CFD Fundamentals (5cfu) – Exercise 1

GROUP 5:

- MATTEO CIGADA (10767277, 270632)
- FEDERICO D'AGOSTINI (10788635, 281037)
- SIMONE D'INCÀ(10837331, 287102)

Contents

Introduction

The aim of the project is to analyze a 2D flow inside a nozzle. The geometry consists of a rectangular inlet, a nozzle constituted by two circular shaped arcs and a rectangular outlet. The fluid inside the nozzle is ambient air (ρ=1.2kg/m^3, v = 15*10^{-6} m^2/s). The flow is considered incompressible and stationary; furthermore, the flow is assumed incompressible since the speed is considerably lower than the speed of sound.

The following simulations have been performed:

- Case A: laminar case analysis
- Case B: turbulent case analysis with two equations model with wall functions
- Case C: turbulent case analysis with two equations model without wall functions

The simulations were carried out using OpenFOAM 11, with simpleFoam as the solver. ParaView was used for post-processing.

Geometry

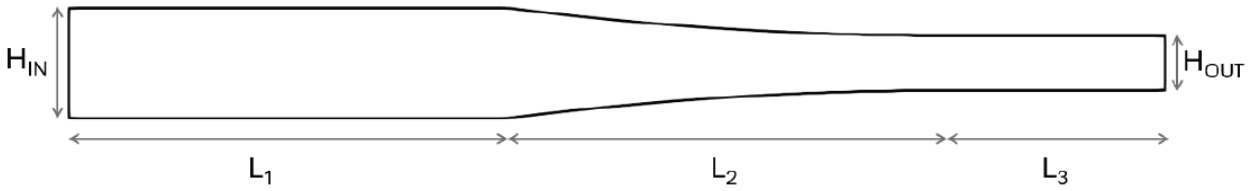


Figure 1

Dimensions	[m]
L_1	1
L_2	1
L_3	0.5
H_{IN}	0.25
H_{OUT}	0.125

Laminar flow

Boundary conditions

The boundary conditions have been imposed according to the rules that guarantee the well-posedness of the problem

- **Pressure $\boldsymbol{p}$** $[m^2/s^2]$:

Inlet	totalPressure (uniform 0.008)
Outlet	fixedValue (0)
topWall	zeroGradient
bottomWall	zeroGradient
Front and back	empty

  - The value at the outlet was provided by the assignment

- **Velocity $\boldsymbol{U}$** $[m/s]$:

Inlet	pressureInletVelocity (uniform 0.126 0 0)
Outlet	zeroGradient
topWall	noSlip
bottomWall	noSlip
Front and back	empty

  - The value at the inlet was provided by the assignment

Setup of the simulation

Mesh

The domain has been divided into 3 blocks; this division is common in all cases.

For this first case, as requested, 5 different meshes have been created, with different refinement levels to achieve grid independence.

In the first case grading was not used in any direction, because acceptable value of skewness, non-orthogonality and aspect ratio were reached for all the meshes generated; furthermore, y^+ was not considered for this laminar case.

Below, the subdivision of the mesh and the checkMesh of the most refined one.

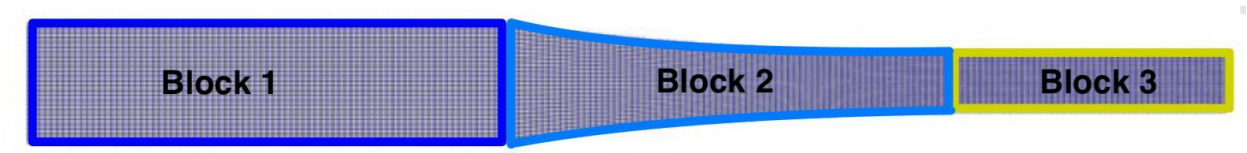


Figure 2

Number of cells	59488
Skewness (max)	0,094
Non orthogonality (max/mean)	7,06° / 1,50°
Aspect ratio	2,61

Domain discretization schemes (fvSchemes)

```
ddtSchemes
{
    default         steadyState;
}

gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      bounded Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         corrected;
}
```

Figure 3

- **Time marching term**: given that the simulation operates under steady-state conditions, the time derivative terms in the equations are set to zero.

- **Gradient term**: a second order accuracy scheme is used. Since no unnatural oscillations were observed in the results, there was no need to use a gradient limiter.
- **Convective flux term**: a second order accuracy scheme is used; "bounded" schemes are typically used for steady state problems and they enhance diagonal dominance.
- **Diffusive flux term**: a second order accuracy scheme is used. The mesh has low non orthogonality, however, using a scheme designed for perfectly orthogonal meshes didn't seem a good option. For this reason, an explicit correction was used; since it didn't introduce unboundedness in the results, there was no need to limit this correction.
- The **interpolation scheme** is set to linear (default option) and for the same reasons described for the diffusive term, an explicit correction was introduced also for the surface normal gradient scheme

Resolution schemes (fvSolution)

```
solvers
{
    p
    {
        solver          GAMG;
    smoother        GaussSeidel;
        tolerance       1e-08;
        relTol          0;
    }


    U
    {
        solver          smoothSolver;
        smoother        GaussSeidel;
        tolerance       1e-08;
        relTol          0;
    }
}


SIMPLE
{
    nNonOrthogonalCorrectors 0;

    residualControl
    {
        p               1e-6;
        U               1e-6;
    }
}

relaxationFactors
{
    fields
    {
     p              0.3;
    }
    equations
    {
        U               0.7;

    }
}
```

Figure 4

- A geometric algebraic multi-grid method has been used for the pressure equation-related resolution matrix with Gauss Seidel as smoother. Since the pressure correction step equation is purely diffusive, the resolution matrix is expected to be

symmetric; for this reason, a “preconditioned conjugate gradient” method with incomplete Cholesky preconditioner is expected to be a valid alternative. This option has been tested, and no significant differences have been observed neither in the results or in the number of iterations needed for convergence.

- A smooth solver is a commonly used option for the velocity equation-related resolution matrix. Gauss Seidel method is used because it is faster compared to other options like Jacobi.
- The number of non-orthogonal correctors has been set to 0 because, considering the low non-orthogonality, there was no need to introduce additional correction steps
- The relaxation factors have been tuned to minimize oscillations in the residuals and to reduce as much as possible the number of iterations needed to reach convergence

Computation of the KPI

- The **volume flow rate at the outlet** is directly computed by OpenFoam through a function that integrates on the outlet patch the component of the velocity normal to the surface (the same result can be obtained by summing the volumetric flux *phi* over the same patch).

```
flowRateOutlet
{
    libs ("libfieldFunctionObjects.so");
    type surfaceFieldValue;
    regionType patch;
    name outlet;
    operation areaNormalIntegrate;
    fields (U);
    //operation sum;
    //fields (phi);
    writeFields off;
    writePrecision 6;
    writeControl        writeTime;
    writeInterval       1;
    enabled             true;
}
```

Figure 5

- The **mean velocity at the outlet** is computed with the same function used for the flow rate; in this case, an average over the area is performed instead of an integration.

```
meanVelocityOutlet
{
    libs ("libfieldFunctionObjects.so");
    type surfaceFieldValue;
    regionType patch;
    name outlet;
    operation areaAverage;
    fields (U);
    writeFields off;
    writePrecision 6;
    writeControl        writeTime;
    writeInterval       1;
    enabled             true;
}
```

Figure 6

- The **shear stresses on the top and bottom wall** are directly computed by OpenFoam through a function that provides at each time step saved the x,y,z components of the shear stress vector. They are computed only on the bottom wall since the problem has been assumed to be symmetric. The output file is read by a Matlab script that extract the data and computes the norm of the vector.

```
bottomWallShearStress
{
    libs ("libfieldFunctionObjects.so");
    type            wallShearStress;
    patches         (bottomWall);
    writeField      off;
    writePrecision 6;
    writeControl    writeTime;
    writeInterval      1;
    enabled             true;
}
```

Figure 7

- The **static and total pressure distributions** along the nozzle are computed through a multi-step procedure. Groups of 13 probes have been distributed along the y direction in equally spaced positions along the longitudinal direction ($\Delta x_{probes} \approx 25\ mm$). At the end of each simulation, averages are performed with the measurements collected by probes with the same x coordinate by means of a Matlab script.

```
probes
{
    libs ("libsampling.so");
    type probes;
    writeControl writeTime;
    writeInterval 1;
    executeControl writeTime;
    executeInterval 1;
    writeField      off;
    writePrecision 6;
    fixedLocations false;
    fields (p U);
    enabled true;

    probeLocations
    (...)}
```

Figure 7

Post Processing

Results

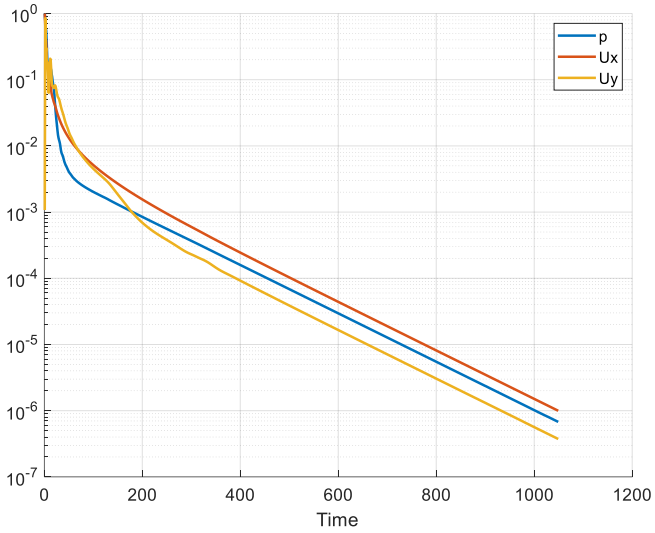


Figure 8

As shown in Figure 8all the residuals went below 10^{-6} after about 1000 iterations. This is considered as an acceptable indicator that convergence has been achieved.

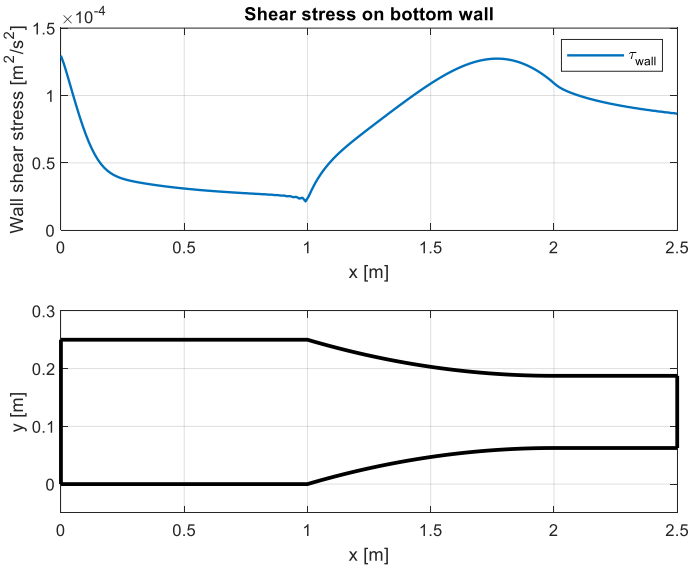


Figure 9

Figure 9, shear stresses on the bottom wall, main peaks and trends are reported

- **Initial peak:** since the flow goes from zero gradient to a sudden introduction of the walls, a peak in shear stress is visible.

- **Discontinuity:** it is induced by the geometrical discontinuity of the nozzle.
- **Constant growth:** along the radius of the arcs that define the nozzle, shear stress increases as well as velocity and its gradient.
- **Final value:** as the flow enters the second constant section area it tends to a constant value higher than the previous, approaching a fully developed laminar flow.

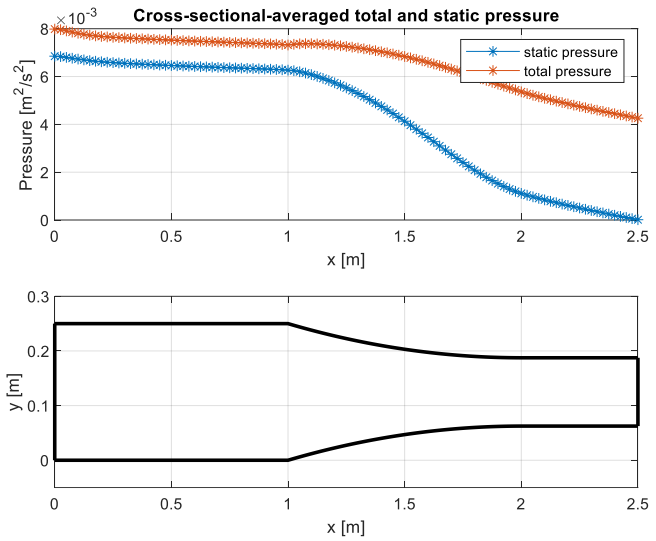
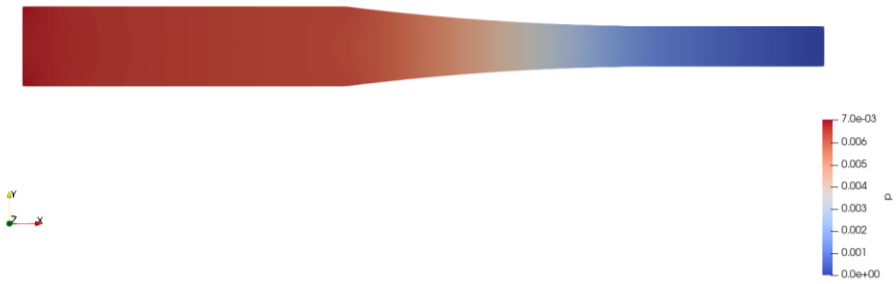


Figure 10

Figure 10, description of the pressure drops along the flow:

- **Initial drop:** initial little drop of total and static pressure due to the formation of the boundary layer
- **Central pronounced drop:** significant pressure drop in correspondence of the change in the cross-section area. Static pressure drop is mainly due to the increased flow speed, whereas the total pressure drop is mainly affected by the increased viscous effects.
- **Stabilization:** in the last constant smaller section both static and total pressure drops stabilized at a constant slope.
  The total pressure boundary condition is respected.

Pressure field



Velocity magnitude field

Figure 11

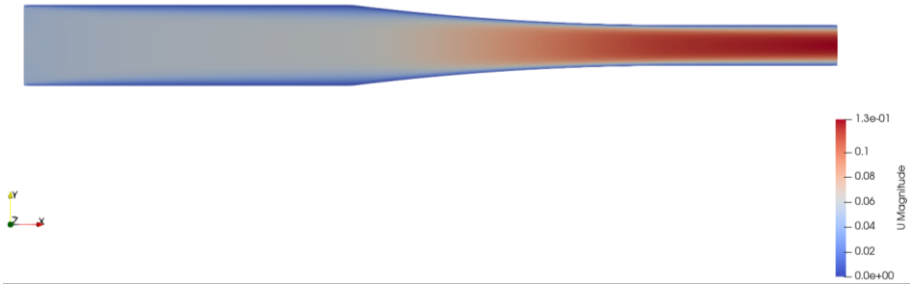


Figure 12

Kpi calculation

- Flow rate at the nozzle exit: **0.01197** m^2/s
    - The flow rate is low, in agreement with the given boundary conditions
- Velocity reduction coefficient: **0.7560**
    - Average speed at the nozzle exit: 0.09562 m/s
    - Isentropic speed at the nozzle exit: 0.126491 m/s
    - The reduction of velocity in comparison with the ideal case for laminar flow is considerable, and it is associated to the following phenomena :
        - Viscous dissipation induced by the wall neglected by the isentropic model
        - Thickness of the boundary layer

Grid independence study

Mesh	sk	Non-hort		aspect-ratio	n celle	it	KPI					
		max	avg				Q[m^2/s]	Vmean [m/s]	phi	delta Q [%]	delta Vmean [%]	delta phi [%]
Mesh00	**0,0961681**	**6,79**	**1,48**	**2,51**	**5000,00**	**1049,00**	**0,0120**	**0,0958**	**0,7575**	**0,00%**	**0,00%**	**0,00%**
~~Mesh01~~	~~0.0999192~~	~~6,59~~	~~1,47~~	~~2,46~~	~~2144,00~~	~~3000,00~~	~~0,0133~~	~~0,1066~~	~~0,8431~~	~~-11,31%~~	~~-11,30%~~	~~-11,30%~~
Mesh02	0.0959731	6,91	1,49	2,54	11400,00	1523,00	0,0120	0,0957	0,7566	10,26%	10,26%	10,26%
Mesh03	0.0950228	7,00	1,50	2,58	26100,00	2208,00	0,0120	0,0957	0,7562	0,06%	0,05%	0,05%
Mesh04	0.0944164	7,06	1,50	2,61	59488,00	3217,00	0,0120	0,0956	0,7560	0,05%	0,03%	0,03%

Figure 13

Since the starting mesh was already reasonably refined, reducing the number of cells as first attempt seemed a reasonable option, however, this led to a non-converging simulation. For this reason, the number of cells in x,y direction was increased by a factor 1.4 for each refinement step.

The simulation is considered grid independent when the difference between consecutive refinements is above a certain threshold. In order to check percentage differences of one refinement iteration with its previous one are considered. Addressing as ϕ_i the value of a generic KPI computed with mesh refinement level i and as ϕ_{i-1} the value of a generic KPI computed with the previous refinement level $i-1$, the delta coefficients have been computed as $\frac{\phi_{i-1}-\phi_i}{\phi_{i-1}}$.

Turbulent flow

Simulations with wall functions

Boundary conditions

- **Pressure $\boldsymbol{p}$** $[m^2/s^2]$:

Inlet	totalPressure (uniform 107.78)
Outlet	fixedValue (uniform 0)
topWall	zeroGradient
bottomWall	zeroGradient
Front and Back	empty

  - The value at the outlet was provided by the assignment

- **Velocity $\boldsymbol{U}$** $[m/s]$:

Inlet	pressureInletVelocity (uniform 14.68 0 0)
Outlet	zeroGradient
topWall	noSlip
bottomWall	noSlip
Front and back	empty

  - The value at the inlet was provided by the assignment

- **Turbulent kinetic energy κ** $[m^2/s^2]$:

Inlet	turbulentIntensityKineticEnergyInlet
Outlet	zeroGradient
topWall	kqRWallFunction
bottomWall	kqRWallFunction
Front and back	empty

- The value at the inlet is computed directly by OpenFoam starting from the value of the turbulence intensity ($I = \frac{u'}{U} = 5\%$) provided by the assignment
- On the walls, the use of wall functions is imposed.

- **Turbulent frequency** ω $[1/s]$:

Inlet	turbulentMixingLengthDissipationRateInlet
Outlet	zeroGradient
topWall	omegaWallFunction
bottomWall	omegaWallFunction
Front and back	empty

- The value at the inlet is computed directly by OpenFoam starting from the value of the mixing length ($l_T = \frac{1}{10} H_{in} = 0.025m$) provided by the assignment
- On the walls, the use of wall functions is imposed.

- **Turbulent kinematic viscosity** (*nut)* ν_T $[m^2/s]$:

Inlet	calculated (uniform 0.0123)
Outlet	zeroGradient
topWall	nutkWallFunction
bottomWall	nutkWallFunction
Front and back	empty

Setup of the simulation

Mesh

In this turbulent flow case study, a mesh was created and designed to respect the target of 30 < y+ < 300,in order to make a correct use of wall functions.

Since the use of wall function condition is limiting the minimum dimension of the cells at the wall, grading was significantly reduced to improve the global refinement without reducing wall cells dimension.

Below, an image of the most refined mesh and a detail near the wall of it are reported (Figure 14 and Figure 15) together with the chekMesh output.

Figure 14

Figure 15

Number of cells	79170
Skeweness (max)	0,13
Non orthogonality (max/mean)	7,03° / 1,36°
Aspect ratio	2,54

Domain discretization schemes (fvSchemes)

Considering the turbulence model that is being used, the following terms need to be discretized

```
ddtSchemes
{
    default         steadyState;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      bounded Gauss linearUpwind grad(U);
    div(phi,omega) bounded Gauss upwind; // this models U*grad(w)
    div(phi,k)      bounded Gauss upwind; // this models rho*U*grad(k)
    div((nuEff*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default        Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         corrected;
}

wallDist
{
    method meshWave;
}
```

Figure 16

Convective flux terms for k and ω: first-order accuracy schemes were employed because, when combined with the second-order scheme for the convective flux of U, they reduced the number of iterations required for convergence. Higher-order schemes were tested, they often led to diverging simulations or an unacceptably high number of iterations. This behavior might be attributed to oscillations that can arise with second-order schemes.

Resolution schemes (fvSolution)

Considering the turbulence model that is being used, an algorithm to solve the linear system associated with the $k-\omega\ SST$ equations is needed. Once again, a multi-grid method was used. In the same way, also some values for the residuals and for the relaxation factors needed to be specified. They have been tuned with the same criteria described previously.

```
    "k|omega"
    {
        solver          GAMG;
        smoother        GaussSeidel;
        tolerance       1e-10;
        relTol          0;
    }
SIMPLE
{
    nNonOrthogonalCorrectors 0;

    residualControl
    {
        p               1e-5;
        U               1e-5;
        omega          1e-5;
        k               1e-5;
    }
}

relaxationFactors
{
    fields
    {
        p               0.3;
    }
    equations
    {
        U               0.7;
        omega           0.6;
        k               0.6;
    }
}
```

Figure 17

Computation of the KPI

The KPIs that need to be computed haven't changed, therefore the procedure followed is the same for the laminar flow.

Simulations without wall functions

Boundary conditions

The boundary conditions for pressure and velocity haven't changed

- **Turbulent kinetic energy** $\kappa\ [m^2/s^2]$:

Inlet	turbulentIntensityKineticEnergyInlet
Outlet	zeroGradient
topWall	fixedValue (uniform 0)
bottomWall	fixedValue (uniform 0)
Front and back	empty

- The value at the inlet is computed directly by OpenFoam starting from the value of the turbulence intensity ($I = \frac{u'}{U} = 5\%$) provided by the assignment
- On the walls, the use of wall functions is imposed

- **Turbulent frequency** $\omega\ [1/s]$:

Inlet	turbulentMixingLengthDissipationRateInlet
Outlet	zeroGradient
topWall	fixedValue (uniform 1e6)
bottomWall	fixedValue (uniform 1e6)
Front and back	empty

- The value at the inlet is computed directly by OpenFoam starting from the value of the mixing length ($l_T = \frac{1}{10} H_{in} = 0.025m$) provided by the assignment

- **Turbulent kinematic viscosity (*nut*)** $\nu_T\ [m^2/s]$:

Inlet	calculated (uniform 0.0123)
Outlet	zeroGradient
topWall	fixedValue (uniform 0)
bottomWall	fixedValue (uniform 0)
Front and back	empty

Setup of the simulation

Mesh

A mesh was created with the primary goal of avoiding the use of wall functions. To achieve this, the first cell was set to have a y+ value smaller than 1, and 10 cells were used to resolve the region up to y+=10 . The desired y+ target was met through the application of a simple grading in the y-direction.

The constraint imposed by the y+ target led to a high aspect ratio, which is not ideal. However, the simulation successfully reached convergence.

Here is reported the Mesh (Figure 18), some detail view of the refinement made in the wall region(Figure 19 Figure 20) and relative checkMesh result.

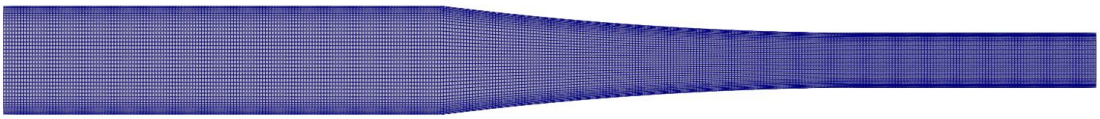


Figure 18

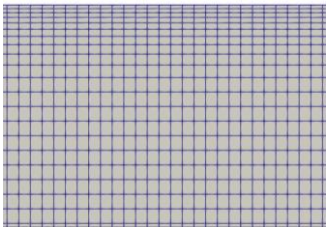


Figure 19

Figure 20

Number of cells	22500
Skeweness (max)	0,156
Non orthogonality (max/mean)	7,14° / 2,07°
Aspect ratio	222,83

Domain discretization schemes (fvSchemes)

The turbulence model has not changed and in the same way, also the discretization schemes haven't changed.

Resolution schemes (fvSolution)

The turbulence model has not changed and in the same way, also the resolution algorithms for linear systems haven't changed.

Computation of the KPIs

The KPIs that need to be computed haven't changed, therefore the procedure followed is the same for the laminar flow

Post Processing

Results

Residual convergence

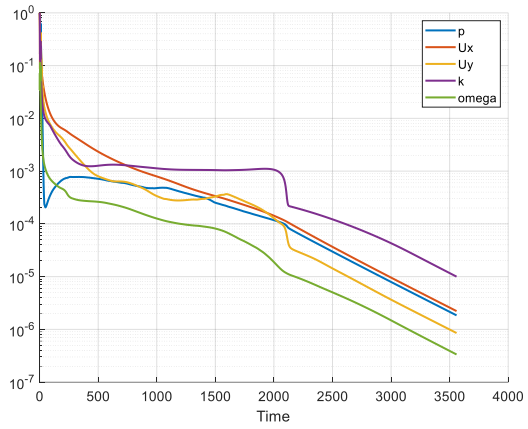


Figure 21 (with wall functions)

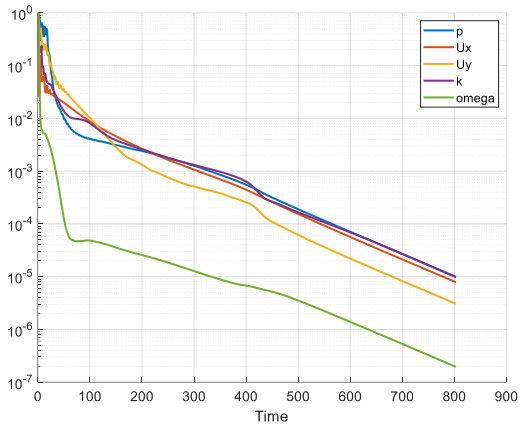


Figure 22 (without wall functions)

- With wall functions: all values drop below the chosen threshold (10^{-6}) after about 3500 iterations
- Without wall functions: all values drop below the chosen threshold (10^{-6}) after about 800 iterations

Shear stress distribution

(bottom wall considered)

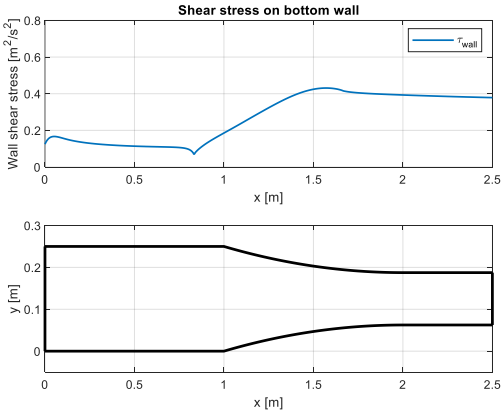


Figure 23 (with wall functions)

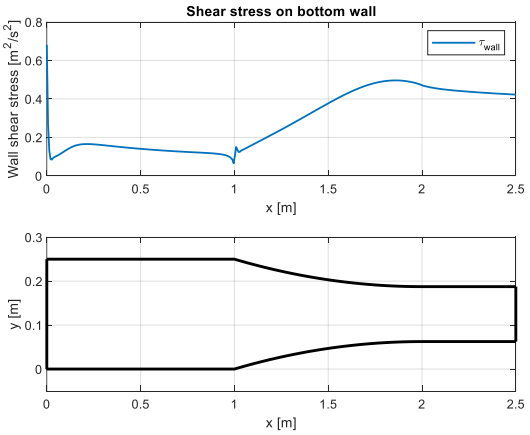


Figure 24 (without wall functions)

Differences between the two cases analyzed:

- In the first constant cross section part, the shear stress associated with the boundary layer development is higher in the case with wall functions.
- In the case without wall functions the peak of shear stress at the end of the converging section is higher than the case with wall functions.

These differences are related to the fact that the solution resolved to the wall can be considered as more accurate than the one where wall functions are used.

Static and total pressure distribution

(bottom wall considered)

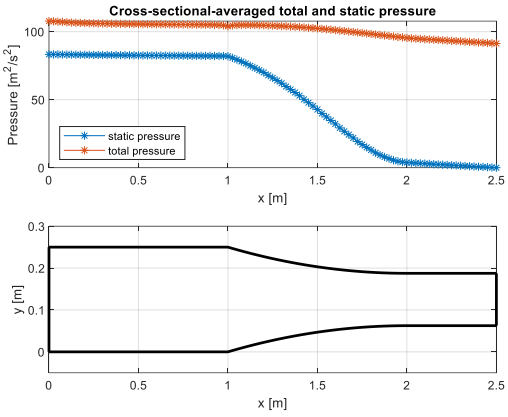


Figure 25 (with wall functions)

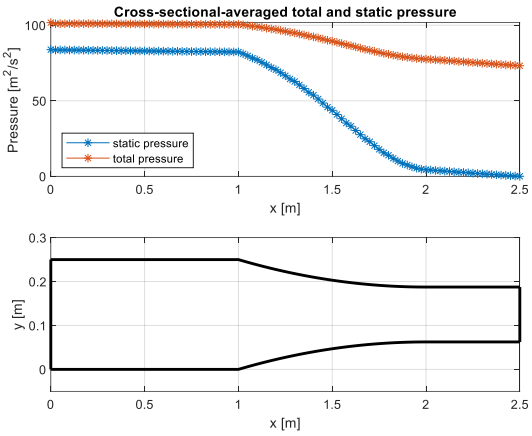


Figure 26 (without wall functions)

Since the difference between the two cases is in the boundary layer modelling approach, the difference is concentrated in the near wall region, so the cross section average value is not affected.

Non-dimensional wall distance

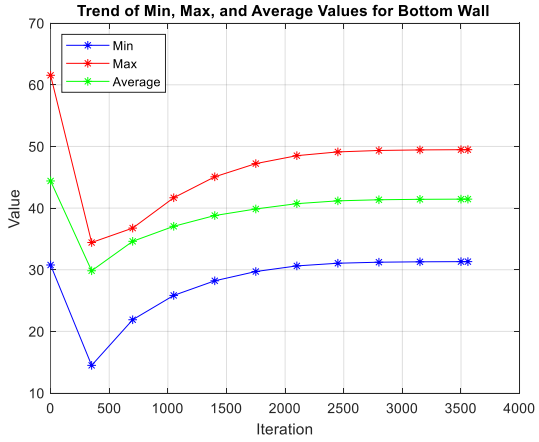


Figure 27 (with wall functions)

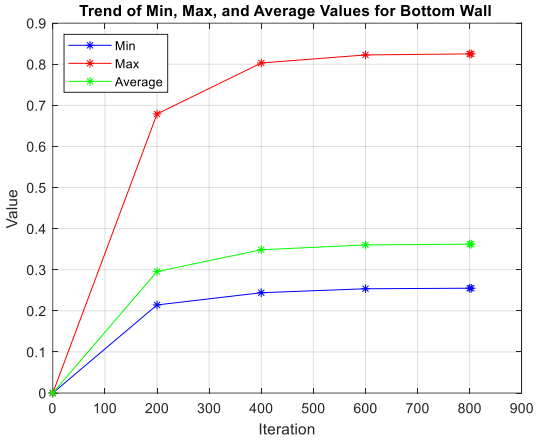


Figure 28 (without wall functions)

The value settles as requested by the two specific models:

- Without wall functions: $y^+ = 30 \div 300$
- With wall functions: $y^+ < 1$

Pressure field

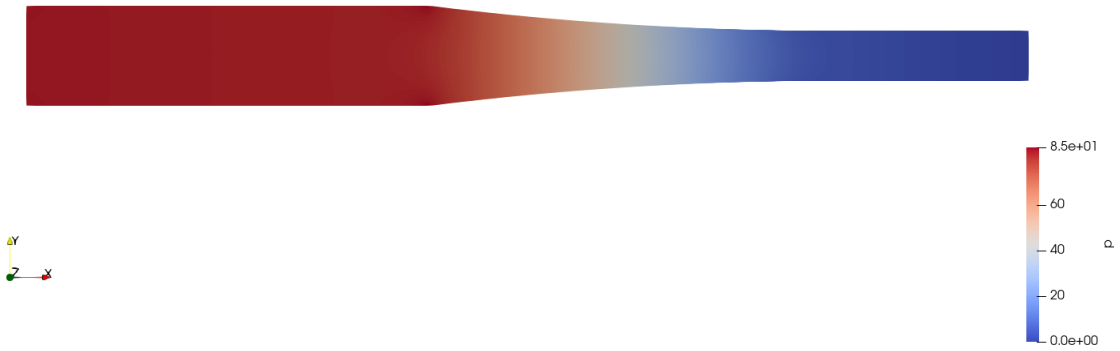


Figure 29 (with wall functions)

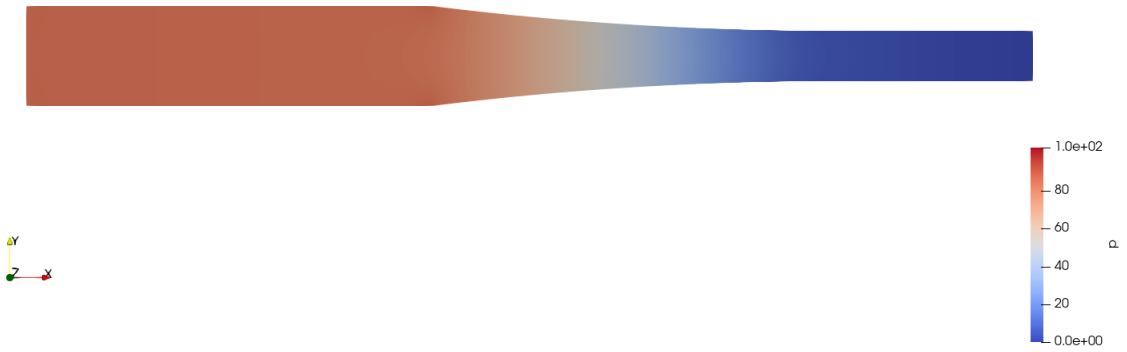


Figure 30 (without wall functions)

Velocity magnitude field

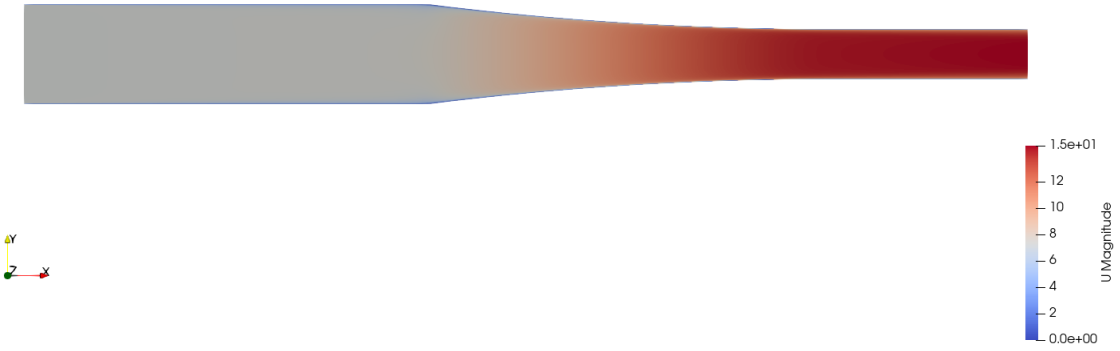


Figure 31 (with wall functions)

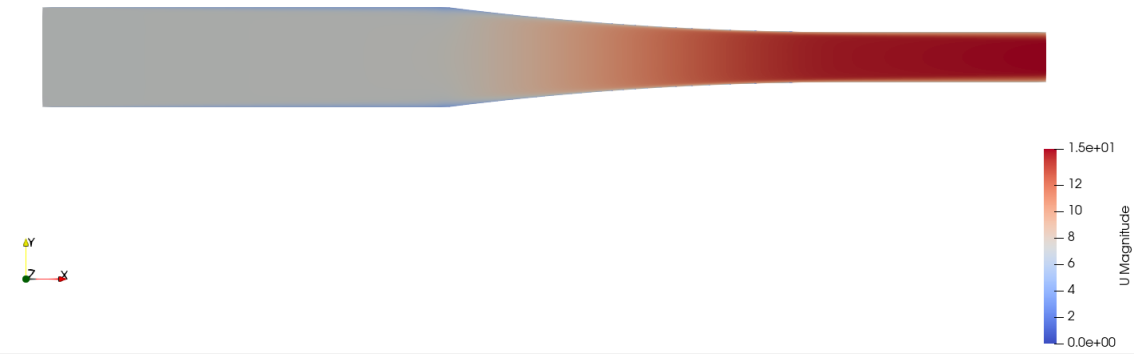


Figure 32 (without wall functions)

Kpi calculation

Case **with** wall functions:

- Flow rate at the nozzle exit**: 1.749670** m^2/s
- Velocity reduction coefficient: **0.953356**
    - Average speed at the nozzle exit: 13.997360 m/s
    - Isentropic speed at the nozzle exit: 14.682189 m/s

Case **without** wall functions:

- Flow rate at the nozzle exit: **1. 731636** m^2/s
    - The flow rate is low, in agreement with the given boundary conditions
- Velocity reduction coefficient: **0. 943530**
    - Average speed at the nozzle exit: 13. 853090 m/s
    - Isentropic speed at the nozzle exit: 14. 682189 m/s

The velocity reduction coefficient is significantly higher than the laminar case for the following reasons:

- Partial influence of the different boundary conditions.
- The thinner boundary layer in turbulent flows, because of the higher Reynold number, implies a greater portion of the flow cross-section is sustaining greater velocities.
- Energy losses in turbulent flows due to wall frictions are distributed differently and tend to have a less pronounced impact on the core flow velocity, that is close to the isentropic case.

Grid independence study

(it was done only for the case with wall functions as requested)

ES 1 B	sk	non-hort		aspect-ratio	n celle	it	KPI					
		Max	avg				Q(m^2/s)	Vmean(m/s)	phi	delta_Q [%]	delta_Vmean [%]	delta_phi [%]
Mesh00	**0,0982063**	**7,02**	**1,50**	**2,57**	**33800,00**	**1868,000**	**1,7510000**	**14,01**	**0,95**	**0,00%**	**0,00%**	**0,00%**
Mesh01	0,11	7,01	1,51	3,61	17390,00	1143,00	1,7534	14,03	0,96	-0,14%	-0,09%	-0,09%
Mesh02	0,13	7,03	1,36	2,54	79170,00	3559,00	1,7497	13,9973	0,9534	0,08%	0,12%	0,12%

Figure 33