

UNIVERSITY OF PERUGIA
Engineering Department
Master Degree in Mechanical Engineering
Academic Year:2012/2013



**IMMERSED BOUNDARY METHODOLOGY:
AN ALTERNATIVE APPROACH TO THE
CFD/3D ANALYSIS.**

Candidate:
Ledjan Hamzaj

Advisor:
Prof. Ing. Francesco Mariani

Co- Advisor:
Ing. Francesco Risi

1. Introduction

The object of this thesis work is to validate the immersed boundary method as a possible alternative to the traditional CFD/3D analysis techniques for industrial applications. For this purpose has been performed some simulations on the RB11.1 prototype front part, realized by the University of Perugia for the Formula Student competition.

Those simulations has been carried out by two different softwares: the first, KARALIT CFD is an innovative software based on the immersed boundary method produced by Karalit s.r.l. and the other one is STAR CCM+ a software based on the traditional approach.

The simulations are compared with the experimental data for the same vehicle geometry acquired from the Engineering Department of the University of Perugia wind tunnel.

2 Immersed boundary method

The term “immersed boundary method” was first used by Peskin (1972) who developed a method to simulate cardiac mechanics and associated blood flow. The distinguishing feature of this method was that the entire simulation was carried out on a Cartesian grid, which did not conform to the geometry of the heart, and a new procedure was formulated to impose the effect of the immersed boundary on the flow.

Since Peskin introduced this method, numerous modifications and refinements have been proposed and a number of variants of this approach now exist.

Nowdays the term immersed boundary is used to include all the methods that simulate viscous flows with immersed boundaries that do not conform to shape of these boundaries

Because of the not conform grid with the solid body, incorporate the boundary conditions request some modifications to the governing equations in proximity of the boundary.

$$\frac{Du}{Dt} = \frac{\partial u}{\partial t} + u \cdot \nabla u = \nu \nabla^2 u - \rho^{-1} \nabla p + f$$

$$\nabla \cdot u = 0$$

body-force term f to the Navier-Stokes equations makes possible to consider the object keeping a simple computational domain based on a Cartesian grid.

2.1 Ghost cell immersed boundary method

The force in N-S equations is correct for the case in which the position of the unknowns on the grid coincides with the immersed boundary. This condition is not always satisfied in case of complex geometries, so is necessary to implement an interpolation method. Many different techniques have been adopted one of them is the *ghost cell method* according to which once the body is immersed into the grid, completely independent of the body itself, the grid cells are labelled based on the position of their centers with respect to the body. With reference to figure 1 cells are classified as:

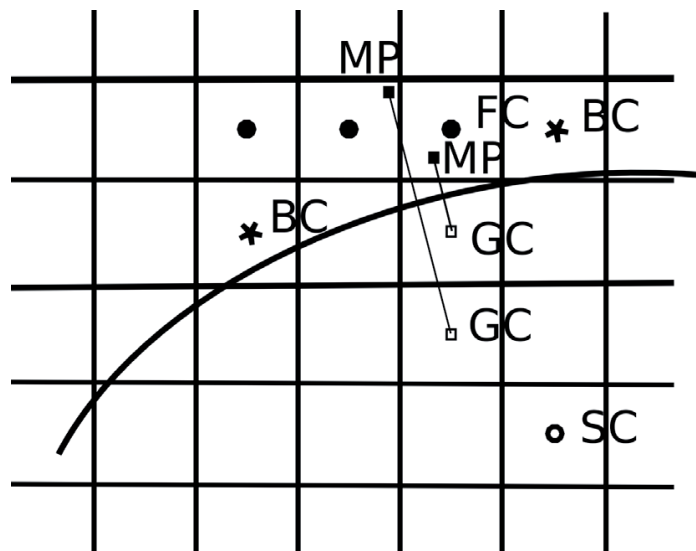


Figura 1 Ghost Cell method

Fully fluid cell (FC) when the entire cell falls inside the fluid region; fully solid cell (SC) when the entire cell falls inside the body; boundary cell (BC) when the

cell is cut by the body but its center is on the fluid side; ghost cell (GC) when the cell is cut by the body but its center is on the solid side.

The position of each ghost cell is reflected into the fluid region by mirroring the cell along the normal to the body surface passing through the cell center. All flow variables are reconstructed at each of those mirror points (MP) via an interpolation procedure that involves a given number of fluid cells surrounding the mirror point itself. The computed variables at mirror points are reflected back to the original ghost cells and are used to enforce the desired boundary conditions.

2.2 Local grid refinement technique

In the context of the immersed boundary techniques, very refined grids are often required close to the high curvature immersed surfaces to properly represent the details of the geometry. Grid resolution can be easily increased in the framework of unstructured meshes by locally inserting grid points and regenerating the connectivity between the points; the same procedure for Cartesian grids is not immediate as it will destroy the implicit ordering of the mesh points. Many approaches for local grid refinement of Cartesian meshes have been proposed; the most successful is based on the work by Powell, De Zeeuw and Pember. It is based on the idea of octree data structure in which every computational cell can be subdivided in eight children-cells. The main difficulty associated with this technique is the inherent complexity of the solution algorithm, which is reflected in high computational costs and memory requirements.

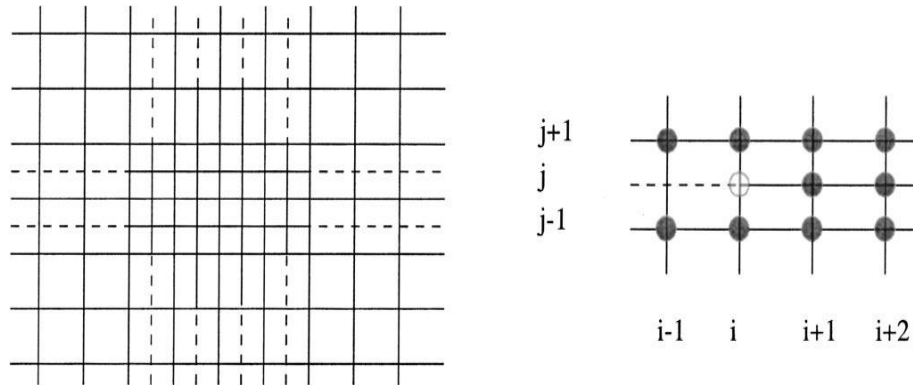


Figura 2 Cartesian mesh with local mesh refinement: dashed lines represent grid lines that are partially deleted

This approach is based on the observation that in a Cartesian mesh the locally refined grid can be viewed as a finer globally refined mesh with some grid line partially deleted. In figure 2 the central region of the grid is refined but the grid lines are extended to the boundary of the computational domain; in this way, a (i,j) ordering of the vertices can be retained.

3 KARALIT CFD

KARALIT CFD is a computational fluid dynamics software based on the *ghost cell method*, which belongs to the more general class of mathematical methods known as the *immersed boundary method*. This is not the only innovation of this software, another key point is the presence the *Apps*, basically templates that aim to simplify the interaction between users and IB method.

Before starting with the presentation of the tests performed in this thesis, would be useful to explain how to create a new simulation on KARALIT CFD and how to manage this simulation with the GUI.

First of all when creating a new case the software asks the user to choose the app which want to work on as shown in the figure 3.

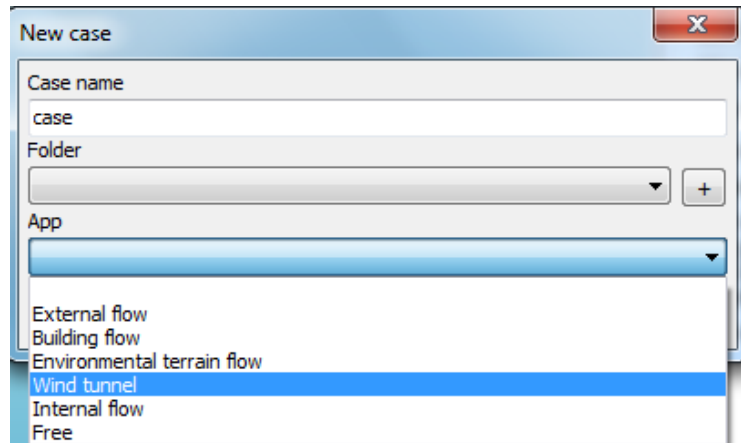


Figura 3 KARALIT CFD available Apps

In this case is chosen the *Wind tunnel* app which is the app used during the RB11.1 front part aerodynamic tests. Thereafter the working area is composed by two windows as shown in figure4: on the left, the visualization window describe step by step the modifications on the computational domain and tested model.

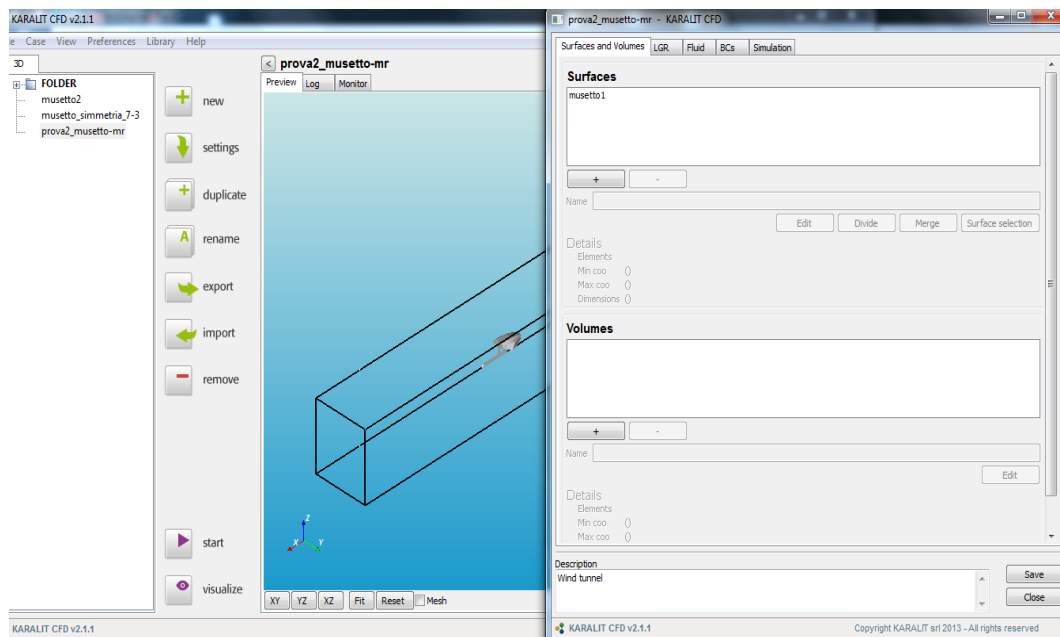
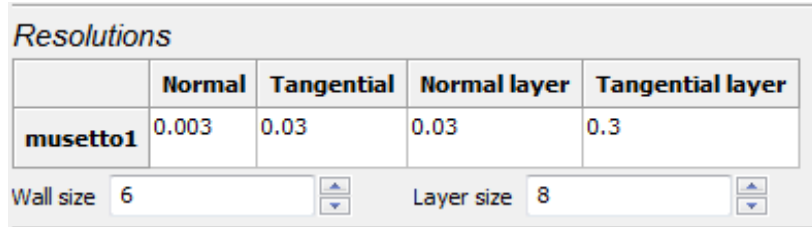


Figura 4 KARALIT CFD graphical user interface

On the right the settings window is composed by five tabs through which the user can define the simulation parameters. The most important is the LGR tab dedicated to the mesh generation, KARALIT CFD is able to generate the

computational grid by setting up few parameters of the *Resolution* section of this tab.



	Normal	Tangential	Normal layer	Tangential layer
musetto1	0.003	0.03	0.03	0.3

Wall size Layer size

Figura 5 Resolution section of LGR tab

Normal: normal dimension of wall cells

Tangential: tangential dimension of wall cells

Normal layer : thickness of intermediate layers in normal direction

Tangential layer: thickness of intermediate layers in tangential direction

Wall size: thickness of wall layer

Layer size: number of intermediate layers

The following figure shows the effect of *Wall size* parameter on a simple geometric model.

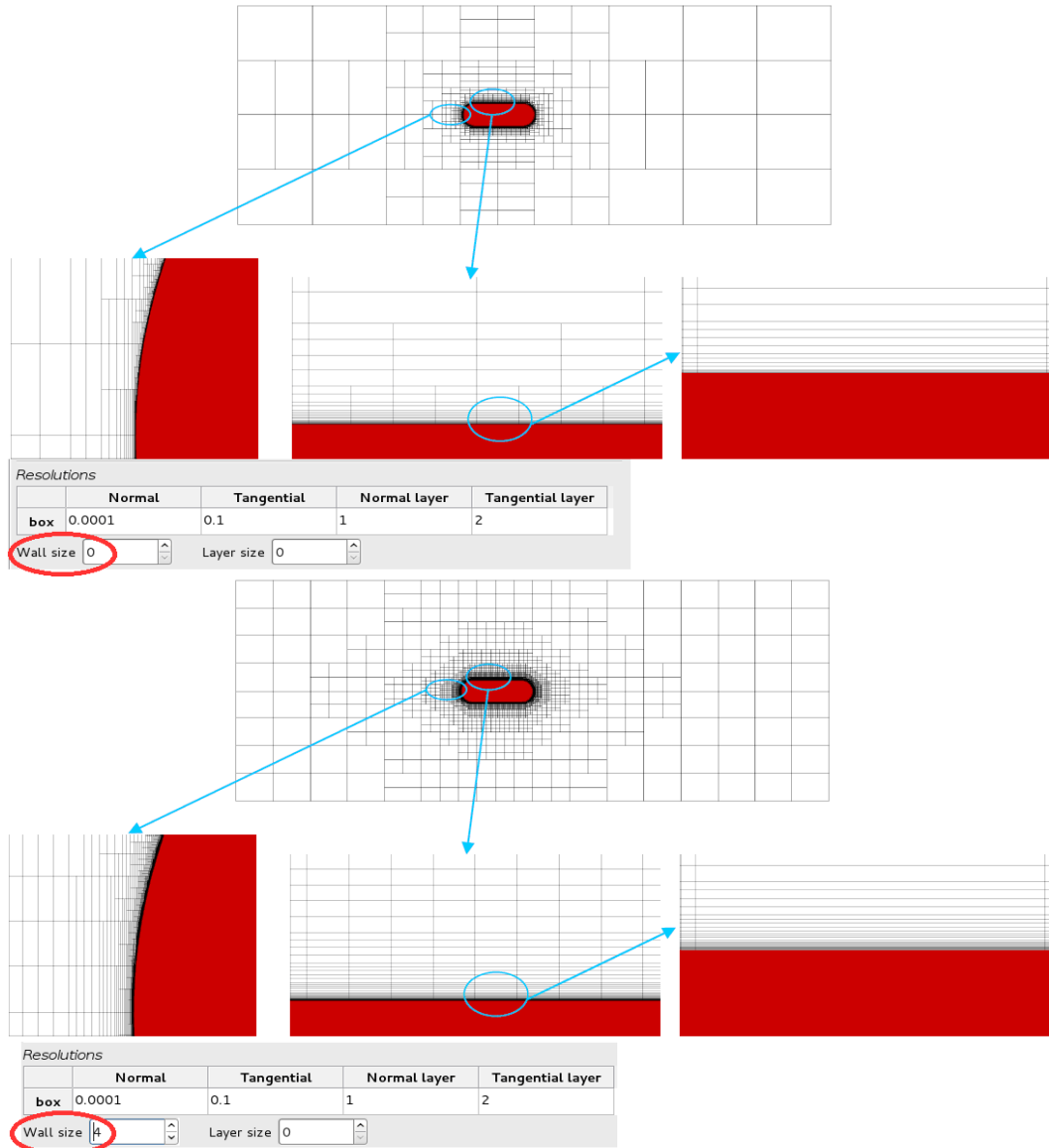


Figura 6 Wall size effect on grid generation

The result is an increment of the wall layer thickness passing from *Wall size*=0 to *Wall size*=4.

This other figure shows the effect of *Layer size* parameter on the previous geometric model.

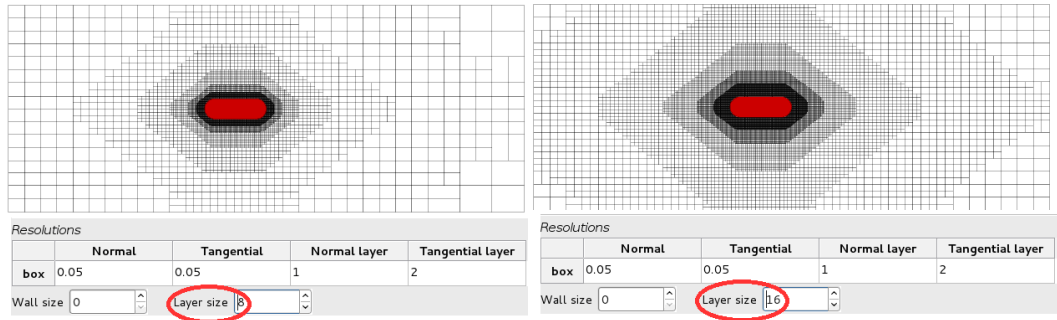


Figura 7 Layer size effect on grid generation

Passing from *Layer size=8* to *Layer size=16* the result is an increment of refinement and thickness on the intermediate layers.

4 Comparative analysis (KCFD vs. Star CCM+)

The wind tunnel tests has been carried out on the full scale model of the front part of RB11.1 vehicle, by setting a flow regime velocity of 80 km/h.

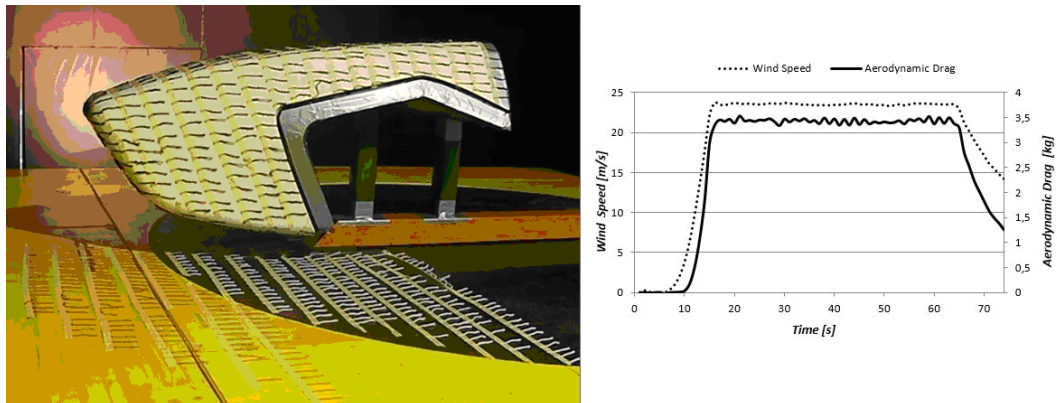


Figura 8 Wool tuft test (left); air velocity and aerodynamic drag time history (right)

From those tests was obtained an average value of the drag force coefficient $C_d=0.43$.

The first step on a CFD analysis is the creation of a CAD model. KARALIT CFD, thanks to immersed boundary method is able to distinguish the fluid cells from the solid cells keeping them out from the governing equation resolution. Even Star CCM+ is able to distinguish the fluid cells from the solid cells using the solid cells as computational domain. Therefore is necessary to model the fluid zone as a solid part and the solid zone as an empty part.

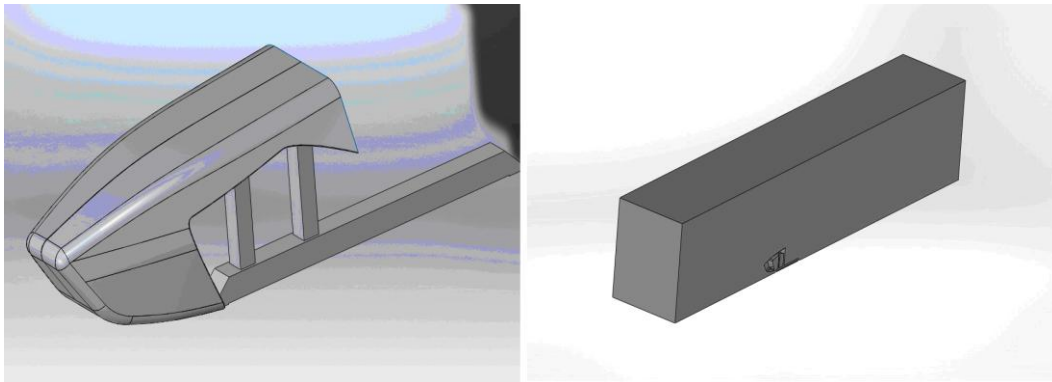


Figura 9 CAD model for KCFD(left); Star CCM+ (right)

To reduce the total cells number and in consequence the computational time was decided to use the symmetry plane of the geometric model dividing in half the computational domain.

During this thesis work was created two types of meshes for each one of the CFD software: a coarse one with about 1.8 million cells and a fine one with about 3.2 million cells.

The first mesh (~1.8 million cells) was generated defining the Resolution section parameters and creating a box window around the model extended even in the wake region as shown by the following figure.

Resolutions

	Normal	Tangential	Normal layer	Tangential layer
musetto1	0.003	0.03	0.03	0.3

Wall size 6 Layer size 8

Set on... Lock on center

Name musetto

Starting x -1.8 Resolution x 0.015

Starting y -0.35 Resolution y 0.05

Starting z 0.1 Resolution z 0.01

Width 3

Depth 0.7

Height 0.619

Figura 10 Mesh 1 computational grid set-up

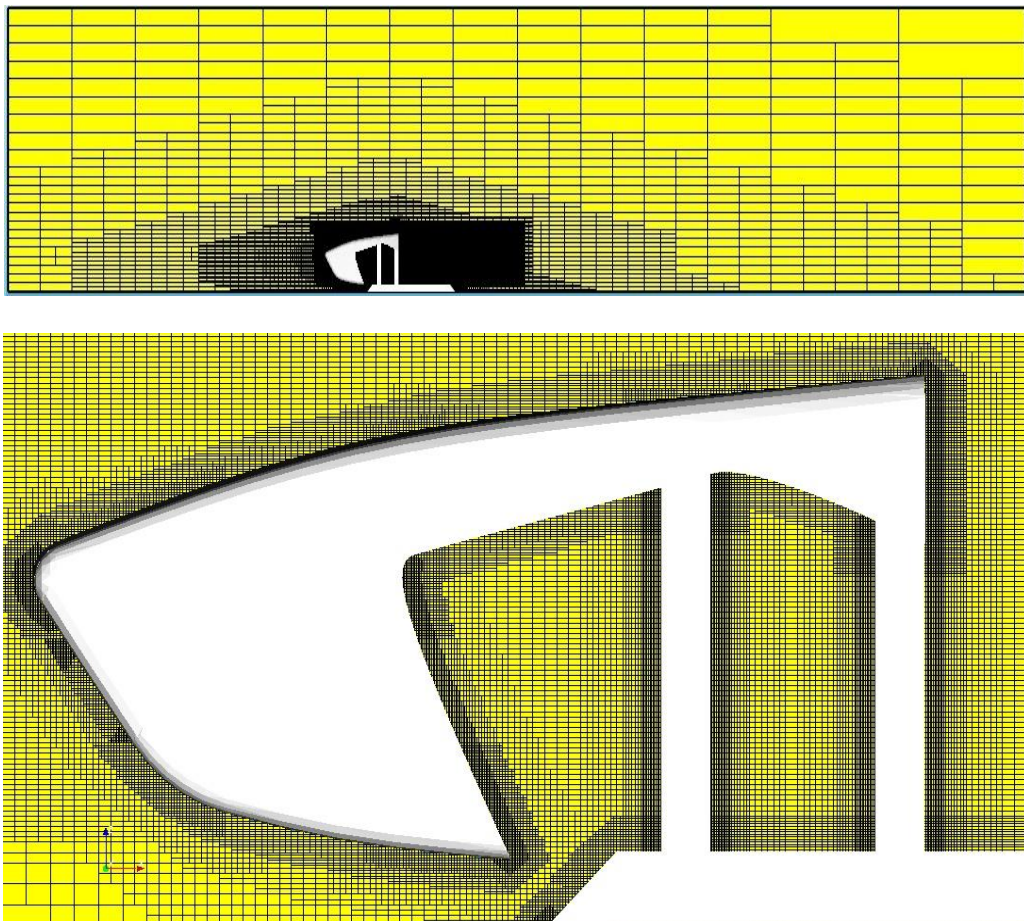


Figura 11 KCFD Mesh1 details

The second mesh (~3.2 million cells) was generated keeping the same box window with the same resolution around the model changing only the Resolution section parameters as below.

<i>Resolutions</i>				
	Normal	Tangential	Normal layer	Tangential layer
musettostl	0.0015	0.02	0.003	0.04

Wall size 0 Layer size 10

Figura 12 Mesh 2 resolution set-up

The result was a more accurate mesh, as shown from the following figures.

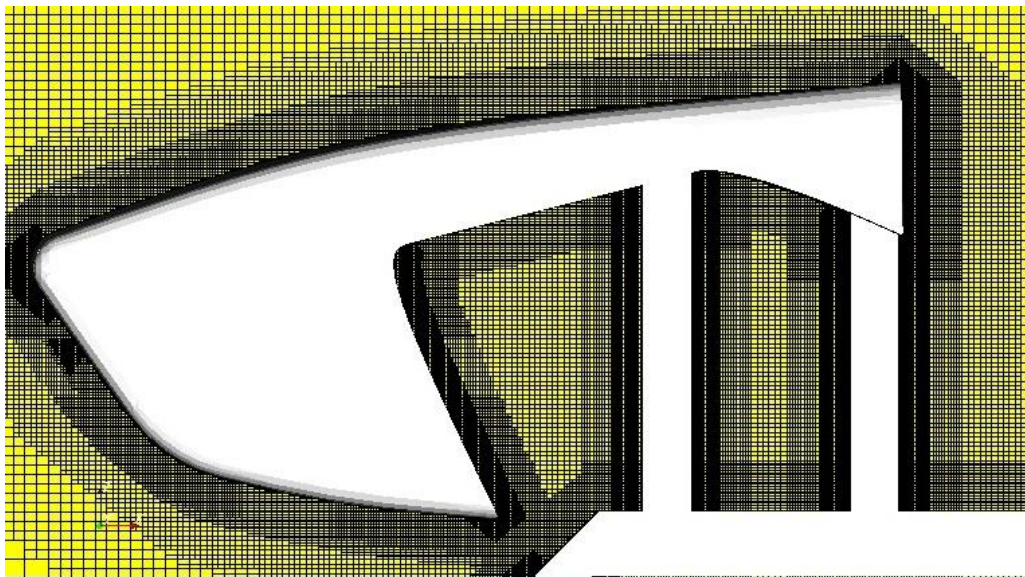
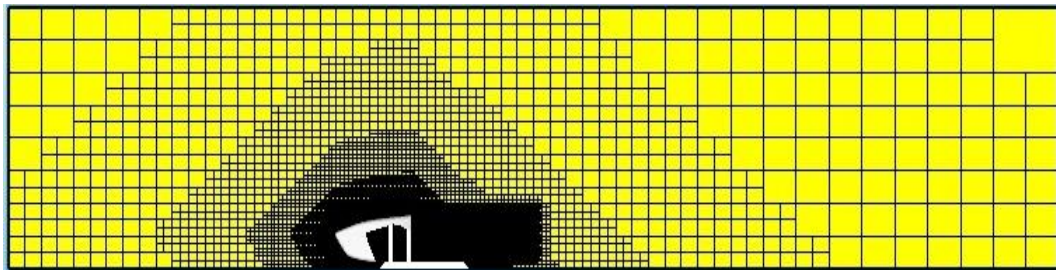
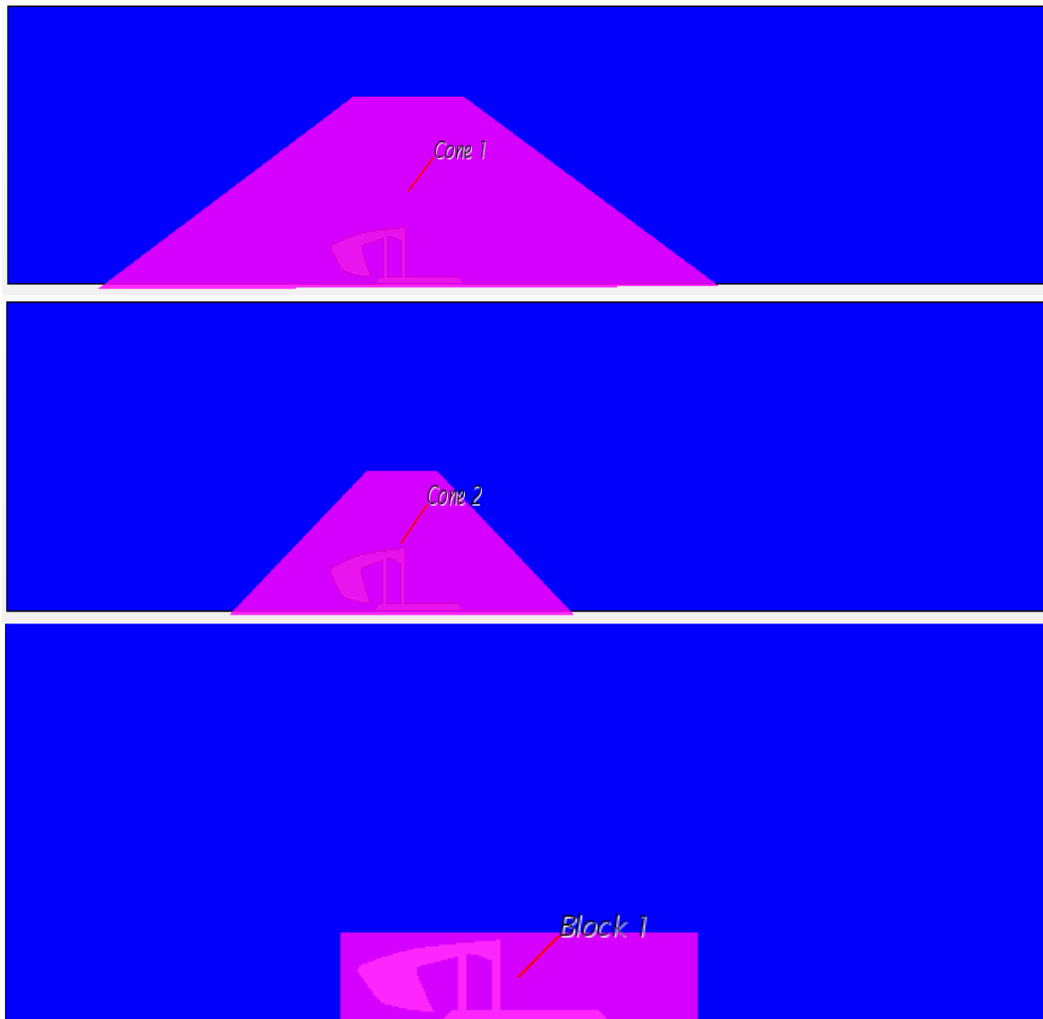


Figura 13 KCFD Mesh2 details

The meshes realized with Star CCM+ had the purpose to reproduce as similar as possible the meshes of KARALIT CFD due to the greater control of this software in grid generation. Using the following Control Volume was possible to

approximately reproduce the intermediate layers, the box window around the body and the wall layer close to the solid body.



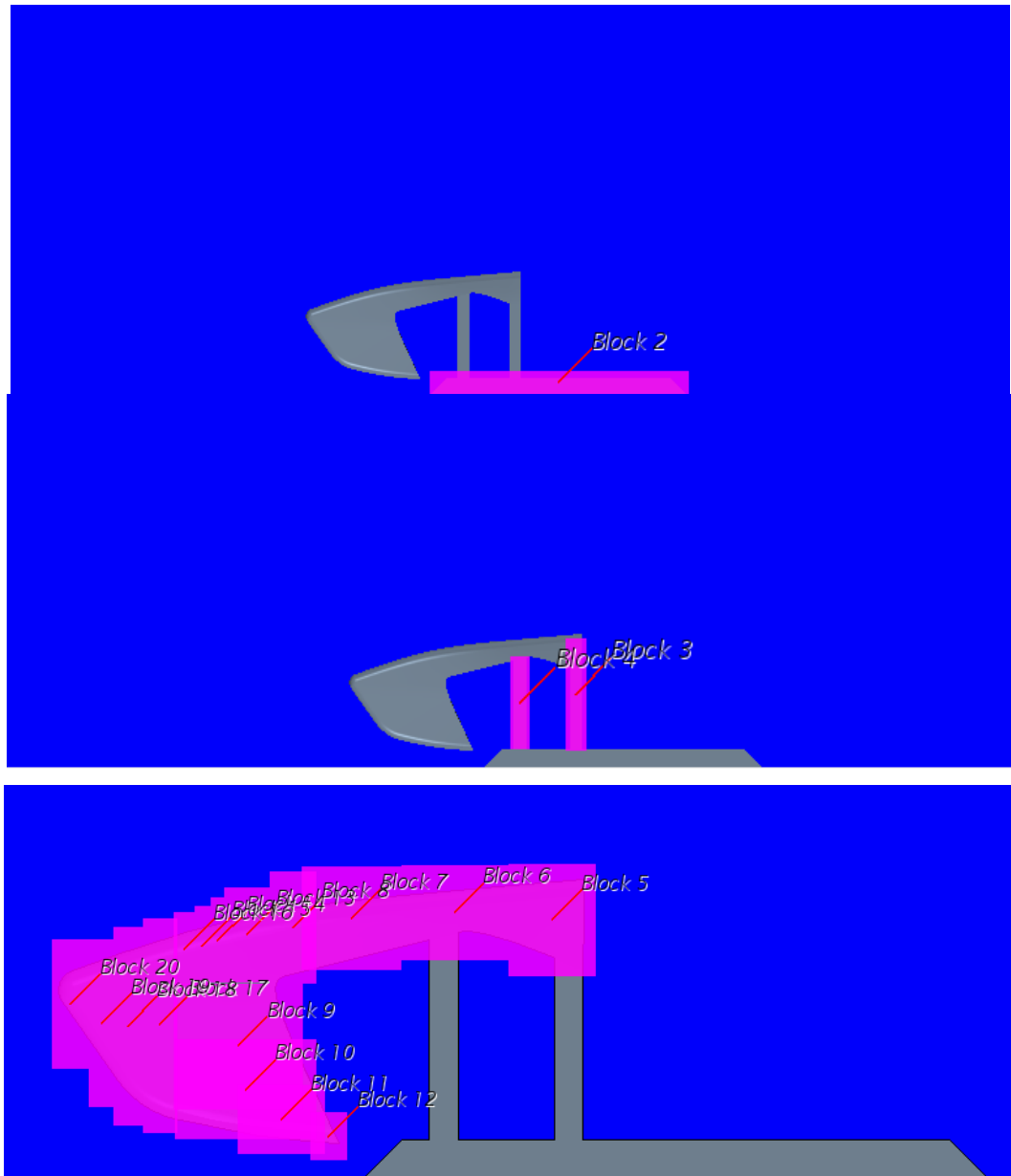


Figura 14 Star CCM+ Control Volumes split

So was possible to create two types of mesh, a coarse one with about 1.8 million of cells and a fine one with about 3.2 million of cells as below.

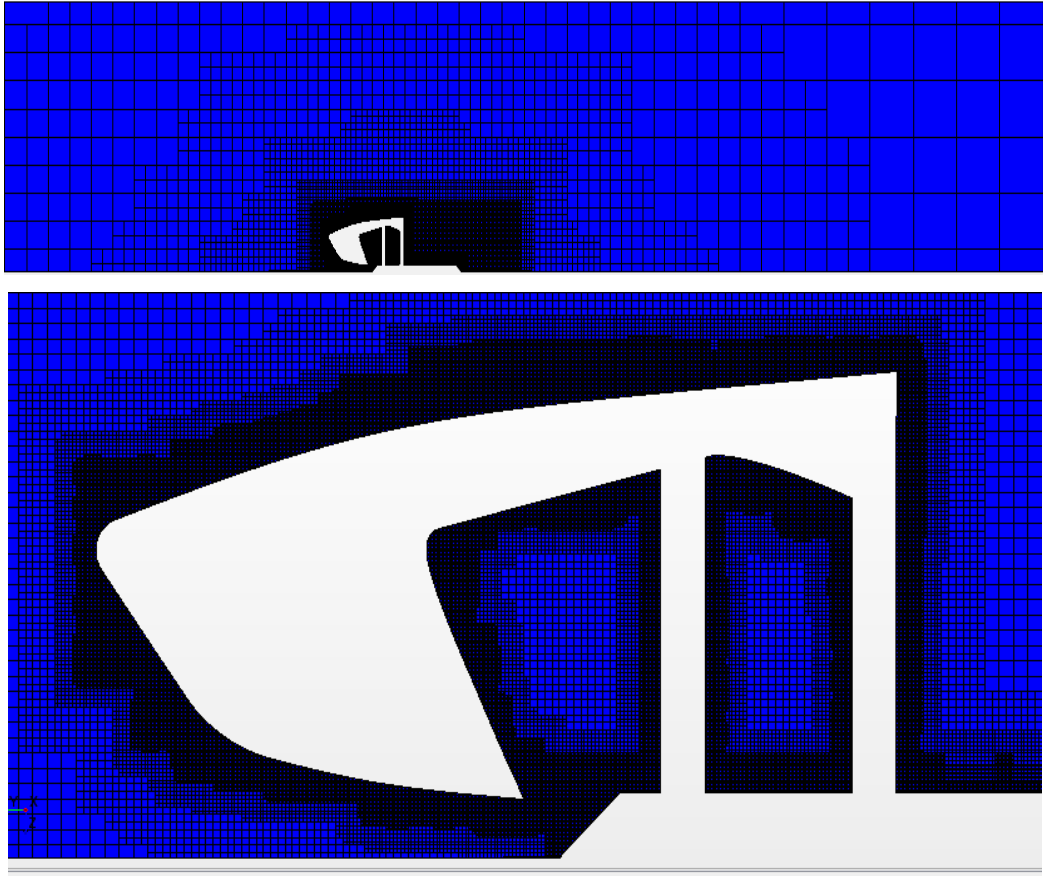


Figura 15 Star CCM+ Mesh1 details



Figura 16 Star CCM+ Mesh2 details

For those simulations the as a computational fluid was chosen air, with 22.2 m/s (~80 km/h) inlet velocity and the following physical conditions:

Tabella 1 Air physique conditions

T	288.15 [K]
ρ	1.225[Kg/ m^3]
μ	$1.8 * 10^{-5}$ [Kg/(m * s)]
P_{amb}	101325[Pa]

The fluid model was defined as:

- Viscous turbulent
- Ideal gas
- Steady
- Coupled flow
- Spalart-Almaras turbulence
- Three dimensional

The Coupled flow model was a forced choice because this is the only model that KARALIT CFD have available. Spalart-Almaras turbulence was chosen as one of the common turbulence models between the two softwares.

Boundary conditions:

- Inlet: *Velocity and Temperature*
- Outlet: *Pressure outlet*
- Lateral walls: *inviscid-adiabatic slip-wall/symmetry*
- Ground: *non slip adiabatic wall*
- Vehicle front part: *non slip adiabatic wall*

5 Results

Once completed the simulations with KARALIT CFD and Star CCM+ the last step was the comparison of:

- Drag force coefficient C_d ;
- Velocity, Pressure etc... distribution;
- Streamlines evolution.

obtained from each software.

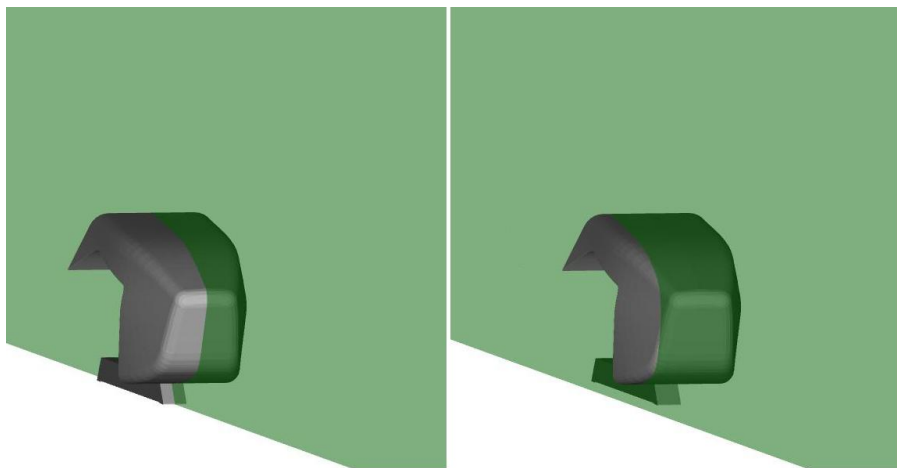


Figura 17 Section planes used in post-processing

Starting with the absolute pressure distribution on the symmetry plane (figure 18) for Mesh1 (~1.8 million cells) can see that there is a great similarity between

KARALIT CFD and Star CCM+. In both cases the frontal part present a compression due to the fluid stagnation of the same entity.

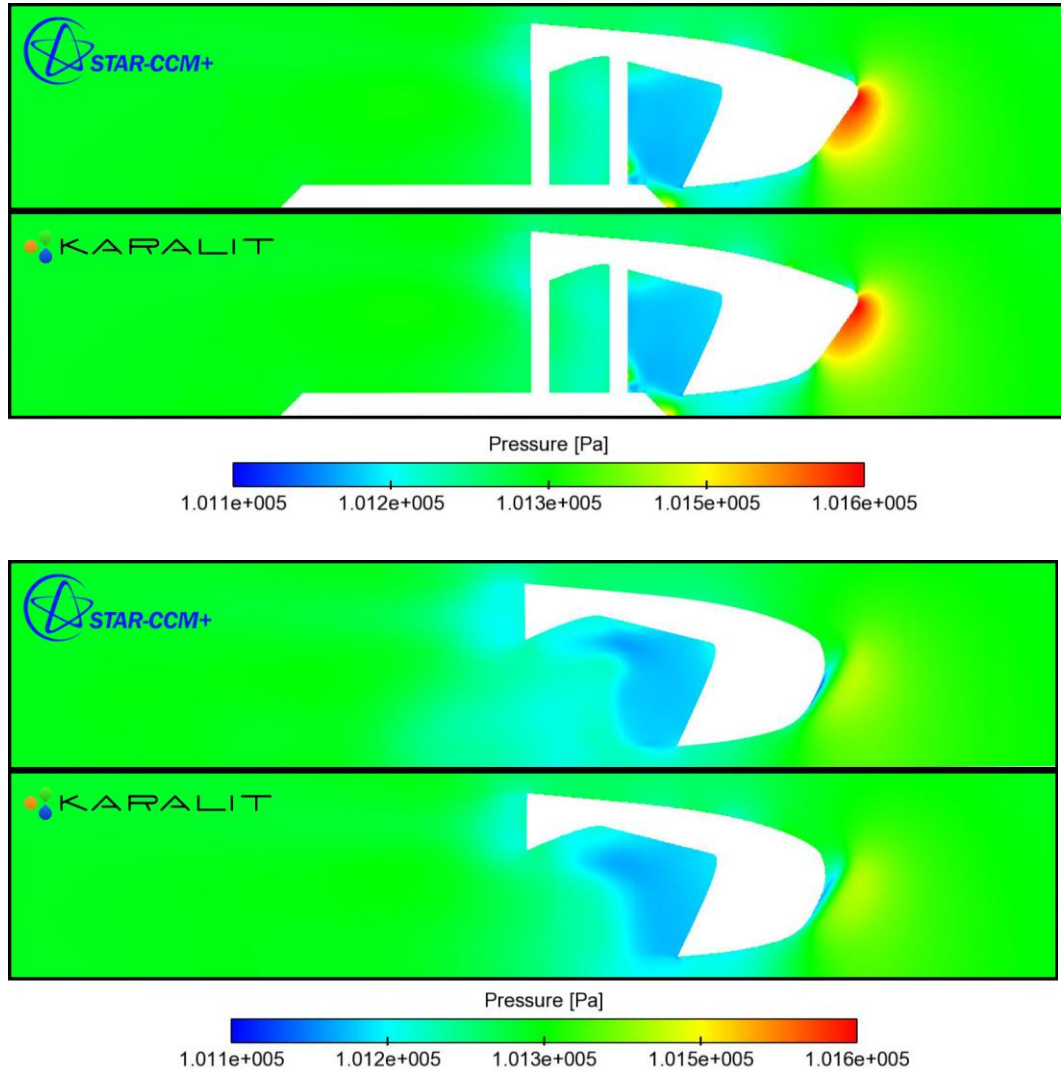


Figura 18 Pressure distribution comparison Mesh1

Figure # shows the velocity field for the two section planes. Both softwares can catch the wake and the deceleration due to the disturbed flow and the dissipative vortex reaction.

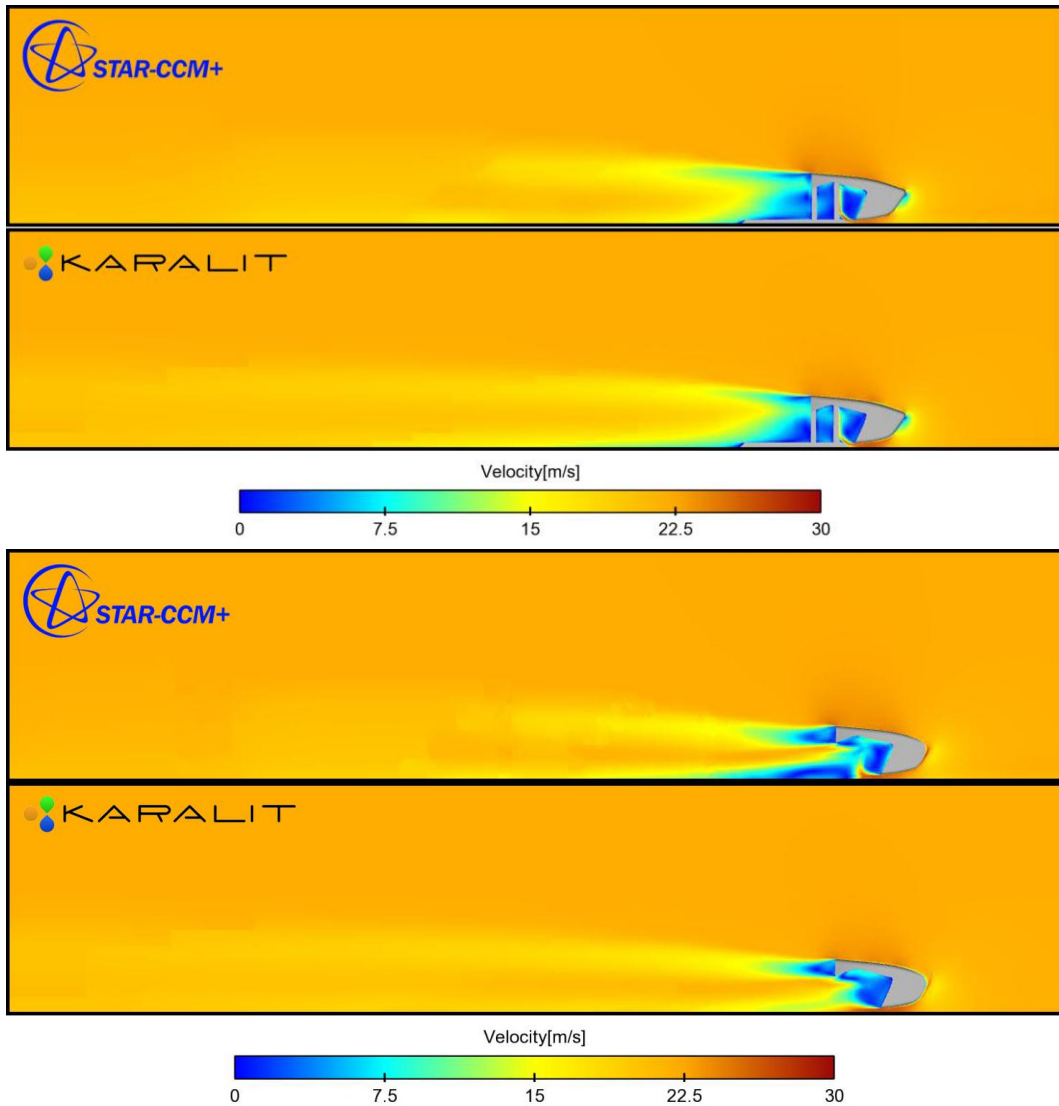


Figura 19 Velocity field comparison Mesh1

What said above can be better observed from the following figures where are shown the streamlines evolution and the velocity vectors field on the symmetry plane.

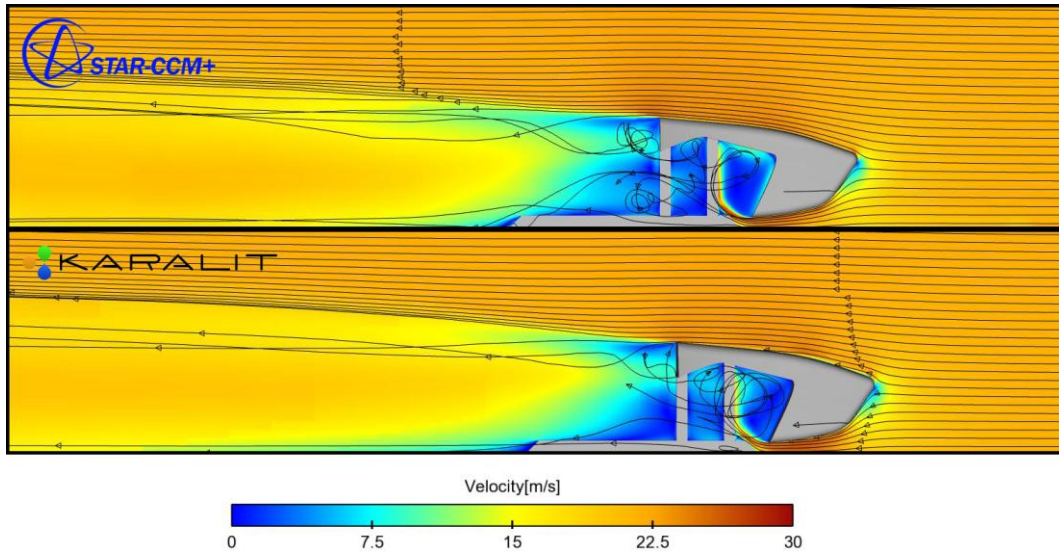


Figura 20 Streamlines evolution Mesh1

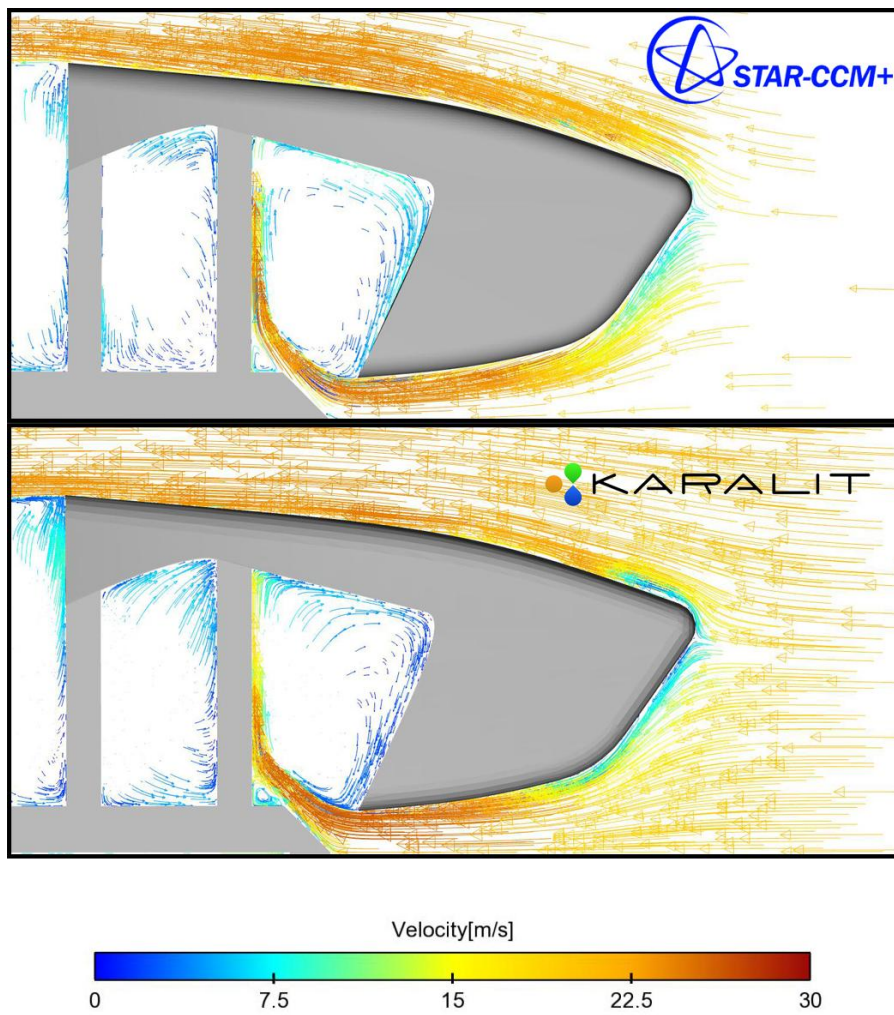


Figura 171 Velocity vectors field Mesh1

Below are presented a series of images that summarize the comparison of the results obtained from Mesh 2 simulation. Following the previous order, first is shown the absolute pressure, the velocity field on the two section planes, the streamlines and the velocity vectors field.

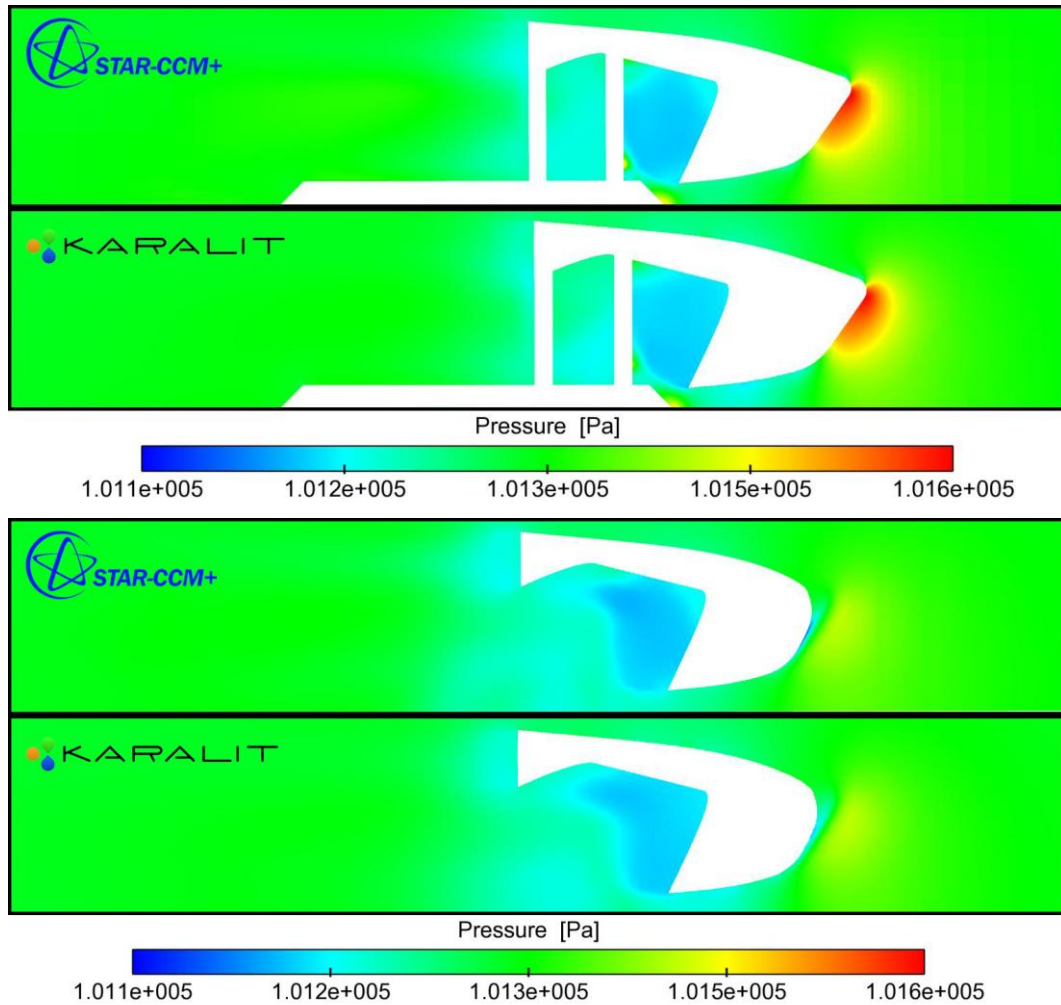


Figura 22 Pressure distribution comparison Mesh2

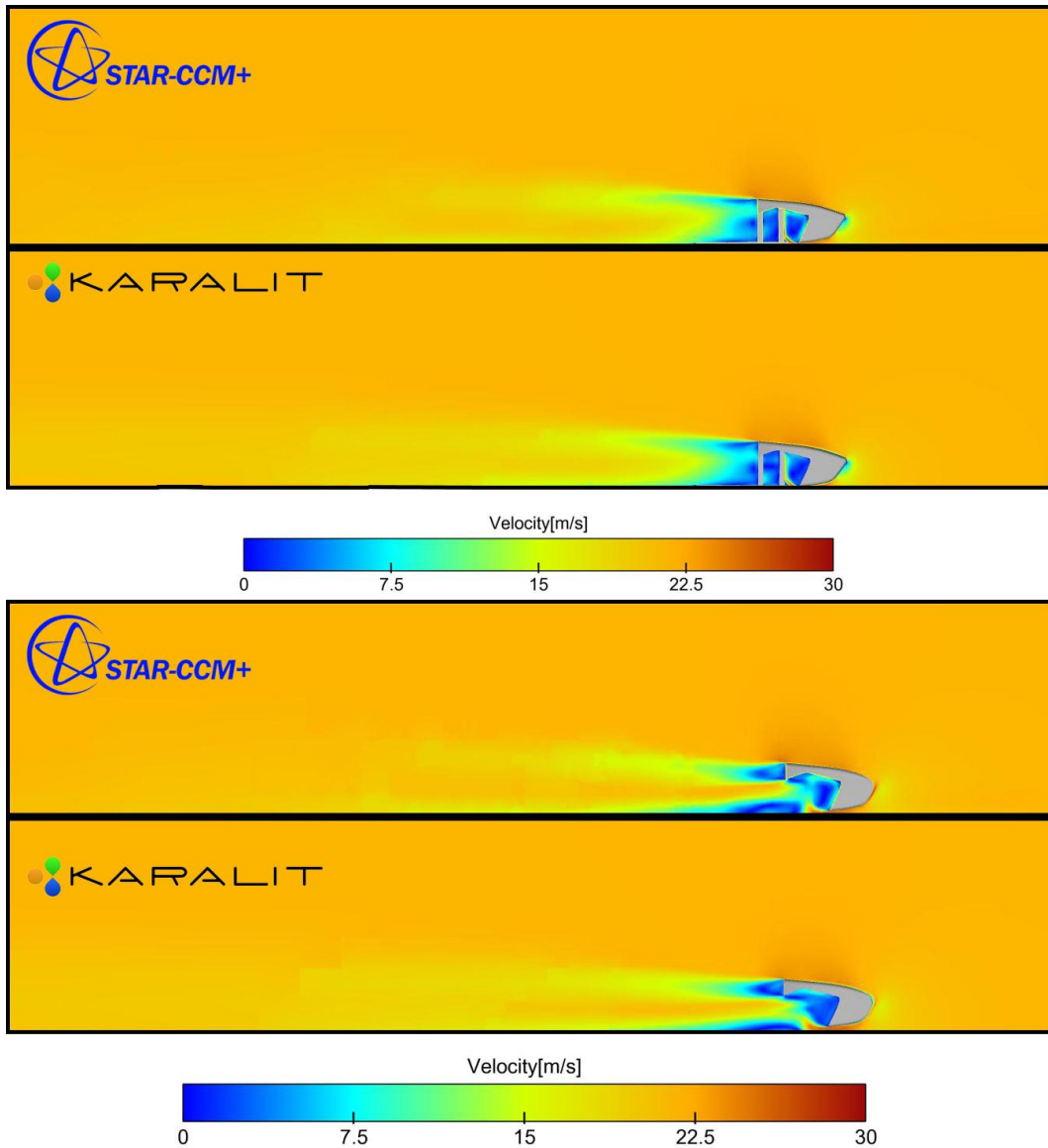


Figura 23 Velocity field comparison Mesh2

Once again the pressure and velocity field turn out to be very similar.

The same thing can be said even for the streamlines (figure 24) which thanks to a greater resolution of the grid can catch better the vortex formation in the wake zone.

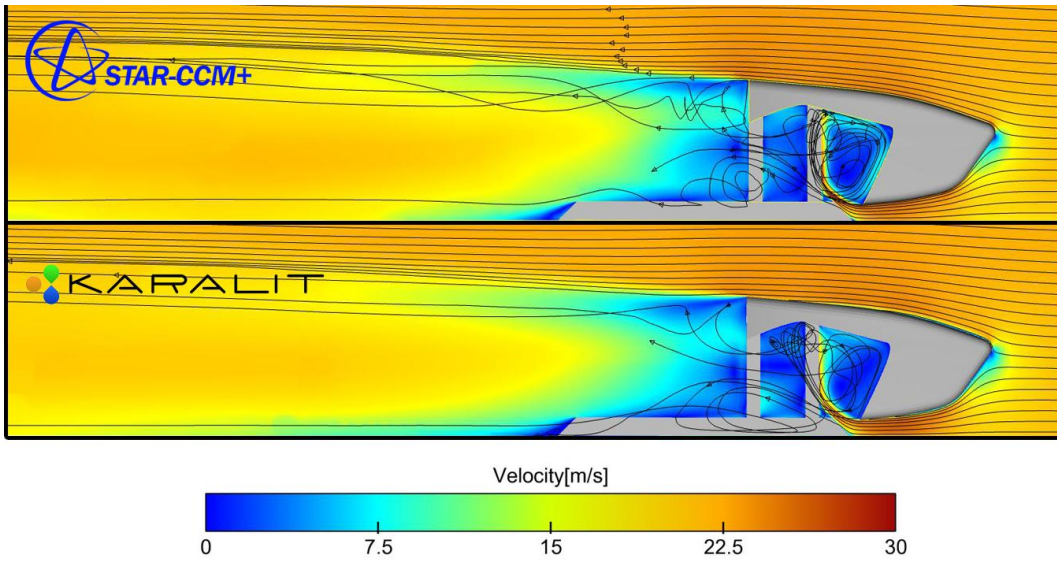


Figura 24 Streamlines evolution Mesh2

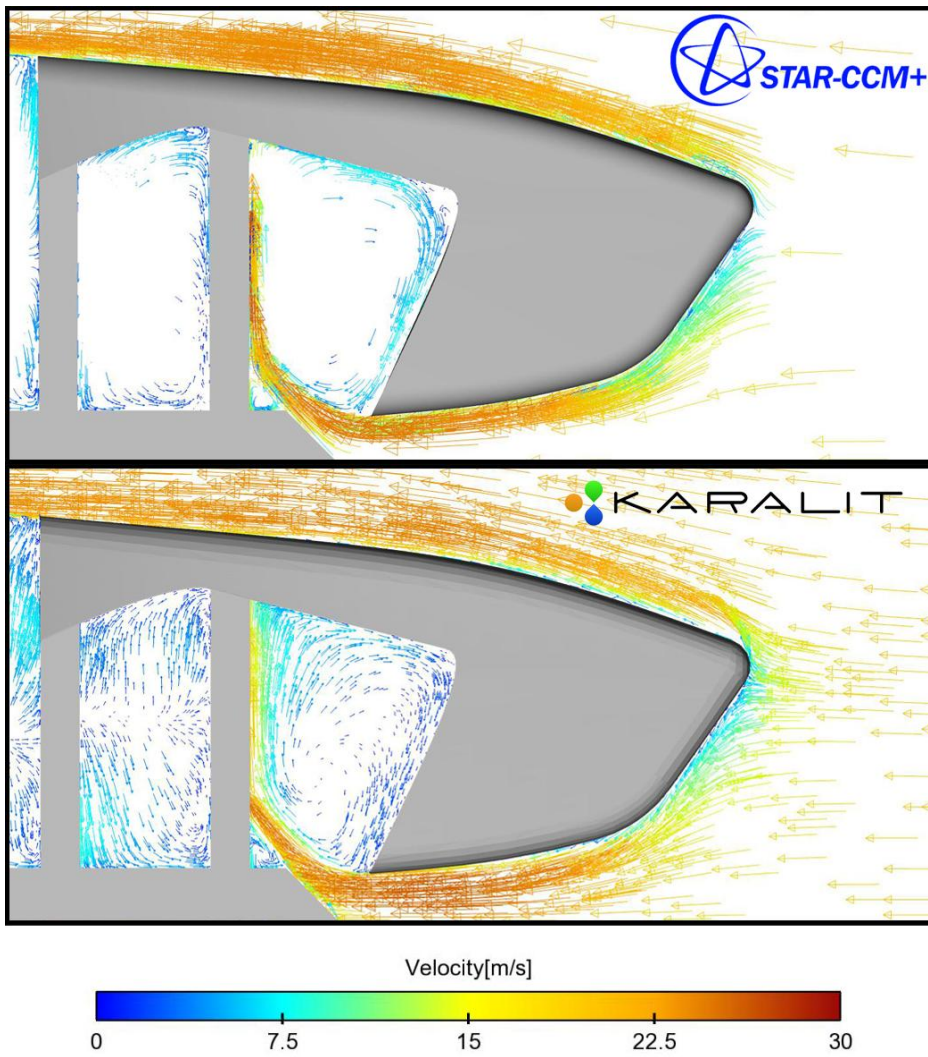


Figura 25 Velocity vectors field Mesh2

At the end, to conclude this comparative analysis, in figure 26 is shown the drag force coefficient calculated from the two softwares for each one of the two meshes, compared with the value obtained from the wind tunnel experimental tests.

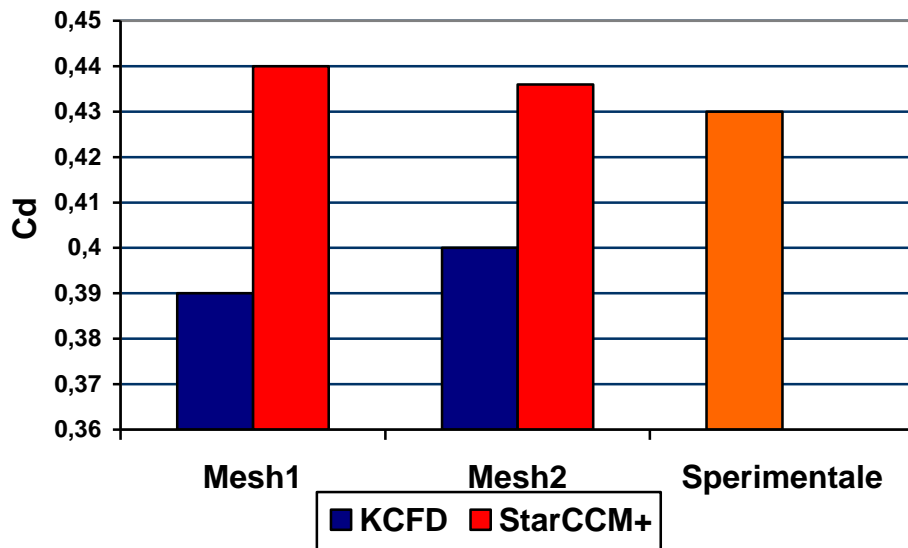


Figura 26 Drag coefficient histogram

The KARALIT CFD results can be considered comparable with both Star CCM+ and wind tunnel tests results. Is very important to notice the drag coefficient C_d trend with the mesh accuracy variation as previously shown.

Conclusions and Future Developments

Using KARALIT CFD during this thesis work brought to results that can be considered comparable with both Star CCM+ simulations and wind tunnel tests experimental results, always showing a substantial accord. This experience definitely consolidate the innovative approach based on the *Immersed Boundary* technology implemented by Karalit code.

The combination of this method with the KARALIT CFD graphical user interface and applications revealed to be a promising approach to reach the primary target: reduce the development time of a complex CFD/3D simulation design. In fact, during the computational grid generation phase, the designer labour is substantially reduced. Sometimes a weak volume discretization control due to the excessive automatism can create a feeling of uncertainty, unlike the already consolidated “*traditional*” methods allow to define a precise and localized mesh even for very complex geometries.

A particular positive vote goes to the graphical user interface (GUI) implemented for all set-up phases; its advanced architecture enables a simple and intuitive development thanks even to a detailed *User’s Guide* and focused tutorials; on the other hand a good knowledge of an external software such as Tecplot, Ensignt (CEI) or Paraview is needed to fill the integrated GUI limited capacity in post-processing for the moment.

During this thesis work has been noticed a non-optimal use of RAM by KARALIT CFD; the two codes computational times were comparable only after its expansion from 16 to 32 GB; KARALIT.

References

- [1] **KARALIT CFD v2.0**, Extended User Guide 2013 Karalit;
- [2] **R. Mittal, G. Iaccarino** – “Immersed Boundary Methods”, Annual Review of Fluid Mechanics, 2005;
- [3] **D. Goldstein, R. Handler, and L. Sirovich**, – “Modeling no-slip flow boundary with an external force field”, Journal of Computational Physics. 1993
- [4] **E.M. Saiki and S. Biringen** , – “Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method ”, Journal of Computational Physics. 1996
- [5] **J. Mohd-Yusof**, – “Combined Immersed Boundaries/B-Splines Methods for Simulations of Flows in Complex Geometries ”, CTR Annual Research Briefs, NASA Ames/Stanford University. 1997
- [6] **D.De Zeeuw and K.G. Powell**, – “An adaptive Cartesian mesh method for the Euler equations ”, Journal of Computational Physics 1993
- [7] **R.J. Pember, B.J. Bell, P. Colella, W.J. Crutchfield, and M.L. Welcome**, – “An adaptive Cartesian grid method for unsteady compressible flow in irregular regions”, Journal of Computational Physics 1995
- [8] **E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof**, “Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations ”, Journal of Computational Physics, 2000
- [9] **Y.H. Tseng, J.H. Ferziger** – “A ghost-cell immersed boundary method for flow in complex geometry”, Journal of Computational Physics, 2003