

3. Bidimensional laminar flow around a circular cylinder

3.1 Description of the case

The second tutorial is based on the analysis of external flow. More specifically, the study encompasses the modelling of a bidimensional laminar flow around a circular cylinder. The analytical solution of the same case modeled with potential flow is known and easy to interpret; however the physical mechanism which rules *real* flow around a circular cylinder as well as its mathematical treatment are highly complex. Some examples are the boundary layer detachment and oscillatory effects that without a CFD study would be very difficult to determine either analytically or experimentally. The case includes simulations made with two different Reynolds numbers to observe the behaviour of the flow in such different regimes.

3.2 Hypotheses

- Incompressible flow
- Laminar flow
- Newtonian flow
- Bidimensional flow ($\frac{\partial}{\partial z} = 0$)
- Negligible gravitatory effects

3.3 Physics of the problem

The problem encompasses a circular cylinder with a diameter of $D = 0.1$ m submerged in a stream with an undisturbed speed of $V = 5$ m/s and ambient pressure. The problem statement is shown at Figure 3.1.

3. Bidimensional laminar flow around a circular cylinder

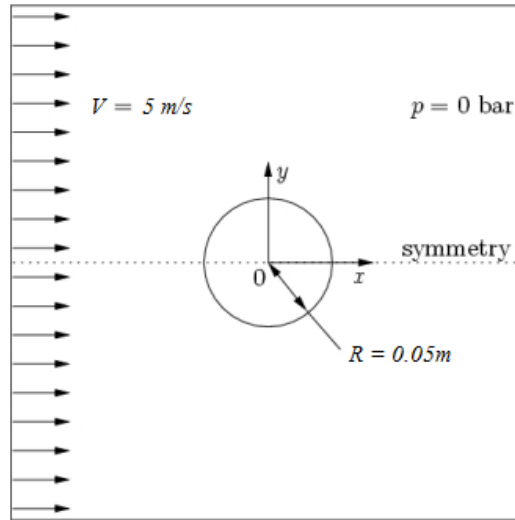
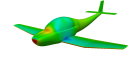


Figure 3.1: Flow around a circular cylinder

Unlike Chapter 2, in the current chapter there is no easy analytical solution to describe the behaviour of the fluid (except for potential flow, which is an ideal case). However, it is necessary to keep in mind the main equations and dimensionless numbers involved in the problem:

The continuity equation,

$$\nabla \cdot \mathbf{U} = 0 \quad (3.1)$$

The momentum equation,

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \mathbf{U} \quad (3.2)$$

The Reynolds number,

$$Re = \frac{D|\mathbf{U}|}{\nu} \quad (3.3)$$

The Strouhal number (which describes oscillating flow mechanisms),

$$St = \frac{wD}{2\pi|\mathbf{U}|} \quad (3.4)$$

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

$$C_D = \frac{F_D}{\frac{1}{2}\rho|\mathbf{U}|^2S} \quad (3.5)$$

The behaviour of the flow around a circular cylinder is very complex and highly de-

pendent on the Reynolds number. The boundary layer in laminar regimes detaches at an angle of $\theta = 82^\circ$ as a consequence of the existence of an adverse pressure gradient. Furthermore, due to the symmetry of the cylinder, a curious phenomenon appears named *von Kármán street*, consisting of alternate vortices emitted by the cylinder. The vortices are detached periodically and their frequency is directly related to the Strouhal number.

For external flow, the transition from laminar regime to turbulent regime occurs at $Re \approx 3 \times 10^5$. However, the very complex structure of vortex shedding deserves special mention; the relation between the behaviour of the flow and the Reynolds number is shown at Figure 3.2:

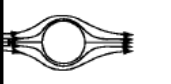
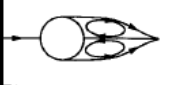

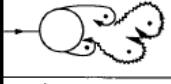

	No separation. Creeping flow	$Re < 5$
	A fixed pair of symmetric vortices	$5 < Re < 40$
	Laminar vortex street	$40 < Re < 200$
	Transition to turbulence in the wake	$200 < Re < 300$
	Wake completely turbulent. A: Laminar boundary layer separation	$300 < Re < 3 \times 10^5$ Subcritical

Figure 3.2: Flow structure depending on the Reynolds number, extracted from [2]

As the simulation is going to be run with laminar flow, the Reynolds number is not going to be set higher than 195. This will allow to observe the von Kármán street and its transition while guaranteeing that the whole fluid domain remains laminar. Additionally, another simulation with $Re = 30$ will be run to check if the results of the CFD simulation adapt to the scheme shown at Figure 3.2.

The drag coefficient is a function of the Reynolds number and decreases when the regime turns to turbulent. It can be seen at Figure 3.3.

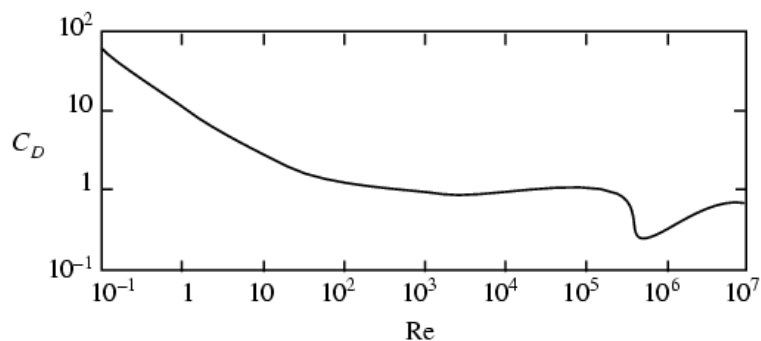
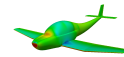


Figure 3.3: Drag coefficient as a function of the Reynolds number in an infinite circular cylinder

3.4 Pre-processing with $Re = 195$

The following codes contain the information to simulate the cylinder with $Re = 195$ using icoFoam. The case directory is named `cylinder195` and will be located within `FoamCases`. Its structure of directories and subdirectories is very similar to the one used in Chapter 2.

3.4.1 Mesh generation

The mesh for the study of the cylinder is not going to be uniform. Some areas of the domain need to contain a higher cell density (mainly the walls of the cylinder and its prolongation). Consequently, it is necessary to divide the domain in different blocks as Figure 3.4 shows.

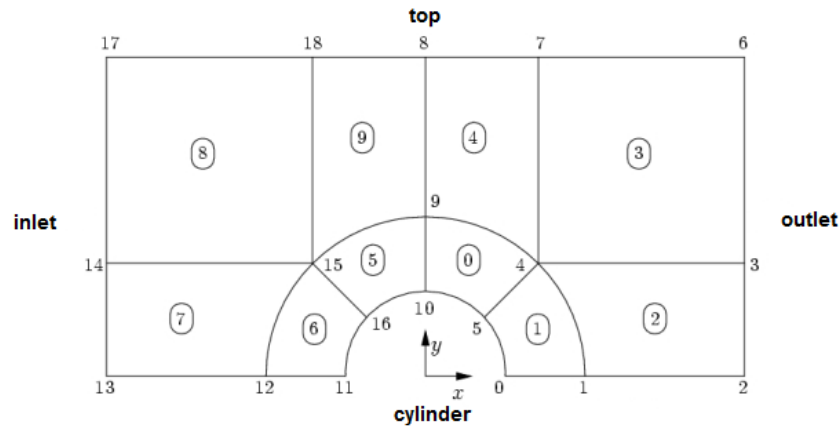


Figure 3.4: Half of the scheme used for the creation of the mesh, extracted from [1]

In Figure 3.4 there is only half of the mesh, which is contained between $z = -0.05$ m and $z = 0.05$ m. The numbers indicate the numbering used for the vertices and the blocks in the *blockMeshDict* file.

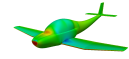
In the *bidimensional cylinder* case, *constant/blockMeshDict* must contain the following instructions:

```

1  /*-----* C++ *-----*/
2  |=====|
3  |  \ \  /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  |  \ \  /  O p e r a t i o n | Version: 2.2.1 |
5  |  \ \  /  A n d      | Web: www.OpenFOAM.org |
6  |  \ \  /  M a n i p u l a t i o n |
7  \*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     object        blockMeshDict;
14 }
15 // *****
16
17 convertToMeters 0.1;
18
19 vertices
20 (
21     (0.5 0 -0.5)
22     (1 0 -0.5)
23     (10 0 -0.5)
24     (10 0.707107 -0.5)
25     (0.707107 0.707107 -0.5)
26     (0.353553 0.353553 -0.5)
27     (10 2 -0.5)
28     (0.707107 2 -0.5)
29     (0 2 -0.5)

```

3. Bidimensional laminar flow around a circular cylinder



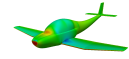
```
30      (0 1 -0.5)
31      (0 0.5 -0.5)
32
33      (-0.5 0 -0.5)
34      (-1 0 -0.5)
35      (-2 0 -0.5)
36      (-2 0.707107 -0.5)
37      (-0.707107 0.707107 -0.5)
38      (-0.353553 0.353553 -0.5)
39      (-2 2 -0.5)
40      (-0.707107 2 -0.5)
41
42      (0.5 0 0.5)
43      (1 0 0.5)
44      (10 0 0.5)
45      (10 0.707107 0.5)
46      (0.707107 0.707107 0.5)
47      (0.353553 0.353553 0.5)
48      (10 2 0.5)
49      (0.707107 2 0.5)
50      (0 2 0.5)
51      (0 1 0.5)
52      (0 0.5 0.5)
53
54      (-0.5 0 0.5)
55      (-1 0 0.5)
56      (-2 0 0.5)
57      (-2 0.707107 0.5)
58      (-0.707107 0.707107 0.5)
59      (-0.353553 0.353553 0.5)
60      (-2 2 0.5)
61      (-0.707107 2 0.5)
62
63
64      (10 -0.707107 -0.5)
65      (0.707107 -0.707107 -0.5)
66      (0.353553 -0.353553 -0.5)
67      (10 -2 -0.5)
68      (0.707107 -2 -0.5)
69      (0 -2 -0.5)
70      (0 -1 -0.5)
71      (0 -0.5 -0.5)
72
73      (-2 -0.707107 -0.5)
74      (-0.707107 -0.707107 -0.5)
75      (-0.353553 -0.353553 -0.5)
76      (-2 -2 -0.5)
77      (-0.707107 -2 -0.5)
78
79      (10 -0.707107 0.5)
80      (0.707107 -0.707107 0.5)
81      (0.353553 -0.353553 0.5)
82      (10 -2 0.5)
83      (0.707107 -2 0.5)
84      (0 -2 0.5)
85      (0 -1 0.5)
86      (0 -0.5 0.5)
```

```

87
88     (-2 -0.707107 0.5)
89     (-0.707107 -0.707107 0.5)
90     (-0.353553 -0.353553 0.5)
91     (-2 -2 0.5)
92     (-0.707107 -2 0.5)
93 );
94
95 blocks
96 (
97     hex (5 4 9 10 24 23 28 29) (80 20 1) simpleGrading (10 1 1)
98     hex (0 1 4 5 19 20 23 24) (80 20 1) simpleGrading (10 1 1)
99     hex (1 2 3 4 20 21 22 23) (200 20 1) simpleGrading (1 1 1)
100    hex (4 3 6 7 23 22 25 26) (200 40 1) simpleGrading (1 1 1)
101    hex (9 4 7 8 28 23 26 27) (20 40 1) simpleGrading (1 1 1)
102    hex (16 10 9 15 35 29 28 34) (20 80 1) simpleGrading (1 10 1)
103    hex (11 16 15 12 30 35 34 31) (20 80 1) simpleGrading (1 10 1)
104    hex (12 15 14 13 31 34 33 32) (20 20 1) simpleGrading (1 1 1)
105    hex (15 18 17 14 34 37 36 33) (40 20 1) simpleGrading (1 1 1)
106    hex (9 8 18 15 28 27 37 34) (40 20 1) simpleGrading (1 1 1)
107
108
109    hex (40 45 44 39 53 58 57 52) (20 80 1) simpleGrading (1 10 1)
110    hex (0 40 39 1 19 53 52 20) (20 80 1) simpleGrading (1 10 1)
111    hex (1 39 38 2 20 52 51 21) (20 200 1) simpleGrading (1 1 1)
112    hex (39 42 41 38 52 55 54 51) (40 200 1) simpleGrading (1 1 1)
113    hex (44 43 42 39 57 56 55 52) (40 20 1) simpleGrading (1 1 1)
114    hex (48 47 44 45 61 60 57 58) (80 20 1) simpleGrading (10 1 1)
115    hex (11 12 47 48 30 31 60 61) (80 20 1) simpleGrading (10 1 1)
116    hex (12 13 46 47 31 32 59 60) (20 20 1) simpleGrading (1 1 1)
117    hex (47 46 49 50 60 59 62 63) (20 40 1) simpleGrading (1 1 1)
118    hex (44 47 50 43 57 60 63 56) (20 40 1) simpleGrading (1 1 1)
119 );
120
121 edges
122 (
123     arc 0 5 (0.469846 0.17101 -0.5)
124     arc 5 10 (0.17101 0.469846 -0.5)
125     arc 1 4 (0.939693 0.34202 -0.5)
126     arc 4 9 (0.34202 0.939693 -0.5)
127     arc 19 24 (0.469846 0.17101 0.5)
128     arc 24 29 (0.17101 0.469846 0.5)
129     arc 20 23 (0.939693 0.34202 0.5)
130     arc 23 28 (0.34202 0.939693 0.5)
131     arc 11 16 (-0.469846 0.17101 -0.5)
132     arc 16 10 (-0.17101 0.469846 -0.5)
133     arc 12 15 (-0.939693 0.34202 -0.5)
134     arc 15 9 (-0.34202 0.939693 -0.5)
135     arc 30 35 (-0.469846 0.17101 0.5)
136     arc 35 29 (-0.17101 0.469846 0.5)
137     arc 31 34 (-0.939693 0.34202 0.5)
138     arc 34 28 (-0.34202 0.939693 0.5)
139
140
141     arc 0 40 (0.469846 -0.17101 -0.5)
142     arc 40 45 (0.17101 -0.469846 -0.5)
143     arc 1 39 (0.939693 -0.34202 -0.5)

```

3. Bidimensional laminar flow around a circular cylinder



```
144     arc 39 44 (0.34202 -0.939693 -0.5)
145     arc 19 53 (0.469846 -0.17101 0.5)
146     arc 53 58 (0.17101 -0.469846 0.5)
147     arc 20 52 (0.939693 -0.34202 0.5)
148     arc 52 57 (0.34202 -0.939693 0.5)
149     arc 11 48 (-0.469846 -0.17101 -0.5)
150     arc 48 45 (-0.17101 -0.469846 -0.5)
151     arc 12 47 (-0.939693 -0.34202 -0.5)
152     arc 47 44 (-0.34202 -0.939693 -0.5)
153     arc 30 61 (-0.469846 -0.17101 0.5)
154     arc 61 58 (-0.17101 -0.469846 0.5)
155     arc 31 60 (-0.939693 -0.34202 0.5)
156     arc 60 57 (-0.34202 -0.939693 0.5)
157 );
158
159 boundary
160 (
161
162     top
163     {
164         type symmetryPlane;
165         faces
166         (
167             (7 8 27 26)
168             (6 7 26 25)
169             (8 18 37 27)
170             (18 17 36 37)
171         );
172     }
173
174     bottom
175     {
176         type symmetryPlane;
177         faces
178         (
179             (49 50 63 62)
180             (50 43 56 63)
181             (43 42 55 56)
182             (42 41 54 55)
183         );
184     }
185
186     inlet
187     {
188         type patch;
189         faces
190         (
191             (14 13 32 33)
192             (17 14 33 36)
193
194             (46 13 32 59)
195             (46 49 62 59)
196         );
197     }
198
199     outlet
200     {
```

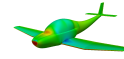


```

201         type patch;
202         faces
203         (
204             (2 3 22 21)
205             (3 6 25 22)
206
207             (38 51 21 2)
208             (41 54 51 38)
209         );
210     }
211
212     cylinder
213     {
214         type wall;
215         faces
216         (
217             (10 5 24 29)
218             (5 0 19 24)
219             (16 10 29 35)
220             (11 16 35 30)
221
222             (48 11 30 61)
223             (45 48 61 58)
224             (40 45 58 53)
225             (0 40 53 19)
226         );
227     }
228
229     frontAndBack
230     {
231         type empty;
232         faces
233         (
234             (5 10 9 4)
235             (24 23 28 29)
236             (0 5 4 1)
237             (19 20 23 24)
238             (1 4 3 2)
239             (20 21 22 23)
240             (4 7 6 3)
241             (23 22 25 26)
242             (4 9 8 7)
243             (28 23 26 27)
244             (16 15 9 10)
245             (35 29 28 34)
246             (12 15 16 11)
247             (31 30 35 34)
248             (13 14 15 12)
249             (32 31 34 33)
250             (14 17 18 15)
251             (33 34 37 36)
252             (15 18 8 9)
253             (34 28 27 37)
254
255             (45 40 39 44)
256             (58 57 52 53)
257             (40 0 1 39)

```

3. Bidimensional laminar flow around a circular cylinder



```
258         (53 52 20 19)
259         (39 1 2 38)
260         (52 51 21 20)
261         (39 38 41 42)
262         (52 55 54 51)
263         (44 39 42 43)
264         (57 56 55 52)
265         (47 48 45 44)
266         (60 57 58 61)
267         (12 11 48 47)
268         (31 60 61 30)
269         (13 12 47 46)
270         (32 59 60 31)
271         (49 46 47 50)
272         (62 63 60 59)
273         (50 47 44 43)
274         (63 56 57 60)
275     );
276 }
277 );
278
279 mergePatchPairs
280 (
281 );
282
283 // ***** //
```

It can be seen that as the cylinder has curved edges, it is necessary to use the `arc` instruction to obtain the circular geometry. This instruction must be followed by the labels of the connected vertices and an interpolation point contained in the trajectory of the arc.

In the current *blockMeshDict* there are four different types of patches: *wall*, *empty*, *symmetryPlane* and *patch*. All were used in the *plane-parallel plates* case except *symmetryPlane*, which is used in the *top* and *bottom* patches of the current case to indicate that there are no physical walls in the top and bottom borders; the flow must behave as if the domain would extend infinitely in the *y*-direction.

After running `blockMesh` and checking the results with `checkMesh` it is possible to observe the mesh with `ParaView`.

Caution:

It is necessary to have the majority of the required files for the simulation within the case directory to create the mesh. Wait until the three main directories (*0*, *constant* and *system*) and their files are ready to run `blockMesh`. Otherwise, it is also possible to use a solved case as a *dummy* file to run `blockMesh`

The mesh of `cylinder195` is shown at Figures 3.5, 3.6 and 3.7:

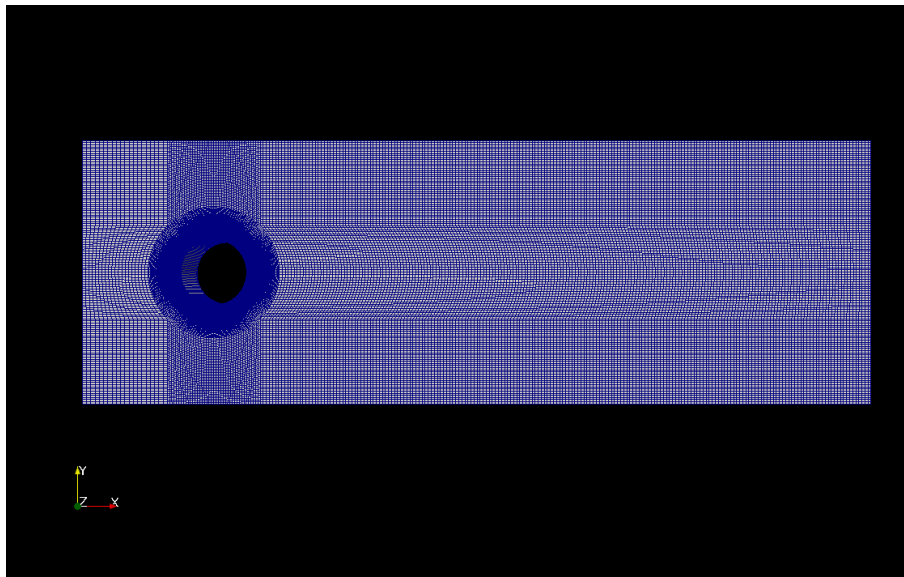


Figure 3.5: Mesh of the *bidimensional cylinder* case

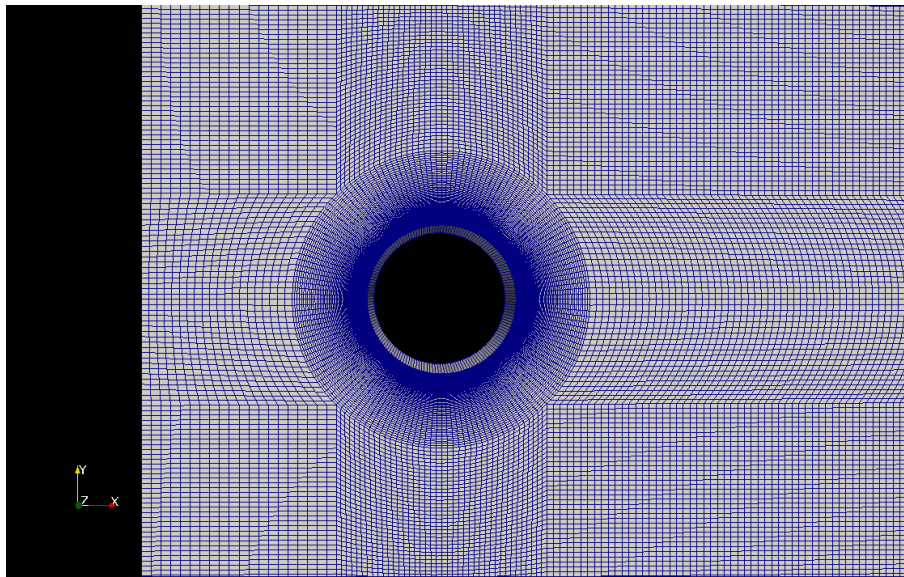


Figure 3.6: Detail of the mesh of the *bidimensional cylinder* case

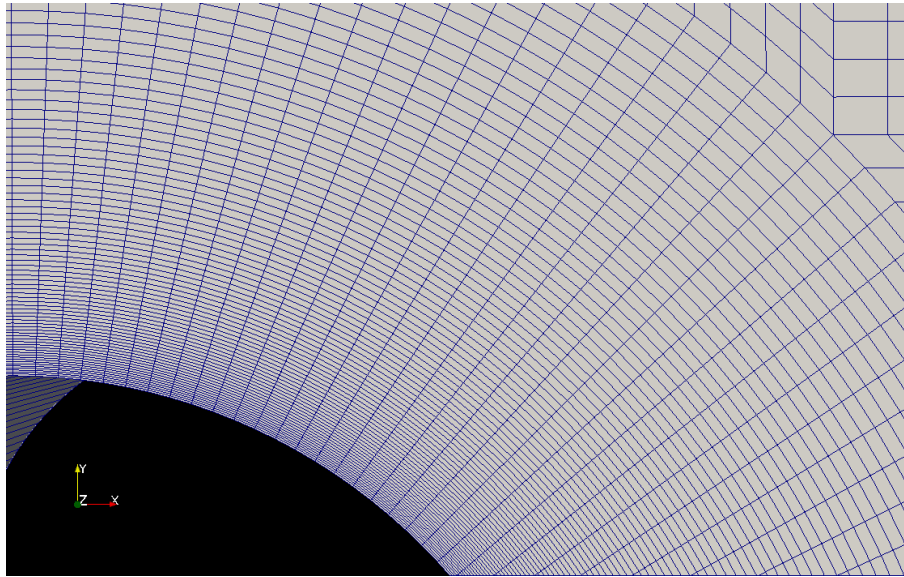
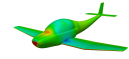


Figure 3.7: Detail of the mesh gradation on the walls of the *bidimensional cylinder* case

Advice:

On the walls and downstream the cylinder it is necessary to have a high refined mesh. It will allow the user to compute the drag coefficient and to observe the von Kármán street both with high accuracy. However, if the simulation is too much time-consuming or it is done with a Reynolds number such that no vortices are generated, it is recommended to reduce the refinement of the mesh

3.4.2 Boundary and initial conditions

The files (located in *0*) containing the information related to the pressure and the velocity fields are the following:

```
1  /*----- C++ -----*/
2  |=====|
3  |  \ \  /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  |  \ \  /  O p e r a t i o n | Version: 2.2.1 |
5  |  \ \  /  A n d      | Web: www.OpenFOAM.org |
6  |  \ \  /  M a n i p u l a t i o n |
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     top
24     {
25         type      symmetryPlane;
26     }
27
28     bottom
29     {
30         type      symmetryPlane;
31     }
32
33     inlet
34     {
35         type      freestreamPressure;
36     }
37
38     outlet
39     {
40         type      freestreamPressure;
41     }
42
43     frontAndBack
44     {
45         type      empty;
46     }
47
48     cylinder
49     {
50         type      zeroGradient;
51     }
52 }
53
54 // *****

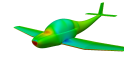
```

```

1  /*----- C++ -----*/
2  |=====|
3  |  \ \  /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  |  \ \  /  O p e r a t i o n  | Version: 2.2.1 |
5  |  \ \  /  A n d      | Web: www.OpenFOAM.org |
6  |  \ \  /  M a n i p u l a t i o n  |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volVectorField;
13     object        U;

```

3. Bidimensional laminar flow around a circular cylinder



```
14 }
15 // * * * * *
16
17 dimensions          [0 1 -1 0 0 0 0];
18
19 internalField        uniform (5 0 0);
20
21 boundaryField
22 {
23     top
24     {
25         type          symmetryPlane;
26     }
27
28     bottom
29     {
30         type          symmetryPlane;
31     }
32
33     inlet
34     {
35         type          freestream;
36         freestreamValue  uniform (5 0 0);
37     }
38
39     outlet
40     {
41         type          freestream;
42         freestreamValue  uniform (5 0 0);
43     }
44
45     cylinder
46     {
47         type          fixedValue;
48         value         uniform (0 0 0);
49     }
50
51     frontAndBack
52     {
53         type          empty;
54     }
55 }
56
57 // * * * * *
```

The **freestreamPressure** condition is acting as **zeroGradient** but with a more accurate physical behaviour. **freestream** acts as **fixedValue** when the flow is ingoing, and as **zeroGradient** when it is outgoing. This kind of boundary conditions are widely used for external flow simulations.

3.4.3 Physical properties

Within `constant` one finds the information related to the kinematic viscosity and the `RASProperties` file to compute the wall shear stress or other utilities requesting the `RAS` dictionary.

```

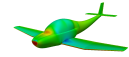
1  /*-----* C++ *-----*/
2  |=====|
3  |  \ \  /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  |  \ \  /  O p e r a t i o n | Version: 2.2.1 |
5  |  \ \  /  A n d      | Web: www.OpenFOAM.org |
6  |  \ \  /  M a n i p u l a t i o n |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "constant";
14     object        transportProperties;
15  }
16  // *****
17
18  transportModel  Newtonian;
19
20  nu              nu [0 2 -1 0 0 0 0] 2.564103e-03;
21
22  // *****

```

```

1  /*-----* C++ *-----*/
2  |=====|
3  |  \ \  /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  |  \ \  /  O p e r a t i o n | Version: 2.2.1 |
5  |  \ \  /  A n d      | Web: www.OpenFOAM.org |
6  |  \ \  /  M a n i p u l a t i o n |
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        laminar;
19
20  turbulence       off;
21
22  printCoeffs     off;
23
24  // *****

```



3.4.4 Control

```
1  /*----- C++ -----*/
2  |=====|
3  |  \ \  /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  |  \ \  /  O p e r a t i o n | Version: 2.2.1 |
5  |  \ \  /  A n d      | Web: www.OpenFOAM.org |
6  |  \ \  /  M a n i p u l a t i o n |
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      icoFoam;
19
20 startFrom         startTime;
21
22 startTime         0;
23
24 stopAt            endTime;
25
26 endTime           1.75;
27
28 deltaT            0.00001;
29
30 writeControl       timeStep;
31
32 writeInterval      1000;
33
34 purgeWrite         0;
35
36 writeFormat        ascii;
37
38 writePrecision     6;
39
40 writeCompression   off;
41
42 timeFormat         general;
43
44 timePrecision      6;
45
46 runtimeModifiable true;
47
48 // *****
```

Advice:

The case reaches steady before $t = 1.75$ s. Despite it, `endTime` has been set to 1.75 to widely appreciate the von Kármán street and its periodical distribution. As the mesh is very refined and the solver is transitory, the simulation may be very slow

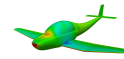
3.4.5 Discretization and linear-solver settings

```

1  /*----- C++ -----*/
2  |=====|
3  |  \ \  /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  |  \ \  /  O p e r a t i o n | Version:  2.2.1                |
5  |  \ \  /  A n d             | Web:      www.OpenFOAM.org        |
6  |  \ \  /  M a n i p u l a t i o n |                               |
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       Euler;
21  }
22
23  gradSchemes
24  {
25     default       Gauss linear;
26     grad(p)       Gauss linear;
27  }
28
29  divSchemes
30  {
31     default       none;
32     div(phi,U)    Gauss linear;
33  }
34
35  laplacianSchemes
36  {
37     default       none;
38     laplacian(nu,U) Gauss linear orthogonal;
39     laplacian((1|A(U)),p) Gauss linear orthogonal;
40  }
41
42  interpolationSchemes
43  {
44     default       linear;
45     interpolate(HbyA) linear;
46  }
47

```

3. Bidimensional laminar flow around a circular cylinder



```
48 snGradSchemes
49 {
50     default            orthogonal;
51 }
52
53 fluxRequired
54 {
55     default            no;
56     p                  ;
57 }
58
59 // *****

1  /*-----* C++ *-----*\
2  |=====|
3  |  \ \  /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  |  \ \  /  O p e r a t i o n | Version:  2.2.1                    |
5  |  \ \  /  A n d              | Web:      www.OpenFOAM.org         |
6  |  \ \  /  M a n i p u l a t i o n |                               |
7  \*-----*/
8  FoamFile
9  {
10     version            2.0;
11     format              ascii;
12     class               dictionary;
13     location            "constant";
14     object               transportProperties;
15 }
16 // *****

17
18 solvers
19 {
20     p
21     {
22         solver          PCG;
23         preconditioner   DIC;
24         tolerance        1e-06;
25         relTol           0;
26     }
27
28     U
29     {
30         solver          PBiCG;
31         preconditioner   DILU;
32         tolerance        1e-05;
33         relTol           0;
34     }
35 }
36
37 PISO
38 {
39     nCorrectors          2;
40     nNonOrthogonalCorrectors 3;
41     pRefCell              0;
42     pRefValue              0;
43 }
```

44

45 // ***** //

Advice:

As it can be seen, the instruction `nNonOrthogonalCorrectors` has been set to 3 instead of 0 (as it was in the *plane-parallel plates* case). It is so because when checking the mesh with `checkMesh`, it is possible to observe that there are *mesh non-orthogonalities*. Although globally the mesh is OK to be run, it helps in obtaining more physically accurate results

At the end of the pre-processing, the structure of directories and subdirectories within `cylinder195` should be as follows:

$$\text{cylinder195} \left\{ \begin{array}{ll} 0 & \left\{ \begin{array}{l} p \\ U \end{array} \right. \\ \text{constant} & \left\{ \begin{array}{l} \text{polyMesh} \left\{ \text{blockMeshDict} \right. \\ \text{transportProperties} \\ \text{RASProperties} \end{array} \right. \\ \text{system} & \left\{ \begin{array}{l} \text{controlDict} \\ \text{fvSchemes} \\ \text{fvSolutions} \end{array} \right. \end{array} \right.$$

3.5 Post-processing

3.5.1 Results of the simulation with $Re = 195$

The evolution over time of the velocity is as follows:

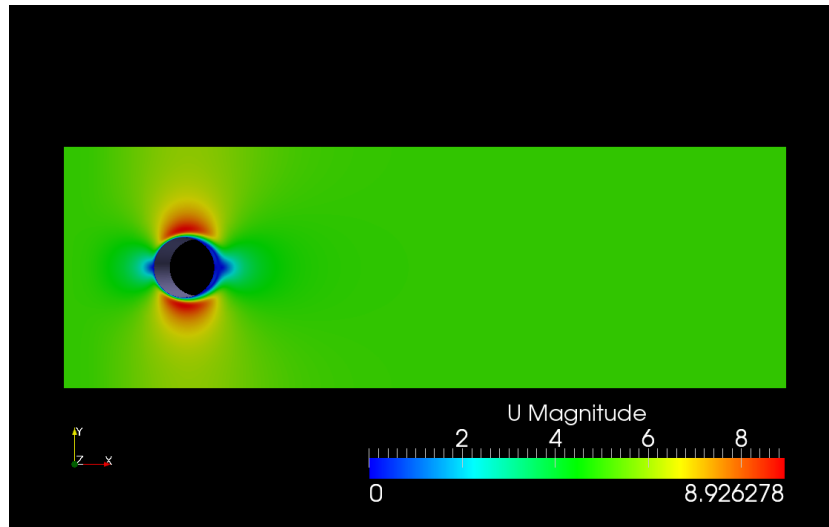
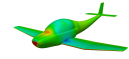


Figure 3.8: Velocity field around the bidimensional cylinder at $t = 0.01$ s (m/s)

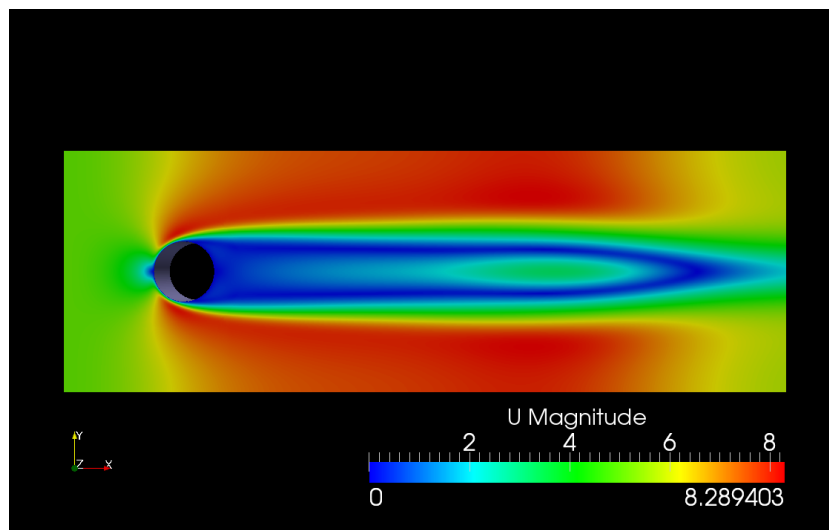


Figure 3.9: Velocity field around the bidimensional cylinder at $t = 0.6$ s (m/s)

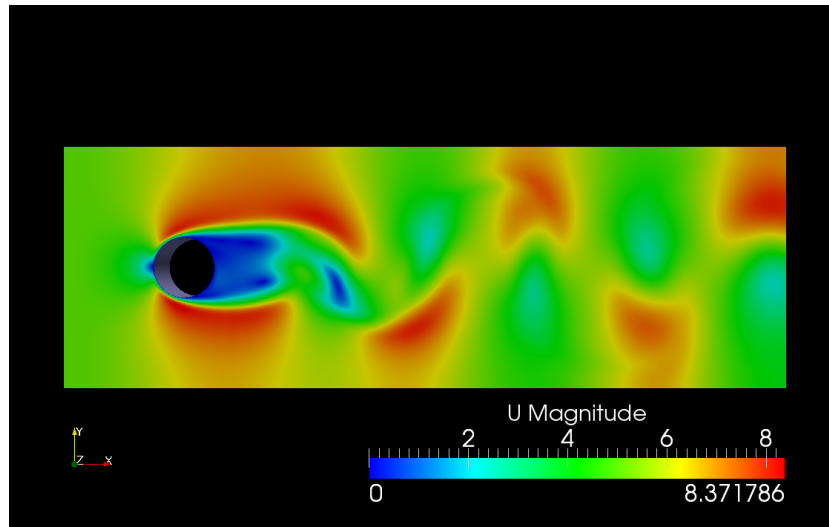


Figure 3.10: Velocity field around the bidimensional cylinder at $t = 1.13$ s (m/s)

The evolution over time of the pressure is as follows:

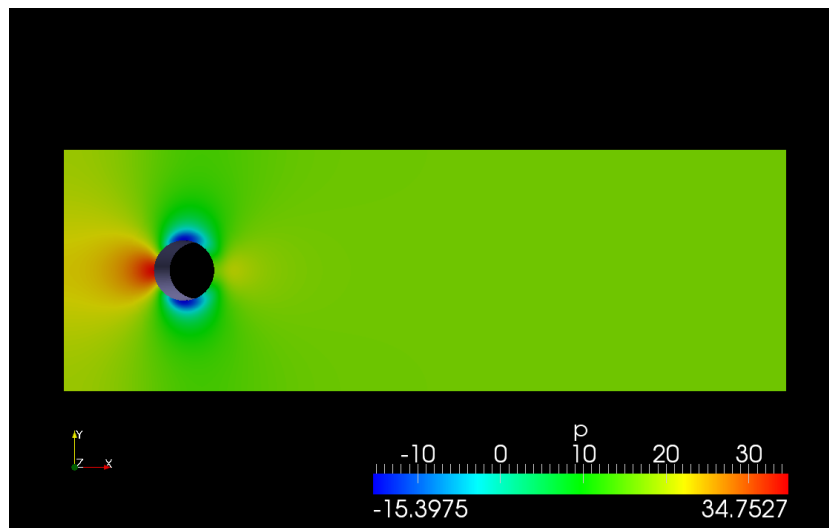


Figure 3.11: Pressure field around the bidimensional cylinder at $t = 0.01$ s (m^2/s^2)

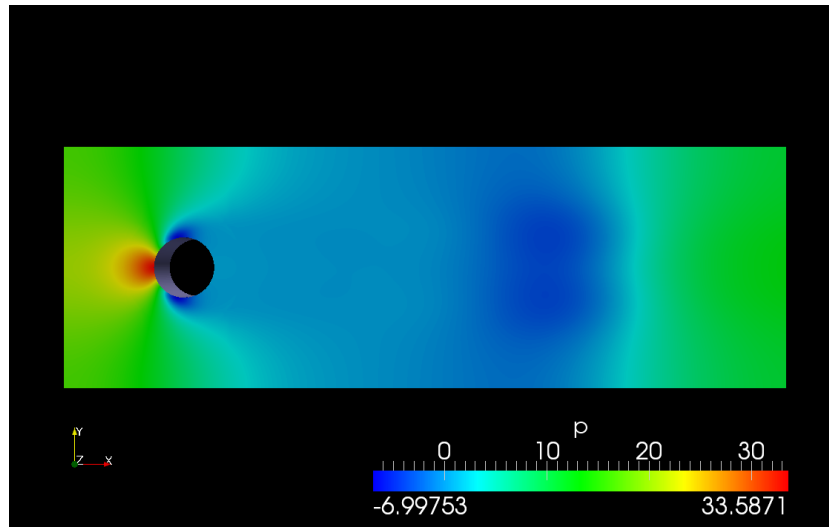
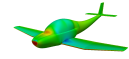


Figure 3.12: Pressure field around the bidimensional cylinder at $t = 0.6$ s (m^2/s^2)

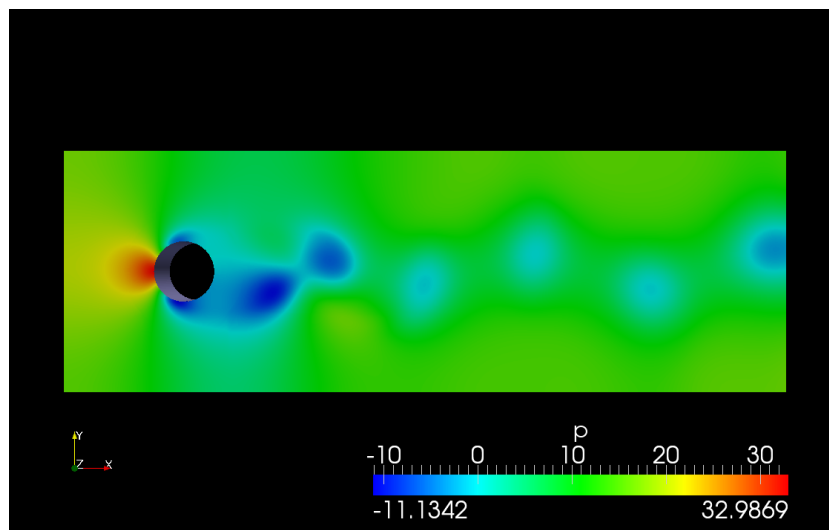


Figure 3.13: Pressure field around the bidimensional cylinder at $t = 1.13$ s (m^2/s^2)

The evolution over time of the streamlines is as follows:

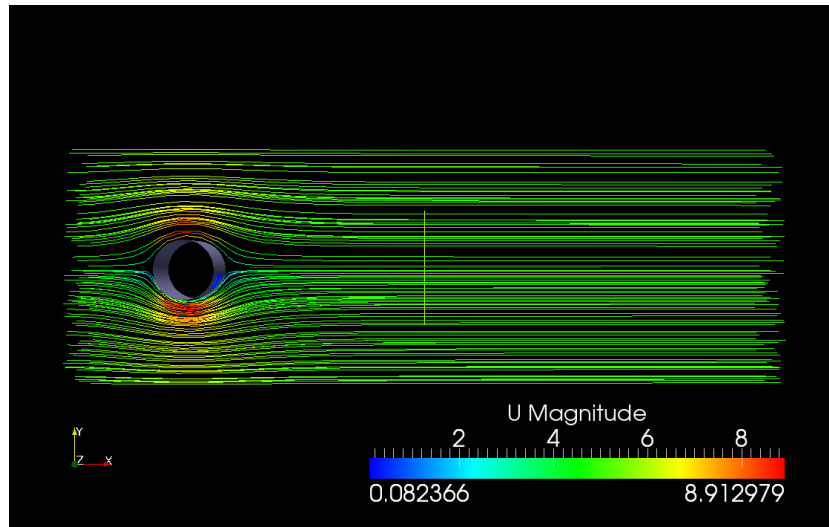


Figure 3.14: Streamlines around the bidimensional cylinder at $t = 0.01$ s (m/s)

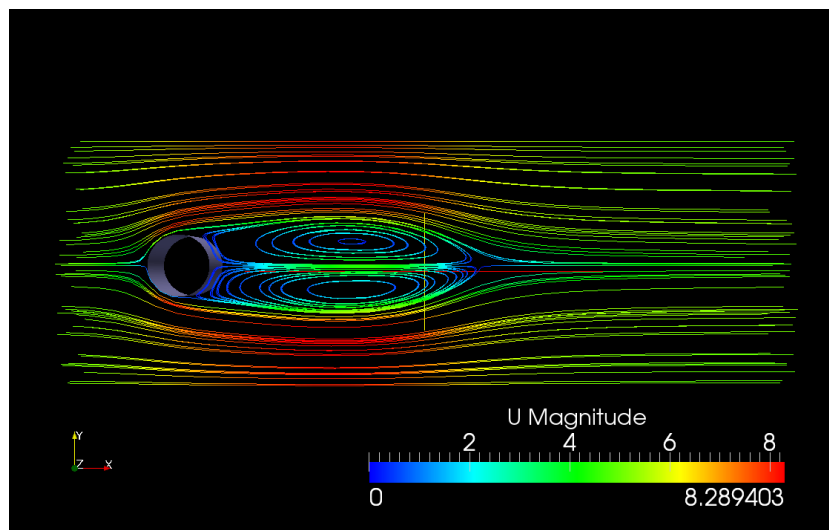


Figure 3.15: Streamlines around the bidimensional cylinder at $t = 0.4$ s (m/s)

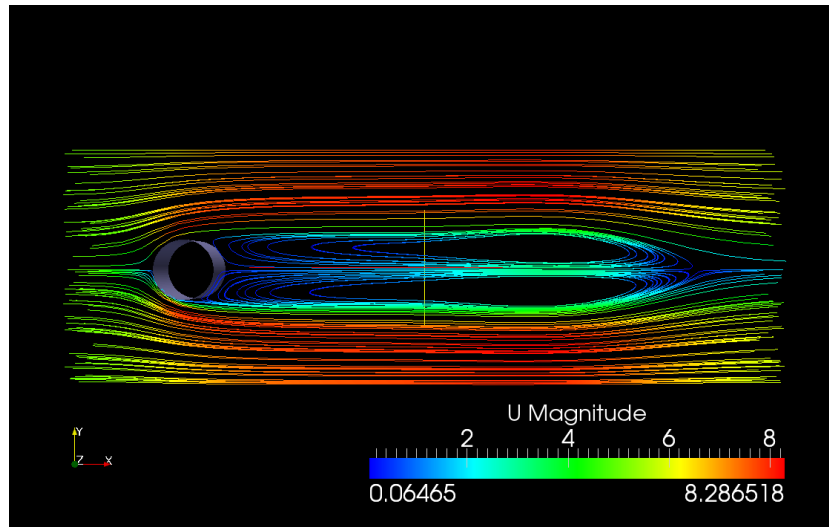
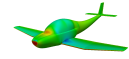


Figure 3.16: Streamlines around the bidimensional cylinder at $t = 0.6$ s (m/s)

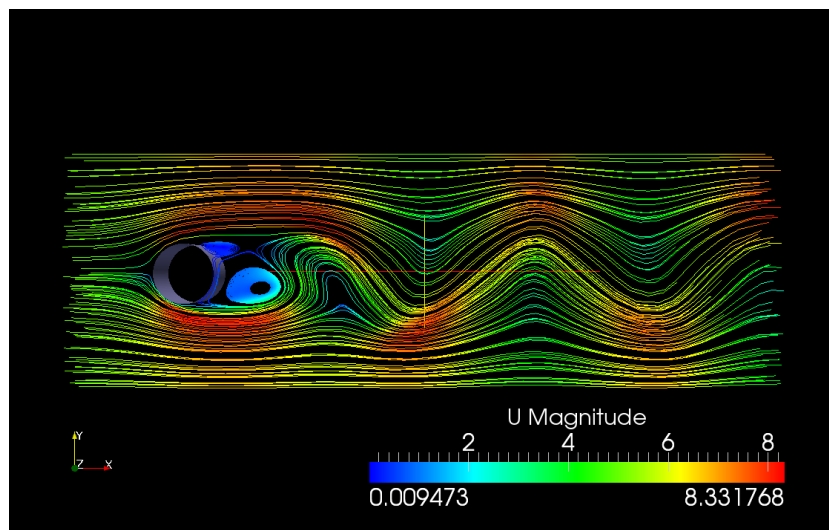


Figure 3.17: Streamlines around the bidimensional cylinder at $t = 1.13$ s (m/s)

A detail of the vector field near the wall of the cylinder is shown to appreciate the boundary layer detachment:

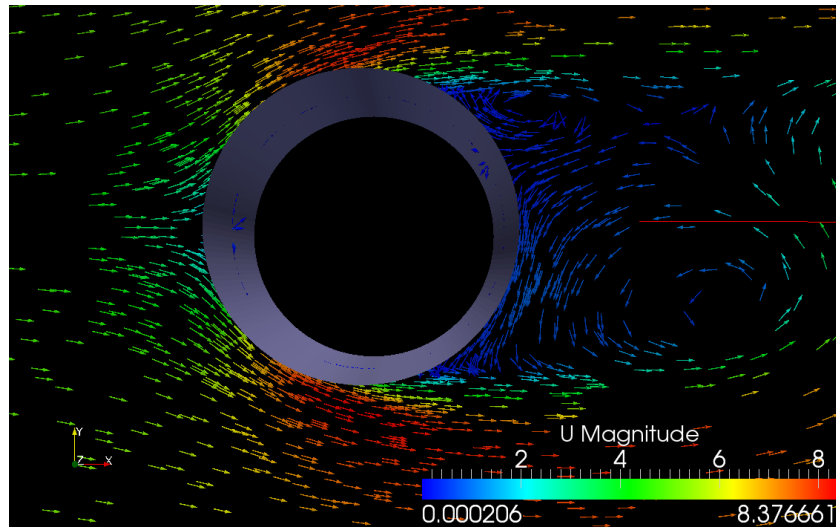


Figure 3.18: Velocity vectors around the bidimensional cylinder at $t = 1.13$ s (m/s)

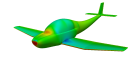
Advice:

A way to introduce the shape of the cylinder to provide more realism to the results is shown in Section 3.6.5

3.5.2 Comparative between cases with $Re = 30$ and $Re = 195$

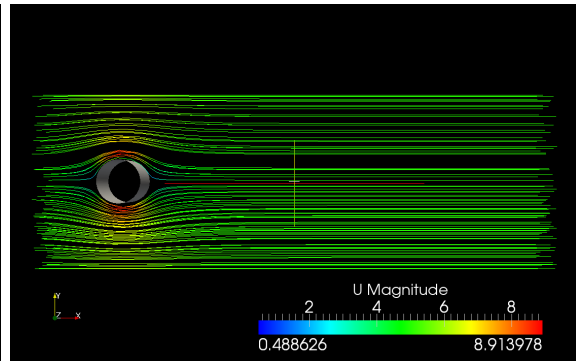
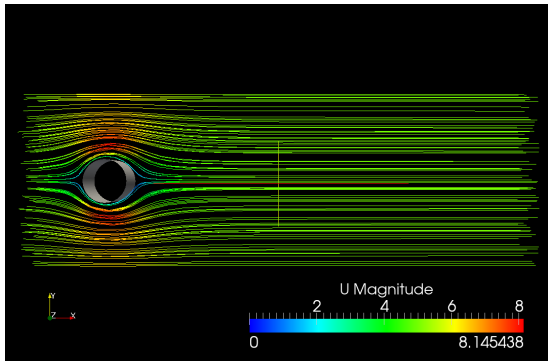
Additionally, the simulation has been done for another different Reynolds number ($Re = 30$). The only instruction that needs to be changed in the *OpenFOAM*[®] code is for instance the kinetic viscosity in *constant/transportProperties*. According to Equation 5.3, by maintaining the value of the inlet velocity while increasing the kinetic viscosity it is possible to simulate the bidimensional cylinder in such regime (case *cylinder30*):

3. Bidimensional laminar flow around a circular cylinder

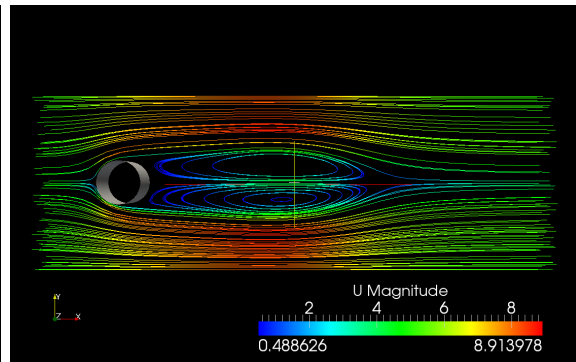
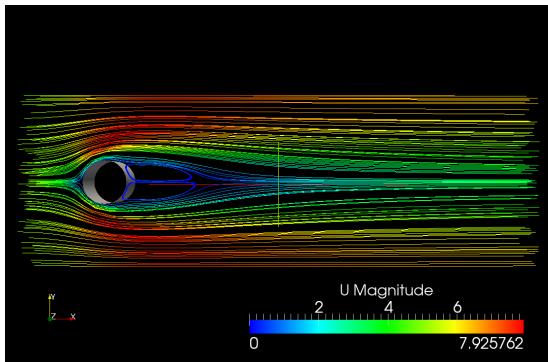


$Re = 30$

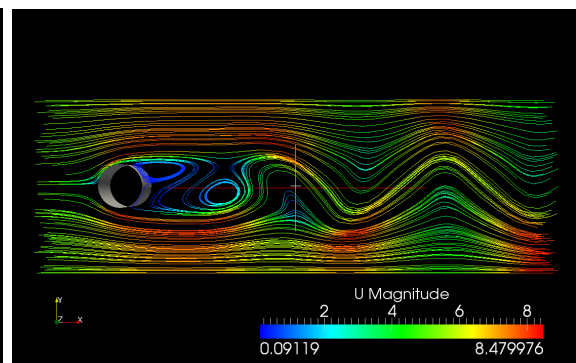
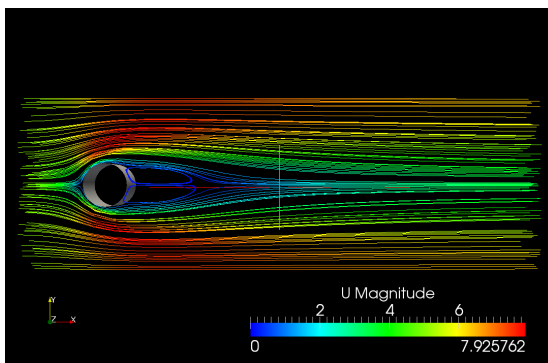
$Re = 195$



$t = 0.01$ s



$t = 0.4$ s



$t = 1$ s

As it was shown in Section 3.3, for $Re = 30$ the alternate vortices are not detached.

3.6 Additional utilities

3.6.1 Vorticity

As with the computation of the flow rate and the wall shear stress, it is also possible to compute the vorticity in the fluid field by using *OpenFOAM*® utilities. The vorticity is a pseudovector field that describes the local spinning motion of a fluid near some point, as would be seen by an observer located at that point and traveling along with the fluid. Mathematically, the vorticity is the curl of the velocity field:

$$\vec{\omega} = \nabla \times \mathbf{U}$$

The vorticity of a two-dimensional flow is always perpendicular to the plane of the flow. It plays a relevant role in the current chapter due to the existence of vortices generated by the cylinder forming the von Kármán street.

To execute it, type within the case directory

```
vorticity
```

Within the directories of each time step it has appeared a new file. Then, to observe the vorticity field, open *ParaView* and select it by clicking the **vorticity** box contained within **Volume Fields**. For the *cylinder195* case, the results are:

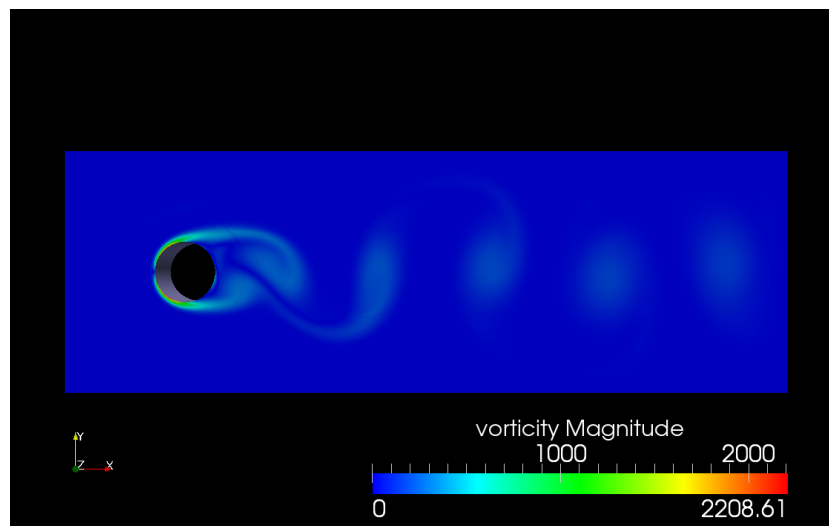
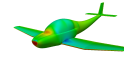


Figure 3.19: Vorticity field around the bidimensional cylinder at $t = 1.75$ s



It offers a clear-cut representation of the von Kármán vortices at $Re = 195$.

3.6.2 Computation of the aerodynamic coefficients

One of the main interests when studying external flows is the computation and analysis of the aerodynamic forces that the fluid exerts to solid objects. The drag is the force parallel to the flow velocity and the lift is the force perpendicular to the flow velocity. The dimensionless drag force is presented in Equation 5.6 and for an infinite circular cylinder its dependence on the Reynolds number is shown at Figure 3.3.

With *OpenFOAM*[®], to compute the aerodynamic forces that a fluid exerts on solid walls, add the following code after the last instruction of `controlDict`:

```
1  functions
2  (
3  forces
4  {
5  type forces;
6  functionObjectLibs ("libforces.so");
7  patches (cylinder); // Patch where the force exerted by the fluid is calculated
8  pName p;
9  UName U;
10 rhoName rhoInf;
11 rhoInf 1000; // Reference density of the fluid
12 CofR (0 0 0); // Origin for moment calculations
13 outputControl timeStep; // Time criterion used to print the results
14 outputInterval 100; // How often (according to outputControl) the results are
    printed
15 }
16 forceCoeffs
17 {
18 type forceCoeffs;
19 functionObjectLibs ("libforces.so");
20 patches (cylinder); // Patch where the force exerted by the fluid is calculated
21 pName p;
22 UName U;
23 rhoName rhoInf;
24 rhoInf 1000; // Reference density of the fluid
25 CofR (0 0 0); // Origin for moment calculations
26 liftDir (0 1 0);
27 dragDir (1 0 0);
28 pitchAxis (0 0 1);
29 magUInf 5; // Free stream velocity
30 lRef 0.1; // Reference length (diameter of the cylinder)
31 Aref 0.01; // Reference area (cross sectional area of the cylinder)
32 outputControl timeStep; // Time criterion used to print the results
33 outputInterval 100; // How often (according to outputControl) the results are
    printed
34 }
```

35);

Advice:

For incompressible cases, the value of `rhoInf` is irrelevant for the computation of the dimensionless coefficients

Now, when the case is rerun, a new directory named *postProcessing* appears next to *0*, *constant* and *system*. This directory contains two subdirectories with information concerning the evolution of the aerodynamic forces and moments and their dimensionless coefficients.

3.6.3 Plotting the results with Gnuplot

Once the aerodynamic forces have been computed with the instruction shown in Section 3.6.2, it is useful to plot the results. Besides showing the behaviour of the forces with time, it allows an understanding of the convergence (or divergence) of the case. For the *bidimensional cylinder* case, it is possible to claim that if the drag coefficient converges then the case converges too.

First of all the user has to have **Gnuplot** installed. It is a portable command-line driven graphing utility for Linux, MS Windows, OSX and many other platforms. It is widely used to plot data obtained with *OpenFOAM*®. First, access the file containing the required data:

```
cd FoamCases/cylinder195/postProcessing/forceCoeffs/0
```

Secondly, execute **Gnuplot** by typing:

```
gnuplot
```

Finally plot the values of the drag coefficient (third column) in front of the time (first column) by typing:

```
plot './forceCoeffs.dat' u 1:3 w l
```

The plot is:

3. Bidimensional laminar flow around a circular cylinder

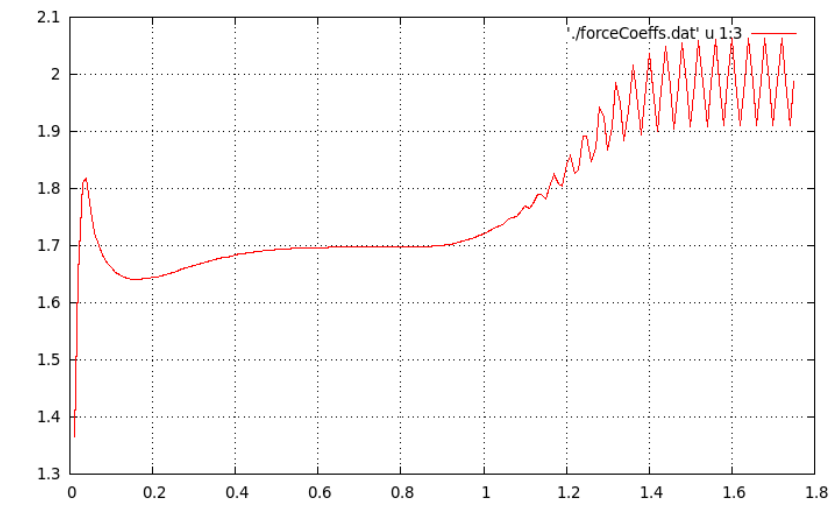
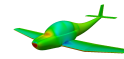


Figure 3.20: Drag coefficient (ordinate axis) of the bidimensional cylinder at $Re = 195$ in front of time (abscissa axis)

It is also possible to plot the values of the lift coefficient (fourth column) in front of the time (first column) by typing:

```
plot './forceCoeffs.dat' u 1:4 w l
```

The plot is:

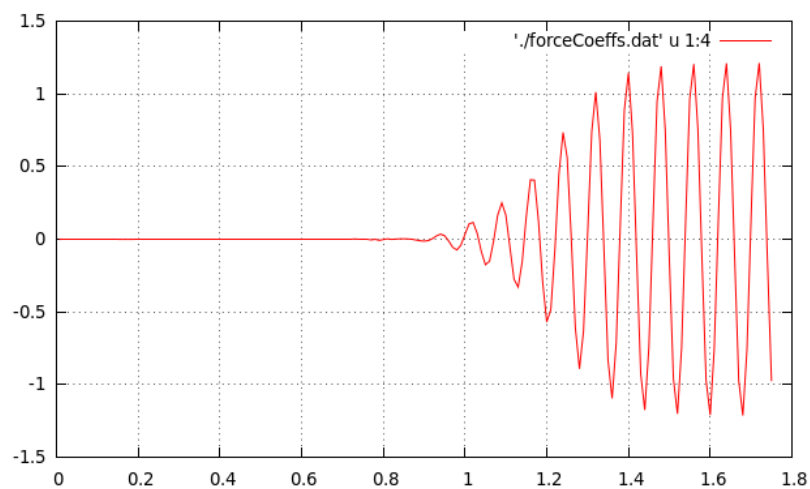


Figure 3.21: Lift coefficient (ordinate axis) of the bidimensional cylinder at $Re = 195$ in front of time (abscissa axis)

3.6.4 Computation of the stream function

As it was explained in Section 2.6.2, the streamlines offer a clear understanding of the behaviour of the flow; they represent the trajectories of particles in steady fluids. Related to it, there exists a scalar function (stream function) such that the flow velocity components can be expressed as the derivatives of this function, also being used to plot the streamlines. Mathematically it is related to the velocity as:

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}$$

Since streamlines are tangent to the velocity vector of the flow, the value of the stream function must be constant along a streamline.

To obtain the stream function of the velocity, type:

```
streamFunction
```

To view the results with ParaView, it is necessary to select the `streamFunction` box located within Point Fields. Here are the results:

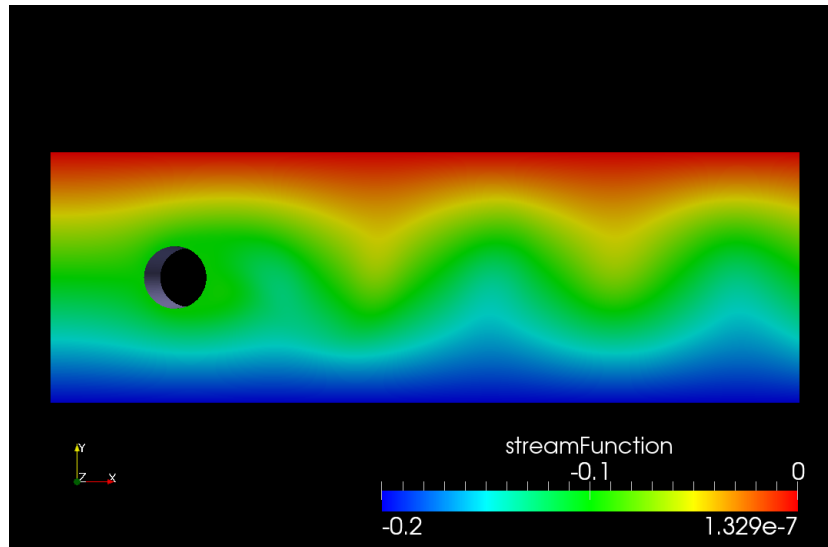
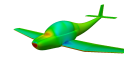


Figure 3.22: Stream function of the velocity of the *bidimensional cylinder* case for $Re = 195$ at $t = 1.75$ s

As it can be seen, it follows the same trend as the streamlines but with a continuous appearance. The colour indicates the value of the stream function in a particular point (note that this utility generates a point field), taking the $\psi = 0$ streamline as



the one at the **top** patch of the domain.

It can be proved that the volumetric flow rate between two streamlines is equal to the difference between their stream functions. This helps in the validation of the `streamFunction` utility. At the inlet:

$$Q = V \cdot S = 5 \cdot 0.4 \cdot 0.1 = 0.2 \text{ m}^3/\text{s}$$

$$Q = \psi_n - \psi_0 = 0.2 \text{ m}^3/\text{s}$$

3.6.5 Conversion to VTK

It is possible to convert data from *OpenFOAM*® to VTK format. For instance, access `cylinder195` and type:

```
foamToVTK
```

A new directory appears within the case containing the VTK data. Since it is a worldwide used format, it is also possible to open it with **ParaView**.

*Example: Once in **ParaView**, click on the "open" icon, access the VTK directory of the case and click on "cylinder". Within the **Pipeline Browser**, a new module has appeared. It is the **cylinder** patch whose shape can be used to give more realism to the results of the simulation. This same procedure can be carried out for each one of the defined patches of the case.*