Open∇FOAM®

Escola Superior d'Enginyeries Industrials, Aeroespacial i Audiovisual de Terrassa

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Simulation of the first stages of a turbofan using OpenFOAM

Degree: Aerospace Engineering

Course: Application of Open-Source CFD to Engineering Problems

Delivery date: 09-12-2016

Students:

Herrán Albelda, Fernando

Martínez Viol, Víctor

Morata Carranza, David

Contents

1 Introduction

During the development of the course *'Application of Open-Source CFD to engineering problems'* we have learned the basics of how to use and solve real-world cases and situations related with fluid mechanics using OpenFOAM, an open source CFD tool. During the first days of the course, several possible projects were presented and we had to choose one of them. After agreeing with the professor, we decided to simulate a project that was not on that list. Since we are really interested in propulsion, we thought that it was a good idea to try to simulate the flow inside the first stages of compression of a turbofan engine.

To do that, we have been gathering lots of information of this type of engines and their typical working conditions in order to obtain realistic results.

These type of engine is the most used propulsion system in the aerospace industry. It presents several advantages to other systems such as the turbohelix or the turbojet; for example, this kind of engine takes advantage of the flow that goes through the fan (that cannot be more compressed or heated given that the combustion chamber has a limited volume) and results in a higher thrust. Thus, it is no surprise that state-of-the-art planes such as the Airbus 380 or the Boeing 747 use this kind of propulsion system.

2 Case

2.1 Description of the case

The aim of this project is to analyze the airflow inside the first stages of a turbofan. The turbofan consists of an initial stage where a fan is placed. This first stage increases the pressure of the air that goes through it and, as it can be seen in 2.1, a part of the incoming airflow goes to the low pressure compression stage (main or primary flow) and the other goes to the conducts (secondary flow). As the secondary flow advances through the conduct, the main flow enters the engine core where it goes through the low pressure and high pressure compression stages. By increasing the pressure of the air, the density also increases; thus, a higher mass flow can be mixed with the fuel and burned in the combustion chamber. After this process, the gases go through the turbines, interchanging energy with them (pressure and velocity, mainly), and they arrive at the nozzle, where the air is accelerated, and a thrust force is obtained.
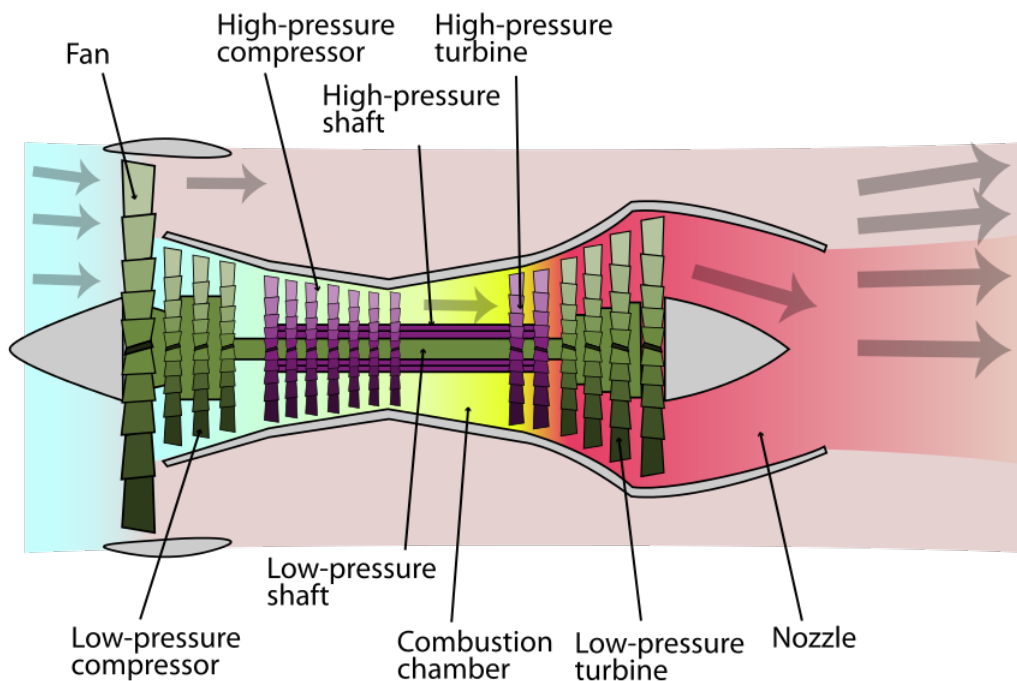

Figure 2.1: Parts of a turbofan

To simulate the flow in the first stages of the turbofan, we have downloaded the following turbine model from *GrabCad*, shown in 2.2. This model is pretty similar with the one shown above (2.1) and the simulation will take place between the back side of the fan and the back side of the second compressor.

The geometry of the *blockMesh* as well as the refined mesh of this parts will be shown and discussed in the next section.
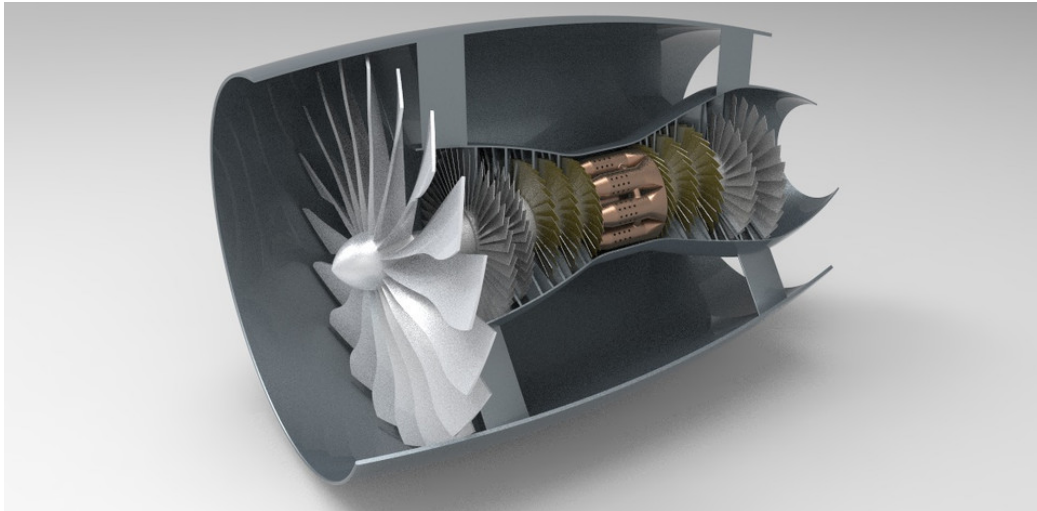

Figure 2.2: Model used

2.2 Hypotheses

Given that we are not able to use a supercomputer and the computational power that is available to us is very limited (since we are using our own personal computers to run this simulation), several hypotheses have to be made to alleviate the calculation time. These hypotheses are presented below.

- Viscous flow

- Incompressible flow

- Newtonian flow

- Stationary flow

Clearly, the flow is not incompressible; the aim of the turbofan is to compress the air to pursue a more efficient combustion. However, the compressible solvers are really difficult to use and the computation time is also higher given that we have a high number of control volumes (as discussed in the next section).

So, the simulation will take into account that it is a tridimensional, that it behaves as a newtonian fluid and it will be run under stationary flow conditions (that is: the velocity in the inlet is always uniform and has a fixed value).

3 Mesh generation

3.1 blockMesh

The definition of the *blockMeshDict* is the first part that needs to be defined in order to simulate the case. Only the first two stages of the turbofan engine will be simulated, so we have to make sure that the *blockMesh* includes them.

The domain of the mesh is a 0.58x2.3x2.3m rectangular prism and the vertices are as follows:

```
vertices
(
(0.77 0 0)
(1.35 0 0)
(1.35 2.3 0)
(0.77 2.3 0)
(0.77 0 2.3)
(1.35 0 2.3)
(1.35 2.3 2.3)
(0.77 2.3 2.3)
)
```

Once the boundaries of the *blockMesh* are defined, the number of cells that it will have has to be set. It has to be taken into account that a very dense mesh will not be efficient when simulating the case. On the other hand, a very coarse mesh will not be efficient either because additional divisions will have to be set when generating the refined mesh with *snappyHexMesh*. Thus, a compromise solution between a mesh with a very high number of cells and a very low number of cells has to be attained.

This basic mesh has been divided every 0.05m. It means that we have done 12 divisions in the x direction, 46 on the y direction and 46 more on the z direction. Obviously, the vertex numeration has been kept the same as the one that comes as default in every tutorial case; so, it was easier to

define the inlet face of the *blockMesh* as well as the outlet face that will be used to define the boundary conditions.

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  4.0                                   |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

convertToMeters 1;

vertices
(
    (0.77 0    0)
    (1.35 0    0)
    (1.35 2.3 0)
    (0.77 2.3 0)
    (0.77 0    2.3)
    (1.35 0    2.3)
    (1.35 2.3 2.3)
    (0.77 2.3 2.3)
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (12 46 46) simpleGrading (1 1 1)
);

edges
```

```
(
);

boundary
(
    frontAndBack
    {
        type patch;
        faces
        (
            (3 7 6 2)
            (1 5 4 0)
            (0 3 2 1)
            (4 5 6 7)
        );
    }
    inlet
    {
        type patch;
        faces
        (
            (0 4 7 3)
        );
    }
    outlet
    {
        type patch;
        faces
        (
            (2 6 5 1)
        );
    }
);

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

The log obtained when the mesh has been generated is as follows:

```
Writing polyMesh

--------

Mesh Information

--------

boundingBox:  (0.77 0 0) (1.35 2.3 2.3)

nPoints:  28717

nCells:  25392

nFaces:  79396

nInternalFaces:  72956

--------

Patches

--------

patch 0 (start:  72956 size:   2208) name:  frontAndBack

patch 1 (start:  75164 size:   2116) name:  inlet

patch 2 (start:  77280 size:   2116) name:  outlet


End
```
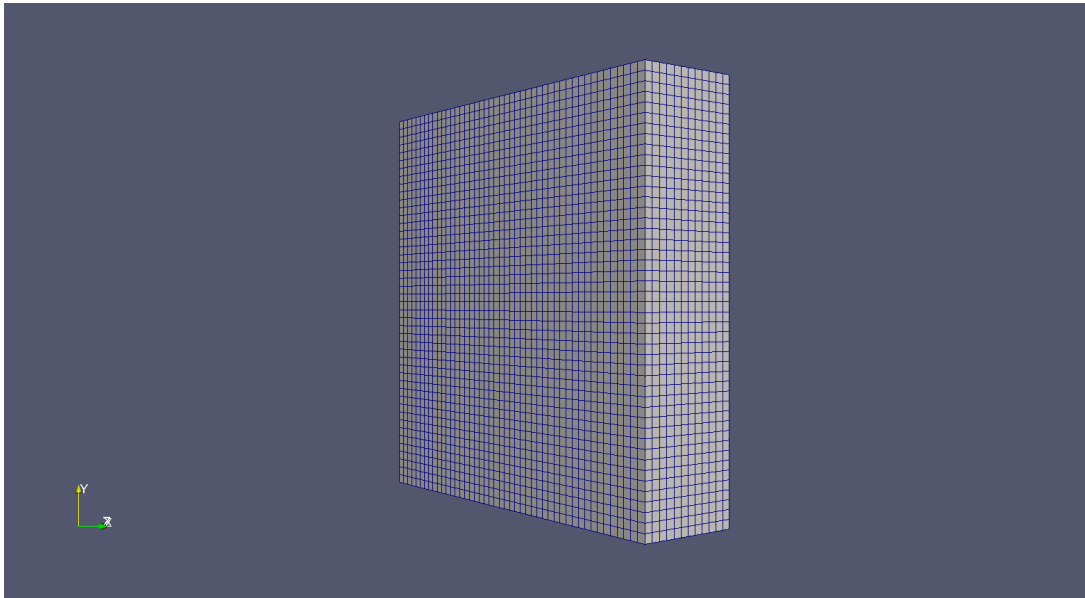

Figure 3.1: blockMesh

9

3.2   Mesh refinement

To refine the mesh, the *snappyHexMesh* utility, included in OpenFoam, which is used and several parameters have to be modified in order to obtain a dense mesh that is suitable for the simulation of this complex geometry. Additionally, it has to be considered that a particular geometry is has a relative velocity; this is, the rotor is rotating, and so the first and second stages of the Low Pressure Compressor of the turbofan engine, while the nacelle and the combustor are static.

FALTA EXPLICAR SURFACE EXTRACT

A comparison between the coarse mesh and the refined mesh generated is presented below.

3.2.1   Coarse mesh

The number of control volumes in the mesh is 40.
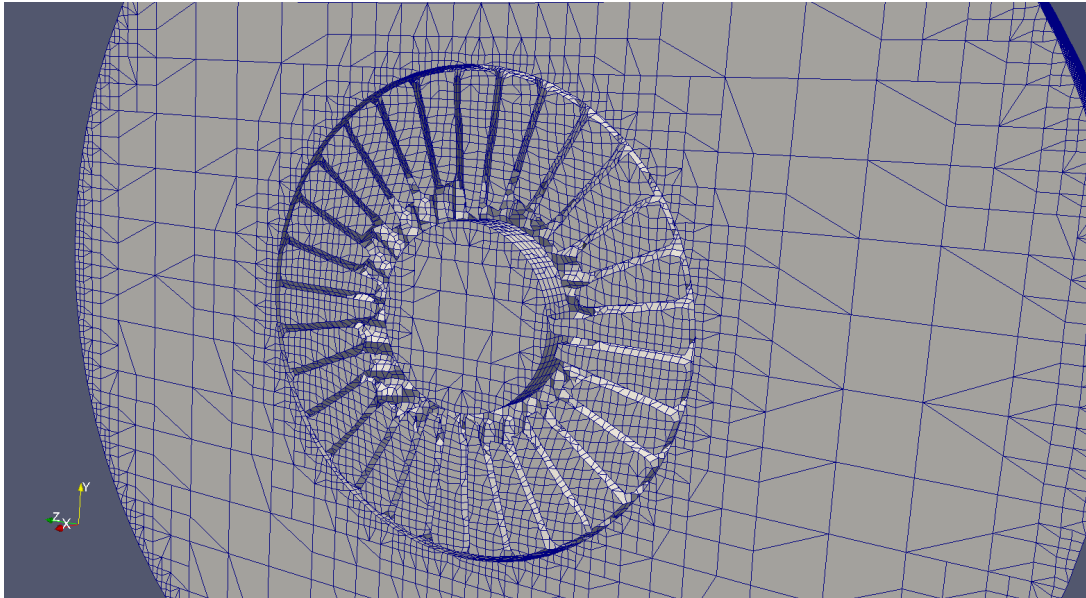
Figure 3.2: Detail of the coarse mesh
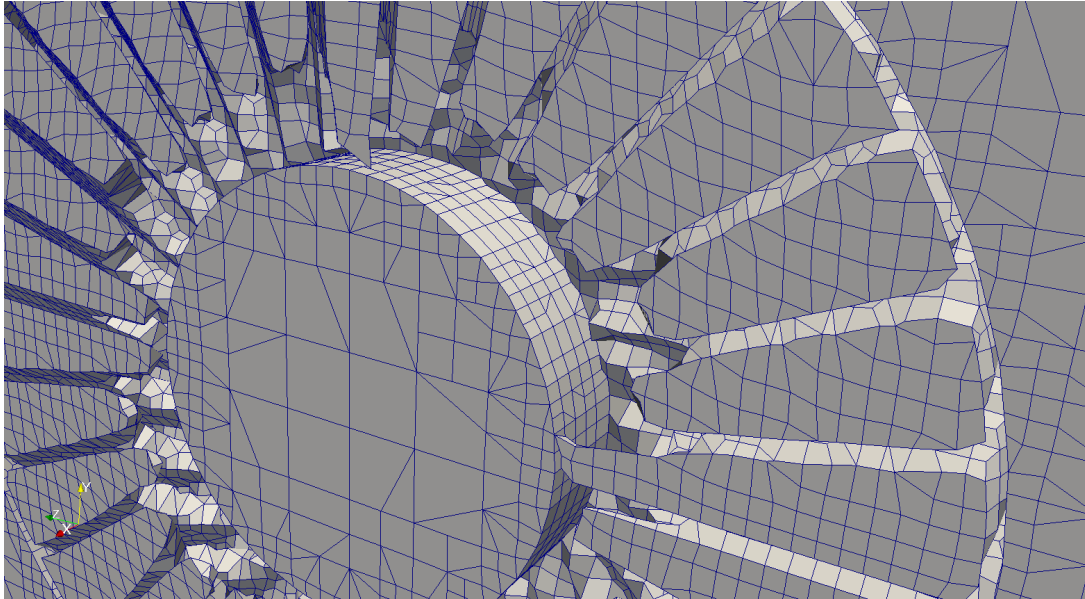
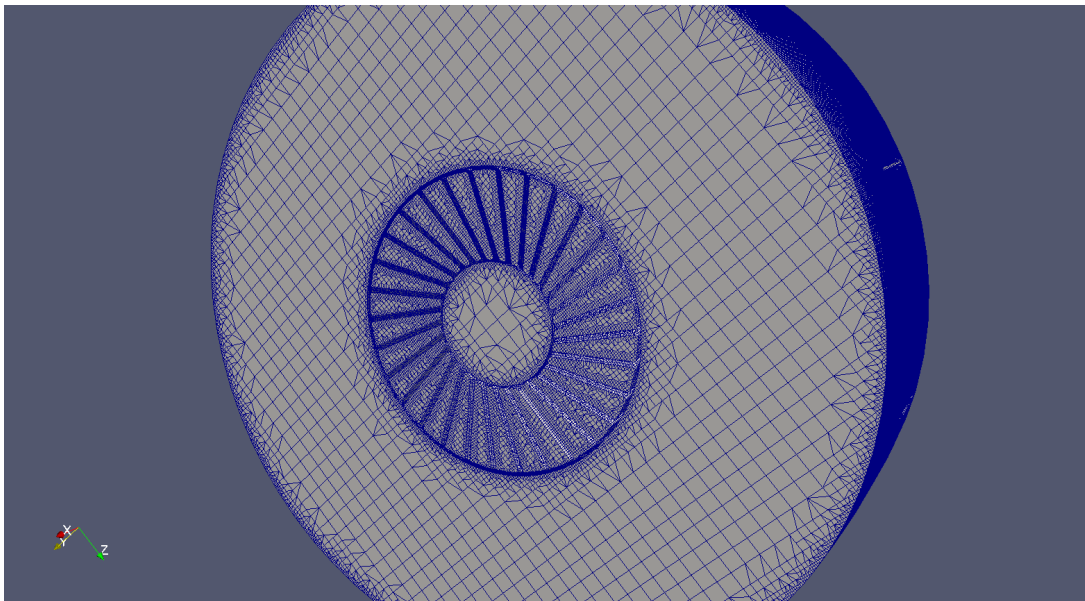Figure 3.3: Detail of the coarse mesh

Figure 3.4: Detail of the standard mesh
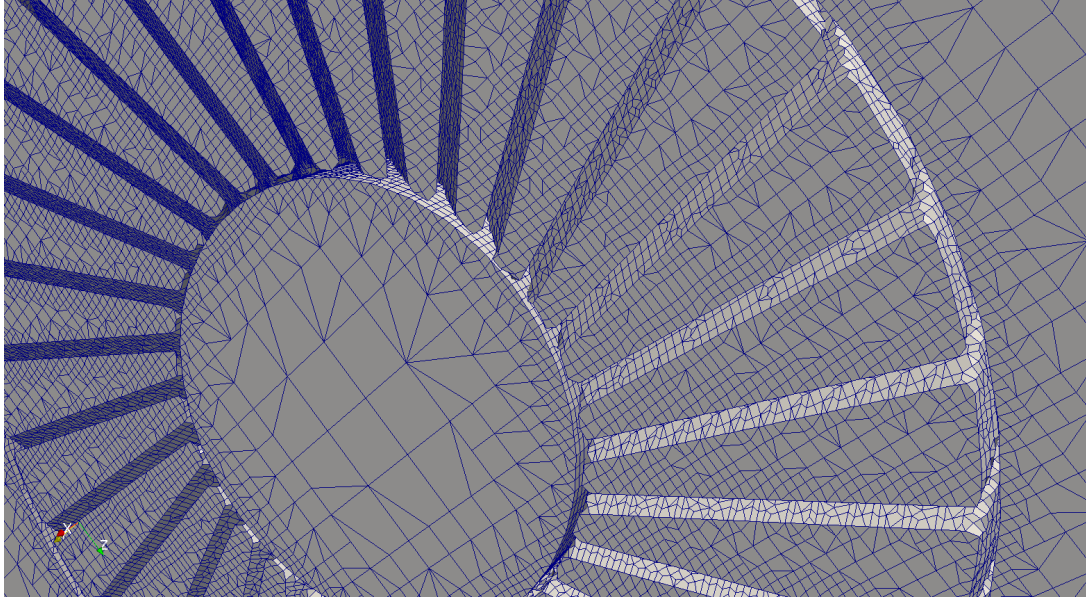
3.2.2 Standard mesh

3.2.3 Dense mesh

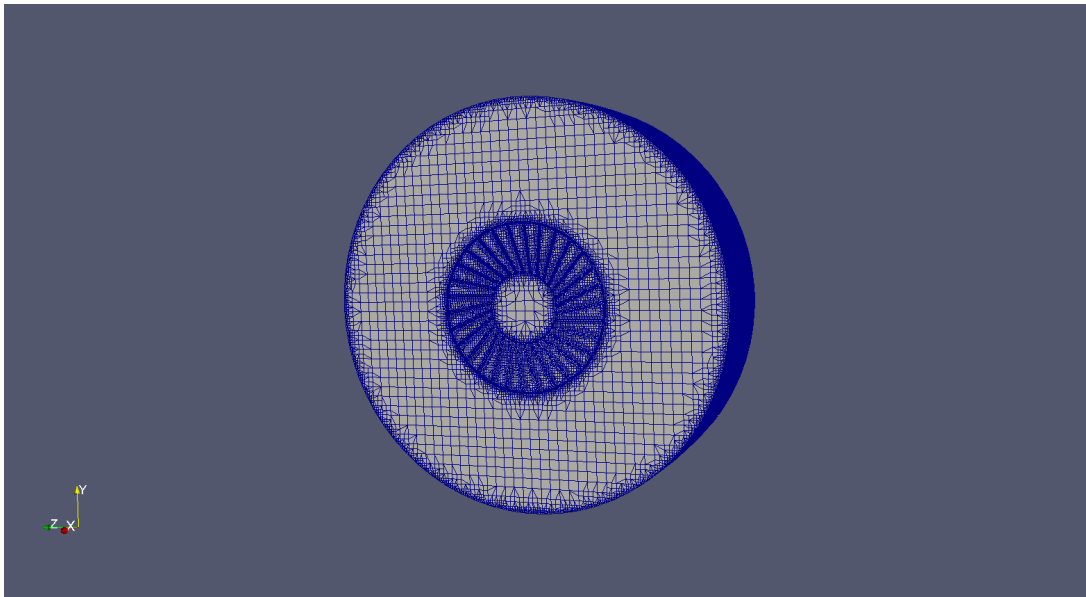Figure 3.5: Detail of the standard mesh
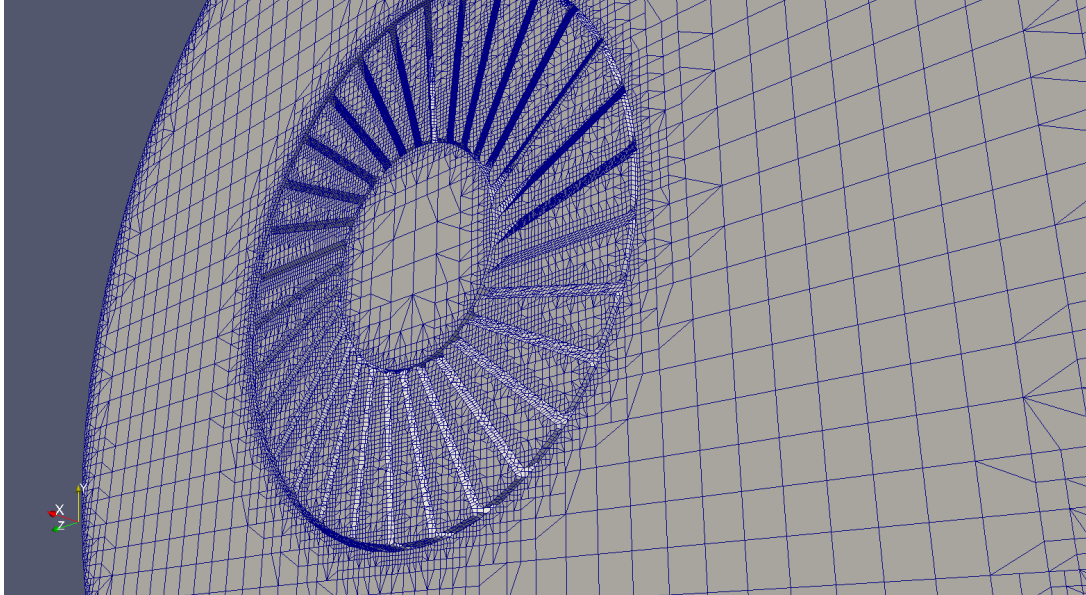

Figure 3.6: Detail of the coarse mesh
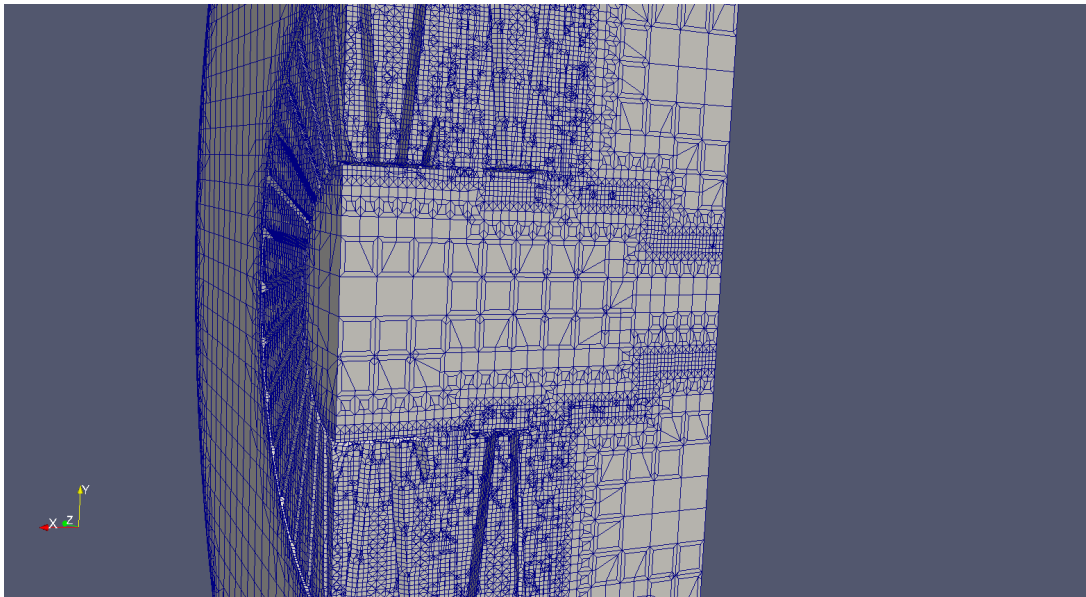
Figure 3.7: Detail of the dense mesh

Figure 3.8: Detail of the dense mesh

4 The problem

EMPTY

5   Simulation

EMPTY