

# Supersonic nozzle flow

---

By Louenas Zemmour

[louenas.zemmour@etu.sorbonne-universite.fr](mailto:louenas.zemmour@etu.sorbonne-universite.fr)

How can we predict  
and visualize a  
supersonic nozzle  
flow in OpenFOAM ?

# What challenges ?

A rigorous and logical path must be followed and will be further detailed

- Geometry
  - Inviscid
  - Viscous case
  - Conclusion
-

# Introduction



# Why ?

Well why not ?

- Physical phenomena involved are of critical importance
  - Few tutorials/litterature using OpenFoam
  - Open-source tool
-



# Inviscid flow



# Solvers and boundary conditions

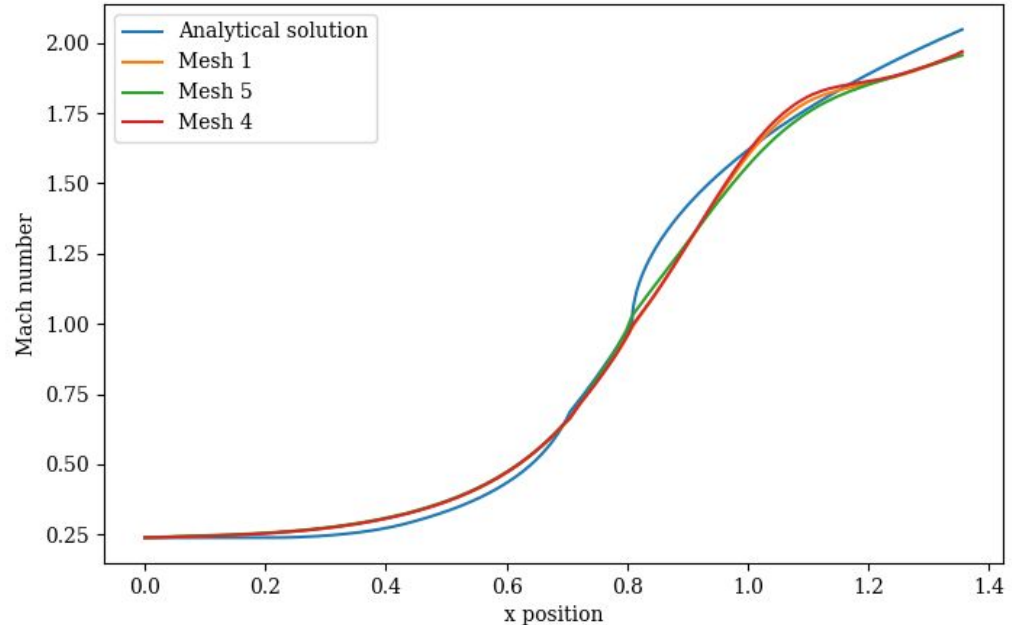
1. SonicFoam, choice based on data from literature.
2. Smooth solver for pressure, velocity and energy
3. PreConditioned Conjugate Gradient (PCG) for density
4. Inlet BC: total pressure and temperature.  
ZeroGradient velocity
5. Far field BC: waveTransmissive for p, U, T
6. Ambient pressure and temperature: standard  
atmosphere at sealevel

```
17 solvers
18 {
19     "p.*"
20     {
21         solver      smoothSolver;
22         smoother     symGaussSeidel;
23         tolerance    1e-06;
24         relTol       0;
25     }
26
27     "(U|e)[.]*"
28     {
29         $p;
30         tolerance    1e-05;
31         relTol       0;
32     }
33
34     "rho.*"
35     {
36         solver      PCG;
37         preconditioner DIC;
38         tolerance    1e-05;
39         relTol       0;
40     }
41 }
42
```



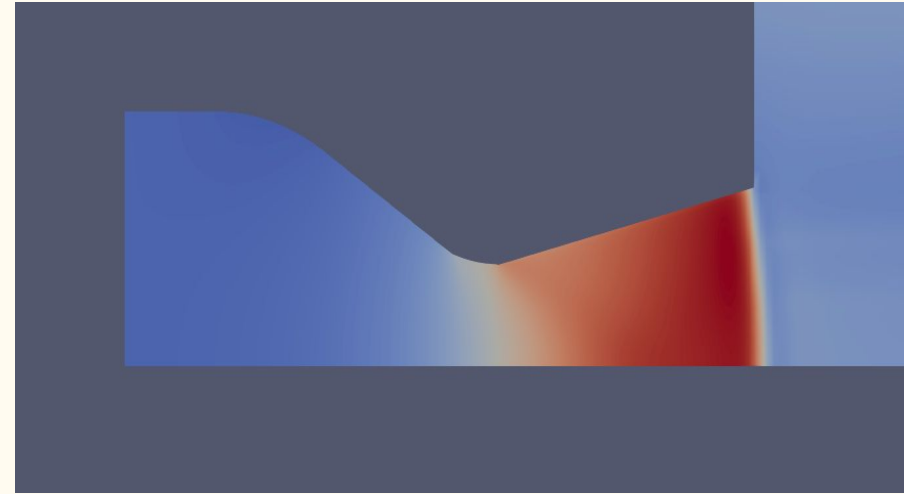
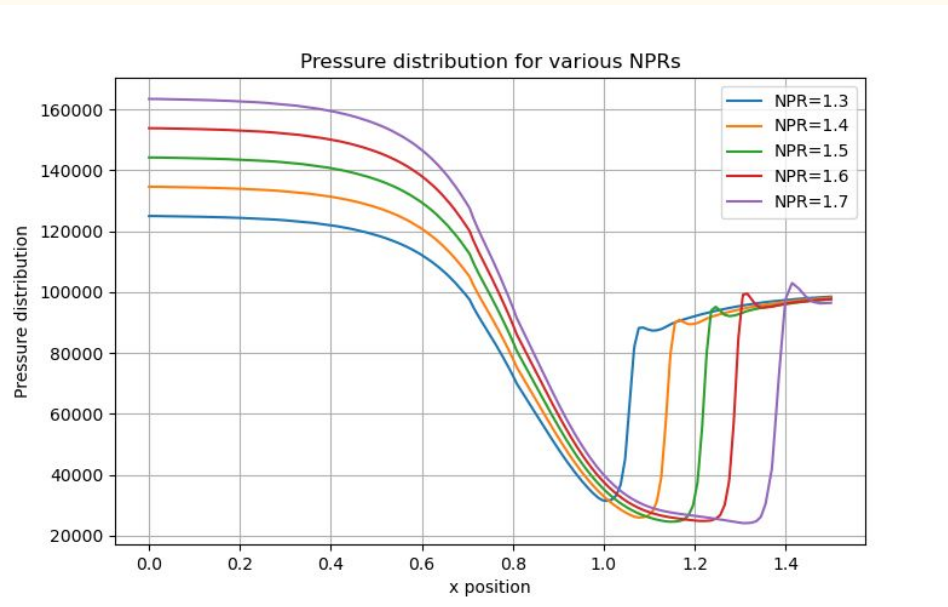
# Mesh convergence and validation

1. Simulation for an adapted nozzle were compared with the analytical solution given by the quasi 1D Euler model
2. Mesh chosen has 35234 points



# Nozzle Pressure Ratio

1. Defined as the ratio between the total and static pressure at the exit



# Viscous case

---

Insert a turbulence  
model to visualize  
shockwave/boundary  
layer interactions

# How ?

Following basic CFD good mores

- Turbulence model and boundary conditions
- The  $y^+$  and first cell size
- Time integration
- Results

---

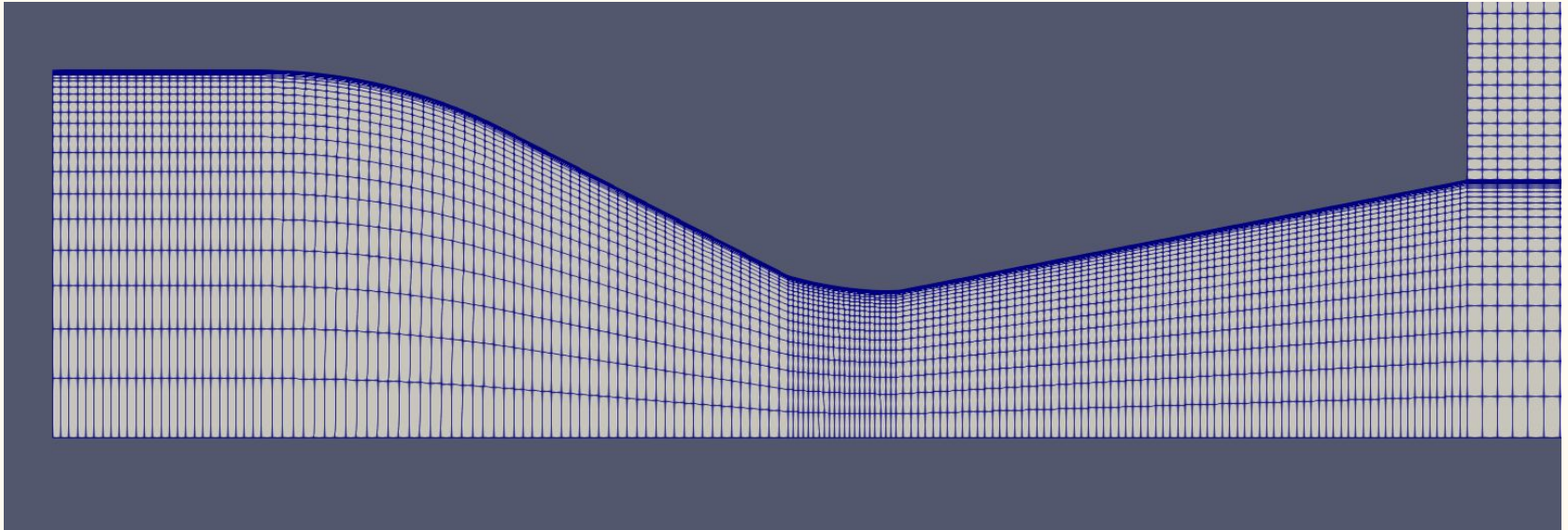
# Turbulence model

1. We need a model that gives good results at walls, the  $k$ - $\omega$  SST is the best suited for this case.
2. The initial conditions for the TKE and the specific dissipation rate are estimated according to the relations given in the OpenFoam user guide page.
3. Boundary conditions are zero uniform value for  $k$  and calculated value for  $\omega$

```
21 boundaryField
22 {
23     inlet
24     {
25         type            turbulentIntensityKineticEnergyInlet;
26         intensity        0.05;
27         value            uniform 5;
28     }
29
30     "(outlet|farfield)"
31     {
32         type            inletOutlet;
33         inletValue       uniform 5;
34         value            uniform 5;
35     }
36
37     symmetry
38     {
39         type            symmetryPlane;
40     }
41
42
43     upperWall
44     {
45         type            uniformFixedValue;
46         uniformValue     0;
47     }
48 }
```

# $y^+$ and the first cell size.

1. To capture the boundary layer and shock wave interactions we have to simulate the viscous sublayer and the  $y^+$  must be of order 1
2. The mesh was refined and the final maximum  $y^+$  on the wall was: 0.4 which is a very good result.



# Time integration

1. In order to study the unsteadiness of such a configuration, the semi-implicit Crank-Nicolson scheme has been used to allow for higher CFL numbers and bigger time steps.
2. The schemes and time step values are to be found in respectively the fvSchemes and ControlDict files in the system directory.

```
16
17 ddtSchemes
18 {
19     default          CrankNicolson 0.8;
20 }
21
22 gradSchemes
23 {
24     default          Gauss linear;
25 }
26
27 divSchemes
28 {
29     default          none;
30
31     div(phi,U)       Gauss limitedLinearV 1;
32
33     div(phiid,p)      Gauss limitedLinear 1;
34     div(phi,e)        Gauss limitedLinear 1;
35     div(phi,K)        Gauss limitedLinear 1;
36     div(phiiv,p)      Gauss limitedLinear 1;
37
38     turbulence        Gauss upwind;
39     div(phi,k)         $turbulence;
40     div(phi,omega)     $turbulence;
41
42     div((((rho*nuEff)*dev2(T(grad(U)))))) Gauss linear;
43 }
```



Results:



# Discussion

## On the numerical simulations

- OpenFoam is a very performant opensource software for CFD. SonicFoam provide relatively accurate results and is shock capturing.
- More detailed data about time convergence will be provided in the report

## Variables that affected the outcome

- Mesh that could have been finer
- However, the simulations were already quite costly and the lack of numerical ressources affected significantly the results.

# Conclusion

We have been able to provide a basis to predict and visualize supersonic nozzle flow using OpenFoam. More investigation must be held regarding the different compressible solvers, schemes and boundary conditions.

Also more physical analysis have to be made regarding the results and compared with data within the literature.

# Thank you for your attention !

Python codes, scripts and data files can be found on:

[https://github.com/LouenasZm/supersonic\\_nozzle](https://github.com/LouenasZm/supersonic_nozzle)

Or by scanning the following QR code.

The OpenFOAM case files will be added to the repository as soon as possible.

