

Simulation of cylinder flow at different Reynolds numbers

Izabela Gesla

June 4, 2021

This project was made in OpenFoam 4.0 based on Asmaa Hadane open tutorial ¹. The aim was to visualise different structures in the laminar flow around a cylinder varying on Reynolds number.

Contents

1	Introduction	2
2	Dataset and analysis	2
2.1	Initial parameters	2
2.2	Inlet velocity	10
3	Results	11
3.1	Case 1	11
3.2	Case 2	11
3.3	Case 3	12
3.4	Case 4	12
3.5	Case 5	14
4	Conclusion	16

¹<https://www.youtube.com/watch?v=Udt3RhkgKw>

1 Introduction

This project aims to visualise structures that can be observed around a circular cylinder. There is couple of flow regimes determined for laminar flow, the graphic representation is shown in the Figure 1 [1].

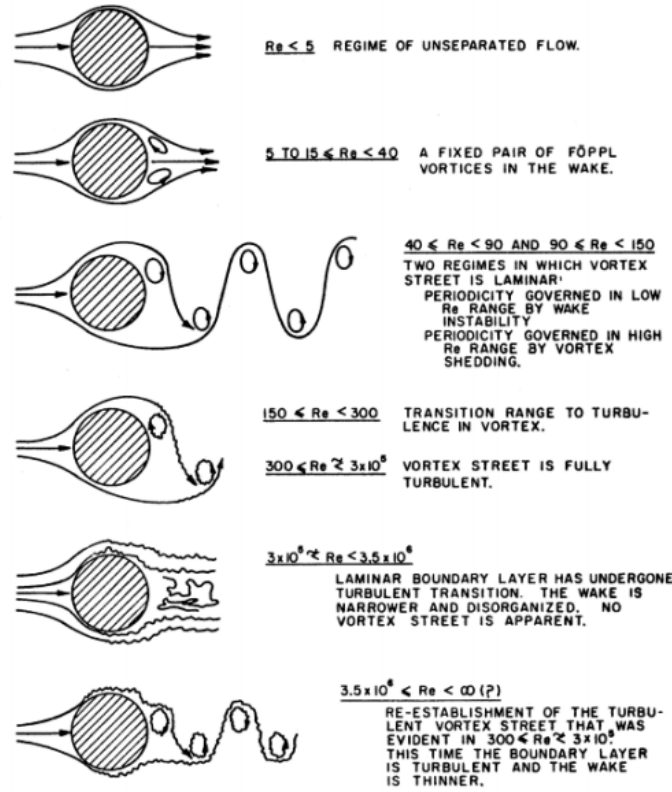


Figure 1: Flow regimes [1].

Structures that are created when considering flow around a circular cylinder can be examined by using numerical simulations, such as solver SimpleFoam in OpenFoam. This specific solver uses the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm to solve problems that are incompressible. In this particular case only laminar case will be examined, so in file "turbulenceProperties" in RASProperties the simulationType will be set to laminar.

In this report, the visualisation of different structures will be presented using parameters presented in the tutorial and modifying inlet velocity in folder "0" in file "U".

2 Dataset and analysis

2.1 Initial parameters

2.1.1 U

Analyses was carried out by varying inlet velocity in order to develop different structures. Base file "U" represents initial conditions of inlet velocity. In order to note the parameter in the file below - for only report purpose - the inlet velocity will be represented as "x", which will later be replaced by the values of corresponding inlet velocities.

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    location     "0";
    object       U;
}

dimensions      [0 1 -1 0 0 0 0];

internalField    uniform (0 0 0);

boundaryField
{
    inlet
    {
        type          fixedValue;
        value          uniform (x 0 0);
    }
    outlet
    {
        type          zeroGradient;
    }
    wall
    {
        type          fixedValue;
        value          uniform (0 0 0);
    }
    obstacle
    {
        type          fixedValue;
        value          uniform (0 0 0);
    }
    frontAndBack
    {
        type          empty;
    }
}

```

2.1.2 p

Other settings files will remain unchanged. File "p" in folder file "0" will remain as:

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    location     "0";
    object       p;
}

```

```

dimensions      [0 2 -2 0 0 0 0];

internalField    uniform 0;

boundaryField
{
    inlet
    {
        type      zeroGradient;
    }
    outlet
    {
        type      fixedValue;
        value      uniform 0;
    }
    wall
    {
        type      zeroGradient;
    }
    obstacle
    {
        type      zeroGradient;
    }
    frontAndBack
    {
        type      empty;
    }
}

```

2.1.3 transportProperties

File "transportProperties" located in "constant" folder file contains the information about the viscosity of a fluid represented by the Greek number ν (nu). The viscosity will be later used to calculate Reynolds number and velocity corresponding to it's value.

```

FoamFile
{
    version      2.0;
    format        ascii;
    class         dictionary;
    location      "constant";
    object        transportProperties;
}

transportModel  Newtonian;

nu              nu [0 2 -1 0 0 0 0] 1.5e-5;

```

2.1.4 turbulenceProperties

Next, file "turbulenceProperties", which includes the simulation model used, which is RAS - Reynolds Averaged Simulation turbulence closures based on linear and non-linear eddy viscosity models, and Reynolds stress transport models.

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       RASProperties;
}
simulationType laminar;
```

2.1.5 blockMeshDict

In "system" folder, there is also a file "blockMeshDict" which will define our geometry and from which we can deduce what is the diameter of the cylinder. The points of the edges defined as arc has a coordinates 2.5 and -2.5, so from that we know that the diameter has to be 5. At the beginning of the file there is a information about the scaling factor, which is convertToMeter and it equals to 0.01, so we know that units are in cm. This information will be next used to calculation of Re number and velocity.

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant/polyMesh";
    object       blockMeshDict;
}

convertToMeters 0.01;

vertices
(
    // pre-block
    (-25   -10   0.5) // 0
    (-10   -10   0.5) // 1
    (-10   -10  -0.5) // 2
    (-25   -10  -0.5) // 3

    (-25    10   0.5) // 4
    (-10    10   0.5) // 5
    (-10    10  -0.5) // 6
    (-25    10  -0.5) // 7

    // obstacle blocks
    ( 10   -10   0.5) // 8
    ( 10   -10  -0.5) // 9

    (-1.767766953 -1.767766953 0.5) // 10
```

```

( 1.767766953 -1.767766953 0.5) // 11
( 1.767766953 -1.767766953 -0.5) // 12
(-1.767766953 -1.767766953 -0.5) // 13

(-1.767766953 1.767766953 0.5) // 14
(-1.767766953 1.767766953 -0.5) // 15

( 1.767766953 1.767766953 0.5) // 16
( 1.767766953 1.767766953 -0.5)

( 10 10 0.5) // 18
( 10 10 -0.5) // 19

// post-block
( 65 -10 0.5) // 20
( 65 -10 -0.5) // 21
( 65 10 0.5) // 22
( 65 10 -0.5) // 23
);

blocks
(
  // pre-block
  hex ( 0 1 2 3 4 5 6 7) ( 60 1 30) simpleGrading (1 1 1)

  // obstacle blocks
  hex ( 1 8 9 2 10 11 12 13) (30 1 30) simpleGrading (1 1 1) // bottom
  hex ( 1 10 13 2 5 14 15 6) (30 1 30) simpleGrading (1 1 1) // left
  hex (14 16 17 15 5 18 19 6) (30 1 30) simpleGrading (1 1 1) // top
  hex (11 8 9 12 16 18 19 17) (30 1 30) simpleGrading (1 1 1) // right

  // post-block
  hex ( 8 20 21 9 18 22 23 19) (180 1 30) simpleGrading (1 1 1)
);

edges
(
  arc 10 11 ( 0 -2.5 0.5)
  arc 12 13 ( 0 -2.5 -0.5)
  arc 14 10 (-2.5 0 0.5)
  arc 15 13 (-2.5 0 -0.5)
  arc 14 16 ( 0 2.5 0.5)
  arc 15 17 ( 0 2.5 -0.5)
  arc 16 11 ( 2.5 0 0.5)
  arc 17 12 ( 2.5 0 -0.5)
);

boundary
(
  inlet
  {
    type patch;
    faces
    (
      ( 0 4 7 3)
    );
  }
  outlet
  {
    type patch;
    faces

```

```

        (
            (20 21 23 22)
        );
    }
    wall
    {
        type wall;
        faces
        (
            ( 4 5 6 7)
            ( 5 18 19 6)
            (18 22 23 19)

            ( 0 1 2 3)
            ( 1 8 9 2)
            ( 8 20 21 9)
        );
    }
    obstacle
    {
        type wall;
        faces
        (
            (10 11 12 13)
            (11 16 17 12)
            (14 16 17 15)
            (10 14 15 13)
        );
    }
    frontAndBack
    {
        type empty;
        faces
        (
            ( 0 1 5 4)
            ( 1 10 14 5)
            (14 16 18 5)
            (11 8 18 16)
            ( 1 8 11 10)
            ( 8 20 22 18)

            ( 3 7 6 2)
            ( 2 6 15 13)
            (15 6 19 17)
            (12 17 19 9)
            ( 2 13 12 9)
            ( 9 19 23 21)
        );
    }
};

mergePatchPairs
(
);

```

2.1.6 controlDict

File "controlDict" is a dictionary used to specify the main case controls, e.g. timing information, write format, and optional libraries that can be loaded at run time. In these simulations, we will carry 10 000 interactions with 100 write time interval. This will be specifically important to visualise changes in flow at higher Re number.

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       controlDict;
}

application     simpleFoam;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         10000;

deltaT          1;

writeControl     timeStep;

writeInterval    100;

purgeWrite       0;

writeFormat      ascii;

writePrecision   6;

writeCompression compressed;

timeFormat       general;

timePrecision    6;

runTimeModifiable yes;
```

2.1.7 fvSchemes

```
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
```



```

    object      fvSchemes;
}

ddtSchemes
{
    default      steadyState;
}

gradSchemes
{
    default      Gauss linear;
}

divSchemes
{
    default      none;
    div(phi,U)    bounded Gauss limitedLinearV 1;
    div(phi,k)     bounded Gauss limitedLinear 1;
    div(phi,epsilon) bounded Gauss limitedLinear 1;
    div((nuEff*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default      Gauss linear corrected;
}

interpolationSchemes
{
    default      linear;
}

snGradSchemes
{
    default      corrected;
}

```

2.1.8 fvSolution

```

FoamFile
{
    version      2.0;
    format        ascii;
    class        dictionary;
    location      "system";
    object        fvSolution;
}

solvers
{
    "(p|U|k|epsilon)"
    {
        solver      GAMG;
        tolerance    1e-06;
        relTol       0;
    }
}

```

```

        smoother      GaussSeidel;

        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator faceAreaPair;
        mergeLevels 1;
    };
}

SIMPLE
{
    nNonOrthogonalCorrectors 1;
    consistent true; // SIMPLEC

    residualControl
    {
        "(p|U|k|epsilon)" 1e-08;
    }
}

relaxationFactors
{
    p            0.5;
    U            0.1;
    k            0.3;
    epsilon      0.3;
}

```

2.2 Inlet velocity

In order to determine the velocity needed for the initial conditions, the Reynold's number formula will be used.

$$Re = \frac{vd}{\nu} \quad (1)$$

We know which Reynolds number corresponds to its flow regimes from graphic representations in the Figure 1 [1]. So the values of Re were selected and then velocities were set.

Case	"x" inlet velocity [m/s]	Reynolds number
Case 1	0.0009	3
Case 2	0.009	30
Case 3	0.03	100
Case 4	0.054	180
Case 5	0.3	1 000

Table 1: Velocity and Re

Case 4 is in fact a case that was studied in the tutorial that was mentioned in the introduction. However, to fully understand the creation of the vortex structures in the laminar flow the analyses will be carried out from almost motionless (Re=3) up to nearly turbulent flow (Re =1 000, turbulent flow from 2 100).

3 Results

The results of the simulation carried out in OpenFoam will be presented below. The standard initialization of calculation was made by commands blockMesh to generate geometry and mesh, then simpleFoam, a solver that employs the SIMPLE algorithm to solve the continuity and momentum equations [2] and paraview to visualise results in post-processing application ParaView.

```
blockMesh
simpleFoam
paraview
```

Results presented for Case 1,2 and 3 are visualisations at last timestep of simulation (10 000 iteration), showing fully developed flow at certain flow regime. For case 4 and 5 the visualisation presented will include also steps between initial conditions and fully developed flow.

3.1 Case 1

For this case $Re = 3$, so the stream in Figure 2 represents the unseparated streaming flow.

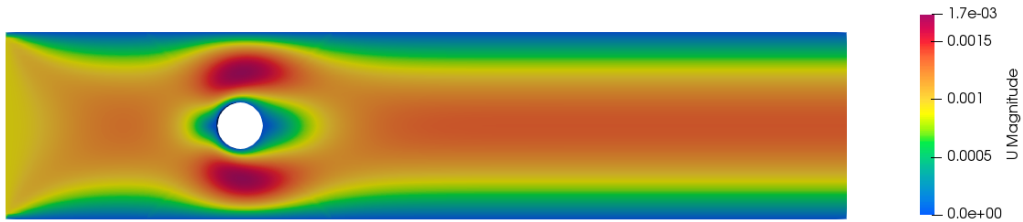


Figure 2: Case 1

3.2 Case 2

For this case $Re = 30$ and we can see in Figure 3 that stream is slightly different from first one, the sides velocities are higher and at the back of the cylinder we can observe a zero velocity area, which can indicate creation of vortexes.

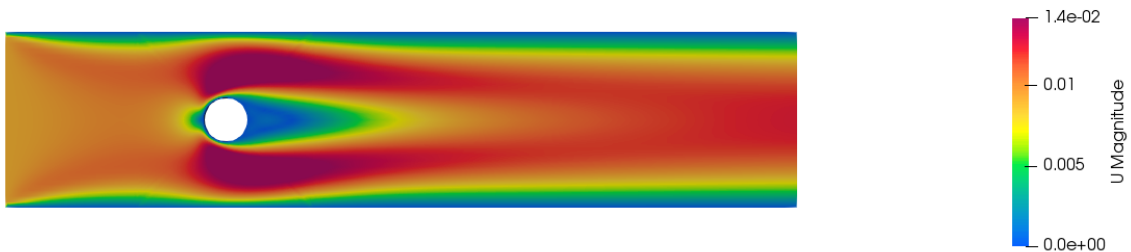


Figure 3: Case 2

3.3 Case 3

For this case $Re = 100$, differently from the graphic representation in Figure 1[1], we cannot observe yet any vortices, but it is shown that the stream is stronger and the borders are clearly marked.

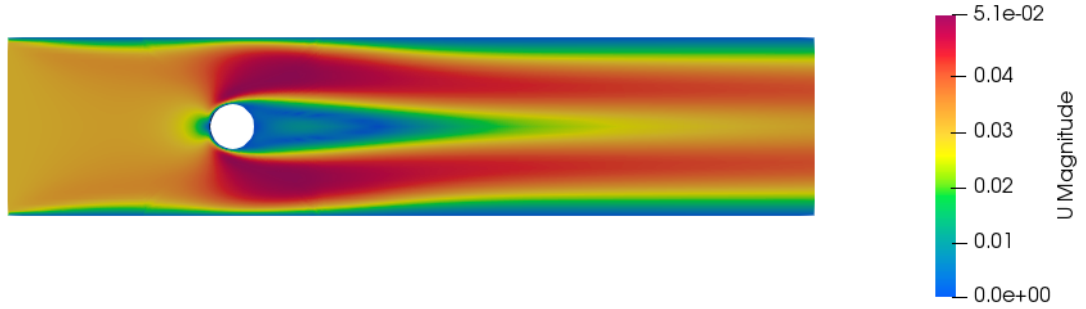


Figure 4: Case 3

3.4 Case 4

For this case $Re = 180$, that indicate the flow regime is a transition range to turbulence in vortex. That means we can observe the creation of the vortex around a cylinder. The final form of the developed flow is shown in the Figure 5, the creation of the vortices in the Figure 6.

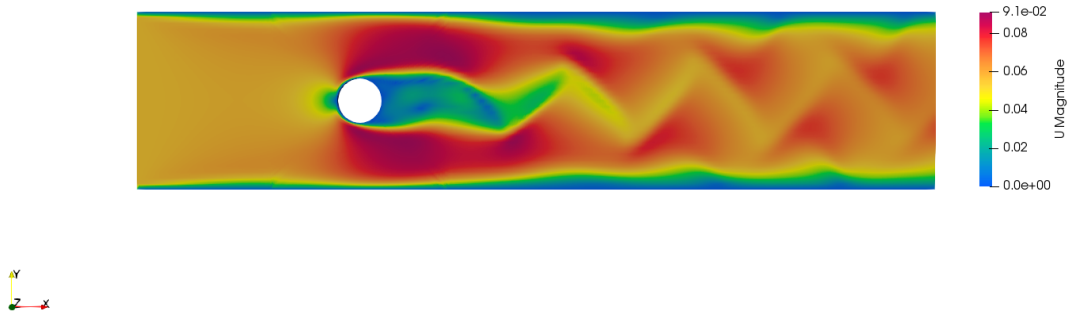


Figure 5: Case 4 - developed flow

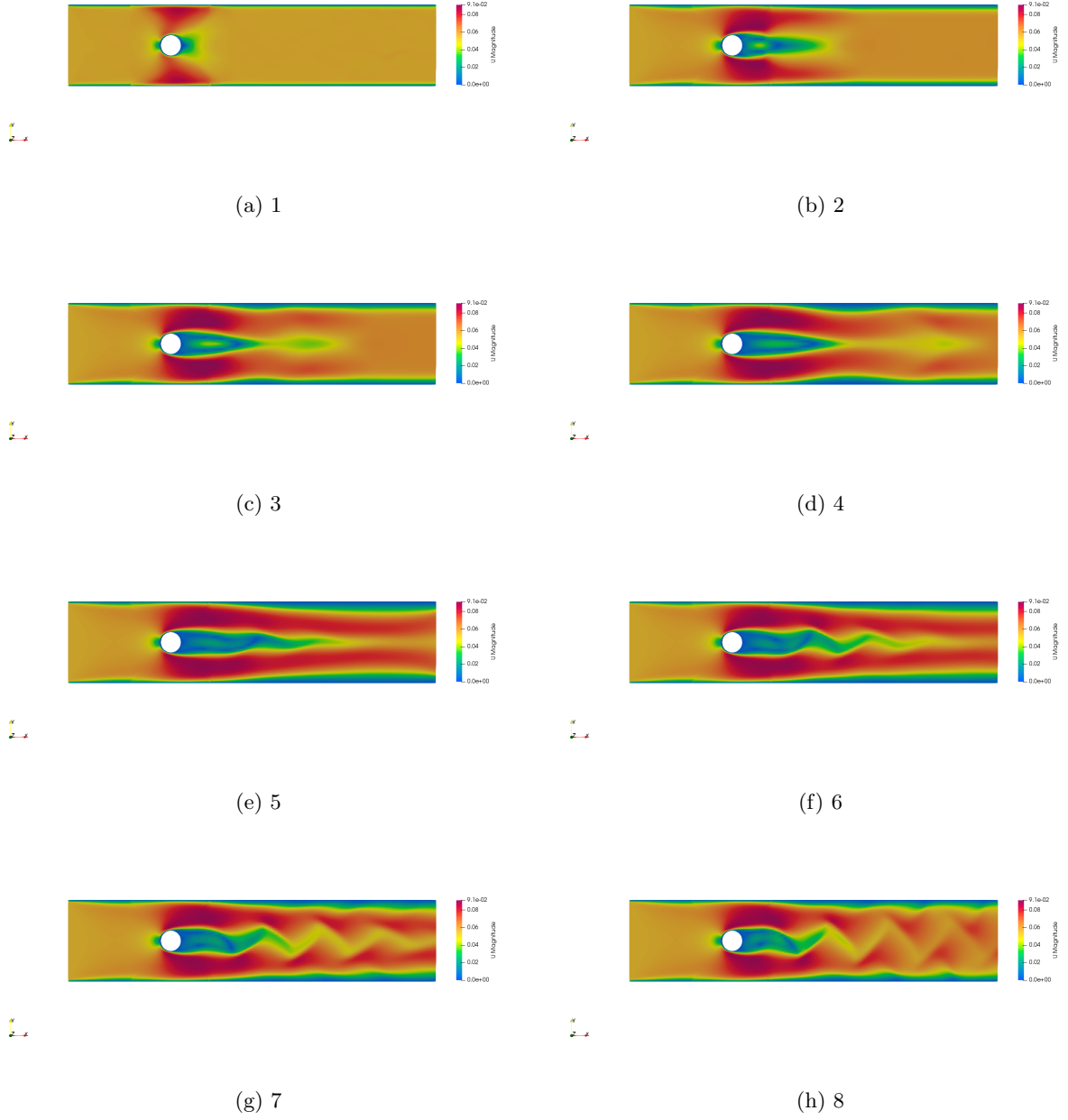


Figure 6: Flow development in Case 4

3.5 Case 5

For this case $Re = 1000$, that indicate the flow regime is defined as the boundary layer is laminar up to the separation point, the vortex street is turbulent and the wake flow field is increasingly three-dimensional. That process is notable in the Figure 9, where steps illustrate the creation of the turbulent vortex. The unsteady state when vortex was creating is marked in the Figure 7, and the final form of the developed flow is shown in the Figure 8.

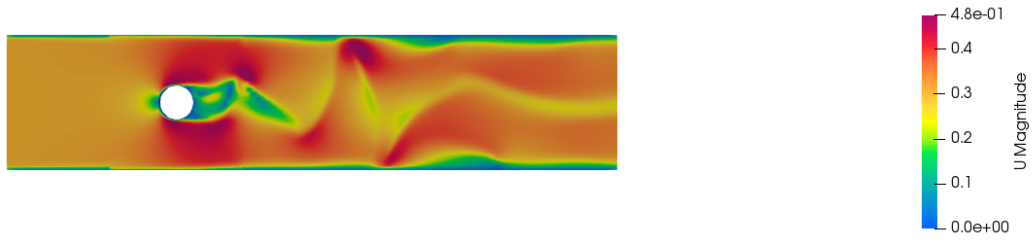


Figure 7: Case 5 - creation

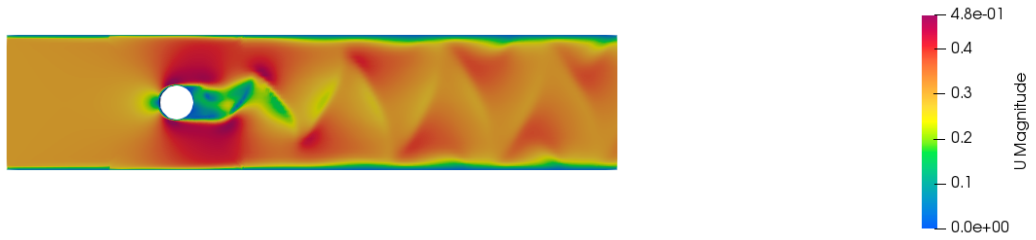


Figure 8: Case 5 - developed flow

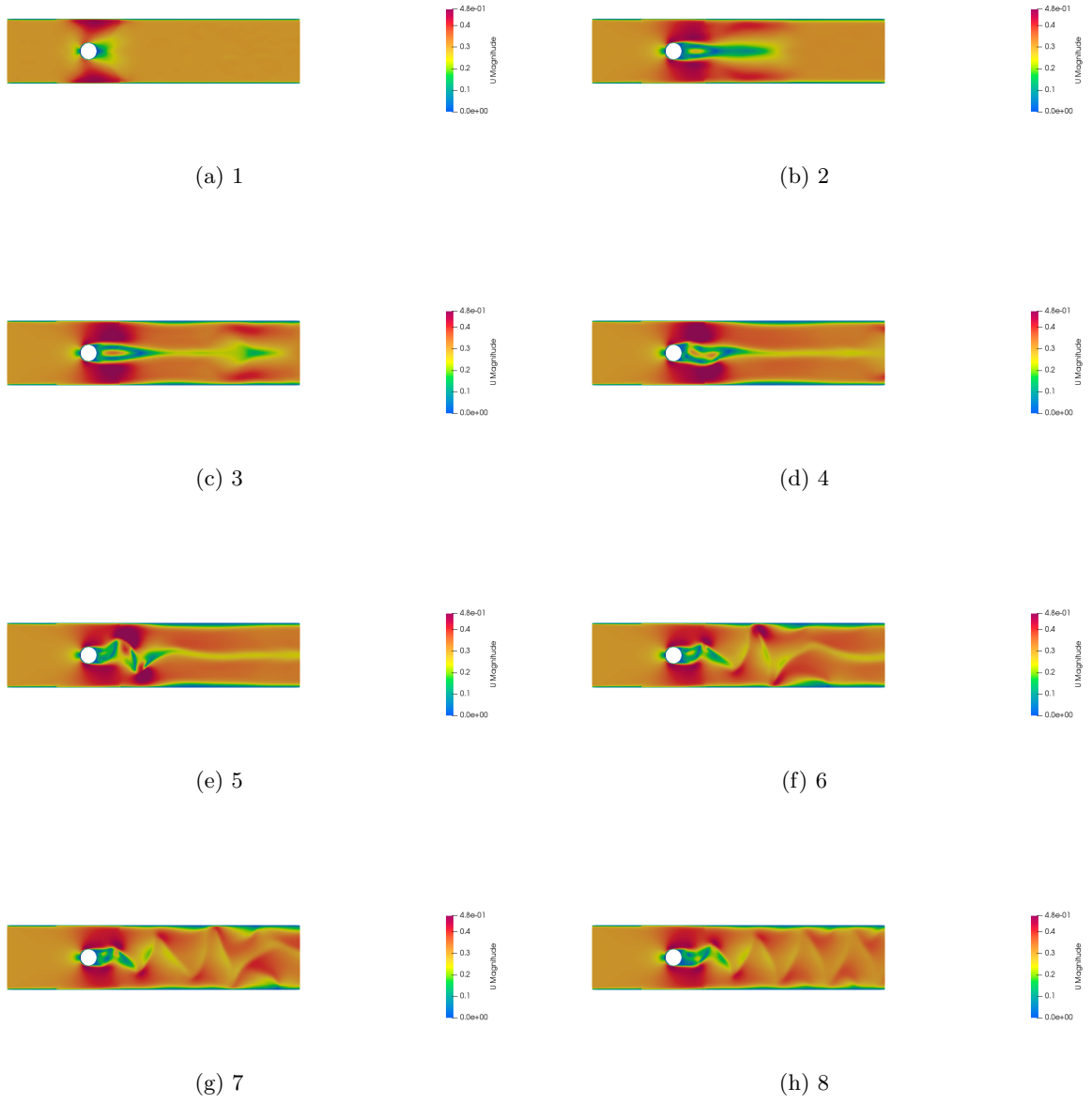


Figure 9: Flow development in Case 5

4 Conclusion

Thanks to OpenFoam, a toolbox with numerical solvers and post-processing utility ParaView, it is possible to visualise different flow regimes. The subject of this report analysis was a cylinder in laminar flow ($Re < 2300$). The results clearly confirm the structures described in the literature for different flow regimes [1].

References

- [1] J.H. Lienhard. Synopsis of lift, drag, and vortex frequency data for rigid circular cylinders. *Technical Extension Service*, 1966.
- [2] OpenCFD Ltd. Openfoam: User guide v2012. the open source cfd toolbox.