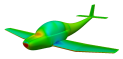


5. Aerodynamics of a 2D airfoil NACA 23012

5.0.12 Description of the case

The current chapter studies flow around a 2D airfoil. Although it encompasses the study of external flow, there are significant differences with Chapter 3, such as the fact that an airfoil is a streamlined body, and thus the behaviour of the flow around it changes substantially. Besides the physics of the problem, Chapter 5 includes new features as the introduction of turbulence models and the creation of a mesh without using blockMesh.

5.0.13 Hypotheses

- Incompressible flow
- Turbulent flow
- Newtonian flow
- Bidimensional flow ($\frac{\partial}{\partial z} = 0$)
- Negligible gravitatory effects
- Sea level conditions
- RAS turbulence modelling with wall functions

5.0.14 Physics of the problem

The problem deals with an airfoil NACA 23012 flying at a speed of $V = 45$ m/s at sea level. As the medium is air ($\nu = 1.5 \times 10^{-5}\ m^2/\text{s}$), the Reynolds number is

$$Re = \frac{Vc}{\nu} = \frac{45 \cdot 1}{1.5 \times 10^{-5}} = 3 \times 10^6$$

where c is the chord of the airfoil and is equal to 1 m. The chord is the imaginary straight line joining the leading and trailing edges. The problem statement is shown at Figure 5.1.

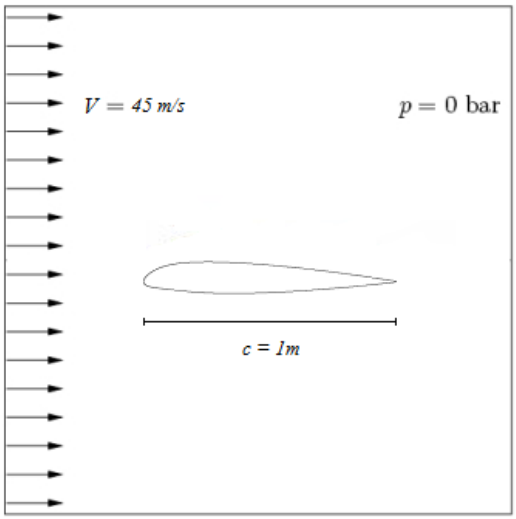


Figure 5.1: Airfoil NACA 23012 flying at 45 m/s and ambient pressure

The airfoil NACA 23012 has been chosen because of its high aerodynamic performance at low flying velocities. It presents a high maximum lift coefficient ($C_{l_{max}}$) and a high stall angle (α_{stall}). Its geometric characteristics are:

- Thickness: 12%
- Maximum thickness position: 30%
- Maximum chamber: 1.83%
- Maximum chamber position: 13%

The shape of the NACA 23012 is shown at Figure 5.2.

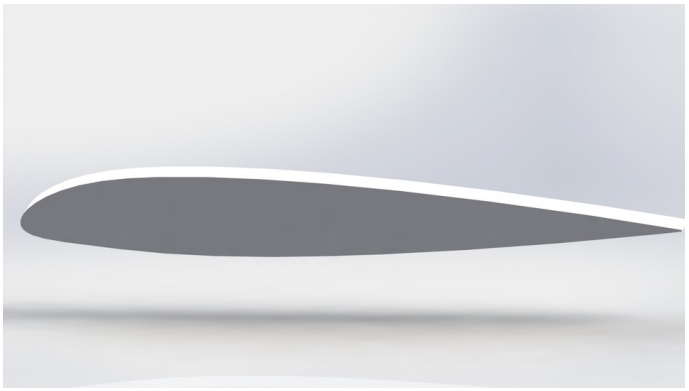


Figure 5.2: NACA 23012

In the current chapter there is no easy analytical solution to describe the behaviour of the fluid. However, it is necessary to keep in mind the main equations and dimensionless numbers involved in the problem:

The continuity equation,

$$\nabla \cdot \mathbf{U} = 0 \tag{5.1}$$

The momentum equation,

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\nabla^2 \mathbf{U} \tag{5.2}$$

The Reynolds number,

$$Re = \frac{c|\mathbf{U}|}{\nu} \tag{5.3}$$

The lift coefficient (dimensionless force perpendicular to the flow),

$$C_l = \frac{L}{\frac{1}{2}\rho|\mathbf{U}|^2 S} \tag{5.4}$$

The drag coefficient (dimensionless force parallel to the flow),

$$C_d = \frac{D}{\frac{1}{2}\rho|\mathbf{U}|^2 S} \tag{5.5}$$

The moment coefficient (dimensionless moment of the airfoil calculated at the aerodynamic center),

$$C_m = \frac{M}{\frac{1}{2}\rho|\mathbf{U}|^2 Sc} \tag{5.6}$$

For a low velocity flow and a surface with a given rugosity, C_l and C_d depend on the angle of attack and the Reynolds number:

$$C_l = f(Re, \alpha) \qquad \text{or} \qquad C_d = f(Re, \alpha)$$

The angle of attack (α) is the angle between the chord of the airfoil and the direction of the fluid velocity.

At low values of α, there is a pressure gradient on the upper surface of the airfoil but it is not strong enough to detach the boundary layer. The flow around the airfoil is smooth, the drag is low and the lift excellent. The relation between the lift coefficient and the angle of attack is approximately linear. When α increases, the pressure gradient grows. Normally, at an angle of attack between 15º and 20º, the flow is completely detached on the upper surface. When it happens, the airfoil stalls.

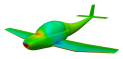
5.0.15 Pre-processing with $\alpha = 15^o$

The following codes contain the information to simulate the case with $Re = 3 \times 10^6$ using simpleFoam (steady-state solver for incompressible turbulent flow) and the Spalart-Allmaras turbulence model. As the current case includes new features and files, a general directory is going to be created with the name AirfoilCase and containing two files to create the mesh (without using blockMesh), as well as a subdirectoy containing the cases solved for different angles of attack (2DAirfoil). Within 2DAirfoil the names of the cases are going to be alpha0 ($\alpha = 0^o$), alpha5 ($\alpha = 5^o$), etc. In particular, the current pre-processing section contains the instructions to simulate the airfoil for $\alpha = 15^o$ (alpha15 case).

5.0.15.1 Mesh generation

As it happened with Chapter 3, the mesh is not going to be uniform. It is necessary to divide the mesh by regions, especially to provide a high refinement to the walls of the airfoil and downstream. However, in the current case, the geometry is more complex than a simple circular shape. Therefore, the mesh is going to be created with GNU Octave and Gmsh Mesh Generator. Then, it will be converted to *OpenFOAM*®.

The first step is to install these two softwares: GNU Octave is a high-level interpreted language, primarily intended for numerical computations. Gmsh Mesh Generator



contains 4 modules, for geometry description, meshing, solving and post-processing. They are both free software.

Secondly, it is necessary to create two gedit sheets. The first is going to be named *foilgmsh.m* and contains the code shown in the Appendix, Chapter A.

As it can be seen, this code was developed by an external author, who provided it selflessly. Before the instructions, it is explained how to use it and what parameters are necessary to define. The code can be entirely found in:

$$code - saturne.org/forum/old\_forums\_files/614676387/foilgmsh.m$$

Afterwards, a Script is going to be created to execute the previous code and introduce the output to Gmsh Mesh Generator. The results are going to be two new files, containing the created geometries and mesh. This mesh will be later converted to the $OpenFOAM^{®}$ format. The Script must be named *executeAirfoilMesher* and contains:

```
1  #!/bin/bash
2
3  archi="naca23012.dat" #File containing the coordinates of the airfoil (.dat
       extension)
4  alfa=15
5  yplus=100
6  eter="a"
7  Re=3000000
8  M=0.128
9  T0=300
10 N="[100, 40, 30, 30]"
11 bump=1
12 cd /home/<linuxUserName>/<Desktop>/AirfoilCase #If different, write how to access
       foilgmsh.m
13 octave --silent --eval "foilgmsh('$archi', $alfa, $yplus, '$eter', $Re, $M, $T0,
       '$N', $bump)" #Executes the foilgmsh function
14 gmsh -3 naca23012.dat.geo #Meshes with Gmsh Mesh Generator
```

So, once the programs have been installed and the gedit sheets containing the previous codes are ready:

1.- Make the Script executable as it was shown in Section 4.0.11.3

2.- Copy *foilgmsh.m* and *executeAirfoilMesher* within AirfoilCase

3.- Copy a file with the coordinates of the airfoil (in this case the NACA 23012) with the extension *.dat* named *naca23012.dat* within AirfoilCase. It can be easily found in Internet

4.- Access AirfoilCase and type in the terminal:

```
./executeAirfoilMesher
```

5.- Select one of the created files (*naca23012.dat.msh*) and copy it inside an empty directory named alpha15

6.- Copy a *system* folder from another case as it is necessary to convert the mesh to $OpenFOAM^{®}$ (it does not matter what time-control setting it contains; it will be later correctly defined)

7.- Access 2DAirfoil/alpha15

8.- Type:

```
gmshToFoam naca23012.dat.msh
```

9.- The *constant* directory appears

10.- Acces *constant/polyMesh/boundary*, change the name of the first patch (symmetry) to frontAndBack and define it as empty instead of patch (both, type and physicalType). Keep the second name (airfoil) but change its patch type to wall. Finally, change the third patch name (walls) to topAndBottom and its patch type to symmetryPlane instead of patch

11.- The mesh is ready

The mesh of the alpha15 case is:

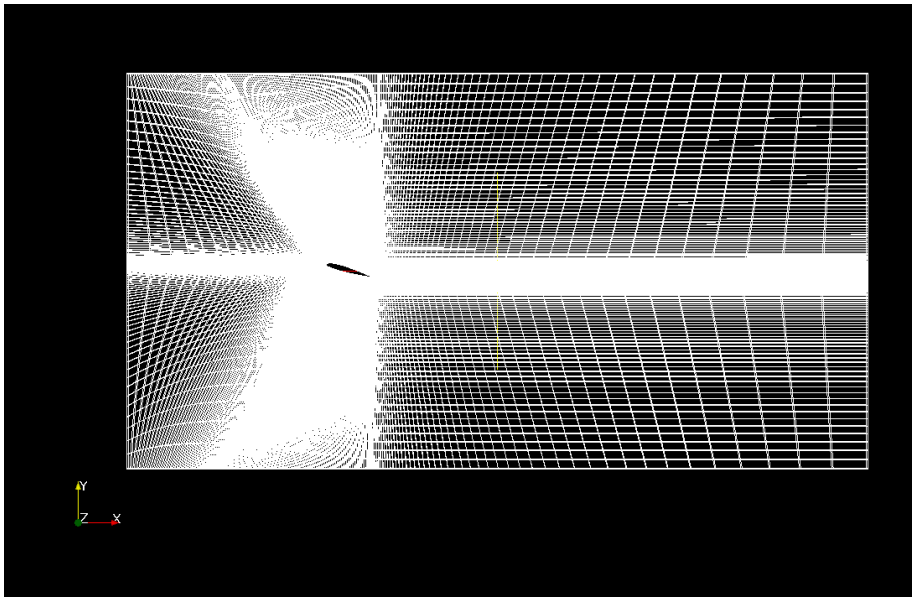


Figure 5.3: Global view of the mesh of the alpha15 case

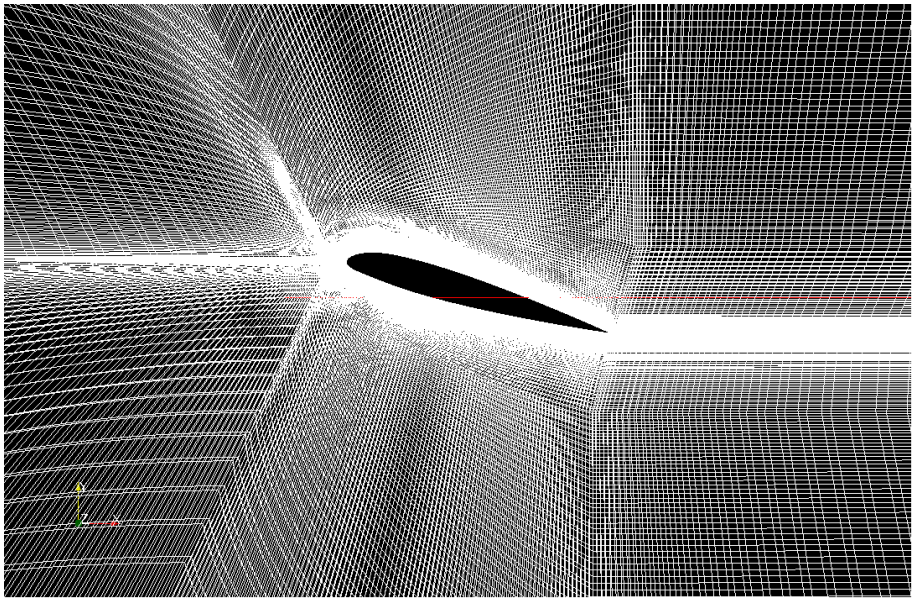


Figure 5.4: Airfoil shape in the mesh of the alpha15 case

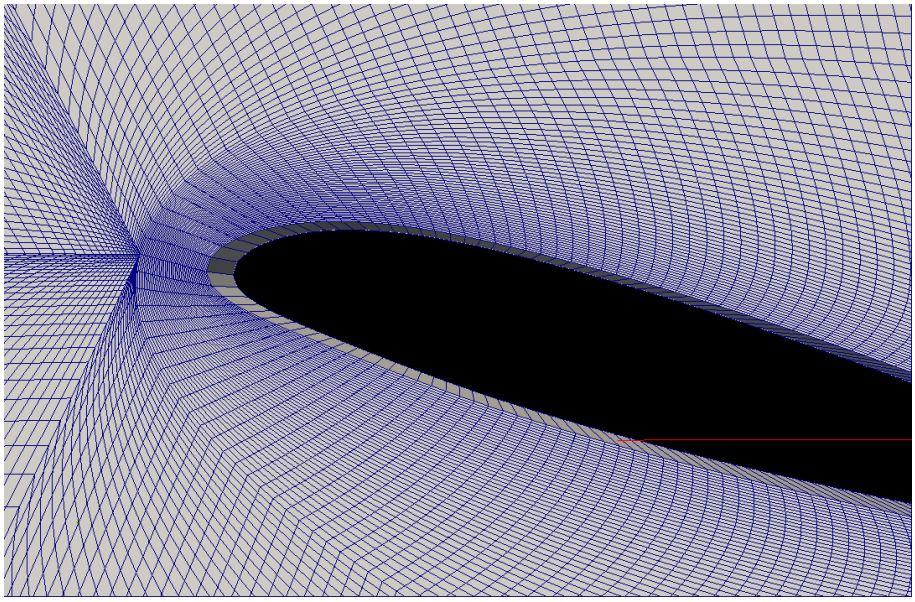


Figure 5.5: Detail of the mesh grading at the walls of the alpha15 case

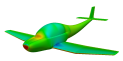
Advice:

> The degree of refinement of the mesh and its relation to the treatment of the turbulent boundary layer is discussed in Section 5.0.17.2. It determines the value written in the fifth line of *executeAirfoilMesher*

5.0.15.2 Boundary and initial conditions

Besides the *p* and *U* files, as the case is set turbulent, two new files are necessary to be created. Their names are *nut* and *nuTilda*. These dictionaries contain the boundary conditions of the parameters used to implement the Spalart-Allmaras turbulence model. Before introducing the user to their calculation, the boundary conditions for p and $\mathbf{U}$ are:

```
1  /*---------------------------------*- C++ -*----------------------------------*\
2  | =========                 |                                                 |
3  | \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
4  |  \\    /   O peration     | Version:  2.2.1                                 |
5  |   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
6  |    \\/     M anipulation  |                                                 |
7  \*---------------------------------------------------------------------------*/
8  FoamFile
9  {
10     version     2.0;
11     format      ascii;
12     class       volScalarField;
13     object      p;
```



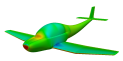
```
14  }
15  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
16
17  dimensions      [0 2 -2 0 0 0 0];
18
19  internalField   uniform 0;
20
21  boundaryField
22  {
23      inlet
24      {
25          type            freestreamPressure;
26      }
27
28      outlet
29      {
30          type            freestreamPressure;
31      }
32
33      airfoil
34      {
35          type            zeroGradient;
36      }
37
38      frontAndBack
39      {
40          type            empty;
41      }
42
43      topAndBottom
44      {
45          type            symmetryPlane;
46      }
47  }
48
49  // ************************************************************************* //
```

```
1   /*--------------------------------*- C++ -*----------------------------------*\
2   | =========                 |                                                 |
3   | \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
4   |  \\    /   O peration     | Version:  2.2.1                                 |
5   |   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
6   |    \\/     M anipulation  |                                                 |
7   \*---------------------------------------------------------------------------*/
8   FoamFile
9   {
10      version     2.0;
11      format      ascii;
12      class       volVectorField;
13      object      U;
14  }
15  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
16
17  dimensions      [0 1 -1 0 0 0 0];
18
19  internalField   uniform (45 0 0);
```

```
20
21  boundaryField
22  {
23      inlet
24      {
25          type            freestream;
26          freestreamValue uniform (45 0 0);
27      }
28
29      outlet
30      {
31          type            freestream;
32          freestreamValue uniform (45 0 0);
33      }
34
35      airfoil
36      {
37          type            fixedValue;
38          value           uniform (0 0 0);
39      }
40
41      frontAndBack
42      {
43          type            empty;
44      }
45
46      topAndBottom
47      {
48          type            symmetryPlane;
49      }
50  }
51
52  // ************************************************************************* //
```

The dictionaries for *nut* and *nuTilda* (they must be saved within *0* with these names) are:

```
1   /*--------------------------------*- C++ -*----------------------------------*\
2   | =========                 |                                                 |
3   | \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
4   |  \\    /   O peration     | Version:  2.2.1                                 |
5   |   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
6   |    \\/     M anipulation  |                                                 |
7   \*---------------------------------------------------------------------------*/
8   FoamFile
9   {
10      version     2.0;
11      format      ascii;
12      class       volScalarField;
13      object      nut;
14  }
15  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
16
17  dimensions      [0 2 -1 0 0 0 0];
18
```



```
19  internalField   uniform 0.0386;
20
21  boundaryField
22  {
23      inlet
24      {
25          type            freestream;
26          freestreamValue uniform 0.0386;
27      }
28
29      outlet
30      {
31          type            freestream;
32          freestreamValue uniform 0.0386;
33      }
34
35      airfoil
36      {
37          type            nutUSpaldingWallFunction;
38          value           uniform 0;
39      }
40
41      frontAndBack
42      {
43          type            empty;
44      }
45
46      topAndBottom
47      {
48          type            symmetryPlane;
49      }
50  }
51
52  // ************************************************************************* //
```

```
1   /*--------------------------------*- C++ -*----------------------------------*\
2   | =========                 |                                                 |
3   | \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
4   |  \\    /   O peration     | Version:  2.2.1                                 |
5   |   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
6   |    \\/     M anipulation  |                                                 |
7   \*---------------------------------------------------------------------------*/
8   FoamFile
9   {
10      version     2.0;
11      format      ascii;
12      class       volScalarField;
13      object      nuTilda;
14  }
15  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
16
17  dimensions      [0 2 -1 0 0 0 0];
18
19  internalField   uniform 0.1929;
20
21  boundaryField
```

```
22  {
23      inlet
24      {
25          type            freestream;
26          freestreamValue uniform 0.1929;
27      }
28
29      outlet
30      {
31          type            freestream;
32          freestreamValue uniform 0.1929;
33      }
34
35      airfoil
36      {
37          type            fixedValue;
38          value           uniform 0;
39      }
40
41      frontAndBack
42      {
43          type            empty;
44      }
45
46      topAndBottom
47      {
48          type            symmetryPlane;
49      }
50  }
51
52  // ************************************************************************* //
```

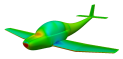
ν_t (*nut*) and $\tilde{\nu}_t$ (*nuTilda*) are parameters required to simulate the case with a turbulence model (as it was said, for this case it is used the Spalart-Allmaras one-equation model). Mathematically, they are determined as:

$$\tilde{\nu_t} = \sqrt{\frac{3}{2}}(|\mathbf{U}|Il) \tag{5.7}$$

where $|\mathbf{U}|$ is the mean flow velocity, I is the turbulence intensity and l is the turbulent length scale. They can be computed as:

$$I \approx 5\%$$

$$l \approx 0.07 \cdot c = 0.07$$



With these assumptions, $\tilde{\nu_t} = 0.1929$. Then, a convenient option is to set $\tilde{\nu_t} = 5\nu_t$ in the freestream. The model then provides fully turbulent results and any regions like boundary layers that contain shear become fully turbulent.

5.0.15.3 Physical properties

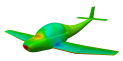
```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      transportProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

transportModel  Newtonian;

rho             rho [ 1 -3 0 0 0 0 0 ] 1.225;

nu              nu [ 0 2 -1 0 0 0 0 ] 1.5e-05;

CrossPowerLawCoeffs
{
    nu0             nu0 [ 0 2 -1 0 0 0 0 ] 1e-06;
    nuInf           nuInf [ 0 2 -1 0 0 0 0 ] 1e-06;
    m               m [ 0 0 1 0 0 0 0 ] 1;
    n               n [ 0 0 0 0 0 0 0 ] 1;
}

BirdCarreauCoeffs
{
    nu0             nu0 [ 0 2 -1 0 0 0 0 ] 1e-06;
    nuInf           nuInf [ 0 2 -1 0 0 0 0 ] 1e-06;
    k               k [ 0 0 1 0 0 0 0 ] 0;
    n               n [ 0 0 0 0 0 0 0 ] 1;
}

// ************************************************************************* //
```

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
```

```
7   \*---------------------------------------------------------------------------*/
8   FoamFile
9   {
10      version     2.0;
11      format      ascii;
12      class       dictionary;
13      location    "constant";
14      object      RASProperties;
15  }
16  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
17
18  RASModel        SpalartAllmaras;
19
20  turbulence      on;
21
22  printCoeffs     on;
23
24  // ************************************************************************* //
```

5.0.15.4 Control

```
1   /*---------------------------------*- C++ -*----------------------------------*\
2   | =========                 |                                                 |
3   | \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
4   |  \\    /   O peration     | Version:  2.2.1                                 |
5   |   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
6   |    \\/     M anipulation  |                                                 |
7   \*---------------------------------------------------------------------------*/
8   FoamFile
9   {
10      version     2.0;
11      format      ascii;
12      class       dictionary;
13      location    "system";
14      object      controlDict;
15  }
16  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
17
18  application         simpleFoam;
19
20  startFrom           startTime;
21
22  startTime           0;
23
24  stopAt              endTime;
25
26  endTime             10000;
27
28  deltaT              1;
29
30  writeControl        timeStep;
31
32  writeInterval       100;
33
34  purgeWrite          0;
```

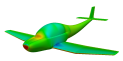


```
35
36  writeFormat          ascii;
37
38  writePrecision       7;
39
40  writeCompression     off;
41
42  timeFormat           general;
43
44  timePrecision        6;
45
46  runTimeModifiable    true;
47
48  functions
49  (
50  forces
51  {
52  type forces;
53  functionObjectLibs ("libforces.so");
54  patches (airfoil);
55  pName p;
56  UName U;
57  rhoName rhoInf;
58  rhoInf 1.225;
59  CofR (0.2704 -0.0012 4.0128);
60  outputControl timeStep;
61  outputInterval 100;
62  }
63  forceCoeffs
64  {
65  type forceCoeffs;
66  functionObjectLibs ("libforces.so");
67  patches (airfoil);
68  pName p;
69  UName U;
70  rhoName rhoInf;
71  rhoInf 1.225;
72  CofR (0.2704 -0.0012 4.0128);
73  liftDir (0 1 0);
74  dragDir (1 0 0);
75  pitchAxis (0 0 1);
76  magUInf 45; //
77  lRef 1; //
78  Aref 0.1; //
79  outputControl timeStep;
80  outputInterval 100;
81  }
82  );
83
84  // ************************************************************************* //
```

5.0.15.5 Discretization and linear-solver settings

```
1  /*--------------------------------*- C++ -*----------------------------------*\
2  | =========                 |                                                 |
```

```
3   | \\      /  F ield          | OpenFOAM: The Open Source CFD Toolbox           |
4   |  \\    /   O peration      | Version:  2.2.1                                 |
5   |   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
6   |    \\/     M anipulation  |                                                 |
7   \*---------------------------------------------------------------------------*/
8   FoamFile
9   {
10      version     2.0;
11      format      ascii;
12      class       dictionary;
13      location    "system";
14      object      fvSchemes;
15  }
16  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
17
18  ddtSchemes
19  {
20      default         steadyState;
21  }
22
23  gradSchemes
24  {
25      default         Gauss linear;
26      grad(p)         Gauss linear;
27      grad(U)         Gauss linear;
28  }
29
30  divSchemes
31  {
32      default           none;
33      div(phi,U)        Gauss linearUpwind grad(U);
34      div(phi,nuTilda)  Gauss linearUpwind grad(nuTilda);
35      div((nuEff*dev(T(grad(U))))) Gauss linear;
36  }
37  laplacianSchemes
38  {
39      default         none;
40      laplacian(nuEff,U) Gauss linear corrected;
41      laplacian((1|A(U)),p) Gauss linear corrected;
42      laplacian(DnuTildaEff,nuTilda) Gauss linear corrected;
43      laplacian(1,p) Gauss linear corrected;
44  }
45  interpolationSchemes
46  {
47      default        linear;
48      interpolate(U) linear;
49  }
50
51  snGradSchemes
52  {
53      default         corrected;
54  }
55
56  fluxRequired
57  {
58      default         no;
59      p               ;
```

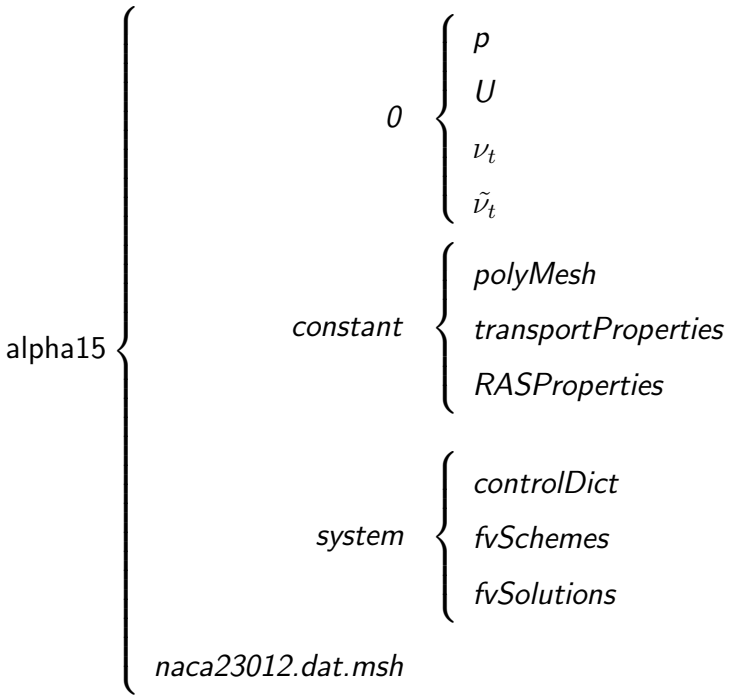


```
60  }
61
62  // ************************************************************************* //
```

```
1   /*--------------------------------*- C++ -*----------------------------------*\
2   | =========                 |                                                 |
3   | \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
4   |  \\    /   O peration     | Version:  2.2.1                                 |
5   |   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
6   |    \\/     M anipulation  |                                                 |
7   \*---------------------------------------------------------------------------*/
8   FoamFile
9   {
10      version     2.0;
11      format      ascii;
12      class       dictionary;
13      location    "system";
14      object      fvSolution;
15  }
16  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
17
18  solvers
19  {
20      p
21      {
22          solver           GAMG;
23          tolerance        1e-06;
24          relTol           0.1;
25          smoother         GaussSeidel;
26          nPreSweeps       0;
27          nPostSweeps      2;
28          cacheAgglomeration true;
29          nCellsInCoarsestLevel 10;
30          agglomerator     faceAreaPair;
31          mergeLevels      1;
32      }
33
34      U
35      {
36          solver           smoothSolver;
37          smoother         GaussSeidel;
38          nSweeps          2;
39          tolerance        1e-08;
40          relTol           0.1;
41      }
42
43      nuTilda
44      {
45          solver           smoothSolver;
46          smoother         GaussSeidel;
47          nSweeps          2;
48          tolerance        1e-08;
49          relTol           0.1;
50      }
51  }
52
```

```
53  SIMPLE
54  {
55          nCorrectors                 1;
56          nNonOrthogonalCorrectors    2;
57          pRefCell                    0;
58          pRefValue                   0;
59          residualControl
60      {
61          p                           1e-5;
62          U                           1e-5;
63          nuTilda                     1e-5;
64      }
65  }
66       relaxationFactors
67      {
68
69       fields
70      {
71          p                           0.3;
72      }
73       equations
74      {
75          U                           0.7;
76          nuTilda                     0.7;
77      }
78  }
79
80  // ************************************************************************* //
```

At the end of the pre-processing, the structure of directories, subdirectories and files within alpha15 should be as follows:



5.0.16 Post-processing

5.0.16.1 Results of the simulation for $\alpha = 15^{o}$

The velocity field:

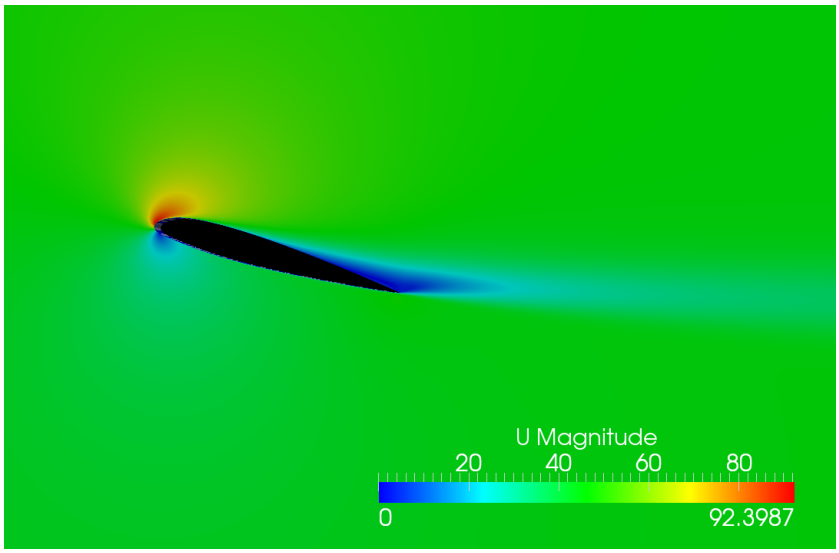


Figure 5.6: Velocity field around the NACA 23012 at $\alpha = 15^{o}$ (m/s)

The pressure field:

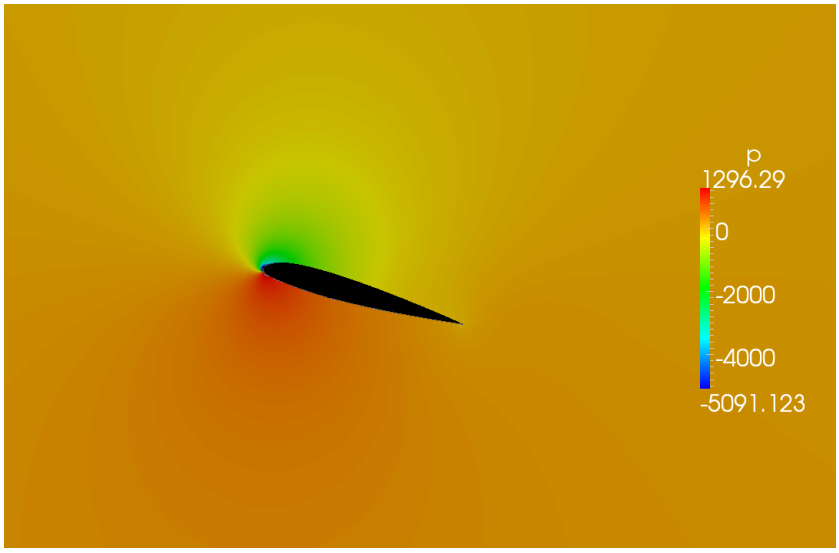


Figure 5.7: Pressure field around the NACA 23012 at $\alpha = 15^{o}$ (m^2/s^2)

The streamlines around the airfoil:

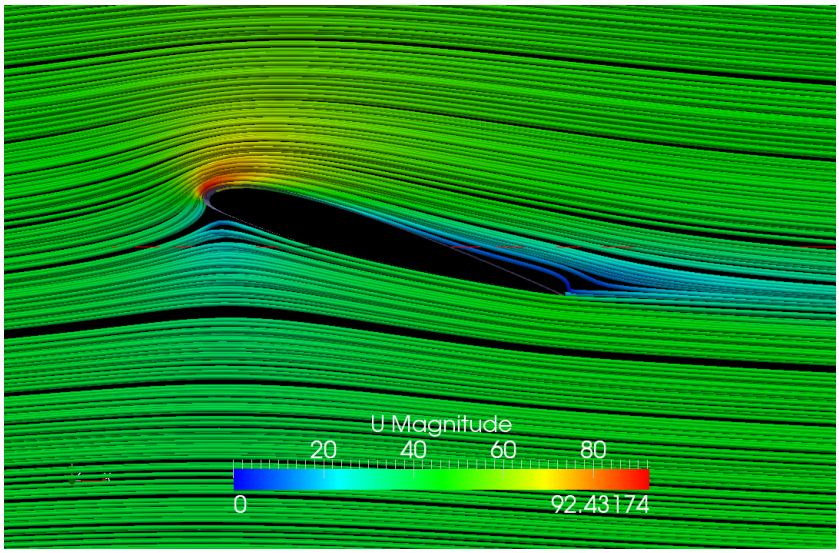


Figure 5.8: Streamlines around the NACA 23012 at $\alpha = 15^{o}$ (m/s)

It is possible to observe that at $\alpha = 15^{o}$ the flow has started to detach due to the strong pressure gradient on the upper surface. However, 15^{o} is not the maximum angle of attack (α_{stall}) because the fluid is not fully detached and therefore stall has not yet happened.

The vector field at the trailing edge (to appreciate the boundary layer detachment):

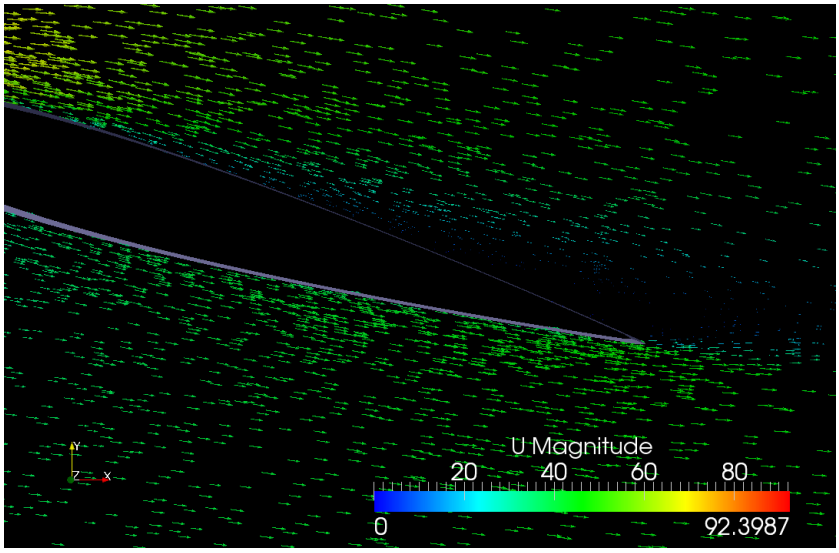


Figure 5.9: Velocity vectors around the NACA 23012 at $\alpha = 15^{o}$ (m/s)

The distribution of $\tilde{\nu}_t$ in the domain:

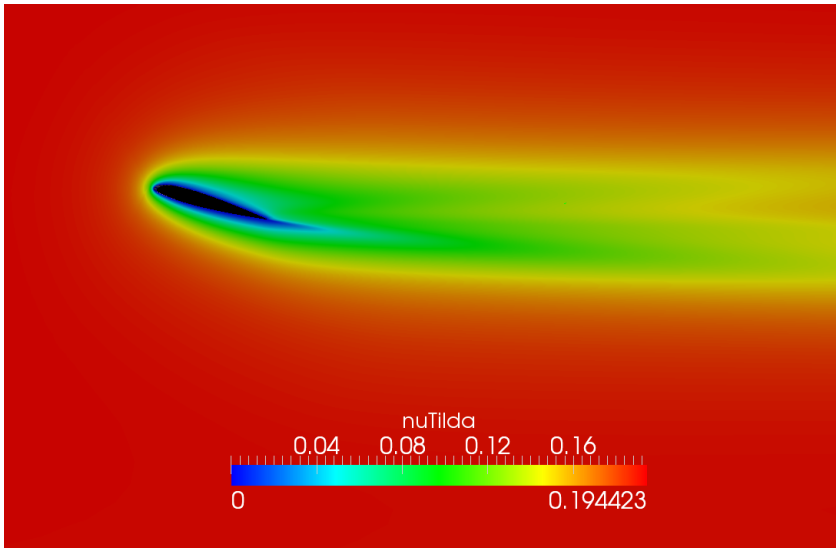


Figure 5.10: Distribution of $\tilde{\nu}_t$ in the domain of the alpha15 case

As it was explained in Chapter 3, the plot of the aerodynamic forces in front of the time gives information about the convergence of the case. In Figure 5.11 it is represented the lift coefficient:

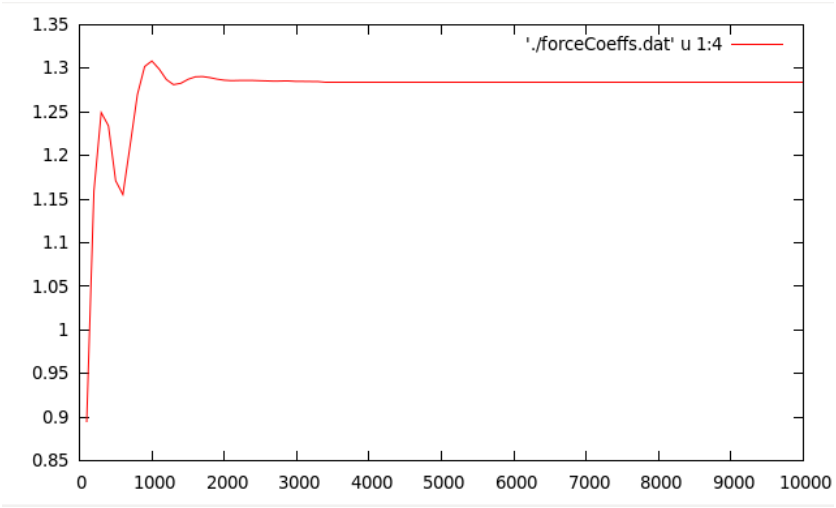


Figure 5.11: Evolution of the lift coefficient (ordinate axis) with the time (abscissa axis) in the alpha15 case

As it can be seen, the case has converged rapidly.

5.0.16.2 Results of the simulation for a range of α. Plot of the main aerodynamic curves

To simulate the NACA 23012 for different configurations (according to α), it is necessary to change this parameter before meshing. It has to be done in the parameter definition within *executeAirfoilMesher*. At the fourth line of the Script, the user can write the angle of attack (in degrees) at which the simulation will be carried out.

Caution:

> For $\alpha > 15^{o}$, as the flow detaches, there are unsteady phenomena. Thus, simpleFoam would not be the appropriate solver for these cases

Caution:

> Although the angle of attack changes from one case to the other, the velocity boundary conditions have to be set equal for all cases. This is because the flow velocity is not modified, but the geometric characteristics of the airfoil shape in the mesh

Some representative results of the behaviour of the NACA 23012 for different configurations are:

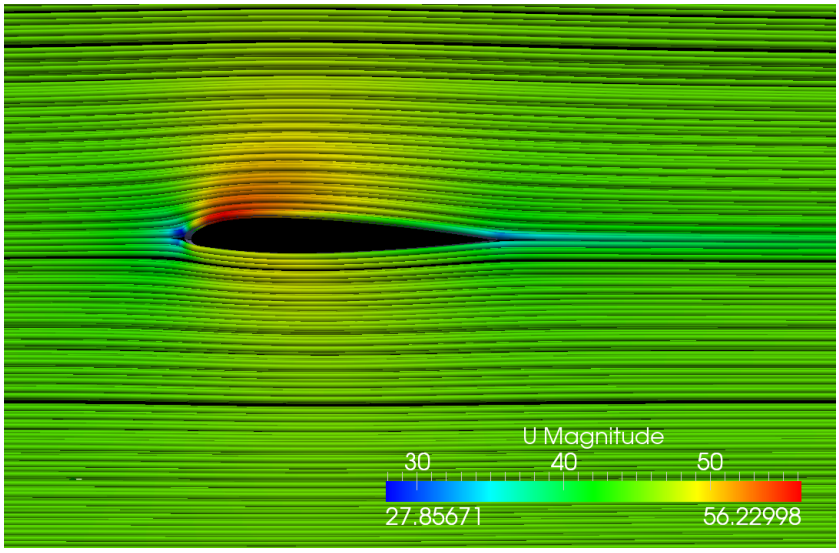


Figure 5.12: Behaviour of the flow around the NACA 23012 at $\alpha = 0^{o}$ (m/s)

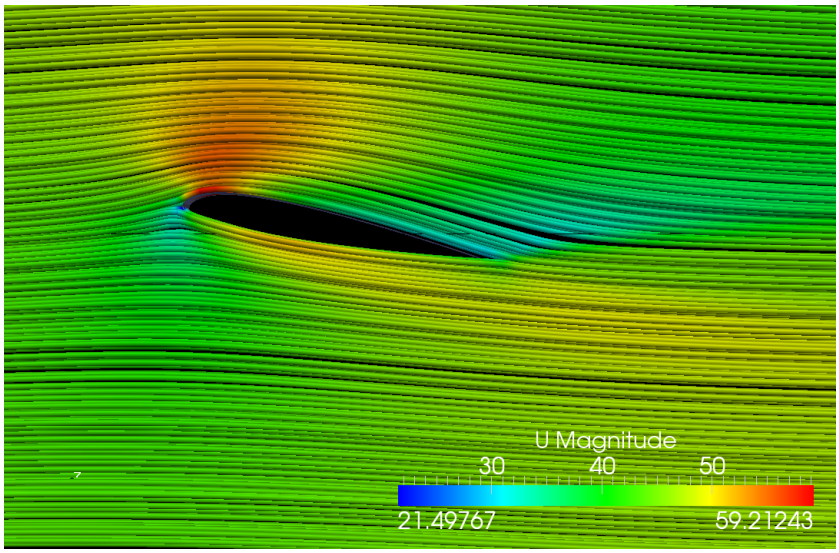


Figure 5.13: Behaviour of the flow around the NACA 23012 at $\alpha = 10^{o}$ (m/s)

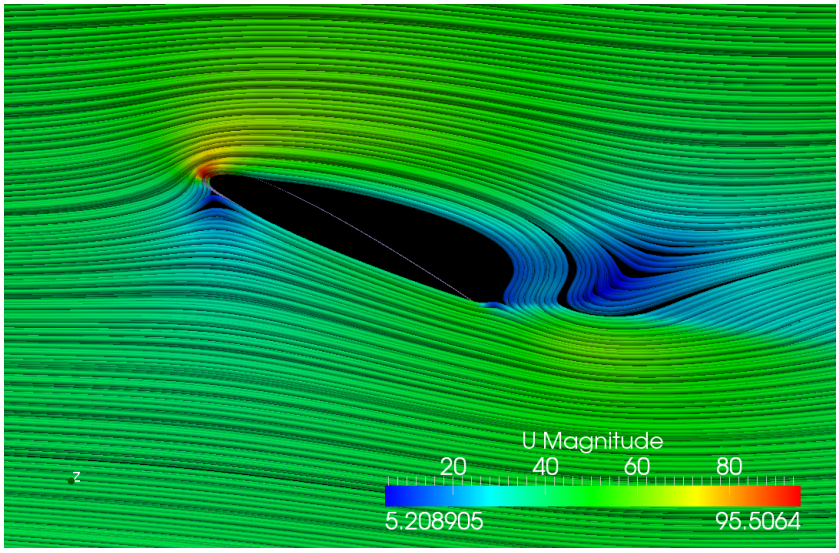


Figure 5.14: Behaviour of the flow around the NACA 23012 at $\alpha = 20^{o}$ (m/s)

By using these simulations, the main aerodynamic curves have been plot (in a range between -10^{o} and 15^{o} with intervals of 5^{o}). It can be done by copying the C_l and C_d obtained with the simulations and plot their relation using Gnuplot and the instructions shown in Section 3.6.3. The curves are:

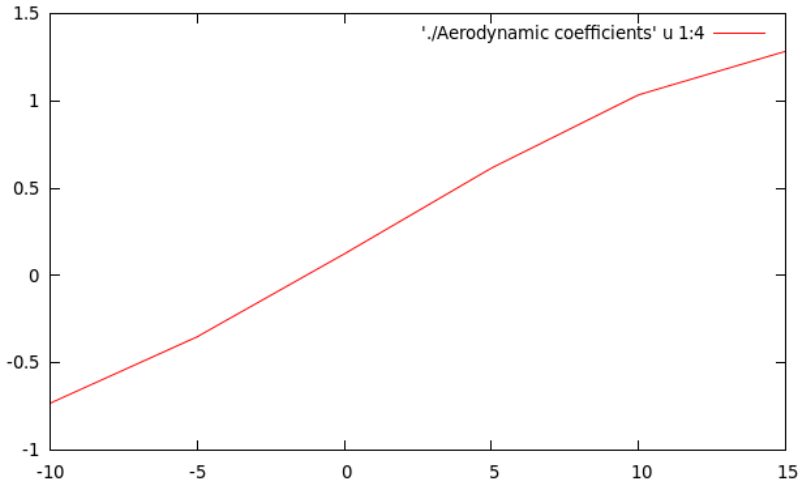


Figure 5.15: Relation between C_l (ordinate axis) and α (abscissa axis) for the NACA 23012

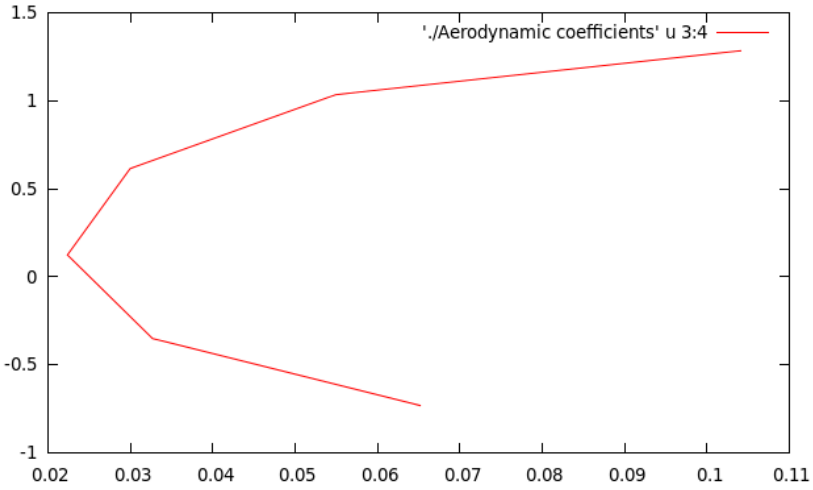


Figure 5.16: Relation between C_l (ordinate axis) and C_d (abscissa axis), polar curve, for the NACA 23012

It can be seen that for near zero angles of attack, the relation between C_l and α is linear. However, it is shown that this trend starts to stabilize when high values of $|\alpha|$ are achieved due to the boundary layer detachment.

5.0.17 Additional utilities

5.0.17.1 Mesh conversion

The user can generate meshes using other packages and convert them into the format that *OpenFOAM*® uses. This has been done in Section 5.0.15.1 to convert a mesh exported from Gmsh Mesh Generator by typing:

```
gmshToFoam
```

Furthermore, it is possible to convert meshes exported from other packages using instructions such as:

```
ansysToFoam
```

```
fluentMeshToFoam
```

5.0.17.2 Computation of y^+

y^+ is the dimensionless wall distance for a wall-bounded flow. It can be expressed as:

$$y^+ = \frac{u_\star y}{\nu} \tag{5.8}$$

where $u_\star$ is the friction velocity $\left(u_\star \equiv \sqrt{\frac{\tau_w}{\rho}}\right)$, y is the distance to the nearest wall and ν is the kinematic viscosity of the fluid. y^+ is commonly used in boundary layer theory and in defining the law of the wall.

y^+ plays a relevant role in the treatment of the boundary layer. The subdivision of the near-wall region in a turbulent boundary layer can be summarized as follows (*Fluent, 2005*):

- $y^+ < 5$: viscous sublayer region (velocity profile is assumed to be laminar and viscous stress dominates the wall shear)
- $5 < y^+ < 30$: buffer region (both viscous and turbulent shear dominate)
- $30 < y^+ < 300$: fully turbulent portion or log-law region (turbulent shear predominates)

In the alpha15 case, a wall function has been used to model the behaviour of the boundary layer. To use a wall function approach for a particular turbulence model with confidence, it is necessary to ensure that the y^+ values are within a certain range. As in the current case a RAS model has been implemented, it is necessary to guarantee that the first node of the mesh falls within the log-law region. It is to ensure that $30 < y^+ < 300$.

Caution:

> In addition to the concern about having a mesh with y^+ values that are too large, it is necessary to be aware that if the y^+ is too low then the first calculation point will be placed in the viscous sublayer region and the wall function will also be outside its validity

Caution:

> If an attached flow is modeled, then generally a wall function approach can be used. If flow separation is expected and the accurate prediction

of the separation point will have an impact on the results then it would be advisable to resolve the boundary layer with a finer mesh

OpenFOAM® presents a utility to calculate and report y^+ for all the wall patches. Note that as it depends on the local Reynolds number, it can only be obtained in the post-processing. Consequently only approximate values can be estimated when meshing.

As in the current case a RAS turbulence model has been used, the instruction (written in the terminal) to obtain y^+ is:

```
yPlusRAS
```

To view the results with ParaView it is necessary to select the `yPlus` box located within Volume Fields, choose the time at which the results will be consulted and select the wall patch (airfoil). The results:

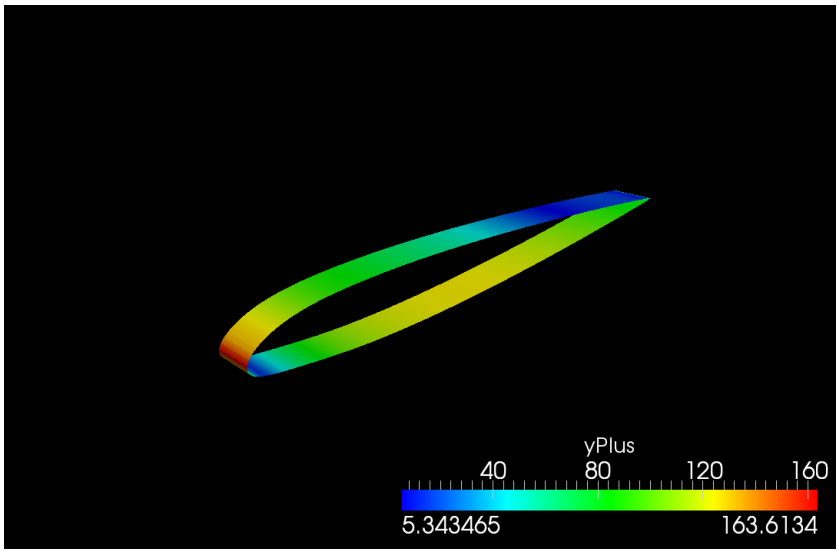


Figure 5.17: Distribution of y^+ at the wall of the airfoil in the alpha15 case at $t = 10000$ s

Note that there are some regions where y^+ is not within the required range. As it can be seen, it is basically located in the region of boundary layer detachment. In these regions the wall function approach would be out of its validity.

5.0.17.3 Isobars around a body

In this section it is explained how to create a contour plot across a plane. It will be used to plot the isobars around the NACA 23012 at $\alpha = 15^{o}$.

First of all the user has to launch ParaView, click on `internalMesh` and select the pressure field. Then, with the `alpha15.OpenFOAM` module highlighted, go to Filters -> Alphabetical -> Slice. The Slice filter allows the user to specify a cutting plane making the case truly two-dimensional. For the current case click on $\underline{Z}$ Normal so the plane will cut the domain in the adequate direction. Afterwards, select the `Slice1` module and go to Filters -> Alphabetical -> Contour.

To appreciate the isobars, select the `Slice1` and `Contour1` modules while keeping `alpha15.OpenFOAM` deselected. In the Properties menu of the `Contour` module, select the pressure field in the Contour By option and in the Isosurface panel click on New Range. Make sure that the `From` and `To` values coincide with the maximum and minimum of the legend and set the `Steps` option to 25. Finally, go to the Display menu and in the Color panel choose `Solid Color` in Color by instead of the pressure. Choose a color of your choice to represent the isobars. The results are then:

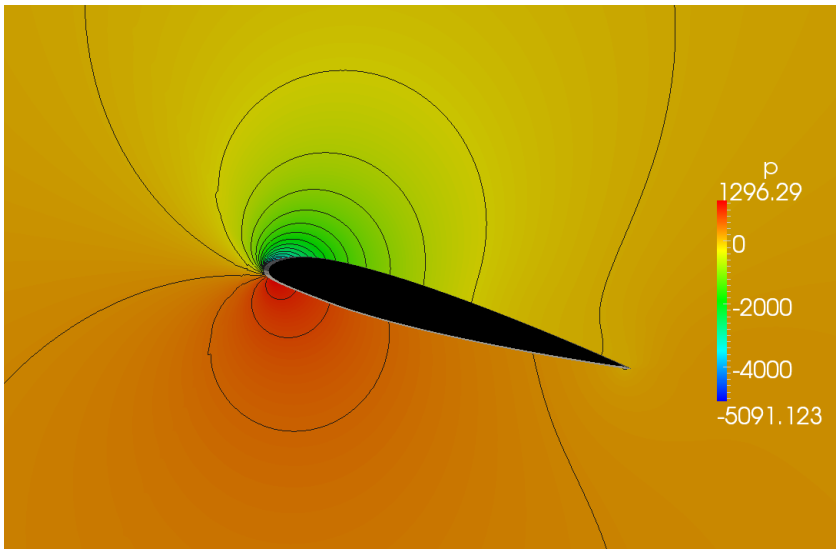


Figure 5.18: Isobars around the NACA 23012 at the alpha15 case at $t = 10000$ s (m^2/s^2)

5.0.17.4 Tube-like streamlines

In Section 2.6.2 it was shown how to plot the streamlines. Additionally, in the current section it is explained how to represent tube-like streamlines. It may add more

realism to the simulation or simply a different way of representing the streamlines. The differences between both configurations can be observed in the following figures:

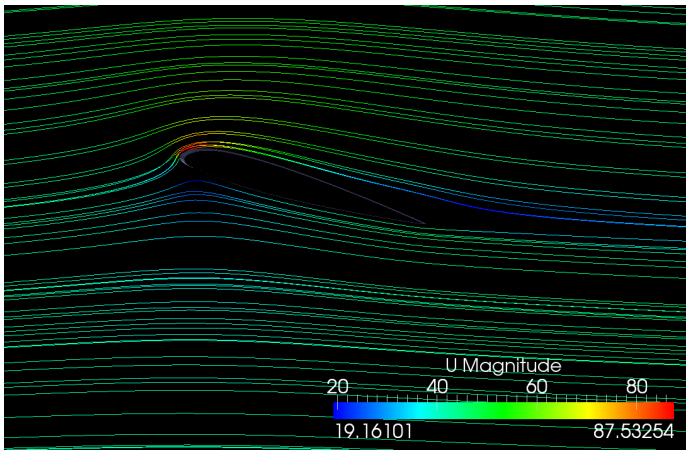


Figure 5.19: Streamlines around the NACA 23012 at the alpha15 case at $t = 10000$ s (m/s)

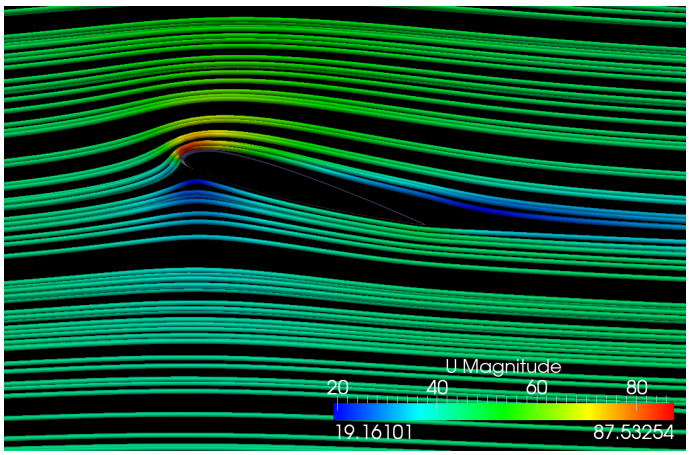


Figure 5.20: Tube-like streamlines around the NACA 23012 at the alpha15 case at $t = 10000$ s (m/s)

To obtain a configuration like Figure 5.20 it is first necessary to plot the streamlines as it was shown in Section 2.6.2. In the current case, Number of Points has been set to 3000 and Radius to 2. Once it is generated, it is necessary to select the module of the streamlines and go to Filters -> Alphabetical -> Tube. Finally, the thickness of the tube-like streamlines can be adjusted by changing the Radius option. In Figure 5.20 this parameter has been set to 0.01.