Warsaw University of Technology | Faculty of Power and Aeronautical Engineering

Modelling the airflow through rocket nozzle in OpenFOAM 8

Karol Gorbacevski

Warsaw 2022

Abstract

Exhaust gas and particle flow through nozzle has been a topic of research among propulsion engineers for the last century. It is one of the crucial elements when designing a rocket or jet engine. This paper analyzes how students, aerospace amateurs and engineers can analyze airflow through nozzle. To answer this question I used open source CFD software OpenFOAM. My results showed how using only limited computational power it is possible to model a compressible air flow through various nozzles.

Contents

0.1 Introduction 2

0.2 Numerical modelling 3

0.3 Pre processing and implementing the task in OpenFOAM 3

0.3.1 OpenFOAM 3

0.3.2 Geometry and mesh 3

0.3.3 Boundary conditions and solver settings 5

0.4 Calculations and results 5

0.5 Conclusions 9

0.6 References 10

0.1 Introduction

There are many problems that require the numerical solution of compressible flow in nozzles. One example of these problems is gas exhaust from plane or rocket nozzle. This report is devoted to numerical modelling of airflow through rocket jet nozzle. Propelling nozzle is placed in rocket engine to expand and accelerate combustion products to high supersonic velocities.

First scientific works on converging-diverging tubes started in late 18th century by Giovanni Battista Venturi and progressed in following centennial by German and Swedish engineers. Gustaf de Laval applied his own converging-diverging nozzle design for use in his impulse turbine in the year 1888. Later his nozzle was first applied in rocket design by Robert Goddard, American engineer and aerospace pioneer. Starting from cold-war era up to now, most modern rocket engines that employ hot gas combustion use de Laval nozzles. Most of the rockets today are reaching supersonic velocities that it is why it is our field of interest in this paper.

Sonic and supersonic flow conditions can only be attained when the critical pressure prevails at the throat, that is, when p2/p1 is equal or less than quantity defined below:

$$p_t/p_1 = [2/(k+1)]^{\frac{k}{k-1}} \tag{1}$$

At the location of critical pressure, namely the throat, the Mach number is always one. In the rocket propulsion systems supersonic nozzles are the subject of interest.The ratio of the inlet and outlet pressure in rocket nozzles must be design to be sufficiently large in order to induce supersonic flow. Another important factor for the successful rocket nozzle and simulation is choking factor. Which states that it is impossible to increase sonic speed at the throat or change the mass flow in nozzle by changing the exit pressure. Choking does take place in critical pressure area, namely the throat. From continuity:

$$\dot{m} = \frac{A_t v_t}{V_t} = A_t p_1 k \frac{\sqrt{[2/k+1]^{\frac{k+1}{k-1}}}}{\sqrt{kRT_1}} \tag{2}$$

Where $\dot{m}$ is equal to mass flow rate at any cross section within the nozzle. The ratio between throat and any downstream area for supersonic nozzles might be expressed as:

$$\frac{A_t}{A_y} = \frac{V_t v_y}{V_y v_t} = \left(\frac{k+1}{2}\right)^{\frac{1}{k-1}} \left(\frac{p_y}{p_1}\right)^{\frac{1}{k}} \sqrt{\frac{k+1}{k-1}\left[1 - \left(\frac{p_y}{p_1}\right)^{\frac{k-1}{k}}\right]} \tag{3}$$

Eq. (3) shows the inverse nozzle expansion ratio. For low altitude rockets ratio is typically between 3 and 30, for higher altitudes (100km and higher) the ratio is between 40 and 200, sometimes reaching 400.

0.2 Numerical modelling

Navier-Stokes equation for compressible gas/fluid in vector form is as follows:

$$\rho\left[\frac{\partial v}{\partial t} + (v \cdot \nabla)v\right] = -\nabla p + \mu\Delta v + (\zeta + \frac{1}{3}\mu)\nabla(\nabla \cdot v) + \rho f \qquad (4)$$

We use compressible flow due to the fact that velocities reach high numbers and incompressible flow is not accurate at this point. This equation can be solved by numerical methods. Simulating and understanding the high velocity turbulent flows is still a challenging problem. There are two methods to consider in numerical method: DNS or turbulence models. DNS uses very fine grids to resolve turbulent eddies directly. It requires massive amounts of computing power and therefore it is not efficient way to calculate turbulences. A turbulence model is derived from Navier-Stokes equations and tries to resolve only larger eddies. RANS and LES are the examples of these models.

0.3 Pre processing and implementing the task in OpenFOAM

0.3.1 OpenFOAM

OpenFOAM (for "Open-source Field Operation And Manipulation") is a C++ toolbox for the development of customized numerical solvers, and pre-/post-processing utilities for the solution of continuum mechanics problems, most prominently including computational fluid dynamics (CFD).

0.3.2 Geometry and mesh

In order to minimize the computing power and calculations needed I decided to build a geometry and respectively mesh as 2D-axisymmetry, as shown below:

Geoemtry was created in *Salome 9.7* software and divided into 7 groups. Whole geometry's width is 0.5 m and height 0.3 m. Exact spline vertex points for nozzle can be found in text document: 'nozzle jet.txt'. Inverse nozzle expansion is set to be between 3 and 30. The beginning of global coordinate system is in left bottom corner of the combustion chamber/nozzle, as shown on the figure above.

To create a mesh I also used the *Salome 9.7* software. The geometry was divided into 7 mesh groups: Inlet, outlet1, outlet2, outlet3, wall1, wall2, nozzleWall.

Surfaces defined by me are shown on a figure above as follows: inlet - green; nozzleWall - red; outlet1 - yellow; outlet2 - light blue (top surface); outlet3 - pink; wall1 - blue; wall2 - orange. The mesh consists of rectangles and is not very fine. The mesh is fined in nozzle's throat for more accurate results.

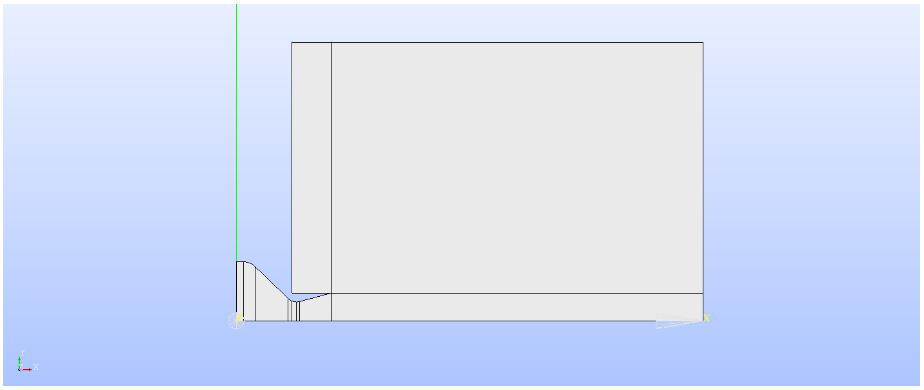

Figure 1: 2D axisymmetrical geometry created

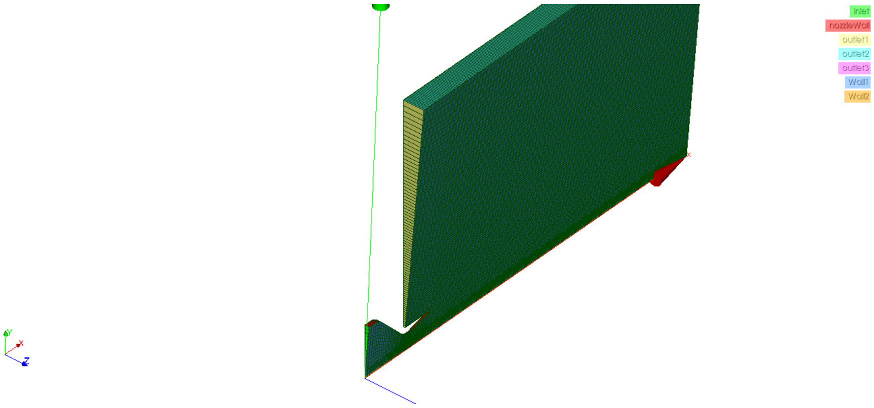

Figure 2: Mesh created in Salome 9.7, iso-view

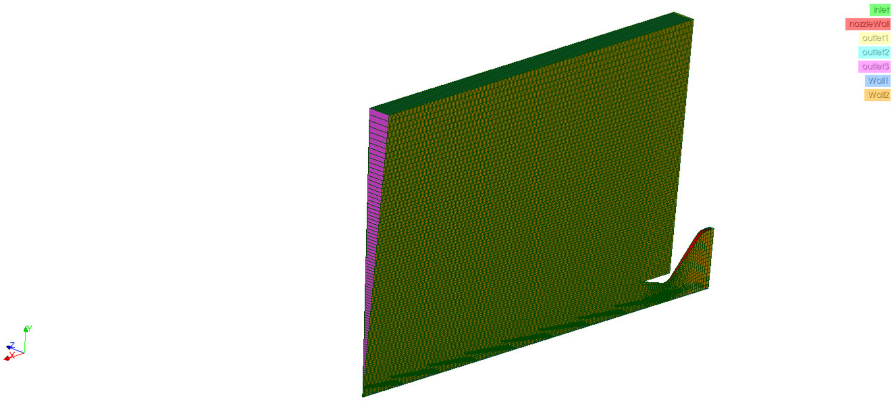

Figure 3: other side of the mesh, iso-view

0.3.3 Boundary conditions and solver settings

Boundary conditions are set in *0* folder. There are two main attributes, p and U to run calculations. In this simulation we do not impose any inlet velocity.

In terms of pressure it is important to keep in mind that ratio between the inlet and outlet pressures in all rocket propulsion systems must be designed large in order to achieve supersonic velocities. Only when absolute pressure in chamber drops below approximately 1.78 atm there will be subsonic flow in the divergent part of the nozzle (sea - level). That is why in our simulation we proceed with the pressure ratio equal to 10:1 in order to build a relatively stable supersonic flow.

Solver can be found in folder *systems* in file *controlDict*. For compressible simulations appropriate solver is rhoPimpleFoam. Material and chemical abilities of gas are set for typical air. Writing interval is set to 0.0001, start time to 0, $\Delta t = 1e-06$ and end time to 0.01.

0.4 Calculations and results

After few hours of calculations finally the result showed up. In order to make more accurate calculations the mesh was refined by using function *refineMesh* and afterwards the function *mapFields* was used in order to make computations faster. Mesh was refined by the factor of 2. *MapField* does take already solved existing computations and map those on a more fined mesh. It is an efficient way to calculate complex and fined meshes firstly by counting the simple meshes and step by step fining and mapping on already solved cases. Otherwise if we jump straight forward to calculating very fined meshes it will take hours or even days to complete iterations. Moreover there is a risk that model was set incorrectly and all time spent calculating is wasted. By using *ParaFOAM* function we can visualize the results in *ParaView 5.6.0* software. The contours for pressure, temperature and Mach numbers are shown below:

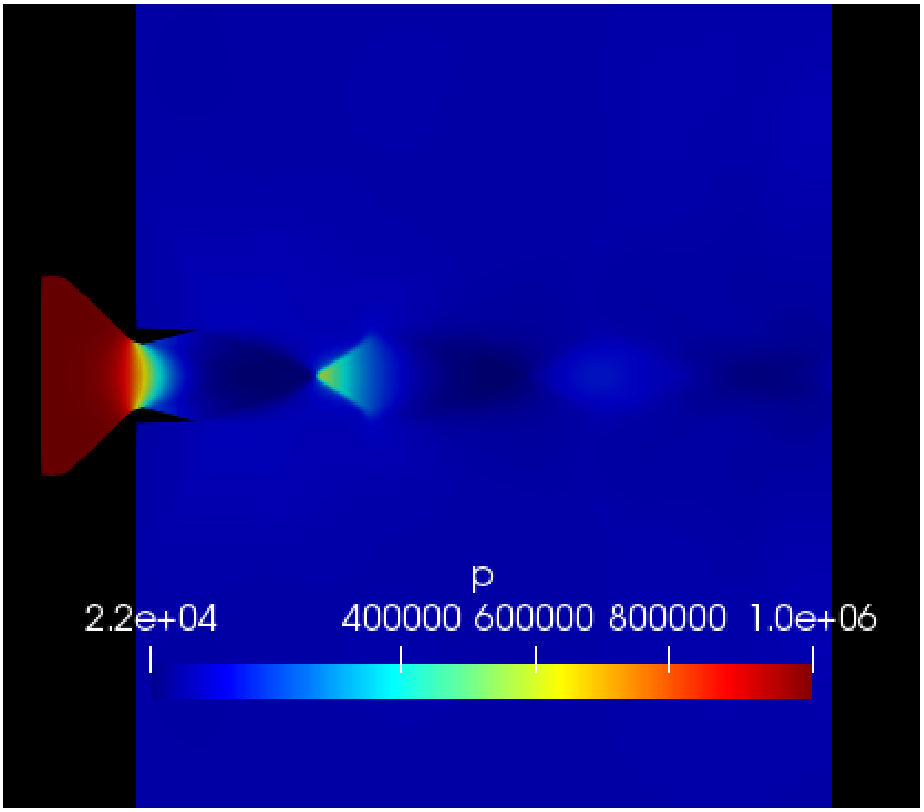

Figure 4: Pressure results, [Pa]

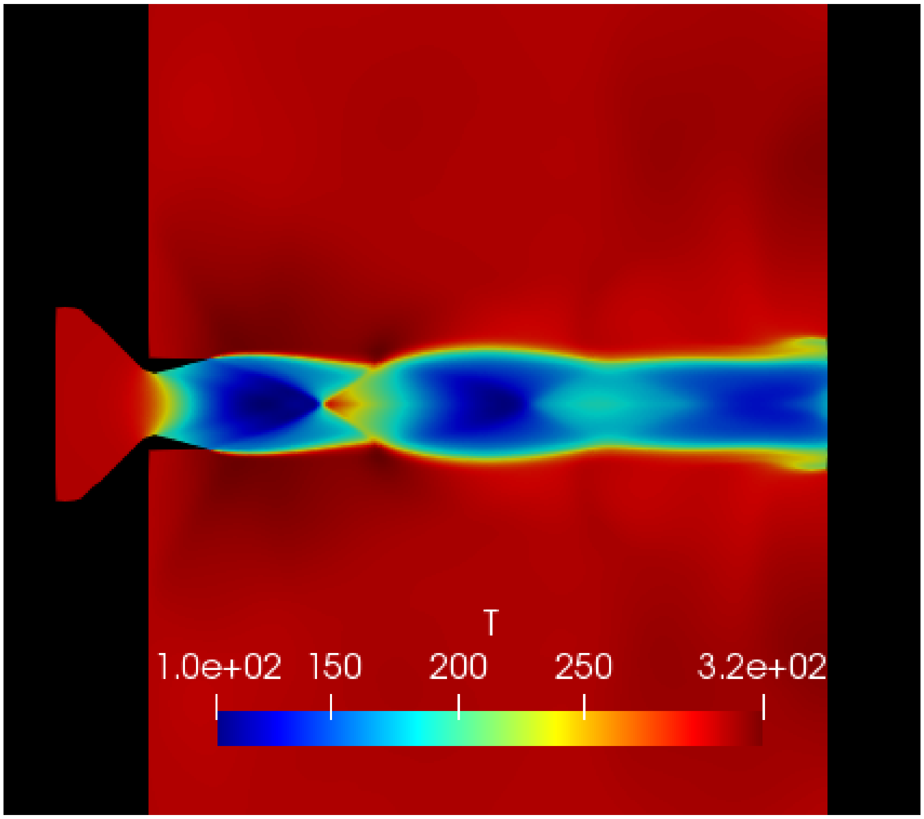

Figure 5: Temperature results, [K]

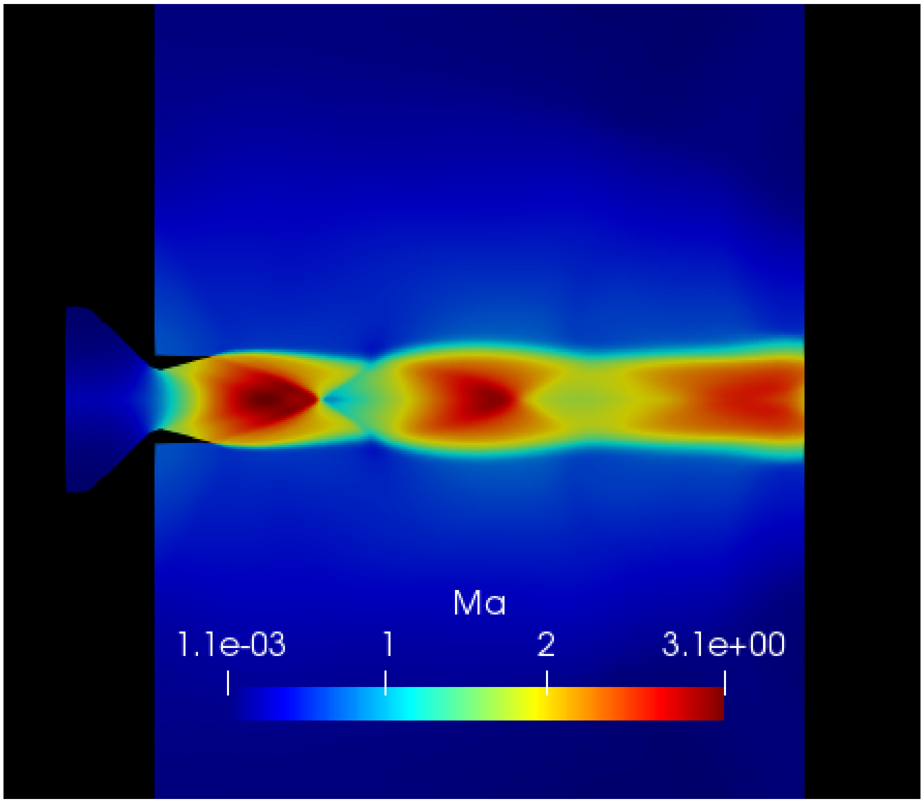

Figure 6: Mach number results

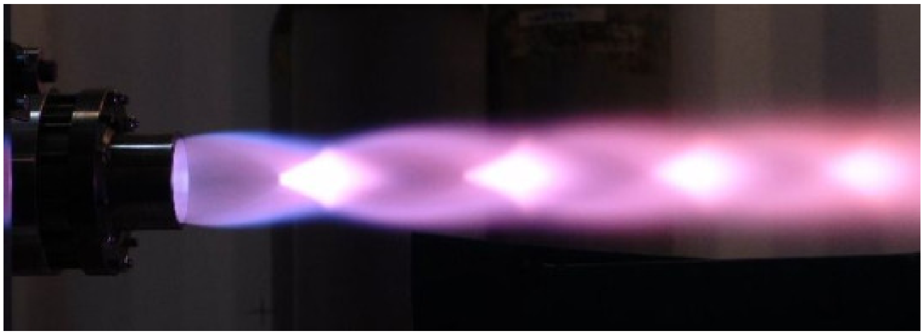

Figure 7: Shock diamonds also called Mach diamonds during engine test

We can observe the increase in a velocity outside the throat area, thus proving us the formulas we use in rocket science and that it is a supersonic nozzle. The velocity is reaching the Mach 3 in certain places. As it was expected from Bernoulli law, the decrease in pressure and temperature after the throat area is also highly visible. The flow is typical for rockets and jet engines. Animation showing flow is uploaded and placed in my repository. In the animation we can observe the turbulences in the flow. It is caused by an unstable flow. In order to improve it we need to run calculations for a longer period of time and let the flow stabilise. After stabilization the Mach diamonds will be seen very clearly.

0.5 Conclusions

OpenFOAM can serve as a good tool to simulate and visualize the flow of gases in nozzle. For rocketry engineers as well as engineers dealing with jet engines it is important to check the possible outcome of exhaust gases and quality of flow. These tests may lead to change in geometry, initial chamber pressure or many other specifications in order to make a rocket more efficient and safe. OpenFOAM deals greatly with this task and as we can see with limited computational power it is possible to simulate compressible flows. As listed above, in section *Calculations and results*, refining the mesh after initial low grid calculations and then mapping the results of low grid on the fined mesh is an efficient way to calculate fine meshes. Unfortunately OpenFOAM has its limits and for a big projects (ex. *heat exchange and flow in Raptor rocket engine*) aerodynamic tunnels and more sophisticated software may be needed.

0.6 References

1. C. J. Hwang and G. C. Chang, *Numerical study of gas-particle flow in a solid rocket engine*

2. George P. Sutton and Oscar Biblarz, *Rocket propulsion elements ninth edition*

3. `https://en.wikipedia.org/wiki/De_Laval_nozzle`

4. Philipp Birken, *Numerical methods for the unsteady compressible Navier-Stokes equations*

5. https://www.openfoam.com/documentation/guides/latest/doc/guide-applications-solvers-compressible-rhoPimpleFoam.html

6. Anuar Kagenov, *How to install ParaView, Salome, Ubuntu wsl2, OpenFOAM in Windows 10*
   https://www.youtube.com/watch?v=bRnbrdjuXm4

7. Anuar Kagenov, *Simulation of Free Supersonic Jet using OpenFOAM*
   https://www.youtube.com/watch?v=JG7AvESdZq8