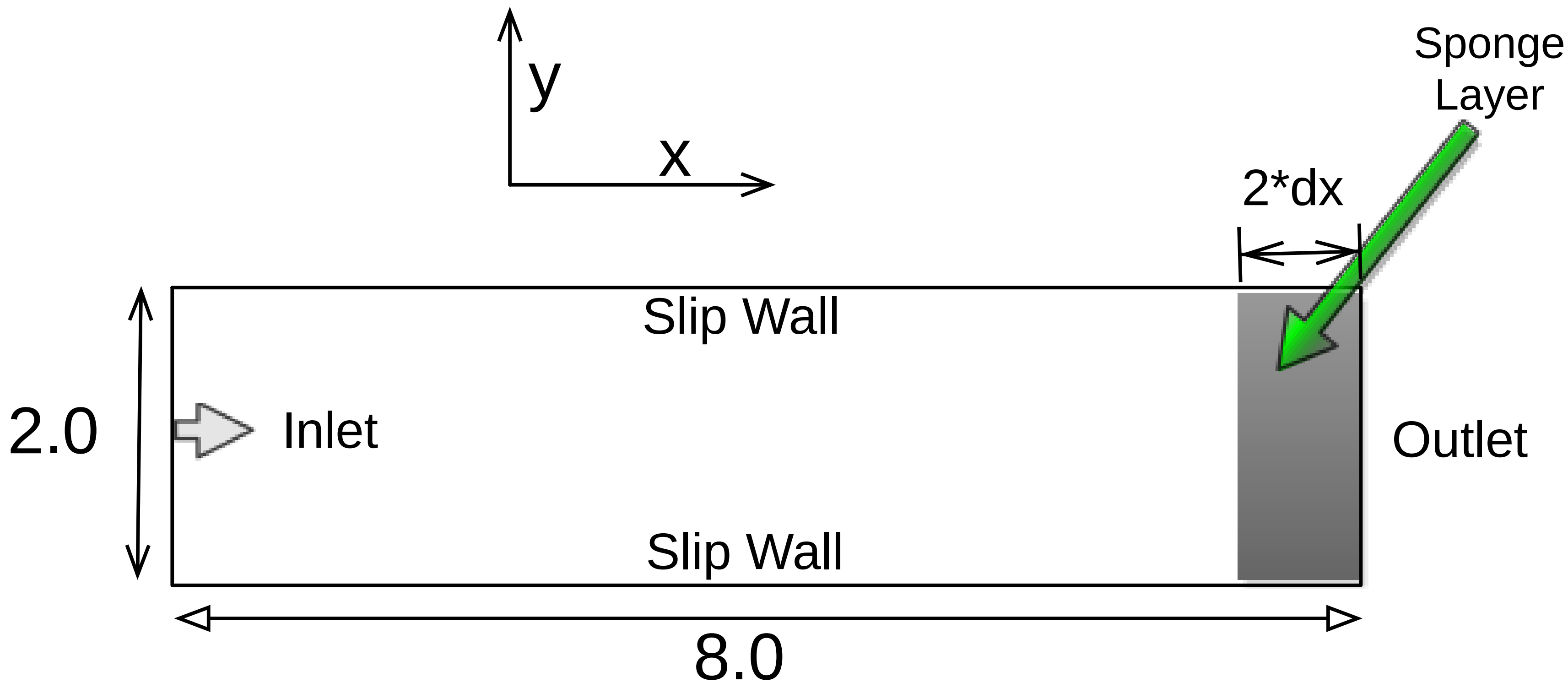
Subsonic jet

Post-processing and Visualization

Outlook

- Testcase –Subsonic jet
- Post-processing
- Visualizing the results
- Conclusion

Subsonic Jet



Jet radius = 0.1

Varying Parameters

Name	Total Number of elements[1]	Order of the scheme (m+1)[2]	Total degrees of freedom
finest	65280	2	1,044,480
fine	4032	8	1,032,192
coarse	240	33	1,045,440

1 - Total number of elements can be varied with varying the level parameter in the mesh input file

2 - Order of the scheme can be set my the variable “m” (max. poly degree) in the solver input file

Evaluating High Order Results

- How to visualize the polynomial solution after the simulation?
- Need to produce a representation, that can be understood by the visualization tool
- We voxelize the information within the elements and provide a finer resolved mesh to represent the variation within elements

Controlling the Number of Voxels

- The number of voxels to use per element is configured by the ply_sampling table:

```
ply_sampling = {
  nlevels = 3,
  method = 'fixed'
}
```

- nlevels refers to the refinement of elements by bisection in each direction (factor of 8 elements per level)

Exercise

- Post process all restart files obtained from the last simulation into VTU file format suitable for visualization.
- We set the number of voxels to be constant for all the three test cases, resulting in the same resolution.
  - This is done by setting the subsampling level (as shown in the last slide)
- Visualize the output using Paraview

Post-processing multiple restart files with Ateles-Harvesting

- We use a script, which invokes multiple independent instances of Atl-Harvesting. The script requires –
  - A template harvester input-file
  - The path to the input restart files
  - The desired output folder
  - The path to the Atl-Harvesting executable
  - mpiexec option, if you want to run it in parallel
- All these options can be set in the input file (series.config)

Setting up the config file for using harvester script (series.config)

- `template: harvest_series.template`
- `files: restart/*.lua`
- `lua: /shared/akad-cfd-s/bin/lua`
- `harvester: /shared/akad-cfd-s/bin/atl_harvesting`
- `out: output`
- `run: mpirun –n 10`

Template harvest_series file

- Here as a basic user it is sufficient to set the subsampling and provide information about the solver input file
- Set up the template Atl-Harvesting file (harvest_series.template)
  - `require('ateles')`
  - `ply_sampling = {`

```
    nlevels = 1, --for finest mesh
    -- nlevels = 3, --for fine mesh
    -- nlevels = 5, --for the coarse
    method = 'adaptive'
  }
```

Template harvest_series file

```
tracking = {
  {
    label = 'visu',
    variable = {
      'density', 'pressure',
      'mach_number', 'velocity'
    },
    shape = { kind = 'global' },
    folder = '$!out!$',
    output = { format = 'vtk' }
  }
}
```

Post-Processing

- The steps for post-processing are provided in visualize.job
- Please adapt that job script
- It makes use of the harvest_series script:

```
python3 $KURS/bin/harvest_series.py ↲
  -c series.config
```

Workflow

- Go to the directory, where you set up your run

```
cd $MYWS/hpcfdx8/mysetup
```

- Copy the script files into the current folder

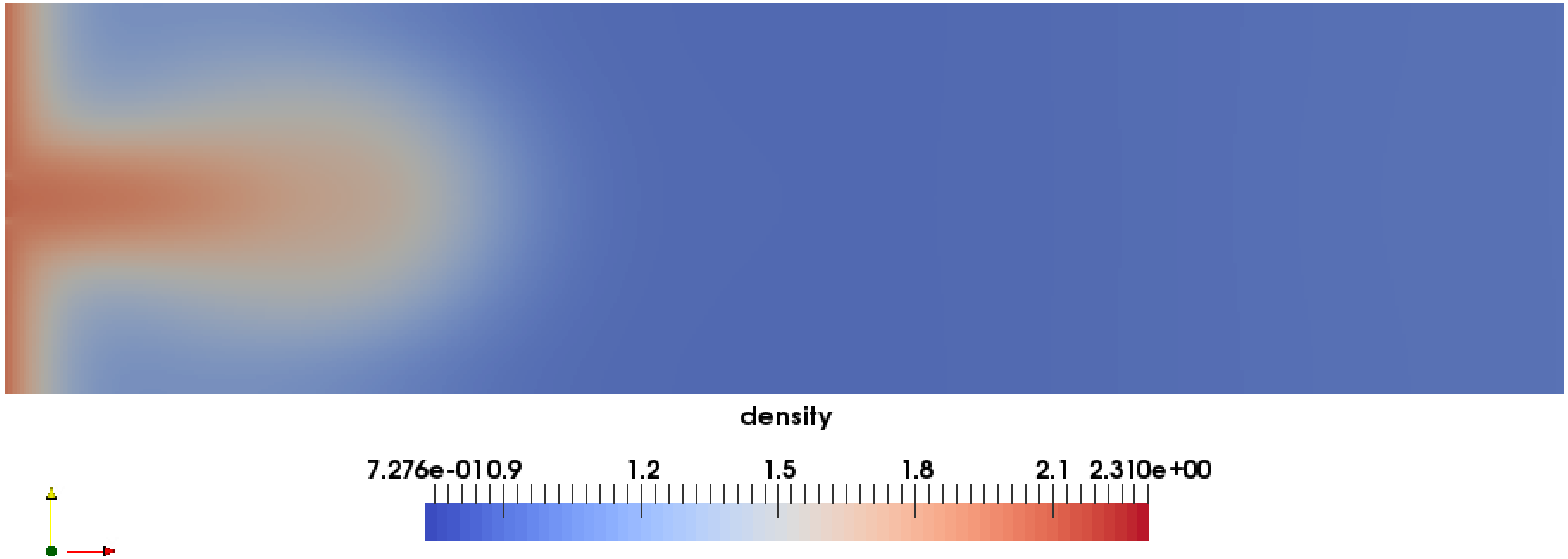
```
cp ../harvest_series.template ↵
  ../series.config .
```

- Set up the parameter file (series.config) for the harvest_series script
- Make changes to the harvest_series template file
- Execute the harvest_series script

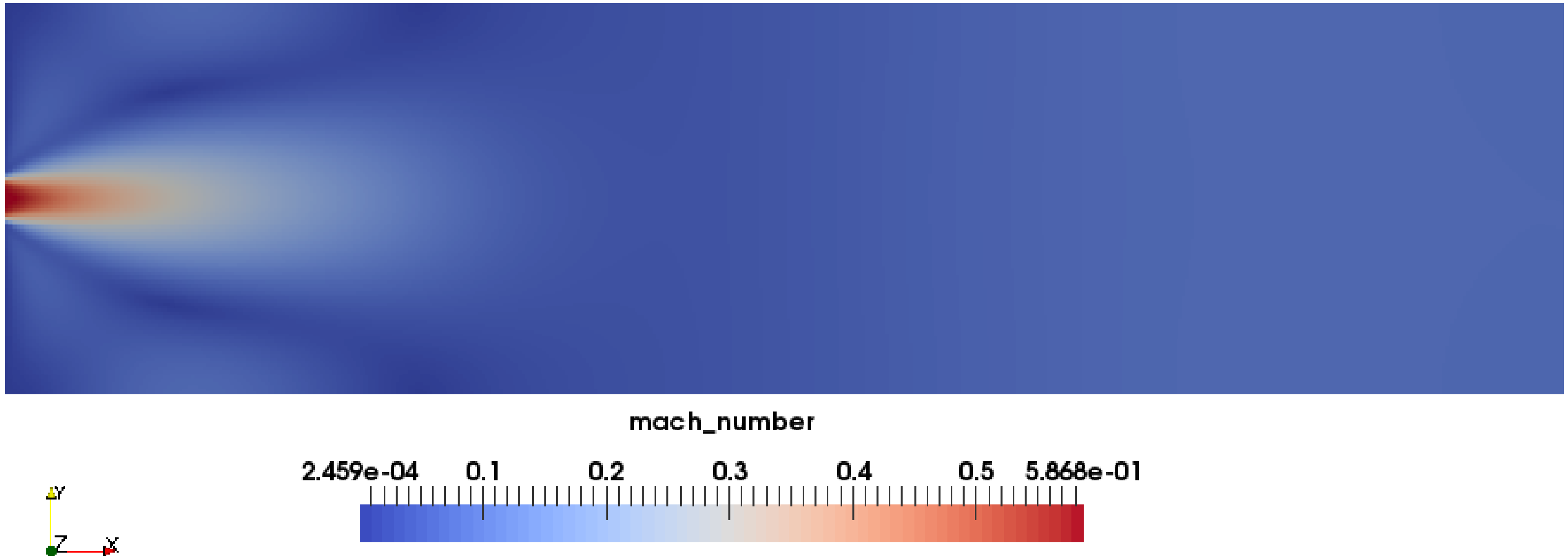
Workflow (Visualizing the results)

- Open Paraview as described on https://geb.inf.tu-dresden.de/hpcfd/pages/Visualisierung.html
- Go to the folder to be visualized and open the "jet_*.pvd" file
- Play around with Paraview, by visualizing the simulation results at different time-steps

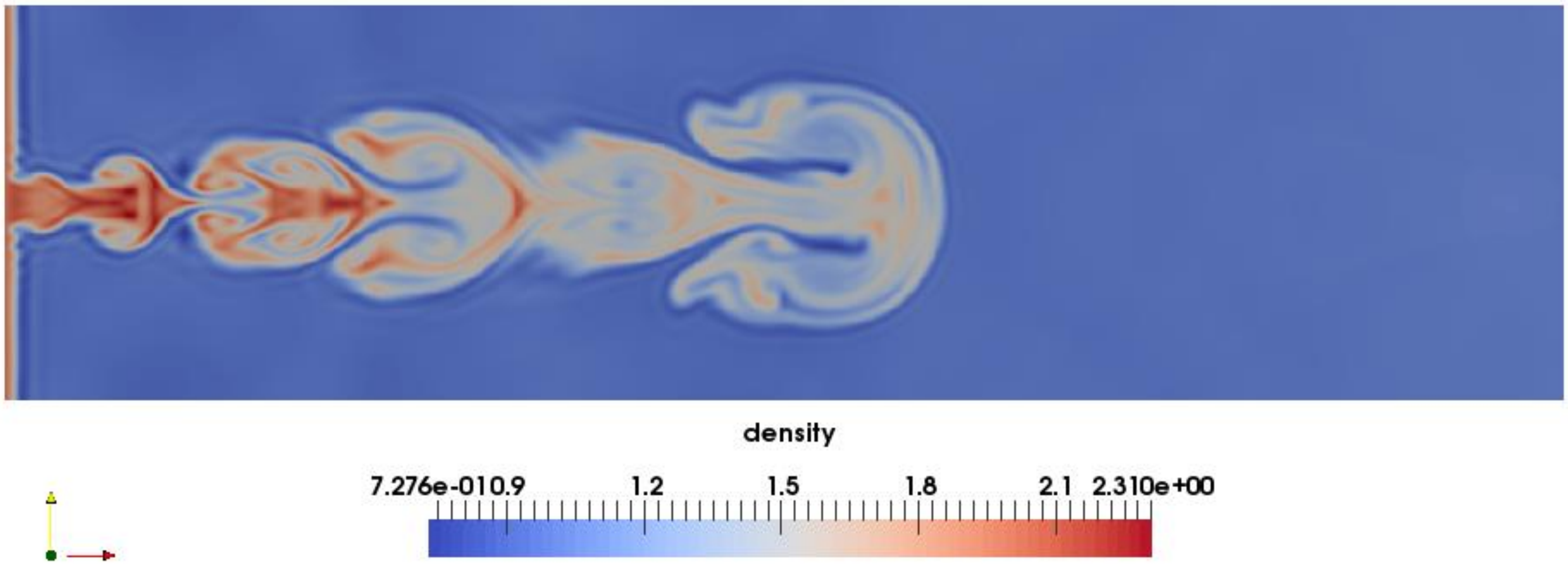
Testcase: finest mesh (instantaneous density)



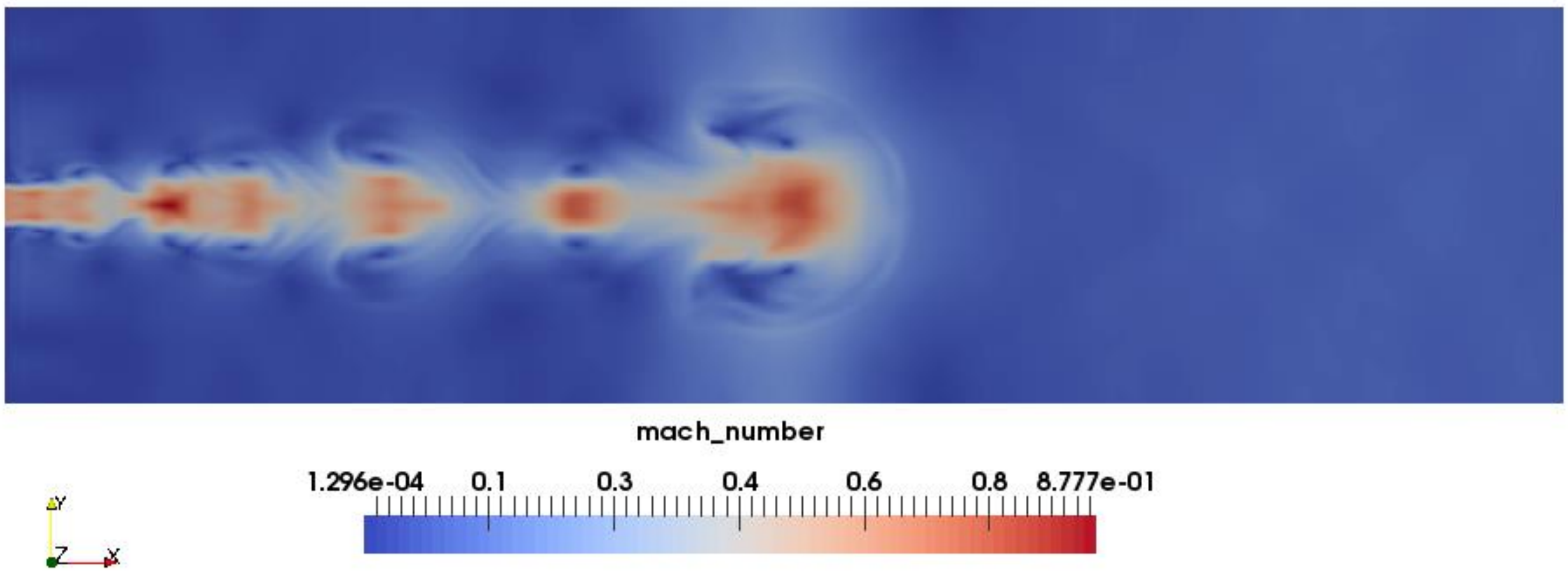
Testcase: finest mesh (instantaneous Mach number)



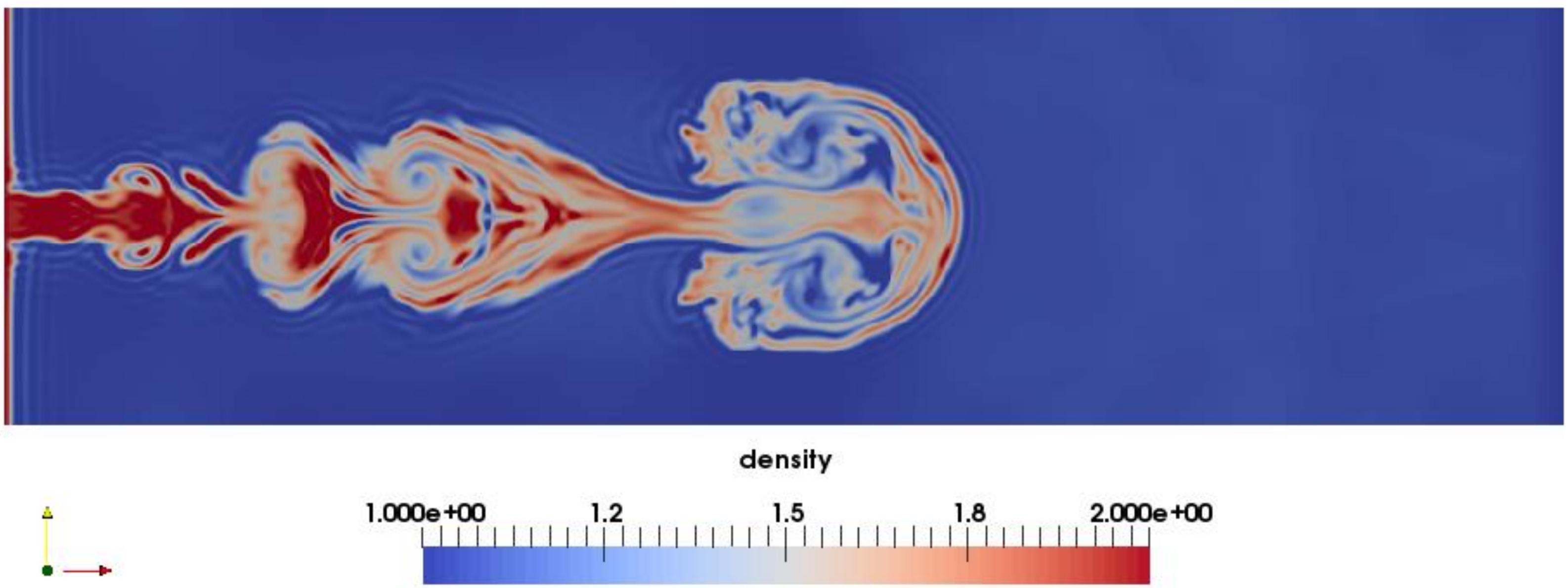
Testcase: fine mesh (instantaneous density)



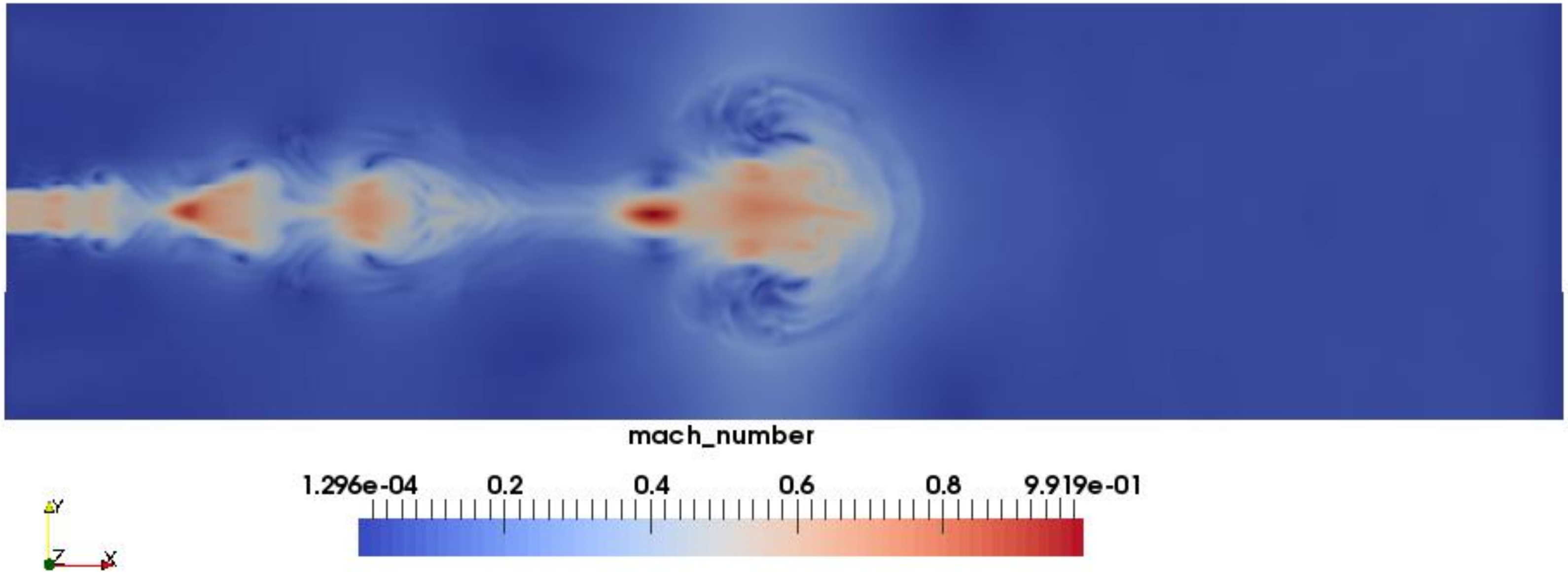
Testcase: fine mesh (instantaneous Mach number)



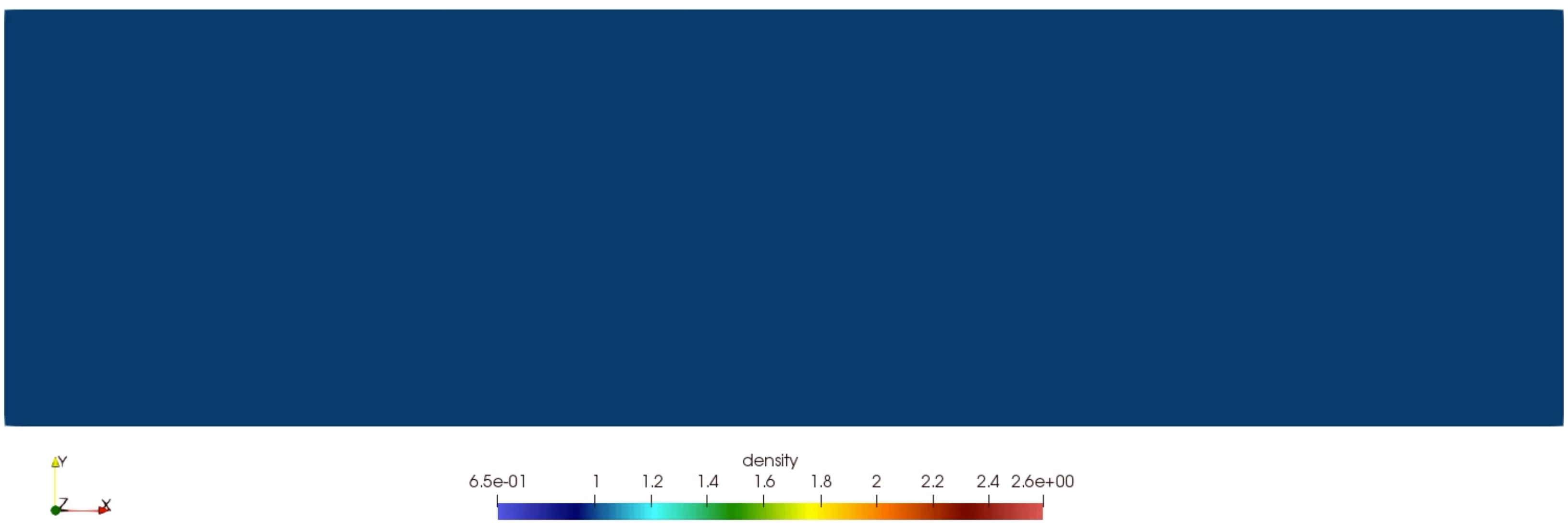
Testcase: coarse mesh (instantaneous density)



Testcase: coarse mesh (instantaneous Mach number)



Testcase: coarse mesh



Conclusion

- Solving same number of degree of freedoms – refinement in **p** gives more detailed profiles when compared to refinement in **h**
- With increase in **p**, the time-step size becomes smaller and more computational effort is required

$$\Delta t < C_{cfl} \frac{h}{p^2}$$