2D Combustor case – Compressible flows

Federico Piscaglia *
Dept. of Aerospace Science and Technology (DAER), Politecnico di Milano, Italy

Abstract. Lab handout for the study of compressible flow inside a combustor-like geometry. Topics covered are the setup of the compressible solver `rhoPimpleFoam` in low- and high- speed configurations, starting from the incompressible case.

1 Learning outcome

The software used is the open-source CFD software OpenFOAM®-8 by the OpenFOAM Foundation. In this Laboratory you will learn how to:

- set up a case with a compressible single-phase flow, including heat transfer;
- set up a transonic flow case;
- provide understanding on the choice between compressible and incompressible formulation based on the physics to model;
- analyse the results.

2 Introduction

In this Lab we will consider flows with variable density. Density ρ can vary because of the effect of the temperature and/or of the pressure. Variations in flow density affect continuity, momentum and, if the gravity is accounted, can generate buoyancy:

1) The simplest way to consider a density change is by the volumetric expansion coefficient β:

$$(\rho - \rho_0) = -\beta\rho(T - T_0) \tag{1}$$

with T_0 is an appropriate reference temperature. **The above linearized relationship is almost exact for liquids, but it is acceptable for gases just if temperature gradients are limited.** Eq. (**??**) is known as the Boussinesq approximation and it is widely used to solve buoyancy problems.

2) With large temperature and pressure gradients, Eq. (**??**) must be replaced by an equation of state such as, in case of a perfect gas:

$$\frac{p}{\rho} = RT \tag{2}$$

Changes in density due to the pressure are not accounted when the Boussinesq approximation is used. Pressure might change by the effect of source terms (moving boundaries, combustion, etc.) or because of velocity. Should this

*Tel. (+39) 02 2399 8620, E-mail: federico.piscaglia@polimi.it

be the case, incompressible flow equation are not valid anymore and they must be replace with their compressible counterpart. Commonly, the breakthrough point between compressible and incompressible flows is considered to be at $M \approx 0.3$. The variation of density due to high Mach number affects the solution the of pressure-velocity coupling problem, and terms that can be neglected at $M < 0.3$ now must be taken into account. If the flow is transonic, details of the numerical procedure for transonic flows is reported in Appendix **??**.

3 Boussinesq solver

In the past (until OpenFOAM-6), the solver for heat transfer problems based on the Boussinesq approximation was `buoyantPimpleFoam`. With the recent releases of OpenFOAM, the solver has been updated. In the recent releases, the Boussinesq approximation is implemented as an equation of state in the `thermophysicalProperties` file.

```
thermoType
{
    type            heRhoThermo;
    mixture         pureMixture;
    transport       const;
    thermo          eConst;
    equationOfState Boussinesq;
    specie          specie;
    energy          sensibleInternalEnergy;
}
```

With respect to the compressible flow solver `rhoPimpleFoam`, the solver for buoyancy flow includes the contribution of gravity in the momentum equation, the effect of radiation. As mentioned, the Boussinesq approximation can be selected with any kind of solver in the recent releases of OpenFOAM. The formulation of the solver `buoyantPimpleFoam` is fully compressible and therefore more general.

3.1 Prepare

The starting point is the setup of the final assignement of the previous lab, where the gas flow velocity was set to $U = 10$ m/s at the inlet patch and $U_f = 15$ m/s (gas velocity at the fuel inlet).

```
user@host:run$ cp -r combustor2D-2flows combustor2D-BBP
```

3.2 Set fluid properties

The gravitational acceleration "$\boldsymbol{g}$" and `thermophysicalProperties` files must be copied in the `constant` folder. The `hotRoomBoussinesq` tutorial helps.

```
user@host:combustor2D-BBP$ cp -r $FOAM_TUTORIALS/heatTransfer/buoyantPimpleFoam/
```

```
user@host:combustor2D-BBP$ cp -r $FOAM_TUTORIALS/heatTransfer/buoyantPimpleFoam/                ho
```

In "constant/g" $\rightarrow$ set $\boldsymbol{g} = -9.81 m/s^2$ in Y-direction;

In "constant/thermophysicalProperties" :

1. set volumetric expansion coefficient $\beta = 3.4e-3$;
2. add Prandtl total number $Prt = 0.85$ among the transport quantities;

3. check:

```
T0    300;
Pr    0.7;
```

3.3 Temperature equation

3.3.1 Create a field T

1. by copying the pressure:
   `user@host:combustor2D-BPP$ cp 0/p 0/T`
2. Adjust name in header;
3. Adjust units of measurement [K]
   `dimensions [0 0 0 1 0 0 0];`
4. Set fixedValues T=300 K at the main inlet, T=350 K at fuel inlets, zero gradient at the outlet and walls. You can set a uniform flow temperature T=300K for the internal field.

3.4 Pressure equation

3.4.1 Create a pressure field ($p\_rgh = p - \rho gh$)

1. by copying the pressure:
   `user@host:run/combustor2D-BPP$ cp 0/p 0/p_rgh`
2. Within p_{rgh}:
   - Adjust name in header;
   - Adjust units of measurement $[kg/(ms^2)]$
     `dimensions [1 -1 -2 0 0 0 0];`
3. Within p
   - Adjust units of measurement $[kg/(ms^2)]$
     `dimensions [1 -1 -2 0 0 0 0];`
   - change B.C. as calculated (value 0);

3.5 Add turbulent thermal diffusivity

1. Start from the turbulent viscosity:
   `user@host:run/combustor2D-BPP$ cp 0/nut 0/alphat`
2. Change name in the header;
3. Set units of measurement to $kgm^{-1}s^{-1}$ `dimensions [1 -1 -1 0 0 0 0];`
4. Set wall function on solid walls:

```
wall
{
    type            compressible::alphatWallFunction;
    value           uniform 0;
}
```

5. Set α_t to "calculated" on all other boundaries (but the empty ones) and use a starting value of 0.

3.5.1 Modify discretization schemes

In system/fvSchemes:

1. comment the `grad(U)`;
2. comment the `div(phi,v2)`;
3. comment the divergence of non linear stresses;
4. add or modify:

```
div(phi,U) Gauss upwind;
div(phi,k) Gauss upwind;
div(phi,epsilon) Gauss upwind;
div(phi,omega)   Gauss upwind;
div(((rho*nuEff)*dev2(T(grad(U))))) Gauss linear;

div(phi,e)        Gauss upwind;
div(phi,K)        Gauss linear;
div(phiv,p)       Gauss linear;
```

3.5.2 Modify linear solver

In system/fvSolution:

1. add:

```
"rho.*"
{
    solver          diagonal;
}
```

2. add "e" to the existing solvers:

```
"(U|e|k|epsilon|omega)"
{
    solver          smoothSolver;
    smoother        symGaussSeidel;
    tolerance       1e-05;
    relTol          0;
}

"(U|e|k|epsilon|omega)Final"
{
    $U
    relTol          0;
}
```

3. Change EVERY "p" with "p_rgh" in the linear solvers and reduce relTol to 0.01:

```
p_rgh
{
    solver          GAMG;
    tolerance       1e-06;
    //  suggested in  unsteady  flow  solvers
```

```
        //  AND  small  number of  outer iterations
        relTol          0.0;
        smoother        GaussSeidel;
    }

    p_rghFinal
    {
        $p_rgh;
        relTol 0;
    }
```

4. adjust PIMPLE loop correctors and residual controls:

```
    PIMPLE
    {
        nCorrectors 2;
        nOuterCorrectors 1;
        nNonOrthogonalCorrectors 0;
        consistent no;
        outerCorrectorResidualControl
        p
        {
            tolerance 1e-4;
        }

        U
        {
            tolerance 1e-5;
        }
    }
```

5. adjust relaxation factors:

```
    relaxationFactors
    {
      fields
      {
          "p.*" 1.0;
      }
      equations
      {
          "U.*"           1.0;
          "(k|epsilon)" 1.0;
      }
    }
```

6. remove SIMPLE;

3.6 Adjust run parameters

- max CFL=1
- $\Delta t = 10^{-5}$ s
- end time: 0.05 s
- delete unnecessary Function Objects

3.7 Run

```
user@host:run/combustor2D-BPP$ buoyantPimpleFoam > log.BBP
```

IMPORTANT NOTE: the `buoyantPimpleFoam` solver is NOT the correct solver to use in this case! This solver is though for confined rooms where flow transport is not dominant and gravity has an important role to determine the motion of the flow. The combustor case simulated is open at its boundaries, so the effect of gravity of the flow motion is negligible. You should NOT expect different results from `rhoPimpleFoam`.

4 Compressible solver – low speed

The proper way to account for density change is to use a compressible solver, i.e., a solver that takes into account density changes by solving both the equation of state and the energy conservation. In OpenFOAM, the compressible counterpart of `pimpleFoam` is `rhoPimpleFoam`.

Main differences in the setup with respect to the incompressible case are:

- Fluid thermodynamic behavior must be defined.
- Energy equation is added to the solver (see Sec. **??** for details).
- Pressure in compressible flows is the thermodynamic pressure: $[p] = \text{kg} \cdot \text{m}^{-1}\text{s}^{-2}$.
- Heat transfer with walls must be accounted for.

4.1 Prepare

The starting point is the incompressible **unsteady** setup with $U = 10$ m/s and $U_f = 15$ m/s, being U_f the gas inlet velocity at the fuel nozzles (see the case setup of Lab03)..

```
user@host:run$ cp -r d-combustor2Flows combustor2D-RPF
```

(note: `d-combustor2Flows` comes from Lab02.

4.2 Define thermophysical properties

1. Copy the template file:

   ```
   $ cp -r $FOAM_TUTORIALS/compressible/rhoPimpleFoam/LES/pitzDaily/constant/   \
     thermophysicalProperties constant/.
   ```

2. Change energy to:

   ```
   energy          sensibleEnthalpy;
   ```

3. Change thermo to:

   ```
   thermo hConst;
   ```

4. Add the following line in the subdict “thermodynamics”

   ```
   Cp      1004;
   ```

5. Add the following line in the dictionary main body:

   ```
   dpdt    yes;
   ```

4.3 Setup the energy equation solution

4.3.1 Create a field T

1. by copying the pressure:
   ```
   user@host:combustor2D-BPP$ cp 0/p 0/T
   ```
2. Adjust name in header;
3. Adjust units of measurement [K]
   ```
   dimensions [0 0 0 1 0 0 0];
   ```
4. Set fixedValues T=300 K at the main inlet, T=350 K at fuel inlets, zero gradient at the outlet and walls. You can set a uniform flow temperature T=300K for the internal field.
   Note: try changing the temperature level at $T_{fuel\_inlet} = 400$ K.

4.3.2 Add discretization schemes for energy

In "system/fvSchemes" add the following lines:

```
div(phi,h)                          Gauss limitedLinear 1; //convection of enthalpy
div(phi,K)                          Gauss limitedLinear 1; //convection of kinetic energy
div(((rho*nuEff)*dev2(T(grad(U))))) Gauss linear; //compressible stress tensor
```

4.3.3 Add solvers and relaxation

Add the following lines to "system/fvSolution":

```
"rho.*"
{
    solver          diagonal;
    tolerance       1e-09;
    relTol          0;
}

h
{
    solver          smoothSolver;
    smoother        symGaussSeidel;
    tolerance       1e-05;
    relTol          0.1;
}
hFinal
{
    $h
    relTol          0;
}
```

Note: you can add "h" to the already existing "U" solver.

```
relaxationFactors
{
```

```
    // ...
    equations
    {
        //...
        "(h|e)" 0.5;
    }
}
```

ATTENTION: adjust convergence criteria.

4.3.4 Set thermodynamic pressure

Open the file "0/p" and:

1. Set units of measurement to `[1 -1 -2 0 0 0 0]`
2. Set internal field to 101325;
3. Set the value of pressure at the inlet boundary:

```
inlet
{
    type        fixedValue;
    value       uniform 101325;
}
```

4. Set a non-reflecting BC on the outlet:

```
outlet
{
    type        waveTransmissive;
    rho         rho;
    gamma       1.4;
    fieldInf    101325;
    lInf        10;
    value       uniform 101325;
}
```

Note: the non-reflecting outlet BC avoids any reflected wave from the outlet, that would be generated if using a plain "fixedValue" BC and may possibly impair the solution stability. In compressible flows, the use of non-reflecting BCs improves robustness.

4.4 Add turbulent thermal diffusivity

1. Start from the turbulent viscosity:

   `user@host:run/combustor2D-BPP$ cp 0/nut 0/alphat`

2. Change name in the header;
3. Set units of measurement to $kgm^{-1}s^{-1}$ `dimensions [1 -1 -1 0 0 0 0];`
4. Set wall function on solid walls:

```
wall
{
    type            compressible::alphatWallFunction;
    value           uniform 0;
}
```

5. Set α_t to "calculated" on all other boundaries (but the empty ones) and use a starting value of 0.

4.5 Adjust run parameters

- max CFL=5
- $\Delta t = 10^{-5}$ s
- end time: 0.05 s
- delete unnecessary Function Objects

4.6 Run

```
user@host$ rhoPimpleFoam > log.rhoPimpleFoam 2>&1 &

user@host$ tail -f log.rhoPimpleFoam
```

5 Compressible solver – high speed

If the velocity of flow is increased beyoind $M \approx 0.3$, local pressure gradients also influence gas compressibility, that becomes significant. Let's try and see what happens at higher velocity.

5.1 Prepare

The starting point is the **compressible** unsteady setup with $U = 10$ m/s and $U_f = 15$ m/s (U_f is the gas inlet velocity at fuel nozzles).

```
user@host:run$ cp -r combustor2D-RPF combustor2D-highRe
```

5.2 Change velocity

Increase all velocities by a factor of 10: $U = 100$ m/s and $U_f = 150$ m/s (U_f is the gas inlet velocity at fuel nozzles).

5.3 Adjust run parameters

- max CFL=5
- $\Delta t = 5 \cdot 10^{-7}$ s
- end time: 0.05 s
- delete unnecessary Function Objects
- Add a Function Object to compute (and write) the Mach number each time the results are written on disk.

5.4 Run

```
user@host$ rhoPimpleFoam > log.rhoPimpleFoam
```

6 Hands-on

Set all wall temperature to 1500 K and the fuel inlet temperature to 700 K. Run all cases both at low and high gas speed.

Appendix

A Buoyant flows

When heat is added to a fluid and the fluid density varies with temperature, a flow can be induced due to the force of gravity acting on the density variations. Such buoyancy-driven flows are termed natural-convection (or mixed-convection) flows; for these flows pressure does not change substantially, but the temperature and/or concentration of solutes can cause large variation in fluid properties. Buoyant flows differ from forced convective flows in some significant aspects. In particular, in subsonic forced convective flows, the coupling between the momentum and energy equations tends to be one-way with momentum affecting the advection term in the energy equation. The energy equation typically does not feed back into the momentum equation directly. In buoyant flows, the coupling is direct and two-way with the density gradient in a gravity field appearing in the momentum equations. There is little disagreement about the effect of buoyancy on the mean flow. Variations in density, viscosity, Prandtl number and specific heat increase the non-linearity of the equations. The sequential solution method can be applied to these flows in almost the same way they are applied to flows with variable temperature. In buoyant flows fluid properties are recalculated after each outer iteration and they are treated as known during the next outer iteration.

A.1 Equations

In many practical problems (incompressible flows) it is not necessary evaluate the whole pressure p, but it is sufficient to know the pressure field $\tilde{p}$ correct with the hydrostatic component:

$$\tilde{p} = p - (-\rho_0 g H) = p + \rho_0 g H \tag{3}$$

where:

- ρ_0 is a reference density
- g the gravitational constant
- H is the distance from the lowest geodetic high in the domain (measured opposite to the gravitational field)

Note: the momentum equation

$$\frac{\partial(\rho \boldsymbol{v})}{\partial t} + \nabla(\rho \boldsymbol{v}\boldsymbol{v}) = \rho g - \nabla p + \nabla \cdot (\mu \nabla \boldsymbol{v}) \tag{4}$$

may be simplified, for an isothermal flow field without mean velocity $\boldsymbol{v}$ ($\boldsymbol{v}$=0), as:

$$0 = \rho g - \nabla p \rightarrow \nabla p = \rho g \tag{5}$$

Since $\nabla = \frac{\partial}{\partial z}$ and H is a vector oriented along the opposite verse of the vector $\boldsymbol{g}$:

$$\nabla H = \frac{\partial H}{\partial z} = -1 \tag{6}$$

it follows:

$$\nabla \tilde{p} = \nabla p + \rho_0 g \nabla H \rightarrow \nabla p + \rho_0 g \frac{-\boldsymbol{g}}{g} = \nabla p - \rho_0 \boldsymbol{g} \tag{7}$$

Being:

$$\tilde{p} = p - (-\rho_0 g H) = p + \rho_0 g H$$
$$\nabla \tilde{p} = \nabla p - \rho_0 \boldsymbol{g}$$
$$\nabla p = \nabla \tilde{p} + \rho_0 \boldsymbol{g}$$

The buoyancy term in the momentum equation can be written as:

$$-\nabla p + \rho \boldsymbol{g} = -\left(\nabla \tilde{p} + \rho_0 \boldsymbol{g}\right) + \rho \boldsymbol{g} = -\nabla \tilde{p} - (\rho_0 - \rho)\boldsymbol{g} \tag{8}$$

For **very low temperature variations of the fluid flow**:

$$(\rho - \rho_0) = -\beta \rho (T - T_0) \tag{9}$$

where

$$\beta = -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right)_P \tag{10}$$

and

$$-\nabla p + \rho \boldsymbol{g} = -\nabla \tilde{p} - \rho\beta(T - T_0)\boldsymbol{g} \tag{11}$$

The resulting momentum equation takes the form:

$$\frac{\partial(\rho \boldsymbol{v})}{\partial t} + \nabla(\rho \boldsymbol{v}\boldsymbol{v}) = -\nabla \tilde{p} - \rho\beta(T - T_0)\boldsymbol{g} + \nabla \cdot (\mu \nabla \boldsymbol{v}) \tag{12}$$

Some notes:

- for steady flows, multigrid method can result in a substantial speed-up;
- if the variation property is significant, the convergence may be slowed considerably.
- The importance of buoyancy forces in a mixed convection flow can be measured by the ratio if the Grashof and Reynolds numbers:

$$\frac{Gr}{Re^2} = \frac{g\beta\Delta T L}{v^2} \tag{13}$$

  When this number approaches or exceeds unity, strong buoyancy contributions to the flow should be expected. Conversely, if it is very small, buoyancy forces may be ignored.

- `buoyantSimpleFoam` might perform poorly with low-quality grids. The problem lies in the calculation of `grad(x)`. If x is a linearly varying field, the calculation of the gradient does not return a uniform field value on highly non-hortogonal girds meshes. In presence of large gradients, such as that induced by the hydrostatic force ($\rho\, g$), trying to balance the hydrostatic force by grad(p) fails and produces large spurious momentum sources on poor mesh elements.
- If the gravitational acceleration $\vec{g}$ is set to zero, the buoyancy force will be also zero. The issue with low-quality grids should disappear, since there will be no hydrostatic pressure gradient and the solver will recover to the SIMPLE flow solver, where the temperature solution is uncoupled.

B Energy equation in OpenFOAM®

Energy equation can be written in OpenFOAM® in basically three forms:

1. Based on temperature
2. Based on enthalpy
3. Based on internal energy

The first form is used in the Boussinesq family of solvers and basically reads:

$$\frac{\partial T}{\partial t} + \nabla \cdot (\varphi T) - \nabla \cdot \alpha_{eff} \nabla T = S_T \tag{14}$$

Where the total change of temperature in a control volume ($DT/Dt = \partial T/\partial t + \nabla \cdot (\phi T)$) is caused by diffusion and source terms. α_{eff} is the effective thermal diffusivity and accounts for both conductivity and turbulent diffusivity (through the turbulent Prandtl number).

Conservation of enthalpy or internal energy are implemented in the `rho*` family of solvers:

$$\frac{\partial \rho h}{\partial t} + \nabla \cdot (\varphi h) + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\varphi K) - \frac{\partial p}{\partial t} - \nabla \cdot \alpha_{eff} \nabla h = S_h \tag{15}$$

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\varphi e) + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\varphi K) + \nabla \cdot (\varphi p) - \nabla \cdot \alpha_{eff} \nabla e = S_e \tag{16}$$

Both equations are implemented in the `rho*` solvers, and the variable that is actually solved for to enforce energy conservation is selected in the "constant/thermophysicalProperties".

Notes:

- Pressure derivative in the enthalpy formulation must be activated in the "constant/thermophysicalProperties": `dpdt yes;`
- Usually h is preferred for steady flows, combustion
- e is preferred for strongly compressible flows (e.g. shock tube)

C Solution at arbitrary Mach number

REFERENCE: J.H. Ferziger and M. Perić, Computational methods for fluid dynamics, 3rd edition, Springer, 2002. Pages 309-315.

Governing equations of compressible flow. In a fully compressible flow, one has to consider three governing equations: momentum (**??**), energy (**??**) , continuity (**??**):

$$\frac{\partial}{\partial t}\int_\Omega \rho \boldsymbol{v} \mathrm{d}\Omega + \int_S \rho \boldsymbol{v}\boldsymbol{v}\cdot\boldsymbol{n}\mathrm{d}S = \int_S \mathrm{T}\cdot\boldsymbol{n}\mathrm{d}S + \int_\Omega \rho\boldsymbol{b}\mathrm{d}\Omega \tag{17}$$

$$\frac{\partial}{\partial t}\int_\Omega \rho h \mathrm{d}\Omega + \int_S \rho h\boldsymbol{v}\cdot\boldsymbol{n}\mathrm{d}S = \int_S k\nabla T\cdot\boldsymbol{n}\mathrm{d}S + \int_\Omega \left[\boldsymbol{v}\cdot\nabla p + \mathrm{S}\colon\nabla\boldsymbol{v}\right]\mathrm{d}\Omega + \frac{\partial}{\partial t}\int_\Omega p\mathrm{d}\Omega \tag{18}$$

$$\frac{\partial}{\partial t}\int_\Omega \rho \mathrm{d}\Omega + \int_S \rho\boldsymbol{v}\cdot\boldsymbol{n}\mathrm{d}S = 0 \tag{19}$$

and the equation of state (**??**):

$$p = \rho RT \tag{20}$$

Here the pressure has the meaning of actual thermodynamic pressure, whose absolute value is of critical importance.

Pressure-correction procedure. By discretizing the momentum equation (**??**), one can get the predicted i-th velocity component in cell P at iteration m:

$$u_{i,P}^{m*} = \frac{Q_{u_i}^{m-1} - \sum_l A_l^{u_i} u_{i,l}^{m*}}{A_P^{u_i}} - \frac{\Delta\Omega}{A_P^{u_i}}\left(\frac{\partial p^{m-1}}{\partial x_i}\right) \tag{21}$$

which originates a predicted mass flux through the cell faces $\dot{m}^*$

The discretized continuity equation is written as:

$$\frac{\rho^{m-1}-\rho^n}{\Delta t} + \sum_f \dot{m}_f^* = Q_m^* \tag{22}$$

where Q_m^* is an imbalance term to be eliminated by a correction procedure. Each predicted mass flux differs from its corrected counterpart because of two contribution: velocity and density:

$$\dot{m}_f^m = (\rho^{m-1} + \rho')_f (v_n^{m*} + v_n')_f S_f \tag{23}$$

thus a mass correction can be defined (last term is of higher order and usually neglected):

$$\dot{m}'_f = S_f[(\rho^{m-1}v'_n)_f + (v_n^{m*}\rho')_f + \cancel{(\rho' v'_n)}] \tag{24}$$

The first term in Eq. **??** is expressed in terms of the pressure gradient in the same way as in the incompressible solver:

$$S_f(\rho^{m-1}v'_n)_f = -(\rho^{m-1}S\Delta\Omega)_f \left[\frac{1}{A_P^{v_n}}\right]_f \left[\frac{\partial p'}{\partial n}\right]_f \tag{25}$$

while the expression for the density corrector depends upon the pressure-density relationship. For a perfect gas, considering fixed temperature, one has:

$$\rho' \approx \left.\frac{\partial \rho}{\partial p}\right|_T p' = \psi p' \tag{26}$$

where ψ is the gas compressibility: $\psi = (RT)^{-1}$.

Hence, the second term in Eq. **??** becomes:

$$(v_n^{m*}\rho' S)_f = \left[\frac{\psi \dot{m}^*}{\rho^{m-1}}\right]_f p'_f \tag{27}$$

thus leading to the final form for the mass correction:

$$\dot{m}'_f = -(\rho^{m-1}S\Delta\Omega)_f \left[\frac{1}{A_P^{v_n}}\right]_f \left[\frac{\partial p'}{\partial n}\right]_f + \left[\frac{\psi \dot{m}^*}{\rho^{m-1}}\right]_f p'_f \tag{28}$$

Now, the continuity equation written for mass flux correction is:

$$\frac{\rho'_P \Delta\Omega}{\Delta t} + \sum_f \dot{m}'_F + Q_m^* = 0 \tag{29}$$

and by expressing ρ'_P by Eq. (**??**) one can get the discretized pressure equation for compressible flows:

$$A_P p'_P + \sum_l A_l p'_l = -Q_m^* \tag{30}$$

From Eq. (**??**) one can see that the pressure corrector in enters the final equation in two ways: the first one is in a diffusive term (that will be discretized by a laplacian), as it happens in incompressible flows, the second one is a convective term, that will be approximated by a divergence operator. The relative importance of diffusion with respect to convection is proportional to $1/\mathrm{Ma}^2$, therefore the convective contribution can be neglected at low speed.

Solver setup. In OpenFOAM®, the solution of compressible flows using the above outlined pressure correction is activated by the keyword `transonic` in the `fvSolution` dictionary. Consequently, the setup of the case must be modified accordingly, as follows:

- A discretization operator must be set for $(\psi\dot{m}^*/\rho^{m-1})p$

  ```
  div(phid,p) Gauss limitedLinear 1;
  ```

- Matrix of the linear system represented by Eq. (**??**) is not symmetric anymore and an appropriate solver has to be selected:

  ```
  p
  {
      solver          PBiCG;
      preconditioner DILU;
      tolerance       1e-06;
      relTol          0.01;
  }
  ```

D Non-reflecting outlet

REFERENCE: T. Poinsot and D. Veynante, Theoretical and numerical combustion, 2nd edition, Edwards, 2005. Pages 449-451.

If a constant static pressure is set on the domain outlet, negative pressure waves are reflected into the domain. This not only can be problematic for some applications (e.g. acoustics), but it can also render the convergence more difficult. In particular, numerical disturbances are seen by the solver as pressure waves and reflected as well, possibly leading to numerical instability.

A way to overcome this is to use a non-reflecting boundary condition. In its simplest formulation the one-dimensional, inviscid advection of a wave is solved at the outlet to simulate an additional domain length:

$$\frac{\partial p}{\partial t} + \nabla \cdot [(c+u)p] = 0 \tag{31}$$

The name of the BC in OpenFOAM® is "waveTransmissive". Optionally, the value of the field at infinity p_∞ and its distance from the patch L_∞ can be specified:

```
outlet
{
    type      waveTransmissive;
    rho       rho;
    gamma     1.4;
    fieldInf  101325;
    lInf      10;
    value     uniform 101325;
}
```

The underlying assumption of Eq. (**??**) is that pressure waves and velocity are normal to the outlet and that viscous-related phenomena are negligible. If these conditions do not hold, the full set of N-S equation must be solved at the outlet to simulate a real non-reflecting boundary.