likewise in the discretised equations. This would imply:

$$a_p = \sum a_n$$

However, when discretising a heat transfer equation where the source term is function of temperature (that is, ' $S_p \neq 0$ '), this rule is violated. Even in such cases, this rule should not be forgotten, but should be applied envisaging a special case of the equation.

A more intuitive way of looking at Rule 4: In the absence of a source term, when all the neighbouring temperatures ' T_n ' are equal, the centre temperature ' T_p ' must be equal to them.

Only a poorly discretised equation would not predict ' $T_p = T_n$ ' under such circumstances.

The above section outlined the important practice of numerical discretisation which forms the bedrock of CFD methods. The CFD evaluation of the free jet is done using Ansys FLUENT, an industry standard in terms of CFD software. Along with OpenFOAM and many other reliable solvers of industrial importance, it uses a numerical method based on the FVM approach.

The above outline hopefully imparts a basic idea of the numerical methodology of CFD.

A Basic Introduction to Turbulence Modelling

Turbulence refers to irregular, chaotic fluctuations in fluid flow, which significantly affect the transport of momentum, energy, and scalar quantities. In Computational Fluid Dynamics (CFD), solving all turbulent scales directly (as done in Direct Numerical Simulation or DNS) is computationally very expensive. Therefore, most practical simulations use the Reynolds-Averaged Navier–Stokes (RANS) equations, which require turbulence models to approximate the effects of turbulence on the mean flow.

Important Turbulence Quantities:

- k : Turbulent kinetic energy – represents the intensity of turbulence (energy contained in eddies).
- ϵ (epsilon): Turbulent dissipation rate – the rate at which turbulent energy is converted into heat.
- ω (omega): Specific dissipation rate – defined as ϵ/k , and represents the frequency of turbulence.

Several turbulence models are used in engineering applications. Each model balances computational cost with physical fidelity and is suited to different types of flows. The most commonly used models are described below:

1. k – ϵ Model

- Solves two equations: one for turbulent kinetic energy (k) and one for its dissipation rate (ϵ).
- Widely used in industrial flows, especially where the flow is fully developed.
- Performs well in free shear flows like jets and wakes.
- Less accurate near walls and in cases with strong flow separation.

2. k – ω Model

- Uses k and ω as the variables.
- Provides better accuracy near walls without requiring wall functions.
- More sensitive to free-stream boundary conditions (especially ω).
- Better suited for flows with pressure gradients and separation.

3. k – ω SST (Shear Stress Transport) Model

- A hybrid model that combines the advantages of k – ϵ and k – ω models.
- Uses k – ω near the wall for better accuracy, and k – ϵ in the free stream to reduce sensitivity.
- Very popular in simulations involving boundary layer separation, adverse pressure gradients, and complex flows.
- One of the most reliable general-purpose turbulence models.

4. Spalart–Allmaras Model

- A single-equation model that solves for a modified turbulent viscosity.
- Developed specifically for aerospace applications.
- Performs well for external flows over aerofoils, wings, and fuselages.
- Computationally efficient and accurate for attached flows.
- Less effective for complex flows involving large separations or recirculation.

In summary, turbulence modelling is an essential part of CFD that enables practical simulation of complex flows. The $k-\omega$ SST and Spalart–Allmaras models are particularly popular in aerospace and engineering applications due to their accuracy and efficiency.

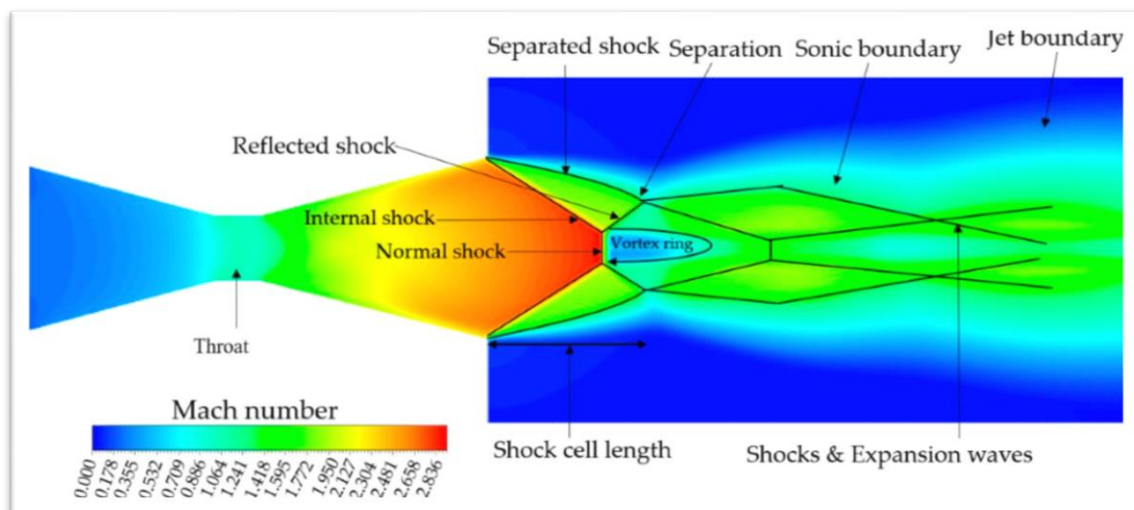


Why CFD?

In this project, Computational Fluid Dynamics (CFD) is employed to simulate the high-speed free jet emerging from a clustered rocket nozzle configuration. The complex interaction of multiple supersonic exhaust plumes in close proximity introduces nonlinear flow phenomena such as shock-shock interactions, shear layer instabilities, and asymmetrical plume behaviour. Capturing these effects accurately is crucial for understanding the performance, structural loading, and thermal behaviour of the launch vehicle's propulsion system.

A physical experiment or hot-fire test of such configurations is expensive and limited in terms of detailed internal flow field visualization. CFD provides a cost-effective and high-resolution alternative that enables the detailed study of flow parameters like Mach number, pressure distribution, and jet symmetry. By numerically solving the compressible Navier–Stokes equations using a density-based solver, the simulation can capture the propagation of shocks and expansion waves with high fidelity. This predictive capability is particularly valuable during the design and validation stages of heavy-lift launch vehicles.

Furthermore, the ability to model the system in three dimensions and under steady-state conditions allows engineers to assess the flow behaviour in a realistic yet computationally manageable way, supporting informed design decisions and performance optimization.



Analytical Solution

To validate the numerical (CFD) results, a simplified quasi one-dimensional (quasi-1D) analytical model is developed for the nozzle and free jet. This analytical approach is based on isentropic flow relations and assumes a steady, inviscid, compressible flow of an ideal gas through an axisymmetric converging-diverging nozzle. It further assumes that the flow is isentropic (i.e., adiabatic and reversible) with no shock waves, no viscous effects, and uniform flow properties at each cross-section. The solution also neglects any multi-dimensional or turbulent effects, making it purely geometry-driven. Also, the geometry of the nozzle is simplified to a simple converging and diverging section, neglecting the geometric complexities like the fillets.

Such an analytical solution is essential as a baseline for validating the CFD results. While it cannot capture complex flow phenomena like boundary layer growth, shock interactions, or jet mixing, it helps establish whether the CFD setup reproduces the expected trends and magnitudes in Mach number, pressure, and temperature distributions under ideal conditions. A close match with this analytical model indicates that the CFD simulation is physically consistent and correctly implemented.

Isentropic relations used in the modelling:

$$\frac{A}{A^*} = \frac{1}{M} \left(\frac{1 + \frac{\gamma-1}{2} M^2}{\frac{\gamma+1}{2}} \right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

$$\frac{T}{T_o} = \frac{(\gamma - 1) * M^2}{2} + 1$$

$$\frac{P}{P_o} = \left(\frac{T}{T_o} \right)^{\frac{\gamma}{\gamma-1}} = \left(\frac{(\gamma - 1) * M^2}{2} + 1 \right)^{\frac{\gamma}{\gamma-1}}$$

(The thermodynamic symbology retains its usual meaning)

In the free jet region downstream of the nozzle exit, the centreline Mach number is known to decrease due to successive expansion and compression waves. A commonly used empirical approximation for the decay of Mach number in a supersonic jet is:

$$M(x) = M_e * \left(1 - \frac{x - x_e}{k * D_e} \right)$$

(Here, the subscript 'e' denotes the values at the exit plane. For example, ' x_e ' denotes the 'x' coordinate of the exit plane.)

In the empirical model used to approximate the Mach number variation in the free jet, the parameter k controls how quickly the jet slows down as it mixes with the surrounding atmosphere. The value of 'k' is usually chosen between '10' and '20' for such jets. It depends on factors like the nozzle exit Mach number, nozzle size, and ambient conditions.

In this model, ' $k = 13$ ' is chosen because it gives a realistic length for the supersonic core of the jet. For most high-speed jets, the core extends about '10' to '15' nozzle diameters downstream before the jet starts to mix and slow down. This value is commonly observed in experiments and works well for cold gas jets in open air.

MATLAB code:

```

close all;
clear;
clc;

%Modelling the nozzle:
r_1=0.0045; r_t=0.00205; r_2=0.006;
l_con=0.00327;l_div=0.015;
A_t=pi*r_t^2; gamma=1.4;
%Defining the domain
l_domain=0.950;
x_con=linspace(0,l_con,100);
x_div=linspace(l_con,l_div+l_con,500);
x_jet=linspace(l_div+l_con,l_domain,5000);
%Converging section modelling
slope_con=(r_t-r_1)/l_con;
R_con=@(x) slope_con*x+r_1;
A_con=@(x) pi*(R_con(x))^2;
%Divergining Section modelling
slope_div=(r_2-r_t)/l_div;
R_div=@(x) slope_div*(x-l_con)+r_t;
A_div=@(x) pi*(R_div(x))^2;

%Plotting the nozzle profile:
figure()
x_nozzle=[x_con,x_div];
r_nozzle=[R_con(x_con),R_div(x_div)];
plot(x_nozzle,r_nozzle)
xlabel('Length along the nozzle (in m)');
ylabel('Radial distance from centre line in (m)');
title('Nozzle Profile');

Mach_rel = @(M, A) (1/M)*((2/(gamma+1))*(1+(gamma-1)/2*M^2))^( (gamma+1)/(2*(gamma-1))) -
(A/A_t);

Mach_solve=@(M_guess,A) fzero(@(M)Mach_rel(M,A),M_guess);

M_con=zeros(size(x_con));
M_div=zeros(size(x_div));
M_guess=[0.2, 2.5];
for i = 1:length(x_con)
    if abs(A_con(x_con(i)) - A_t) < 1e-8
        M_con(i) = 1;
    end
end

```

```
else
M_con(i) = Mach_solve(0.7, A_con(x_con(i)));
    end
end
for i = 1:length(x_div)
    if abs(A_div(x_div(i)) - A_t) < 1e-8
M_div(i) = 1;
    else
M_div(i) = Mach_solve(2.5, A_div(x_div(i)));
    end
end
M_nozzle=[M_con,M_div];
M_exit=M_nozzle(end);
%Plotting the nozzle profile:
figure()
x_nozzle=[x_con,x_div];
plot(x_nozzle,M_nozzle)
xlabel('Length along the nozzle (in m)');
ylabel('Mach Number');
title('Mach Number Profile Along the Nozzle');

%Pressure and Temperature Calculations:
T_o=300; P_o=6700000;
T = @(M) T_o/(1+(gamma-1)/2*M^2);
P = @(M) P_o/(1+(gamma-1)/2*M^2)^(gamma/(gamma-1));
P_nozzle=arrayfun(@(M) P(M),M_nozzle);
T_nozzle=arrayfun(@(M) T(M),M_nozzle);

figure()
plot(x_nozzle,P_nozzle)
xlabel('Length along the nozzle (in m)');
ylabel('Pressure (in Pascals)');
title('Pressure Along the Nozzle');
figure()
plot(x_nozzle,T_nozzle)
xlabel('Length along the nozzle (in m)');
ylabel('Temperature (in K)');
title('Temperature Along the Nozzle');
```

```
%Modelling the free jet:
x_domain=[x_nozzle,x_jet];
k=12;
M_jet_fun = @(x) max(0, M_exit*(1-(x-(l_con+l_div))/(k*2*r_2)));
M_jet=arrayfun(@(x)M_jet_fun(x),x_jet);
M_domain=[M_nozzle,M_jet];

figure()
plot(x_domain,M_domain)
xlabel('Length in the streamwise direction (in m)');
ylabel('Mach Nubmer');
title('Mach Number across the domain');

P_jet=arrayfun(@(M)P(M), M_jet);
T_jet=arrayfun(@(M)T(M), M_jet);
P_domain=[P_nozzle,P_jet];
T_domain=[T_nozzle,T_jet];

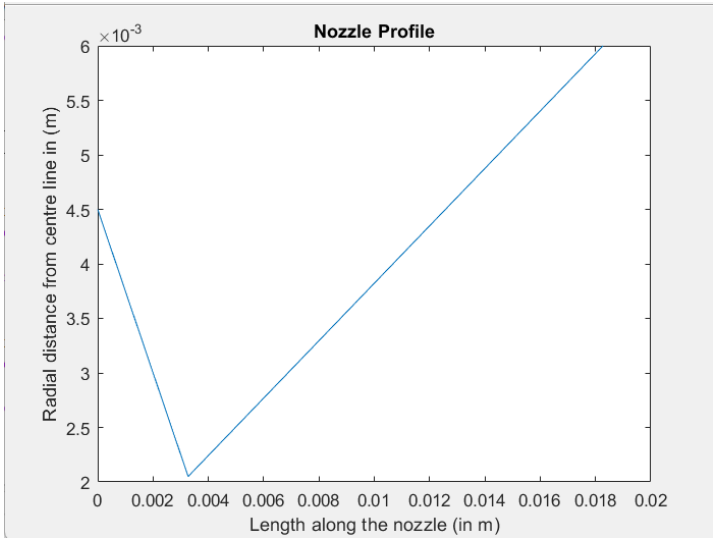
figure()
plot(x_domain,P_domain)
xlabel('Length in the streamwise direction (in m)');
ylabel('Pressure (in Pascals)');
title('Pressure across the domain');

figure()
plot(x_domain,T_domain)
xlabel('Length in the streamwise direction (in m)');
ylabel('Temperature (in K)');
title('Temperature across the domain');

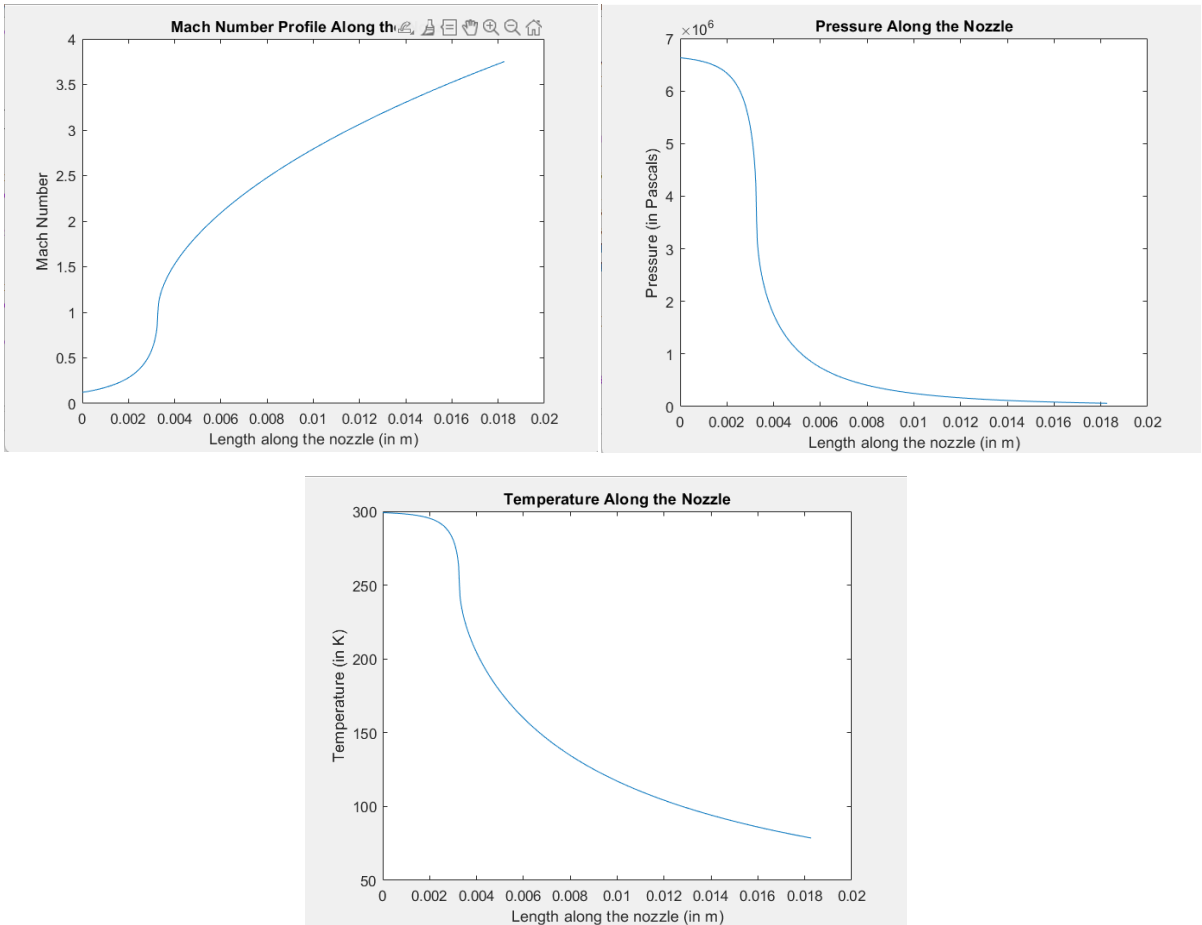
%Calculating the mass flow rate:
R=287; %for air as the working fluid
No_of_Nozzles=9;
m_fr_chocked=A_t*P_o*((gamma/(R*T_o))^(1/2))*(2/(gamma+1))^(gamma/(2*(gamma-1)));
m_fr_chocked_total=No_of_Nozzles*m_fr_chocked;
```

Plots

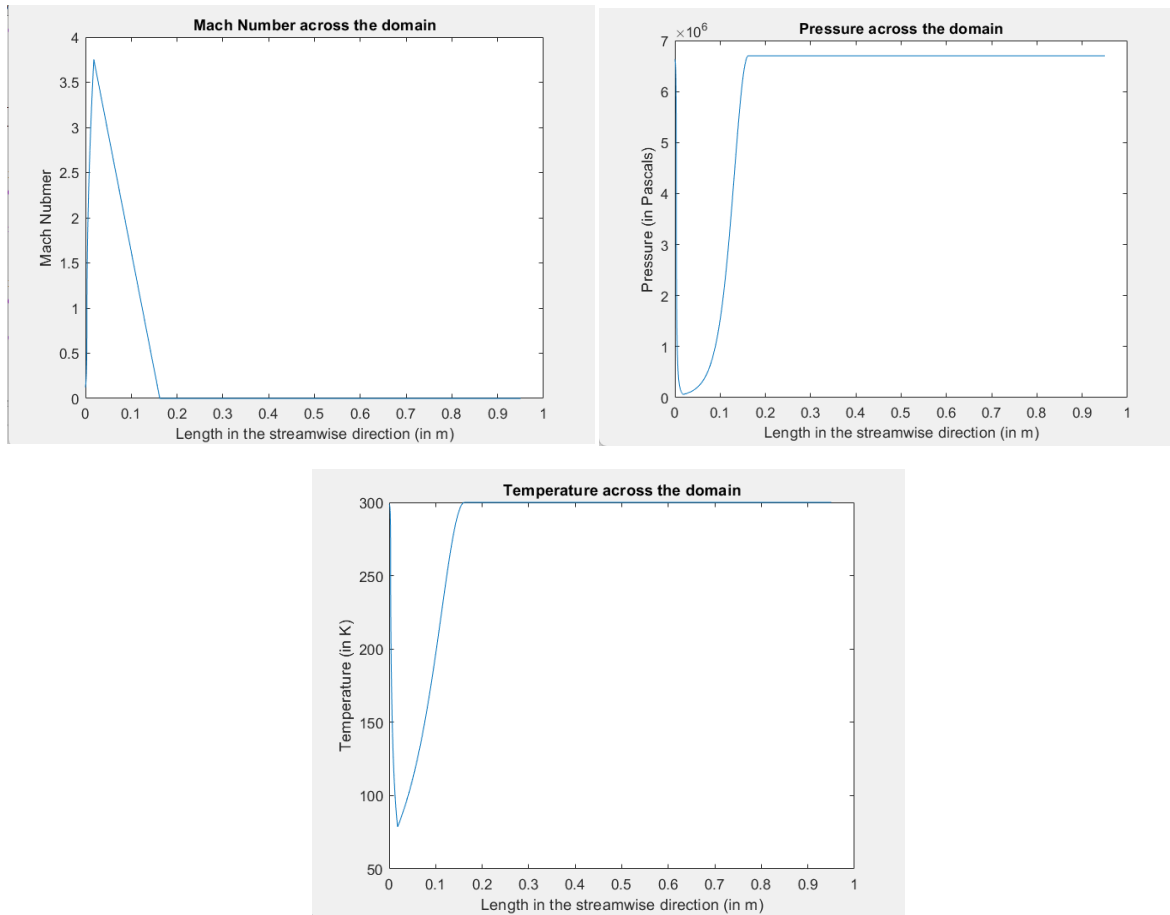
Nozzle Profile:



Variation along nozzle:



Variation from Nozzle to 950mm downstream the inlet plane:



On exiting the nozzle, the jet mixes with the ambient air outside in a chaotic, turbulent and an irreversible manner. Shock waves, shear layers and other such phenomena makes the process outside the nozzle highly isentropic.

However, the temperature plot profile is close to physical reality and is a good way to validate the numerical solution. This is because the temperature plot is less susceptible to these entropy changes. The equivalence of total temperature across any domain is usually analytically shown by the steady flow energy equation. Since the flow in the centreline is fairly inviscid and is well isolated from the ambient air by strong shear layers, the total temperature can be assumed to be the nearly equal and the modelling of the temperature profile can be done using the below equation:

$$\frac{T}{T_o} = \frac{(\gamma - 1) * M^2}{2} + 1$$

This is directly derived from the steady flow energy equation. There is no shaft work and along the centreline, the assumption of no heat addition/removal is reasonable.

The familiar equation that would be used to model the pressure variation with Mach number is derived from the above equation.

$$\frac{P}{P_o} = \left(\frac{T}{T_o}\right)^{\frac{\gamma}{\gamma-1}} = \left(\frac{(\gamma - 1) * M^2}{2} + 1\right)^{\frac{\gamma}{\gamma-1}}$$

However, this is heavily reliant on the flow being isentropic. Also, the total pressure decreases exponentially with respect to entropy change. Thus, the pressure profile cannot be modelled accurately without using complex empirical/mathematical models. *Thus, the “Pressure across the domain” plot should not be considered in validating the CFD solution.*

Despite the many convenient assumptions, the above plots can play an important role in validating the numerical solution. On comparing the basic shape of the profile, it can be ascertained that the CFD solution is consistent with the basic analytical prediction.

Mass Flow Rate Outputs:

```
m_fr_chocked =  
0.2064
```

```
m_fr_chocked_total =  
1.8578
```

Potential Challenges

This section deals with the potential challenges in setting up the numerical solution. To study the free jet, the nozzles' CAD geometry will have to be affixed with an additional body for the fluid to empty into. This additional body must simulate the effects of the ambient atmosphere. The challenge here is to model this additional domain in manner where it is sizeable enough to capture the most drastic effects seen by the developing jet while being compact enough to save computational effort. Computational economy is one of the key deciding metrics of a good CFD set-up.

Another challenge is that the mesh sizing requirements vary drastically along the domain. The region around the nozzles and the expected region of the jet needs a much finer resolution than the far-field regions of the domain. The body sizing must be done in such a manner that the change in the size of the mesh elements must not be too drastic over space. This can lead to the creation of a batch of 'bad quality' mesh elements, characterised by low orthogonality, high aspect ratios and skewness. This can hinder the solution- cause it to unphysical results or even cause the solution to "diverge" (the residuals balloon to unreasonably high values, indicating that the solution is not converging to a stable solution). The key is to avoid these bad elements by not giving a drastic difference in mesh sizing along the different parts of the domain (regions of interest and far field region) or by giving a taking a more "stepped approach" where the mesh elements near the free jet increases slowly while the changes are more drastic away from the region of interest) in an effort to keep any bad elements created away from the region of interest.

Other difficulties would involve setting up the right boundary conditions so that the problem is "well-posed", that is, the solver has all the necessary information at the boundaries to initialise and solve for the pressure and velocity fields along with the density variation in the domain. Also the domain coupled with the right combination of boundary conditions yield a physically realistic solution.

Explaining the Work-Flow

Firstly, a coarse mesh is used with relaxed convergence criteria. This is purely for the sake of testing for solver stability with different combinations and then validating the results with the analytical calculations.

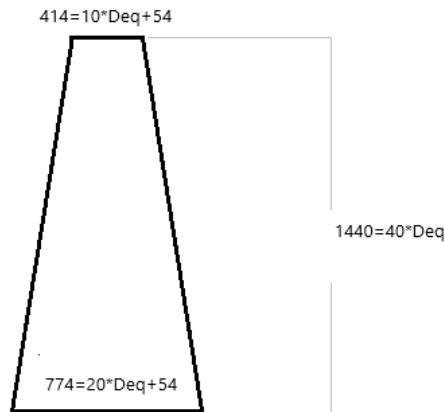
Then a finer mesh would be run with the solver set-up that works the best, hoping to capture all the different jet characteristics like the plume interaction, the shock-diamonds, etc. with a reasonable resolution. Here, a more scrutinizing convergence criteria will be used. If the solver is showing any numerical instability like the residuals oscillating, different solver setting will have to be tried to coax the residuals below the set thresholds. The goal is to bring the residuals for the energy and the turbulence equations to below ' $1e-6$ ' while the rest are kept at a relatively relaxed ' $1e-3$ '. If the jet details are not being captured in detail, a finer mesh will have to be tried.

This work-flow boils down to a decision loop that is more or less based on educated trial and error. The more experienced a CFD analyst is, the smaller the number of trial set-ups are needed before arriving at a converged, physically realistic solution that is also computationally efficient.

Coarse Mesh Trial Runs

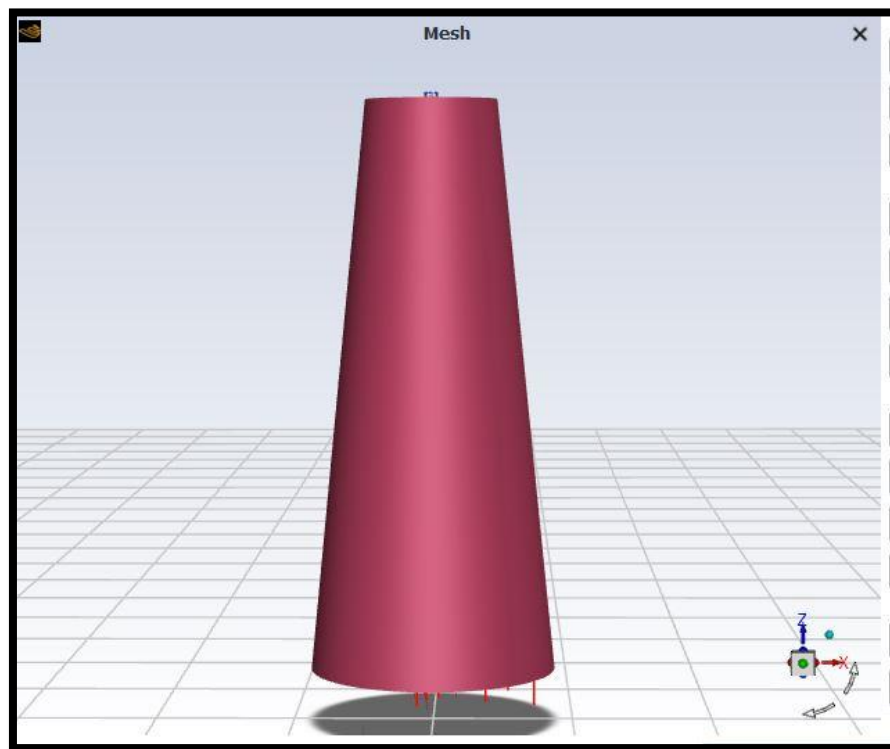
For reasons pertaining to confidentiality, the nozzle profile is not shown.

The schematic of the affixed domain:



(All dimensions are in mm.)

All the dimensions are in terms of the equivalent diameter of the cluster. The nozzle's exit diameter is 12mm. And as there are nine nozzles, a nozzle with an equal exit area would have an exit diameter of 36mm.



The domain housing the free jet is in the shape of a frustum of a cone. The choice of shape for the domain is purely for the sake of computational economy.

Meshing Strategy

For the initial runs, the body of influences have been suppressed for physics. The entire domain (nozzle geometry and the frustum) was meshed with the “Automatic Sizing” method.

The “Capture Curvature” was set to “Yes”. This feature generates a mesh where the cells are sized and placed so as to effectively capture the curvature in the domain.

The effectiveness of this is best measured by the “Curvature Normal Angle” parameter. This is the angle between the normals of the adjacent faces. This was left at the default values of ‘18’ degrees (default).

The “Capture Proximity” was set to “Yes”. This feature generates a mesh where the cells are properly sized in accordance with the proximity between the different faces of the geometry.

The effectiveness of this method is best measure by the “Proximity Gap Factor”. This was left at the default value at ‘3’.

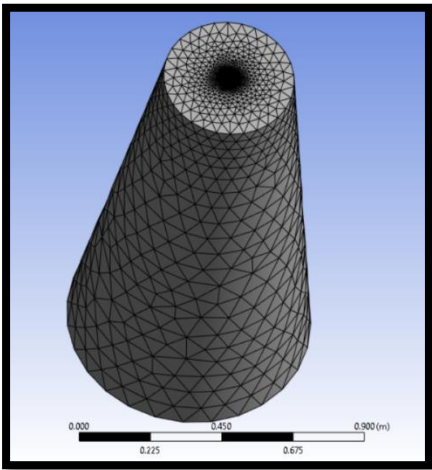
The setting here should generate a coarse mesh that adapts well to the complexities in the nozzle geometry as well as the vastness of the frustum shaped domain.

Use Adaptive Sizing	No
<input type="checkbox"/> Growth Rate	Default (1.2)
<input type="checkbox"/> Max Size	Default (0.18234 m)
Mesh Defeaturing	Yes
<input type="checkbox"/> Defeature Size	Default (4.5584e-004 m)
Capture Curvature	Yes
<input type="checkbox"/> Curvature Min Size	Default (9.1169e-004 m)
<input type="checkbox"/> Curvature Normal Angle	Default (18.0°)
Capture Proximity	Yes
<input type="checkbox"/> Proximity Min Size	Default (9.1169e-004 m)
<input type="checkbox"/> Proximity Gap Factor	Default (3.0)
Proximity Size Sources	Faces and Edges
Bounding Box Diagonal	1.8234 m
Average Surface Area	8.4988e-002 m²
Minimum Edge Length	1.3326e-002 m

(The Mesh Sizing Details)

Defaults	
Physics Preference	CFD
Solver Preference	Fluent
<input type="checkbox"/> Element Size	Default (9.1169e-002 m)
Export Format	Standard
Export Preview Surface Mesh	No

(Mesh Details)



(Generated Mesh)

The above mesh was updated to fluent for testing out different boundary conditions. The mesh was converted to “Polyhedral” mesh which is going to ensure faster convergence.

Basic Solver Set up

The chosen solver was the “Density Based (dbns)” solver. Though more computationally expensive and difficult to converge, this is the choice model for dealing with simulations where compressibility effects of the fluid are significant.

The turbulence model chosen was the “kappa-Omega SST (Shear Stress Transport) model. Near the walls, it switches to more of “Kappa-Omega” characteristics while away from the wall it switches to more of the “Kappa-Epsilon” characteristics. This hybrid nature brings in the best of both worlds of the individual “Kappa-Omega” and “Kappa-Epsilon” models values making it exceptionally stable.

The energy equation was turned on.

In the operating conditions, the operating pressure was set to ‘0 Pascals’. The gravity equation was kept turned off. The effect of gravity on these flows is negligible for study. There is no need to add this body force term and increase the computational effort.

In the cell zones, the fluid (air) was set to compressible gas. Keeping everything else at the default constants the viscosity was set to the “Sutherland” model (three coefficient method). The convergence criteria for residuals were left at the default value of ‘1e-3’. This is not good enough to get a physical simulation of such complex flows with supersonic velocities with complexities caused by the shock diamonds in the free jet’s stream. The coarse mesh case was just for the sake of getting initial contours and to observe the behaviour of the residuals from the various equations involved (continuity, energy, turbulence transport equations for ‘kappa’ (turbulent kinetic energy) and ‘omega’ (specific turbulence dissipation rate) and the three momentum equations.

The reference values are then computed from inlet.

Boundary Conditions

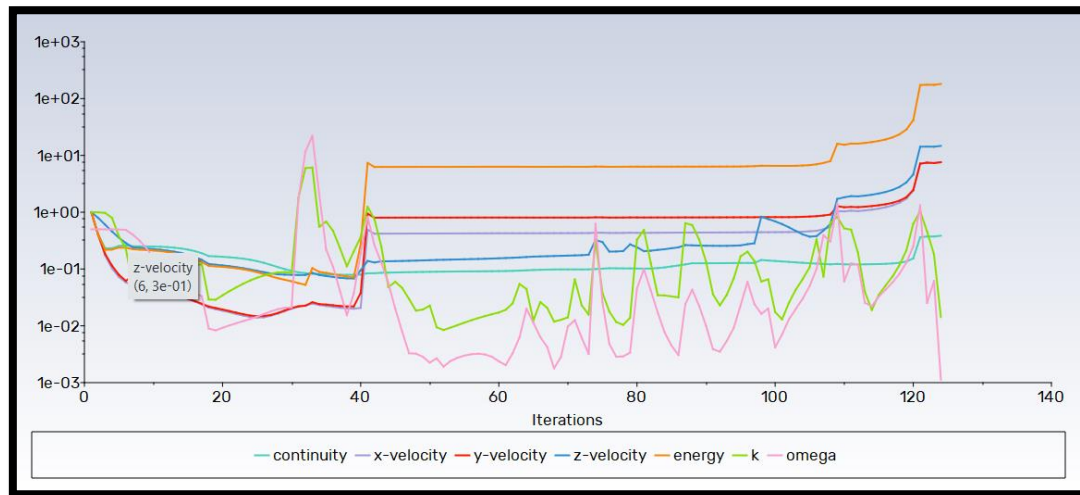
Many different combinations of boundary conditions were tried out to make sure than the final, more refined mesh would be set-up as a well-posed problem.

The resulting residuals were studied.

(The Nozzle walls are defined as adiabatic walls. The nozzle inlet are defined with the pressure inlet condition (Total Gauge Pressure: 6700000Pa, Initial/Supersonic Total Gauge Pressure: 6700000Pa, Total Temperature: 300K) unless stated otherwise).

BC1:

Pressure outlet (gauge Pressure: 101325 Pascals (atmospheric pressure at sea level)) given to all other boundaries (up/downstream faces and the lateral side of the frustum).

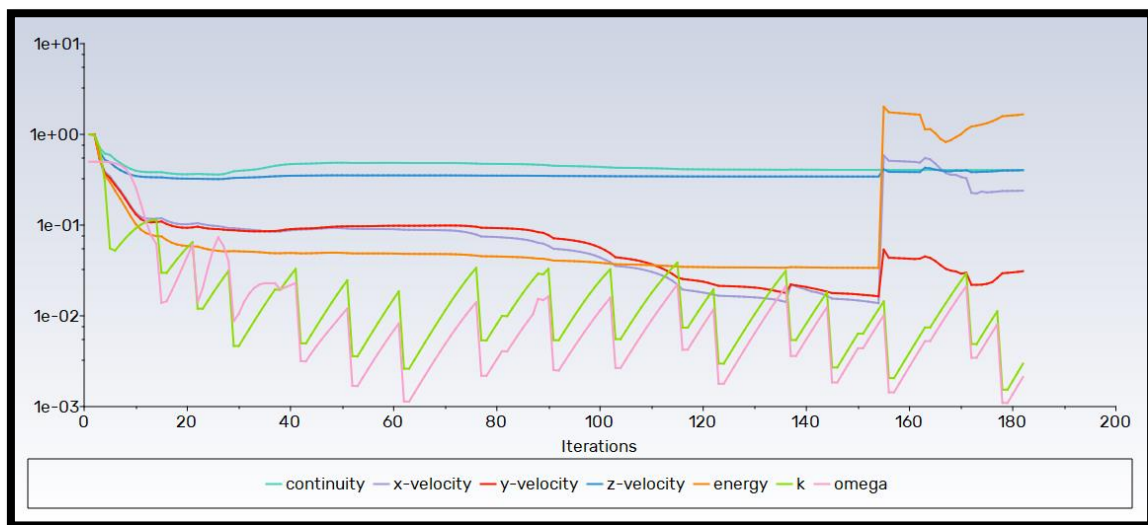


The solution blew up even before the completion of a hundred iterations.

BC2:

The inlet at the nozzles were given the “Mass Flow Rate” inlet condition (Mass Flow Rate: 1.8578 Kg/sec (choked flow’s mass flow rate, at the given conditions, the flow is expected to expand and accelerate to supersonic conditions), Initial/Supersonic Total Gauge Pressure: 6700000Pa).

Pressure outlet (gauge Pressure: 101325 Pascals (atmospheric pressure at sea level)) given to all other boundaries (up/downstream faces and the lateral side of the frustum).

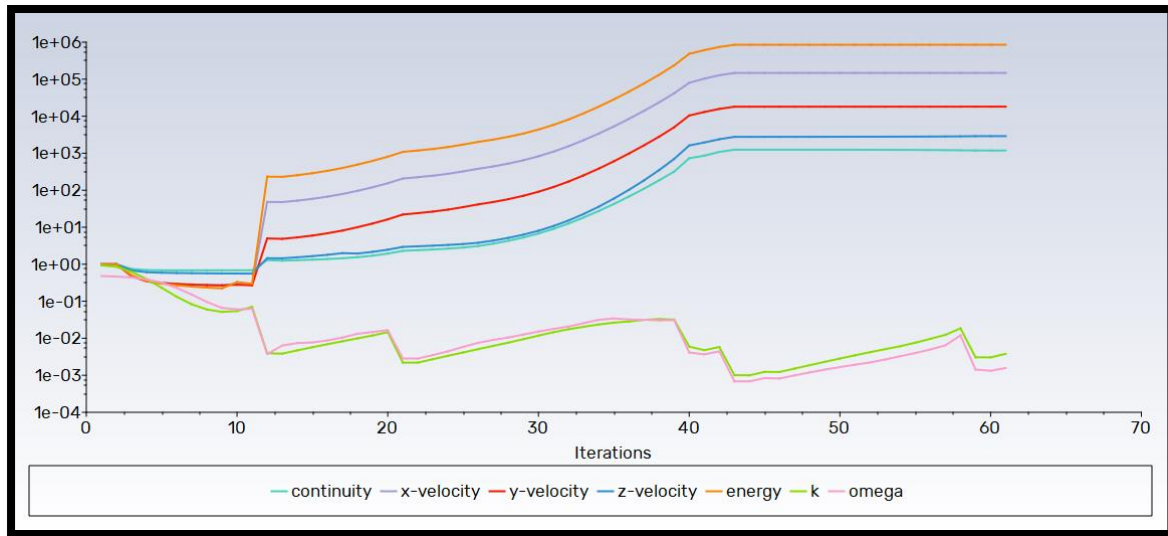


The residuals for the turbulence transport equations were oscillating while the rest of the residuals flat-lined and then blew up after a hundred and fifty five iterations or so.

BC3:

The inlet at the nozzles was again given the “Mass Flow Rate” inlet condition (Mass Flow Rate: 1.8578 Kg/sec, Initial/Supersonic Total Gauge Pressure: 6700000Pa).

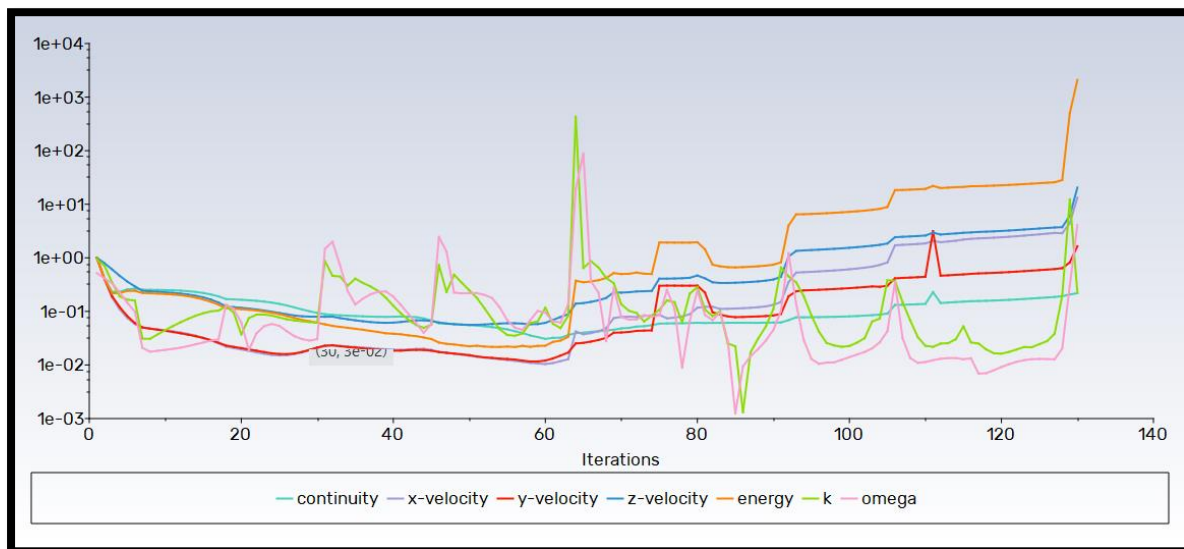
The downstream face of the frustum was given the “Pressure Outlet” BC (gauge Pressure: 101325 Pascals (atmospheric pressure at sea level)) while the upstream face and the lateral sides were given the “Pressure Far-Field” BC (Gauge Pressure: 101325 Pascals).



The residuals diverged almost immediately. The solution was only stable for the first ten iterations.

BC4:

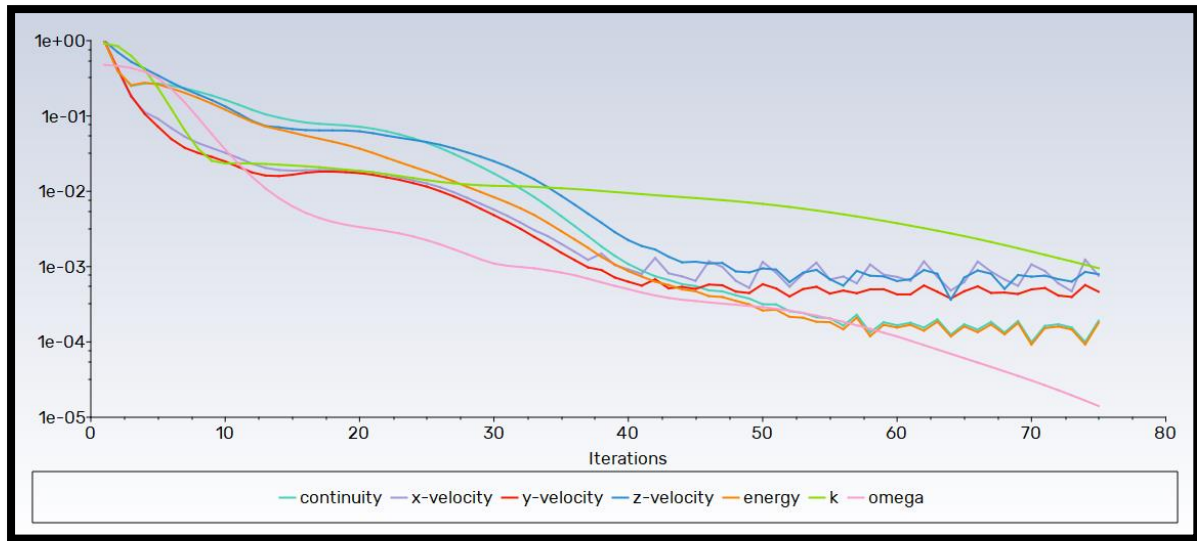
The upstream face was given the “Pressure Inlet” boundary condition (Total Gauge Pressure: 101325 Pascals, Initial/Supersonic Total Gauge Pressure: 101325 Pascals, Total Temperature: 300K), the lateral face was given the “Pressure Far-Field” BC (Gauge Pressure: 101325 Pascals) and the downstream face was assigned the “Pressure Outlet” BC (Gauge Pressure: 101325 Pascals).



The residuals were oscillating, the solver did not show any sign of stability and eventually the solution blew up.

BC5:

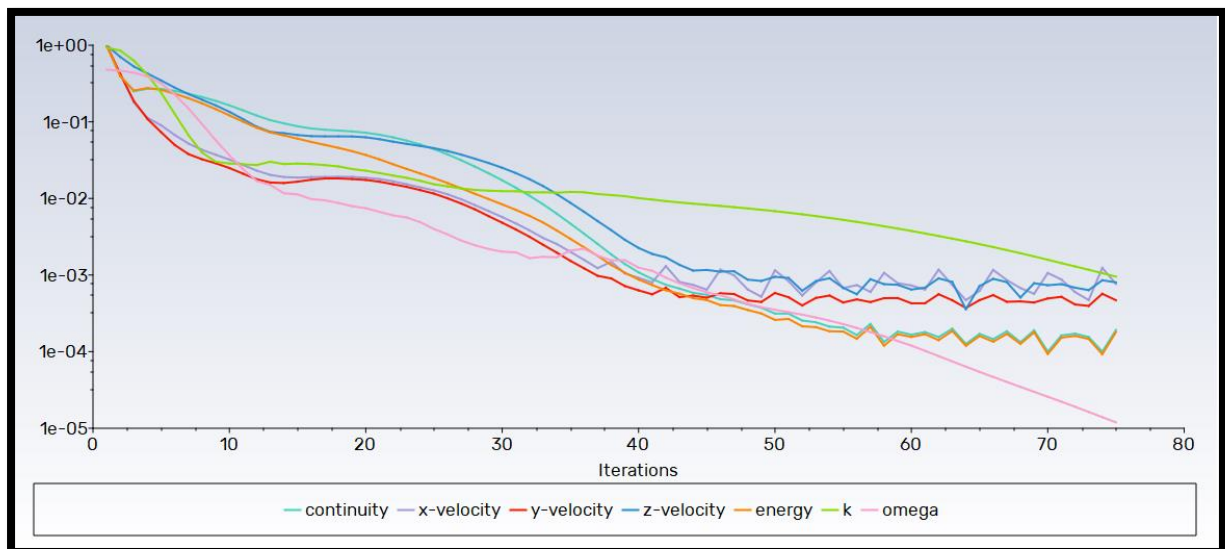
The up/downstream faces and the lateral faces were given the “Pressure Far-Field” BC (Gauge Pressure: 101325 Pascals).



The residuals almost converged. The residuals flat-lined and were oscillating just above the convergence criteria.

BC6:

The upstream faces and the lateral faces were given the “Pressure Far-Field” BC (Gauge Pressure: 101325 Pascals) while the downstream face was given the “Pressure Outlet” BC.

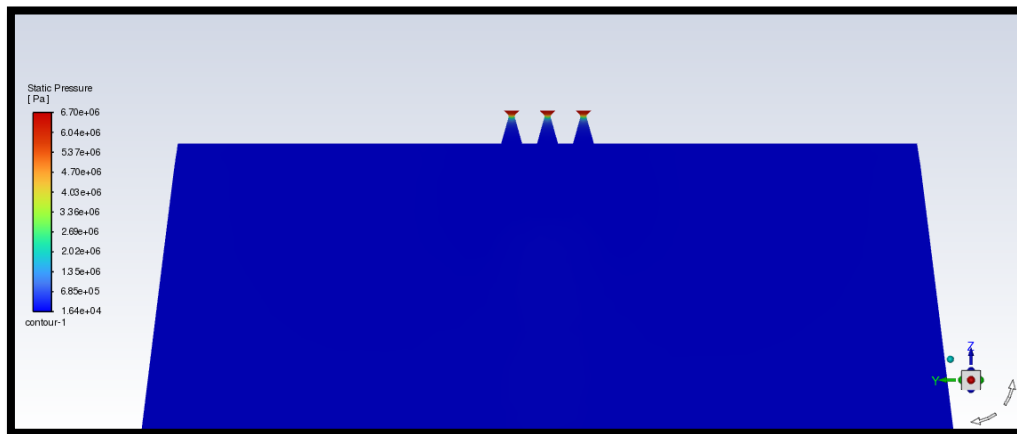
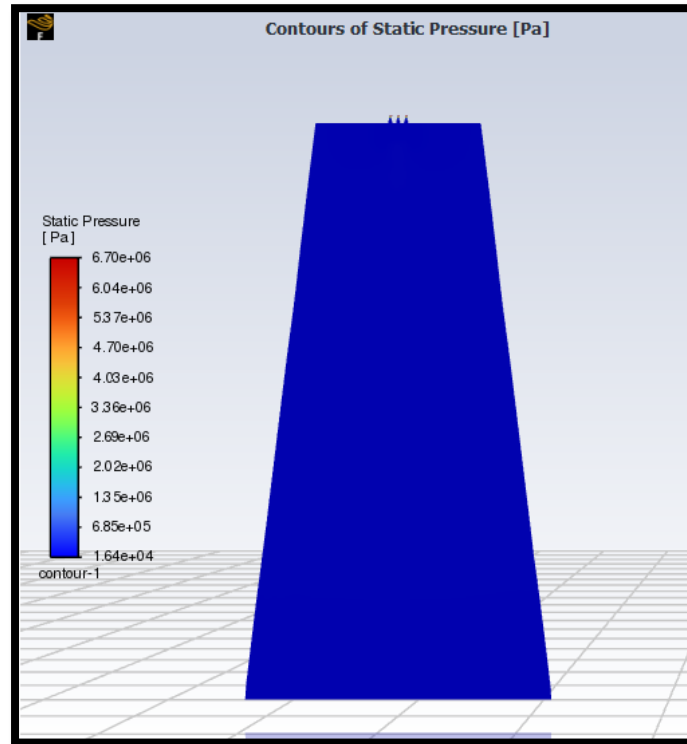


This case fulfilled the set convergence criteria. However, the same oscillations were seen in the residuals and they seemed to flat-line rather than continuing a negative slope towards lower values.

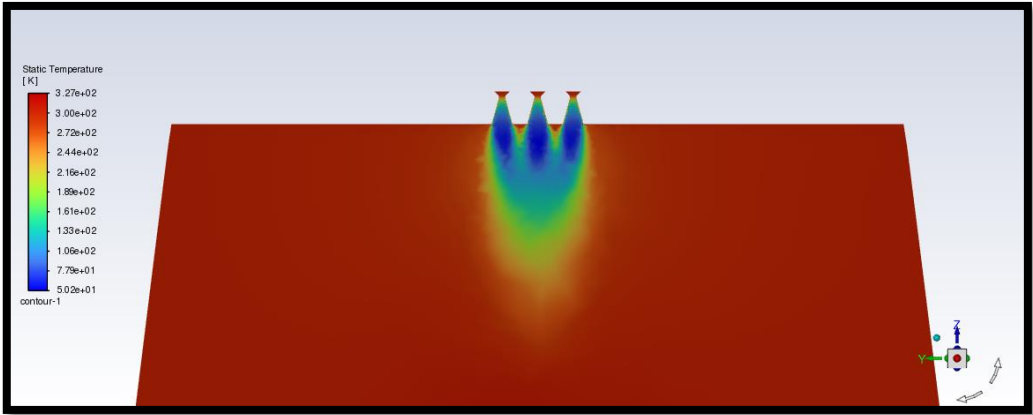
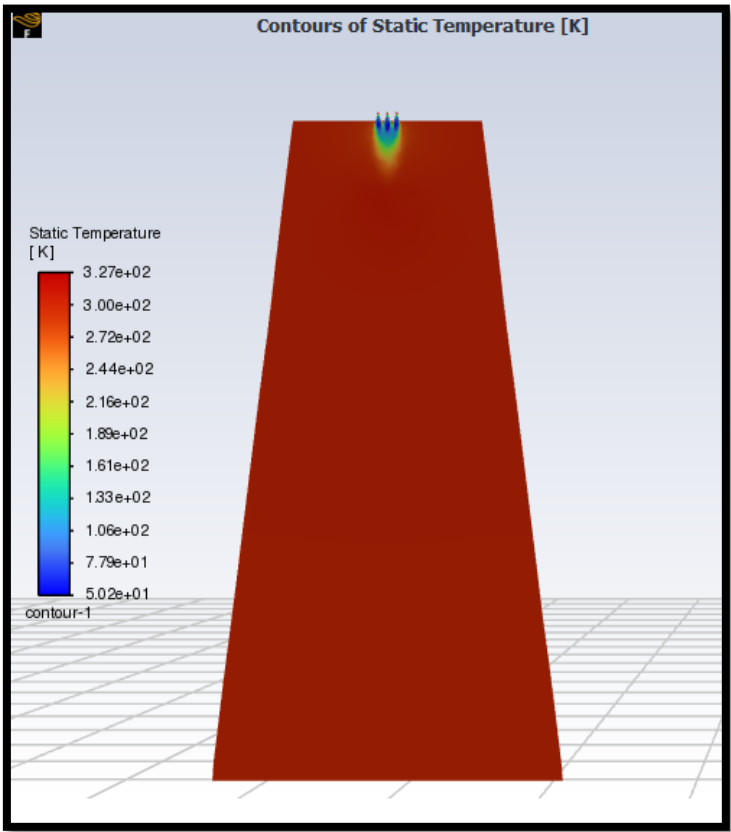
Processing the Converged Case (BC5):

Contours

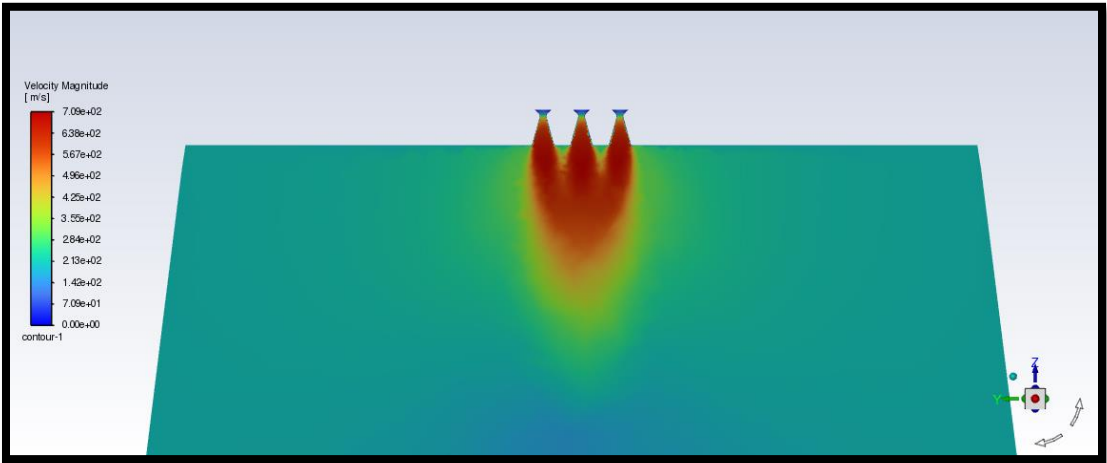
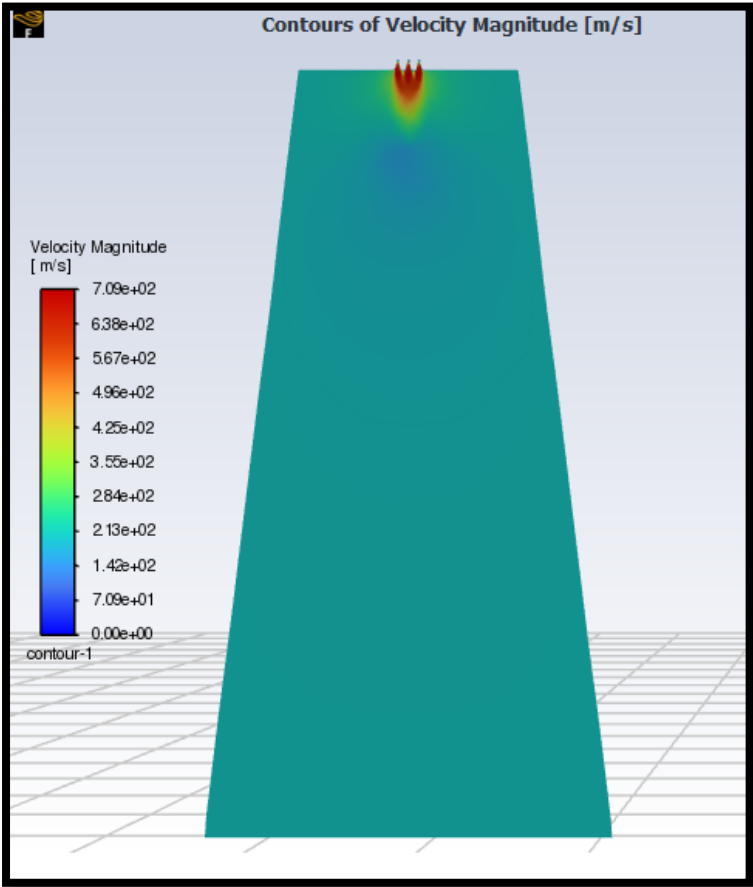
Static Pressure Contours:



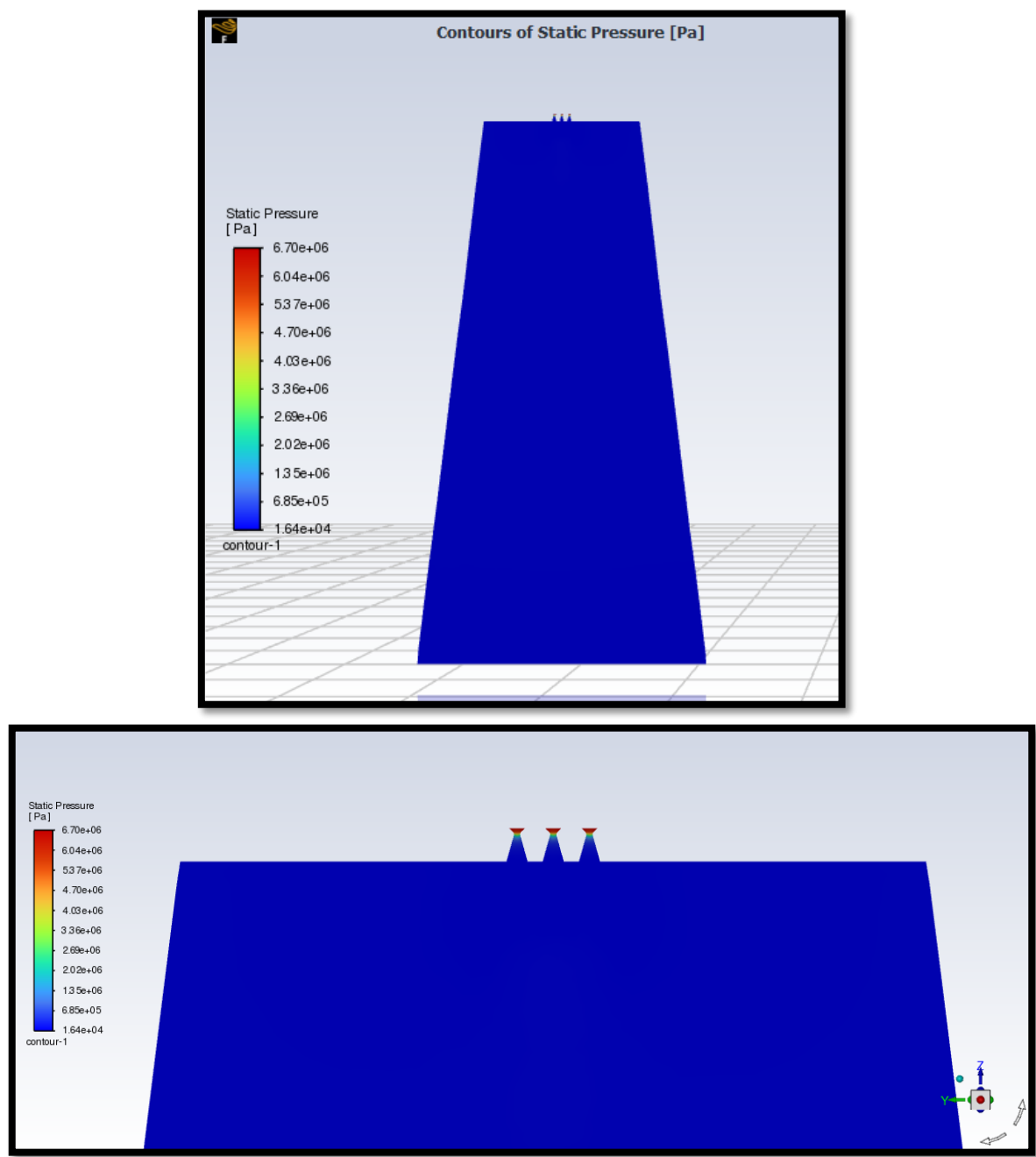
Temperature Contours:



Velocity Magnitude Contours:



Mach Number Contours:



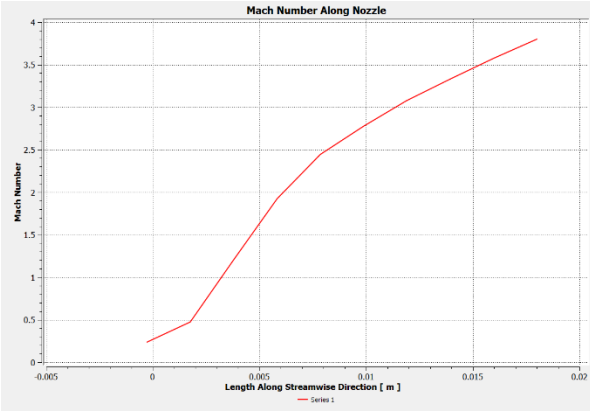
Mass Flow Rate (Computed at Inlet)

Mass Flow Rate	[kg/s]
inlet	1.8383317

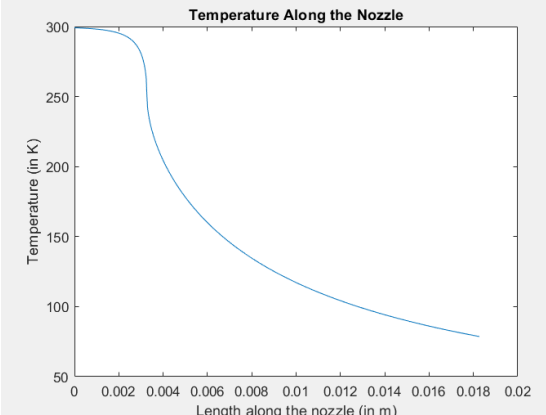
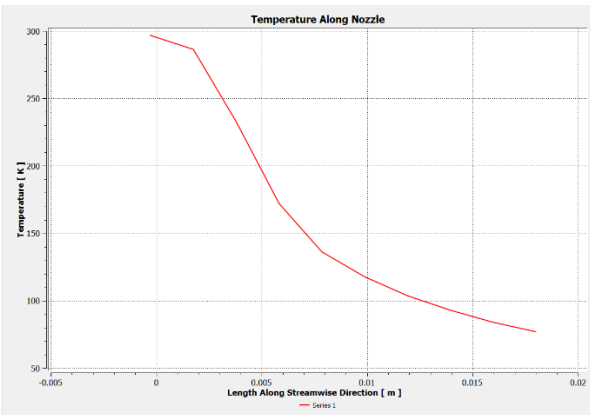
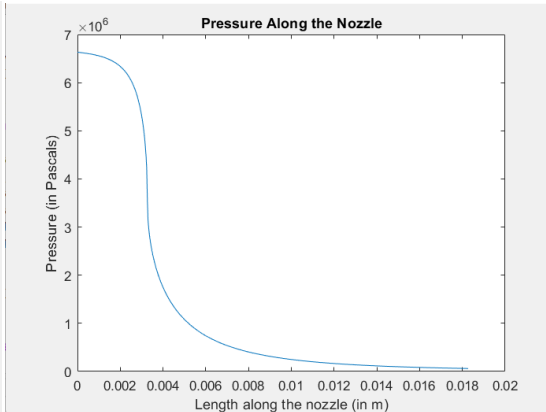
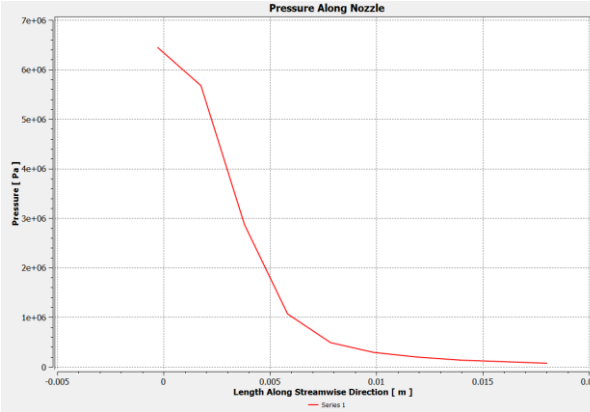
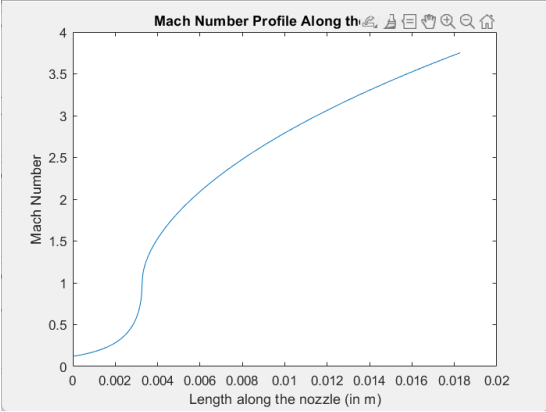
Validation (Comparing Plots from CFD-POST with Plots from MATLAB)

Along Nozzle:

(From CFD-POST)

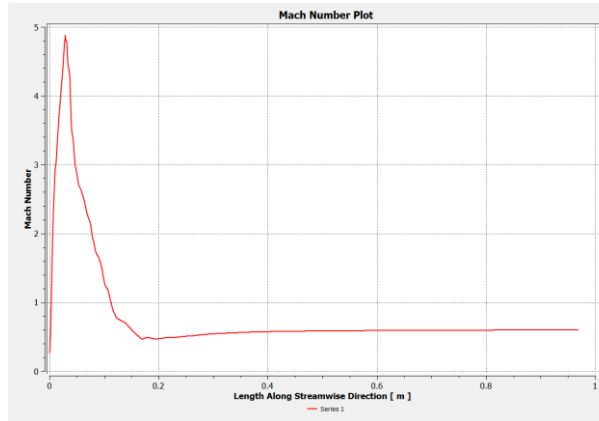


(From MATLAB)

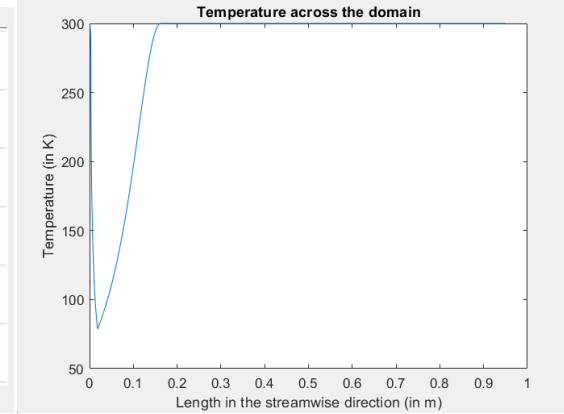
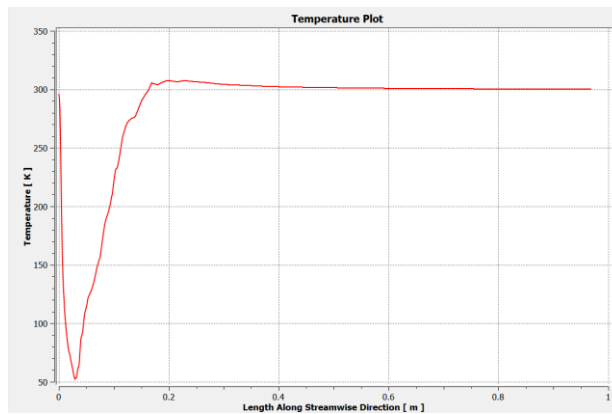
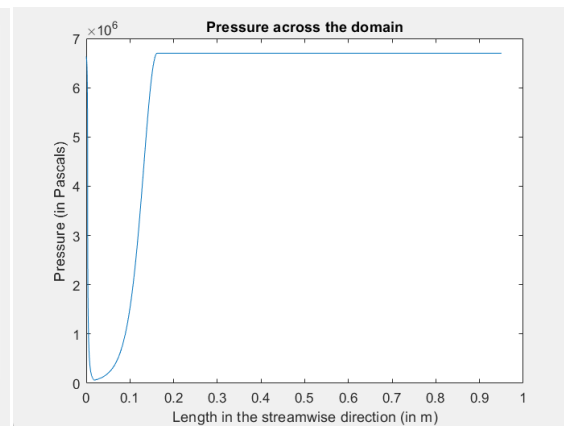
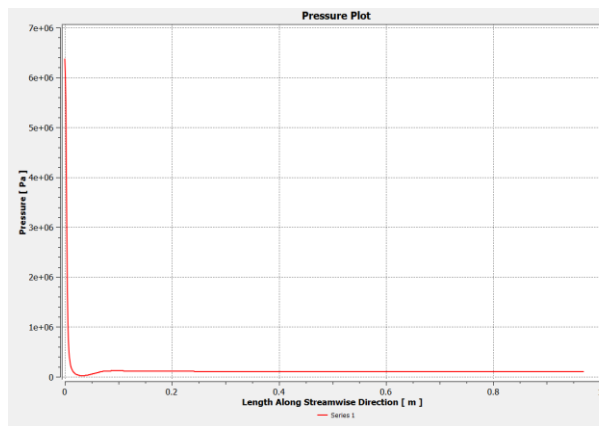
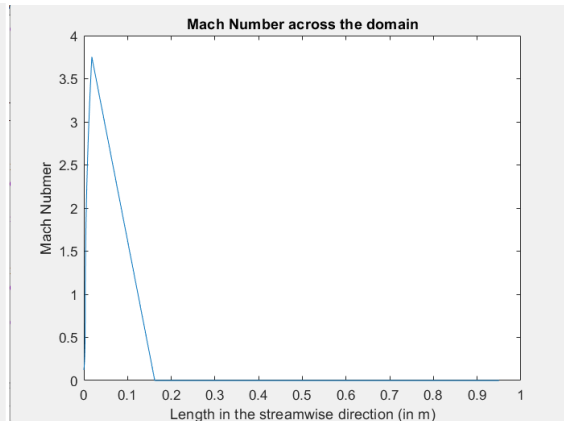


From Nozzle Inlet Plane to 950mm Downstream:

(From CFD-POST)



(From MATLAB)



Note on Pressure Profile Validity:

The pressure profile beyond the nozzle, though presented here for completeness, is derived from isentropic relations that cease to be valid once the jet exits into the ambient environment. In reality, the flow outside the nozzle is subject to strong viscous mixing, shear-layer growth, and non-isentropic processes such as shock waves and expansion fans—none of which are captured by the quasi-one-dimensional model used here. As a result, the analytical pressure curve incorrectly trends back toward stagnation pressure downstream, which is unphysical. In a real or simulated supersonic free jet, the pressure should instead oscillate about atmospheric pressure due to shock-cell structures before gradually equilibrating. This discrepancy is acknowledged and has been addressed in the section discussing the MATLAB code. As such this pressure plot will not be used for validation.

Discussion

Several unphysical characteristics are evident in the current CFD results, due in part to the coarse mesh and relaxed convergence criteria:

- The pressure does not recover to atmospheric values downstream of the nozzle, which is inconsistent with physical expectations for a free jet exiting into quiescent ambient air. The jet appears to remain over-expanded far downstream, even when the exit pressure should be equalized through shock adjustments.
- Shock cells and plume structures expected in such jets are either unresolved or diffused due to insufficient spatial resolution.
- Minor oscillations or abrupt gradients in the Mach number and pressure fields suggest numerical diffusion and possible reflections from domain boundaries.

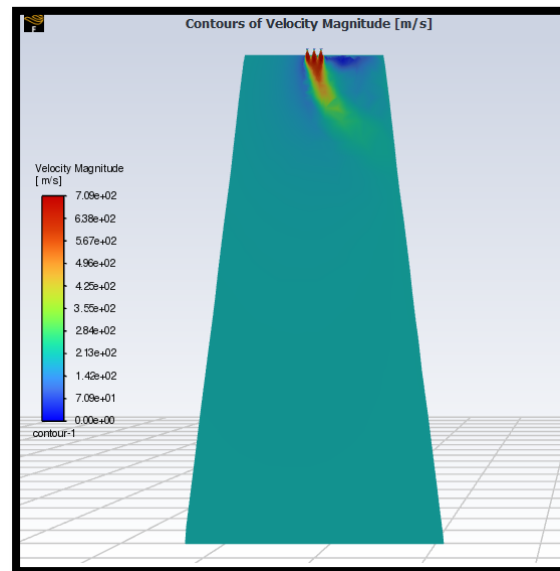
Nonetheless, the general trends in the flow variables from both the plots and the contours match those predicted analytically, which validates the robustness of the simulation set-up, particularly the boundary conditions and thermodynamic definitions. These results justify proceeding to finer mesh simulations, which are necessary to resolve small-scale features like shock diamonds, shear layers, and shock–boundary layer interactions, and to minimize numerical errors inherent in compressible CFD computations.

Learnings from the Coarse Mesh Run

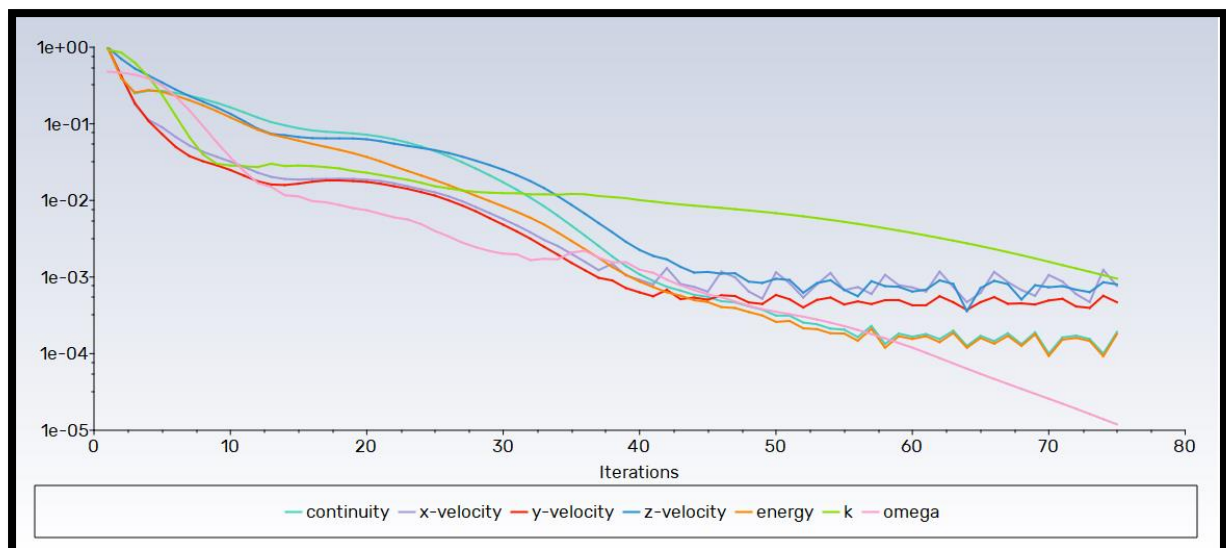
The boundary conditions (BC6) and the basic solver set-up yield results that do not deviate from analytical expectations. This is a good sign for this base-line set-up.

However, the convergence criteria are too low to make this numerical solution a good representation of the actual physics. The jet from the contours is too poorly resolved.

More worryingly, despite the symmetric geometry and boundary conditions, the jet in the 'XZ' plane is skewed away from the centre.



(The solution is clearly not resolved in this plane.)



Strong numerical oscillations had set in. The residuals stopped its desirable downward trend just after forty iterations. If the convergence criteria would have been set any lower, the solver would keep iterating rather pointlessly. Despite the residuals not “blowing up” and diverging, this solution cannot be considered a good representation of the actual physics. However, it served its purpose well as a good base-line to improve upon.

Dealing with the Numerical Oscillations

To deal with these numerical instabilities, a strategy was devised that involves exploiting the “numerical diffusion” that first order schemes are prone to. As the name suggests, this “diffusion” is purely numerical and is unphysical, making it undesirable in numerical solutions. This results from the larger truncation errors of the first order schemes on comparing with its higher order cousins.

Despite their higher accuracy, the higher order schemes (like the second order upwind schemes) are very prone to numerical oscillations that can stagnate the convergence progress of a simulation. These oscillations are not uncommon when running a numerical simulation of a super-sonic jet with many flow intricacies like shocks and plume interferences.

The other part of the strategy involves reducing the “Courant number”. It is a dimensionless number that quantifies how the information from the boundaries travels within the domain. Thus, it is a very effective metric for measuring the probability of a solution converging.

$$C = U \frac{\Delta x}{\Delta t}$$

Here, ‘U’ denotes the velocity of the flow, ‘ Δt ’ denotes the time-step (or a pseudo-time step in this case of a steady state simulation) and ‘ Δx ’ denotes the grid spacing.

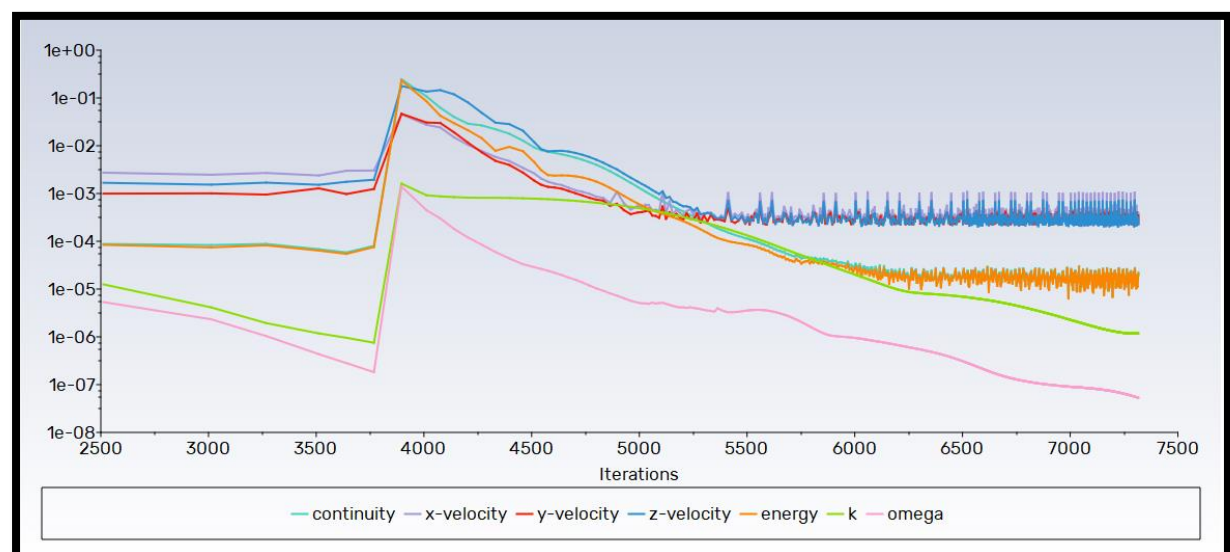
The above formula is for a simplified 1-dimensional case.

A low courant number would thus imply a less aggressive solver. Though slower to converge, it will be more numerically stable.

First order schemes coupled with a low courant number should stabilise the solver enough to meet the convergence threshold.

Trials:

For the first trial, the coarse mesh itself was tried out. By around 3800 iterations the graphs of the residuals was smooth and not showing divergence. However, they were not showing a downward trend. On switching to second order from here, the residuals showed a sharp increase and then after stabilising for a few hundred more iterations, the oscillation set in again.



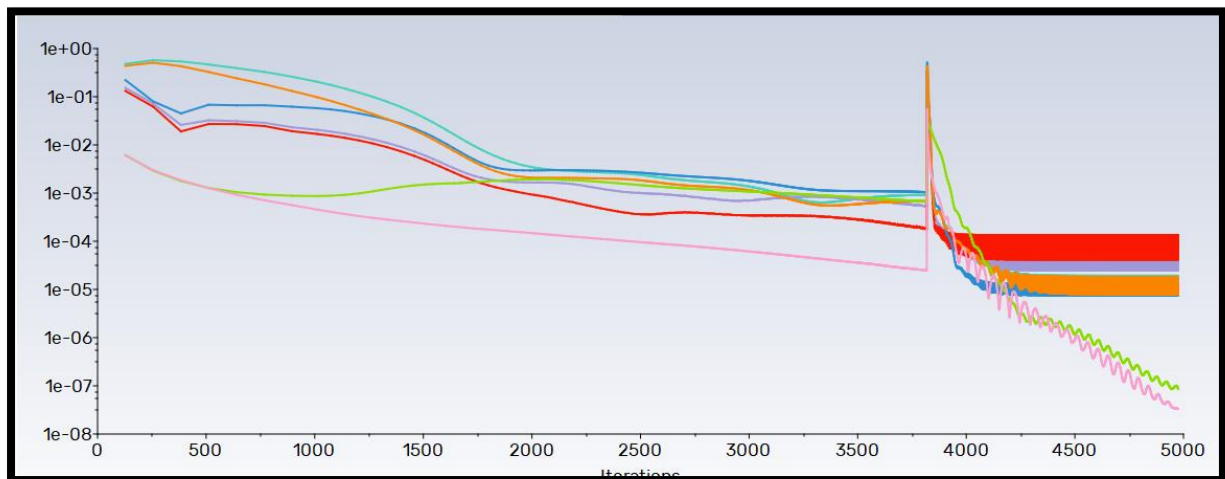
The residuals here were much lower than what was recorded with the previous solver settings.

So, overall, this can be considered a better solution, a step in the right direction.

The unphysical smoothening caused by numerical diffusion from the first order schemes are useful in quenching the numerical oscillations that such solutions of such set-ups usually show. However, they can also lead to inaccuracy. It is known that first order schemes are more accurate when the flow is aligned with a structured mesh.

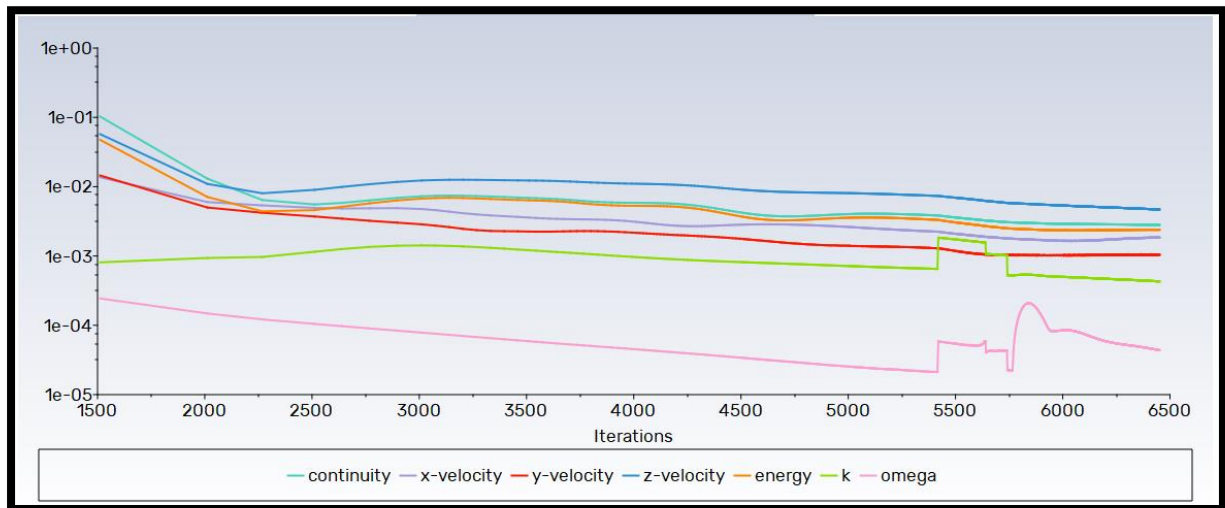
To make the mesh more structured it was decided for further trials to run a mesh that is more structured. By default, the “Automatic” methods discretise the domain with an unstructured mesh that is mostly composed of tetrahedral elements. Thus, for the later runs, the “Multizone” method was used. It creates a more structured mesh that is composed of hexahedral cells.

With a coarse “Multizone” mesh, the residuals actually showed a good downward trend. At around 4000 iterations, the solver was switched back to second order schemes, and the familiar oscillation set in, refusing to abate even after a thousand additional iterations. Moreover, the downward trend of the residuals stopped as a result of this (except for the ‘K’ and ‘Omega’, which had a downward trend, but still showing oscillations).



A more refined mesh was tried. For local body sizing, the body of influences were used. At the nozzle, a size of ‘1mm’ was given, while the mesh for the region for the jet was given ‘7mm’ while global average was set to 15mm.

The residuals started much lower than the previous case, and were showing a steady (but low) downward trend. At around ‘5500’ iterations, the courant number was bumped up to ‘0.2’. This can be seen from the blip in the residual for ‘K’ and ‘Omega’. Despite the residuals continuing their steady downward trend, the reverse flow at the outlet boundary was showing a sharp increase. It was clear that the solver was very sensitive to the outlet. The reverse flow that had gradually resolved and decreased over a span of ‘6000’ iterations began to sharply increase.



To combat this, for the final case, the outflow boundary would be increased from '40Deq' to '50Deq'.

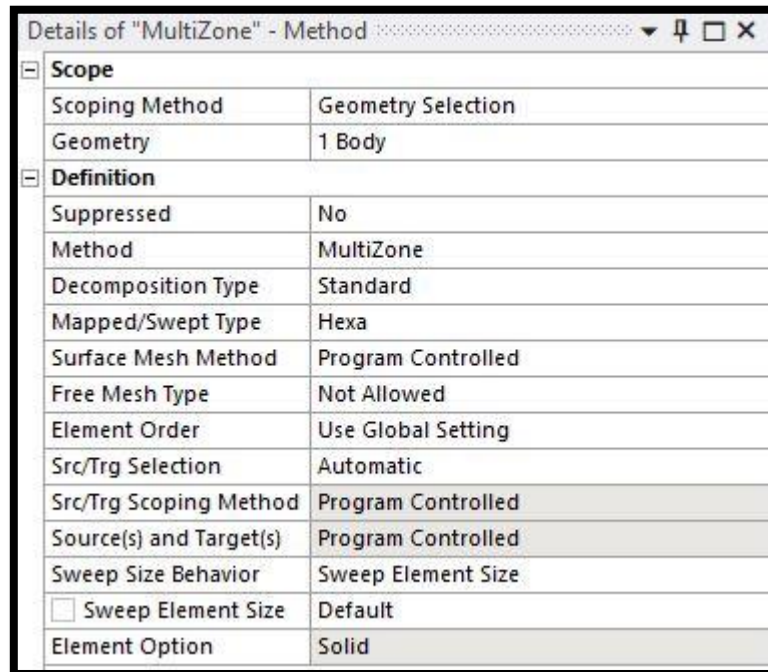
Additionally, the courant number would be kept at '0.1'.

On comparing the above plots, it's easy to visually make out that the residuals are lower in the case with the refined "Multizone" mesh. Also, the residuals show a visible downward trend with no oscillations. So far, all the trial runs have been improvements and have provided valuable insights for finalising the set-up.

Full Run 1

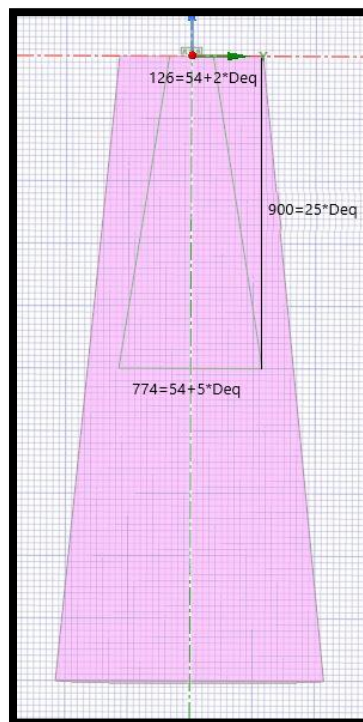
Meshing Strategy:

The “Multizone” method was used with the following details:



Body sizing would be done by defining body of influences (BOIs) around the region of interest.

One BOI encloses the nozzles while the other is located within the frustum shaped domain where the jet is expected to develop.



The BOI for the jet also takes the shape of a frustum for the same reason, computational economy.

Mesh Sizing details for the BOIs (Jet region and the nozzle region):

Details of "Nozzle_Sizing" - Sizing	
Scope	
Scoping Method	Geometry Selection
Geometry	1 Body
Definition	
Suppressed	No
Type	Body of Influence
Bodies of Influence	1 Body
<input type="checkbox"/> Element Size	1.0 mm
Advanced	
<input type="checkbox"/> Growth Rate	Default (1.2)

(For Nozzles, the average element size was set to be 1mm. The minimum dimension corresponds to the throat diameter which measures 4.1mm. Thus a mesh with local sizing as 1mm should capture the geometry well. The generated mesh should fit four individual cells at the throat.)

Details of "Jet_Sizing" - Sizing	
Scope	
Scoping Method	Geometry Selection
Geometry	1 Body
Definition	
Suppressed	No
Type	Body of Influence
Bodies of Influence	1 Body
<input type="checkbox"/> Element Size	4.0 mm
Advanced	
<input type="checkbox"/> Growth Rate	Default (1.2)

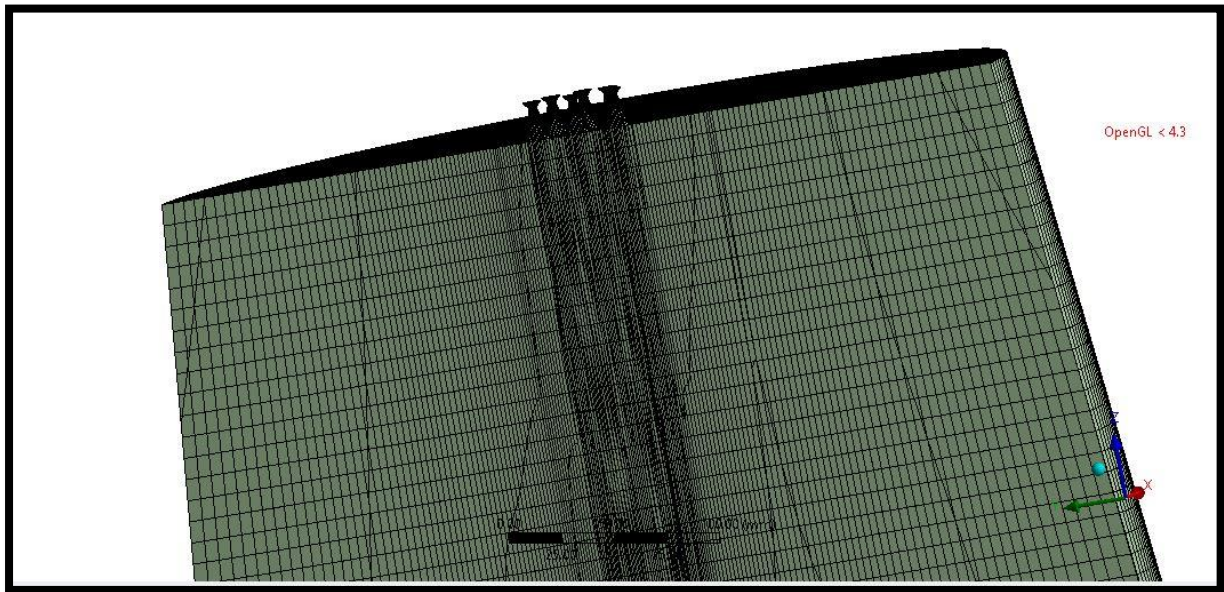
(For Jet region, with the equivalent diameter as 36mm, the local mesh here with the element size as 4mm should capture the complex dynamics in the expected region of the free jet. The generated mesh would have 9 individual cells across the diameter of the equivalent nozzle.)

Details of the generated mesh:

Display	
Display Style	Use Geometry Setting
Defaults	
Physics Preference	CFD
Solver Preference	Fluent
<input type="checkbox"/> Element Size	10.0 mm
Export Format	Standard
Export Preview Surface Mesh	No
Sizing	
Quality	
Inflation	
Advanced	
Statistics	
<input type="checkbox"/> Nodes	2858040
<input type="checkbox"/> Elements	2815783
Show Detailed Statistics	No

(10mm is the global average of the element size for the domain, the number of nodes in this finer mesh is around 2.86 million and the numbers of elements are 2.82 million.)

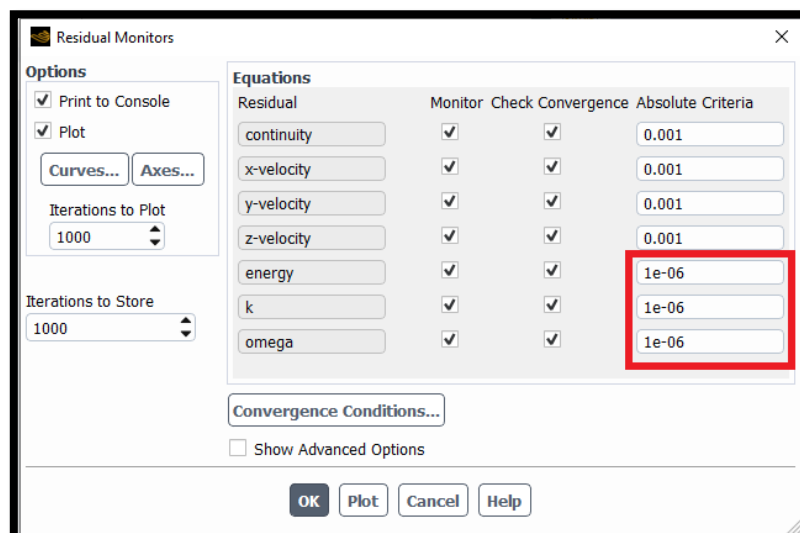
View of the generated mesh from a “Cut Plane”:




Solver Set-Up:

The basic set-up and the boundary conditions were carried out from the previous runs. However, changes were made to the solution methods and controls based on the lessons from the trial runs.

Also, the convergence criteria were updated to ensure a precise solution. For the solution to be declared “converge”, the residuals for the energy and the turbulence transport equations would have to meet the more stringent threshold of ‘1e-6’.



Changes made to the solution “Methods” and “Controls”.

Solution Controls 

Courant Number
0.1


Under-Relaxation Factors

Turbulent Kinetic Energy
0.8

Specific Dissipation Rate
0.8

Turbulent Viscosity
1

Solid
1

Solution Methods 

Formulation
Implicit

Flux Type
Roe-FDS

Spatial Discretization

Gradient
Least Squares Cell Based

Flow
First Order Upwind

Turbulent Kinetic Energy
First Order Upwind

Specific Dissipation Rate
First Order Upwind

Pseudo Time Method
Off

Transient Formulation

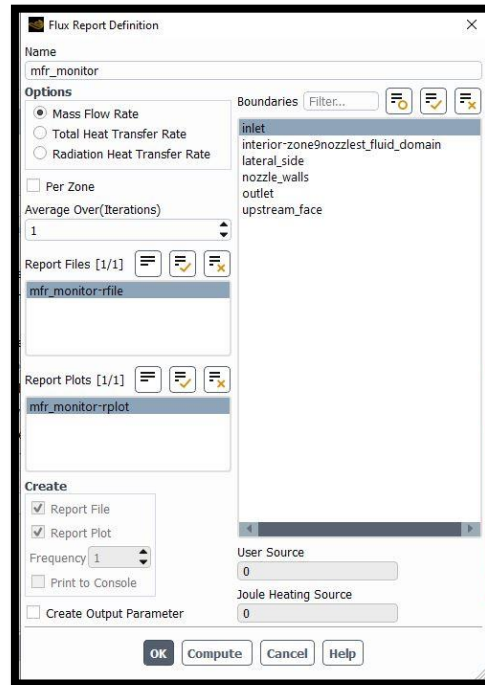
☐ Non-Iterative Time Advancement

☐ Frozen Flux Formulation

☐ Warped-Face Gradient Correction

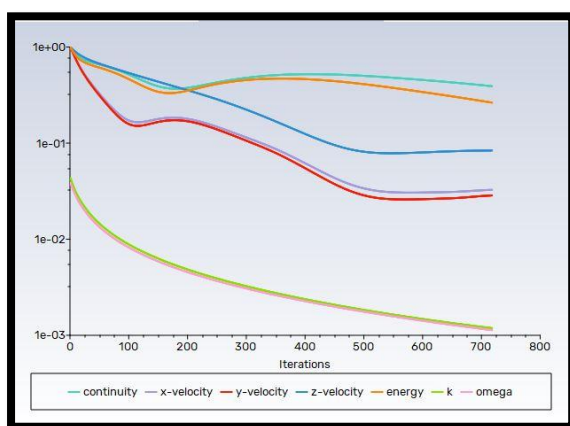
At the steady state, it is expected that flow will be supersonic at the nozzle exit, that is, the flow will be choked at the throat. The choked mass flow rate is calculated to 1.8578 (from the MATLAB code).

Since the flow rate is known, a report definition for the mass flow rate was created as a monitor.

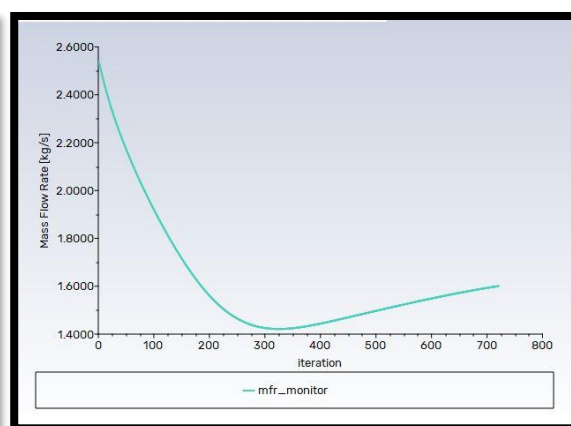


The set up was initialised using the 'Hybrid' method.

The solution was run initially on the system to see how the residuals and the mass flow rate changed over the first few hundred iterations.



(Plot of Scaled Residuals)



(Plot of Mass Flow Rate in 'Kg/sec')

For the '700' iterations that were run on the system, no oscillations or divergence was seen. The solver seemed to be performing smoothly. The mass flow rate looked like it would make the upward trend to reach around 1.8 'Kg/sec'.

This set up was then re-initialised so that it could run using the HPC facilities.

Solution Process	
Update Option	Submit to Remote Solve Manager
RSM Queue	Ansys
RSM Queue Details	
HPC Configuration	SCOF/RO
HPC Queue	ansys
HPC Type	SLURM
Job Name	Workbench
Download Progress Information	<input checked="" type="checkbox"/>
Progress Download Interval	120
Execution Mode	Parallel
Number of Processes	120

(HPC Configuration.)

Results:

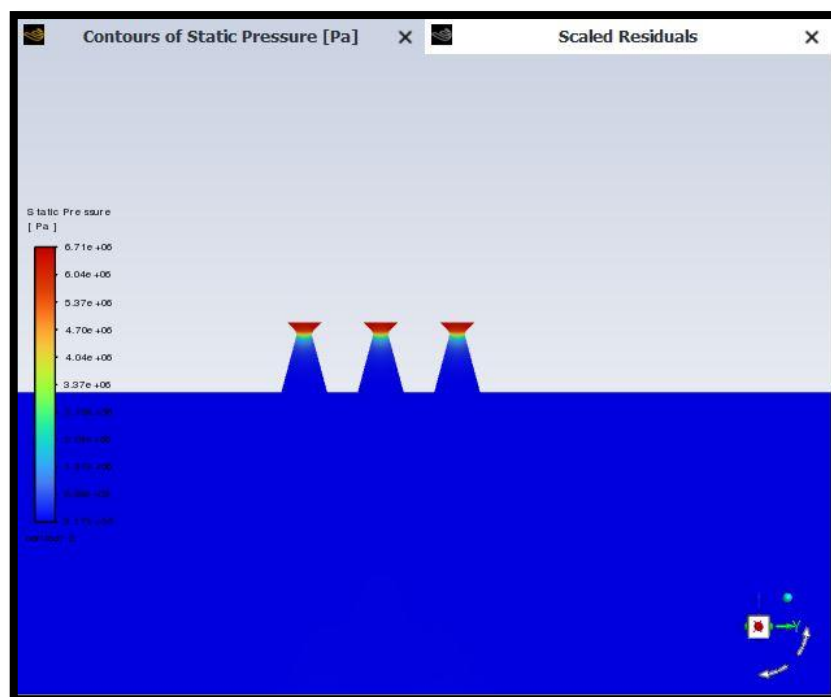
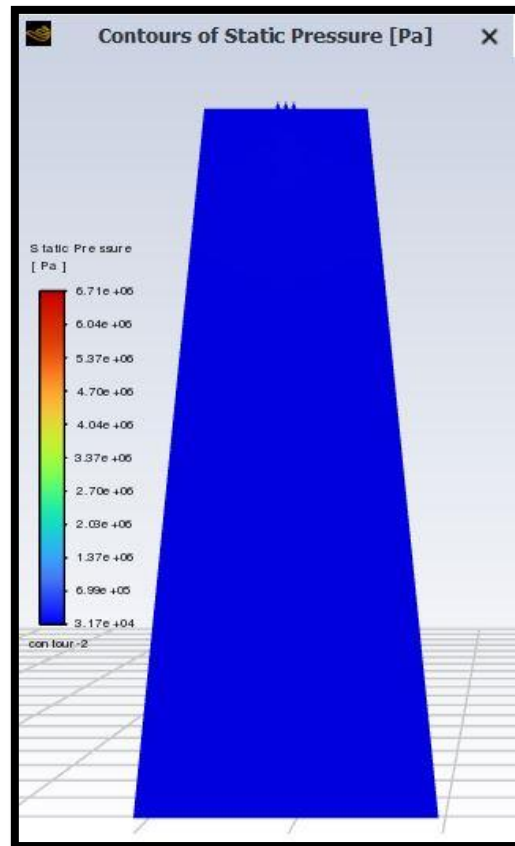
The maximum number of iterations was given to be 150,000.

The solutions converged by 128,500.

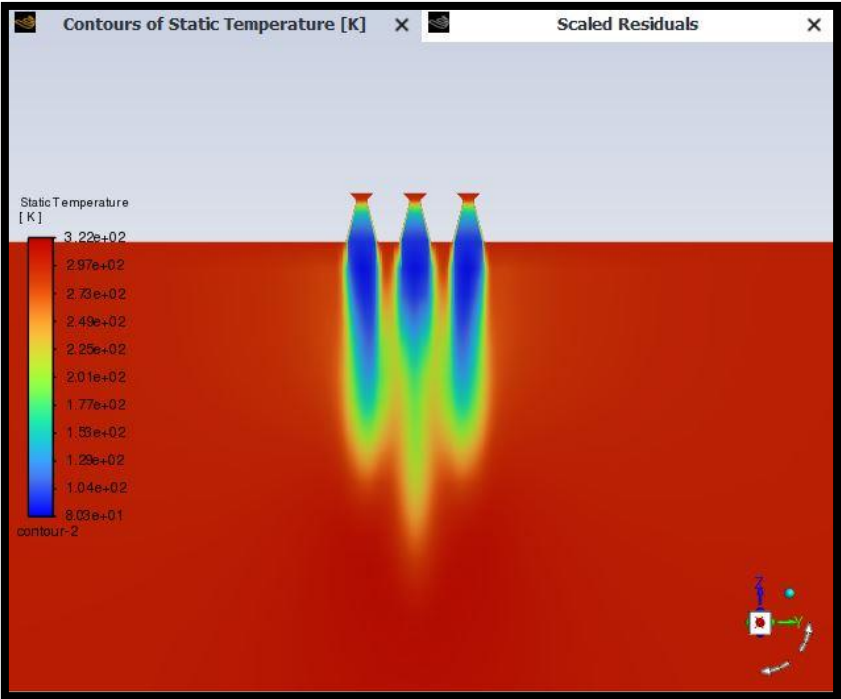
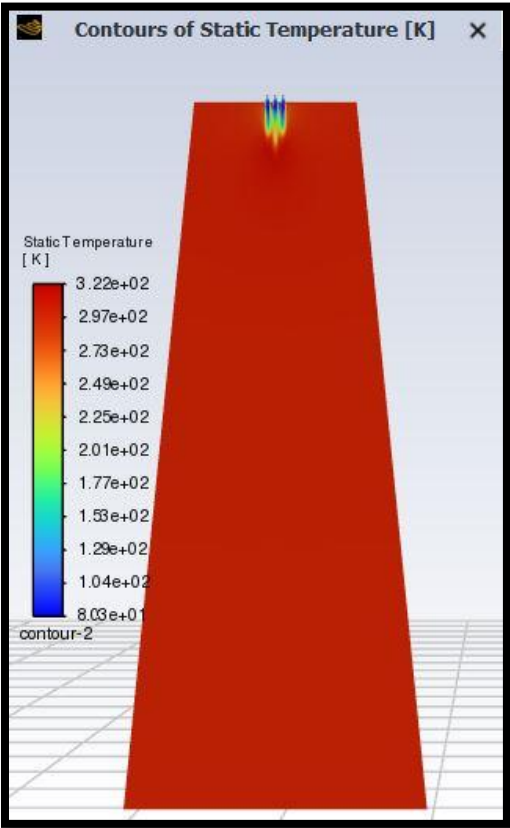
The oscillations set in much later and the residuals were much lower in comparison to the previous simulations. The improved meshing strategy worked. The residuals were able to meet the updated convergence criteria within the set limit of iterations (150,000).

The contours from the solution:

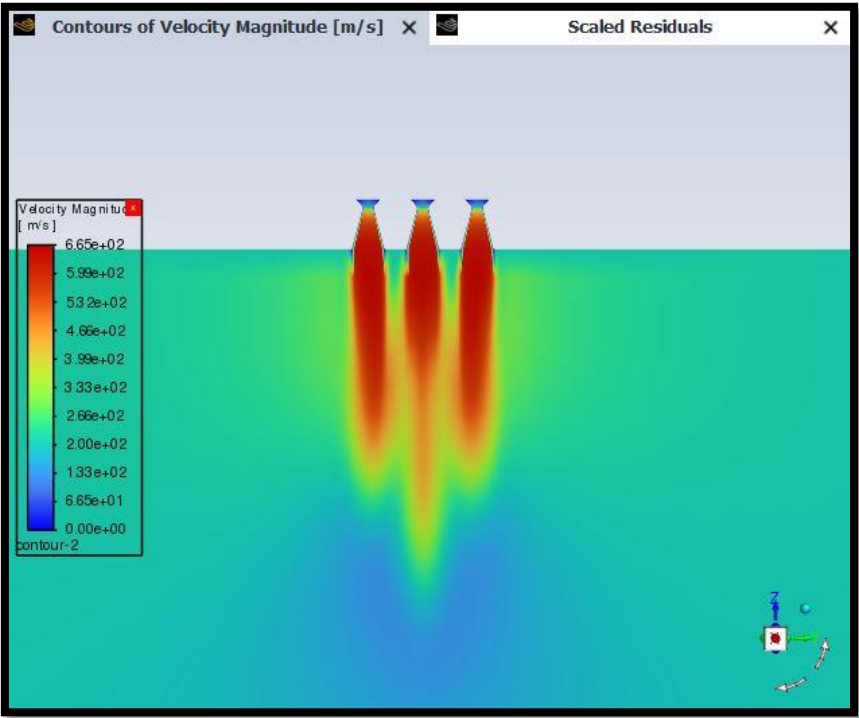
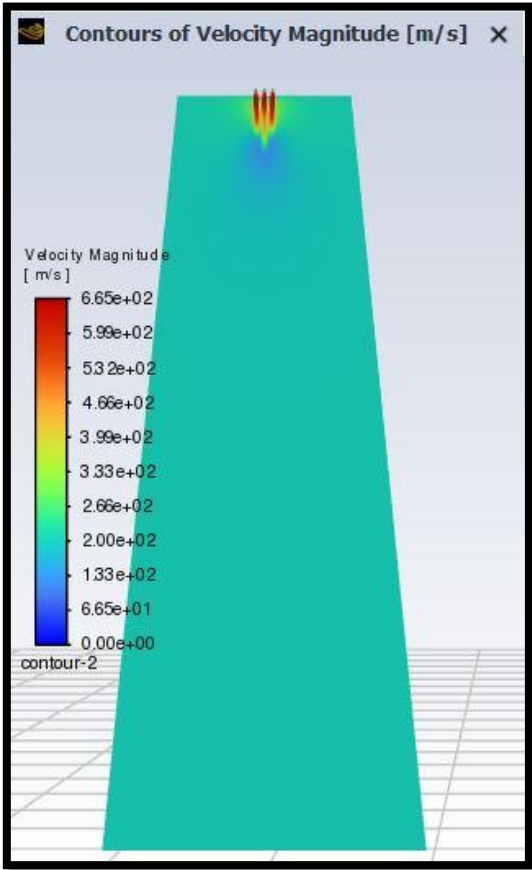
Static Pressure Contours:



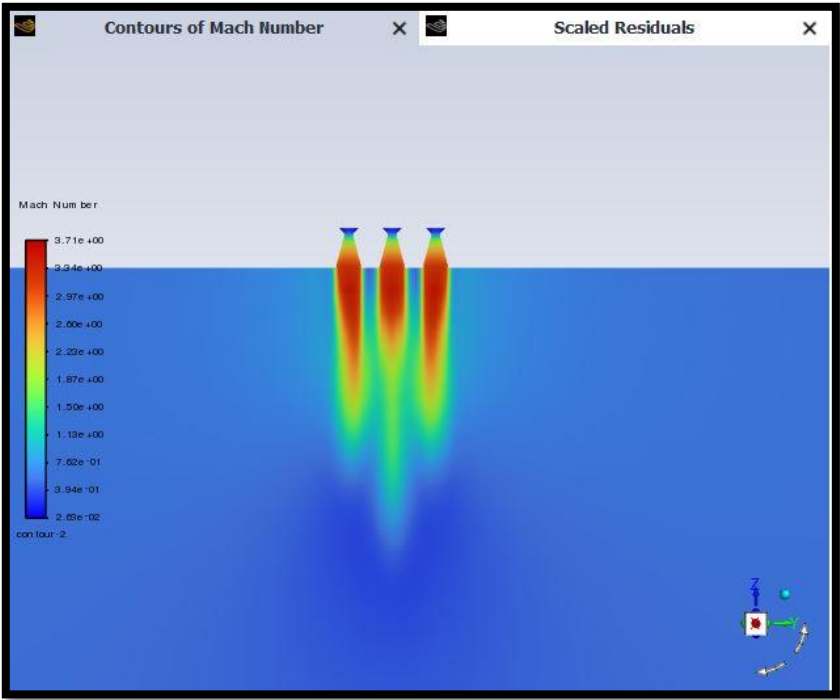
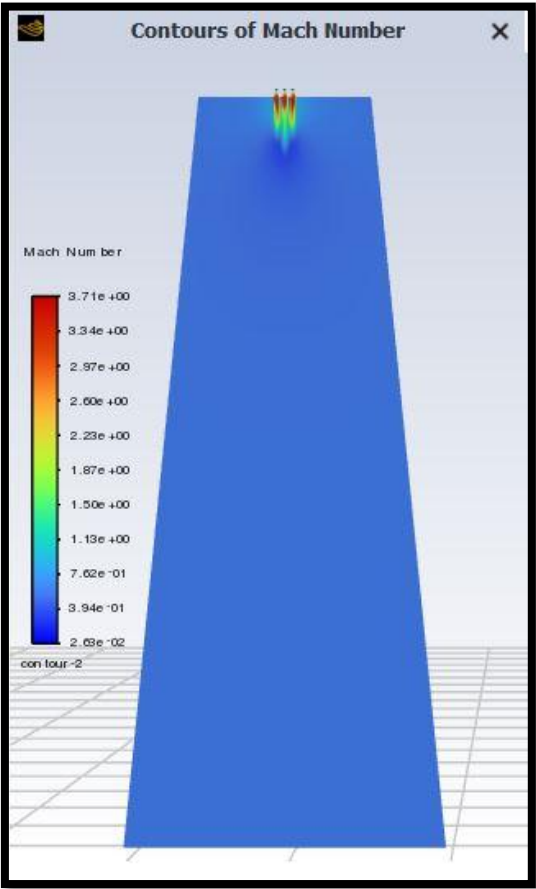
Temperature Contours:



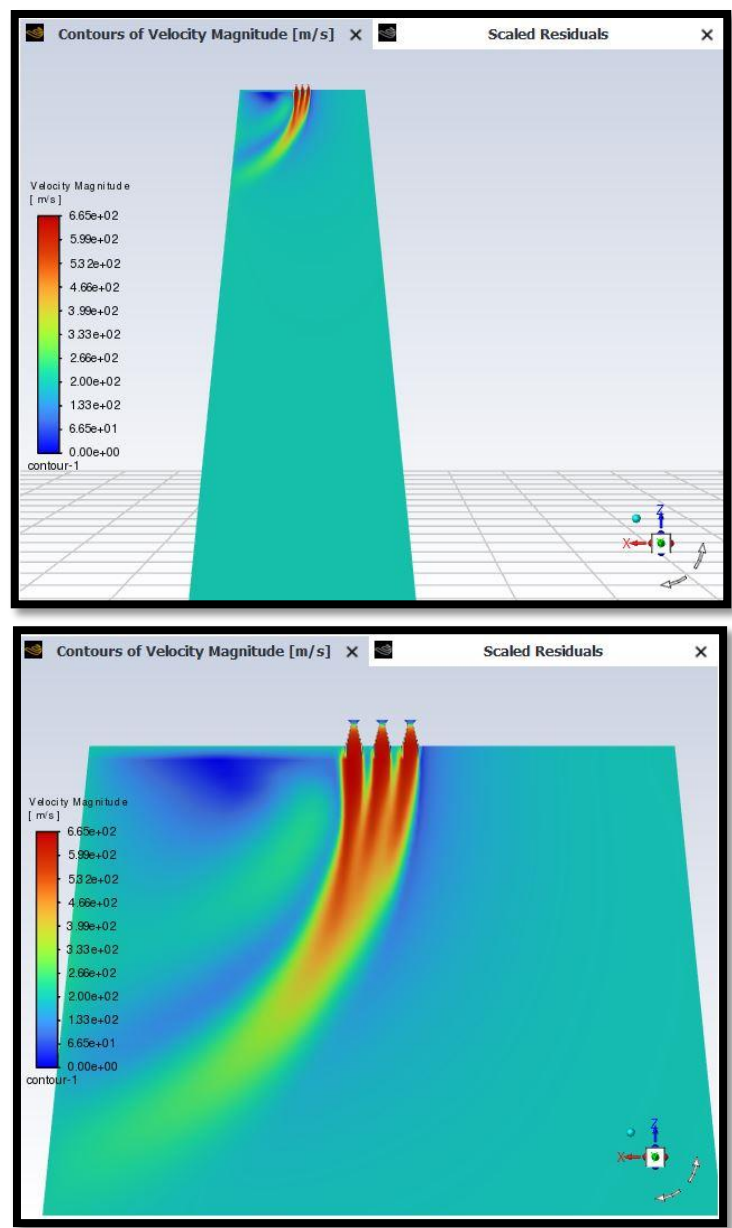
Velocity Magnitude Contours:



Mach Number Contours:



The features are more finely resolved, however, the mesh was not resolved enough to capture the shock interactions. Also the asymmetry in the ‘ZX’ plane persisted. However, it seems like an improvement over the coarser mesh. So, it cannot be ruled out as a mesh independent problem.



The mass flow rate from this simulation is more physically realistic. The computed value from the coarse mesh trials was very close to the analytically calculated value which makes many convenient assumptions.

Report Name	Value	Unit
mfr_monitor	1.6968183	[kg/s]

An Improved Mesh

Seeing that none of the previous solutions showed signs of divergence, the focus should be on the mesh. It was clearly not fine enough to capture the intricacies of such complex flow regimes.

A new mesh strategy was thought out. This time, it was decided for the mesh to be at least as small as ten percent of the smallest defining dimension. Instead of the “equivalent diameter”, the exit diameter of the individual nozzles would be taken for deciding the mesh sizing for the jet.

For the nozzle, the smallest characteristic dimension would be its throat diameter (4.1mm). Thus, the mesh element sizing here would be ‘0.4mm’. For the jet, the smallest characteristic dimension would be taken to be the exit diameter of the individual nozzle (12mm). Thus, following the same scheme, the element size here would be ‘1.2mm’.

A primitive mesh made involved giving ‘5mm’ element size everywhere else in the domain. The smaller size was chosen to prevent drastic change in sizing for the elements between the jet BOI and the rest of the domain. Such an abrupt change in sizing would create a higher possibility of “bad quality” elements (with high aspect ratio, skewness and low orthogonality).

A more obvious drawback is the computational inefficiency of such a mesh. A 5mm’ element size is needlessly fine in the far-field parts of the domain and is thus an unnecessary drain on the computational resources.

The resulting mesh had about ‘11.5’ million mesh elements. The computer had trouble in generating the mesh. Fluent could not even initialise the solution on the local machine.

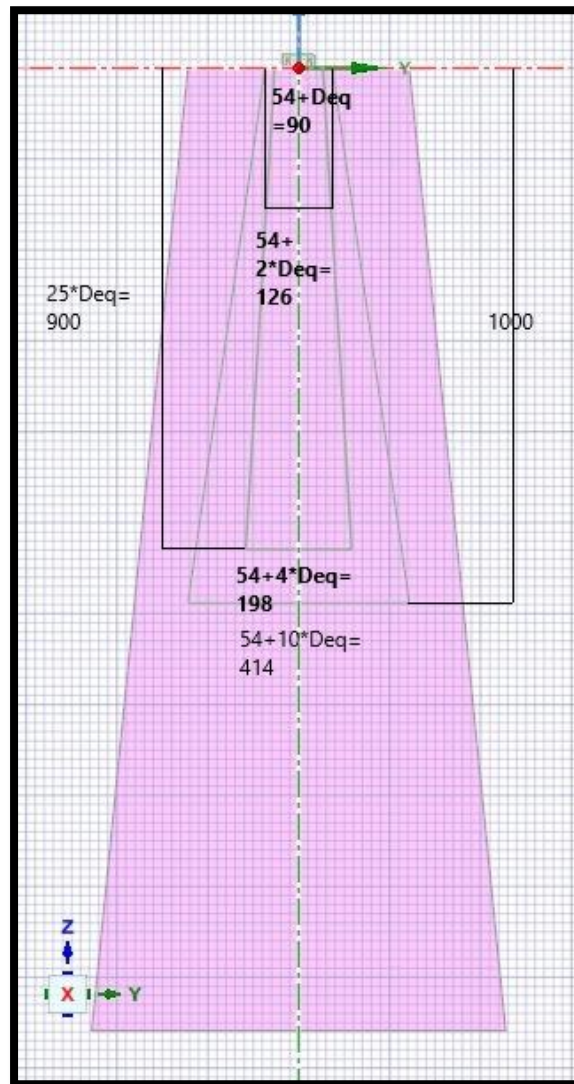
The better approach was to split the ‘BOI’ for the jet into two parts, an inner volume and an outer volume. The volume on the inside can be made necessarily fine (1.2mm), while the volume on the outside can be made more coarse (3.6mm). The exterior of this ‘BOI’ would have a much coarser mesh size of ‘10.8mm’.

This “Stepped Method” of meshing would prevent the creation of the bad quality elements as the size change not drastic across a single step. Even if bad quality elements are created, they would be away from the region of interest (enclosed by the inner jet ‘BOI’ with the fine mesh).

The number of elements was cut to less than half, about ‘4’ million elements.



(This tree shows the ‘stepped’ sizing for the jet ‘BOI’.)



(All Dimensions in 'mm'.)

Mesh Details:

Details of "Mesh"	
Display	
Display Style	Use Geometry Setting
Defaults	
Physics Preference	CFD
Solver Preference	Fluent
<input type="checkbox"/> Element Size	10.8 mm
Export Format	Standard
Export Preview Surface Mesh	No
Sizing	
Quality	
Inflation	
Advanced	
Statistics	
<input type="checkbox"/> Nodes	4152424
<input type="checkbox"/> Elements	4085590
Show Detailed Statistics	No

Multizone Method:

Details of "MultiZone" - Method	
Scope	
Scoping Method	Geometry Selection
Geometry	1 Body
Definition	
Suppressed	No
Method	MultiZone
Decomposition Type	Standard
Mapped/Swept Type	Hexa
Surface Mesh Method	Program Controlled
Free Mesh Type	Not Allowed
Element Order	Use Global Setting
Src/Trg Selection	Automatic
Src/Trg Scoping Method	Program Controlled
Source(s) and Target(s)	Program Controlled
Sweep Size Behavior	Sweep Element Size
<input type="checkbox"/> Sweep Element Size	Default
Element Option	Solid

Mesh Sizing: (Using 'Body of Influence' method.)

Details of "Nozzle_Sizing" - Sizing	
Scope	
Scoping Method	Geometry Selection
Geometry	1 Body
Definition	
Suppressed	No
Type	Body of Influence
Bodies of Influence	1 Body
<input type="checkbox"/> Element Size	0.4 mm
Advanced	
<input type="checkbox"/> Growth Rate	Default (1.2)

Details of "Inner_Fine_Jet_BOI" - Sizing	
Scope	
Scoping Method	Geometry Selection
Geometry	1 Body
Definition	
Suppressed	No
Type	Body of Influence
Bodies of Influence	1 Body
<input type="checkbox"/> Element Size	1.2 mm
Advanced	
<input type="checkbox"/> Growth Rate	Default (1.2)

Details of "Outer_Coarse_Jet_BOI" - Sizing	
Scope	
Scoping Method	Geometry Selection
Geometry	1 Body
Definition	
Suppressed	No
Type	Body of Influence
Bodies of Influence	1 Body
<input type="checkbox"/> Element Size	3.6 mm
Advanced	
<input type="checkbox"/> Growth Rate	Default (1.2)

This improved mesh was sent to the HPC with the same set-up, boundary conditions, solution methods and controls.

As of the moment of concluding the report, the simulation on the current mesh is still running. The higher number of mesh elements (around '4' million) coupled with the low courant number (0.1) would cause a very slow convergence. However, the refined mesh with it's 'stepped' sizing model should be able to capture all the flow complexities associated with such jets in its 3-dimensional entirety. Due to HPC scheduling conflicts and the time-intensive nature of such simulations, the report only documents the important intermediate steps for achieving convergence in such a challenging numerical simulation.

This report informs the reader on how to set-up the 3-dimeanional simulation of the cold flow jet of a 9-nozzle cluster to accurately characterise and study the shock diamonds, the shear layers and other flow intricacies.

Conclusion

This project focused on the computational modelling of a supersonic cold jet emerging from a nine-nozzle clustered rocket configuration, using a combination of analytical and numerical tools. The investigation began with a quasi-one-dimensional MATLAB-based analysis, which provided valuable baseline predictions for Mach number, temperature, and pressure variation along the nozzle and jet axis. These analytical results served as a benchmark for validating the subsequent CFD simulations.

The CFD workflow was designed as a stepwise development process to ensure physical reliability and numerical stability at each stage. Initially, multiple boundary condition configurations were tested to establish a well-posed problem. Early simulations revealed numerical oscillations in pressure and Mach number fields, which were effectively controlled by adopting a first-order upwind scheme and a low Courant number (0.1). Though first-order schemes are more diffusive by nature, this trade-off was acceptable at this stage to prioritize solver stability and minimize divergence.

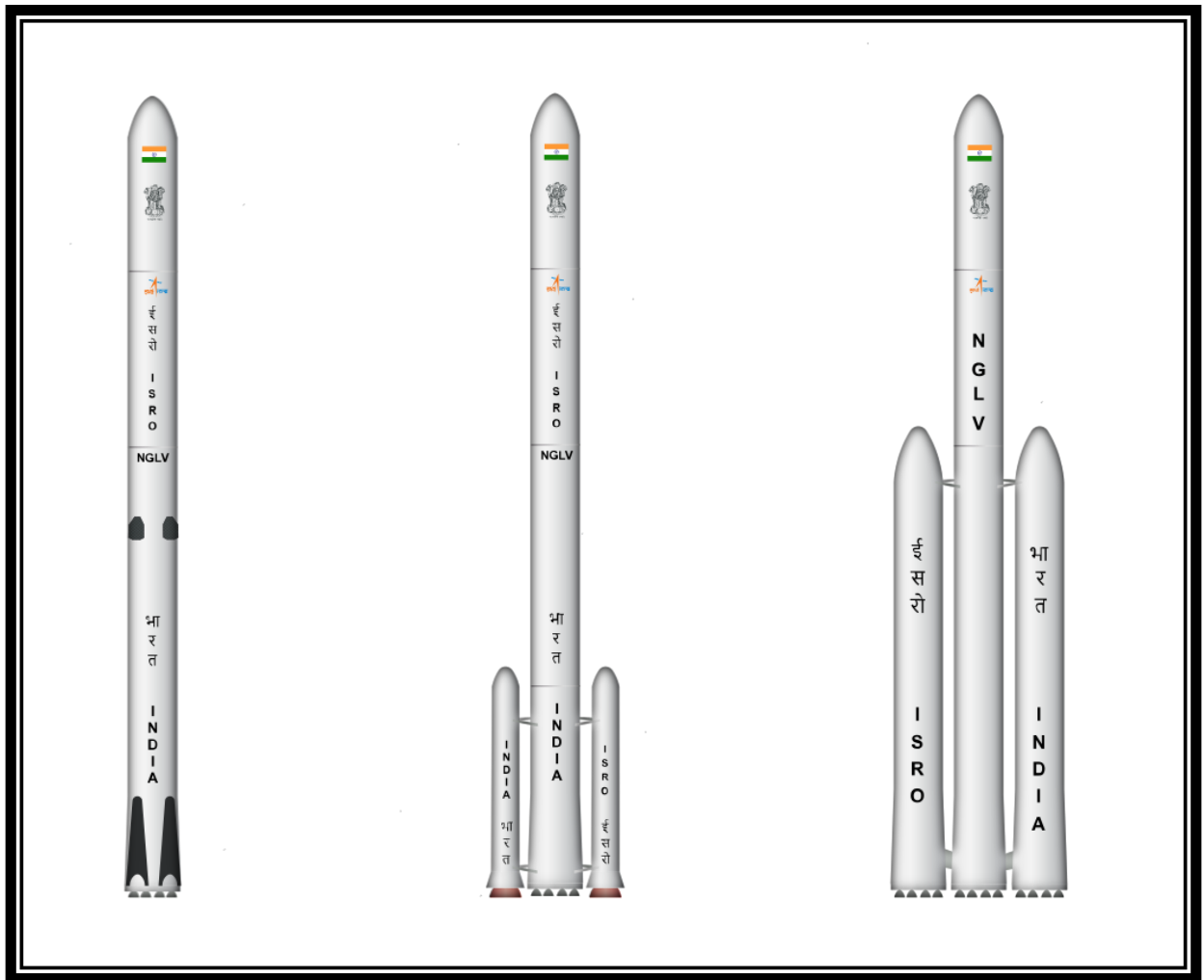
To counteract the smearing effects typically associated with first-order discretisation, a structured mesh using the “Multizone” method was employed. This meshing approach maintained orthogonality and better gradient resolution in critical regions such as the nozzle interior and near-field jet, helping preserve the essential flow features even under conservative numerical schemes.

Trial simulations were conducted across a range of mesh qualities—beginning with a coarse mesh, followed by a structured version of the same, and finally a finer “Multizone” mesh. These runs demonstrated that the simulation setup, particularly the chosen boundary conditions and thermodynamic definitions, produced physically consistent results within the nozzle and early jet region. Notably, the temperature and Mach number distributions matched well with the analytical model, validating the overall CFD strategy.

Some limitations did arise. The jet exhibited mild asymmetry in the ‘ZX’ plane, and the simulation lacked resolution of shock-cell structures typically seen in under expanded supersonic jets. These shortcomings are attributed to the limited mesh refinement and computational constraints. Due to time limitations and hardware issues, a final fully refined simulation could not be completed, though the groundwork has been thoroughly laid for future continuation.

Although higher-order discretisation schemes are typically employed to resolve fine flow features more accurately, this study opted to retain the first-order scheme throughout. This decision was intentional—made to prioritize numerical robustness while the mesh, boundary conditions, and solver behaviour were being tuned. The use of a structured “Multizone” mesh helped offset the increased numerical diffusion, ensuring that critical trends were still captured reliably.

In summary, this project successfully demonstrated a reliable simulation setup for a complex supersonic clustered nozzle system. While further refinement is required to capture detailed shock structures and achieve complete symmetry, the current methodology—validated against analytical results—establishes a strong foundation for future high-fidelity simulations.



(The NGLV family.)

References

- [1] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Corporation, 1980.

- [2] Computational Fluid Dynamics. In: ScienceDirect Topics, <https://www.sciencedirect.com/topics/materials-science/computational-fluid-dynamics> (accessed July 22, 2025).

- [3] J.D. Anderson, Modern Compressible Flow: With Historical Perspective, 3rd ed., McGraw-Hill, New York, 2003.

- [4] A. Krothapalli, L. Lourenco, J. Buchlin, Structure and mixing of a compressible turbulent jet, *J. Fluid Mech.* 246 (1990) 225–256. <https://doi.org/10.1017/S0022112090000058>

- [5] G.N. Abramovich, The Theory of Turbulent Jets, MIT Press, Cambridge, MA, 1963.