Technische Universität München

Fakultät für Maschninenwesen

Bachelor's Thesis in Mechanical Engineering

Detached-Eddy Simulation of a Wall-Mounted Cylinder Flow

Alejandro Contel

Technische Universität München

Fakultät für Maschninenwesen

Bachelor's Thesis in Mechanical Engineering

Detached-Eddy Simulation of a Wall-Mounted Cylinder Flow

Author:	Alejandro Contel
Supervisor:	Marco Kiewat
Advisor:	Dr.-Ing. habil. Thomas Indinger
Submission:	20.09.2016

I assure the single-handed composition of this bachelor's thesis only supported by declared resources.

München, 20.09.2016

(Alejandro Contel)

Abstract

The aim of this project is to study the effect of different turbulence models on the detachment point on the cylinder surface and the vortex shedding frequencies. The turbulence models considered are Spallart-Allmaras DDES, kw-SST DDES, Spallart-Allmaras IDDES and kw-SST IDDES, being the two last ones an improved formulation of the currently used DES model.

In order to perform this study, OpenFOAM, the CFD Open Source tool has been used. To do the meshing, the snappyHexMesh tool is the one considered. Since a suitable computational grid needs to be built, an in-depth study on the meshing configuration has been also held and the grid convergence is shown for certain configurations.

The simulation results cover several magnitudes, such as the drag coefficient, the detachment point and the frequencies of oscillations in the wake of the cylinder as well as on its surface. To validate those results, a literature research prior to running the simulations has been performed, so that they can be easily compared and proved.

Table of contents:

Table of contents

Table of figures:

Index of tables:

Nomenclature and abbreviations

Nomenclature

a	Semi-major axis of the ellipse
α	Detachment point angle
α_{ν_t}	Spalart-Allmaras coefficient
b	Semi-minor axis of the ellipse
C_{b1}	Spalart-Allmaras coefficient
C_{b2}	Spalart-Allmaras coefficient
C_D	Drag coefficient
C_L	Lift coefficient
C_S	Side coefficient
C_k	Spalart-Allmaras coefficient
$C_{\nu 1}$	Spalart-Allmaras coefficient
$C_{\nu 2}$	Spalart-Allmaras coefficient
C_x	x-direction Drag coefficient
$\overline{C_x}$	Time-averaged x-direction Drag coefficient
C_{w2}	Spalart-Allmaras coefficient
C_{w3}	Spalart-Allmaras coefficient
C_{DES}	DES coefficient
C_μ	Empirical turbulence constant
D	Cylinder's diameter
ε, ε_{in}	Turbulence dissipation
f	Frequency
k, k_{in}	Turbulence kinetic energy
l	Length of the plate in the x-direction
l	Turbulence length scale
L	Lift
R	Cylinder's radius
Re	Reynolds number
Pr	Prandtl number
ρ	Density
S	Surface
St	Strouhal number
t	Thickness of the plate
t	Time
T	Period
Tu	Grade of the turbulence
u	x component of the velocity

U	Velocity
$\overline{U}$	Mean velocity
U_m	Average outflow velocity
U_∞	Upstream velocity
v	y component of the velocity
V	Volume
ν	Kinematic viscosity
v_t	Turbulent kinematic viscosity
w	z component of the velocity
ω	Specific dissipation
x, y, z	Cartesian coordinates
x_d	Separation from the center of the cylinder to the trip
Δ	Delta coefficient
μ	Dynamic viscosity
Φ	Phase angle

Abbreviations

3D	3-Dimensional
cDNS	Coarse Direct Numerical Simulation
CFD	Computational Fluid Dynamics
DES	Detached Eddy Simulation
DDES	Delayed Detached Eddy Simulation
DFT	Discrete Fourier Transform
DNS	Direct Numerical Simulation
FFT	Fast Fourier Transform
FVM	Finite Volume Method
GIS	Grid Induced Separation
IDDES	Improved Delayed Detached Eddy Simulation
LES	Large Eddy Simulation
MSD	Modeled Stress Depletion
RANS	Reynolds-Averaged Navier-Stokes
RSM	Reynolds Stress Models
SA	Spalart-Allmaras
SGS	Sub-grid scale
SST	Shear Stress Transport
STL	STereo Lithography
TL	Trip-less
U-RANS	Unsteady Reynolds-Averaged Navier-Stokes
WMLES	Wall-Modeled Large Eddy Simulation

1. Introduction

The usage of unsteady CFD simulation in industrial applications is usually limited to unsteady RANS or hybrid RANS-LES methods like Detached-Eddy Simulation (DES). This project is based on the DES of the flow around a wall-mounted cylinder using OpenFOAM. The goal is, therefore, to study the effect of different turbulence models on the detachment point on the cylinder surface and the vortex shedding frequencies. The turbulence models considered are Spallart-Allmaras DDES, kw-SST DDES, Spallart-Allmaras IDDES and kw-SST IDDES, being the kwSST ones a more recent formulation of the previous models. To do the meshing, the snappyHexMesh tool will be used. Since a suitable computational grid needs to be built, an in-depth study on the meshing configuration has been held and the grid convergence is shown for certain configurations regarding several parameters of study, which can be seen on the results section of this document.

The project is split up in several parts. First of all, a literature study is considered, in order to provide a theoretical background that helps to understand the steps that will be performed when running the simulation and the results that are to be obtained. This study covers from turbulence theory to comparisons between the models that are used and that are expected to be right on the conclusions section. Later on, the simulation procedure is described, from the meshing step until when the results are obtained. This section includes a compilation of the most important parameters and concepts to be taken into account before running the simulations. The most important section is the results one. In this section of the document, all the comparisons and studies can be found. It includes a brief comment on the results, which is later on detailed in the conclusions section.

Finally, on the appendix section, the reader will find some of the relevant files, dictionaries and additional aspects to be taken into account that have been used to run the simulations. They have been separated to avoid excessive information, and therefore, if the reader wants to have a detailed idea of how the simulations have been run, they can always be checked on this section. A first sketch of the cylinder and the plate can be seen at Figure 1, representing just the main measures.

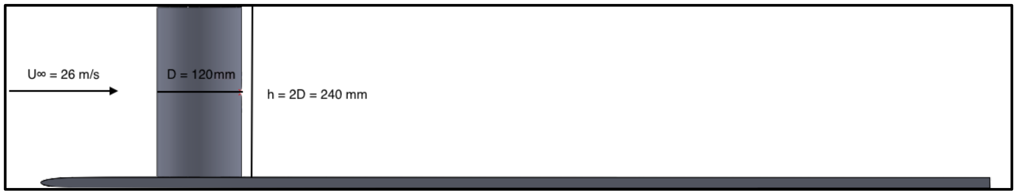


Figure 1 - First sketch with the main measures

2. Theoretical background

2.1. Turbulence

Turbulence can be defined as flow regime characterized by chaotic property changes and behaviors. Several property changes have a significant relevance, which is the case of low momentum diffusion, high momentum convection, and rapid variation of pressure and flow velocity in both space and time. A disorderly motion (deterministic chaos) is expected for the flow, and the origin is instability in the laminar flow.

A strict relationship between turbulence and a non-dimensional variable or factor does not exist. However, the non-dimensional Reynolds number (Re) is related to turbulence. The Reynolds number can be defined as $Re = \frac{UD}{\upsilon}$, being U the maximum velocity of the object relative to the fluid, D the diameter of the cylinder and υ the kinematic viscosity. Even though several environment conditions are involved and every case is different, an approximation can be made stating that flows at Reynolds numbers larger than 5000 tend to be turbulent, while those at low Reynolds numbers usually remain laminar. Therefore, lower Reynolds numbers correspond to laminar flows whereas higher numbers correspond to turbulent ones. In a turbulent flow, differently scaled unsteady vortices are present and interact with each other. Almost all fluid flows that we encounter in daily life are turbulent.

They are, as stated, irregular and chaotic. The flow consists of a spectrum of different scales (eddy sizes). In them, the diffusivity also increases. The exchange of momentum is increased in turbulence at the boundary layers, and the separation is delayed on surfaces such as airfoils. Therefore, the heat transfer is also increased, as well as the wall friction. Turbulent flow is dissipative, which means that kinetic energy in eddies is transformed into thermal energy (from bigger to smaller ones in a hierarchical way).

In short, the factors related to turbulent flows are high-Reynolds numbers, adverse pressure gradients, wall roughness and disturbance in outer flow and possible secondary flows as well. The consequences are a 3D non-stationary flow, efficient mixing, homogenization of properties and increased wall friction.

Furthermore, the velocities in turbulent flows tend to be divided in two parts, one being the time-averaged one (time independent) and another part, which is defined as the fluctuating part. Therefore, the equality $u = \bar{u} + u'$ can be stated, being the first component the time-averaged one and the second one the fluctuating one.

2.2. Origin and structure of the turbulence

Laminar flows reach a critical point in both space and time at which they become unstable under small perturbations. Turbulence arises from instabilities at high Reynolds numbers, causing the properties of the flow to behave and vary in an irregular manner. These perturbations trigger the transition between laminar and turbulent flow.

Turbulence is highly dependent on its surroundings to maintain itself by obtaining energy. The most common source of energy is shear in the mean flow. Therefore, in a non-favorable environment regarding the desired turbulence progression, a turbulent flow would decay and return to a laminar state again, since it would not be able to maintain the turbulent properties for the reasons described. Turbulence is not an instantaneous process, and there are several stages that describe the transition process. Figure 2 illustrates in a clear way in which way this process is developed.

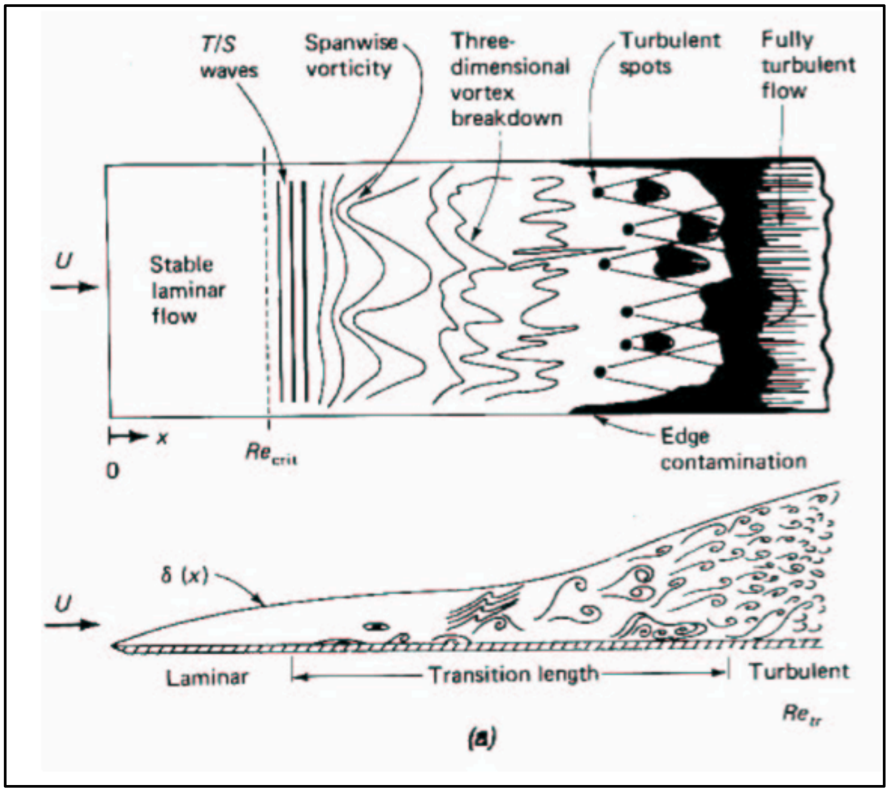


Figure 2 - Origin and development of turbulence

The flow is laminar at first. Following the flow path, the local Reynolds number will increase, and therefore turbulent transition will be observed. When the Reynolds number reaches a critical value, the Tollminen-Schlichting waves, which can be defined as perturbations in perpendicular direction of the flow's motion, appear. However, they are not turbulent but still laminar.

Following the flow motion's direction, the loss of stability on these waves can be observed. A vorticity component in the flow's direction will appear as well. The next phenomenon that can be observed is the chaotic, irregular and most importantly three-dimensional flow pattern. Turbulence is three-dimensional, as it has been previously described. The growth of this layer is observed as the Reynolds number keeps growing and in the end, the completely developed turbulence will be formed, ending the transition process. In short, the typical transition scenario can be described stepwise as laminar flow stage over a short distance, instability due to small perturbations (Tollmien-Schlichting waves formation), linear growth of waves, non-linear saturation and finally, developed turbulence.

The factors influencing transition are the Reynolds number, external perturbations (Pre-turbulence levels and wall roughness), adverse pressure gradients and transverse flow, whereas the wall-related effects can be suction, cooling (heat transfer), curvature (by means of Görtler vortices - secondary flows that appear in a boundary layer flow along a concave wall -), rugosity and compressibility.

Regarding the structure of a turbulent layer, the turbulent flow near a flat wall can be divided up into four regimes, which are called viscous sub-layer, buffer layer, log-law region and free-stream region respectively, if we separate from the wall.

At the wall, the fluid velocity is zero, and for a thin layer above this, the flow velocity is linear with distance from the wall. The region that is therefore defined is defined as the viscous sub-layer. The next known layer is defined as the buffer layer, which is located further away from the wall. In this region, the flow begins the transition to turbulent flow and it eventually reaches fully turbulent flow region. The average flow velocity in this region is related to the log of the distance to the wall, and it is defined as the log-law region. Even further away from the wall, the flow transitions to the free-stream region. The different layers can be observed in Figure 3.

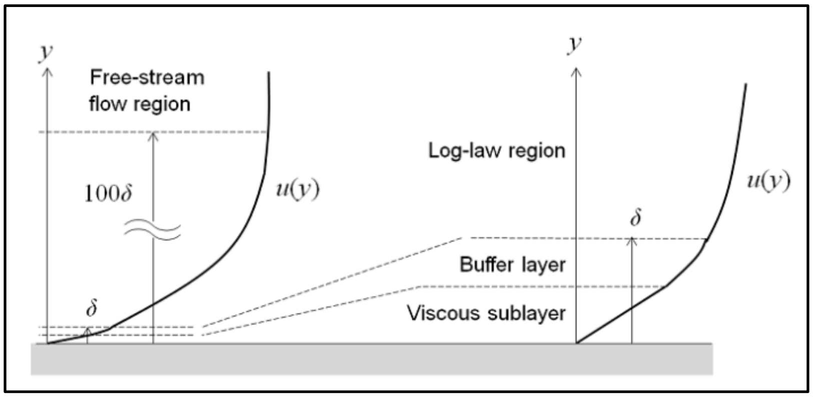


Figure 3 - Turbulence layers

2.2.1. Law of the wall

The law of the wall is a law applicable to the part of the flow that is close enough to the wall or boundary and states that the average velocity of a turbulent flow at a certain point is proportional to the logarithm of the distance from that point to the wall or boundary of the fluid region.

2 different dimensionless magnitudes play a key role regarding the law mentioned, which are the dimensionless velocity (u^+) and the dimensionless wall distance (y^+). The dimensionless velocity can be defined as $u^+ = \frac{U}{u_*}$, being U the local velocity and u_* the friction velocity at the closest wall, whereas the dimensionless wall distance is defined as $u^+ = \frac{u_* * y}{\upsilon}$, where u_* is the friction velocity at the nearest wall, y is the distance to the nearest wall and υ is the local kinematic viscosity of the fluid.

In the log layer the velocity profile can be estimated with the log law, which states that $u^+ = \frac{1}{\kappa}\ln(y^+) + B = \frac{1}{0.41}\ln(y^+) + 5.1$, where κ is the von Karman's constant and can be approximated to 0.41 and B is a constant that can be approximated to 5.1. Close to the wall in the viscous sublayer, both dimensionless magnitudes, u^+ and y^+, are supposed to be equal to each other, being $u^+ = y^+$ for this region. This behavior, as well as the one at the logarithmic region, can be observed at Figure 4, as well as the approximation made by the law described above.

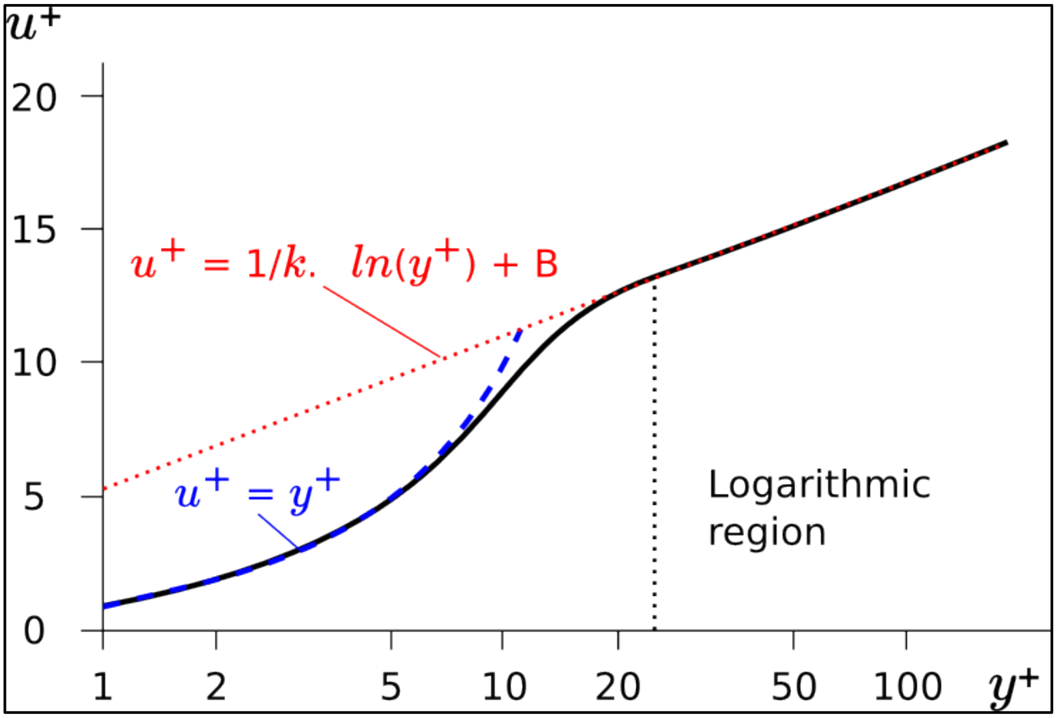


Figure 4 - Law of the wall

2.2.2. Strouhal number

The Strouhal number (St) can be defined as a dimensionless number that describes flow mechanisms that oscillate. It can be defined for this particular study as:

$$St = \frac{fD}{U}$$

Where f is the frequency of the vortex shedding, D is the diameter of the cylinder and U the inlet velocity of the case. For large Strouhal numbers, viscosity dominates fluid flow, whereas for low Strouhal numbers, the high-speed dominates the oscillation.

The two first Strouhal numbers will be considered for this project by considering the FFT spectrum, as it will be shown in the result section.

2.3. Governing equations and laws

2.3.1. Mass conservation

Regardless of the flow assumptions, a statement of the conservation of mass is generally necessary. This is achieved through the mass continuity equation, given as:

$$\frac{\partial \rho}{\partial t} + \rho \frac{\partial}{\partial x_1}(u_1) = 0$$

Breaking the density and velocity into both mean and fluctuating components the following expression can be obtained:

$$\rho = \bar{\rho} + \rho' \qquad u_i = \overline{u_i} + u_i'$$

$$(\bar{\rho} + \rho')\frac{\partial}{\partial x_1}[(\overline{u_i} + u_i)] = 0$$

And if the flow were stationary and incompressible the expression above would simplify in the following one:

$$\frac{\partial}{\partial x}(\overline{u_1} + u_1') + \frac{\partial}{\partial y}(\overline{u_2} + u_2') + \frac{\partial}{\partial z}(\overline{u_3} + u_3') = 0$$

And by temporal-averaging the final expression would be:

$$\frac{\partial \overline{u_1}}{\partial x_1}+\frac{\partial \overline{u_2}}{\partial x_2}+\frac{\partial \overline{u_3}}{\partial x_3}=0$$

And therefore:

$$\frac{\partial \bar{u}_i}{\partial x_i}=0$$

2.3.2. Momentum conservation

The momentum equation is a vector equation that represents the application of Newton’s 2nd Law of Motion to the motion of a gas, and relates the time rate of change of the gas momentum to the forces that act on the gas.

First of all, the incompressible version of the Navier-Stokes Equation needs to be defined, and states that:

$$\frac{\partial u_i}{\partial t}+u_j\frac{\partial u_i}{\partial x_j}=f_i-\frac{1}{\rho}\frac{\partial p}{\partial x_i}+\nu\frac{\partial^2 u_i}{\partial x_j \partial x_j}$$

Where f_i is a vector representing external forces.

The momentum equation can be defined after time-averaging the Navier-Stokes Equation in the following way:

$$\frac{\partial \bar{u}_i}{\partial t}+u_j\frac{\partial \bar{u}_i}{\partial x_j}=\bar{f}_i-\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i}+\nu\frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j}-\frac{\partial \overline{u_i' u_j'}}{\partial x_j}$$

Where the last term is the Reynolds stress tensor.

It is also important to remark that the time-average of a fluctuating quantity is 0, but also that the product of two fluctuating quantities, as it can be seen at the numerator of the last term, is not necessarily equal to 0.

2.4. Separation

All solid objects traveling through a fluid acquire a boundary layer of fluid around them where viscous forces occur in the layer of fluid close to the solid surface [1]. This can be the case of wings from aircrafts, Formula 1 ailerons, parts of a car in motion or just a static cylinder facing the flow's direction, which is the case of study in this project. As it has been defined in previous sections, boundary layers can be either laminar or turbulent, and the properties of them diverge reasonably.

A known phenomenon that could happen is separation. It can occur when the boundary layer faces an adverse pressure gradient enough (which can cause flow reversal) so that the speed of the boundary layer relative to the object is reduced to zero. Then, the fluid flow will detach from the surface of the object, generating vortices or eddies. A clear illustration of these phenomena can be observed in Figure 5.

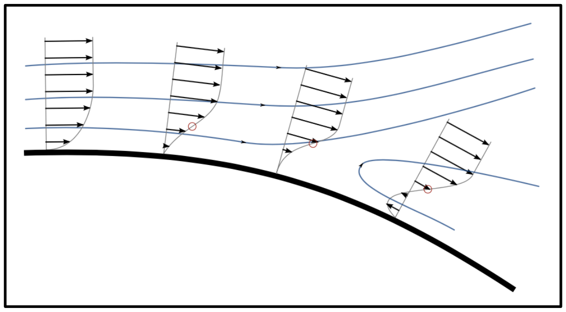


Figure 5 - Detachment process

Depending on whether the boundary layer is turbulent or laminar, the effects and consequences are different. Regarding a laminar boundary layer, low friction can be expected. However, the momentum will be little, and therefore it cannot resist strong adverse pressure gradients, producing an early separation, and therefore, a high form drag. Regarding a turbulent boundary layer, higher friction is expected, but increased momentum due to mixing as well, which offers a better resistance to adverse pressure gradients and thus, a delayed separation, producing a smaller form drag.

Since flow separation can result in increased drag, research in the industry is being conducted in order to study the consequences and how to avoid it. For example, an early flow separation on an airfoil (closer to the leading edge rather than to the trailing edge) due to a higher-than-recommended angle of attack can result in a drag increase and therefore, cause stall (loss of lift), producing severe effects that are not desired if a safe flight wants to be carried out. In this project, the detachment point of the flow facing the cylinder will be studied in several ways by conducting a set of simulations with different parameters, studying how the different magnitudes and configurations influence this detachment and the consequences that this effect might cause on the domain.

2.5. Turbulence Models

The objective of turbulence models is to obtain a physical description of the flow by means of solving a set of equations. A model, therefore, must be built to predict the effects of turbulence. The Navier–Stokes equations govern the velocity and pressure of a fluid flow. Regarding a turbulent flow, the quantities defined can be split up or decomposed into two parts, a mean part and a fluctuating one, as it has been already described. To begin with, an overview to the Navier-Stokes equations must be made. They can be defined as equations that describe the motion of viscous fluid substances. The equalities in the equations come from the fact of applying Newton's second law to the fluid motion, as well as assuming that the stress in the fluid comes from the sum of a diffusing viscous term (proportional to the gradient of velocity) and a pressure term. They can be used to model several relevant processes, such as the weather and air flow around a wing. In this section, an overview of the known turbulence models will be made, by means of describing the main characteristics and features, which will help to understand why certain models will be used in this project.

2.5.1. RANS-based models

Taking into consideration the description above, the Reynolds-averaged Navier–Stokes based models (RANS-based models) are models based on the RANS equations, which can be defined as time-averaged equations of motion for fluid flow. The idea behind the equations is Reynolds decomposition, meaning that an instantaneous quantity is decomposed into its time-averaged and fluctuating quantities. Some approximations can be considered regarding the equations described, depending on the properties of the flow known, in order to give time-averaged solutions to the Navier-Stokes equations. The computation of the Reynolds stresses is the main objective of these turbulence models, and that can be achieved by three main categories of RANS-based turbulence models, which are linear eddy viscosity models, nonlinear eddy viscosity models and Reynolds stress model.

2.5.2. Algebraic turbulence models

Algebraic turbulence models are also known as zero-equation turbulence models. The solution of additional equations is not needed by this kind of models, since they are computed directly from the variables of the flow. However, history effects on the turbulence would not be taken into account regarding 0 equation models, since neither the history effects on turbulence nor convection and diffusion would be considered, making them valid enough for certain situations (simple geometries, for example) but not suitable for further simulations. The two most well known zero equation models are the Baldwin-Lomax model and the Cebeci-Smith model.

2.5.3. One-equation turbulence models

One-equation turbulence models offer a wider variety of options. They are intended to solve one turbulent transport equation, usually the turbulent kinetic energy. A common one-equation model is the Spalart-Allmaras model. Its use is normally focused in the aerospace or automotive industry, showing to provide good results for boundary layers that face significant adverse pressure gradients. It solves a modeled transport equation for the kinematic eddy turbulent viscosity. It adds a single additional variable for a Spalart-Allmaras viscosity and does not use any wall functions, but solves the entire flow field. The turbulent viscosity is calculated from the Boussinesq assumption, which states that the momentum transfer caused by turbulent eddies can be modeled with an eddy viscosity, and is defined as:

$$-\rho\overline{u_i'u_j'} = \mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\rho k\delta_{ij}$$

However, several studies have demonstrated that this model does not perform very well regarding shear flow, separated or decaying turbulence fields. However, it has several advantages as well, which are a reasonable stability and convergence.

2.5.4. Two-equation turbulence models

Finally we have the two-equation models, which are one of the most common types of turbulence models, obviously for several reasons. The most common ones are the k-epsilon and the k-omega models.

2.5.4.1. k-epsilon and k-omega

Two-equation models include two extra transport equations to represent the turbulent properties of the flow, allowing the model to account for history effects like convection and diffusion of turbulent energy, improving the performance with respect to previous less complex models [2]. The most common transport variable is the turbulent kinetic energy, being the second variable the one that is normally different depending on the model chosen. The most common second variables are either the turbulent dissipation or the specific dissipation. In short, the first variable is known to determine the energy of the turbulence, whereas the second one determines its scale. The basis for the two-equation models is the Boussinesq eddy viscosity assumption, which describes that the Reynolds stress tensor is proportional to the mean strain rate tensor. The kinematic turbulent viscosity can be approximated taking the log-law as a reference. The expression that relates the required terms needed to approximate the turbulent viscosity is the following one:

$$\nu_t = \nu(\frac{y^+\kappa}{\ln Ey^+} - 1)$$

Where E=9.8 is equivalent to additive constants in the logarithmic law of the wall and the other magnitudes have already been defined. The k-epsilon model is used in several simulations and has a significant importance, although it is known to give not accurate enough results in cases of strong flow curvature, jet flow or large adverse pressure gradients. However, it gives good results regarding external flow problems around complex geometries. The first transported variable is turbulent kinetic energy and the second one is the turbulent dissipation. As it has been defined, the turbulent kinetic energy defines the energy in the turbulence whereas the turbulent dissipation determines the scale of the turbulence. Wall functions are used in this model, and therefore the flow in the buffer region is not simulated.

The k-omega model, however, uses a different secondary variable, which is the specific dissipation, determining the scale of the turbulence. The initial solution's guess can have a significant impact, and therefore it can be stated that it has a relevant sensitivity to be taken into account, as well as significant difficulty converging as well. It is useful in many cases such as internal flows or flows at which the k-epsilon model does not provide reliable enough results.

2.5.4.2. SST

Until now, both the k-epsilon and the k-omega models have been described. However, there is still an important model not described which combines them both. Its name is SST k-omega model. This turbulence model is a two-equation eddy-viscosity model that has become very popular for several reasons. The shear stress transport (SST) formulation combines advantages from both models. The use of a k-ω formulation in the inner parts of the boundary layer makes the model directly usable all the way down to the wall through the viscous sub-layer, hence the SST k-omega model can be used as a Low-Re turbulence model without any extra damping functions. It switches to a k-epsilon behavior in the free-stream and thereby avoids the common k-omega problem that the model is too sensitive to the inlet free-stream turbulence properties. Therefore, it can be said that it performs well against adverse pressure gradients and separating flow. In short, it can be described as a combination of the k-epsilon in the free stream and the k-omega models near the walls. It does not use wall functions and tends to be most accurate when solving the flow near the wall. The convergence time is probably a minor drawback, since it tends to converge slowly [3 & 4].

2.5.5. Nonlinear viscosity models

Nonlinear viscosity models are a class of turbulence models for the RANS-equations. An eddy viscosity coefficient relates the mean velocity and turbulence fields non-linearly, whereas the Reynold's Stress models (RSM) are higher level, elaborate turbulence models. The Reynolds stresses are computed directly, without considering the eddy viscosity approach. The exact Reynolds stress transport equation accounts for the directional effects of the Reynolds stress fields. It is often defined as the most complete classical turbulence model. In cases of streamline curvature or rotational effects, the Reynolds stress based models can offer much better predictive fidelity. In short, the second moment closure approach offers better accuracy than one or two equation turbulence models and yet is not as computationally demanding as DNS [5].

2.5.6. LES

Large eddy simulation (LES) is a popular technique for simulating turbulent flows. It is a mathematical model for turbulence, which allows to explicitly solving for the large eddies in a calculation and implicitly account for the small eddies by using a subgrid-scale model. The resolved part of the field represent the large eddies, whereas the subgrid part of the velocity represent the small scales whose effect on the resolved field is included through the subgrid-scale model defined. In LES, there is a filtering on the equations. Since in LES there is no time averaging, the filtered variables are functions of space and time. The equations for the filtered variables have the same form as Navier-Stokes equations. LES is intended to reduce the computational cost by decreasing the time range and solved length scales with the use of a low-pass filtering of the Navier–Stokes equations.

In LES, the large-scale motions (large eddies) of turbulent flow are computed directly and only small-scale (sub-grid scale (SGS)) motions are modeled, resulting in a significant reduction in computational cost compared to DNS [6]. LES is more accurate than the RANS approach in several ways, since the large eddies contain most of the turbulent energy and are responsible for most of the momentum transfer and turbulent mixing, and LES captures these eddies directly whereas they are modeled in the RANS approach. Furthermore the small scales tend to be more isotropic and homogeneous than the large ones, and thus modeling the SGS motions should be easier than modeling all scales within a single model (RANS approach). LES can handle many flows that RANS is not able to (for example, separation situations), since in LES large, turbulent scales are resolved [6]. RANS models all turbulent scales, which can be inaccurate, whereas in LES just small, isotropic turbulent scales are modeled, but it is therefore more expensive. However, near the wall region, RANS offers a better prediction, since LES is designed to solve bigger scales.

The subgrid-scale turbulence models are based on the Boussinesq assumption, computing subgrid-scale turbulent stresses from:

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = -2\mu_{t_{SGS}}\overline{S_{ij}}$$

Where $\mu_{t_{SGS}}$ is the subgrid-scale turbulent viscosity, and the last term is the rate-of-strain tensor for the resolved scale.

In LES, the kinematic eddy viscosity is given by the following expression:

$$\nu_t = \nu_{SGS} * (1 - f_{\nu_1})$$

Where ν_{SGS} is the sub-grid scale eddy viscosity and f_{ν_1} is a function called blending function, whose importance is crucial when it comes to low y+, which means at short distances from the wall. This function is limited from 0 to 1 and decreases as the dimensionless distance to the wall increases. Additionally, there is an inverse function, which is f_{ν_2}, which behaves in the opposite way. The definition of the blending function is:

$$f_{\nu_1} = \frac{\chi}{\chi^3 + c_{\nu_1}^3}$$

Where χ is defined as $\frac{\tilde{\nu}}{\nu}$ and c_{ν_1} is a coefficient equal to 7.1. Therefore, the previous expression can be rewritten as:

$$f_{\nu_1} = \frac{(\frac{\tilde{\nu}}{\nu})^3}{(\frac{\tilde{\nu}}{\nu})^3 + 7.1^3} = \frac{\tilde{\nu}^3}{\tilde{\nu}^3 + 357.911\nu}$$

And the second blending function is therefore defined as:

$$f_{\nu_2} = 1 - \frac{\chi}{1 + \chi f_{\nu_1}} = 1 - \frac{\frac{\tilde{\nu}}{\nu}}{1 + \frac{\tilde{\nu}}{\nu} f_{\nu_1}}$$

Regarding the RANS region, the following expression is used:

$$\nu_t = \tilde{\nu} * f_{\nu_1}$$

What this ensures is that close to the wall (on the RANS region), where the dimensionless distance to the wall is close to 0, $\nu_t = \tilde{\nu}$, since f_{ν_1} will tend to 1. Similarly, $\nu_t = \nu_{SGS}$ when $1 - f_{\nu_1}$ equals 1. That happens when since f_{ν_1} is close to or equal to 0, which happens at the LES region, when the dimensionless distance to the wall is big enough.

This explains how the transition between the LES and RANS region is taken into account regarding the kinematic eddy viscosity. The hybrid methods, which are explained in the next section, take this transition into account and that is how they work when it comes to predicting the behavior of this magnitude in the transition between both regions.

2.5.7. Hybrid methods (DES, DDES, IDDES)

The stated difficulties associated with the use of the standard LES models, particularly in near-wall regions, led to the development of hybrid models that attempted to combine the best aspects of both RANS and LES methodologies in a single solution strategy. Detached Eddy Simulation (DES) models switch explicitly, therefore, between RANS and LES model formulations based on the local grid spacing and turbulent length scale. The original purpose of DES was to be run in RANS mode for attached boundary layers and to switch to LES mode in large separated (detached) flow regions. The explicit switch to LES model is, however, not accompanied by a corresponding transfer of modeled (RANS) turbulence to resolved (LES) turbulence. Due to the direct impact of the grid spacing on the RANS model, DES models require more carefully crafted grids to avoid inappropriate behavior. On the other hand, DES models allow a local reduction in eddy-viscosity by grid refinement in the transition region between RANS and LES [7].

However, problematic behavior regarding the DES method was reported by Menter and Kuntz [8], who demonstrated that an artificial separation could be produced for an airfoil simulation when refining the max cell edge length inside the wall boundary layer below a critical value. This effect was termed Grid Induced Separation (GIS) as the separation depends on the grid spacing and not the flow physics. GIS is produced by the effect of a sudden grid refinement that changes the DES model from RANS to LES, without balancing the reduction in eddy-viscosity by resolved turbulence content [7].

Additionally, Spalart [9] defined the Modeled Stress Depletion (MSD), which can be defined as a result of insufficient flow instabilities downstream of the switch from the RANS to the LES model formulation. The flow instability is too weak and it would require many boundary layer thicknesses to allow the formation of a sufficiently developed turbulent LES content to balance the reduction of the RANS model. For that reason, the switch from the RANS to the LES model inside wall boundary layers is not desirable.

GIS could be avoided by shielding the RANS model from the DES formulation for wall boundary layers (Menter and Kuntz [8]), by using the blending functions of the SST model. Later, Spalart et al. [9] proposed a more generic formulation of the shielding function, which depends only on the eddy-viscosity and the wall distance. It can therefore, in principle, be applied to any eddy-viscosity based DES model. The resulting formulation was termed Delayed Detached Eddy Simulation (DDES), which can be described as an improved method.

The development and calibration of DDES shielding functions requires a delicate balance between the need of shielding the boundary layer and the desire of not inhibiting the formation of turbulent structures in the transition zone between attached (RANS) and detached (LES) flow. Overly conservative shielding would allow a high degree of mesh refinement inside the boundary layer without any impact on the RANS model, but would suppress the formation of resolved turbulence in detached flow regions not sufficiently removed from walls.

Another interesting aspect spurring many discussions and model enhancements resulted from the application of the original DES model as a WMLES formulation. These tests indicated that DES could be developed into a suitable WMLES formulation, resulting in the formulation of the IDDES model, Shur et al. [10].

The IDDES model, an even improved DES model, features several blending and shielding functions, which allow using this model both in DDES and WMLES mode. Further details on what improvements brings can be seen on the comparison made on the next section. The behavior and prediction of the kinematic eddy viscosity follows the explanation on the previous section. Taking that into account, the transition between the LES and the RANS region is considered regarding the prediction of this magnitude.

2.5.8. Direct Numerical Simulation (DNS)

A direct numerical simulation (DNS) is a simulation in computational fluid dynamics in which the Navier-Stokes equations are numerically solved without any turbulence model.

This means that the whole range of spatial and temporal scales of the turbulence must be resolved. All the spatial scales of the turbulence must be resolved in the computational mesh, therefore making the whole cost higher, the time longer and the computational resources should be more precise, but the results are more precise than the ones predicted by the models described.

A short comparison between some of the models described can be seen in Figure 6, which helps to understand the differences between them in several ways.

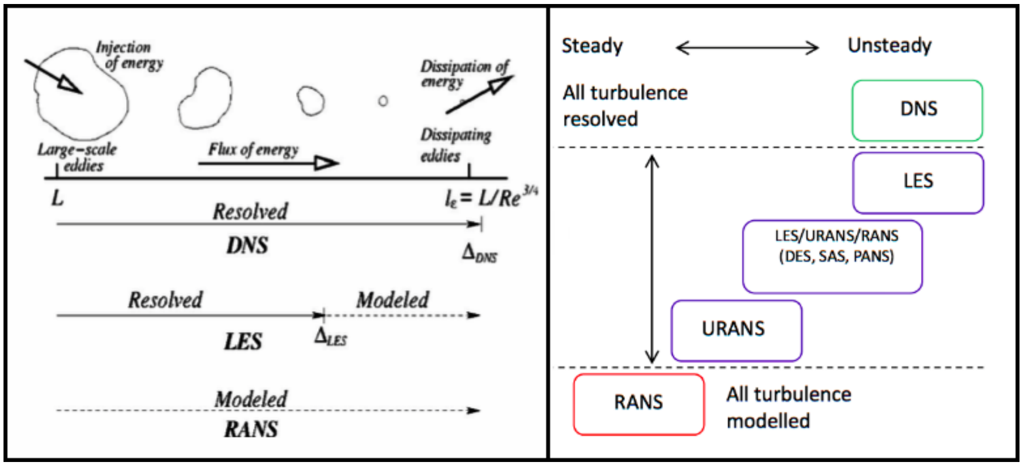


Figure 6 - Short modelling comparison

2.6. Turbulence models comparison

In this section, a comparison of the turbulence models that will be used in the project will be done. Since all models have some pros and cons, the best way to describe them all in an intuitive and understandable way is by designing a table in which they can be listed. Therefore, the main differences can be seen and understood easily. The compilation has been made at Table 1.

Turbulence model	Pros	Cons
DNS	No modeling of turbulence, full turbulent spectrum resolved	Requires extremely fine meshes and timescales Only affordable at low Reynolds numbers Huge computational resources
kε	Robust for a wide range of models Industry applications Low wall-meshing requirements Good results for non-separated flows and fully-turbulent ones Usable to do a gross estimation	Performs poorly against large adverse pressure gradients Poor performance regarding separated flows Difficult to integrate through the viscous sublayer

	of the flow field Fast convergence	Low accuracy regarding detachment point placement Very sensitive to mesh refinement Inaccurate at large Pr values
kω	More accurate near-wall treatment Good performance for wall-bounded and low Re flows Better performance against adverse pressure gradients, separate and free-stream flows Works under complex boundary layers Easily integrated through the viscous sublayer	Underprediction of the amount of separation for large adverse pressure gradients High resolution near the wall required Early separation predicted Problems when facing free-stream boundaries
kω-SST	Robust for a wide range of models Very good near-wall behavior Decent flow separation prediction Good performance against adverse pressure gradients High accuracy regarding boundary layer separations	High resolution mesh required in terms of layers Too large turbulence levels in regions with large normal strain (stagnation regions and regions with strong acceleration)
DES	Copes well with separated and high Re flows Copes with the non-physical	Computational effort and mesh refinement required Possible formation of 1D

	behavior in the vicinity of the boundary layers Better prediction of pressure, forces and moments Easy implementation Good results at moderate cost [11]	eddies and excessive damping of turbulence at the RANS-LES interface Possible mismatch between the modeled log layer and the resolved log layer Very sensitive to length scale and mesh width [11]
Spallart-Allmaras	Computational efficiency and robustness Improved wall-bounded behavior prediction Affordable cost Good behavior against adverse pressure gradients flows in boundary layers [12]	Improvable free-shear flows behavior (specially plane and round jet flows) Problematic rapid change in length scales associated with the transition from wall-bounded to free shear Still not implemented in a wide range of cases [12]
DDES	Good results with separated and high Re flows Solved mismatch between the modeled log layer and the resolved log layer problem Better handling of ambiguous grids Copes with the non-physical behavior in the vicinity of the boundary layers.	Computational effort and mesh refinement required Possible formation of 1D eddies and excessive damping of turbulence at the RANS-LES interface Mainly still for academic use Highly sensitive to mesh refinement

	[13]	Improvable accuracy and results in the near-wall region Possible mismatch between the modeled log layer and the resolved log layer [13]
IDDES	Allows the activation of RANS and LES in different flow regions, well-balanced and powerful numerical approach to complex turbulent flows at high Reynolds numbers Better prediction of the detachment behavior It does not allow the formation of smooth nearly one-dimensional eddies and excessive damping of turbulence at the RANS-LES interface Improved near-wall region results Performs well at high, moderate and even low Re_τ Resolves completely the mismatch between the modeled log layer and the resolved log layer issue [14]	Computational effort and mesh refinement required Academic use, still not much implementation in real industry cases Highly sensitive to mesh refinement Wall-parallel grid needs to be fine enough [14]

Table 1 - Turbulence models main features comparison

3. Simulation setup

Two different scenarios will be designed, based on the purpose on Octavian Frederich's paper [15]. The first one will include a larger domain in which three elements can be distinguished. These elements are a plate, a trip wire (to induce turbulence) and the solid to study, the cylinder. Three different trip wire configurations have been designed, having a circular, rectangular and triangular shape respectively, in order to study the effects of this particular element in the flow behavior after performing the simulations.

The solids have been designed with SolidWorks based on the design parameters that will be described in the upcoming lines, and exported as STL files both separately (by exporting every face) and together as closed surfaces. These STL files can be imported to paraView in order to watch them in a more clear way and observe if the meshing that will be performed is suitable or not. The purpose of designing two scenarios is to obtain a reliable value of the different properties at the beginning of the plate, by means of improving the reliability of the simulation. Therefore, the first scenario will be bigger than the second one. In this scenario, the boundary and initial conditions will be defined and then, the first simulation will be performed to obtain the desired values at the cutting plane, which will be the inlet of the second simulation. The surfaces' measures that are included in the domain (since they are slightly larger than it in the X and Y axes as it is explained in upcoming paragraphs) can be seen at figure 7:

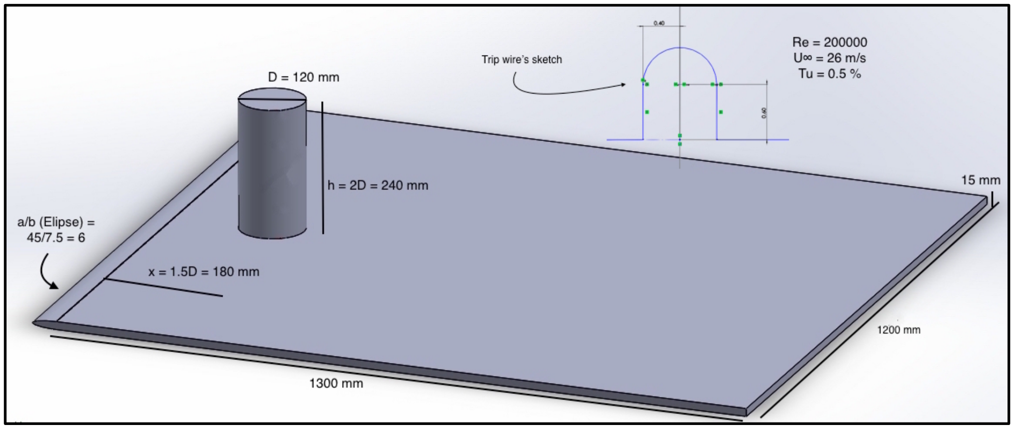


Figure 7 - Sketch of the domain and the main measures

As it can be seen, the cylinder is placed closer to the leading edge than to the trailing edge, in a symmetrical way with respect to the y-axis. The dimensions of the solids have been extracted from Frederich's thesis [15], and so has been the Reynolds number that will be used in this simulation. The different magnitudes and the corresponding values have been summed up in Table 2. x stands for the distance between the middle of the trip wire in the x-axis and the location of the cylinder center.

3. Simulation setup

The STL surfaces are defined in the x-direction from -0.06 to 0.06 for the cylinder and from -0.225 to 1.17 for the plate. In the y-direction, from -0.06 to 0.06 for the cylinder and from -0.7 to 0.7 for the plate. In the z-direction, from -0.01 to 0.24 for the cylinder and from -0.015 to 0.001 for the plate (which means that the STL surface will be "submerged" in the plate surface, in order to avoid some problems with the meshing process, and to avoid further meshing troubles related to Boundary Conditions, the STL surfaces are designed slightly bigger than the domain in the y and the z axis). The trip wire surface is defined by creating a circle with a diameter of 0.4mm followed by straight lines tangent to the circle down to the plate surface with a length of 0.6mm. United to the sketch of the plate, the surface designed in a y-normal slice is then extruded as a solid along the y-axis with SolidWorks in a symmetrical way.

The domains will be defined from -2.605m to 1.12m and from -0.225 to 1.12 in the x-axis for the first case and the final one respectively, from -0.6 to 0.6 (10 times the diameter of the cylinder) in the y-axis and from -0.125 to 0.84 and -0.0075 (half of the plate) to 0.84 in the z-axis for the first and final cases respectively.

3.1. Turbulence parameters

Magnitude	Nomenclature	Value
Turbulence kinetic energy	k, k_{in}	0.02535
Turbulence dissipation	ε, ε_{in}	0.37074375
Kinematic viscosity	υ	0.156
Turbulent viscosity	υ_t	$1.56 * 10^{-4}$
Turbulence length scale	l	$9.798 * 10^{-4}$
Empirical turbulence ct.	C_μ	0.09
Specific dissipation	ω	162.5
DES coefficient	C_{DES}	0.75
Spalart-Allmaras Coefficient	α_{υ_t}	1.5
Spalart-Allmaras Coefficient	C_{b1}	0.1355
Spalart-Allmaras Coefficient	C_{b2}	0.622
Spalart-Allmaras Coefficient	C_{w2}	0.3
Spalart-Allmaras Coefficient	C_{w3}	2
Spalart-Allmaras Coefficient	C_{υ_1}	7.1
Spalart-Allmaras Coefficient	C_{υ_2}	5
Spalart-Allmaras Coefficient	C_k	0.07
Delta coefficient	Δ	2 [24]

Table 2 - Turbulence parameters

Some of the values summed up on the table have been computed, and not just extracted from the literature references. The following calculations have been made to obtain some of the values presented:

$$k = \frac{3}{2} Tu^2 {U_\infty}^2 = \frac{3}{2} 0.005^2 26^2 = 0.02535$$

$$k_{in} = k = 3.75 * 10^{-5} * {U_\infty}^2 = 3.75 * 10^{-5} * 26^2 = 0.02535$$

$$\varepsilon in \ = \ 2.53125 \ \cdot \ 10^{-6} \ \frac{{U_\infty}^3}{D} = 2.53125 \ \cdot \ 10^{-6} \ \frac{26^3}{0.120} = 0.37074375$$

$$\nu_t = 10^{-3} \nu = 10^{-3} * (1.56 * 10^{-7}) = 1.56 * 10^{-4}$$

$$\varepsilon = \varepsilon in = \frac{C_\mu k^2}{\nu_t} = \frac{0.09 * 0.02535^2}{1.56 * 10^{-4}} = 0.37074375$$

$$l = C_\mu \frac{k^{\frac{3}{2}}}{\varepsilon} = 0.09 \frac{0.02535^{\frac{3}{2}}}{0.37074375} = 9.798 * 10^{-4}$$

$$\omega = \frac{\sqrt{k}}{l} = \frac{\sqrt{0.02535}}{9.798 * 10^{-4}} = 162.5$$

An overview of this first setup, whose mesh can be seen in Figure 8 (see the meshing section for more details), can be seen at the Figure 9.

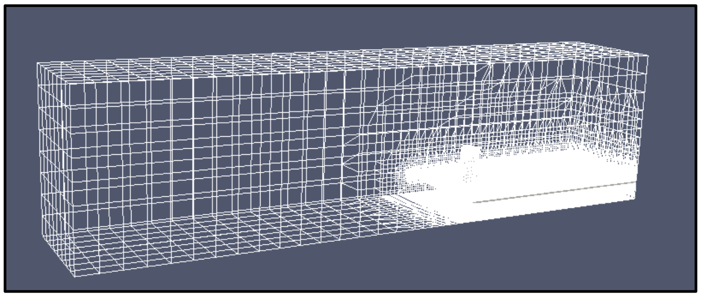


Figure 8 - First simulation domain

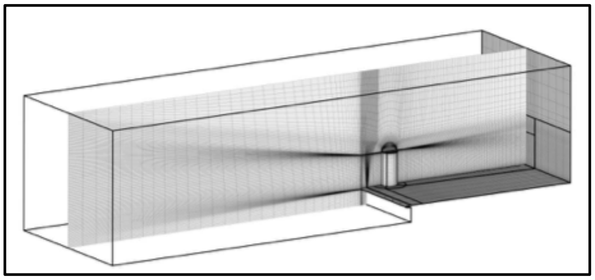


Figure 9 - Sketch of the first simulation

Finally, after obtaining the desired variables, the second environment will be designed, by cutting a plane at the x value corresponding to the beginning of the plate being it the new inlet and importing the results from the previous simulation. The physical properties and the geometrical design will, however, not change from the first configuration to the second one.

3.2. Boundary conditions

It is well known that the domain will contain a couple of solid structures around which the flow will go through and an entire domain that has to be set up. To define what is happening in this particular domain, a set of constraints, which are called the boundary conditions, need to be considered. In this case, there is an inlet, an outlet and the plate and the cylinder are considered as a wall. Furthermore, the condition of symmetry on the upside boundary limit and the side domain limits is present as well. Slip conditions are applied to the upper and lateral boundaries, whereas no-slip conditions are applied to all physical walls. To ensure the non-reflective movement of arbitrary flow structures across the downstream boundary, a convective outflow condition $\frac{\partial \Phi}{\partial t} + U_m \frac{\partial \Phi}{\partial n} = 0$ is used, where U_m is the average outflow velocity to conserve the mass flux and n is the direction normal to the outflow boundary.

Constant velocity profile at the inlet will be considered for the preliminary investigations and the first high-resolution simulations ($U_\infty = ct.$). Due to the proximity of the inlet boundary and cylinder, a spatially variable inflow profile is required taking into account the obstacle blockage. A steady inflow profile has been imposed, which is considered acceptable due to the low turbulence intensity and the expected laminar regions on the cylinder front.

A summary of the boundary conditions (with openFOAM's nomenclature) applied in the first simulation can be seen at Table 3:

Field	**Patch**	**Condition**
U	inlet	fixedValue (U_∞ uniform (26 0 0))
	outlet	inletOutlet (inlet U_∞ uniform (26 0 0))
	upperWall	symmetry
	lowerWall	symmetry
	lateralWall	symmetry
	cylinder	fixedValue uniform (0 0 0)
	plate	fixedValue uniform (0 0 0)
p	inlet	zeroGradient
	outlet	fixedValue uniform 0
	upperWall	symmetry
	lowerWall	symmetry
	lateralWall	symmetry
	cylinder	zeroGradient
	plate	zeroGradient
k	inlet	fixedValue uniform 0.02535

	outlet	zeroGradient
	upperWall	symmetry
	lowerWall	symmetry
	lateralWall	symmetry
	cylinder	kqRWallFunction uniform 0
	plate	kqRWallFunction uniform 0
ω	inlet	fixedValue uniform 162.5
	outlet	zeroGradient
	upperWall	symmetry
	lowerWall	symmetry
	lateralWall	symmetry
	cylinder	omegaWallFunction uniform 162.5
	plate	omegaWallFunction uniform 162.5
ν_t, ν_{Tilda}	inlet	fixedValue uniform 0
	outlet	zeroGradient
	upperWall	symmetry
	lowerWall	symmetry
	lateralWall	symmetry
	cylinder	nutUSpaldingWallFunction uniform 0
	plate	nutUSpaldingWallFunction uniform 0

Table 3 - Boundary conditions

The above explained boundary conditions are based on a literature research, especially based on [21] and [25], which have provided good results in the respective simulations performed.

Regarding the spatial and time discretization, the guidelines that have been followed are inspired by literature researches as well, especially on [22] and [23]. Several configurations have been tried out, but many gave errors, non-convergence or just floating point exceptions, since they were not able to handle the simulations. If the reader wants to see specifically the methods used, the main files regarding discretization and parameters taken into account can be seen in the Appendixes section at the end of the document.

3.3. Case file structure

An OpenFOAM case folder contains 3 initial and major folders, which are the 0, constant and system folders. Furthermore, several additional folders might be created afterwards and can contain post-processing elements or studies, processor folders if the user wants to run a decomposed case to speed up the simulation, a source case folder used to map the fields from a first steady state simulation to initialize the flow, as it will be done in this project or the time folders containing the results after running a simulation. This last case is just a folder containing the 3 major folders of the case which is used as an initialization of the flow, whose magnitudes will be mapped to the 0 folder of the current study case. The most basic case can be seen on Figure 10.

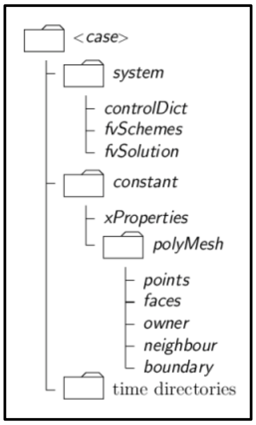


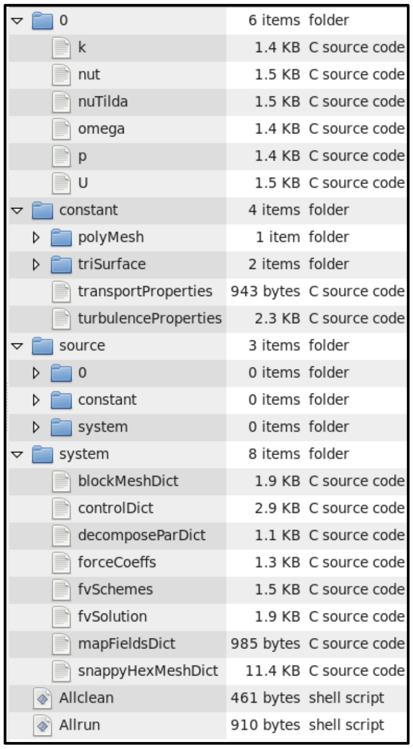
Figure 10 - Basic case structure

The 0 folder contains the initial conditions for all the elements present on the simulation, which can be walls, inlet and outlet, and even empty walls. The different relevant magnitude files, which need to be initialized for each type of simulation (for instance, velocity or pressure conditions), are located here, and must be defined for all the elements without exception before running the case, since it will not work otherwise.

In the constant folder, the user might just see the polyMesh folder, the transportProperties and, if the simulation involves turbulence, the turbulenceProperties file as well. Furthermore, other folders can be included, for instance, if the user wants to use the snappyHexMesh tool to mesh, the triSurface and the extendedFeatureEdgeMesh folders will be located here as well, which contain the STL files needed (in this project's case, the cylinder and the plate) and the corresponding edge extraction. The polyMesh folder, however, contains the files that define the mesh created for the current study.
Finally, the system folder is the most versatile of all. In this folder, the user can place endless dictionaries and functions that will be needed for the simulation. The most important ones are the blockMeshDict file, which defines the first mesh for this project, which will be used as a base for the second mesh using the tool described, the

controlDict file, which contains all the running properties such as the initial and final times, the time step and several other relevant parameters and functions to be included in the current simulation, the fvSchemes and fvSolution files, which define the differential schemes and the solvers that will be used for each variable and the snappyHexMeshDict file, in which the user can define the parameters regarding the use of this tool, such as the refinement levels, the use and placement of refinement boxes, and many other important aspects to consider that can greatly improve the meshing for the case.

The most relevant files of this particular folder and the constant one will be included in the annexes section of this document, so that the reader can understand with which kind of elements we are dealing with and the importance of each one of them regarding the final results. A case sample structure can be seen in Figure 11, which contains mainly the most relevant elements to run a simulation as it has been described in this section. Several redundant files have been hidden but the most important and relevant ones are all located where they should, so that the reader can understand better how the structure of a real case folder works.



```
▽ 0                          6 items  folder
    k                         1.4 KB  C source code
    nut                       1.5 KB  C source code
    nuTilda                   1.5 KB  C source code
    omega                     1.4 KB  C source code
    p                         1.4 KB  C source code
    U                         1.5 KB  C source code
▽ constant                   4 items  folder
  ▷ polyMesh                  1 item  folder
  ▷ triSurface               2 items  folder
    transportProperties    943 bytes  C source code
    turbulenceProperties      2.3 KB  C source code
▽ source                     3 items  folder
  ▷ 0                        0 items  folder
  ▷ constant                 0 items  folder
  ▷ system                   0 items  folder
▽ system                     8 items  folder
    blockMeshDict             1.9 KB  C source code
    controlDict               2.9 KB  C source code
    decomposeParDict          1.1 KB  C source code
    forceCoeffs               1.3 KB  C source code
    fvSchemes                 1.5 KB  C source code
    fvSolution                1.9 KB  C source code
    mapFieldsDict          985 bytes  C source code
    snappyHexMeshDict        11.4 KB  C source code
  Allclean                 461 bytes  shell script
  Allrun                   910 bytes  shell script
```

Figure 11 - Case structure

3.4. Simulation procedure

First of all, we need to extract the eMesh files from the geometries, which are the STL files that have been created previously to the simulation. In order to do that, the command used will be surfaceFeatureExtract. Therefore, the feature edges from the geometry surface will be extracted. The next step will be to create the background or block mesh, with the command blockMesh. This command will check for the entries defined at the corresponding blockMeshDict on the system folder, which can be the vertices of the mesh, the edges, the blocks defined and many other aspects. Once it has been created, its correctness can be checked with the command checkMesh. The output will indicate the user whether there are some errors (orthogonal errors, illegal faces, …) or not, and then the background mesh will be ready.

Since the case is a bit complex and it could take a lot of computational resources and time to compute, the best option is to decompose the case into several processors to run the commands in parallel, reducing the required amount of time significantly. To do that, first of all the number of available processors has to be checked, if the user wants to make use of all of them, and this is done with the command nproc. In our case, the number of available processors is 16, and therefore, the case will be decomposed into 16 processor folders with the command decomposePar –force, being the last –force optional (just in case there are other previous processor folder that are not needed). Once it has been decomposed, the meshing with the snappyHaxMesh utility can begin. When it finishes, both reconstructParMesh and reconstructPar –constant (no time elapsed) must be run in order to recompose the case and form the final mesh before computing.

Once the mesh has been successfully setup, the next step will be to run the first simulation, which will be used as a source for the next ones. A good way to initialize the flow is by optionally performing a potential flow simulation, mapping the fields to a kwSST simulation with just the simpleFoam solver being used in the first domain and afterwards the results will be mapped again as a starting point for the final configuration. If the simulation report gives no errors at all, it can be used as an initialization to the flow for the next simulations, whose results can be seen on the results section. Four different methods will be used, but the base to all of them will be the same, and therefore the fields will be mapped prior to each simulation's run.

Finally, after mapping the fields and setting up the configuration fields for each simulation, the pimpleFoam solver can begin to run and the results can be afterwards analyzed with paraview and external tools, which will help to plot them and understand the behavior presented in the results section.

3.4.1. Meshing

In order to perform a proper meshing, several steps must be followed. The meshing tool that will be used is snappyHexMesh, which is a tool that designs a proper meshing around a solid structure (in this case, around the cylinder and the plate/wire structure), and its working principle is explained in Chapter 5.4 of openFOAM's user guide [16]. It generates 3-dimensional meshes containing hexahedra automatically from triangulated surface geometries, or tri-surfaces from the STL files designed. The mesh approximately conforms to the surface by iteratively refining a starting mesh and morphing the resulting split-hex mesh to the surface. The specification of mesh refinement level is very flexible and the surface handling is robust with a pre-specified final mesh quality.

First of all, once the solids have been designed and extracted in the proper format, several archives must be modified in order to allow openFOAM operate with them with the right syntax.

The first document to modify is the blockMeshDict, which is the blockMesh dictionary. In this document, a mesh with a block shape must be designed. The design parameters and functions can be seen at the user guide [16]. At this document, the right coordinates must be introduced, as well as the orientation of the faces and their definition. The command checkMesh will help us understand whether there is a problem with the mesh or not. Therefore, it is always recommended to check it by just typing this command.

The second relevant document that will be modified is the snappyHexMeshDict. In this dictionary, the geometry parameters will be defined, as well as some refinement parameters, refinement boxes and levels. A deeper explanation of these functions can be seen at the user guide [16].

First of all, the overall domain must be created with the blockMesh command. If everything worked out well, checkMesh should not give any errors in the report. After that, the snappyHexMesh command can be executed. A parallel decomposition in several cores is recommended, since it is a more efficient way of computation and reduces significantly the computation time and resources needed.

Once all these steps are finished, the mesh should be ready, and the simulation can begin after defining the boundary and initial conditions at the respective archives. Several figures of the second setup will help to understand the differences between meshing configurations and how the process can be greatly improved.

First of all, a slice of the blockMesh created can be seen in Figure 12. In it, a really coarse mesh can be seen, which will be the base around which snappyHexMesh will build the final mesh.

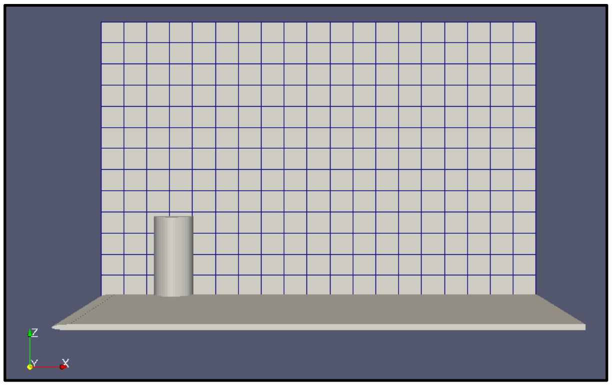


Figure 12 - blockMesh y-normal slice

After using the snappyHexMesh tool, several meshes can be obtained based on the base environments created. The following images, Figures 13, 14 and 15, will compare the results from the coarsest mesh (6-6-4 division) and the finest mesh (19-19-13 division), by using 3 slices in the x, y and z axes respectively. The setup in the snappyHexMeshDict file is the same, and the only thing that changes is the division made in the blockMeshDict. The number of cells used for each domain will be later on detailed on the results section.

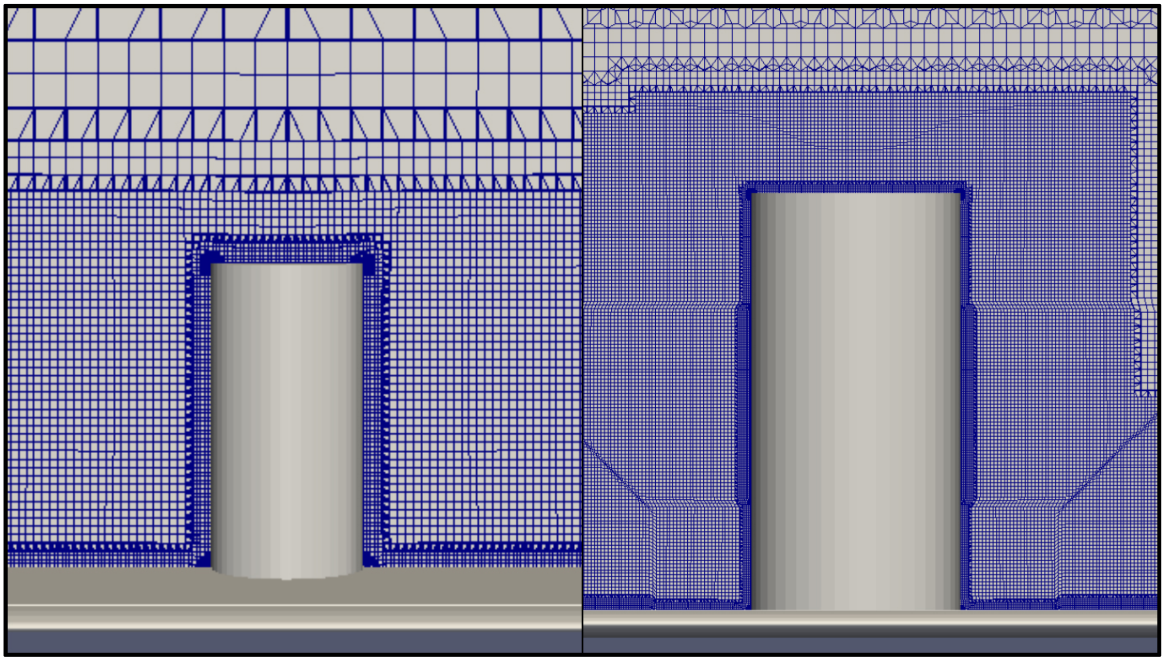


Figure 13 - Coarsest and finest mesh comparison x-normal slice

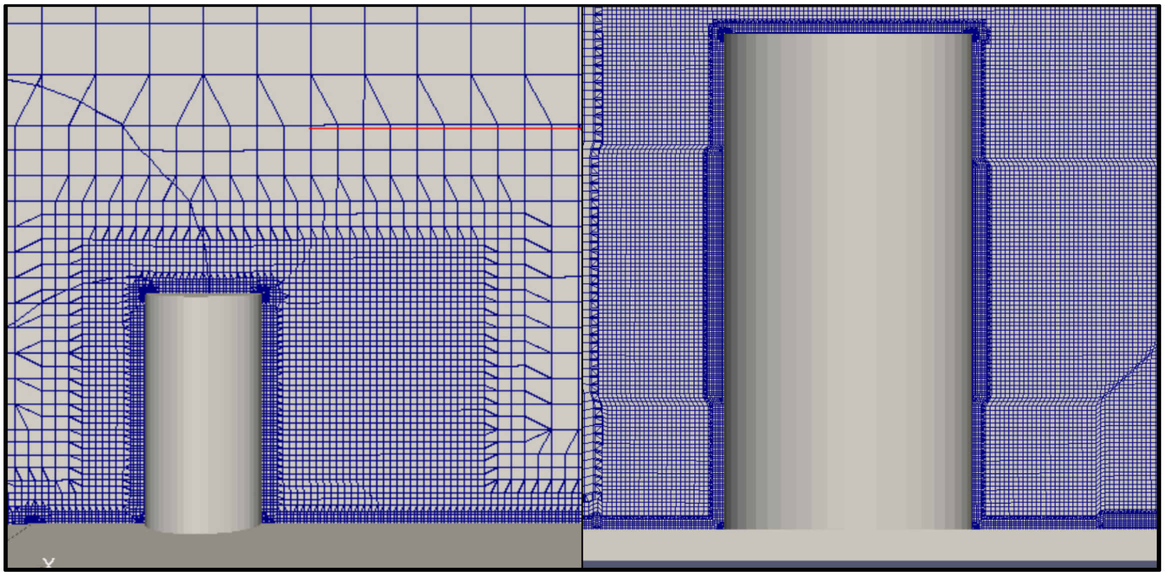


Figure 14 - Coarsest and finest mesh comparison y-normal slice

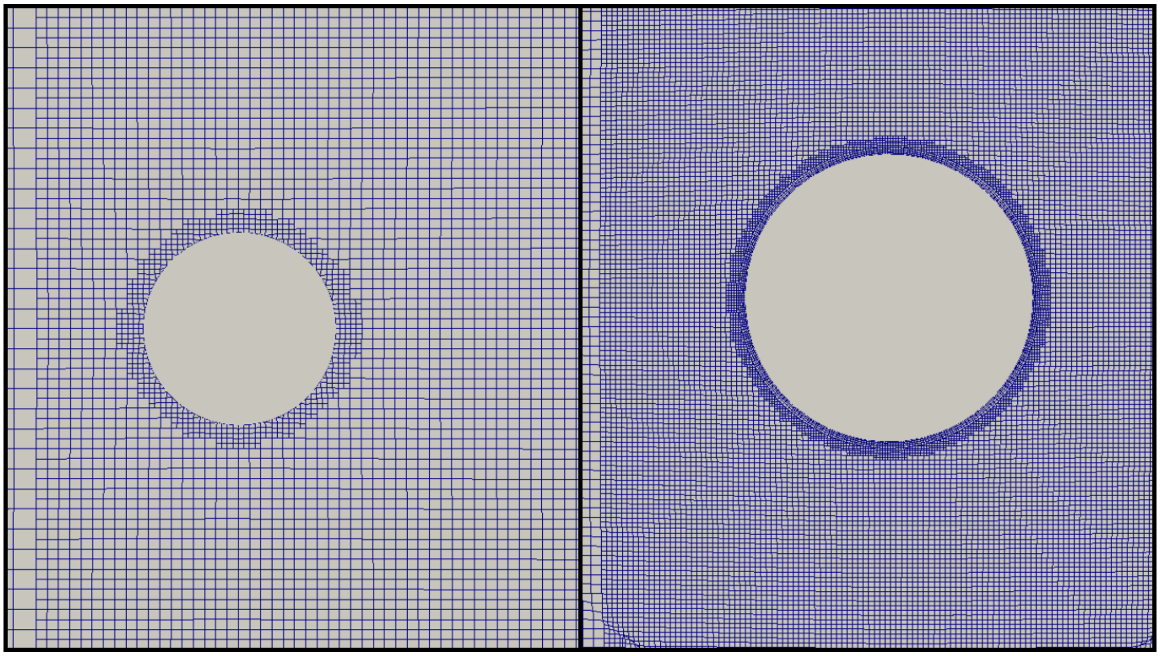


Figure 15 - Coarsest and finest mesh comparison z-normal slice

Additionally, several changes in the snappyHexMeshDict file have been taken into account to mesh properly around the cylinder surface, which suppose a great improvement in the results, such as layer thicknesses, feature angles or surface iterations. A comparison from both the starting and the improved case can be observed in Figure 16, just to give an idea of how this improvement allows the user to get better results.

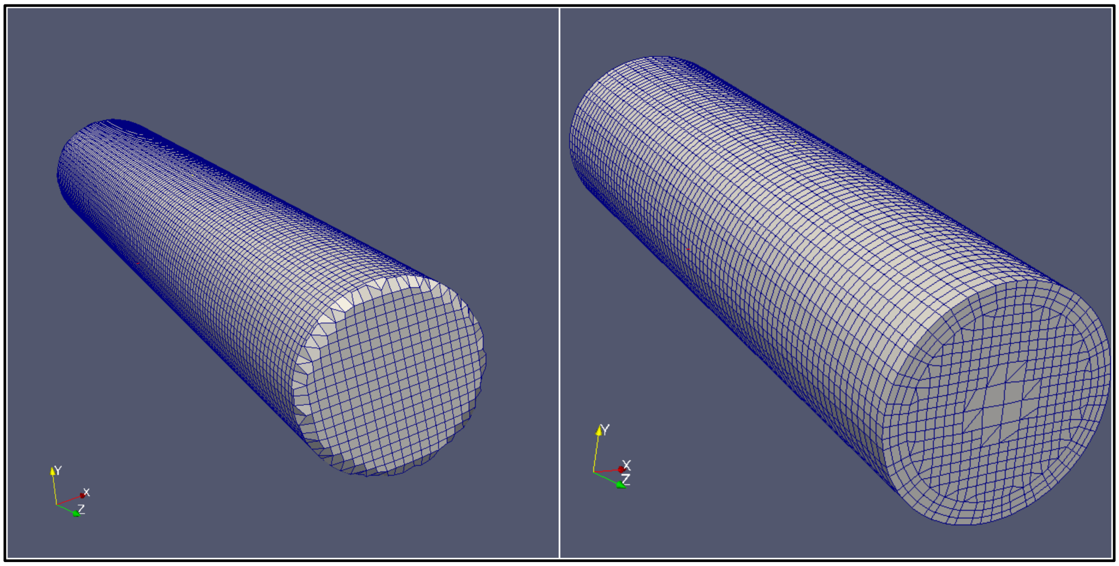


Figure 16 - Meshing around the cylinder improvements

Finally, the final configuration used can be summed up in Table 4, which includes the most relevant parameters considered by this tool and the respective values or configuration used. To view all of them, as well as a short description, the snappyHexMeshDict file will be included in the annexes section.

Element	**Value**
castellatedMesh, snap & addLayers	true
Refinement box refinement	Level 4-5
maxLocalCells	3000000
maxGlobalCells	10000000
minRefinementCells	10
nCellsBetweenLevels	4
Edge refinement level around surfaces	Level 7
Surface based refinement	Level 6-7
nSolveIter	300
nRelaxIter	5
nSurfaceLayers added	4
expansionRatio	1.15
finalLayerThickness	0.9
minThickness	0.05
featureAngle	180
nLayerIter	50
mergeTolerance	1e-6

Table 4 - SnappyHexMesh main parameters

4. Results

4.1. Reference results comparison

As a staring point, taking into account the reference values from the literature can be really helpful to compare both the experimental and simulated values. Several studies regarding this cylinder case have been conducted. They include experimental studies and numerical simulations as well. In Table 5, a comparison of the values coming from the literature reference studies has been made. In it, the key parameter to be compared is the time-averaged drag coefficient in the x direction, but since not all of the studies have been conducted with the same Reynolds number, the different numbers used for each study have been written down as well. The Reynolds number used in this project is 200000, and therefore, the last two values on the table will be the ones that theoretically will provide a better approach to look for and compare. It does not strictly mean that the value that will be obtained in the study must be in the range described, but it is a good reference nonetheless.

Reference	**Re**	$\overline{C_x}$
Kawamura et al. (1984)	32000	0.78
Okamoto & Sunabashiri (1992)	47000	0.73
Fröhlich & Rodi (2004)	43000	0.88
Pattenden et al. (2005)	200000	0.79
Frederich et al. (2008)	200000	0.82
Contel (2016)	200000	0.76

Table 5 - Reference Re and cd results

In Table 6, several values have been selected to compare. The magnitudes studied are the two first Strouhal numbers [17] and the time-averaged drag coefficient in the x direction. The values selected from all the simulations will be the ones corresponding to the finest possible mesh, which should give the better result to consider.

Simulation	St_1	St_2	$\overline{C_x}$
SA-DDES	0.1571	0.2187	0.71
SA-IDDES	0.1741	0.2177	0.76
kwSST-DDES	0.1653	0.1929	0.74
kwSST-IDDES	0.1667	0.2037	0.70
LES (Frederich)	0.16	0.20	0.82
DES (Frederich)	0.15	0.21	0.55
IDDES (Frederich)	0.16	0.20	0.56

Table 6 - Literature references St1, St2 and cd comparison

4.2. Steady-State RANS Simulation

The starting point of the simulations performed is the steady state simulation that is used to initialize the flow and avoid high oscillations and undesired results at the first iterations. As mentioned, each simulation takes into account a background of up to 3 simulations. The first environment is bigger than the final one, and is just used as a starting state before performing the final simulation for each simulation. The mesh is coarser than the mesh that will be used in further configurations, since this step is just used to initialize the flow. A prior step, which is optional, is to initialize the flow with potentialFoam, and then map the fields, run the second simulation and map again the results on the final configuration. This step is, however, avoidable with a significant number of iterations regarding the steady state simulation, or by observing the convergence of the different magnitudes. If the magnitudes show a converged behavior, the number of iterations should be enough as an initialization of the flow.

The convergence of these magnitudes can be checked by plotting the residuals presented at the simpleFoam's log file. A straightforward script can be generated and gnuplot is the tool used to plot the results, which can be seen at Figure 17.

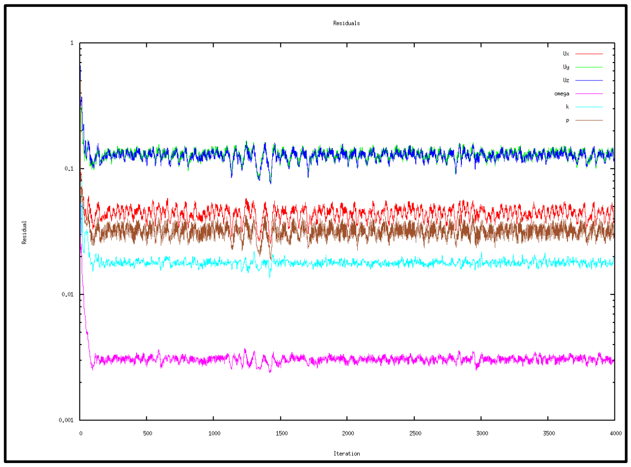


Figure 17 - Residuals of the Steady-State RANS simulation

As it can be seen in Figure 17, after 4000 iterations the magnitudes tend to still oscillate but around a converged-enough value to be considered as the starting point for further configurations, whose meshing and results are better.

After checking the results, some asymmetrical behavior could be seen in the figures that are shown in the next sections, and that can be explained by the fact that the velocity profile is not symmetrical at the Steady-Steate RANS Simulation, due to the fact that the geometries were slightly shifted in the y-axis by 0.015m, which will be expanded in the conclusions section.

4.3. Techniques comparison

The results obtained regarding the 4 main simulations performed are analyzed in this section. An in-depth comparison and analysis of the different magnitudes represented has been made, so that the main differences regarding the magnitudes studied are pointed out. The main objective is to prove that the theoretical differences between the techniques used are present in the simulated cases as well, and that is what has been done. The results are presented by comparing the 4 simulations in the following manner: from left to right, SADDES, SAIDDES and the two slots below correspond to kwSSTDDES and kwSSTIDDES. For several magnitudes, where the differences need to be appreciated in a more careful manner, more zoom is needed or an extra comparison has been made (especially those including additional studies) are placed below the main comparison regarding the magnitude that is being described at the specific section. That means, that if the velocity is being described, and an additional study such as the non trip-less approach has been considered, those comparisons will be placed below the main comparison of the velocity field. It is also important to recall that the magnitudes studied are always the mean values time-averaged from a point at which averaging makes sense, which means that is already converged or close to it.

The colorbars have been scaled in order to display the same range for all simulations, making the results more readable and easier to compare. Furthermore, the color range has been reduced to 20 instead of the default value of 256, making the gradients more visible and easier to compare as well. In most of the cases, two slices have been considered. They are mainly y-normal and z-normal slices, which allow the user to easily read the results. Regarding the slices, they have been done at the origin and at z=D, which is the z-coordinate corresponding to the half of the cylinder, so that the results are easier to compare with literature studies, which also consider the slices at this specific height.

4.3.1. Velocity

Before showing the results obtained, a short description of the theory behind should be made as a starting point, in order to make things easier to understand and to make the comparisons more clear.

An area of reduced pressure manifests itself direct downstream the cylinder top due to the separated shear layers. The flow from the top is vacuumed downwards and the expansion of the separated area falls down quickly. Furthermore, the fluid masses from both the separated layer and the one providing from the top of the cylinder meet face each other, generating an eddy.

Near to those eddies, induced longitudinal velocities are formed contrary to the movement of the fluid, generating a recirculation zone behind the cylinder. Corners, sharp turns and high angles of attack in the case of airfoils represent sharply decelerating flow situations where the loss in energy in the boundary layer ends up leading to separation. Once the flow is separated, the recirculation zone is generated as described before. The earlier the separation is produced, the bigger the recirculation zone will be and the sooner it will be generated, increasing the drag.

To sum up, when the upstream velocity faces the cylinder surface, several eddies are generated at different regions. Horseshoe vortexes are generated at the sides of the cylinder close to the surface of the plate, a recirculation vortex that later on reattaches is generated at the region above the cylinder as the flow faces the sharp edge of the top of the cylinder, side vortexes are generated due to the separation of the flow around the cylinder at the top of it and a bigger vortex structure, the recirculation zone, forms for the reasons described behind the cylinder. Figure 18 sums up the description in a sketch obtained from [15].

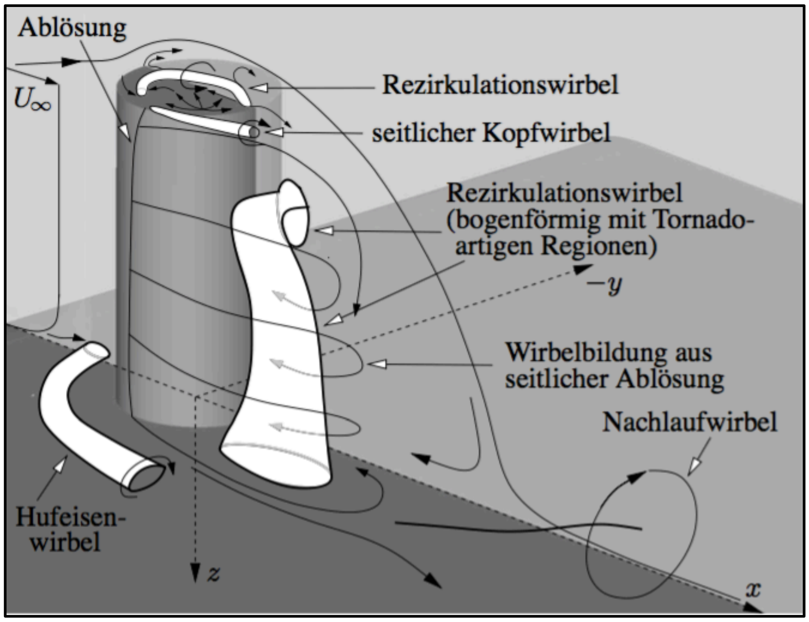

Ablösung
U_∞
Rezirkulationswirbel
seitlicher Kopfwirbel
Rezirkulationswirbel (bogenförmig mit Tornado-artigen Regionen)
$-y$
Wirbelbildung aus seitlicher Ablösung
Nachlaufwirbel
Hufeisen-wirbel
z
x

Figure 18 - Sketch of the vortexes generated [15]

Now that the short theoretical description has been made, the results obtained by slicing the domain are shown. The slices are done at the origin and at z=D, which is an important factor if the results want to be compared with literature ones or with experimental data.

4.3.1.1. y-normal Slice

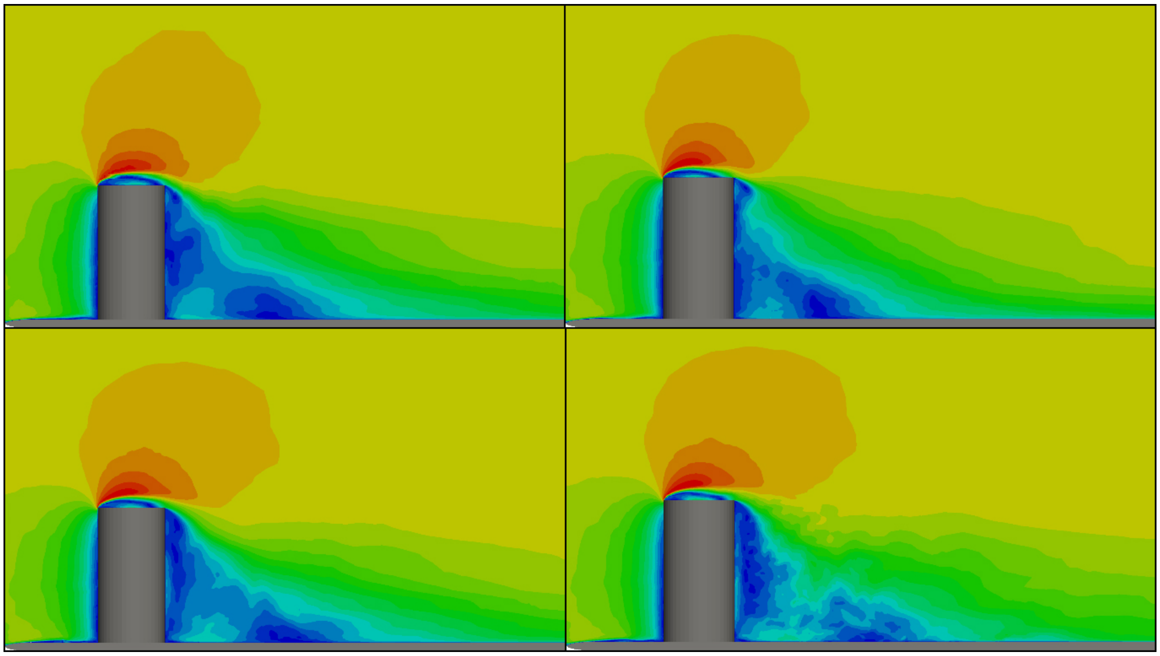


Figure 19 - UMean y-normal slice comparison

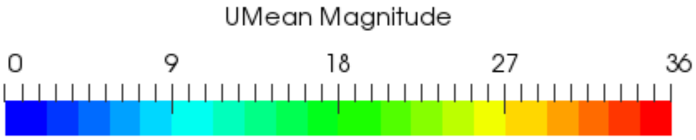


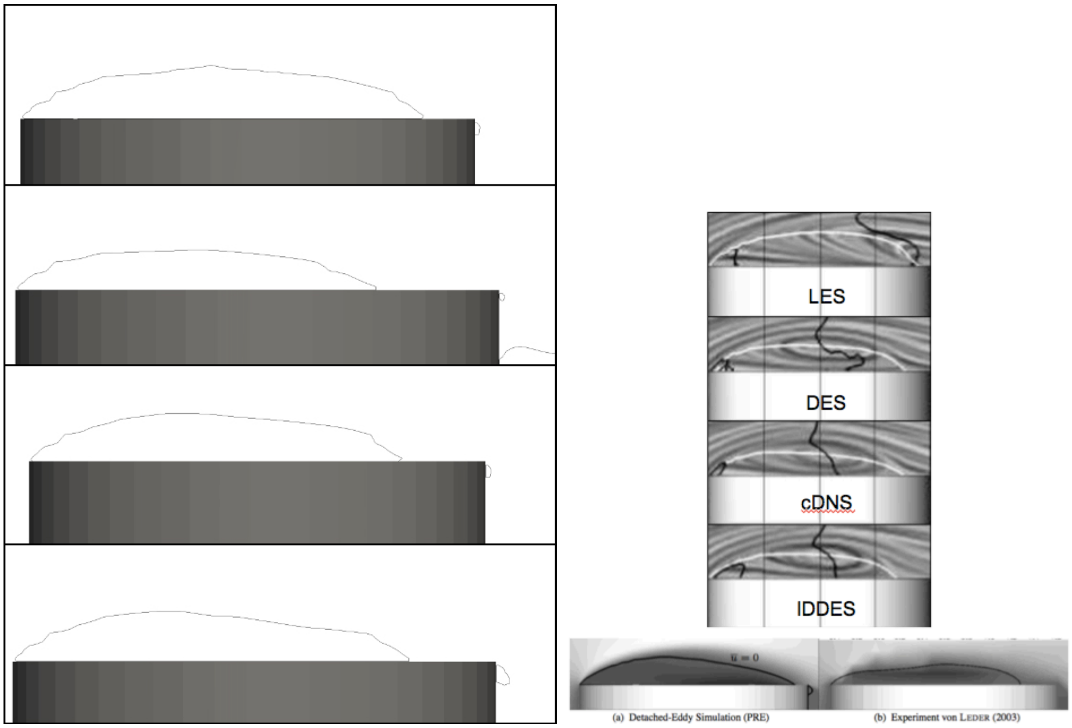
Figure 20 - UMean Magnitude colorbar

To start with, a good sign is that all of them look quite similar. The shapes have some differences but the main trend behind the pictures is similar, and this fact states that the results could be close to the real behavior of the flow, but it must be studied nonetheless, since they all could be wrong, which is highly unlikely but yet possible.

Velocities close to the surfaces, which are the cylinder and the round plate, are 0 or close to this value, as expected. Therefore, the lower velocities are placed close to surfaces, where the main physical phenomena occur. Far upside the cylinder, the velocities are almost equal or equal to the free stream velocity, which is also a good sign, and the biggest component is the one in the x-direction, which also makes sense since the free stream velocity is just defined in this coordinate.

The first recirculation vortex can be seen above the top of the cylinder, also as described in the short theoretical description provided prior to this specific section. This recirculation vortex reattaches later on before the end of the cylinder surface. The recirculation zone can also be appreciated behind the cylinder. Just above the cylinder, the velocities are negative, a fact that can be explained by the presence of adverse pressure gradients, causing the flow to detach from the cylinder surface. A bit above, as they reach a null value and far above they increase again far above to reach a final value close or equal to the free stream velocity one, a fact that makes sense.

The recirculation vortex above the cylinder reattaches faster at the Spalart-Allmaras IDDES simulation, followed by both kwSST simulations, which show a very similar behavior and Spalart-Allmaras DDES is the one that presents a latest reattachment. In order to compare them better, the isosurface $\bar{u} = 0$ has been plotted at the same slice, and the results can be seen at Figure 21.



(a) Detached-Eddy Simulation (PRE)

(b) Experiment von LEDER (2003)

Figure 21 - Reattachment of the recirculation vortex above the cylinder comparison [15]

As it can be seen in Figure 21, the reattachment point regarding the kwSST simulations is almost identical. Spalart-Allmaras DDES presents the latest reattachment point and Spalart-Allmaras IDDES the soonest one. But the comparison does not just end up here. The results have been also compared with the ones from experimental data, by the experiment conducted at [26] and the results from DES simulations performed by Octavian Frederich at [15]. The DES simulation predicted the reattachment point to be at the 94.16% of the diameter of the cylinder. That means that if the cylinder's diameter is 0.12 meters, the reattachment point would be placed at 0.113 meters, regarding the view provided by the slice. The experimental result shows a sooner reattachment point, placed at the 82.48% of the diameter of the cylinder.

Therefore, the DES simulation over predicted the location of this point. The best result would be then the kwSSTDDES one, since its reattachment point is placed at 81.15% of the cylinder's diameter, but also the kwSSTIDDES presents very good results regarding this specific value, since the predicted location is almost identical as in the kwSSTDDES case, whereas the Spalart-Allmaras simulations show very different results, either by over predicting the location, as in the Spalart-Allmaras DDES case (and also in the DES simulation), or by under predicting it by far as in the IDDES case. Another interesting trend to comment is that both IDDES simulations show a sooner reattachment point than the respective DDES ones.

Following with the analysis of the null velocity regions, the isosurface of $\bar{u} = 0$ both in a global and a z-normal slice view and the isoline of $\bar{u} = 0$ and $\bar{w} = 0$ have been represented. The isosurface corresponding to $\bar{u} = 0$ can be seen at Figure 22 in a perspective view and at Figure 35 in a z-normal slice.

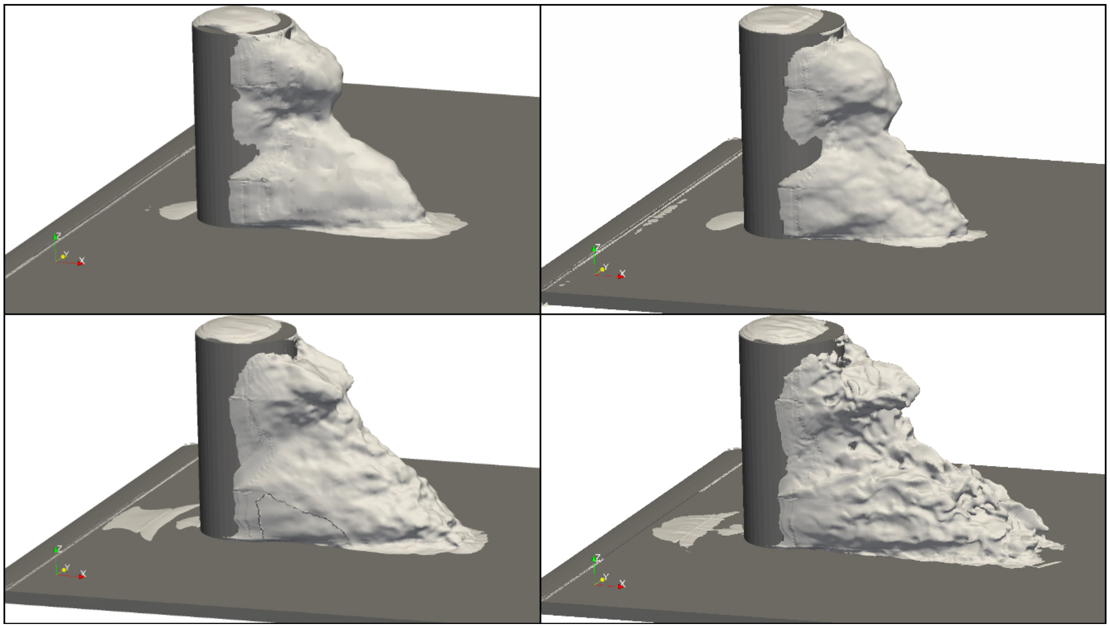


Figure 22 - Isosurface $\bar{u} = 0$ perspective view

As it can be seen at Figure 22, the prediction of the surfaces is more or less equal in all cases, since the shape of the surfaces is quite similar, but there are some differences that need to be pointed out. The first difference is the reattachment point as it has been described before. The trend is exactly the same as described in the previous paragraphs.

The main differences are presented between kwSST and Spalart-Allmaras cases. The isosurface is more extended close to the plate in the kwSST simulations. Both kwSST simulations predict an isosurface close or even reaching the trip wire, whereas Spalart-Allmaras ones do not predict the same and the isosurfaces are not even close of being extended up to that point.

The same happens behind the cylinder, where kwSST simulations predict the surface to be more extended along the x-axis. kwSSTIDDES shows some instabilities that could be ideally removed if the simulation keeps running and averaging longer.

Again, a comparison with the simulations performed by Octavian Frederich at [15] has been made, just to compare if the results obtained are close or not to the other simulations already performed, taking them as a valid reference.

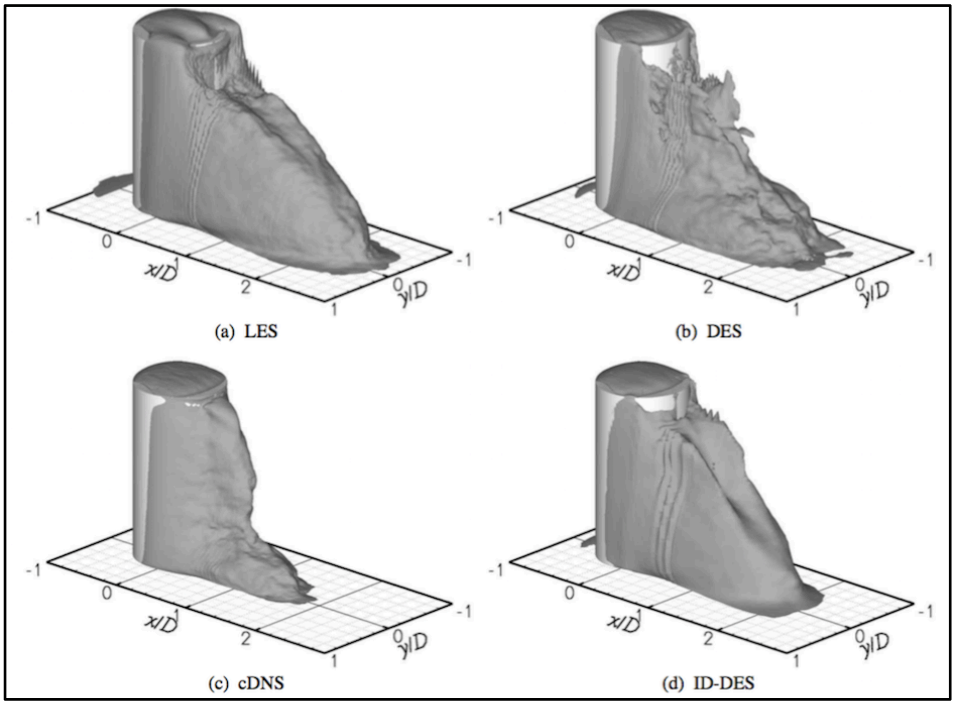


Figure 23 - Isosurface $\bar{u} = 0$ comparison [15]

The results shown at Figure 23 are the ones obtained for LES, DES, cDNS and IDDES simulations. They show that the region just behind the cylinder at the upper part does not present a $\bar{u} = 0$ zone, being the isosurface not present at that specific region whereas both Spalart-Allmaras simulations do as it is shown in Figure 22. Therefore, it can be said that Spalart-Allmaras simulations do not predict exactly the right behavior regarding this region. Therefore, it can be said that considering the results presented, kwSST simulations offer the best behavior prediction.

Going on with the analysis, a comparison between the isolines corresponding to $\bar{u} = 0$ and $\bar{w} = 0$ has been made, which can be seen at Figure 24, and a further comparison with the results on [15] as well, which can be seen at Figure 25. Since enough comparisons have been made regarding the isolines and isosurfaces corresponding to $\bar{u} = 0$, just the pictures of the isolines corresponding to $\bar{w} = 0$ for the same slice have been compared, in order to avoid redundancy since the results are the same.

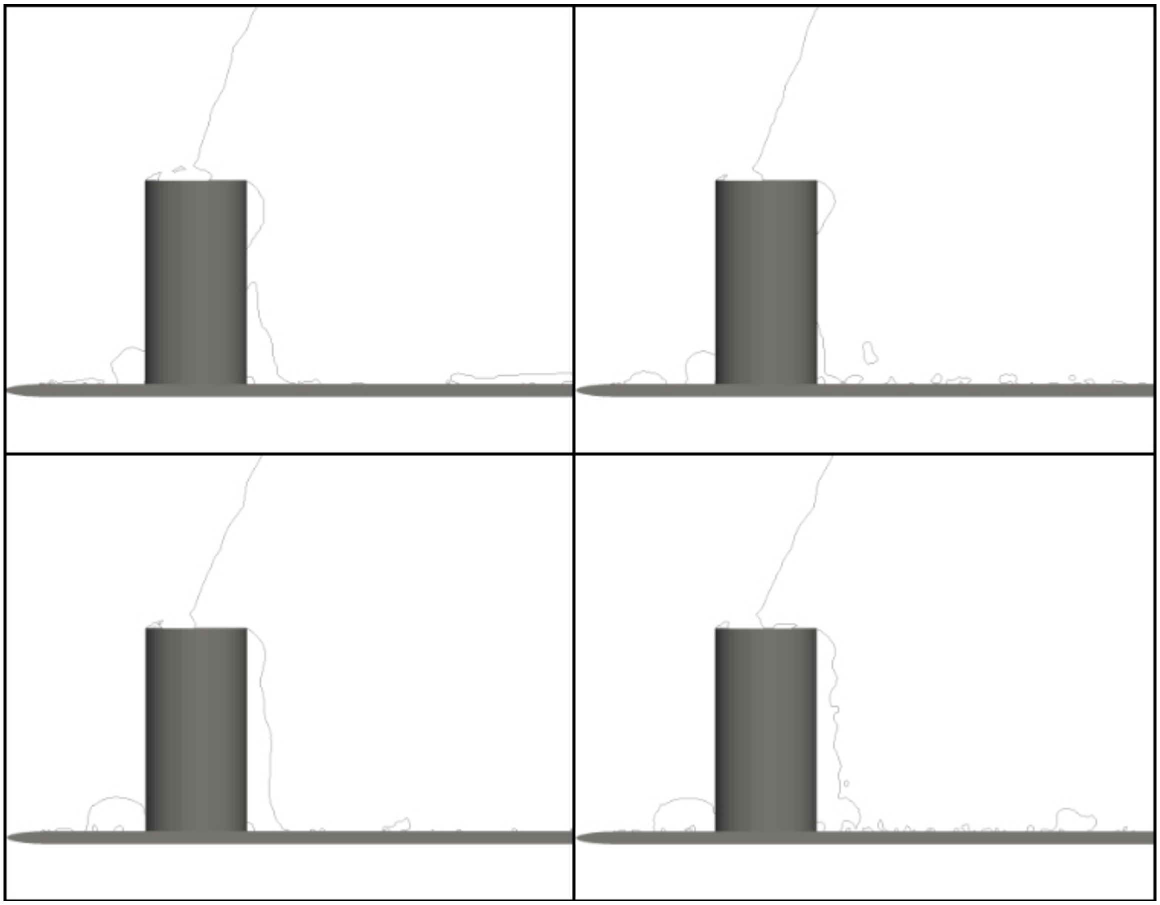


Figure 24 - Isolines $\bar{w} = 0$

In this case, the main trend is the same for all models, with some slight differences. Spalart-Allmaras models predict an isoline of $\bar{w} = 0$ closer to the surface, both in front of and behind the cylinder. The starting point at which the line begins over the cylinder is more or less at half of the cylinder in all cases except for the Spalart-Allmaras DDES, which predicts a starting point again further away compared to the other cases. The main differences are behind the cylinder surface, where both kwSST models show very similar results and again Spalart-Allmaras ones differ regarding the position of this isoline. Whether the results are reliable or not can be determined by comparing them with a valid reference. In this case, they will be compared with the shapes obtained by Octavian Frederich at [15], and the comparison can be seen at Figure 25.

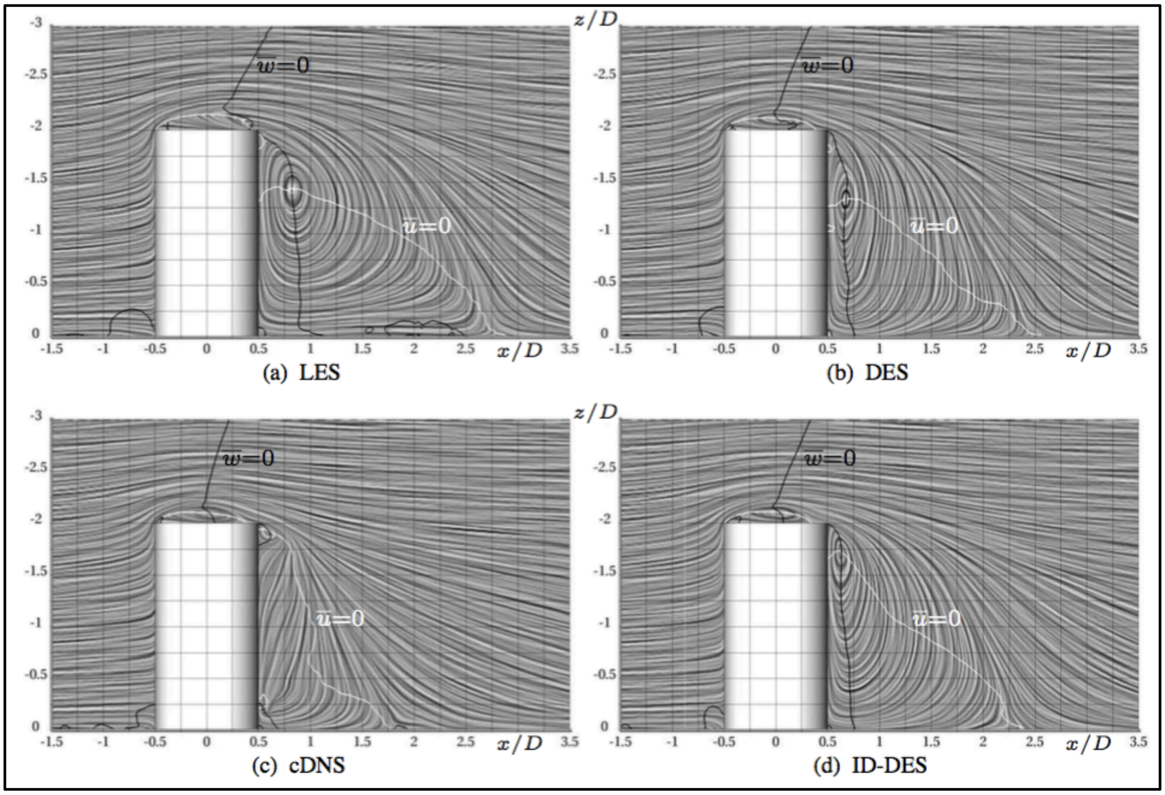


Figure 25 - Isolines u=0 and w=0 obtained by the simulations conducted at [15]

As it can be seen in Figure 25, the results differ significantly. IDDES, cDNS and DES predict the beginning of the isoline above the cylinder at more or less the half of it, just as the results obtained, which can be understandable as good results by looking at this specific comparison, whereas the LES results over predict the placement of this point, just as the Spalart-Allmaras DDES case.

All cases except for the cDNS one show an isoline a bit far away from the surface right behind it, something that can be seen at all the other cases and at both kwSST simulation results. Spalart-Allmaras cases would then present similar results to the cDNS case in this specific study, being the Spalart-Allmaras IDDES case the most similar one to the cDNS results for this isoline, but falling short in the prediction of the reattachment point, which is significantly lower in the Spalart-Allmaras IDDES simulation results.

In addition to all the comparisons made, a comparison between the quotients of the 3 components of the velocity field divided by the free stream velocity vs. the quotient of the z-axis and the diameter of the cylinder has been made. The results have also been compared to those offered by Octavian Frederich at [15], to determine whether they are reliable or not.

Following a straight line starting at the origin regarding the x and y coordinates but just above the cylinder surface regarding the z one, 10000 points have been considered from the surface of the cylinder up to 3 times the cylinder's diameter from this location, in order to determine the behavior of those magnitudes and compare them. The axes are dimensionless, since the velocities have been divided by the free stream velocity and the z axis coordinates by the diameter of the cylinder.

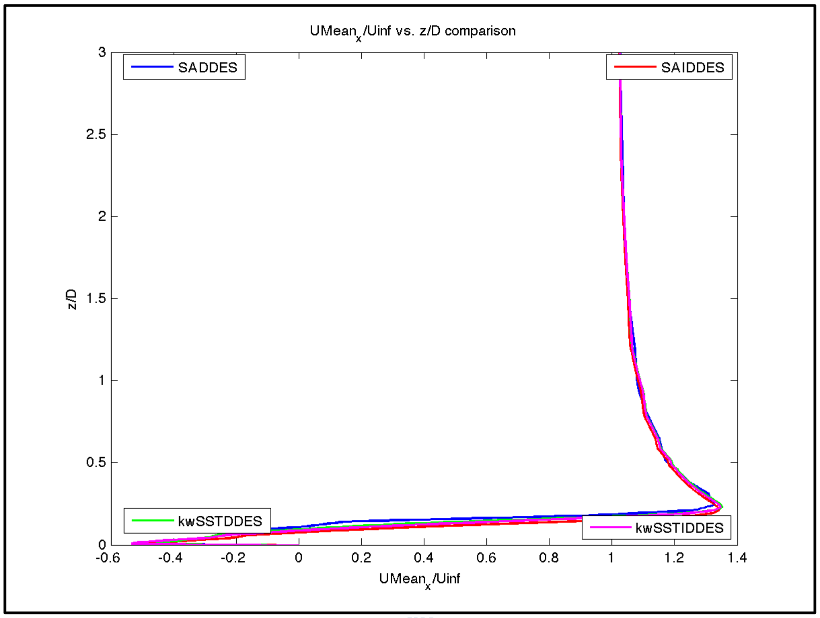


Figure 26 - $\frac{z}{D}$ vs. $\frac{UMean_x}{U_\infty}$ comparison

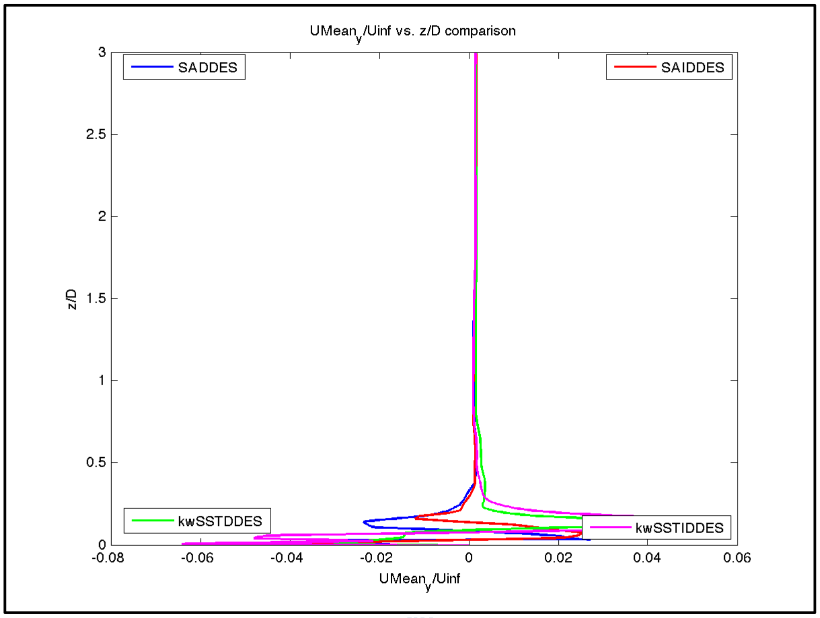


Figure 27 - $\frac{z}{D}$ vs. $\frac{UMean_y}{U_\infty}$ comparison

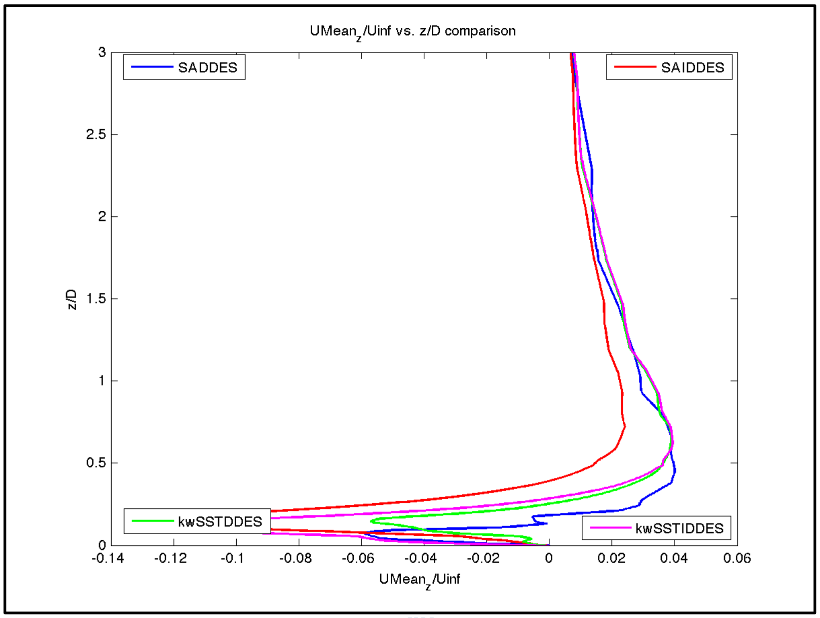


Figure 28 - $\frac{z}{D}$ vs. $\frac{UMean_z}{U_\infty}$ comparison

The most important plot is the first one, since the x component of the velocity is the most relevant one. The free stream velocity is defined just in this direction and therefore, the biggest and most important component of the velocity field is this one. That being said, the analysis can begin.

In the first plot, Figure 26, the first velocities right above the cylinder surface are negative in all simulations as expected, and the trend followed by all of them is almost identical, which is a good sign as well. As the z coordinate increases, the velocities face the isoline of $\bar{u} = 0$ that has been described in the previous analysis, and from that point on, the values are positive and begin to approach the value corresponding to the free stream velocity, which is also a good sign. When the z values are far away of the surface of the cylinder, the values of this velocity component tend to be the same as the free stream velocity, and therefore the plot shows a straight line close to 0, stating that the results make sense.

Regarding Figure 27, the behavior is a bit different right above the cylinder surface. All of them start again with negative magnitudes for the same reason as before, but then, as z increases, they behave differently. All trends reach a positive maximum value, then they retreat back and finally converge, but the way in which they do it is a bit different. Spalart-Allmaras simulations retreat back even to negative values again, which even though they do not reach significant negative values, is not a good sign, whereas both kwSST models just retreat back to values close to the final one, null, and then converge, a trend that is the expected and desired one. The reason behind is pretty straightforward. Once the z coordinate is far away from the surface, the only component present in the velocity should be the x coordinate, since the flow is expected to be approximately equal to the free stream velocity. Therefore, the final x values, when the z coordinate is far away from the surface, should be close to 0 or 0 in an ideal case. What that would mean is that the y component of the velocity at that point is small enough to be considered negligible, and that is what happens in the end for all simulations, but the kwSST ones present more reliable results.

The general trends seen at Figure 28 are more or less similar, but still present some differences regarding the behavior of the magnitudes before converging to the same value in the end. For the first time, both kwSST simulations present significant differences. The velocity magnitudes obtained in the kwSSTIDDES simulation are closer to the ones presented by the Spalart-Allmaras IDDES model, but just at the first y values. When the values begin to converge, they quickly become closer to the ones of the kwSSTDDES simulation. Spalart-Allmaras presents again an overprediction in the velocity values. In the end, they all tend to 0, which also makes sense for the same reason described before.

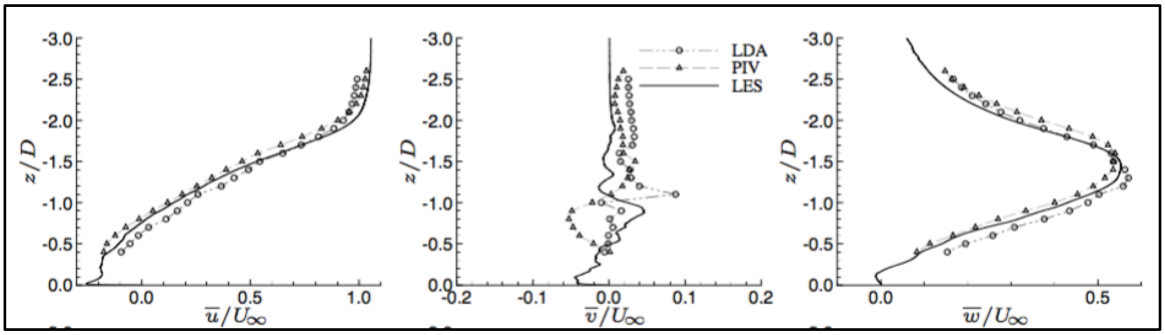


Figure 29 - $\frac{z}{D}$ vs. $\frac{UMean_x}{U_\infty}$, $\frac{UMean_y}{U_\infty}$ and $\frac{UMean_z}{U_\infty}$ comparison

After observing the results obtained, the best way to determine whether they offer good values or trends is by comparing them to the literature references. Again, Octavian Frederich displays the results he got at [15], a fact that allows a quick comparison.

As it can be seen in Figure 29, the trends are similar but not exactly the same, especially the height at which they converge to the final values. The main guidelines are respected, with some specific remarks. For example, all models overpredict the velocity in the x direction, but in the end they all converge to 1 as expected. The same happens on Figure 28, where the trend is similar but they do not exactly fit if comparing the y-axes. Therefore, it can be said that the behavior at the region above the cylinder could be improved either by adding a refinement zone or by adapting the meshing around this region to match the expected results.

4.3.1.2. z-normal Slice

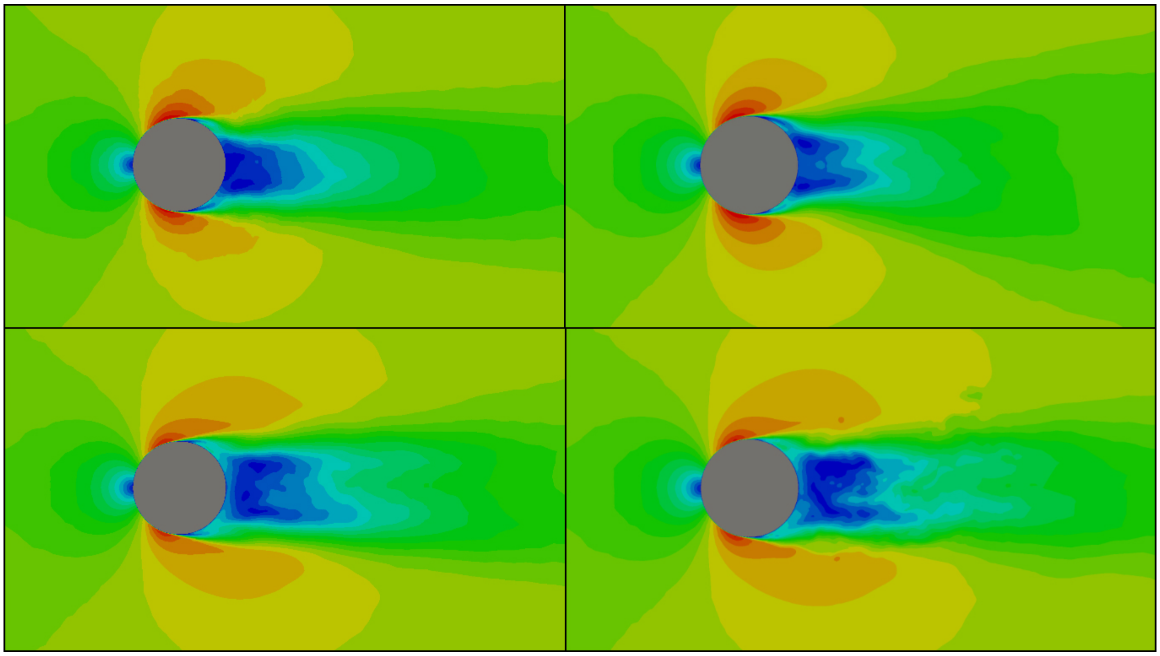


Figure 30 - UMean z-normal slice comparison

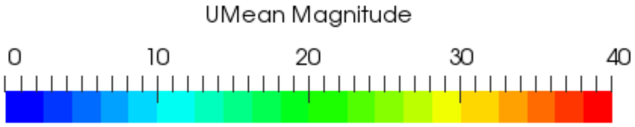


Figure 31 - UMean Magnitude colorbar

After considering several studies focused on y-normal slices, now it is the turn to change the point of view and analyze the results from a z-normal slice. As in the previous case, they present similar and almost symmetrical behaviors, and also the results show similar range values, which is good to know. When the free stream velocity faces the cylinder surface, the flow around it accelerates, until it faces an adverse pressure gradient, friction and external sources that slow down the flow until the separation is produced. The velocity is at its minimum value close to the separation point, at the recirculation zone behind the cylinder and at the stagnation point placed in front of the cylinder, which also makes sense since the flow cannot cross the surface and must disperse as it faces it.

The main differences in this case can be found between kwSST and Spalart-Allmaras simulations. The contours around the cylinder surface are more extended in both kwSST simulations than in the Spalart-Allmaras ones, meaning that the predicted velocities around the cylinder surface are bigger for the kwSST simulations. From this perspective it is almost impossible to distinguish any further differences between the plots and the behavior behind.

In order to see if there are further differences that have not been pointed out, a further analysis has been done, which consists on plotting a dimensionless magnitude obtained by dividing the x component of the velocity by the free stream velocity. This plot can therefore be compared with literature references as well, as seen in Figure 34.

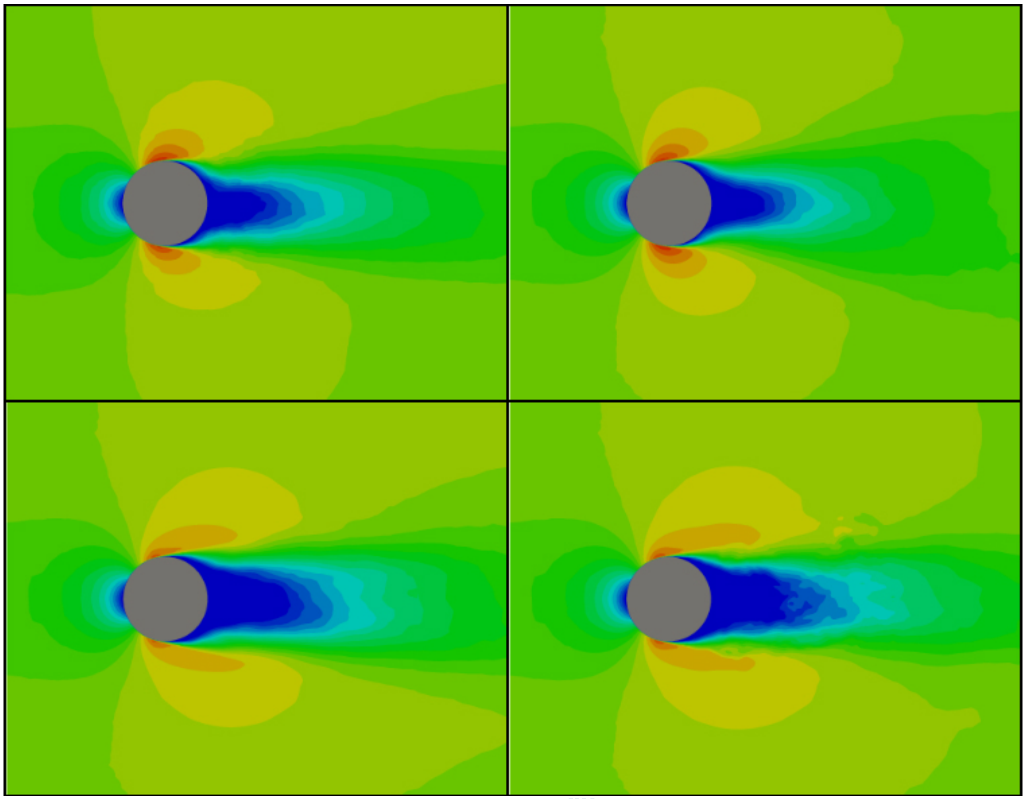


Figure 32 - Dimensionless $\frac{UMean_x}{U_\infty}$ comparison

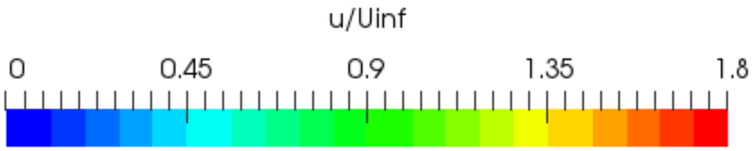


Figure 33 - $\frac{UMean_x}{U_\infty}$ colorbar

As it can be seen in Figure 32, the same trend is followed. Almost no differences present between both kwSST simulations respectively and the same happens with the Spalart-Allmaras ones. However, some differences can be seen between the global trend of the Spalart-Allmaras simulations and the kwSST ones. The recirculation zone is more extended in the kwSST simulations, and the velocities are higher around the cylinder surface as well.

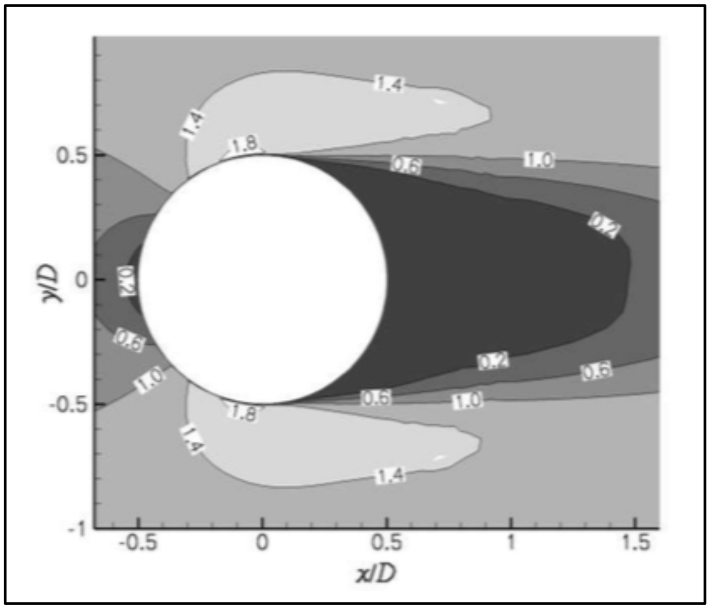


Figure 34 – Contour $\frac{UMean_x}{U_\infty}$ literature plot [15]

Figure 34 shows the results obtained at [15], and can be easily compared with the ones presented above. The contours are more similar to the kwSST simulations, presenting almost identical shapes and values. Therefore, it can be said that these models predict better the velocities around the cylinder surface than the Spalart-Allmaras ones.

To end up with this section, the isosurfaces for the x component of the velocity have been represented by looking parallel to a z-normal slice.

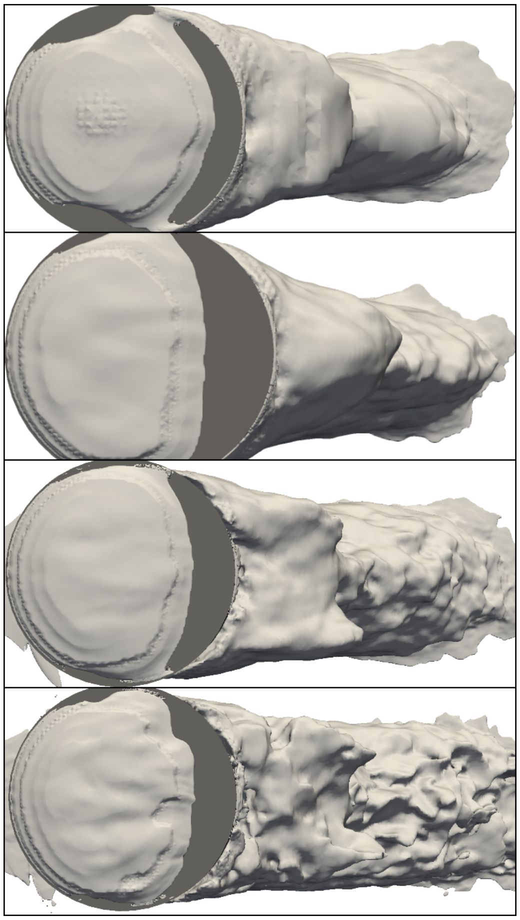


Figure 35 - Isosurface $\bar{u} = 0$ z-normal slice view

In this case, a z-normal slice has been done to observe the behavior of the isosurface by looking at it from above. The reattachment point behavior is the same as explained before. No big differences are presented between both kwSST models, which predict this point very well compared to the experimental results, whereas the Spalart-Allmaras simulation results present many differences, causing an over prediction in the DDES case and an under prediction in the IDDES case.

4.3.2. Pressure

4.3.2.1. y-normal Slice

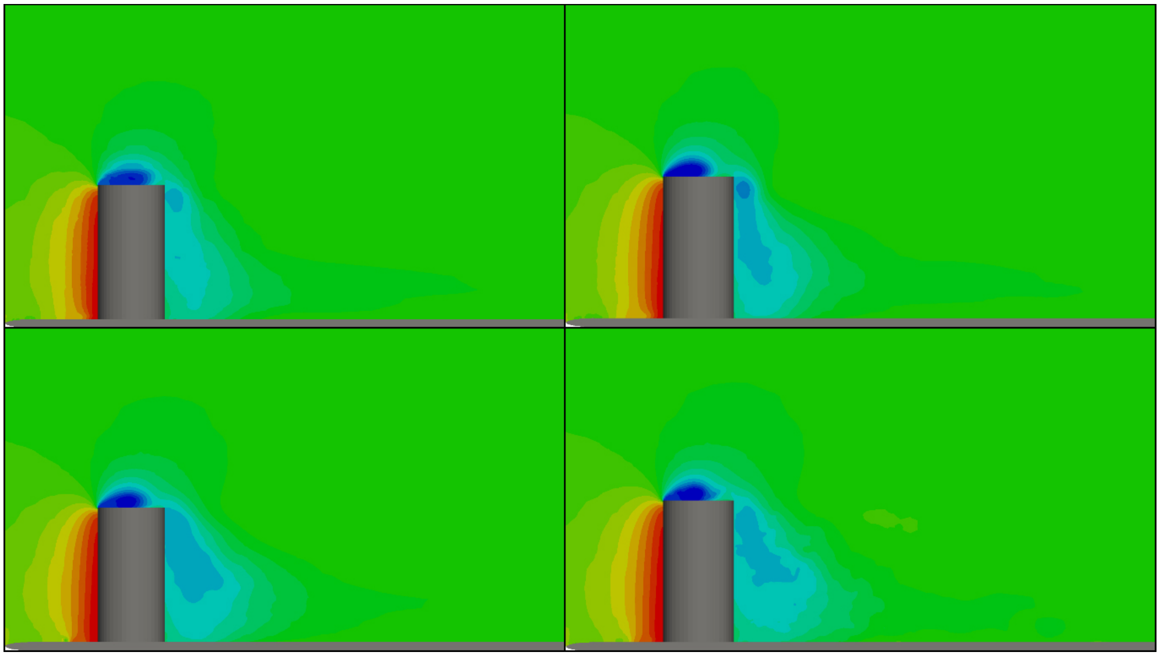


Figure 36 – Mean pressure y-normal slice comparison

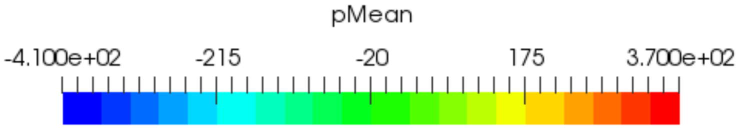


Figure 37 - Mean pressure colorbar

Pressure can be analyzed in parallel with velocity results. In front of the cylinder the behavior is the same for all models. Velocity values tend to 0 close to the walls and therefore, pressure values are at their maximum values, since that is the behavior expected at a stagnation point. Just over the cylinder, where velocities have maximum values, pressure values are at the lowest part of the range of values for the same reason.

Regarding the recirculation zone behind the cylinder, there are some differences that need to be pointed out. First of all, both Spalart-Allmaras simulations present different results for this region. Spalart-Allmaras DDES predicts higher pressures at the recirculation zone than Spalart-Allmaras IDDES. As seen in Figure 19, the velocity results for this region were lower for the DDES case than for the IDDES case, which makes sense after considering the pressure results.

kwSST simulations present almost no differences rather than a slightly different shape at the recirculation zone, where pressures are higher according to the kwSST DDES simulation than those obtained with the kwSST IDDES model but yet again, predicting different shapes if compared with the Spalart-Allmaras results, especially with the Spalart-Allmaras DDES case.

4.3.2.2. z-normal Slice

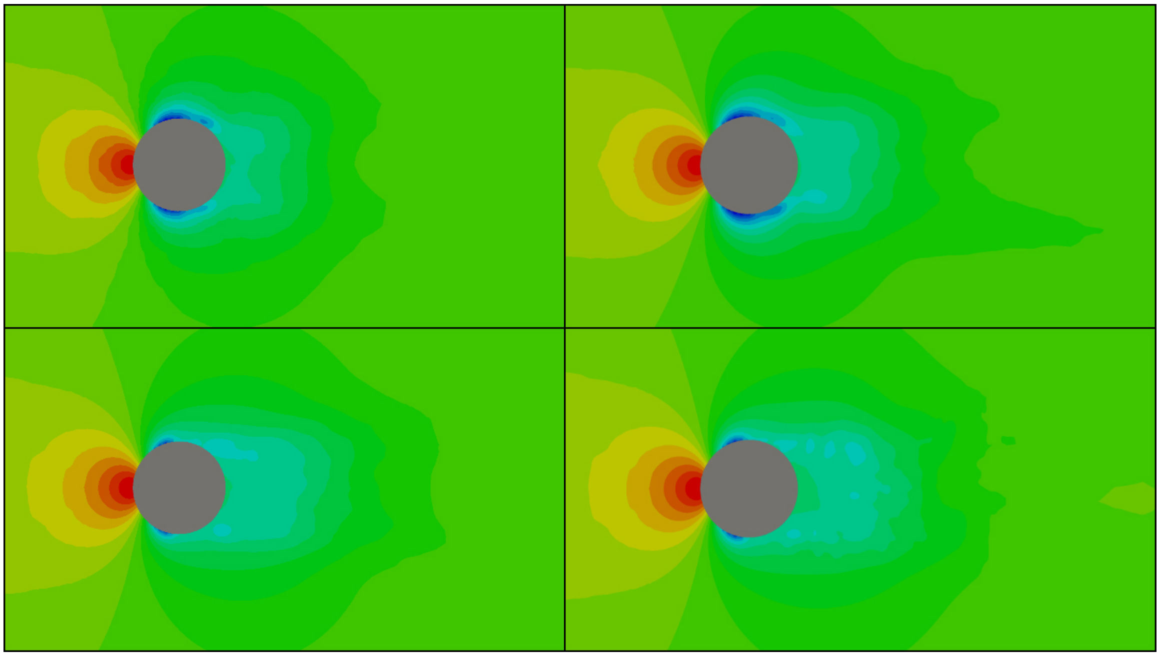


Figure 38 - Mean pressure z-normal slice comparison

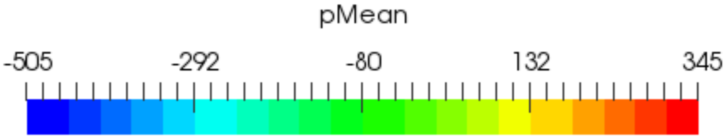


Figure 39 - Mean pressure colorbar

The results obtained after slicing the surface with a z-normal plane make sense after considering the pressure results just described. Close to the stagnation point in front of the cylinder, where the velocity tends to 0, the contours show a high-pressure zone, and where the velocity is higher due to the acceleration that suffers when facing the surface of the cylinder before detaching from it, the pressure is lower.

kwSST models present almost identical results but they differ slightly from Spalart-Allmaras ones yet again. Pressure is lower on these last cases close to the cylinder surface at the upper and lower parts, especially in the Spalart-Allmaras IDDES case.

4.3.3. nuT/nuTilda

4.3.3.1. y-normal Slice

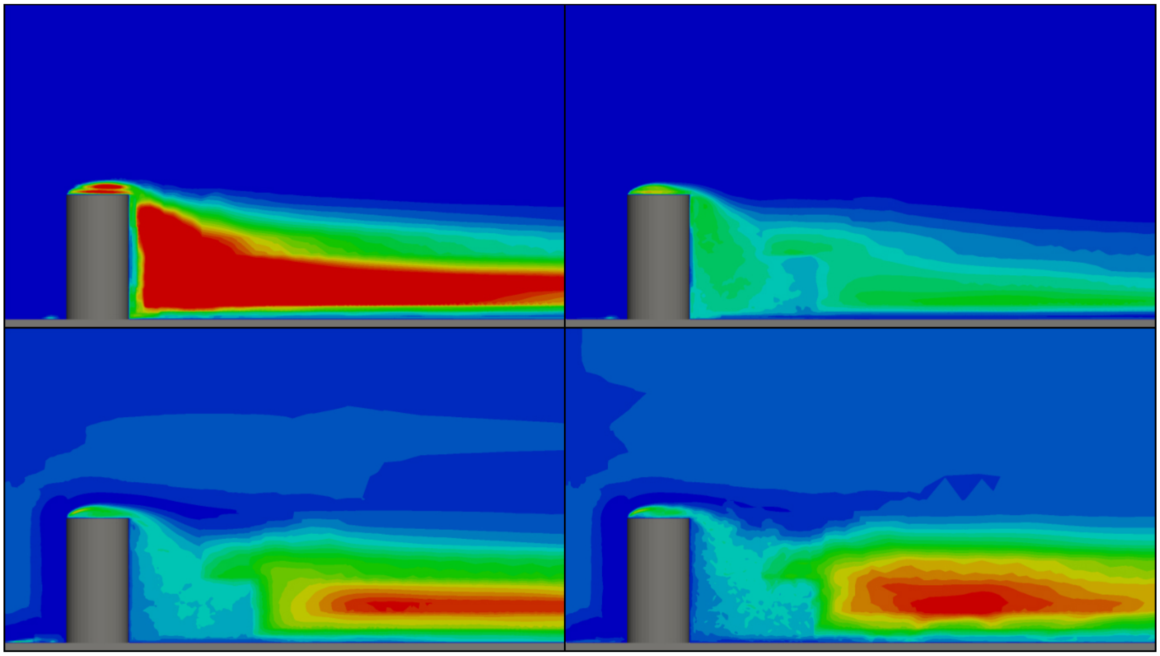


Figure 40 - Mean kinematic eddy viscosity y-normal slice comparison

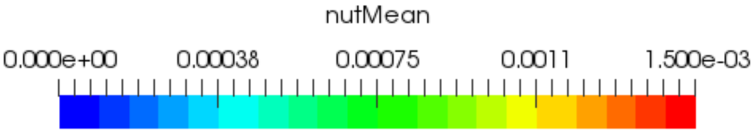


Figure 41 - nutMean colorbar

The modeling of the viscosity in vortexes is considered mostly in LES as an approach. For the determination of the whirl viscosity or fine structure viscosity ν_{SGS} only the speed measure becomes needed, because the linear measure is defined as a rule of the filter Δ.

As it can be seen in Figure 40, the differences regarding this magnitude for the 4 cases studied are significant, especially between both Spalart-Allmaras cases. Spalart-Allmaras DDES obtains higher values both above and behind the cylinder, whereas Spalart-Allmaras obtains lower values in both regions not only comparing with the SA DDES case but also with the kwSST simulations. Not many differences can be seen between both kwSST simulations. The viscosity results in the region above the cylinder are lower in the kwSSTDDES than in the kwSSTIDDES one, and the results at the region behind the cylinder also differ slightly, being the region with highest values smaller in the kwSSTDDES case than in the kwSSTIDDES one and slightly different in shape.

4.3.3.2. z-normal Slice

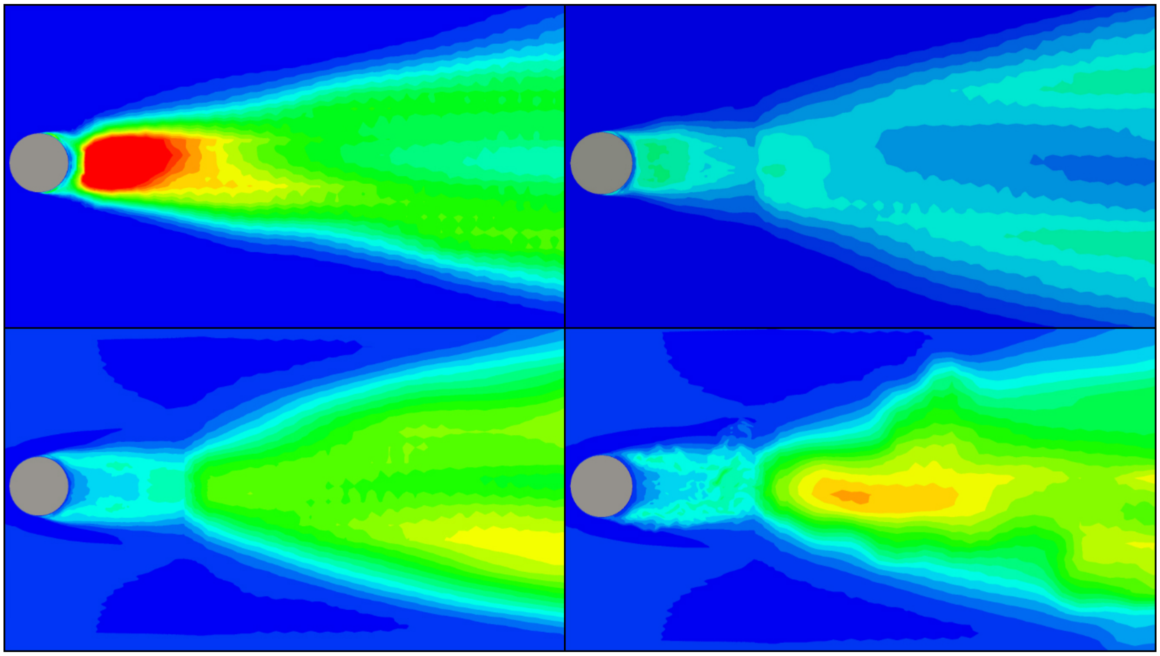


Figure 42 - Mean kinematic eddy viscosity z-normal slice comparison

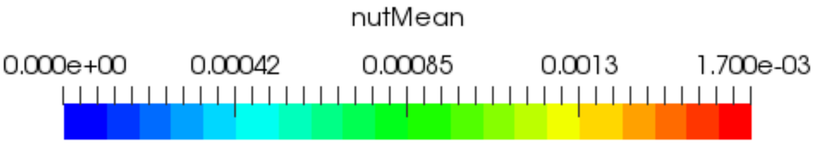


Figure 43 - nutMean colorbar

The same trend can be seen in Figure 42. Spalart-Allmaras DDES results show the highest values of viscosity of all four simulations, and the region with highest values is right behind the cylinder, a trend that is not followed by the other simulations. Spalart-Allmaras IDDES model shows lower values in comparison with not only the SA DDES case but with both kwSST cases as well, following then the behavior observed in the y-normal slices described above.

kwSST simulations show similar region contours but with some differences in where the regions with highest values are located. kwSSTDDES results show a more or less homogeneus value at the wake of the cylinder, whereas kwSSTIDDES shows a higher-than-average values region a bit downstream, behind the cylinder as well.

Therefore, yet again, SADDES is the case that shows the highest values, whereas SAIDDES is the one showing the lowest ones in average, but in this occasion both kwSST simulations present yet slight but greater than before differences between the results obtained for each case.

4.3.4. k

4.3.4.1. y-normal Slice

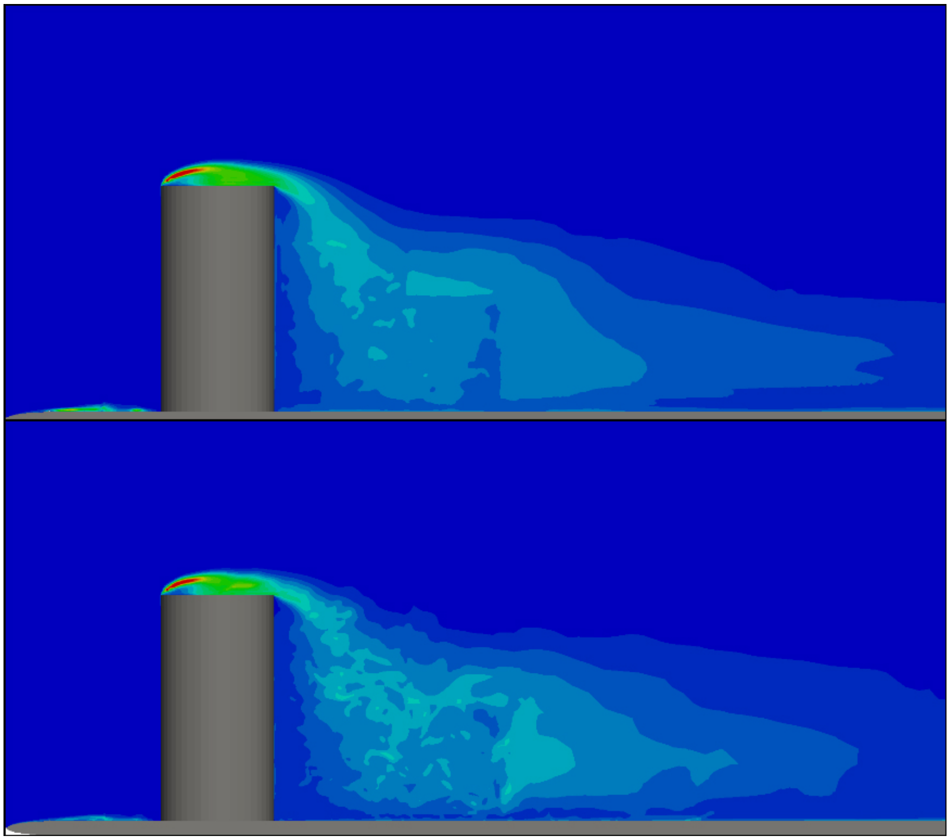


Figure 44 - Mean turbulence kinetic energy y-normal slice comparison

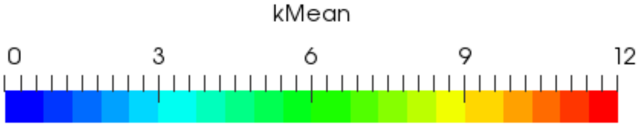


Figure 45 - kMean colorbar

Turbulence kinetic energy can be defined as the mean kinetic energy per unit mass associated with eddies in turbulent flow. It can be produced by fluid shear or friction, for example. In this case, also following the main trend seen up to now, the results between kwSST simulations do not differ a lot between them. In this specific case, the main differences can be seen behind the cylinder and just in front of it, right after the trip wire. kwSSTDDES results show greater values (and therefore, greater energy afafter the wire and in the recirculation vortex above the cylinder. Right behind the cylinder, the behavior shown by the kwSSTIDDES results looks to be more chaotic and unstable than the one observed in the kwSSTDDES case, perhaps because the case could be run longer.

4.3.4.2. z-normal Slice

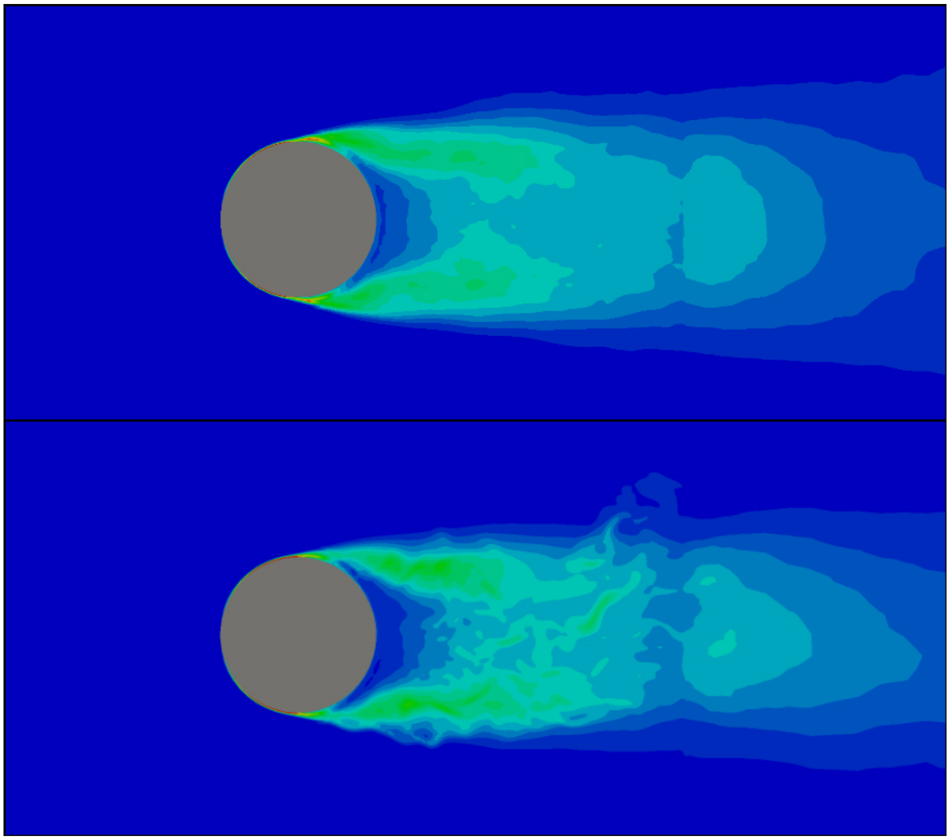


Figure 46 - Mean turbulence kinetic energy z-normal slice comparison

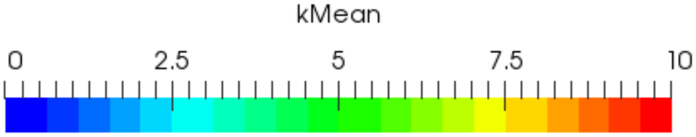


Figure 47 - kMean colorbar

The same happens in this case. The behavior far away from the walls is identical and the main differences can be seen close to the detachment point, where kwSSTDDES results show bigger turbulence kinetic energy values than kwSSTIDDES.

The shape behind the cylinder surface is quite similar, presenting some differences in the closest region to the surface of the cylinder, where again the kwSSTDDES case shows slightly greater turbulence kinetic energy values than the kwSSTIDDES simulation, whose behavior is a bit less stable as well for the same reasons as the ones exposed in the previous slice view.

4.3.5. omega

4.3.5.1. y-normal Slice

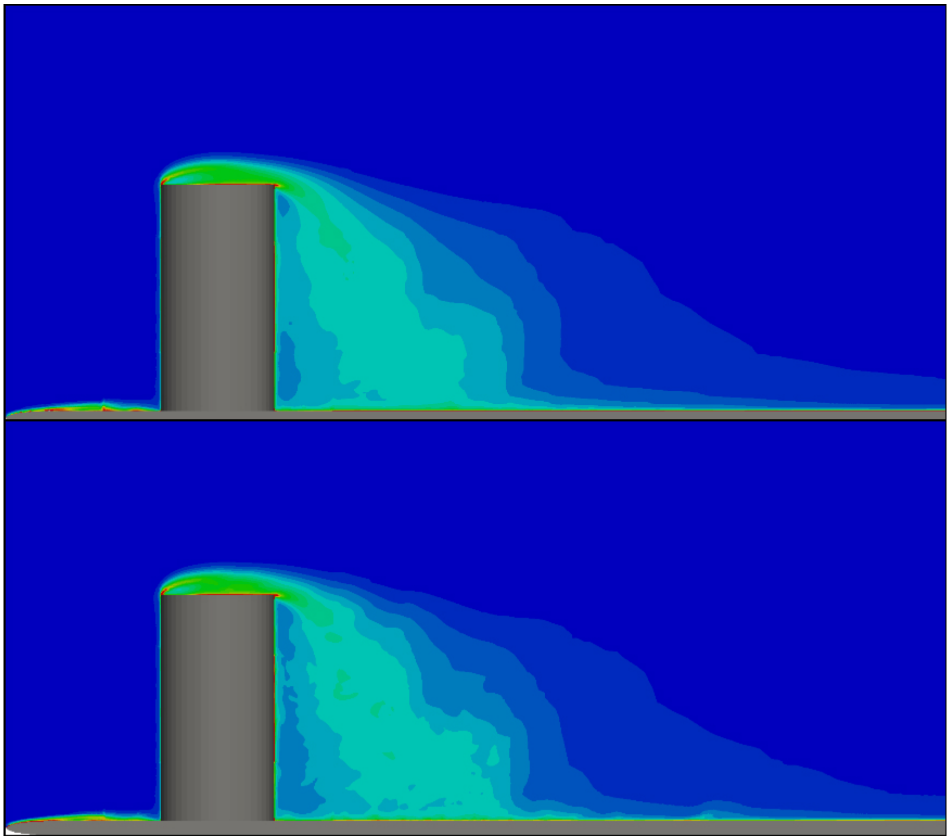


Figure 48 - Mean turbulence specific dissipation y-normal slice comparison

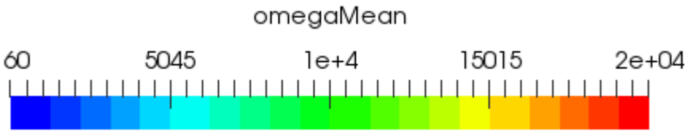


Figure 49 - omegaMean colorbar

The specific dissipation is the variable that determines the scale of the turbulence. The differences between the results obtained are difficult to spot, but mainly focused on the region close to the trip wire and at the recirculation zone behind the cylinder as well.

Regarding the first region, the turbulence scale is greater in the kwSSTDDES than in the kwSSTIDDES simulation, and the same happens at the recirculation zone behind the cylinder.

4.3.5.2. z-normal Slice

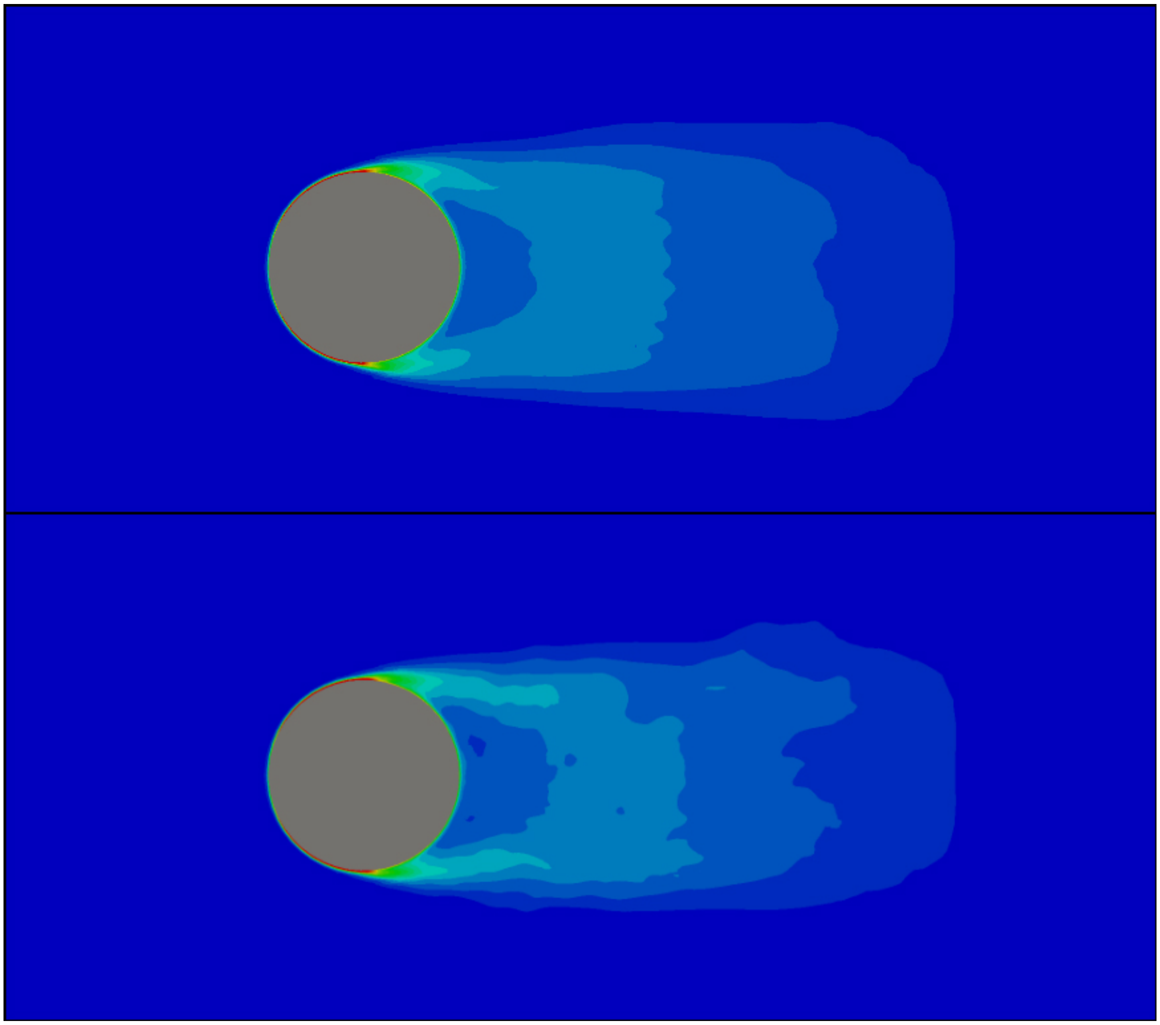


Figure 50 - Mean turbulence specific dissipation z-normal slice comparison

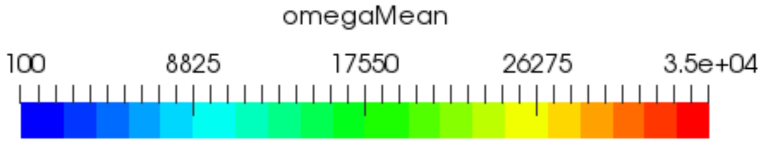


Figure 51 - omegaMean colorbar

Both result pictures look quite similar as well between simulations, which is also a good sign. The main difference can be seen at the values presented right behind the detachment point, where kwSSTDDES predicts that a contour line less prolonged than kwSSTIDDES, which means that the turbulence scale is lower in the first case regarding this region.

4.3.6. DESModelRegions

In this section, a field oscillating between 0 and 1 is analyzed. As it has been said in the theoretical section of this document, the models used are hybrid, and switch between LES and RANS regions depending on where the cell is placed, if close or far away from the wall. 0 means that the region is modeled as a LES region, and alternatively, 1 means that the region is modeled as a RANS region. Two different views have been considered, in order to give the best explanation possible to where the transition is done and how it is done.

4.3.6.1. y-normal Slice

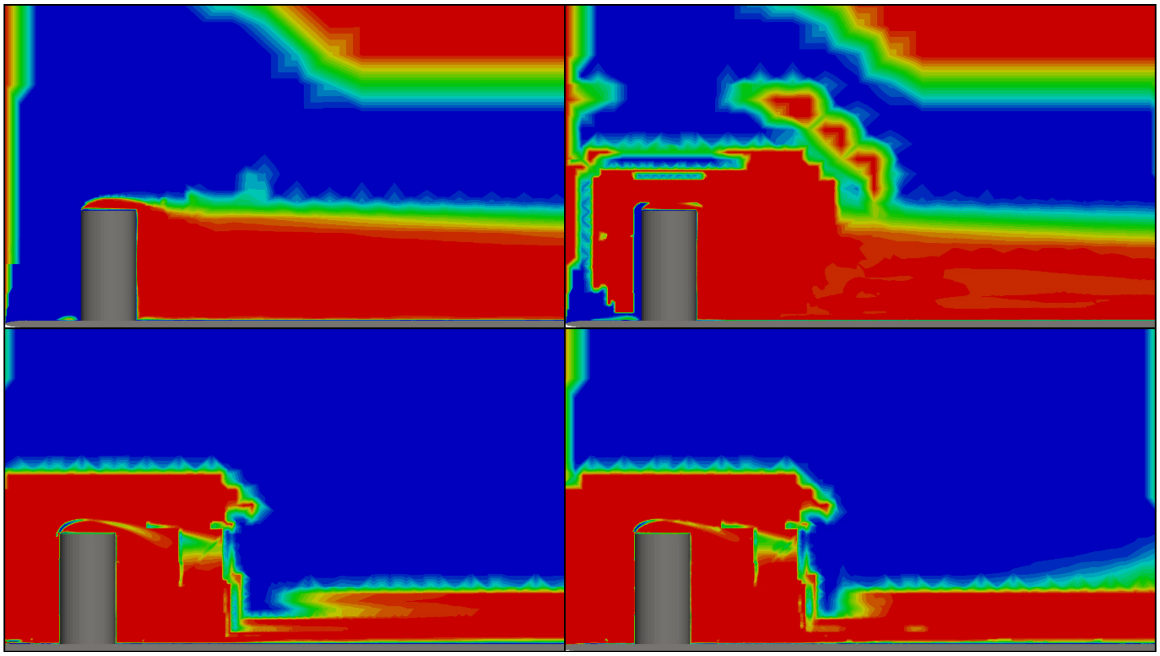


Figure 52 - Mean DESModelRegions y-normal slice comparison

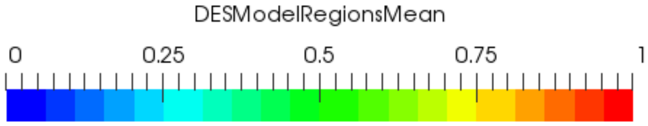


Figure 53 - DESModelRegionsMean colorbar

As it can be seen in Figure 52, there are many differences depending on the method used to simulate. Spalart-Allmaras presents the biggest differences between both models, whereas kwSST simulations model the regions almost identically. The crucial zones to be considered as RANS are the regions close to the walls and the recirculation zone, where eddies are generated. Spalart-Allmaras DDES models the domain even worse than the IDDES case, since the region in front of the cylinder is not even considered as RANS.

The differences between both kwSST simulations are that kwSSTIDDES predicts small regions close to the cylinder as LES whereas kwSSTDDES does not manage to do it, and therefore kwSSTIDDES manages to improve the results of its predecessor in this case.

4.3.6.2. z-normal Slice

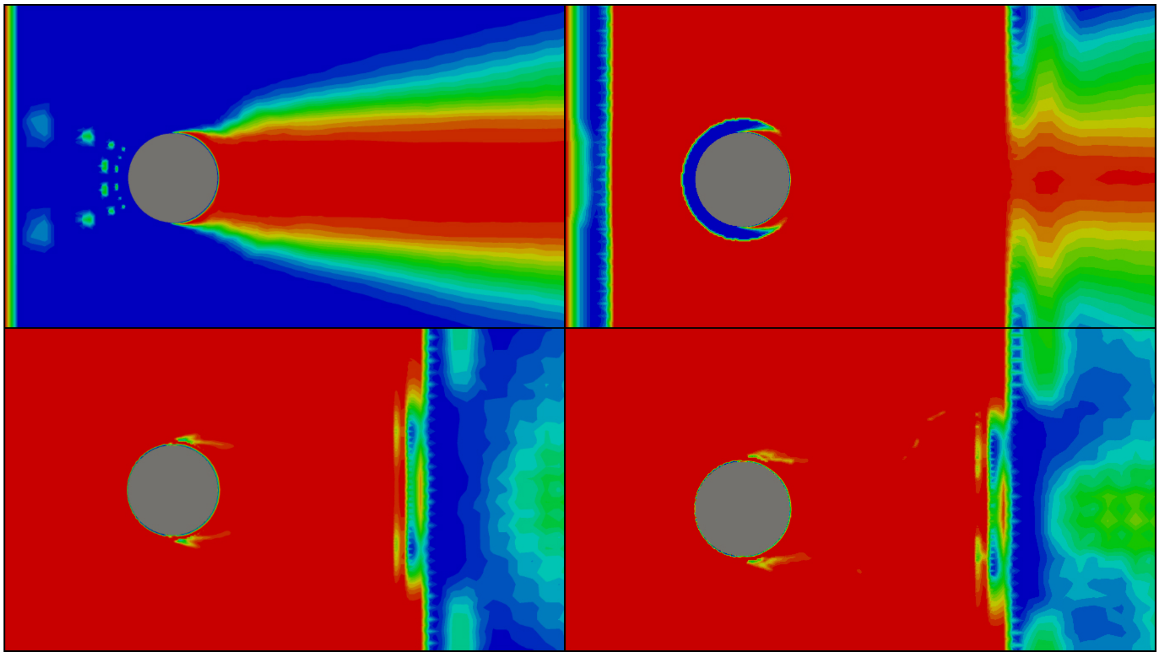


Figure 54 - Mean DESModelRegions z-normal slice comparison

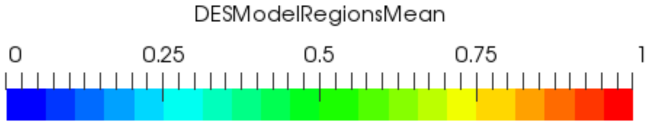


Figure 55 - DESModelRegionsMean colorbar

The same happens in this case. Spalart-Allmaras DDES behaves differently as the other cases. After double-checking the meshing parameters, everything seems to be set in the same way, and therefore it is the model that behaves in the most different way. Spalart-Allmaras IDDES focuses on the zone behind the cylinder and the region where the refinement zone is placed, whereas kwSST models focus less in the region behind the cylinder and more in the region just in front of it, where Spalart-Allmaras IDDES models as a LES region.

Yet again, in the kwSSTIDDES simulation, if the region above the cylinder surface is zoomed, it can be seen that in this last case comparing to the kwSSTDDES one, some regions close to the surface are modeled as LES instead of RANS, which is the same trend followed at the y-normal slice comparison shown above, and states that the kwSSTIDDES performs better at those regions while modeling.

4.3.7. y+

4.3.7.1. y-normal Slice

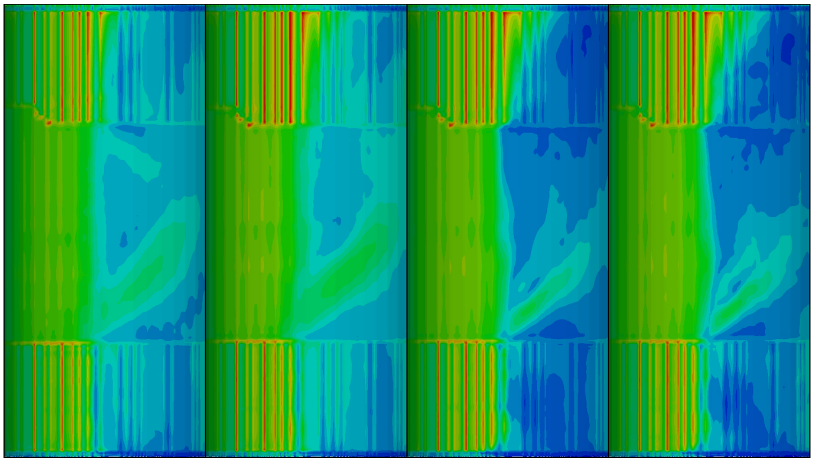


Figure 56 – Mean yPlusLES y-normal slice comparison

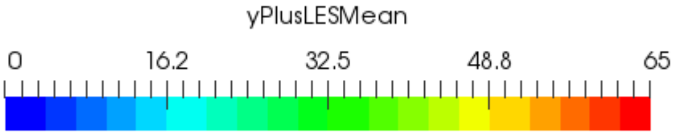


Figure 57 - yPlusLESMean colorbar

The main differences presented regarding this specific study can be spotted between the global trend of Spalart-Allmaras and the one of kwSST simulations, since between Spalart-Allmaras DDES and Spalart-Allmaras IDDES no significant differences are presented, and the same between both kwSST cases.

Each one of the two types mentioned before has its own y+ recommended range, and therefore, they need to be analyzed individually since a global comparison cannot be done due to the fact that the near-wall treatment is not the same in both cases.

According to literature resources, to [19] to be more specific, the best range regarding the yPlus values is either close to 1 or bigger/equal than 30, whereas the recommended range for the yPlus values regarding the kwSST simulations is $11 < y+ < 300$. Therefore, the value for kwSST should be smaller in the kwSST cases than in the Spalart-Allmaras ones, which is what can be seen in Figure 56. However, that is not enough. The mean values and the whole ranges have been computed to see if the conditions above are accomplished or not, and the reasons why the simulation domains could be improved. The averaging has been done by taking into account the average value provided by the yPlus utility at every single time step, and the results can be seen at Figure 58.

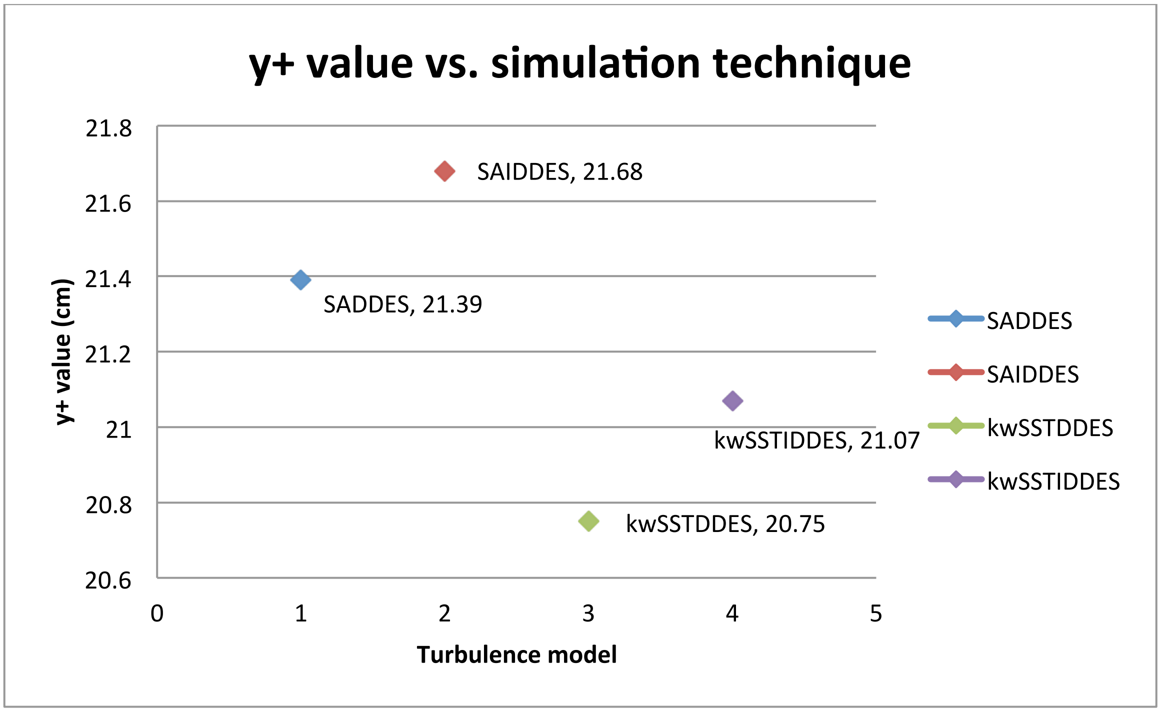


Figure 58 – Average yPlus vs. model

Figure 58 pretty much sums up the improvements that could be made by displaying the y+ average values for each simulation. All simulations present y+ values close to 21, which means that all of them present a similar meshing, which is not always appropriate, especially when several techniques require different settings. kwSST simulations present lower values than Spalart-Allmaras ones, and they fall in the appropriate range, which is good to know, and maybe that is the reason why a good near-wall treatment and results can be appreciated on these simulations. Spalart-Allmaras simulations, however, fall short in being on range, since they should present values higher than 30, and that could be one of the reasons why the near-wall results and treatment are not as good as in the kwSST simulations.

The meshing technique used in all simulations is the same. They all come from the same kind of division and blockMesh design, with a refinement zone placed close to the cylinder surface, where the main phenomena appear, and since the snappyHexMesh utility is an automatic tool, not many changes could be done, rather than presenting a slightly coarser mesh with less cells or more distributed so that the y+ values rise up and then the results could be improved. According to literature studies [19], the y+ average values obtained fall in the range of 10-15, therefore it can be said that the results obtained for the kwSST simulations are not bad at all.

4.3.7.2. z-normal Slice

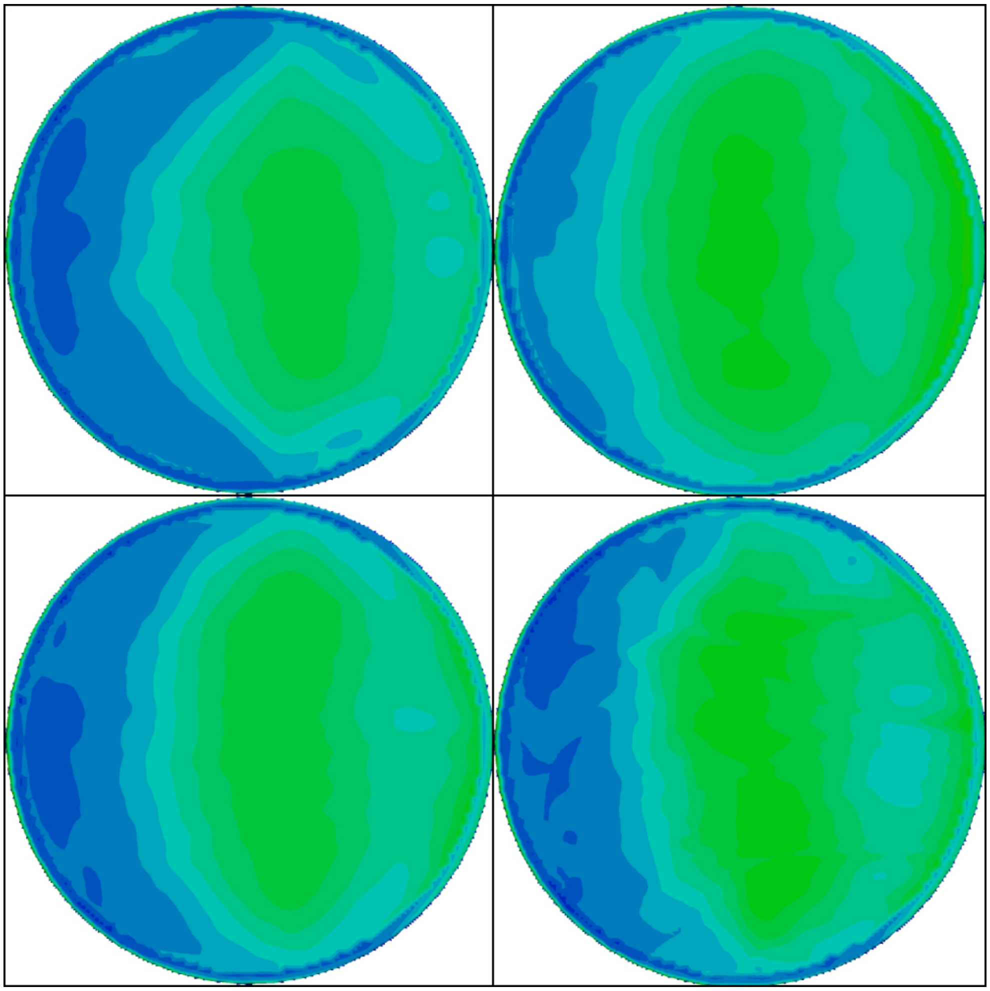


Figure 59 - Mean yPlusLES z-normal slice comparison

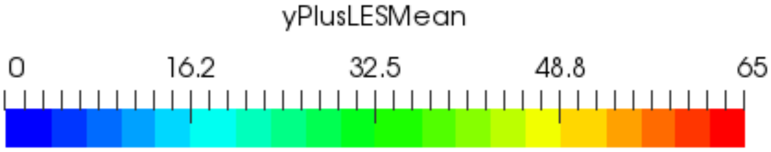


Figure 60 - yPlusLESMean colorbar

Not much can be added here, rather than just displaying where the zones with maximum and minimum values are.

All of them follow similar trends, placing the maximum zones close to half of the cylinder whereas Spalart-Allmaras DDES presents again slightly different results locating the zone with the maximum values a bit further away in the x direction.

4.3.8. Cd and Cl vs. time plots

On this section, the drag coefficient plots vs. time for the final configurations of each technique are analyzed. In this case, they present more similitudes between the same kinds of techniques, which makes the analysis a bit easier.

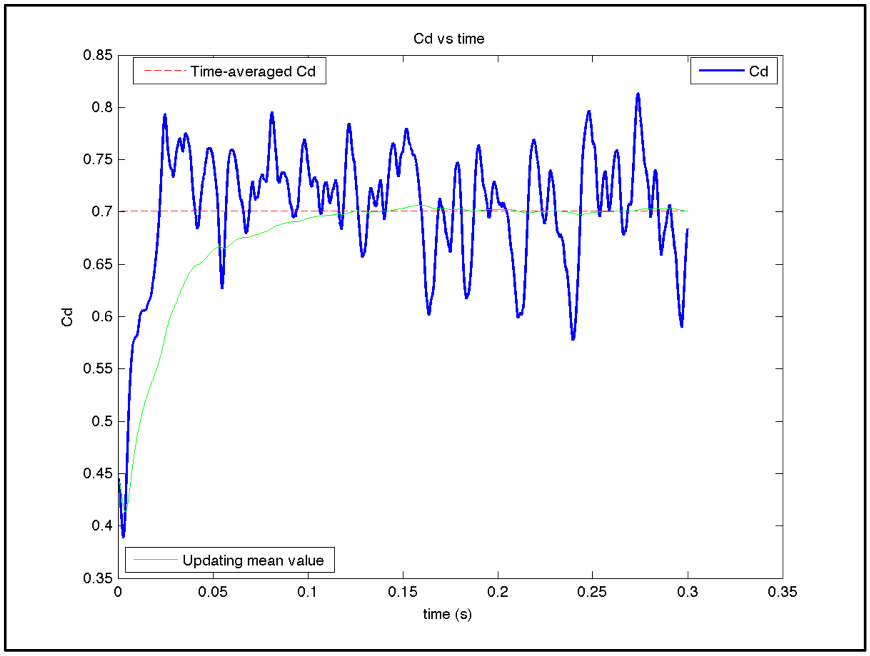


Figure 61 - cd x-direction vs. time SADDES

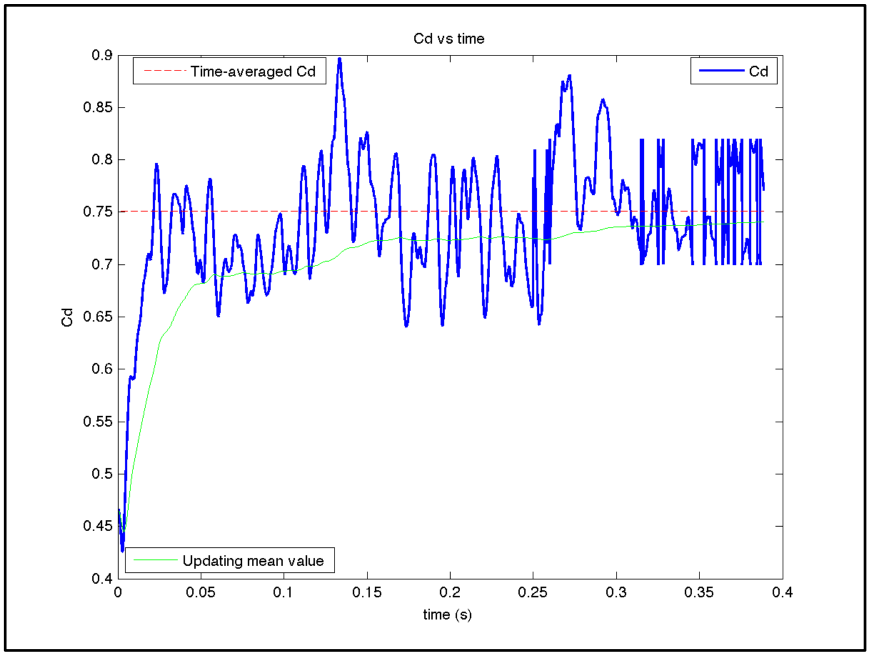


Figure 62 - cd x-direction vs. time SAIDDES

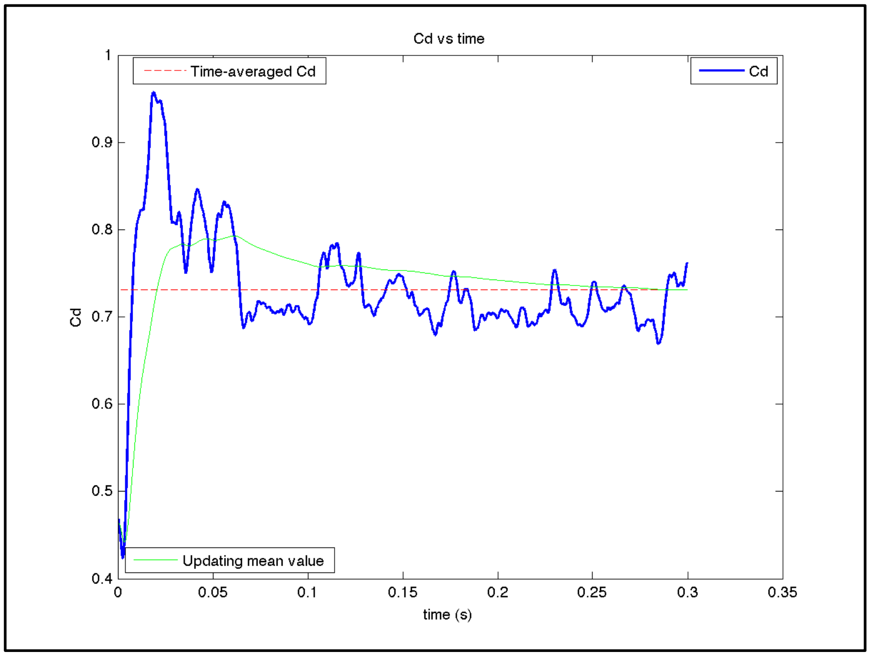


Figure 63 - cd x-direction vs. time kwSSTDDES

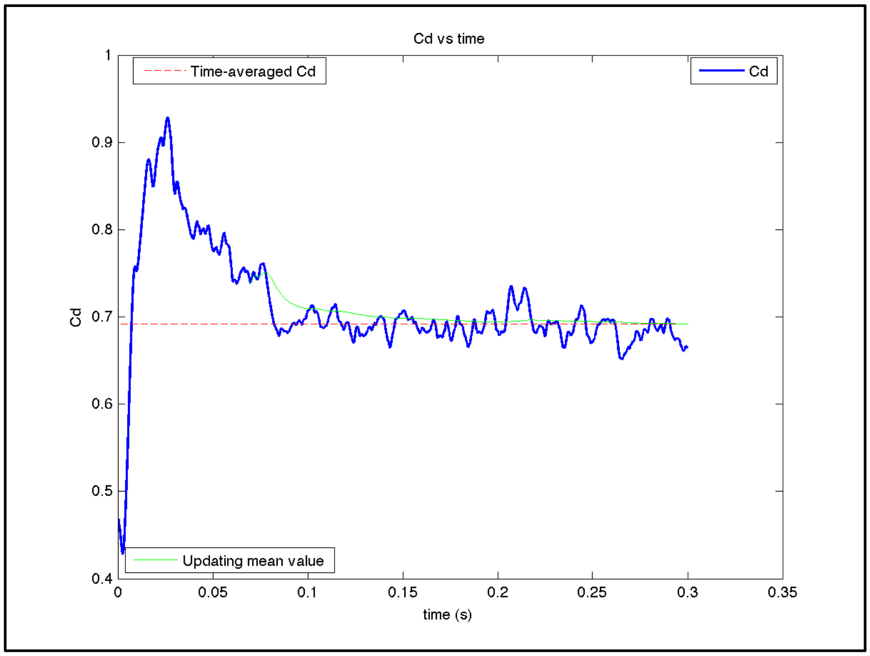


Figure 64 - cd x-direction vs. time kwSSTIDDES

The main difference between kwSST and Spalart-Allmaras when comparing these plots is the oscillations that both techniques present. While kwSST simulations present moderate oscillations around a main value that can be easily seen, Spalart-Allmaras drag coefficient values oscillate constantly and in a more severe way, presenting a difference between the maximum and the minimum value close to 0.2 units, which is significant enough to be mentioned.

Therefore, kwSST simulations reach easily a more or less steady and converged value, whereas Spalart-Allmaras results keep oscillating even after 0.4 physical seconds of time. The time that the flow needs to travel from the inlet to the outlet of the domain is more or less 0.05 seconds.

It is always important to keep that value in mind when analyzing the plots, especially if there is a strange behavior before the completion of that time that should be excluded from the averaging. For example, all simulations present a peak in the cd value close to 0.025 seconds, which can be related to a physical factor, rather than just a coincidence or a reference point.

All simulations present a similar initial value, and the reason is that they all come from the same starting point, which is the mapping of the fields provided by the Steady-State RANS Simulation. The value even goes lower at first, starting to grow quickly afterwards in all cases and reaching the maximum value at the point mentioned above. The averaging has been made in all cases starting from a point late enough to be considered reliable.

Furthermore, the lift coefficient has also been studied for the 4 simulations, obtaining very similar results.

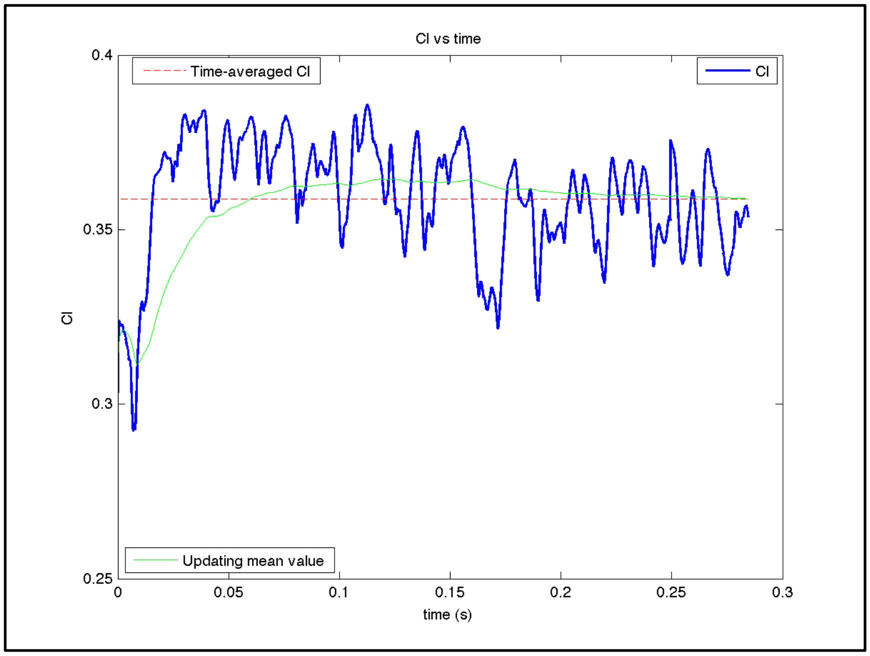


Figure 65 - cl x-direction vs. time SADDES

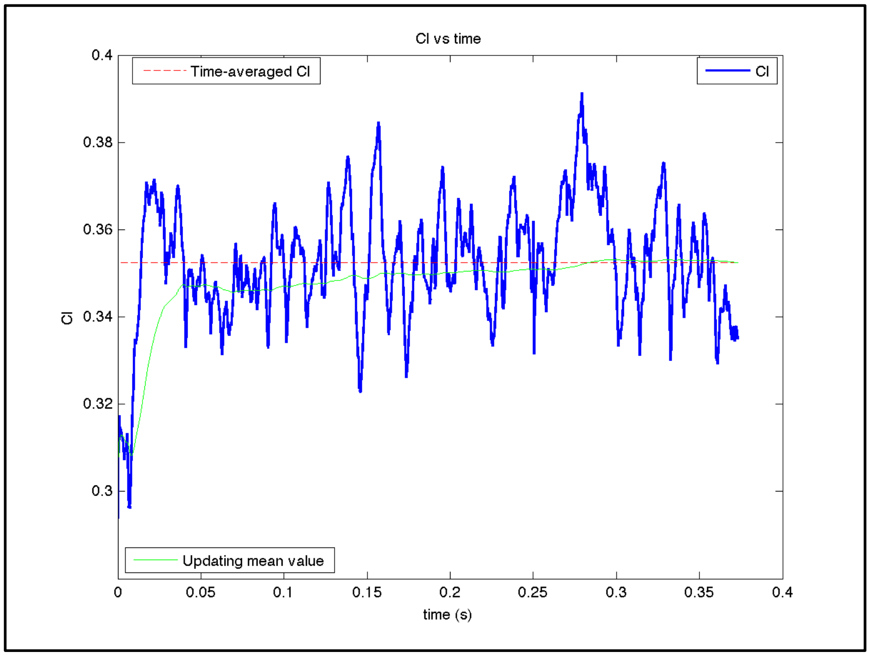


Figure 66 - cl x-direction vs. time SAIDDES

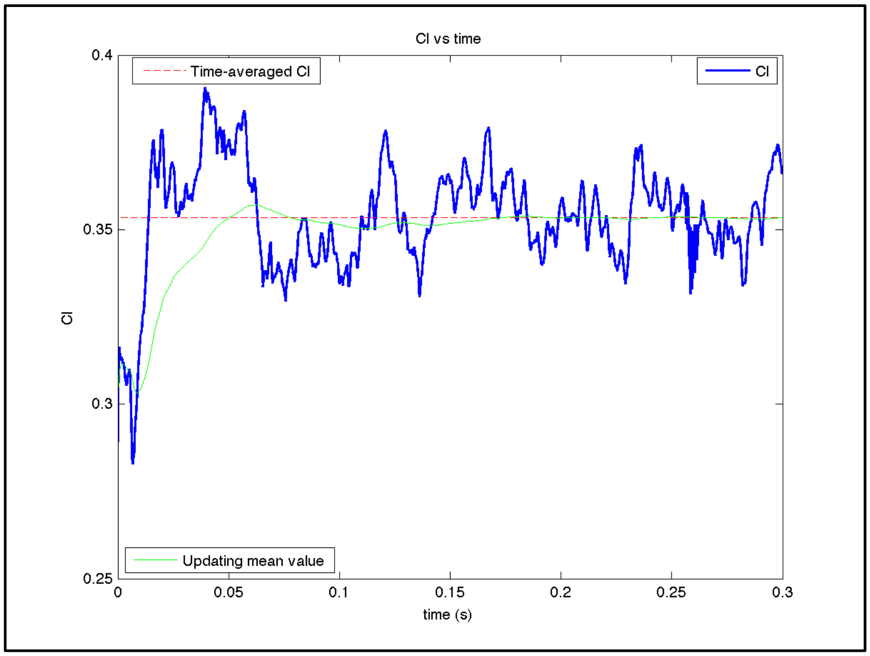


Figure 67 - cl x-direction vs. time kwSSTDDES

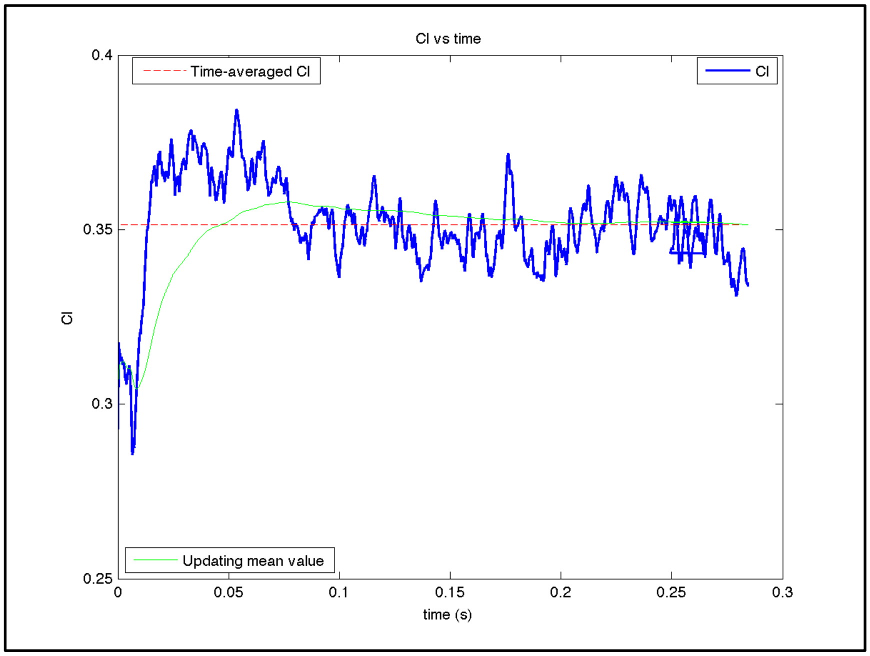


Figure 68 - cd x-direction vs. time kwSSTIDDES

In this case, as it can be seen in the figures, the Cl vs. time plots do not present many differences as the Cd vs. time plots do. They all oscillate more or less around the same value, which is 0.35. The main differences are that the Spalart-Allmaras simulations show bigger oscillation whereas the kwSST ones oscillate a bit less around the average value, but they yet do.

According to [20], the oscillations present in a cylinder case are bigger regarding the lift coefficient values than regarding the drag coefficient ones, but that was not the case in the simulations performed in this study.

4.3.9. Strouhal number

The Strouhal number (St) has been defined for this particular study as:

$$St = \frac{fD}{U}$$

Where f is the frequency of the vortex shedding, D is the diameter of the cylinder and U the inlet velocity of the case.

The time step is related to the Courant number as:

$$Co = \frac{U * \Delta t}{\Delta x} \to \Delta t = \frac{Co * \Delta x}{U}$$

And the Courant number must always accomplish the following criterion:

$$Co \leq 1$$

If feasible results want to be obtained, that must be one of the criteria to be taken into account. Therefore, the time steps used will take that into account. The time steps can be different between simulations, but if a probing wants to be done, they must be the same during the simulation. As long as this is accomplished, the probing is valid. A sketch of where the probes have been placed can be seen at Figures 69 and 70, and has been decided according to [15].

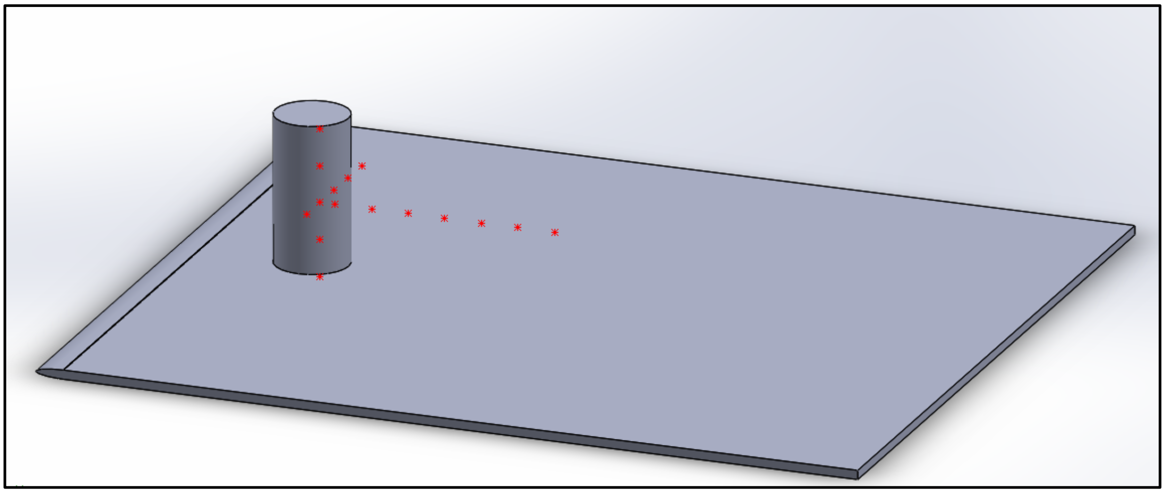


Figure 69 - Probing locations global view

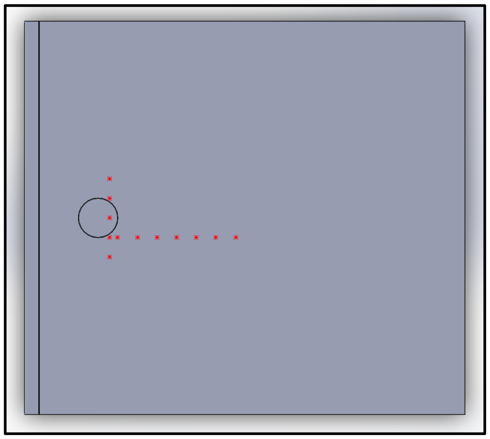


Figure 70 - Probe locations z-normal view

In order to obtain the desired Strouhal numbers in range, enough runtime must be simulated and to determine how much time is at least needed, the Nyquist-Shannon sampling criterion has been followed, which states that the sampling rate must be done at least at double the bandwidth of the signal. The minimum frequency can be related with the Strouhal number with the following expression:

$$fmin = \frac{St * U}{D}$$

According to literature references [15 and 26], the lowest Strouhal number is around 0.16, and therefore the minimum frequency after considering the sampling theorem can be estimated to be around 350 Hz. The sampling rate must be then done taking that into account and the runtime needs to be large enough so that the analysis can be performed safely. To show an example of how the computing of the number has been done, the following plot shows how a sampling result looks like by displaying the SLD vs. the Strouhal number plot after considering the FFT spectrum:

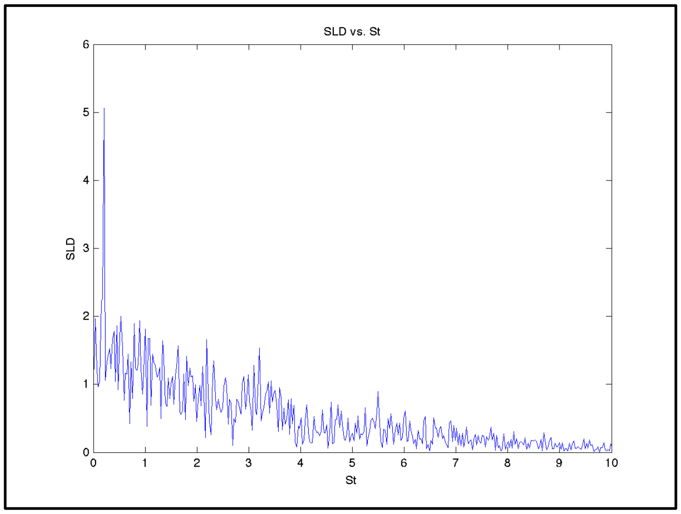


Figure 71 - SLD vs. St plot

Where the two biggest peaks are taken into account to compute the two first Strouhal numbers. The results regarding the first two Strouhal numbers obtained can be seen at the following plot:

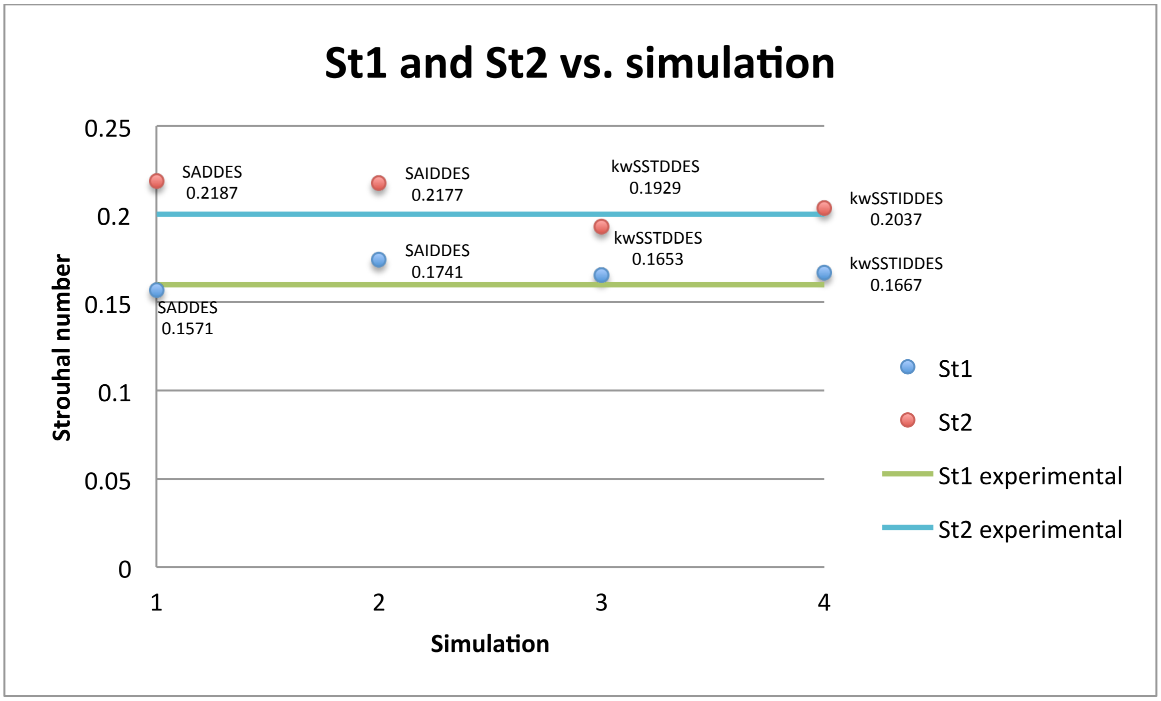


Figure 72 - St1 and St2 values for each simulation comparison

The order is, as always, the following one: Spallart-Allmaras DDES, Spalart-Allmaras IDDES, kwSSTDDES and kwSSTIDDES.

In general, kwSSTDDES simulations predict better the Strouhal numbers than the Spalart-Allmaras ones, which tend to overpredict the values. There are no further relevant differences to comment rather than the IDDES simulations tend to show slightly bigger results if compared to the respective DDES ones.

It can be said that the Strouhal number results agree with the literature research that has been done, since the values are close to the ones obtained in experimental cases.

4.4. Grid convergence study

The aim of this study is to observe how much precision, cells and meshing requirements are needed in order to obtain a good result for the corresponding simulations. Performing such a study requires several different environments, each one with a greater number of cells or meshing tools than the previous one. In an ideal case, the values would get closer at each step towards the finest mesh, but that is not always the case, as it will be described in the following sections.

The meshing technique is however the same, being the setup for blockMesh and snappyHexMesh the only change taken into account between the different domains. The division of the elements has been decided and optimized according to the tool's requirements. That means, basically, trying to have a suitable environment for a hexa-domain, whose cell aspect ratio is the closest possible to 1.

4.4.1. Without layers

The snappyHexMesh meshing tool consists of three steps, as it has already been described in the meshing section. In this particular study, the layer addition step is not considered. In the following tables, the number of cells for each domain, the time-averaged drag coefficient of the x direction and the detachment point have been taken into account to perform the comparison of values. The number of cells is higher than the previous one for the reasons explained above, and the behavior of the drag coefficient and the detachment point is considered after running the simulations, as a post-processing study.

It has been divided into 4 different sections, corresponding to each kind of simulation method used to obtain the results. For the improved cases, the starting point regarding the meshing domain considered comes from the finest mesh of its predecessor.

SA-DDES Simulation:

Environment division	**Number of cells**	**Detachment point (º)**	$\overline{C_x}$
6-6-4	1058763	37.52	0.62
8-8-5	1763371	39.35	0.52
10-10-6	2856548	34.96	0.58
14-14-10	7891525	50.13	0.55
19-19-13	10054387	55.76	0.69

Table 7 - SADDES number of cells, detachment point and cd x-direction values

SA-IDDES Simulation:

Environment division	**Number of cells**	**Detachment point (º)**	$\overline{C_x}$
19-19-13	10054387	61.64	0.71

Table 8 - SAIDDES number of cells, detachment point and cd x-direction values

kwSST-DDES Simulation:

Environment division	**Number of cells**	**Detachment point (º)**	$\overline{C_x}$
6-6-4	1058763	41.08	0.57
8-8-5	1763371	35.66	0.44
10-10-6	2856548	77.21	0.69
14-14-10	7891525	76.54	0.67
19-19-13	10054387	77.12	0.68

Table 9 - kwSSTDDES number of cells, detachment point and cd x-direction values

kwSST-IDDES Simulation:

Environment division	**Number of cells**	**Detachment point (º)**	$\overline{C_x}$
19-19-13	10054387	77.15	0.69

Table 10 - kwSSTIDDES number of cells, detachment point and cd x-direction values

As expected, the number of cells is higher at each environment, since the meshes are less coarse at each simulation regarding the respective predecessor meshes.

Regarding the detachment point results, they are far from ideal for the Spalart-Allmaras simulations. According to the literature references, a detachment point of 81 to 83 degrees is expected, and the results fall short in its attempt to predict it correctly. The coarser meshes for the Spalart-Allmaras DDES simulations show a detachment point around 38 degrees, which is less than half of the value expected. Obtaining such lower detachment points implies that they are placed too far away, and therefore the recirculation zones, where the eddies appear, are smaller and of lower intensity, which derives in lower drag coefficient values, as it is also shown in the tables. The drag coefficient for the last Spalart-Allmaras simulation shows a better drag coefficient value if compared to the experimental results, which implies that the mesh was not appropriate enough and still could be improved, as both this value and the detachment point value for this specific case as well show. The Spalart-Allmaras IDDES case improves the results of its predecessor as expected, predicting a better detachment point, but still too far from the expected results.

KwSSTDDES simulations show better results at the coarser meshes already. It can be seen that at the third environment, the results begin to be close to the experimental results, but still improvable. The detachment point prediction is done way better on these simulations rather than in the Spalart-Allmaras ones. Both the detachment point and the drag coefficient seem to converge from the third environment onwards, which indicates that increasing the number of cells could not necessarily offer better results, if the same meshing style and placement configuration was to be considered. In this case, the IDDES study shows a slightly improvement as well regarding both magnitudes, but the results do not change significantly enough between both final domains for the techniques compared.

To sum up, kwSST simulations show a better behavior as the Spalart-Allmaras ones both for the detachment point and the drag coefficient, which are related as it has been stated before. The improvement shows better stability and reliability of the results even at coarser meshes, not just at the definitive ones. However, there is still much to improve regarding the final results, which can be done by adding layers. The layer addition is the third and final step of the snappyHexMesh utility, and can improve the results greatly, specially at the regions close to the walls, where the further analysis focus as well. Detachment points do not show a convergence in a specific value between the four configurations, but just present close values between both techniques. That is to say, that between the same techniques, the results are similar, being the Improved versions better as expected. The drag coefficient values, however, show a final value for the best meshes that is quite similar, but still far away from the experimental result taken as a reference. The layer addition process is considered on the next section, a factor that improves the meshing around the surfaces greatly and offers better results, which are discussed in-depth as well.

The detachment point vs. domain and the cd x-direction vs. domain plots can be seen at Figure 72 and Figure 73, as a visual reference to the comparison described above.

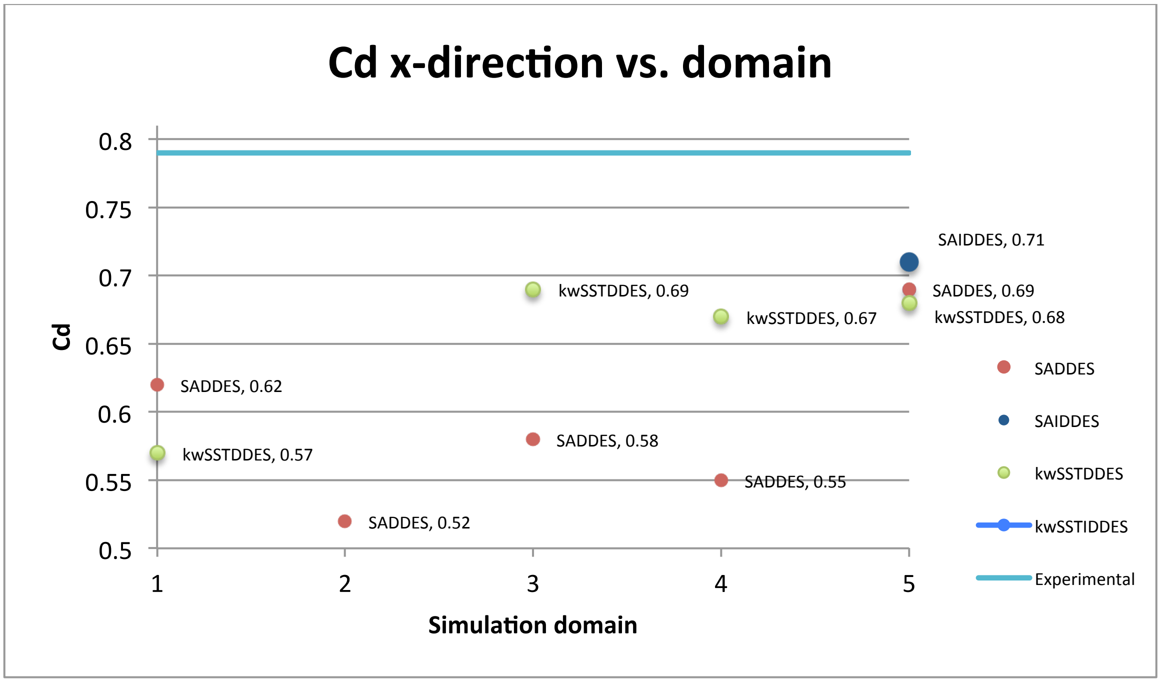


Figure 73 - cd x-direction vs. domain plot (without layers)

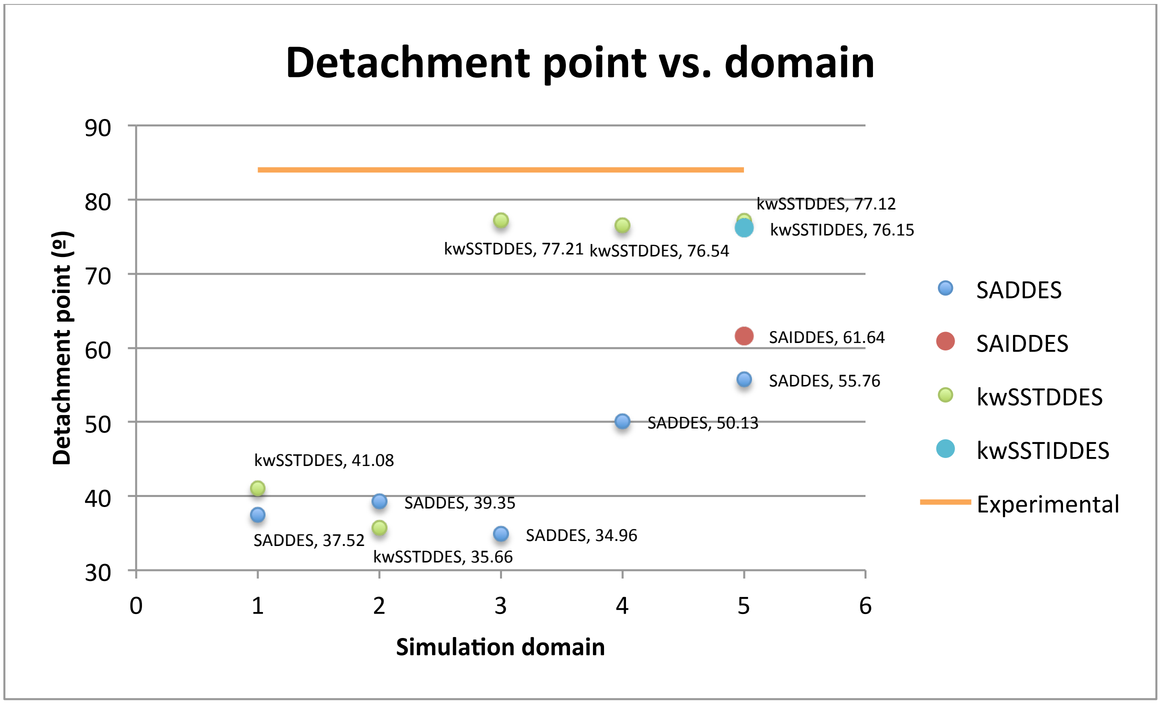


Figure 74 - Detachment point vs. domain plot (without layers)

4.4.2. With layers

In this case, the layer addition process has been considered, and therefore, the number of cells is higher compared to the previous study, since several layers of cells are added to the walls of the solids considered in the simulation. The results are expected to be better after the layer addition process, and the following tables help to observe if the theory is accomplished after the simulations or not.

Furthermore, since these results should be more reliable than the previous ones, which do not include layers on the meshes, several new studies, magnitudes and elements have been carried out and taken into account, a fact that helps even more to validate if the results can be trusted or not.

SA-DDES Simulation:

Environment division	**Number of cells**	**LES %**	**RANS %**	**Detachment point (º)**	$\overline{C_x}$
6-6-4	1487217	26.5	73.5	43.49	0.66
8-8-5	2825580	38.3	61.7	68.58	0.62
10-10-6	4375758	27.4	72.6	64.38	0.71
14-14-10	10765119	30.4	69.6	76.96	0.67
19-19-13	15722344	33.3	64.7	78.92	0.71

Table 11 - SADDES number of cells, detachment point, LES-RAS and cd x-direction values

SA-IDDES Simulation:

Environment division	**Number of cells**	**LES %**	**RANS %**	**Detachment point (º)**	$\overline{C_x}$
19-19-13	15722344	53.2	46.8	80.59	0.76

Table 12 - SAIDDES number of cells, detachment point, LES-RAS and cd x-direction values

kwSST-DDES Simulation:

Environment division	**Number of cells**	**LES %**	**RANS %**	**Detachment point (º)**	$\overline{C_x}$
6-6-4	1487217	14.8	85.2	71.95	0.72
8-8-5	2825580	20.4	79.6	73.44	0.64
10-10-6	4375758	17.2	82.8	74.96	0.73
14-14-10	10765119	20.4	79.6	77.82	0.75
19-19-13	15722344	17.8	82.2	81.37	0.73

Table 13 - kwSSTDDES number of cells, detachment point, LES-RAS and cd x-direction values

kwSST-IDDES Simulation:

Environment division	Number of cells	DES %	RAS %	Detachment point (º)	$\overline{C_x}$
19-19-13	15722344	18.2	81.8	76.54	0.70

Table 14 - kwSSTIDDES number of cells, detachment point, LES-RAS and cd x-direction values

y+ Study:

Simulation	y+ mean value
SA-DDES	21.39
SA-IDDES	21.68
kwSST-DDES	20.75
kwSST-IDDES	21.18

Table 15 - y+ mean values for each simulation

In addition to the detachment point and the drag coefficient, the RANS and LES regions (in percentage) have been considered to perform the analysis, as well as the mean dimensionless distance to the wall. In comparison to the previous results, which were without considering the layer addition process, the results obtained are much better. Therefore it can be stated that, if the snappyHexMesh wants to be used to simulate and obtain reliable results, considering this step is compulsory.

As a starting point, the number of cells with and without considering the layer addition step has been plotted for each configuration in Figure 74. The number of cells is higher when considering the step as expected, being higher the gap between domains at each step, which also makes sense since the proportion between the divisions is a bit higher at each step, always keeping in mind that the cell aspect ratio has to be as close to 1 as possible when doing the initial mesh that is used as a base if good results want to be achieved.

Simulation	Min. value	Max. value
SADDES	0.089	108.495
SAIDDES	0.103	114.470
kwSSTDDES	0.076	105.941
kwSSTIDDES	0.072	103.265

Table 16 - y+ min. and max. values for each simulation

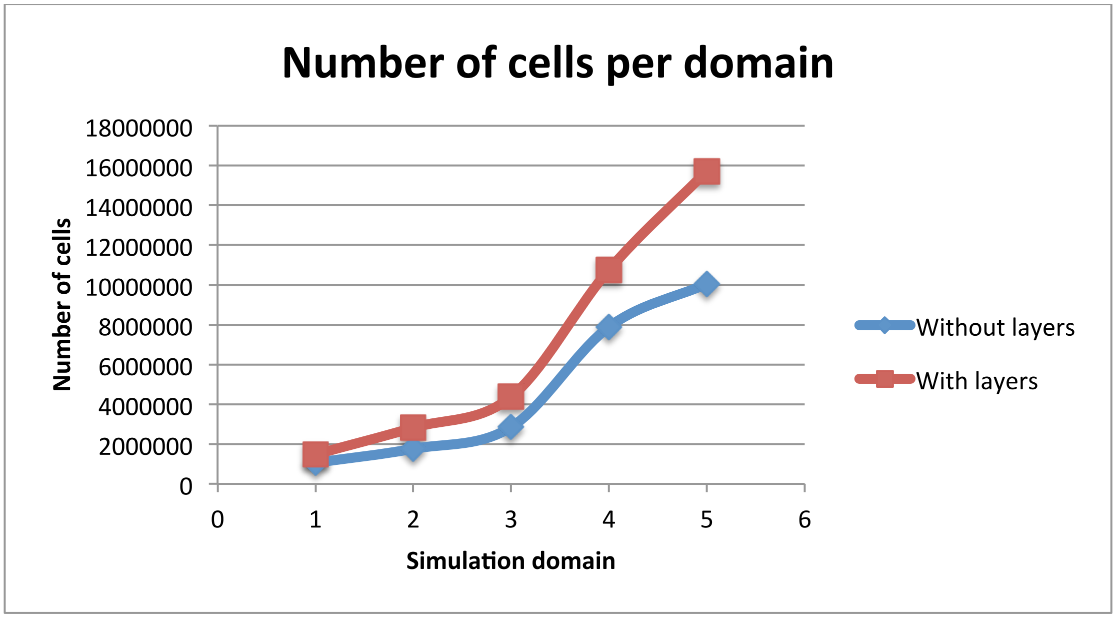


Figure 75 - Number of cells per domain vs. domain

Starting with the Spalart-Allmaras DDES results, they follow a more stable trend as before. The oscillations between configurations are smaller than before, which indicates that the difference between the domains considered is not as big as before. However, the global improvement trend can be seen as well in this case as well. Each mesh gives similar or better results as its predecessor, which is always a good sign, as the meshes get better and better at each step. The LES % vs. domain plot shown at Figure 75 shows significant differences between Spalart-Allmara and kwSST simulations. The kwSST cases show almost identical percentages (with smaller improvements in the IDDES case as it has been commented in the slices comparison) whereas Spalart-Allmaras ones behave differently, being a huge difference present between them as it could be seen in the slice views regarding this specific magnitude. The detachment points oscillate a bit more, being the value of the final configuration close to the experimental results, which shows a good prediction of the detachment point over the surface. As a consequence, the drag coefficient values are higher as expected. The IDDES technique improves the results significantly, predicting a detachment point and drag coefficient values almost identical to the ones from the experimental study, which indicates that the model gives accurate results.

The LES Region's % is lower when considering the kwSST simulations. The detachment point prediction is, as it was without considering the layers added, better even at the worst meshes, presenting a final value even better than the Spalart-Allmaras simulations, almost identical to the value obtained at the experiment performed. In this case, however, the IDDES case does not show an improvement with respect to the DDES predecessor, and that can be related with the kind of mesh produced not being suitable enough for this technique, but the results are still good enough to be considered, falling short in predicting the drag coefficient value and the detachment point.

A graphical representation of the results presented at the tables above can be seen at Figures 75, 76 and 77. The LES % vs. domain can be seen at Figure 75 (plotting the RANS % would be redundant since the sum is 100), the cd x-direction vs. domain at Figure 76 and finally the detachment point vs. domain at Figure 77, so that the user can observe the trends described.

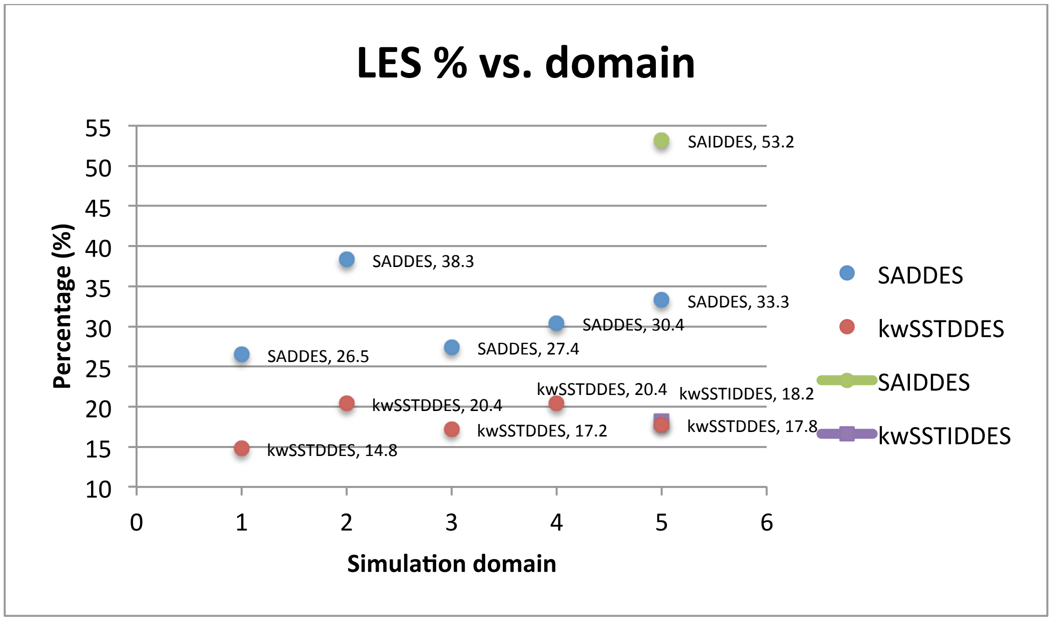


Figure 76 - LES % vs. domain with layers

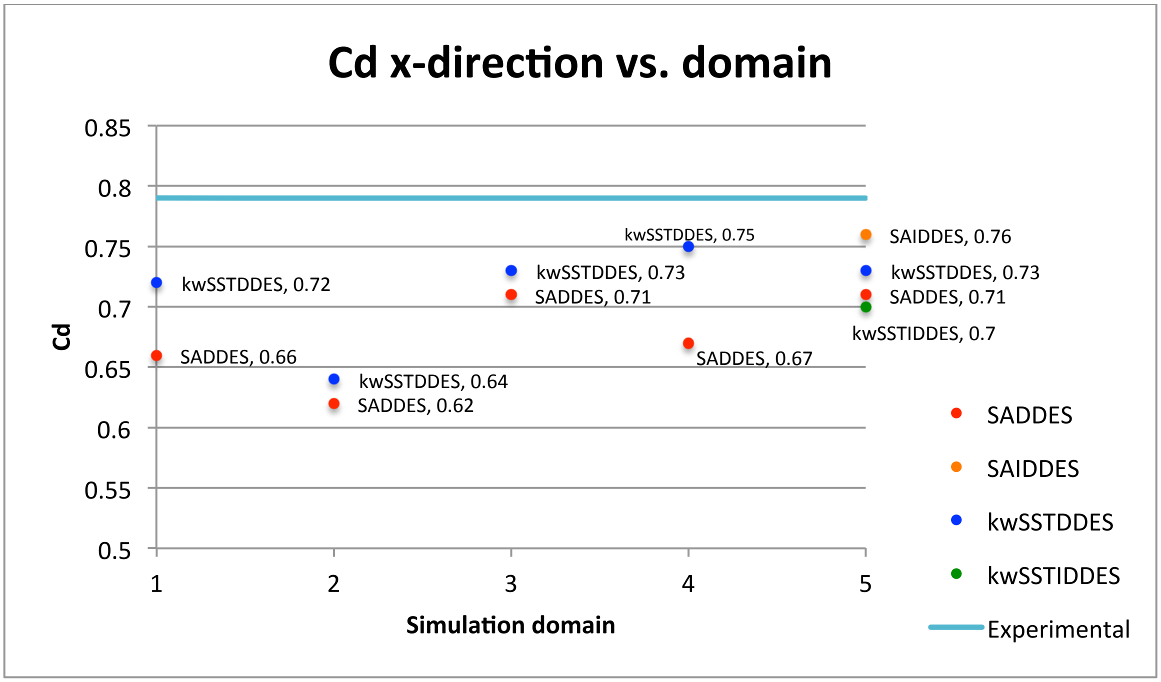


Figure 77 - cd x-direction vs. domain with layers

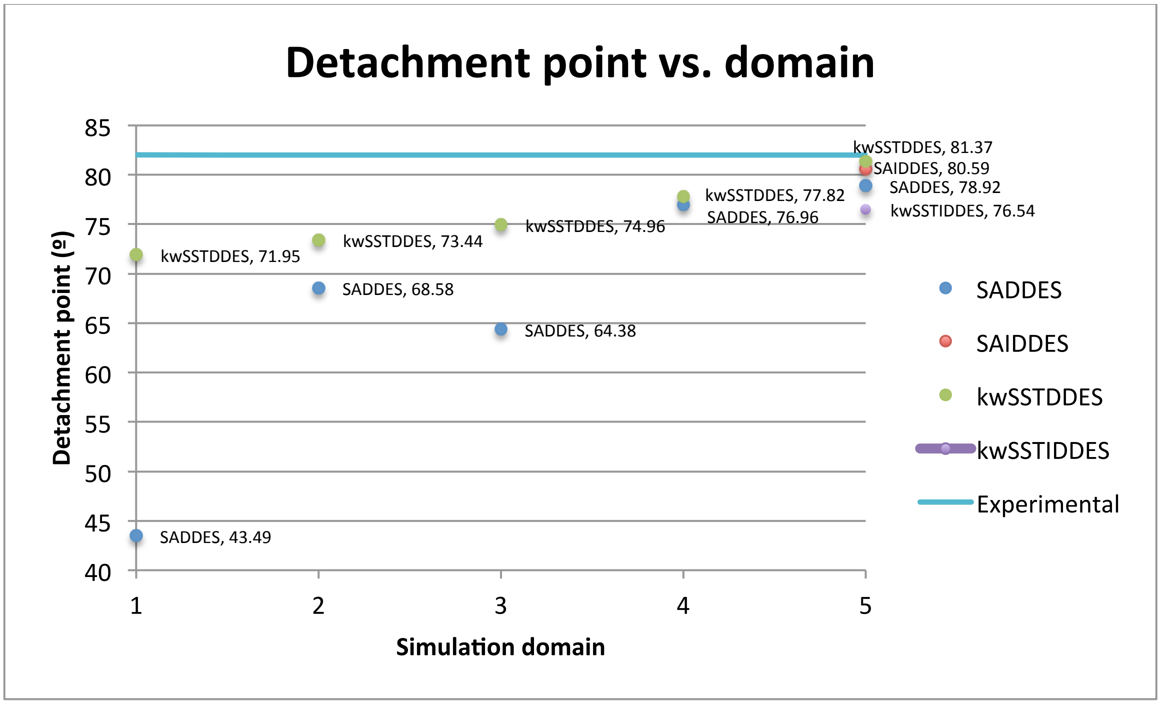


Figure 78 - Detachment point vs. domain with layers

4.5. Cutting the plate in half effect

Another different assumption that we can consider is the fact of “cutting” the plate in half for the simulations. What this consideration allows is to run ideally the same simulation with similar results and less computational time and number of cells, and furthermore the fact that the fluid will not follow the empty path under the plate, which could affect in the results obtained if it goes downstream instead of facing the cylinder surface fully as desired.

To perform this particular study, two different smaller studies have been conducted. In the first one, the drag coefficient value, the computational time it takes to reach a certain milestone and the number of cells are compared between the two finest meshes corresponding to a non-cut and a cut domain. In the second one, the mean velocity in a x-normal slice has been considered both considering the presence of the cylinder and the plate and without them. The mean velocity has been taken into account at a region before the cylinder surface in order to appreciate the expected blocking effect of the cylinder that forces the flow to go underneath the plate. The volumetric flow rate has been computed to obtain the reduction of the mass flow rate afterwards, which can be defined as:

$$\dot{V} = v * A$$

Being v the flow velocity (the mean velocity on both planes in this case) and A the cross sectional surface. The volumetric flow rate is related to the mass flow rate in the following way:

$$\dot{m} = \rho * \dot{V}$$

And by the quotient of both mass fluxes, the percentage of the flow that goes underneath the plate as a consequence of the blocking effect that the cylinder and the plate produce on the flow can be computed. Furthermore, the mean velocity plots are analyzed to see how the field looks like right in front of the cylinder.

4.5.1. Cd study

Simulation	**Number of cells**	**Computational time up to 0.1s** [s]	$\overline{C_x}$
SA-DDES non-cut	24819220	1389164	0.72
SA-DDES cut	15722344	518343	0.71

Figure 79 - Cd half-plate study

4.5.2. Fluid study

Case	Mean velocity (x-Plane cut) $[\frac{m}{s}]$	Cross sectional surface $[m^2]$	Mass flow rate $[\frac{kg}{s}]$	Percentage of the fluid going underneath the plate (%)
Without the surfaces	26.000	0.756	19.656	11.9
Considering them	27.555	0.637	17.555	

Figure 80 - Fluid half-plate study

As it can be seen on the first study, the number of cells increases greatly (the cut case has a number of cells of approximately 63% the amount from the non-cut case) from the first case, which is the cut one to the non-cut one. The main reason is that the domain is smaller due to the fact that the region behind the cylinder is not considered on the cut case, and therefore there are more cells to refine. When running the simulation, the time step is always smaller for the non-cut domain, if the CFL≤ 1 criterion wants to be accomplished to present physical results. Following the same trend, the computational time to reach the first 0.1 seconds, taken as a milestone, is bigger in the non-cut case than in the cut one, and the reason is that the number of cells is bigger as well and therefore the mesh is bigger, which requires more computational time to resolve the domain at each time step. The computation time required for the cut case is approximately 37% of the one required for the non-cut case, which shows a significant reduction, a fact that pays off if the results obtained from the cut case are close enough to the real ones performed experimentally or at least as good as the ones obtained for the non-cut case, a fact that can be observed by looking at the drag coefficient values, which are almost identical between both simulations, and therefore can the cut case be considered as a good approximation.

The second study focuses on computing the magnitudes described above, with aid of the sample utility that openFoam offers, and then the volumetric flow rate and the mass flow rate, whose rate between both surfaces considered gives the percentage of the fluid that the geometry forces to go underneath instead of facing the cylinder surface fully. The results show a 11.9% of fluid going underneath the plate, which is a significant percentage to at least take into account. Ideally, the whole domain should be considered, but the boundary conditions for the wall that is placed below the round plate cannot be reproduced in openFoam successfully.

However, as the previous study shows, the fact of cutting the domain offers results that are close enough to the ones obtained if not cutting it, being the percentage of the fluid a reasonable value as well and therefore it can be stated that cutting the domain as it has been done offers a good approximation result wise.

A comparison between the velocity profiles obtained for two different x-Plane cuts, one placed at the inlet and the other one before the cylinder surface has been made, and can be seen at Figure 80.

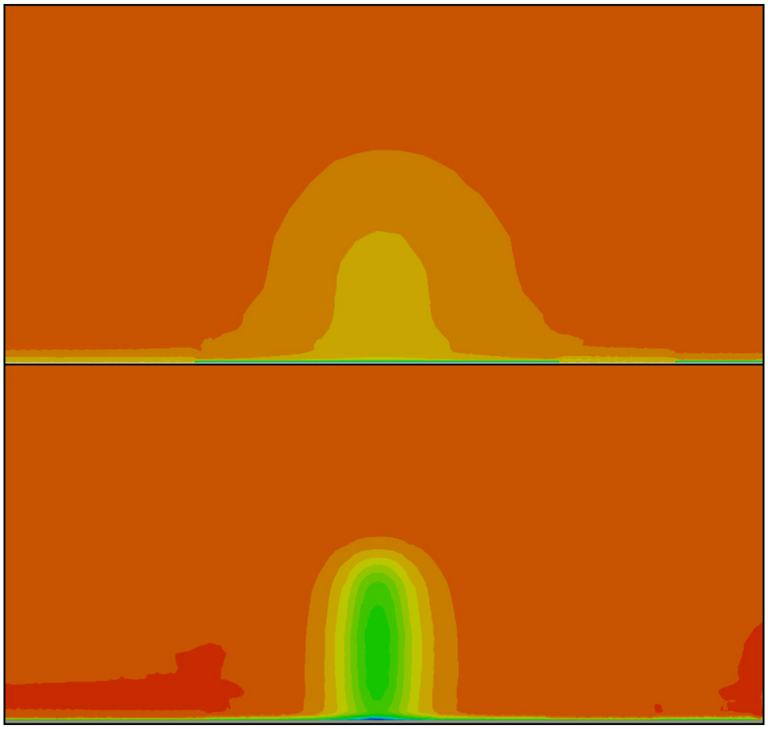


Figure 81 - UMean at two different x-normal slices

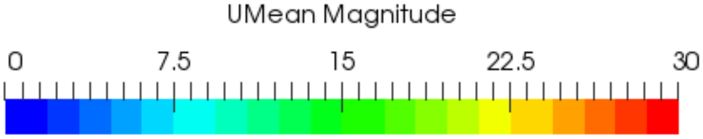


Figure 82 - UMean colorbar

As it can be seen on Figure 80, the velocity profiles present similar values for the whole domain but they present some differences at the region parallel to where the cylinder is placed. That shows the influence of the surface over the flow, whose mean velocity is reduced the closer it gets to where the surface is placed. The shape of the gradients suggests the form of the cylinder, and therefore the statement above can be made, since the cylinder is blocking the field as Figure 80 displays.

4.6. Trip-less approach study

The trip-less approach is a technique or assumption originally made by Travin et al. [18], in which the kinematic viscosity is set to very low values at the inlet of a simulation in order to have better and more realistic detachment points and therefore, drag values. The theoretical explanation why this is produced can be seen at the theoretical section (separation part) of this document in which the difference between the two types of boundary layers (turbulent and laminar) is explained. Therefore, to perform this study, the kinematic viscosity will be reset to zero at the inlet after the mapping of the fields from the steady state simulation (considered as an initialization of the flow) and then the simulations will be run. Therefore, it has been decided to manually adjust the transition behavior of the RANS model to enforce the laminar cylinder boundary layers known to exist on the experiment. The reason behind this fact is that the prediction of laminar to turbulent transition falls outside the scope of RANS models, which tend to predict boundary layer turbulence too early on in their development, a fact that can cause an underestimation of the drag coefficient values, and that is the theory that is tested on this specific study.

The magnitudes that are compared are the detachment point angle and the time-averaged drag coefficient in the x direction. Again, the finest meshes for the simulations selected have been chosen to obtain the best results possible.

According to the original paper, the theoretical behavior of these magnitudes follows the trend that if the trip-less approach is considered, the detachment point angle is expected to be bigger, since the laminar boundary layer (which is accomplished by the fact of resetting the kinematic viscosity at the inlet as it has been described) cannot resist detachment in the same way as a turbulent boundary layer does, and therefore the drag coefficient is expected to be higher [18].

Environment division	**Detachment point (º)**	$\overline{C_x}$
SA-DDES non trip-less	75.04	0.71
SA-DDES trip-less	78.92	0.71
SA-IDDES non trip-less	79.38	0.68
SA-IDDES trip-less	80.59	0.76

Table 17 - Trip-less approach detachment point and cd x-direction comparison

Table 16 shows the comparison between the detachment point and the drag coefficient values for the 4 simulations that are studied. Regarding the DDES cases, the trip-less approach gives the expected results for the detachment point, since it is bigger in the trip-less approach case than in the non trip-less one.

Therefore, this fact states that the separation point is predicted before when applying the approach, which also should produce bigger drag coefficients, since the recirculation zone is expected to be bigger. In this case, however, the drag coefficient does not exactly show the expected result, being the averaged value equal to the one obtained without applying the approach.

However, when comparing the IDDES cases, the trend is successfully accomplished. The detachment point is bigger for the trip-less approach case, which places this point before than in the case where the approach is not considered. The difference in drag coefficient values shows the expected behavior as well, since after considering the approach, the drag coefficient increases greatly up to 0.76, a value that is close to the experimental one, which is 0.79. Therefore, the effectiveness of the approach can be demonstrated by looking at the results presented. A comparison of the drag coefficient value plots can be seen in Figure 82, which displays around which magnitudes they oscillate and the fact that drag coefficient values when considering the trip-less approach are bigger than when not considering it, as it has been stated.

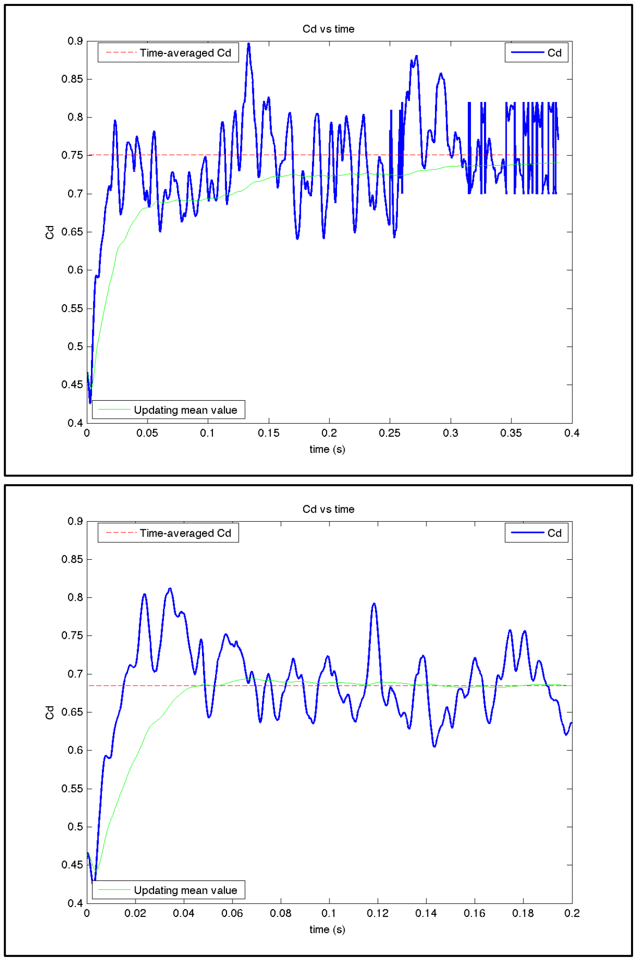


Figure 83 - cd x-direction comparison when considering the trip-less approach (SAIDDES)

4.7. Computation time

The main four cases have been considered to perform this analysis. What has been done is to place all four simulations in the same cluster, with identical running conditions and just with one processor being used. 100 iterations have been run each setup in serial (1 CPU) to be able to check how much the turbulence models cost (in computation time) compared to each other.

The results can be seen at the table below:

Simulation	Average time per iteration [s]	Computation time
SA-DDES	200.7	5 hours 32 minutes
SA-IDDES	201.4	5 hours 37 minutes
kwSST-DDES	194.0	5 hours 23 minutes
kwSST-IDDES	213.6	5 hours 56 minutes

Table 18 - Computation time comparison

Spalart-Allmaras simulations almost present no differences between both cases, since the average time per iteration is almost identical. 5 minutes of difference after 100 iterations is not a significant difference, but if a whole simulation wants to be run up to 1 second of physical time, just to give an example, the difference in computation time could be significant. SADDES is then just 0.35% faster than SAIDDES

kwSST cases present a greater difference in comparison to Spalart-Allmaras ones. kwSSTDDES is the case that took less time to compute, having the lowest average time per iteration and therefore, the lowest computation time for the number of iterations given. Contrary to what has been seen to Spalart-Allmaras, the difference in computation time is already significant at 100 iterations, and therefore, if the same trend is followed for further iterations, the differences in computation time if the user wants to simulate up to 1 second of physical time or more can be up to days, which is something worth considering. kwSSTDDES is then 10.1% faster than kwSSTIDDES.

Another interesting trend to comment is that both DDES cases took less time to compute than the corresponding IDDES ones, resulting the just mentioned increase significantly greater for the kwSST simulations.

Finally, a surprising fact about the behavior of the kwSST cases was that even though they are 2-equation, and thus, more complex models than the Spalart-Allmaras ones, this behavior is not reflected by the fact that the kwSSTDDES case was the fastest one, probably due to the fact that only 100 iterations were taken into account.

5. Conclusions

The most important objective in this section is to show whether the theoretical comparison made in the theoretical background section can be demonstrated or not by the results obtained, which is one of the best ways to check if the results obtained make sense or not, apart from comparing them to experimental results found on the literature research. The main differences between the models have been summed up in order to easily compare between the theoretical and expected behavior and the one that the simulations have shown, as it can be seen in the results section. Finally, a short comparison of the results obtained has also been made, explaining briefly which model or models are considered the best after comparing the results obtained with the expected trends.

As a starting point, the Spalart-Allmaras cases are considered. The Spalart-Allmaras model is a simple, 1 equation turbulence algebraic model that was derived using empirical relationships, dimensional analysis, and Galilean invariance with the objective of producing a turbulent transport model that was fast, numerically stable, and reasonably accurate for both shear layers and boundary layers. It was originally focused on cases related to the automotive and aerospace industries, and by taking some assumptions its results are not accurate enough around complex geometries. It solves a modeled transport equation for the kinematic eddy turbulent viscosity and adds a single additional variable for a Spalart-Allmaras viscosity, without considering any wall functions, but solving the entire field instead. However, this model shows lower accuracy in fields that exhibit shear flow or separated flow. Since the case studied does exhibit separated flow, the results are expected to be improvable, if compared with other models, and that is what has happened by looking at the results obtained.

As it can be seen in the results section, the results obtained close to the separation zone are still improvable. The detachment point prediction is better in the kwSST cases, just to compare them with other models. Spalart-Allmaras simulations needed greater refinement to begin predicting the detachment point location, whereas kwSST models predicted them way better already in domains with a significant lower number of cells. Furthermore, the drag coefficient values keep oscillating even after 10 times the time required to physically reaching the end of the domain if considering the free stream velocity. Other comparisons and studies also show this improvable flow prediction in regions close to the surface. A couple of examples could be the isolines of null value, which were either too far away or too soon over the surface of the cylinder or the mean velocity in the x-axis contours, which have also been compared to other simulations as well.

The shapes and contours around the surfaces were not as expected or shown in experimental studies and further simulations, and therefore the theoretical limitations of this model have been proven.

The kwSST model is a two-equation turbulence model. It includes two extra transport equations to represent the turbulent properties of the flow, allowing the model to account for history effects like convection and diffusion of turbulent energy, improving the performance to previous less complex models, such as the Spalart-Allmaras model. It has an accurate near-wall treatment, provided the resolution near it is high enough. Its complexity extends as well to more complex geometries, in comparison with the Spalart-Allmaras model, making it more robust, but the most important pro regarding this specific model is the fact that it theoretically gives good results for adverse pressure gradients, separated flows and boundary layer treatment, facts that are expected to be significant improvements for the case studied in this project.

The results obtained with this model are not far from the experimental and simulated ones that have been found in the literature research, which is a great sign. The handling of adverse pressure gradients and the separation of the flow was close the expected results as well. Velocity contours, as it has been shown in the result section, are close enough to the literature results, and the prediction of the detachment point is close to the experimental results even at coarse meshes, obtaining good values in the finest cases. Another aspect that has been proven is the isosurfaces of null value and the behavior of the magnitudes studied in the results section, which follow the expected trends if compared to the experimental results taken as a reference. One of the downsides of this kind of model is the difficulty to converge, and it is something that has also been proven by the fact that the kwSSTIDDES simulation is the one that took the longest computation time and was run until longer physical times but still presented some instabilities in the flow prediction after analyzing them in post-processing.

After stating the main differences between both models tested and the quality of the results obtained in comparison with the references that were available to check the validity of those results, it can be stated that the kwSST cases offered better results for the case studied, improving the ones obtained in the Spalart-Allmaras cases.

Finally, the differences between DDES and IDDES have been summed up to compare if they result in a significant improvement or not when comparing the results obtained. The main advantages of IDDES are a better prediction of the detachment behavior, improved near-wall region results and the versatility in the range of possible Reynolds numbers at which it performs well, among others. The main advantage presenting by the last one is the wall-modeling capability.

It is mainly for academic purposes and it is not implemented in industry cases yet, but some experiments have already been conducted to value its possible implementation on a near future, despite the fact that the modeling is quite complex.

One of the downsides of this model is, as stated, the complexity. That means, among other aspects and considerations, that it is highly sensitive to near-wall refinement, grid spacing and structuring. In order to see significantly improved and better results close to the walls, a very specific and complex meshing must be performed, which is not always easy and it requires a high computation cost and a huge number of cells as well. The IDDES cases do not present significant differences with respect to the DDES ones, especially for the kwSST simulations. One of the reasons could be that the meshing is not done in a way that takes advantage of the IDDES advantages fully, and therefore an improved meshing could be considered, if better results were desired.

Therefore, in order to improve the results or at least take more advantage of the benefits that this model can theoretically offer, other tools could be used to mesh or just change the meshing parameters available for the snappyHexMesh tool taking into account grid design guidelines that are given for this specific kind of simulations, which are not easy to follow and are quite different for each case. In this study, however, the differences could not be appreciated fully since the meshing is more standardized between models (with slight modifications but quite similar in general), and therefore they cannot be pointed out either by looking at the results obtained or by comparing them to other relevant simulations conducted in the past, due to a lack of references for this model, perhaps because it is still a model that needs a wider implementation.

The asymmetrical behavior that many plots show can be explained by the fact that from the Steady-State RANS Simulation the velocity profile was not symmetrical. When building the STL files on SolidWorks and importing them to OpenFOAM, many problems can appear. The main problem is that, in order to be coherent and compare in the easiest way the results with other experiments and papers, the y and z-axis need to be shifted, but also the fact that when exporting the geometries, the triangles and vertices created, even at maximum resolution, could not define the surface as it is designed, especially in cases like a cylinder where the round shape is not easy to define. When shifting those axes, a slight shifting of 0.015m in the y-direction was made, causing an asymmetrical placement of the surfaces and thus, after performing the Steady-State simulation, importing the initial conditions to the final cases, expanding the slightly asymmetrical behavior on those as well. In order to prevent that in future cases or further studies, the STL geometries have been first of all placed symmetrically, checking that there is no shifting in any direction whatsoever, and then expanded to prevent problems if they match exactly the size of the domain. The meshing utility is designed to cut the domain where it is specified and therefore no problems show up.

6. Overview

The main objective of this section is to point out the possible changes that would be made if the project had to be started again or possible suggestions that could improve the results already obtained.

Those suggestions can be related to specific parameters that could be changed, alternative meshing configurations, a better design of certain elements of this project or just the fact that some simulations could be run faster.

To start with, the first aspect that must be considered is the mesh. SnappyHexMesh has a great variety of parameters that can define the meshing that wants to be done. Some of them are really advanced, and by changing them, floating-point exceptions were faced, problems rose up and the simulation could not start or just stopped after a few iterations, which led to a more standard meshing with still some specific improvements with the objective of improving the result.

Other suggestions can be related to the refinement. This meshing utility can adapt the refinement to both general, specific regions and of course around a specific surface. It is therefore likely to obtain better results if the refinement is increased around the surfaces of interest, but the downside of doing that is the computational time and effort that a finer mesh would require, since for the final case the number of cells is close to 16 million, which is already a significant number to take into account when meshing and running a simulation afterwards within such a domain. Another improvement that can be done is changing the location and shape of the refinement zone, as well as the refinement within it.

According to the literature research that has been done, a better refinement zone could be created around the cylinder and focusing at the wake as well, as suggested in Figure 83.

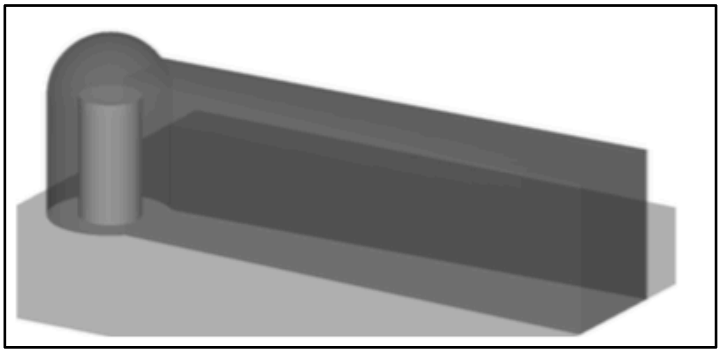


Figure 84 - Possible improvement of the refinement zone [15]

Another problem that has been faced is the convergence of some models, especially the Spalart-Allmaras ones while looking at the drag coefficient plots, which still oscillate even after 7 times past the physical time that it takes to the fluid to cross the whole domain.

The problem behind it is a lack of time. Running a simulation in the fastest cluster with 16 processors working fully can take up to 10 days to simulate 0.25 seconds, which is just a quarter of a physical second. The domain is big and the number of cells is close to 16 millions, and therefore the simulations could not be run longer, but still the results obtained are not bad and can be mapped if a further study wants to be performed right afterwards.

Something than can always be changed is the time and spatial discretization. This project has followed some guidelines according to successful previous simulations as specified in the bibliography but if the results want to be improved, they can be changed with this objective, and the same would happen with the boundary conditions.

Apart from the mesh configuration and design, the running time and further refinements, the only aspects that might be worth changing if the results want to be improved are the boundary conditions and the steady-state RANS simulation that is used as in initialization of the flow.

Regarding the boundary conditions, the procedure has been to consider guidelines based on successful literature reports, but every case is different and therefore something might be better in a different way, like for example a specific boundary condition related to a turbulent magnitude, using an improved boundary condition that has been included in newer versions of openFoam and so on.

The initialization of the flow already shows good convergence but it can always be better. A different and improved simulation could be used as an initialization but the results would not change drastically in the end. The number of oscillations would be significantly reduced at the beginning but the averaging should give similar results in the end.

OpenFoam has some problems and bugs related with the resetting of the averages as well. Sometimes, when the simulations have been stopped for some reason, after starting the simulation again some averages have been reset deleting the previous ones even with the flag of reset set to off. What this has caused is that the averaging afterwards did not take as many samples as desired and some instabilities could show up in the results, which has not exactly been the case but it is always something to consider, especially after looking at some results obtained with the kwSSTIDDES model.

7. Bibliography & references

[1] Anderson, John D. (2004), “Introduction to Flight”, McGraw-Hill. ISBN 0-07-282569-3.

[2] Hossain N. (2012), “Modeling Thermal Turbulence Using Implicit Large Eddy Simulation”.

[3] Menter, F. R. (1993), “Zonal Two Equation k-ω Turbulence Models for Aerodynamic Flows”, AIAA Paper 93-2906.

[4] Menter, F. R. (1994), “Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications”, AIAA Journal, vol. 32, no 8. pp. 1598-1605.

[5] Andersson B., Andersson R., Hakkanson L., Mortensen M. (2012), “Computational fluid dynamics for engineers”, Cambridge University Press, ISBN 978-1-107-01895-2.

[6] Carlomagno G.M., Brebia C.A., (2011), “Computational methods and experimental measurements XV”, WIT Press, 2011, ISBN 978-1-84564-922-7.

[7] Gritskevich, Garbaruk, Schütze, Menter F. R., (2011), “Development of DDES and IDDES Formulations for the k-ω Shear Stress Transport Model”

[8] Menter F.R. and Kuntz M., (2002), “Adaptation of Eddy-Viscosity Turbulence Models to Unsteady Separated Flow Behind Vehicles.” Proc. Conf. The Aerodynamics of Heavy Vehicles: Trucks, Buses and Trains, Asilomar, Ca, 2002

[9] Spalart P. R., Jou W.H., Strelets M., Allmaras S.R. (1997), “Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach”

[10] Shur, M.L., Spalart, P.R., Strelets, M.K., Travin, A.K., (2008), “A hybrid RANS-LES approach with delayed-DES and wall-modeled LES capabilities”. Int. J. Heat Fluid Flow 29, 1638–1649, 2008

[11] P.R. Spalart, W-H Jou, M. Strelets, S.R. Allmaras, “Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach”, in: C. Lin, Z. Lin (Eds.), Advances in DNS/LES, Creyden Press, Columbus, OH, 1997, pp. 137–147.

[12] P.R. Spalart, S.R. Allmaras, “A one-equation turbulence model for aerodynamics flows”, La Recherche Aerospatiale 1 (1994) 5–21.

[13] Travin A. K., Shur M. L., Spalart P. R., et al., “Improvement of delayed detached-eddy simulation for LES with wall modeling”, in: Wesseling P, Oate E, Périaux J, eds. European Conference on Computational Fluid Dynamics. The Netherlands: TU Delft, 2006.

[14] M.L. Shur, P.R. Spalart, M.K. Strelets, A.K. Travin, “A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities”, Int. J. Heat Fluid Flow 29 (2008) 1638–1649.

[15] Frederich O., (2010), “Numerische Simulation und Analyse turbulenter Strömungen am Beispielder Umströmung eines Zylinderstumpfes mit Endscheibe”.

[16] Greenshields, Christopher J., (2015), “OpenFOAM – The Open Source CFD Toolbox User Guide”

[17] Strouhal, V., (1878), “Ueber eine besondere Art der Tonerregung”, Annalen der Physik und Chemie, 3rd series, 5 (10) : 216–251

[18] Travin A., Shur M., Strelets M., Spalart P., (1999), “Detached-Eddy Simulations Past a Circular Cylinder”, Federal Scientific Center “Applied Chemistry”, St. Petersburg 197198, Russia and Boeing Commercial Airplanes, P.O. Box 3707, Seattle, WA 98124, U.S.A.

[19] Wilcox D., (1994), “Turbulence Modelling for CFD”, DCW Industries Incorporated and the University of Michigan, Michigan USA, ISBN 0963605100, 9780963605108

[20] Mockett C., (2009), “A comprehensive study of detached-eddy simulation”, Fakultät V – Verkehrs- und Maschinensysteme der Technischen Universität Berlin, Berlin, Germany, ISBN 978 3 7983 2159 5

[21] Guerrero J., (2015), “Turbulence modeling in OpenFOAM®”, University of Genoa, DICCA Dipartimento di Ingegneria Civile, Chimica e Ambientale, Genoa, Italy

[22] Spalart P., (2001), “Young person’s guide to Detached-Eddy Simulation Grids”, Boeing Commercial Airplanes, Seattle, Washington, USA

[23] Verhoeven O., (2011), “Trailing Edge Noise Simulations using IDDES in OpenFOAM”, Delft University of Technology, Netherlands

[24] Keith A. Weinman, Harmen van der Ven, Charles R. Mockett, Tobias A. Knopp, Johan C. Kok, Rodolphe T.E. Perrin and Frank H. Thiele, (2006), “A study of grid convergence issues for the simulation of the massively separated flow around a stalled airfoil using des and related methods”, German Aerospace Center, Technical University of Berlin, National Aerospace Laboratory, Berlin, Germany, European Conference on Computational Fluid Dynamics

[25] Lindblad D., (2014), “Implementation and run-time mesh refinement for the $k - \omega$ SST DES turbulence model when applied to airfoils”, Chalmers University of Technology, USA

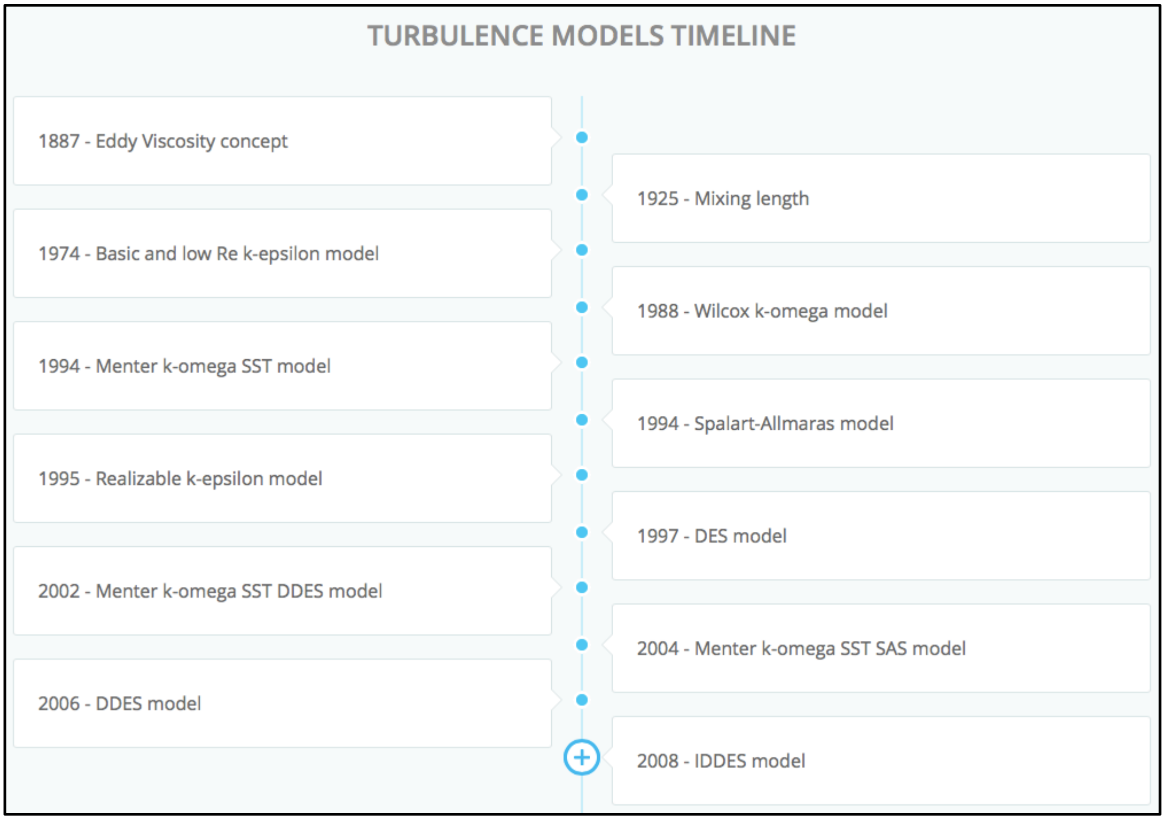
[26] Pattenden, R. J., Turnock, S. R. & Zhang, X., (2005), “Measurements of the flow over a low-aspect-ratio cylinder mounted on a ground plane. Experiments in Fluids 39 (1), 10–21.

8. Appendixes

In this section, the reader can find several additional elements that contribute in the general understanding of this project. The elements included will be mainly simulation files, in which the different parameters and configurations can be seen in-depth.

8.1. Turbulence Models timeline

A brief infographic, which can be seen in Figure 84, has been created to sum up the advance of these models through time as they have been described in this project. With the help of this content, the reader will be able to understand better how these models supposed improvements with respect to the corresponding predecessors and how much time it took to improve them.



TURBULENCE MODELS TIMELINE

1887 - Eddy Viscosity concept

1925 - Mixing length

1974 - Basic and low Re k-epsilon model

1988 - Wilcox k-omega model

1994 - Menter k-omega SST model

1994 - Spalart-Allmaras model

1995 - Realizable k-epsilon model

1997 - DES model

2002 - Menter k-omega SST DDES model

2004 - Menter k-omega SST SAS model

2006 - DDES model

2008 - IDDES model

Figure 85 - Turbulence Models timeline

8.2. turbulenceProperties file

In these files, the type of simulation must be specified, which will be LES and later on, the LES model will be selected, depending on which one is being used. Turbulence must be set to on and then, the coefficients that have been defined in the report must be included before running the simulation.

8.2.1. Spallart-Allmaras

For this particular case, both DDES and IDDES will use the same kind of file, and the only difference will be found in the declaration of the coefficients and the LESModel parameter, which will be IDDES instead of DDES.

```
simulationType  LES;

LES
{
       LESModel      SpalartAllmarasDDES;
        turbulence    on;
       printCoeffs    on;
       delta          maxDeltaxyz;

       maxDeltaxyzCoeffs
       {
           deltaCoeff 2;
       }

       SpalartAllmarasDDESCoeffs
       {
           alphaNut       1.5;
           Cb1            0.1355;
           Cb2            0.622;
           Cw2            0.3;
           Cw3            2.0;
           Cv1            7.1;
           Cv2            5.0;
           CDES           0.75;
           ck             0.07;
       }
}
```

If IDDES cases want to be considered, instead of using a delta of “maxDeltaxyz”, the function used is IDDESDelta, with the same deltaCoeff and a cw of 0.15, as default.

8.2.2. kwSST

In this case, which is very similar to the previous one, since no further coefficients need to be specified, the file is really simple and easy to understand. The same variables have to be defined with the exception of the Spallart-Allmaras coefficients.

```
simulationType  LES;

LES
{
       LESModel        kOmegaSSTDDES;
       turbulence      on;
       printCoeffs     on;
       delta           maxDeltaxyz;

       maxDeltaxyzCoeffs
       {
           deltaCoeff 2;
       }
}
```

8.3. blockMeshDict

In this file, the background mesh is defined. First of all, the vertices of the box must be defined (values are in meters already), and then the blocks as well. The division is also defined, which has been one of the most relevant parameters that allow coarser or finer meshes. Then, the different patches and walls to consider must be defined by indicating their faces and the type of element that is being defined. The example below is for the final cases, where only half of the plate is considered and the inlet is set at the leading edge of the plate.

```
convertToMeters 1;

vertices
(
    (-0.225 -0.6 -0.0075)
    (1.12 -0.6 -0.0075)
    (1.12 0.6 -0.0075)
    (-0.225 0.6 -0.0075)
    (-0.225 -0.6 0.84)
    (1.12 -0.6 0.84)
    (1.12 0.6 0.84)
    (-0.225 0.6 0.84)
);
blocks
(
    hex (0 1 2 3 4 5 6 7) (19 19 13) simpleGrading (1 1 1)
);
```

```
edges
(
);

boundary
(
    inlet
    {
        type patch;
        faces
        (
            (0 4 7 3)
        );
    }
    outlet
    {
        type patch;
        faces
        (
            (2 6 5 1)
        );
    }
    lateralWall
    {
        type symmetry;
        faces
        (
            (0 1 5 4)
            (3 7 6 2)
        );
    }
    upperWall
    {
        type symmetry;
        faces
        (
            (4 5 6 7)
        );
    }
    lowerWall
    {
        type symmetry;
        faces
        (
            (0 3 2 1)
        );
    }
);
mergePatchPairs
(
);
```

8.4. controlDict

In this dictionary file, the most important aspects regarding the simulation run are held. First of all, the application used to run is defined and so are the running times and the time step. Several other parameters can be also defined, such as the precision, format elements or adjust the time step manually to not exceed the maximum Courant number, which is physically defined to 1.

In this file, the user can also define the functions to be used during the run of the simulation. In this project's case, the functions created are thought to compute the mean value of all the variables, obtain the percentages of LES and RAS at each iteration and probe to obtain the frequency spectrum and do the Strouhal number analysis and computation.

```
application    pimpleFoam;

startFrom      startTime;

startTime      0;

stopAt         endTime;

endTime        0.1;

deltaT         1e-5;

writeControl   timeStep;

writeInterval  250;

purgeWrite     2;

writeFormat    binary;

writePrecision 10;

writeCompression uncompressed;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;

adjustTimeStep  no;

maxCo           1;
```

```
functions
{
    #include "readFields"
    #include "forceCoeffs"
    #include “probes”

    desField
    {
        type            DESModelRegions;
        functionObjectLibs ("libutilityFunctionObjects.so");
        log             true;
        enabled on;
        storeFilter on;
        outputControl outputTime;
    }

    yPlusLES
    {
        type               yPlus;
        functionObjectLibs ("libutilityFunctionObjects.so");
        outputControl      outputTime;
        log                no;
    }

    meanValues
    {
        type            fieldAverage;
        functionObjectLibs ("libfieldFunctionObjects.so");
        enabled                 true;
        cleanRestart            false;
        outputControl           outputTime;
        resetOnOutput           false;
        resetOnRestart          false;
        fields
        (
            U
            {
                mean        on;
                prime2Mean  off;
                base        time;
            }
            p
            {
                mean        on;
                prime2Mean  off;
                base        time;
            }
            k
            {
                mean        on;
                prime2Mean  off;
                base        time;
            }
            omega
            {
                mean        on;
                prime2Mean  off;
                base        time;
            }
            nut
            {
                mean        on;
                prime2Mean  off;
                base        time;
            }
            nuTilda
            {
                mean        on;
                prime2Mean  off;
                base        time;
            }
            yPlusLES
            {
                mean        on;
                prime2Mean  off;
                base        time;
            }
            DESModelRegions
            {
                mean        on;
                prime2Mean  off;
                base        time;
            }
        );
    }
}
```

8.5. forceCoeffs

This small file will enable the computation of the lift and drag coefficients for each simulation. The user must set the patches or elements about which the coefficient values should be computed, the direction of each magnitude according to the coordinate system of reference, the inlet velocity, the diameter of the cylinder and finally the projected surface of the patch, which is, in this case, the cylinder.

```
forceCoeffs1
{
    type        forceCoeffs;
    functionObjectLibs ("libforces.so");

    outputControl   timeStep;
    timeInterval    1;

    log         yes;
    patches     ( Cylinder );

    rhoName     rhoInf;
    rhoInf      1;

    liftDir     (0 0 1);
    dragDir     (1 0 0);

    CofR        (0 0 0);
    pitchAxis   (0 1 0);

    magUInf     26;

    lRef        0.120;
    Aref        0.0288;
}
```

8.6. fvSchemes

The fvSchemes dictionary in the system directory sets the numerical schemes for terms, such as derivatives in equations that are calculated during a simulation. The schemes for the different magnitudes are defined according to the parameters designed by the user.

```
ddtSchemes
{
    default backward;
}
gradSchemes
{
    default         Gauss linear;
    grad(nut)   cellLimited Gauss linear 1;
    grad(U)         cellLimited Gauss linear 1;
}
```

```
divSchemes
{
    default         none;
    div(phi,U)      Gauss LUST unlimitedGrad(U);
    div(phi,k)      Gauss limitedLinear 1;
    div(phi,nut) Gauss limitedLinear 1;
    div(phi,nuTilda) Gauss limitedLinear 1;
    div((nuEff*dev2(T(grad(U))))) Gauss linear;
}
laplacianSchemes
{
    default         Gauss linear limited corrected 0.33;
}
interpolationSchemes
{
    default         linear;
}
snGradSchemes
{
    default         limited corrected 0.33;
}
wallDist
{
    method meshWave;
}
```

8.7. fvSolution

The equation solvers, tolerances and algorithms are controlled from the fvSolution dictionary in the system directory. Each magnitude has its corresponding solver and, by introducing several parameters, the results can improve and the computation time can be reduced. The tolerances can also be indicated in this file, which can be a timesaver if the user just wants to reach a specific tolerance value to validate the results. Regarding the solver used, the user can also modify relevant parameters, such as correctors that make the simulation results more reliable.

```
solvers
{
    p
    {
        solver           GAMG;
        tolerance       1e-05;
        relTol           1e-03;
        smoother       GaussSeidel;
        nPreSweeps    0;
        nPostSweeps   2;
        cacheAgglomeration on;
        agglomerator     faceAreaPair;
        nCellsInCoarsestLevel 100;
        mergeLevels    1;
    }
```

```
   pFinal
    {
        $p;
        smoother      DICGaussSeidel;
        tolerance      1e-05;
        relTol          1e-02;
    }

   "(U|k|omega|B|nut|nuTilda)"
    {
        solver          smoothSolver;
        smoother      symGaussSeidel;
        tolerance      1e-05;
        relTol          1e-02;
    }

  "(U|k|omega|B|nut|nuTilda)Final"
    {
        $U;
        solver          smoothSolver;
        smoother      symGaussSeidel;
        tolerance      1e-05;
        relTol          1e-02;
    }
}

PIMPLE
{
    nCorrectors     1;
    nNonOrthogonalCorrectors 2;
}

relaxationFactors
{
    "U.*"           1;
    "nut.*"         1;
}
```

8.8. sampleDict

This dictionary file will help to compute the mean velocity by taking a density of points defined by the user and importing them. The surface studied will be a plane at the location desired by the user and around the axis defined. The most relevant parameters are the definition of the box from which it takes points to, the number of points desired (the more, the better) and the fields taken into account to do this extraction of values.

By taking equally distanced points on a defined region, the cell size can be taken into account and the results can be reliable enough rather than using other methods to compute these mean values.

```
interpolationScheme cellPoint;

setFormat           raw;

fields              ( UMean );

sets
(
    samplePlane_Uinlet
    {
        origin  (-0.225 -0.6 -0.125);
        type    array;
        axis    xyz;
        coordinateSystem
        {
            e1      (0 1 0);
            e2      (0 0 1);
            origin  (-0.225 -0.6 -0.125);
        }
        coordinateRotation
        {
                 type    axesRotation;
                 e1      (0 1 0);
                 e2      (0 0 1);
        }
        pointsDensity   (100 100 100);
        spanBox         (-0.225 0.6 0.6);
    }
);
```

8.9. snappyHexMeshDict

In this file, all of the parameters described in the previous sections of the document are detailed and some of them, which can cause trouble and have a relative importance, are highlighted with a short description.

```
castellatedMesh     true;
snap                true;
addLayers           true;

geometry
{
    Cylinder.stl
    {
        type triSurfaceMesh;
        name Cylinder;
    }
    Plate_round.stl
    {
        type triSurfaceMesh;
        name Plate_round;
    }
refinementBox
    {
        type searchableBox;
        min (-0.12 -0.3 0.0);
        max ( 0.3   0.3 0.3);
    }
};

castellatedMeshControls
{
    maxLocalCells   3000000;
    maxGlobalCells 10000000;
    minRefinementCells 10;
    maxLoadUnbalance 0.10;
    nCellsBetweenLevels 4;

features
    (
        {
            file "Cylinder.eMesh";
            level 7;
        }

       {
            file "Plate_round.eMesh";
            level 7;
        }
    );
    refinementSurfaces
    {
        Cylinder
        {
          level (6 7);
            patchInfo
            {
               type wall;
            }
        }
      Plate_round
        {
            level (6 7);
            patchInfo
            {
                type wall;
            }
        }
}
resolveFeatureAngle 95;

refinementRegions
    {
        refinementBox
        {
            mode inside;
            levels ((4 5));
        }
    }

    locationInMesh (0 0 0.4); //Over the cylinder
    allowFreeStandingZoneFaces true;
}
snapControls
{
    nSmoothPatch 5;
    tolerance 2.0;
    nSolveIter 300; //- Number of mesh displacement relaxation iterations.
    nRelaxIter 5;

    nFeatureSnapIter 10;

    implicitFeatureSnap false;
    explicitFeatureSnap true;
    multiRegionFeatureSnap false;
}

addLayersControls
{
    relativeSizes true;

    layers
    {
       Cylinder
       {
             nSurfaceLayers 4;
       }
       Plate_round
       {
             nSurfaceLayers 4;
       }
     }
    expansionRatio 1.15; // Expansion factor for layer mesh
    finalLayerThickness 0.9; // Wanted thickness of final added cell layer.
    minThickness 0.05; // Minimum thickness of cell layer
    nGrow 1;
    featureAngle 180;
    slipFeatureAngle 70;
    nRelaxIter 3;
    nSmoothSurfaceNormals 1;
    nSmoothNormals 3;
    nSmoothThickness 10;
    maxFaceThicknessRatio 0.9;
    maxThicknessToMedialRatio 0.9;
    minMedianAxisAngle 130;
    nBufferCellsNoExtrude 0;
    nLayerIter 50;
}

meshQualityControls
{
    maxNonOrtho 65;
    maxBoundarySkewness 20;
    maxInternalSkewness 4;
    maxConcave 80;
    minVol 1e-13;
    minArea 1e-13;
    minTetQuality 1e-30;
    minTwist 0.05;
    minDeterminant 0.001;
    minFaceWeight 0.02;
    minVolRatio 0.01;
    minTriangleTwist -1;
    nSmoothScale 4;
    errorReduction 0.75;
}
// Advanced

writeFlags
(
    scalarLevels
    layerSets
    layerFields     // write volScalarField for layer coverage
);

mergeTolerance 1e-6;
```