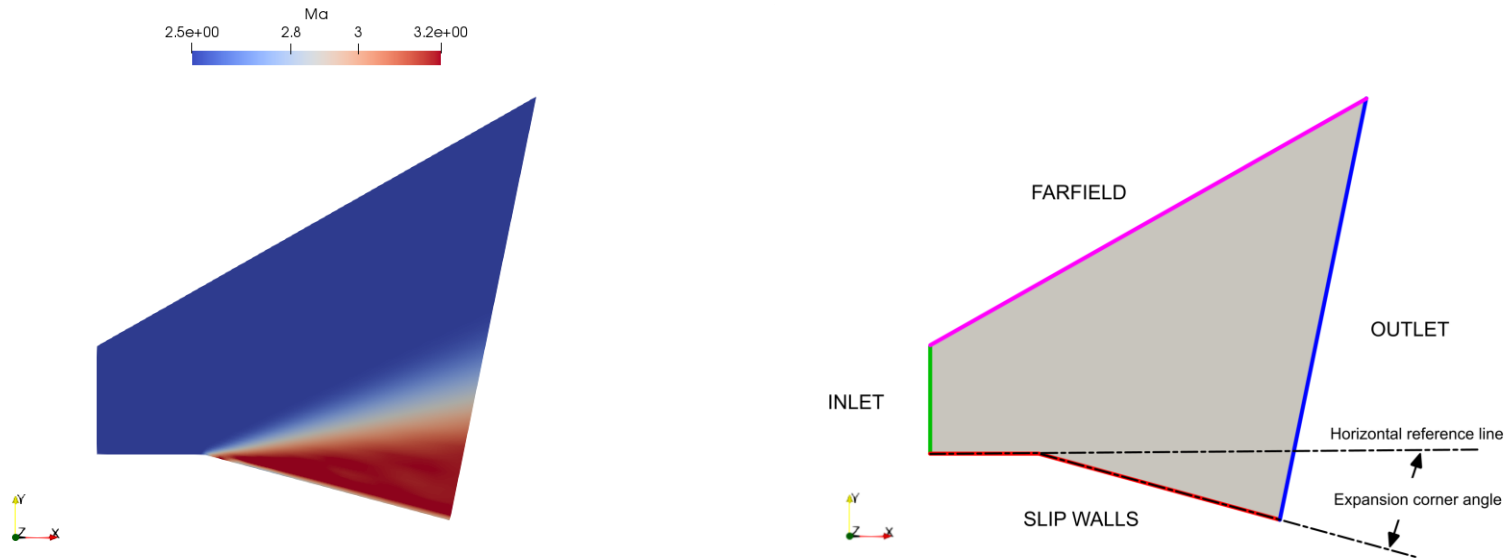


High speed compressible flow – Expansion fan

Expansion corner – Expansion fan



Physical and numerical side of the problem:

- In this case, we simulate a flow past by an expansion corner.
- The expansion corner is inclined 15 degrees below the horizontal line and the inlet Mach number is equal to 2.5.
- We are going to use a steady compressible solver with zero viscosity.
- Therefore, the governing equations of the problem are the compressible Euler equations.
- This problem has an analytical solution and plenty of validation data.

High speed compressible flow – Expansion fan

- Let us run this case. Go to the directory:

```
$PTOFC/expansion_fan
```

- \$PTOFC is pointing to the directory where you extracted the training material.
- In the case directory, you will find the `README.FIRST` file. In this file, you will find the general instructions of how to run the case. In this file, you might also find some additional comments.
- You will also find a few additional files (or scripts) with the extension `.sh`, namely, `run_all.sh`, `run_mesh.sh`, `run_sampling.sh`, `run_solver.sh`, and so on. These files can be used to run the case automatically by typing in the terminal, for example, `sh run_solver`.
- We highly recommend you to open the `README.FIRST` file and type the commands in the terminal, in this way, you will get used with the command line interface and OpenFOAM® commands.
- If you are already comfortable with OpenFOAM®, use the automatic scripts to run the cases.

High speed compressible flow – Expansion fan

What are we going to do?

- We will use this case to learn how to setup supersonic flow cases.
- We will compare the numerical solution with the analytical solution.
- We will run the case with a robust numerics, but you are invited to try different setups and compare the different outcomes.
- To find the numerical solution we will use the solver `rhoSimpleFoam` with zero viscosity (Euler equations).
- `rhoSimpleFoam` is a steady solver for turbulent flow of compressible fluids.
- After finding the numerical solution we will do some sampling.
- Then we will do some plotting (using gnuplot or Python) and scientific visualization.

High speed compressible flow – Expansion fan

Running the case

- Let us first convert the mesh from Fluent format.
- In the terminal window type:

1. `$> foamCleanTutorials`
2. `$> rm -rf 0 > /dev/null 2>&1`
3. `$> fluentMeshToFoam -2D 1 ./mesh/ascii.cas`
4. `$> checkMesh`

- If you want to generate the mesh using `blockMesh`, in the directory **system** you will find the dictionary `blockMeshDict`, which has been already parametrized.
- In this case we are using a medium mesh and the wedge angle is 15 degrees.
- If you want to try different meshes or wedge angles, feel free to modify the dictionary `blockMeshDict`.

High speed compressible flow – Expansion fan

Selecting thermophysical properties

```
1 thermoType
2 {
3     type          hePsiThermo;
4     mixture       pureMixture;
5     transport     const;
6     thermo        hConst;
7     equationOfState perfectGas;
8     specie        specie;
9     energy        sensibleEnthalpy;
10 }
11
12 mixture
13 {
14     specie
15     {
16         nMoles      1;
17         molWeight   28.9;
18     }
19     thermodynamics
20     {
21         Cp          1005;
22         Hf          0;
23     }
24     transport
25     {
26         mu          0.0;
27         Pr          0.713;
28     }
29 }
```

- The thermophysical properties are set in the dictionary *thermophysicalProperties*.
- This dictionary file is located in the directory **constant**.
- In the sub-dictionary **thermoType** (lines 1-10), we define the thermophysical models. Many of these options are hardwired with the solver used.
- The **transport** keyword (line 5) concerns evaluating dynamic viscosity. In this case the viscosity is constant.
- The thermodynamic models (**thermo** keyword) are concerned with evaluating the specific heat Cp (line 6). In this case Cp is constant.
- The **equationOfState** keyword (line 7) concerns to the equation of state of the working fluid. In this case, we are using the ideal gas equation model.

$$\rho = \frac{p}{RT}$$

High speed compressible flow – Expansion fan

Selecting thermophysical properties

```
1 thermoType
2 {
3     type          hePsiThermo;
4     mixture       pureMixture;
5     transport     const;
6     thermo        hConst;
7     equationOfState perfectGas;
8     specie        specie;
9     energy        sensibleEnthalpy;
10 }
11
12 mixture
13 {
14     specie
15     {
16         nMoles      1;
17         molWeight   28.9;
18     }
19     thermodynamics
20     {
21         Cp          1005;
22         Hf          0;
23     }
24     transport
25     {
26         mu          0.0;
27         Pr          0.713;
28     }
29 }
```

- The form of the energy equation to be used is specified in line 9 (**energy**).
- In this case we are using enthalpy formulation (**sensibleEnthalpy**). In this formulation, the following equation is solved,

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\rho \mathbf{u} h) + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho \mathbf{u} K) - \frac{\partial p}{\partial t} = \nabla \cdot (\alpha_{eff} \nabla e) + \rho \mathbf{g} \cdot \mathbf{u} + S$$

- In the sub-dictionary **mixture** (lines 12-29), we define the thermophysical properties of the working fluid (air in this case).
- In line 17, we define the molecular weight.
- In line 21, we define the specific heat **Cp**. The heat of formation **Hf** is defined in line 22 (not used in this case).
- As we are using the transport model **const** (line 5), we need to define the dynamic viscosity **mu** and Prandtl number **Pr** (lines 26 and 27).
- As we want to solve the Euler equations, we set the viscosity to zero (line 26). We also define the Prandtl number in line 27 but is not used in this case as we are solving the Euler equations.

High speed compressible flow – Expansion fan

Selecting thermophysical properties

- When solving the enthalpy formulation of the energy equation,

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\rho \mathbf{u} h) + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho \mathbf{u} K) - \frac{\partial p}{\partial t} = \nabla \cdot (\alpha_{eff} \nabla e) + \rho \mathbf{g} \cdot \mathbf{u} + S$$

the pressure work term $\partial p / \partial t$ can be excluded from the solution.

- This has a stabilizing effect on the solution, specially if you are using steady solvers.
- To turn off the pressure work term $\partial p / \partial t$, set the option `dpdt` to `no` in the *thermophysicalProperties* dictionary.

...

...

...

dpdt no;

High speed compressible flow – Expansion fan

Selecting turbulence model

- As we are solving the Euler equations (no viscosity), there is no turbulence involved.
- Nevertheless, we need to set the turbulence model to laminar in the dictionary *turbulenceProperties*.
- This dictionary is located in the directory `constant`.

```
simulationType    laminar;
```

- At this point we are done with the physical properties.
- Let us define the boundary conditions.

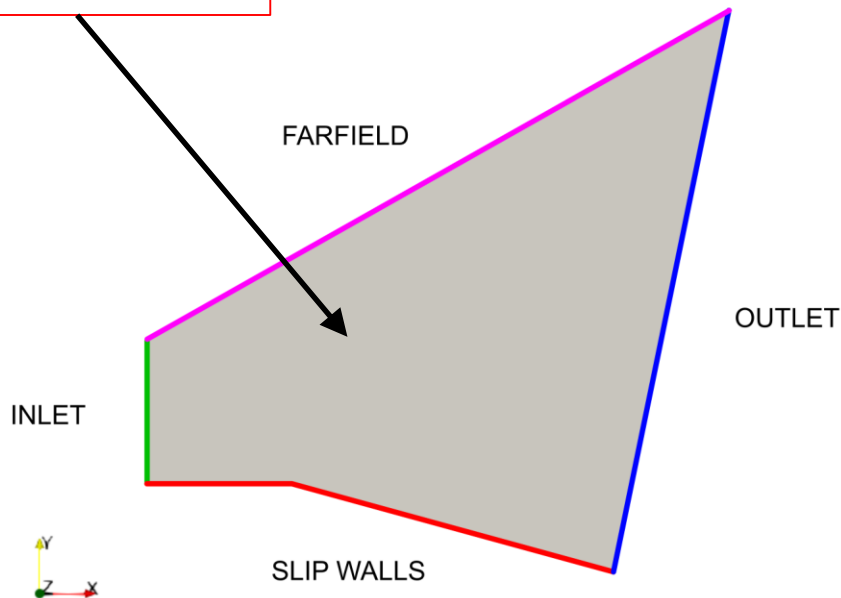
High speed compressible flow – Expansion fan

Boundary and initial conditions

- The boundary and initial conditions are defined as follows.

Initial conditions:

$U \rightarrow (867.7192 \ 0 \ 0)$
 $p \rightarrow 303961$
 $T \rightarrow 300$



Boundary conditions

INLET:

$U \rightarrow (867.7192 \ 0 \ 0)$
 $p \rightarrow 303961$
 $T \rightarrow 300$

fixedValue

OUTLET-FARFIELD:

$U \rightarrow \text{inletOutlet}$
 $p \rightarrow \text{zeroGradient}$
 $T \rightarrow \text{inletOutlet}$

SLIP WALLS:

$U, p, T \rightarrow \text{slip}$

Note: the inlet velocity is equivalent to a Mach number of 2.5. The speed of sound used was for air at 303961 Pa and 300 Kelvin.

High speed compressible flow – Expansion fan

Selecting the discretization schemes

```
1  ddtSchemes
2  {
3      default          steadyState;
4  }
5
6  gradSchemes
7  {
8      default          cellLimited Gauss linear 1;
9      grad(p)          cellLimited Gauss linear 0.333;
10     grad(h)          cellLimited Gauss linear 0.333;
11 }
12
13 divSchemes
14 {
15     default          none;
16     div(phi,U)      bounded Gauss vanLeerV;
17     div(phi,K)      bounded Gauss limitedLinear 1;
18     div(phi,h)      bounded Gauss limitedLinear 1;
19     div(phiid,p)    Gauss limitedLinear 1;
20     div((nuEff*dev2(T(grad(U)))) Gauss linear;
21 }
22
23 laplacianSchemes
24 {
25     default          Gauss linear limited 1;
26 }
27
28 interpolationSchemes
29 {
30     default          linear;
31 }
32
33 snGradSchemes
34 {
35     default          limited 1;
36 }
```

- The discretization schemes are set in the dictionary *fvSchemes* located in the directory **system**.
- When dealing with compressible flows and strong discontinuities (such as shock waves), it is of paramount importance to set a robust and accurate numerics, as the one used in this case.
- In this case, we use a steady solver(line 3).
- In lines 6-11 we define the gradient discretization. When dealing with shock waves, it is recommended to use an aggressive limiter for **grad(U)** (line 8).
- Be careful not to add very aggressive limiters to **grad(p)** and **grad(h)** (line 8), as they may add a lot of numerical diffusion.
- In this case, we are using a stable and accurate limiter for **grad(p)** and **grad(h)** (lines 9 and 10).

High speed compressible flow – Expansion fan

Selecting the discretization schemes

```
1  ddtSchemes
2  {
3      default          steadyState;
4  }
5
6  gradSchemes
7  {
8      default          cellLimited Gauss linear 1;
9      grad(p)          cellLimited Gauss linear 0.333;
10     grad(h)          cellLimited Gauss linear 0.333;
11 }
12
13 divSchemes
14 {
15     default          none;
16     div(phi,U)       bounded Gauss vanLeerV;
17     div(phi,K)       bounded Gauss limitedLinear 1;
18     div(phi,h)       bounded Gauss limitedLinear 1;
19     div(phiid,p)     Gauss limitedLinear 1;
20     div((nuEff*dev2(T(grad(U)))) Gauss linear;
21 }
22
23 laplacianSchemes
24 {
25     default          Gauss linear limited 1;
26 }
27
28 interpolationSchemes
29 {
30     default          linear;
31 }
32
33 snGradSchemes
34 {
35     default          limited 1;
36 }
```

- In lines 13-21 we define the discretization of the convective terms.
- Notice that for velocity (line 16) we are using a TVD scheme.
- TVD schemes are highly recommended when you are dealing with strong discontinuities (such as shock waves).
- In lines 17-20 we define the discretization schemes for the variables related to the energy equation. In general, the setup used is accurate and stable.

$$\frac{\partial \rho h}{\partial t} + \underbrace{\nabla \cdot (\rho \mathbf{u} h)}_{\text{div}(\mathbf{phi}, h)} + \frac{\partial \rho K}{\partial t} + \underbrace{\nabla \cdot (\rho \mathbf{u} K)}_{\text{div}(\mathbf{phi}, K)} - \frac{\partial p}{\partial t} = \nabla \cdot (\alpha_{eff} \nabla e) + \rho \mathbf{g} \cdot \mathbf{u} + S$$

- Line 19 is related to the transonic correction used. this correction is set in the dictionary *fvSolution*.

High speed compressible flow – Expansion fan

Selecting the discretization schemes

```
1 ddtSchemes
2 {
3     default          steadyState;
4 }
5
6 gradSchemes
7 {
8     default          cellLimited Gauss linear 1;
9     grad(p)          cellLimited Gauss linear 0.333;
10    grad(h)          cellLimited Gauss linear 0.333;
11 }
12
13 divSchemes
14 {
15     default          none;
16     div(phi,U)       bounded Gauss vanLeerV;
17     div(phi,K)       bounded Gauss limitedLinear 1;
18     div(phi,h)       bounded Gauss limitedLinear 1;
19     div(phiid,p)     Gauss limitedLinear 1;
20     div((nuEff*dev2(T(grad(U)))) Gauss linear;
21 }
22
23 laplacianSchemes
24 {
25     default          Gauss linear limited 1;
26 }
27
28 interpolationSchemes
29 {
30     default          linear;
31 }
32
33 snGradSchemes
34 {
35     default          limited 1;
36 }
```

- For steady state cases, the bounded form can be applied to the divSchemes.
- In this case, **div(phi,U)** is defined as **bounded Gauss vanLeerV**.
- The terms **div(phi,K)** and **div(phi,h)** are defined as **bounded Gauss limitedLinear 1**.
- This adds a linearized, implicit source contribution to the transport equation of the form,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) - (\nabla \cdot \mathbf{u})\mathbf{u} = \nabla \cdot (\Gamma \nabla \mathbf{u}) + S$$

- This term removes a component proportional to the continuity error. This acts as a convergence aid to tend towards a bounded solution as the calculation proceeds.
- At convergence, this term becomes zero and does not contribute to the final solution.

High speed compressible flow – Expansion fan

Selecting the solution method and linear solvers

```
1 solver
2 {
3   p
4   {
5     solver      PBiCGStab;
6     preconditioner DILU;
7     tolerance   1e-06;
8     relTol      0.001;
9     minIter     2;
10  }
11
12  u
13  {
14    solver      PBiCGStab;
15    preconditioner DILU;
16    tolerance   1e-08;
17    relTol      0.001;
18    minIter     2;
19  }
20
21  h
22  {
23    solver      PBiCGStab;
24    preconditioner DILU;
25    tolerance   1e-08;
26    relTol      0.001;
27    minIter     2;
28  }
29
30  rho
31  {
32    solver      diagonal;
33  }
34
35 }
36
```

- The solution method, corrections and linear solvers are set in the dictionary *fvSolution* located in the directory **system**.
- In this case, we are using the linear solver **PBiCGStab** for all variables except **rho**.
- In compressible solvers, **rho** is computed from the thermodynamical variables, therefore, we use a diagonal solver (back substitution).
- It is recommended to set the minimum number of iterations to 2 or 3, this increases the stability and helps at linearizing the linear system.
- The minimum number of iterations is set by using the keyword **minIter**.

High speed compressible flow – Expansion fan

Selecting the solution method and linear solvers

```
1 SIMPLE
2 {
3     transonic          yes;
4     consistent         yes;
5
6     residualControl
7     {
8         P      1e-4;
9         U      1e-4;
10        h      1e-4;
11    }
12
13    nNonOrthogonalCorrectors  2;
14
15    pMin      10000;
16    pMax      1000000;
17 }
18
19 relaxationFactors
20 {
21     fields
22     {
23         p      0.7;
24         rho    0.7;
25     }
26     equations
27     {
28         p      0.7;
29         U      0.7;
30         h      0.7;
31     }
32 }
33
```

- In this case we are using the **SIMPLE** pressure-velocity coupling (line 1).
- In line 3, we enable the transonic correction.
- This option is recommended when the critical Mach number is exceeded (usually for Mach number values above 0.5).
- In line 4 we are enabling the **SIMPLEC** pressure-velocity coupling formulation.
- This formulation is more stable and faster than the **SIMPLE** formulation.
- In lines 6-11 we are defining the residuals control for the field variables. If the solver reach this tolerance it will automatically stop.
- In line 13 we define the non-orthogonal corrections (these corrections are related to the mesh quality).
- In lines 15-16 we define the minimum and maximum values of pressure. This is not compulsory, but it is recommended to follow this practice to avoid reaching unrealistic values (and of course the values used should be realistic).
- In lines 19-32 we define the under-relaxation factors (URF).
- Remember, URF values are they are problem dependent.
- An optimum choice of under-relaxation factors is one that is small enough to ensure stable computation but large enough to move the iterative process forward quickly.

High speed compressible flow – Expansion fan

fvOptions – Limiting temperature min/max values

- As for the minimum and maximum values of pressure in the dictionary *fvSolution* (**pMin** and **pMax**), it is also possible to add a limit to the minimum and maximum values of the temperature field.
- This can be done by adding a limit in the form of a source term in the *fvOptions* dictionary (located in the directory **system**).
- This limiting of the minimum and maximum values is not compulsory. However, it is recommended to follow this practice to avoid reaching unrealistic values and eventually divergence.
- No need to mention that the limits defined should be realistic.
- Also, it is recommended to use this form of limiter during the initial iterations and until the solution has stabilized.
- In the lines below, a limiter for the temperature is shown (there is a similar version for the velocity field).

```
limitT  
{  
  type          limitTemperature;  
  active        yes;  
  min           100;  
  max           1000;  
  selectionMode all;  
}
```

← User given name of the limiting source term

← Source term to limit temperature

← Turn on/off limit

← Allowed minimum and maximum values

← Selection mode, in this case we are applying the limit to the whole domain. It can also be applied to a region or cellSet.

High speed compressible flow – Expansion fan

Running the case

- You will find this tutorial in the directory `$PTOFC/supersonic_wegde`
- In the terminal window type:

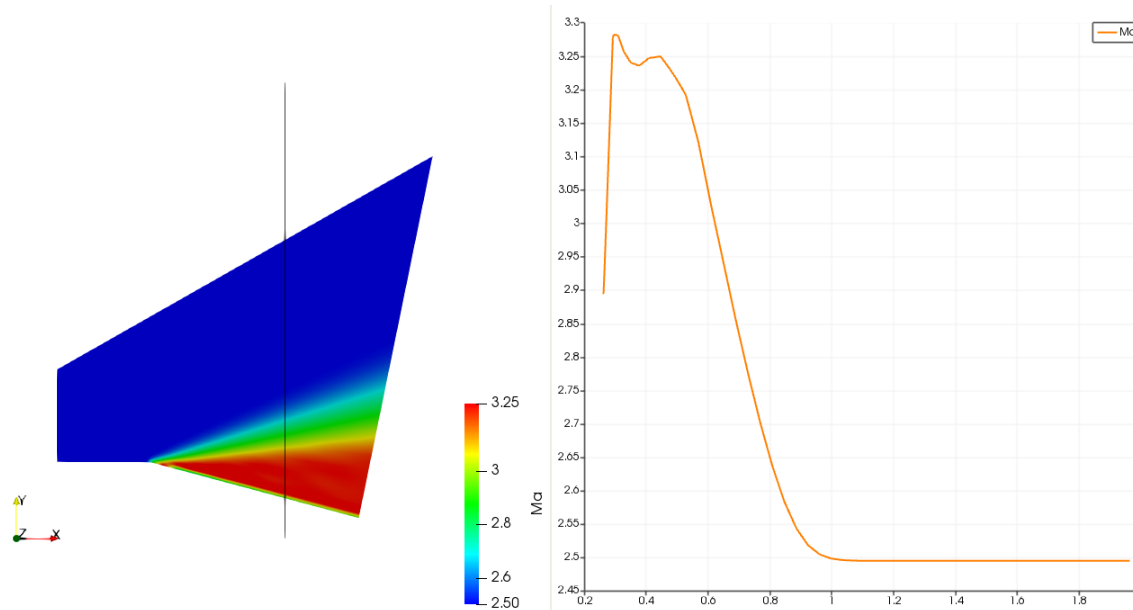
1. `$> cp -r 0_empty 0`
2. `$> renumberMesh -overwrite`
3. `$> rhoSimpleFoam | tee log.solver`
4. `$> rhoSimpleFoam -postProcess -func MachNo`
5. `$> paraFoam`

- In line 4 we compute the Mach number.
- You cannot use the pseudo-transient formulation with steady solvers (`rhoSimpleFoam` in this case).

High speed compressible flow – Expansion fan

Visualizing the solution in Paraview

- At this point, you can visualize the solution using Paraview and compare the output with the analytical solution.



Analytical solution [1, 2]

	Mach number
Mach number after expansion fan/corner	3 .2370

- [1] Modern compressible flow: with historical perspective. Third edition. McGraw-Hill.
- [2] Compressible aerodynamics calculator. <http://www.dept.aoe.vt.edu/~devenpor/aoe3114/calc.html>