



UNIVERSITAT  
ROVIRA i VIRGILI

Universitat Rovira i Virgili

Escola Tècnica Superior d'Enginyeria Química

**unir**

LA UNIVERSIDAD  
EN INTERNET

Universidad Internacional de La Rioja

Escuela Superior de Ingeniería y tecnología

Máster Interuniversitario en Mecánica de Fluidos  
Computacional

# Estudio de la implementación de OpenFOAM en el Sector de la Automoción

Trabajo fin de estudio presentado por:	Jordi Lanuza Fabregat
Tipo de trabajo:	Simulación Numérica
Director/a:	Àlex Fabregat
Fecha:	20/07/2022

## Resumen

El presente trabajo de fin de máster se enmarca en un proyecto que el autor desarrolla en su entorno profesional como ingeniero de CFD en una empresa del sector de la automoción.

El objetivo consiste en determinar la viabilidad de la implementación de una plataforma de CFD de código abierto (OpenFOAM) que pueda complementar o, eventualmente, reemplazar a las actuales soluciones comerciales (STAR-CCM+, ANSYS) para la simulación y caracterización de las propiedades aerodinámicas de vehículos, lo que permita poder aumentar el potencial de simulación y su competitividad en el mercado.

Aunque ya son notables los esfuerzos dentro de este sector para facilitar esta transición hacia soluciones de código abierto, cada empresa necesita adaptar las soluciones computacionales de acuerdo con sus necesidades.

El estudio presenta y discute los resultados preliminares de la validación de OpenFOAM como plataforma CFD para la resolución de un vehículo de referencia usando geometrías simplificadas en las que se omiten elementos como retrovisores o ruedas. Los resultados numéricos obtenidos se han comparado con resultados experimentales para condiciones de flujo externo y geometrías similares. Posteriormente, el análisis se extiende a una geometría completa que permita una caracterización más precisa. Las distintas mallas computacionales, generadas con la herramienta ANSA, se han comparado, permitiendo el análisis del impacto de la resolución y la cuantificación de errores con respecto a los resultados experimentales para configuraciones similares.

Finalmente, se comparan diferentes parámetros aerodinámicos obtenidos por el autor con los resultados de un caso de referencia en el sector automovilístico. La similitud entre los resultados presentes y los reportados por los distintos equipos provenientes de la industria y la academia sugiere que la herramienta OpenFOAM se posiciona favorablemente como una alternativa viable a las soluciones comerciales.

**Palabras clave:** OpenFOAM, Aerodinámica, Automoción, CFD, DrivAer, ANSA

## Abstract

This master's thesis is part of a project that the author develops in his professional environment as CFD engineer in an automotive company.

The objective is to determine the feasibility of implementing an open source CFD platform (OpenFOAM) that could complement or, eventually, replace the current commercial solutions (STAR-CCM+, ANSYS) for the simulation and characterization of the vehicle's aerodynamics properties, leading to an increase of the simulation's potential and competitiveness in the market.

Although there have been significant efforts in the sector to facilitate this transition towards open-source solutions, each company needs to adapt the computing solutions according to their needs.

The study presents and discusses preliminary results of the validation of OpenFOAM as a CFD platform for the resolution of a bluff body in which some elements, such as mirrors or wheels, are omitted. The numerical results obtained are then compared with experimental results for similar external flow conditions and geometries. Later, the analysis is extended to a more complete geometry for a more precise characterization. Different computational meshes, generated with ANSA, are compared, allowing the analysis of the impact of the resolution and the error quantification with respect to experimental results for similar configurations.

Finally different aerodynamics parameters obtained are compared with the results of a benchmark case in the automotive industry. The similarity between the results and those reported by different teams from industry and universities suggests that OpenFOAM is favourably positioned as a viable alternative to commercial solutions.

**Keywords:** OpenFOAM, Aerodynamics, Automotive, CFD, DrivAer, ANSA

## Glosario de símbolos

Re	-	Número de Reynolds
$\mathbf{u}$	m/s	Vector velocidad ( $u_x, u_y, u_z$ )
U	m/s	Magnitud de la velocidad
$u, u_x$	m/s	Velocidad en x
$v, u_y$	m/s	Velocidad en y
$w, u_z$	m/s	Velocidad en z
p	Pa	Presión estática
q	Pa	Presión dinámica
I	-	Matriz identidad
$\sigma$	Pa	Tensor de esfuerzos
$\sigma_v$	Pa	Tensor de esfuerzos viscosos
$\epsilon$	$s^{-1}$	Velocidad de deformación
$\nu$	$m^2/s$	Viscosidad cinemática
$\mu$	$Kg \cdot m^{-1} \cdot s^{-1}$	Viscosidad dinámica
e	J/kg	Energía interna por unidad de masa
L	m	Longitud característica
$\rho$	$Kg/m^3$	Densidad
g	$m/s^2$	Constante de la gravedad
L	N	Fuerza de sustentación (Lift)
D	N	Fuerza de resistencia (Drag)
$C_L$	-	Coefficiente de sustentación
$C_D$	-	Coefficiente de resistencia
A	$m^2$	Área frontal del automóvil

$\delta$	m	Grosor de capa límite
$u_\tau$	m/s	Velocidad de fricción
$y^+$	-	Distancia adimensional normal a la pared
$U^+$	-	Velocidad adimensional
Pe	-	Número de Peclet
k	$m^2/s^2$	Energía cinética turbulenta
$\omega$	$s^{-1}$	Ratio específico de disipación ( $\omega = \epsilon/k$ )
$\epsilon$	$m^2/s^3$	Disipación turbulenta

## Abreviaciones

RANS	<i>Reynolds Averaged Navier-Stokes Equations</i>
TUM	Technical University of Munich
CFD	Dinámica de fluidos computacional ( <i>Computational Fluid Dynamics</i> )
OEM	<i>Original Equipment Manufacturer</i>
DES	<i>Detached Eddy Simulation</i>
LES	<i>Large Eddy Simulation</i>
DNS	<i>Direct Numerical Simulation</i>
TFM	Trabajo Final de Máster
SGS	Subescala ( <i>Subgrid-scale</i> )
RH	Ride Height (Altura del suelo a la parte inferior del coche)

## Índice de contenidos

1. Introducción .....	15
1.1. Motivación .....	15
1.2. Planteamiento del trabajo .....	16
1.3. Estructura del trabajo .....	17
2. Contexto y Estado del arte .....	18
2.1. Introducción Teórica .....	18
2.1.1. Ecuaciones de gobierno.....	18
2.1.2. Flujos viscosos .....	19
2.1.3. Flujos turbulentos.....	21
2.1.4. Tipos de simulación .....	22
2.1.5. Capa límite .....	27
2.1.6. Resistencia (Drag).....	30
2.1.7. Coeficientes de Lift/Drag.....	32
2.2. Tipos de simulación en el sector.....	34
3. Objetivos y Metodología de trabajo .....	35
3.1. Objetivo general.....	35
3.2. Objetivos específicos .....	35
3.3. Metodología de trabajo .....	35
4. OpenFOAM.....	37
4.1. Introducción.....	37
4.2. Solvers.....	38
4.2.1. potentialFoam .....	38
4.2.2. simpleFoam .....	38
4.3. Esquemas Numéricos.....	38

4.4. Modelo de turbulencia .....	40
5. Introducción de los Modelos.....	42
6. Modelo Simplificado .....	43
6.1. Modelo de turbulencia .....	43
6.1. Modelo de turbulencia .....	44
6.2. Malla .....	45
6.3. SetUp.....	48
6.4. Resultados.....	50
6.4.1. Comparativa entre mallas y esquemas .....	51
6.4.2. Comparativa resultados experimentales .....	53
7. Modelo de DrivAer .....	58
7.1. Geometría .....	58
7.2. Condiciones de la simulación.....	59
7.3. Proceso de mallado.....	59
7.4. SetUp.....	62
7.5. Resultados.....	63
7.5.1. Modelo de DrivAer sin ruedas .....	63
7.5.2. Modelo de DrivAer con ruedas.....	69
7.5.3. Modelo segunda conferencia AutoCFD.....	73
8. Conclusiones y Trabajo futuro .....	75
Referencias bibliográficas.....	76
9. ANEXOS .....	81
Anexo A. Figuras adicionales .....	81
Anexo B. Información Adicional .....	85
a. Estructura carpetas OpenFOAM .....	85

b.	Ampliación esquemas numéricos usados .....	87
i.	gradSchemes .....	87
ii.	divSchemes.....	89
iii.	laplacianSchemes .....	91
c.	Ecuaciones URANS .....	92
d.	Modelo de Turbulencia k- $\omega$ SST.....	93
e.	Tabla resumen simulaciones modelo simplificado .....	94
	Anexo C. Ficheros OpenFOAM .....	95
a.	0.....	95
b.	constant.....	102
c.	system .....	103

## Índice de figuras

Figura 2.1 Esfuerzos en un elemento del fluido ((Versteeg & Malalasekera, 1995), Fig 2.3 - p15).....	20
Figura 2.2 Distribución del espectro de energía turbulenta (a) y de la disipación turbulenta (b) ((Leschziner, 2015), Fig 2.4 - p13) .....	22
Figura 2.3 Promediado temporal de la velocidad para un flujo estadísticamente estacionario (izqda.) y para un flujo no estacionario (dcha.) ((Ferziger et al., 2002), Fig 10.24 - p397) .....	23
Figura 2.4 Señal de una velocidad turbulenta $U(t)$ y la misma señal promediada con dos ventanas de tiempo ((Leschziner, 2015), Fig 3.1 – p35).....	24
Figura 2.5 Flujo esquemático de la capa límite sobre una placa plana semi-infinita ((Frank M. White, 2008) , Fig 9.10 - p383) .....	28
Figura 2.6 Perfil promedio de velocidad en una capa límite turbulenta ((Frank M. White, 2006), Fig 12.8 - p587).....	29
Figura 2.7 Resistencia aerodinámica – Explicación gráfica simplificada (Fuente: Tatra - The Legacy of Hans Ledwinka) .....	31
Figura 2.8 Perfil capa límite sometido a un gradiente adverso de presión (Fuente: <a href="http://www.examhill.com/wp-content/uploads/2016/11/Figure-12.png">http://www.examhill.com/wp-content/uploads/2016/11/Figure-12.png</a> ).....	32
Figura 2.9 Lightyear 0 (Fuente: <a href="https://lightyear.one/lightyear-0">https://lightyear.one/lightyear-0</a> ) .....	33
Figura 2.10 Modelo Mercedes VISION EQXX (Fuente: <a href="https://www.diariomotor.com/">https://www.diariomotor.com/</a> ).....	33
Figura 2.11 Modelo Mercedes EQS (Fuente: <a href="https://www.mercedes-benz.es">https://www.mercedes-benz.es</a> ).....	33
Figura 4.1 Estructura de carpetas en OpenFOAM (inspirado en (Bastian Nebenführ, 2010), Fig 2.6 – p16).....	37
Figura 4.2 Distribución de $\phi$ cerca de dos fuentes a diferentes número de Peclet. (a) Difusión pura i (b) Difusión i convección ((Versteeg & Malalasekera, 1995) ,Fig 5.9 - p144).....	39
Figura 4.3 Malla cerca de la pared para un flujo resuelto cerca de la pared (dcha.) y usando funciones de pared (izqda) ((Bastian Nebenführ, 2010) , p14).....	41

Figura 5.1 Layout del Túnel de viento de 2.4 x 1.8 m de baja velocidad (Fuente: <a href="https://www.cranfield.ac.uk">https://www.cranfield.ac.uk</a> ) .....	42
Figura 5.2 Sección de test con el modelo durante los ensayos .....	42
Figura 5.3 Modelos DrivAer (Fuente: (TUM School of Engineering and Design, n.d.)).....	42
Figura 6.1 Dibujo esquemático del modelo simplificado ((Lanuza, 2016), Fig4.3, p26) .....	43
Figura 6.2 Dimensiones del dominio (Siemens, 2020) .....	44
Figura 6.3 Condiciones de contorno del dominio .....	44
Figura 6.4 Geometría modelo simplificado .....	45
Figura 6.5 Set Up malla superficial .....	46
Figura 6.6 Set Up Malla capa límite .....	46
Figura 6.7 Volúmenes de refinamiento .....	46
Figura 6.8 Malla Modelo Simplificado – Difusor 4º y RH=40mm .....	47
Figura 6.9 Skewness.....	47
Figura 6.10 Non-orthogonality .....	47
Figura 6.11 Aspect Ratio .....	47
Figura 6.12 Jacobian .....	47
Figura 6.13 fvSchemes.....	49
Figura 6.14 fvSolution.....	49
Figura 6.15 Residuales – Modelo Difusor 4º, RH=80mm y <i>linearUpwindV</i> .....	50
Figura 6.16 Residuales – Modelo Difusor 4º, RH=60mm y <i>linearUpwind</i> .....	50
Figura 6.17 Residuales – Modelo Difusor 10º, RH=40mm y <i>linearUpwindV</i> .....	50
Figura 6.18 Residuales – Modelo Difusor 10º, RH=40mm y <i>linearUpwind</i> .....	50
Figura 6.19 Contorno de $y^+$ - Modelo Simplificado – Difusor 10º y RH=60mm .....	51
Figura 6.20 Residuales – Modelo Difusor 10º, RH=40mm, malla tetra, <i>gradScheme(cellLimited leastsquares)</i> .....	51

Figura 6.21 Residuales – Modelo Difusor 10º, RH=40mm, malla tetra, gradScheme(Gauss Linear).....	51
Figura 6.22 Residuales – Modelo Difusor 10º, RH=60mm, malla tetra, gradScheme(Gauss Linear).....	52
Figura 6.23 Residuales – Modelo Difusor 10º, RH=60mm, malla tetra, gradScheme(Gauss Linear), con 2 correctores de non-ortogonalidad .....	52
Figura 6.24 Residuales – Modelo Difusor 10º, RH=100mm, malla tetra, gradScheme(Gauss Linear).....	52
Figura 6.25 Residuales – Modelo Difusor 10º, RH=100mm, malla Hexa, gradScheme(Gauss Linear).....	52
Figura 6.26 Residuals – Modelo Difusor 10º, RH=100mm, divScheme (bounded Gauss vanLeer).....	53
Figura 6.27 Residuals – Modelo Difusor 10º, RH=100mm, divScheme(bounded Gauss linearUpwind grad(k/omega)) .....	53
Figura 6.28 Comparativa resultados $C_L$ 2015 vs 2016 ((Lanuza, 2016) , Fig 5.3 – p38).....	54
Figura 6.29 Comparativa resultados $C_D$ 2015 vs 2016 ((Lanuza, 2016) , Fig 5.4 – p39) .....	54
Figura 6.30 Comparativa simulación con resultados experimentales - $C_L$ .....	54
Figura 6.31 Comparativa simulación con resultados experimentales - $C_D$ .....	54
Figura 6.32 Comparativa linearUpwind vs linearUpwindV - $C_L$ .....	55
Figura 6.33 Comparativa linearUpwind vs linearUpwindV - $C_D$ .....	55
Figura 6.34 Contorno de velocidades en $y=0$ – linearUpwindV .....	55
Figura 6.35 Contorno de velocidades en $y=0$ - linearUpwind .....	56
Figura 6.36 Contorno de vorticidad en $x=0.89m$ , $x=0.95m$ y $x=1.07m$ .....	57
Figura 6.37 Isosuperficie de la energía cinética turbulenta (k).....	57
Figura 7.1 Modelo DrivAer NotchBack – Con ruedas .....	58
Figura 7.2 Modelo DrivAer NotchBack – Sin ruedas .....	58
Figura 7.3 Zona unión neumático-asfalto.....	58

Figura 7.4 Volúmenes virtuales – Mallado con offset de las superficies .....	60
Figura 7.5 Contorno de $y^+$ malla $y^+\approx 1$ .....	61
Figura 7.6 Malla zona capa límite $y^+\approx 1$ .....	61
Figura 7.7 Contorno de $y^+$ malla $y^+\approx 30$ .....	61
Figura 7.8 Malla zona capa límite $y^+\approx 30$ .....	61
Figura 7.9 Malla – Zoom zona de contacto neumático y asfalto .....	62
Figura 7.10 Refinado de la malla en la zona de contacto y estela de la rueda. ....	62
Figura 7.11 Malla de baja calidad unión llanta-neumático .....	63
Figura 7.12 Malla Modelo 1 - DrivAer sin ruedas.....	64
Figura 7.13 Historial de convergencia Modelo 1 - DrivAer sin ruedas.....	64
Figura 7.14 Malla Modelo 2 - DrivAer sin ruedas.....	65
Figura 7.15 Historial de convergencia Modelo 2 - DrivAer sin ruedas.....	65
Figura 7.16 Contorno de velocidad Modelo 2 en la iteración 4000, 4500 y 5000 - DrivAer sin ruedas .....	66
Figura 7.17 Malla Modelo 3 - DrivAer sin ruedas.....	67
Figura 7.18 Malla Modelo 4 - DrivAer sin ruedas.....	68
Figura 7.19 Gráfico $C_L$ Modelo final – DrivAer con ruedas – últimas 2000 iteraciones .....	70
Figura 7.20 Gráfico $C_D$ Modelo final – DrivAer con ruedas – últimas 1000 iteraciones.....	70
Figura 7.21 Contorno de velocidades Modelo Final (iteración 4000, 4500 y 5000) – DrivAer con ruedas .....	71
Figura 7.22 Contorno de velocidades Modelo Final (zoom final luneta trasera) – DrivAer con ruedas .....	71
Figura 7.23 Isosuperficie del coeficiente de presión total 0 ( $C_{pT}=0$ ) – DrivAer con ruedas .....	72
Figura 7.24 Resumen resultados Workshop AutoCFD - $C_D$ (Hupertz et al., 2021) .....	73
Figura 7.25 Resumen resultados Workshop AutoCFD – $C_{Lr}$ y $C_{Lf}$ (Hupertz et al., 2021) .....	73
Figura 7.26 Historial de convergencia modelo AutoCFD.....	74

Figura 7.27 Valor de los coeficientes últimas 2300 iteraciones .....	74
Figura 9.1 Gráfico $C_L$ Modelo Simplificado – Difusor 4º y RH=40mm – últimas 2000 iteraciones .....	81
Figura 9.2 Gráfico $C_D$ Modelo Simplificado – Difusor 4º y RH=40mm – últimas 1000 iteraciones .....	81
Figura 9.3 Gráfico $C_L$ Modelo 2 – DrivAer sin ruedas – últimas 2000 iteraciones.....	81
Figura 9.4 Gráfico $C_D$ Modelo 2 – DrivAer sin ruedas – últimas 1000 iteraciones .....	81
Figura 9.5 Gráfico $C_L$ Modelo 3 – DrivAer sin ruedas – últimas 2000 iteraciones.....	81
Figura 9.6 Gráfico $C_D$ Modelo 3 – DrivAer sin ruedas – últimas 1000 iteraciones .....	81
Figura 9.7 Detalle caja de refinamiento – Resaltado Size box 2 .....	82
Figura 9.8 Skewness Modelo 4 – Modelo DrivAer sin ruedas.....	82
Figura 9.9 Non-orthogonality Modelo 4 – Modelo DrivAer sin ruedas.....	82
Figura 9.10 Gráfico $C_L$ Modelo 4 – DrivAer sin ruedas – últimas 2000 iteraciones.....	82
Figura 9.11 Gráfico $C_D$ Modelo 4 – DrivAer sin ruedas – últimas 1000 iteraciones .....	82
Figura 9.12 Historial de convergencia modelo 4 – DrivAer sin ruedas .....	83
Figura 9.13 Historial de convergencia modelo final – DrivAer con ruedas.....	83
Figura 9.14 Skewness modelo final – DrivAer con ruedas .....	83
Figura 9.15 Non-orthogonality modelo final – DrivAer con ruedas .....	83
Figura 9.16 Sección malla Y0 modelo final – DrivAer con ruedas .....	84
Figura 9.17 Estructura de carpetas en OpenFOAM (inspirado en (Bastian Nebenführ, 2010), Fig 2.6 – p16) .....	85
Figura 9.18 Discretizado 2D – Malla estructurada y alineada con el flujo .....	87
Figura 9.19 Discretizado 2D – Malla no estructurada 2D.....	88
Figura 9.20 Esquemas TVD – Representación diferentes funciones limitadoras $\Psi(\mathbf{r})$ ((Versteeg & Malalasekera, 1995) , Fig 5.25 - p171) .....	90

## Índice de tablas

Tabla 1. <i>Esquemas numéricos usados.</i> .....	40
Tabla 2. <i>Funciones de pared usadas</i> .....	41
Tabla 3. <i>Condiciones de contorno en las paredes – Modelo simplificado</i> .....	49
Tabla 4. <i>Casos simulados – Modelo simplificado</i> .....	53
Tabla 5. <i>Valores de malla modelo 3 - Modelo DrivAer sin ruedas.</i> .....	66
Tabla 6. <i>Valores de malla modelo 4 - Modelo DrivAer sin ruedas.</i> .....	67
Tabla 7. <i>Esquemas numéricos simulación 1 - Modelo DrivAer con ruedas.</i> .....	69
Tabla 8. <i>Valores de malla modelo final - DrivAer con ruedas.</i> .....	83
Tabla 9. <i>Unidades ficheros condiciones iniciales (Ej. U)</i> .....	85
Tabla 10. <i>Esquemas usados</i> .....	90
Tabla 11. <i>Constantes del modelo SST <math>k-\omega</math></i> .....	93
Tabla 12. <i>Casos simulados – Modelo simplificado</i> .....	94

## 1. Introducción

En este primer apartado se pretende presentar una idea clara de este Trabajo Final de Máster, la motivación de éste, el planteamiento y la estructura del trabajo.

### 1.1. Motivación

El objetivo de este trabajo consiste en hallar un proceso robusto para la simulación CFD de vehículos, principalmente automóviles, usando herramientas CFD de código libre que permitan una mayor customización y una reducción costes. En este trabajo se pretende centrarse en el estudio de un proceso para un tipo de simulación muy recurrente, y cada día con mayor relevancia para el sector, la aerodinámica externa de los vehículos. El diseño de vehículos más eficientes es particularmente relevante en el contexto actual, con normativas de emisiones cada vez más restrictivas.

La resistencia al aire de los vehículos en entornos no urbanos implica el consumo del 50% de la energía mecánica producida por la unidad de potencia (Agarwal, 2013). Por tanto, la optimización de la aerodinámica del vehículo, sobre todo en los vehículos eléctricos, donde el limitante en el rango de éstos cuando salen de entornos urbanos es la resistencia al aire, cobra aún mayor importancia. El número de simulaciones, y el detalle de éstas, se ha incrementado de forma significativa. En este contexto, las empresas especializadas del sector, ya sean OEMs o consultorías, compiten por aumentar su capacidad para analizar el mayor número de geometrías o condiciones de trabajo posibles, garantizando a la vez, unos costes computacionales competitivos.

Para poder incrementar el potencial de simulaciones se debe invertir en una mayor potencia de cálculo, aumentar el personal y/o, en el caso de softwares comerciales, la compra de nuevas licencias. Todo esto puede derivar en un incremento importante de los costes. Por tanto, implementar un software gratuito como OpenFOAM, para la realización de este tipo de simulaciones, implica una reducción significativa de los mismos.

OpenFOAM es un software libre de simulación que permite una gran adaptabilidad al usuario, debido a que todo el código es público y se puede modificar en cualquier momento. Sin embargo, no tiene las virtudes de los códigos comerciales, los cuales se han adaptado para generar procesos más robustos que mejoran la convergencia, incluso con mallas de

peor calidad. En OpenFOAM, el trabajo de encontrar el set up adecuado, para el tipo de simulación que se lleva a cabo, recae en el usuario, y aún así, sigue siendo muy dependiente de la calidad de malla. Según experiencias de usuarios, pequeños cambios de malla o geometría pueden afectar muy significativamente a la convergencia de la simulación (Bastian Nebenführ, 2010).

Se tiene conocimiento de empresas que han conseguido hacer un set up en OpenFOAM, con procesos y tiempos de computación comparables a los de los softwares comerciales, como, por ejemplo, la empresa Engys con su software HELIX, el cual venden actualmente junto con sus servicios.

Por lo tanto, se pretende, en esta primera etapa, hacer un estudio de la viabilidad de poder implementar esta herramienta para simulaciones de aerodinámica externa, sin incluir la rotación de ruedas y con los sistemas de refrigeración y entradas al motor cerradas, dejando para una segunda etapa la inclusión de estos elementos.

Las simulaciones se llevarán a cabo en uno de los superordenadores de la empresa, el cual tiene instalada la versión 7 de OpenFOAM, perteneciente a "OpenFOAM Foundation", y la versión 21.12, perteneciente a "esi-OpenCFD".

## 1.2.Planteamiento del trabajo

Se propone realizar un estudio por etapas, añadiendo complejidad a la simulación en cada etapa. Primero se empezará con un coche simplificado, sin ruedas ni retrovisores. A continuación, se usarán los diferentes modelos de DrivAer proporcionados por la Universidad Tecnológica de Múnich (TUM – Technical University of Munich) (TUM School of Engineering and Design, n.d.), los cuales tienen diferentes geometrías con diferentes niveles de detalle.

En este primer estudio nos centraremos en simulaciones RANS (*Reynolds Averaged Navier-Stokes equations*) con el modelo de turbulencia k- $\omega$  SST, utilizando ANSA para realizar el tratamiento de las superficies, preparados de modelo y mallados.

Basándonos en literatura, se escogerán y se hará un análisis de los diferentes esquemas numéricos.

### 1.3. Estructura del trabajo

Primero se hace una introducción teórica de las ecuaciones, y conceptos importantes, a tener en cuenta a la hora de la resolución del problema. Se presentan las ecuaciones que rigen la física del problema, la ecuación de continuidad y las ecuaciones de Navier-Stokes. A continuación, se introducen los conceptos de viscosidad, turbulencia y capa límite, así como los tipos de simulación más usadas en el sector. Finalmente, se presentarán los conceptos de sustentación y resistencia al aire.

Una vez hecha la introducción teórica, se presentará el software de simulación, los diferentes esquemas numéricos, modelo de turbulencia y tratamiento de pared usados para la resolución del problema.

Una vez introducidos los modelos y alguna de las simplificaciones asumidas, se presentan los modelos que se van a simular. Primero, se analiza un modelo como de coche sin ruedas ni retrovisores. Los resultados se comparan con valores experimentales obtenidos del túnel de viento de la universidad de Cranfield. Una vez se ha obtenido un set up que funcione en varias configuraciones, se empieza a trabajar con un modelo más complejo y realista de coche. Los resultados obtenidos se comparan con resultados obtenidos por otros autores. Para ambos modelos, se realiza la malla con ANSA y se va iterando hasta cumplir los criterios de malla requeridos por OpenFOAM.

Finalmente, se enumeran las conclusiones principales obtenidas y se definen próximos pasos a realizar para cumplir con el objetivo final.

## 2. Contexto y Estado del arte

Actualmente, el CFD es una herramienta ampliamente usada en el sector de la automoción ya que se puede utilizar desde las primeras fases de diseño, permitiendo reducir el número de modelos a escala fabricados para su ensayo en túneles de viento, y, a la vez, aumentando el número de iteraciones que se pueden realizar durante el proceso de diseño. Además, nos permite empezar a trabajar en la aerodinámica del vehículo en una etapa muy inicial del diseño, evaluando el efecto de los cambios que puedan tener un impacto sobre la misma, y así, acabar consiguiendo un vehículo mucho más eficiente.

Como se ha comentado en el capítulo anterior, en este trabajo nos centraremos en la simulación de la aerodinámica externa del vehículo. Por ello, clasificaremos el tipo de simulaciones que se suelen realizar en el sector para este tipo de problema en particular.

Notar, que el gran limitante, al escoger entre uno u otro tipo de simulación, es la potencia de cálculo disponible.

### 2.1. Introducción Teórica

La dinámica de fluidos computacional (CFD) es el estudio mediante simulaciones por ordenador del movimiento de un fluido, la transferencia de calor y fenómenos asociados, como reacciones químicas (Versteeg & Malalasekera, 1995).

#### 2.1.1. Ecuaciones de gobierno

Las ecuaciones, que trata de resolver la mecánica de fluidos computacional, parten de las leyes físicas de conservación:

- Conservación de la masa.
- Conservación de cantidad de movimiento (segunda ley de Newton). Indica que la ratio con el que cambia la cantidad de momento del fluido debe ser igual a la suma de fuerzas ejercidas sobre éste.
- Conservación de la energía (primera ley de la termodinámica)

La dinámica de fluidos computacional está regida por las siguientes ecuaciones diferenciales:

- Ecuación de la continuidad

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \mathbf{u}) = 0 \quad \text{Ecuación 2-1}$$

$$\nabla(\rho \mathbf{u}) = \frac{\partial \rho u_i}{\partial x_i} = \frac{\partial \rho u_x}{\partial x} + \frac{\partial \rho u_y}{\partial y} + \frac{\partial \rho u_z}{\partial z} \quad \text{Ecuación 2-2}$$

$$\mathbf{u} = (u_x, u_y, u_z) = (u, v, w) \quad \text{Ecuación 2-3}$$

- Ecuaciones de la conservación de la cantidad de movimiento

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = \frac{\partial \sigma_{ij}}{\partial x_j} - \rho g \vec{k} \quad \text{Ecuación 2-4}$$

$$\sigma_{ij} = -p \mathbf{I} + \sigma_{vij} \quad \text{Ecuación 2-5}$$

Siendo  $\sigma_{vij}$  el tensor de esfuerzos viscosos.

Se tiene una ecuación de la conservación de la cantidad de movimiento para cada dirección (x, y, z).

- Ecuación de la conservación de energía

$$\frac{\partial \varepsilon}{\partial t} + \frac{\partial \rho \varepsilon u_j}{\partial x_j} = \frac{\partial u_i \sigma_{ij}}{\partial x_j} - \rho g \vec{k} \cdot \mathbf{u} + \dot{Q} \quad \text{Ecuación 2-6}$$

$$\varepsilon = \rho e + q \quad ; \quad q = \frac{1}{2} \rho U^2 \quad \text{Ecuación 2-7}$$

Siendo e la energía interna por unidad de masa.

### 2.1.2. Flujos viscosos

Como hemos podido ver, en las ecuaciones de cantidad de movimiento aparece el tensor de esfuerzos viscosos. Si obtenemos este tensor a partir del esquema de la Figura 2.1:

$$\sigma_{ij} = -p \mathbf{I} + \sigma_{vij} = \begin{pmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{pmatrix} \quad \text{Ecuación 2-8}$$

El tensor de esfuerzos es un tensor simétrico  $\sigma_{ij} = \sigma_{ji}$ , lo cual es necesario para satisfacer el equilibrio de momentos en los tres ejes del elemento (Frank M. White, 2006).

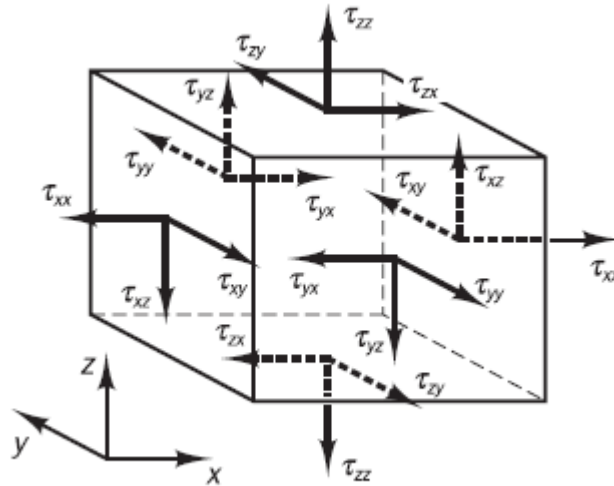


Figura 2.1 Esfuerzos en un elemento del fluido ((Versteeg & Malalasekera, 1995), Fig 2.3 - p15)

En el caso de que el flujo estuviera en reposo, los esfuerzos viscosos desaparecen y el tensor de esfuerzos queda como  $\sigma_{ij} = -p\delta_{ij}$  (siendo  $\delta_{ij}$  la función delta de Kronecker), donde  $\delta_{ij}$  es 1 si  $i=j$  y 0 en los demás casos.

Por tanto, nos queda por definir el tensor de esfuerzos viscosos. Este tensor es una función lineal de las velocidades de deformación ( $\epsilon_{ij}$ ) (Frank M. White, 2006). La ley experimental de Navier-Poisson establece la siguiente relación para un fluido isótropo y homogéneo entre el tensor de esfuerzos viscosos y el tensor de deformación lineal:

$$\sigma_{vij} = 2\mu \left( \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) + \lambda \frac{\partial u_i}{\partial x_i} \delta_{ij} \quad \text{Ecuación 2-9}$$

En esta ley experimental, aparece la viscosidad aparente ( $\lambda$ ) multiplicando la divergencia de la velocidad. Esto implica que, para flujos incompresibles, este término desaparecerá, quedando el tensor de esfuerzos viscos descrito de la siguiente forma:

$$\sigma_{vij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad \text{Ecuación 2-10}$$

Con esto, la ecuación de la conservación de cantidad de movimiento queda:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = - \frac{dp}{dx_i} + \frac{\partial}{\partial x_j} \left( \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) - \rho g \vec{k} \quad \text{Ecuación 2-11}$$

### 2.1.3. Flujos turbulentos

Cuando se prepara una simulación, es importante tener claro qué tipo de flujo estamos resolviendo. El parámetro adimensional que nos sirve para diferenciar entre dos tipos de comportamientos diferenciados del flujo es el número de Reynolds (Frank M. White, 2008). El número de Reynolds es la ratio entre fuerzas inerciales del fluido y las viscosas. Se define como:

$$Re = \frac{UL}{\nu} \quad \text{Ecuación 2-12}$$

Flujos donde las fuerzas viscosas predominan sobre las inerciales, se consideran flujos laminares. Están caracterizados por ser un flujo ordenado, donde no hay presencia de remolinos y/o vórtices y no hay una componente de la velocidad normal a la componente principal del fluido (Leschziner, 2015).

Los flujos turbulentos se definen como caóticos, pero, al igual que los flujos laminares, están regidos por las ecuaciones de Navier-Stokes. Por tanto, si se definen correctamente y de forma precisa las condiciones iniciales para la resolución de las ecuaciones, se puede predecir el comportamiento del flujo (Leschziner, 2015).

Una de las propiedades principales de la turbulencia es la vorticidad. Se puede entender la turbulencia como un conjunto de remolinos o vórtices de diferentes tamaños, velocidades y sentidos de rotación. Estos remolinos interactúan entre ellos deformándose y estirándose. Esta interacción lleva a una reducción del diámetro de los remolinos, por tanto, aumentando su velocidad de rotación (Leschziner, 2015). La energía turbulenta se distribuye a lo largo de todas las escalas de remolinos, siendo  $\Lambda$  la escala más grande de remolino y  $\eta$  la más pequeña. En flujos turbulentos completamente desarrollados, la distribución de la energía a lo largo de las escalas tiene un comportamiento universal (Figura 2.2) (Leschziner, 2015). Como podemos ver en la distribución de la energía turbulenta, se diferencian tres zonas:

- Zona A: Es la zona en la que se extrae energía del campo medio y se transfiere a las escalas de menor tamaño (concepto de cascada).
- Zona B: Se denomina el rango inercial. En esta zona los remolinos más grandes se fragmentan en remolinos más pequeños. Se considera que esta zona no se ve afectada por la viscosidad ni por los remolinos de escalas mayores (Leschziner, 2015).

Como vemos en el gráfico, en esta zona el espectro de la energía turbulenta ( $E(\kappa)$ ) es proporcional a  $\kappa^{-5/3}$ . Kolmogorov propuso una función para definir este espectro en esta zona:

$$E(\kappa) = C\varepsilon^{2/3}\kappa^{-5/3} \quad \text{Ecuación 2-13}$$

- Zona C: Es la zona donde se acaba disipando la energía convirtiéndose en calor debido a la viscosidad del fluido (Leschziner, 2015). Como se ha mencionado anteriormente, el menor tamaño de remolino vendrá definido por la viscosidad del fluido.

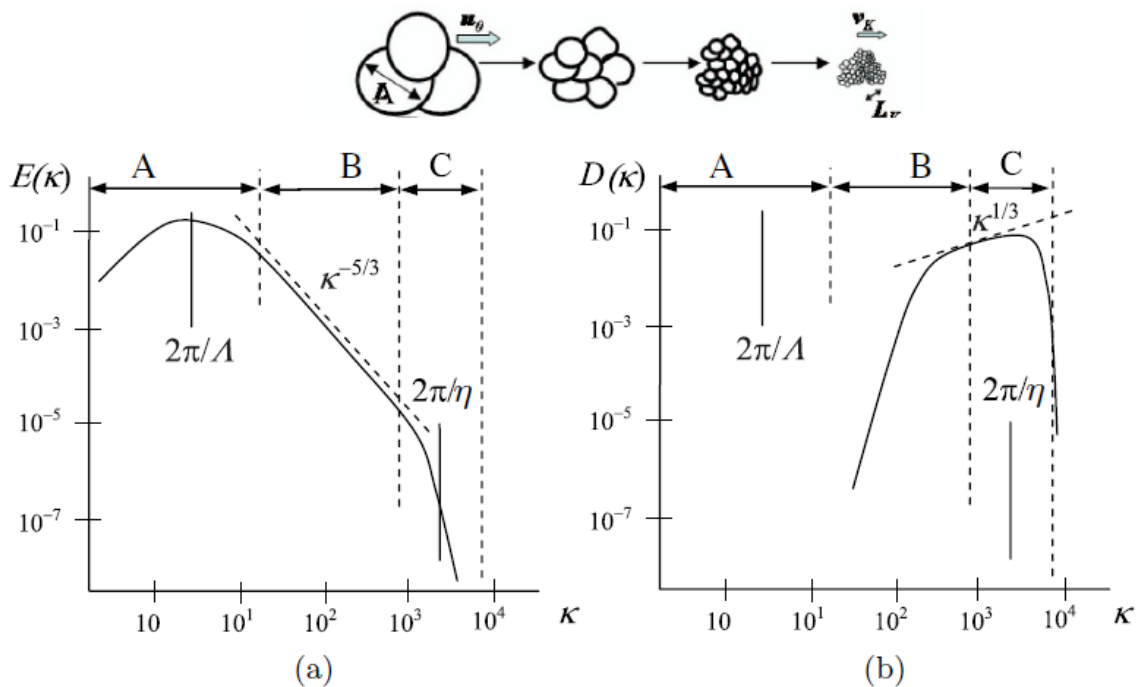


Figura 2.2 Distribución del espectro de energía turbulenta (a) y de la disipación turbulenta (b) ((Leschziner, 2015), Fig 2.4 - p13)

#### 2.1.4. Tipos de simulación

Si quisiéramos resolver completamente el flujo tendríamos que capturar desde las escalas más grandes hasta las escalas turbulentas más pequeñas del flujo. Esto se conoce como *Direct Numerical Simulation* (DNS). Este tipo de simulaciones tienen un coste computacional prohibitivo, debido a la cantidad de celdas que se necesitan para resolverlas. Por ejemplo, en el caso del sector de la automoción, donde el Re oscilan alrededor de  $10^6$ , la ratio entre la escala de los remolinos más pequeños y los más grandes serán del orden de  $3 \cdot 10^5$ .

Por ello, a nivel industrial, se utilizan diferentes métodos simulación que modelen mayor o menor parte de la turbulencia.

#### 2.1.4.1. RANS

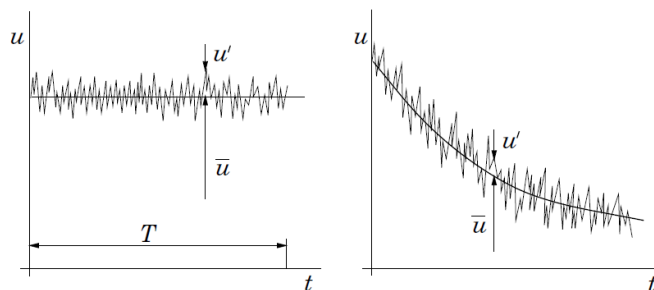
En este tipo de simulaciones, se modelan todas las escalas de la turbulencia usando modelos de turbulencia. Este tipo de simulación es la menos costosa computacionalmente. Antes de introducir el modelo que se ha usado en las simulaciones, hay que introducir las ecuaciones de Navier-Stokes promediadas temporalmente (RANS – *Reynolds Averaged Navier-Stokes equations*). El método para hacer el promediado fue propuesto por Osborne Reynolds en 1895 (Osborne Reynolds, 1895). Osborne Reynolds primero propuso descomponer la variable instantánea en la suma del valor promedio y las fluctuaciones instantáneas:

$$\phi(x_i, t) = \overline{\phi(x_i)} + \phi'(x_i, t) \quad \text{Ecuación 2-14}$$

donde

$$\overline{\phi(x_i)} = \frac{1}{T} \int_t^{t+T} \phi(x_i, t) \quad \text{Ecuación 2-15}$$

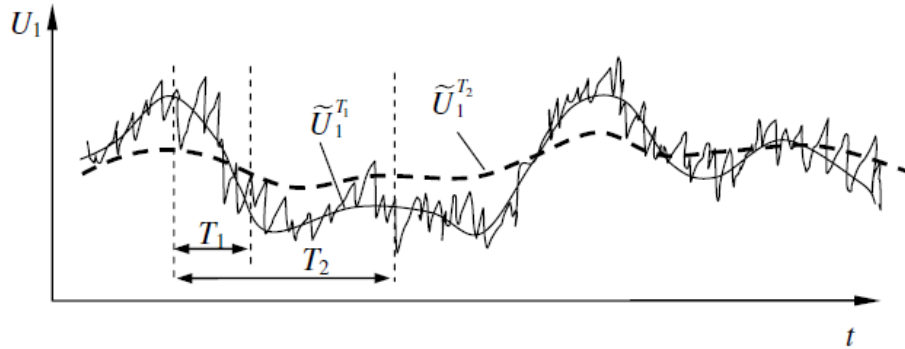
En esta última ecuación tenemos que t es el tiempo, y T es el intervalo de tiempo sobre el que se promedia. Este tiempo T debe ser suficientemente grande para que el promedio no dependa de la ventana de tiempo escogida.



**Figura 2.3 Promediado temporal de la velocidad para un flujo estadísticamente estacionario (izqda.) y para un flujo no estacionario (dcha.) ((Ferziger et al., 2002), Fig 10.24 - p397)**

Aunque no se entra en detalle, también existe la versión no estacionaria de las ecuaciones RANS (URANS – Unsteady Reynolds Average Navier-Stokes equations) (Anexo B.c). La diferencia es que el tiempo T es mayor que las fluctuaciones propias de la turbulencia, pero, en el caso de existir estructuras coherentes y deterministas en el flujo, éstas deberían

sobrevivir (Ferziger et al., 2002). En este caso es importante seleccionar correctamente el tiempo T para el promedio para poder capturar correctamente estas estructuras (Figura 2.4).



**Figura 2.4** Señal de una velocidad turbulenta  $U(t)$  y la misma señal promediada con dos ventanas de tiempo  
 ((Leschziner, 2015), Fig 3.1 – p35)

Volviendo a un valor de  $T \rightarrow \infty$ , condiciones estacionarias, si introducimos la Ecuación 2-14 las ecuaciones de Navier-Stokes y de la conservación tenemos:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho(\bar{u}_i + u'_i)}{\partial x_i} = 0 \quad \text{Ecuación 2-16}$$

$$\frac{\partial \rho(\bar{u}_i + u'_i)}{\partial t} + \frac{\partial \rho(\bar{u}_i + u'_i)(\bar{u}_j + u'_j)}{\partial x_j} = - \frac{d(\bar{p} + p')}{dx_i} + \frac{\partial \bar{\sigma}_{v_{ij}}}{\partial x_j} \quad \text{Ecuación 2-17}$$

Donde

$$\bar{\sigma}_{v_{ij}} = \mu \left( \frac{\partial(\bar{u}_i + u'_i)}{\partial x_j} + \frac{\partial(\bar{u}_j + u'_j)}{\partial x_i} \right) \quad \text{Ecuación 2-18}$$

Si a continuación promediamos temporalmente estas ecuaciones nos queda:

$$\frac{\partial \rho \bar{u}_i}{\partial x_i} = 0 \quad \text{Ecuación 2-19}$$

$$\frac{\partial(\rho \bar{u}_i \bar{u}_j + \rho \overline{u'_i u'_j})}{\partial x_j} = - \frac{d\bar{p}}{dx_i} + \frac{\partial \bar{\sigma}_{v_{ij}}}{\partial x_j} \quad \text{Ecuación 2-20}$$

$$\bar{\sigma}_{v_{ij}} = \mu \left( \frac{\partial(\bar{u}_i)}{\partial x_j} + \frac{\partial(\bar{u}_j)}{\partial x_i} \right)$$

Habiendo calculado el término lineal de la siguiente forma:

$$\overline{(\bar{u}_i + u'_i)(\bar{u}_j + u'_j)} = \bar{u}_i \bar{u}_j + \overline{u'_i u'_j} \quad \text{Ecuación 2-21}$$

Y sabiendo que, por definición, el promedio de la variable fluctuante es 0 ( $\overline{u'_i} = 0$ ) y que el promedio de una variable ya promediada es la misma variable promediada ( $\overline{\bar{u}_i} = \bar{u}_i$ ).

Tras hacer esto, nos aparece un nuevo término que se conoce como el tensor de Reynolds ( $-\rho \overline{u'_i u'_j}$ ). Reescribiendo las ecuaciones RANS nos queda:

$$\frac{\partial \rho \bar{u}_i}{\partial x_i} = 0 \quad \text{Ecuación 2-22}$$

$$\frac{\partial \rho (\bar{u}_i \bar{u}_j)}{\partial x_j} = -\frac{d\bar{p}}{dx_i} + \frac{\partial}{\partial x_j} \left( \mu \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \rho \overline{u'_i u'_j} \right) \quad \text{Ecuación 2-23}$$

Una vez tenemos las ecuaciones planteadas, la idea detrás de los modelos de turbulencia es proponer un modelo que obtenga el tensor de Reynolds. Al no tener acceso a los valores de las fluctuaciones durante la simulación, el modelo de la viscosidad turbulenta para el tensor de Reynolds asume que éste se puede aproximar como un incremento de la viscosidad. Por tanto, el tensor de Reynolds tendría una forma similar que la disipación viscosa (Ferziger et al., 2002):

$$-\rho \overline{u'_i u'_j} = \mu_t \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \frac{2}{3} \rho \delta_{ij} k \quad \text{Ecuación 2-24}$$

Donde  $\mu_t$  es la viscosidad turbulenta y  $k = \frac{1}{2} \overline{u'_i u'_i}$  es la energía cinética turbulenta. A partir de este punto, hay muchos modelos diferentes para resolver este tensor. Estos diferentes modelos se llaman modelos de turbulencia.

#### 2.1.4.2. Large Eddy Simulations (LES)

Como se ha comentado anteriormente, la turbulencia tiene varias escalas. Las más energéticas son las escalas más grandes (Figura 2.2), y son las más efectivas para el transporte de las propiedades. Por tanto, tiene sentido resolver estas escalas sin necesidad de modelarlas. Eso es lo que conseguimos con las simulaciones LES. Este tipo de simulaciones siguen siendo costosas, pero mucho menos que una simulación DNS.

Como en el caso anterior, las ecuaciones de Navier-Stokes se filtran, pero esta vez, se filtran espacialmente. Las ecuaciones que se obtienen son similares a las que se obtienen en URANS, excepto por el término adicional que aparece después del filtrado. En el caso de URANS aparece el tensor de Reynolds y aquí aparece tensor de esfuerzos de subescala ( $\tau_{ij}^S$ ). Las ecuaciones tras el filtrado quedan (en este apartado  $\bar{\quad}$  indica filtrado temporal):

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho \bar{u}_i}{\partial x_i} = 0 \quad \text{Ecuación 2-25}$$

$$\frac{\partial \rho \bar{u}_i}{\partial t} + \frac{\partial \rho (\bar{u}_i \bar{u}_j)}{\partial x_j} = \frac{\partial \tau_{ij}^S}{\partial x_j} - \frac{d\bar{p}}{dx_i} + \frac{\partial}{\partial x_j} \left( \mu \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \right) \quad \text{Ecuación 2-26}$$

$$\tau_{ij}^S = -\rho (\overline{u_i u_j} - \bar{u}_i \bar{u}_j) \quad \text{Ecuación 2-27}$$

Para resolver el tensor de subescala, se necesita un modelo como se utilizó uno para el tensor de Reynolds. El primer modelo de subescala propuesto fue el modelo de Smagornisky (Leschziner, 2015). Este modelo se basa en la hipótesis de la existencia de una analogía entre los esfuerzos viscosos y los esfuerzos de subescala, idea similar a la usada para resolver flujos turbulentos con RANS. Por tanto, el tensor de esfuerzo se define como:

$$(\tau_{ij})_{SGS} = \tau_{ij} - \frac{1}{3} \delta_{ij} \tau_{kk} = -\mu_{SGS} \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) = -2\mu_{SGS} \bar{S}_{ij} \quad \text{Ecuación 2-28}$$

Una vez hecha esta analogía, solo queda definir la viscosidad de subescala. A partir de la suposición que hemos hecho, se puede suponer que las unidades de esta viscosidad serán  $m^2/s$ . Por tanto, serán igual a una escala de velocidad (m/s) por una escala de longitud (m). La escala de longitud de las escalas de SGS son del mismo tamaño que el volumen de control ( $\Delta$ ), mientras que la escala de velocidad se puede calcular como la norma del tensor deformación  $|\bar{S}|$  por la escala de longitud (Ferziger et al., 2002).

La magnitud del tensor de deformación se calcula de la siguiente forma:

$$|\bar{S}| = (2 * \bar{S}_{ij} * \bar{S}_{ji})^{1/2} \quad \text{Ecuación 2-29}$$

Siendo

$$\bar{S}_{ij} = \frac{1}{2} \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \quad \text{Ecuación 2-30}$$

Haciendo un análisis dimensional ((Ferziger et al., 2002)), se obtuvo la siguiente fórmula para la viscosidad de subescala:

$$\mu_{SGS} = C_s \rho \Delta^2 |\bar{S}| \quad \text{Ecuación 2-31}$$

A partir de aquí, existen diferentes modelos que buscan modelar el tensor de esfuerzos de subescala de una forma diferente a la propuesta por Smagorinsky.

#### 2.1.4.3. DES y Modelos híbrido LES-RANS

Otro tipo de simulación, y actualmente muy usado en la industria de la automoción, es la simulación DES, de sus siglas en inglés *Detached Eddy Simulations*, o modelos híbridos LES-RANS. La idea detrás de este tipo de simulaciones es una combinación entre LES y RANS operando en diferentes subdominios (Leschziner, 2015). En el caso de la automoción se resuelve con LES la mayor parte del dominio y, en la capa límite y cerca de ella, se usa RANS (ANSYS, 2021). Este tipo de modelos están especialmente pensados para simulaciones con números de Reynolds elevados. Uno de los principales problemas para este tipo de modelos es la construcción de la malla (Spalart, 2001) y como se conectan matemáticamente ambos subdominios (Leschziner, 2015).

Aunque el primer modelo de este tipo fue el de Spalart-Allmaras en 1992, en la actualidad existen otros modelos híbridos como el modelo combinado con SST k- $\omega$  (ANSYS, 2021; Leschziner, 2015).

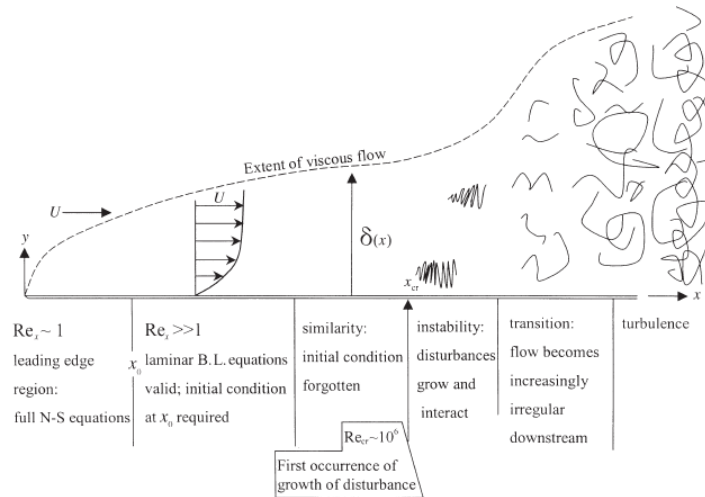
#### 2.1.5. Capa límite

Otra de las claves a la hora de realizar una simulación es como se resuelve la capa límite.

La capa límite es la zona próxima a la pared, donde los efectos de la viscosidad son importantes. En esta zona, el flujo pasa de estar en reposo en la pared (condición de *no slip*), a la velocidad del flujo libre. El grosor de la capa límite ( $\delta$ ) se considera como la distancia normal entre la pared y donde la velocidad es el 99% de la velocidad del flujo libre.

En el desarrollo de una capa límite se pueden diferenciar tres zonas (Figura 2.5). Al principio, encontramos la capa límite laminar, la cual se caracteriza por un flujo regular, donde los

elementos de fluido fluyen a lo largo de una línea de corriente y no se mezclan entre diferentes líneas de corriente. Posteriormente, empiezan a aparecer perturbaciones en el flujo (zona de transición) que van aumentando hasta que el flujo pasa a ser turbulento en la capa límite.



**Figura 2.5 Flujo esquemático de la capa límite sobre una placa plana semi-infinita ((Frank M. White, 2008) , Fig 9.10 - p383)**

En este estudio, vamos a lidiar con flujos turbulentos. Por esta razón, nos centraremos en el análisis de la capa límite turbulenta. De nuevo, se pueden diferenciar varias zonas dentro de la capa límite turbulenta.

Primero, es importante introducir el concepto de  $y^+$ .  $y^+$  es una variable adimensional que nos indica a qué distancia normal de la pared y en qué zona de la capa límite turbulenta nos encontramos. Se define como:

$$y^+ = y \frac{u_\tau}{\nu} \quad \text{Ecuación 2-32}$$

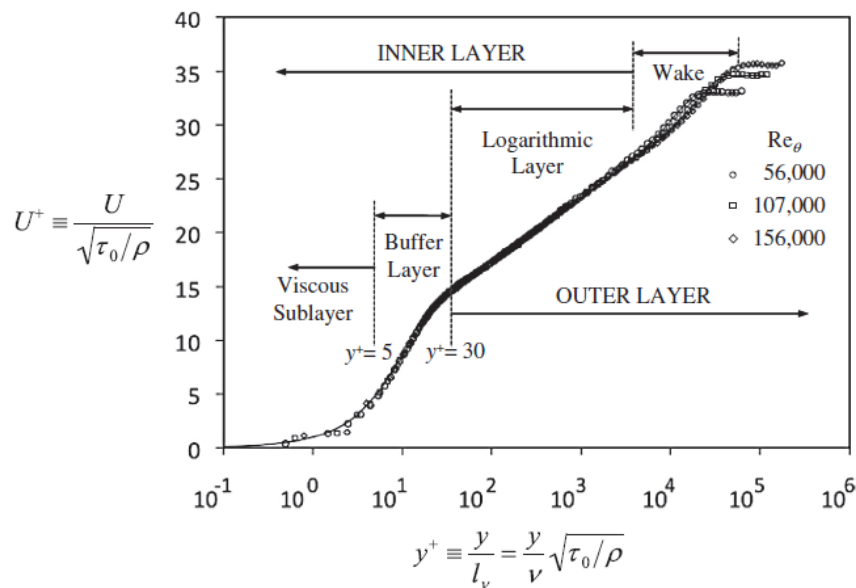
siendo

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad \text{Ecuación 2-33}$$

$$U^+ = \frac{U}{u_\tau} \quad \text{Ecuación 2-34}$$

Si se grafica el perfil de la capa límite usando estas variables adimensionales, vemos que se obtiene un perfil similar para distintos tipos de flujos turbulentos completamente

desarrollados (Frank M. White, 2008). Este comportamiento se pudo confirmar experimentalmente en el 2010 (Oweis et al., 2010).



**Figura 2.6 Perfil promedio de velocidad en una capa límite turbulenta ((Frank M. White, 2006), Fig 12.8 - p587)**

En la capa límite, podemos identificar dos zonas, dependiendo de los esfuerzos dominantes en esa zona (Figura 2.6). La zona exterior se caracteriza por efectos viscosos negligibles similar a los que se tienen fuera de la capa límite (Frank M. White, 2006).

En la zona interior de la capa límite, los esfuerzos viscosos son dominantes. En esta zona de la capa límite, el perfil de velocidades depende solo de los parámetros próximos a la pared (distancia a la pared,  $\rho$ ,  $\tau_w$ ,  $\nu$ ), y no de la velocidad del flujo externo ( $U_\infty$ ) o de las longitudes características de nuestra geometría (Frank M. White, 2006).

Centrándonos en la capa interna, vemos que nos encontramos tres zonas diferenciadas (Figura 2.6):

- Sub-capa viscosa ( $y^+ < 5$ ) la cual se caracteriza por  $U^+ = y^+$ .
- Región logarítmica ( $y^+ > 30$ ): El perfil de velocidades sigue en esta zona una función logarítmica  $U^+ = \frac{1}{\kappa} \ln(y^+) + B$ , donde  $\kappa$  es la constante de Von Karman y  $B=5.2$  (Pope, 2000).
- Zona de transición (Buffer región) ( $5 < y^+ < 30$ ). En esta zona el perfil de velocidades no sigue ninguna de las dos tendencias.

Cuando se está preparando la simulación, es importante conocer cómo queremos modelar la capa límite. En el caso de utilizar ecuaciones RANS, tenemos la opción de usar modelos de turbulencia que son capaces de resolver la capa límite correctamente (SST k- $\omega$ ). Esto implica que deberemos tener mallas en las que el centro de la primera celda este en la sub-capa viscosa ( $y^+ \approx 1$ ), lo que implicará un mayor número de celdas.

Existe otra opción que consiste es usar modelos de turbulencia con funciones de pared. Con las funciones de pared, se trata de modelar la zona próxima a la pared. En este caso, el valor de la primera celda deberá estar en la región logarítmica ( $y^+ > 30$ ). Es importante evitar la zona de transición ya que, en esa zona, ninguna de las opciones resuelve correctamente la capa límite.

Se pueden encontrar varias páginas que, dando el  $Re$ , la longitud característica del caso y el  $y^+$  deseado, permite calcular el grosor de la primera celda de la capa límite aproximadamente. En capítulos posteriores se entrará en mayor profundidad en el modelo de turbulencia escogido y el tratamiento de pared usado.

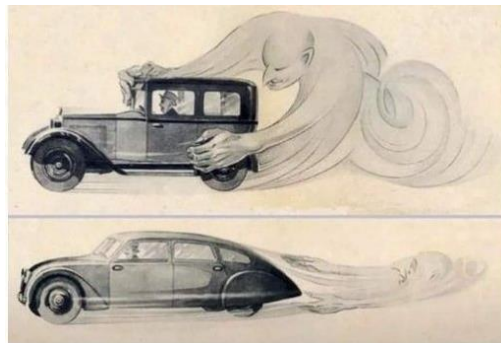
#### 2.1.6. Resistencia (Drag)

Resolver correctamente la capa límite es determinante para poder obtener una buena distribución de presiones sobre el coche y que, por tanto, lleve a unas predicciones correctas de las fuerzas sobre el vehículo. En nuestro caso, será especialmente importante para obtener un valor correcto de resistencia al avance.

La resistencia al avance a la que está sometido el vehículo se puede descomponer en dos fuentes: el drag de fricción y el drag de forma (Anderson, 2017).

- Drag de forma (o *Pressure Drag*). Es la fuerza resultante, en la dirección del flujo, de todas las fuerzas de presión ejercidas sobre todas las superficies del cuerpo. La componente principal de este tipo de drag es la separación del flujo (si existe). En los vehículos, existirá esta componente en la parte posterior, generando una estela con presiones bajas que “retiene” el coche (Figura 2.7), la cual se trata de minimizar. La forma y valor de esta depresión generada tras el coche depende, entre otros factores, de la capa límite en el punto de separación. Otros componentes de este drag son:

- Drag inducido (o *Vortex Drag*). Esta resistencia se asocia a los vórtices que se obtienen debido a la generación de sustentación o *downforce*.
- Resistencia por ondas de choque (wave drag). Asociado a la energía disipada por las ondas de choque. Al trabajar con Mach bajos, no aparecen este tipo de fenómenos.
- Resistencia de Fricción (*Skin Friction Drag*). Es la resultante, en la dirección del flujo, de todos los esfuerzos cortantes sobre todas las superficies del cuerpo.

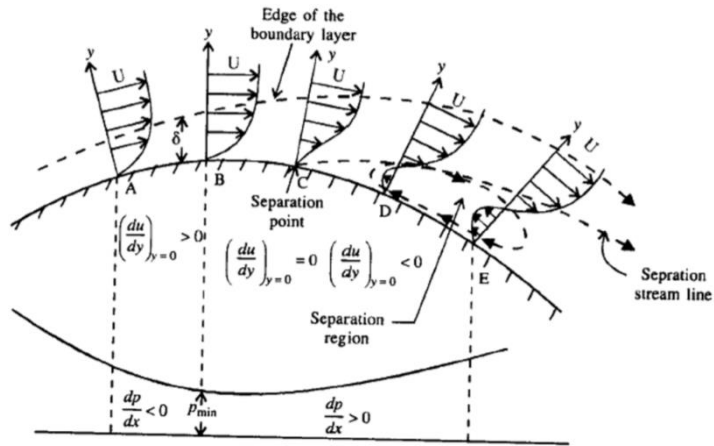


**Figura 2.7 Resistencia aerodinámica – Explicación gráfica simplificada (Fuente: Tatra - The Legacy of Hans Ledwinka)**

Vemos la necesidad resolver la capa límite correctamente. Una mala resolución puede llevar a sobrestimar, o subestimar, el gradiente de presión máximo que puede afrontar la capa límite, y que aparezcan, o no, zonas de separación y recirculación del flujo, que lleven a una predicción errónea de la fuerza resultante sobre el vehículo.

La separación en una superficie continua se rige por el gradiente adverso de presión y la viscosidad del fluido (Lin, 1992). Para que el flujo no se separe, la capa límite debe tener suficiente momento para afrontar la pérdida de energía cinética asociada a este gradiente adverso de presión y a la disipación viscosa (Lin, 1992). Esta pérdida de energía cinética es más significativa cerca de la pared, ya que el momento allí es menor. Conforme el perfil de velocidades cerca de la pared avanza, frente a este gradiente adverso de presión, va perdiendo momento y el perfil va variando (Figura 2.8). El punto de separación es donde  $\left(\frac{du}{dy}\right)_w = 0$ . Pasado este punto, aparecen zonas de recirculación con flujo adverso. Con esto, se puede ver la importancia de resolver correctamente la capa límite y poder obtener una buena predicción del momento a lo largo de la capa límite y, por tanto, no sobre- o subestimar el punto de separación y la fuerza de ésta.

Respecto a la resistencia de fricción sucede algo similar. Si no se resuelve correctamente el perfil de la capa límite, los esfuerzos cortantes obtenidos sobre la superficie tampoco serán correctos, lo que llevará, de nuevo, a una predicción errónea de las fuerzas resultantes sobre el vehículo.



**Figura 2.8 Perfil capa límite sometido a un gradiente adverso de presión (Fuente: <http://www.examinill.com/wp-content/uploads/2016/11/Figure-12.png>)**

Es importante también considerar que, aparte de tener un buen modelo que resuelva correctamente la capa límite, deberemos tener una malla adecuada para ello. En casos donde la geometría es compleja, como es el segundo modelo simulado, se complica significativamente conseguir una buena malla a lo largo de todo el vehículo. Una mala malla, aunque sean pocas celdas, puede derivar en predicciones incorrectas, fluctuaciones, la no convergencia o, incluso, la divergencia de los resultados.

### 2.1.7. Coeficientes de Lift/Drag

Finalmente, otro de los conceptos importantes a tener en cuenta al analizar los resultados, son el coeficiente de sustentación ( $C_L$ ) y de resistencia del coche ( $C_D$ ). Estos coeficientes se definen como:

$$C_L = \frac{L}{\frac{1}{2} \rho U^2 A} \quad \text{Ecuación 2-35}$$

$$C_D = \frac{D}{\frac{1}{2} \rho U^2 A} \quad \text{Ecuación 2-36}$$

En el sector del motorsport el  $C_L$  es un parámetro muy relevante a la hora de diseñar el coche, ya que permite aumentar la velocidad en curvas medias y rápidas. Sin embargo, en el sector del automóvil, lo que se busca es que el coche no tenga más de un valor definido de  $C_L$ . Lo que sí importa es la resistencia al aire del vehículo y por tanto el  $C_D$ .

Recientemente algunos fabricantes utilizan el  $C_D$  como estrategia de marketing para mostrar la eficiencia del vehículo. Un ejemplo de esto lo hizo Mercedes en la presentación del modelo Vision EQXX con un  $C_D$  de 0.17 o con el modelo EQS con un  $C_D$  de 0.20, referente en el sector actualmente junto con Lightyear y su Lightyear 0, con un  $C_D < 0.19$ .



Figura 2.9 Lightyear 0 (Fuente: <https://lightyear.one/lightyear-0>)



Figura 2.10 Modelo Mercedes VISION EQXX  
(Fuente: <https://www.diariomotor.com/>)



Figura 2.11 Modelo Mercedes EQS (Fuente: <https://www.mercedes-benz.es>)

En la fase de diseño, normalmente, el  $C_D$  se da con tres decimales. Cuando se habla de que se han incrementado o reducido el drag 1 punto (o *count* en inglés), se hace referencia al tercer decimal (de 0.251 a 0.250).

## 2.2. Tipos de simulación en el sector

En agosto del 2021 hubo una conferencia relevante para el sector, donde se pidió a empresas del sector (Ford, Audi, Stellantis, BETA-CAE Systems...), universidades (Loughborough University, University of Oxford, Ohio State University) y centros de investigación (Barcelona Supercomputing Center), que simularan con el tipo de simulación y software que quisieran, dos casos diferentes, con diferentes mallas dadas por los organizadores (University of Oxford, 2021). Participaron 22 empresas, proporcionando 65 resultados diferentes. Se simuló el modelo de DrivAer Notchback con el *underbody* detallado y las ruedas estáticas. Este análisis nos puede dar una idea de que tipos de simulación se llevan a cabo en el sector. Además, se hizo una comparativa de los diferentes casos clasificándolos según el tipo de simulación, analizando las principales diferencias y particularidades de los resultados obtenidos (Hupertz et al., 2021).

Si analizamos los resultados presentados, vemos como se entregaron resultados de simulaciones RANS, de modelos híbridos RANS-LES y algunos códigos propios. Vemos como simulaciones LES y DNS parecen quedar fuera del alcance de las empresas, debido a su alto coste computacional y el tiempo requerido para obtener resultados. Esto también se refleja en diferentes estudios donde simulaciones RANS (Bezavada & Kodali, 2012; Heft et al., 2012a; R. F. Soares et al., 2017; R. F. Soares & José De Souza, 2018) y modelos híbridos (Collin et al., 2016; Ekman et al., 2020; Islam & Thornber, 2017) se usan para resolver una gran variedad de casos. Además, en varios artículos se hace una comparativa entre estos diferentes métodos (Ashton & Revell, 2015; Aultman et al., 2021; R. Soares et al., 2015), al igual que se hizo en la conferencia de AutoCFD (Hupertz et al., 2021).

Si solo nos centramos en los resultados obtenidos, vemos como las simulaciones con modelos híbridos RANS-LES, ofrecen una mayor precisión tanto en las distribuciones de coeficiente de presión ( $C_p$ ) como en valores de  $C_D$  y  $C_L$  (Ashton et al., 2016; Ashton & Revell, 2015; Hupertz et al., 2021). Sin embargo, es importante tener en cuenta el incremento del coste de computación al usar modelos híbridos, pudiendo incrementar el tiempo de computación hasta 17 veces (Ashton et al., 2016).

## 3. Objetivos y Metodología de trabajo

### 3.1. Objetivo general

El objetivo general del Trabajo Final de Máster es asentar unas bases para poder implementar un software de código libre para simulaciones CFD en el sector de la automoción, que permita una mejor customización teniendo acceso al código y, a su vez, permita reducir los costes derivados del uso de este tipo de programas comerciales. En este caso el software que se quiere utilizar es OpenFOAM.

### 3.2. Objetivos específicos

El objetivo principal se puede descomponer en varios objetivos específicos, con el fin de cumplir el objetivo principal de este TFM. Estos objetivos específicos son:

CONOCER OpenFOAM como herramienta de simulación CFD así como las recomendaciones de usuarios previos.

INDAGAR en las posibilidades de OpenFOAM como software de simulación en el ámbito industrial. Hasta la fecha, se conoce a OpenFOAM como una muy buena herramienta para investigación (donde se pueden obtener mallas de mejor calidad).

APRENDER a utilizar el mallador de ANSA. Hasta la fecha, no se cuenta con experiencia en la empresa para realizar mallas volumétricas con este software.

ESTABLECER un proceso tanto de malla como de simulación que permita resolver los casos deseados.

ANALIZAR los resultados obtenidos VALIDÁNDOLOS con resultados experimentales.

### 3.3. Metodología de trabajo

Para poder alcanzar los objetivos planteados se siguen una serie de pasos durante la investigación.

Primero se realiza una investigación en trabajos previos, foros, documentos de mejores prácticas, conferencias o cursos disponibles online, así como de las recomendaciones de *set ups* más robustas para este tipo de simulaciones. A su vez, se empieza a investigar la herramienta de ANSA y sus posibilidades como mallador. Se prueban diferentes tipos de

opciones dentro del programa para conseguir un procedimiento robusto que consiga una malla de buena calidad para usar en OpenFOAM.

A continuación, se simula una versión simplificada de un coche sin ruedas ni retrovisores, de la que se tienen resultados experimentales de (Lanuza, 2016). Se prueban varias configuraciones para garantizar un modelo robusto con este coche, antes de pasar a una versión más compleja.

Finalmente, se simulará un caso de benchmark muy analizado en el sector de la automoción, el modelo de DrivAer de la TUM (TUM School of Engineering and Design, n.d.).

## 4. OpenFOAM

### 4.1. Introducción

OpenFOAM es un software gratuito y de código abierto. Actualmente hay dos versiones de este software, una desarrollada por “OpenFOAM Foundation” y la otra por “esi-OpenCFD”. Aun estando desarrollados por dos grupos independientes, las capacidades de ambos softwares son muy similares. Ambos son softwares basados en librerías de C++ que proporcionan dos tipos de aplicaciones, la primera son los *solvers* diseñados para resolver el problema en cuestión y la segunda es la propia manipulación de la información (Greenshields, 2011; OpenCFD Ltd., 2022).

Para poder trabajar con OpenFOAM es importante estar familiarizado con la estructura de carpetas usada. A continuación, se mostrará el formato usado en este trabajo que corresponde a un modelo RANS con el modelo de turbulencia SST k- $\omega$ . De todas formas, cualquier otro tipo de simulación, tendrá una estructura similar.

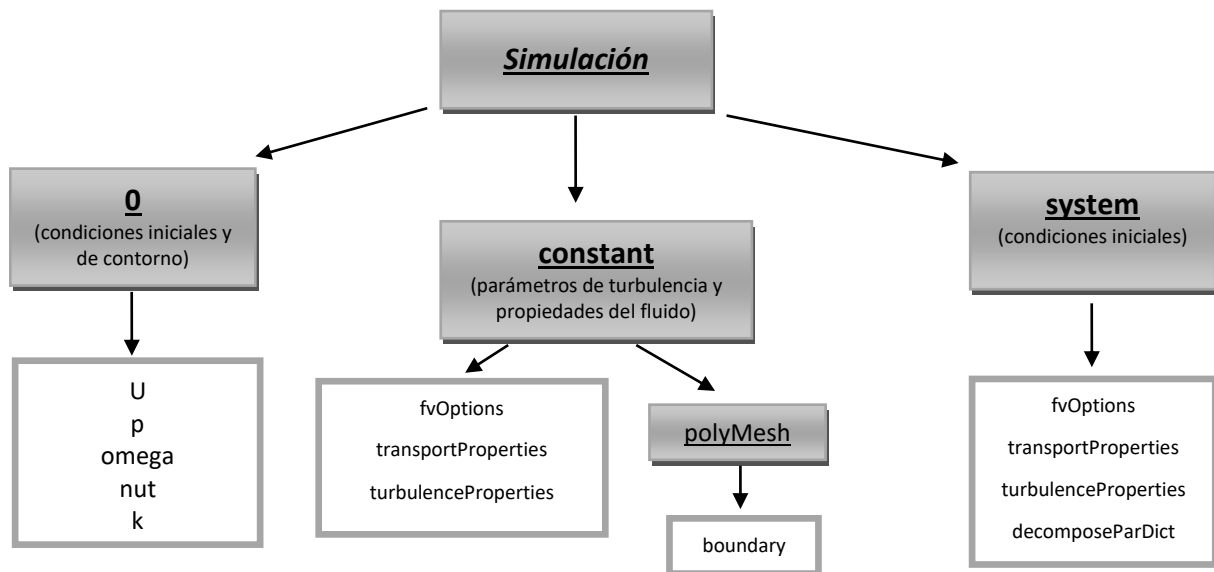


Figura 4.1 Estructura de carpetas en OpenFOAM (inspirado en (Bastian Nebenführ, 2010), Fig 2.6 – p16)

La Figura 4.1 muestra un esquema de cómo debe ser la estructura de las carpetas. Excepto el nombre de la carpeta “Simulación”, es importante mantener el nombre de las carpetas y de los ficheros.

Podemos encontrar una descripción detallada, en la que se indica que debe introducirse en cada una de las carpetas y ficheros, en el Anexo B.a.

## 4.2. Solvers

OpenFOAM ofrece una diversa variedad de *solvers* dependiendo del tipo de flujo y sistema que se quiera analizar. Sin embargo, solo se explicarán los dos usados en este estudio.

### 4.2.1. potentialFoam

Este *solver* resuelve un flujo potencial. Se ha utilizado para obtener una solución del campo interno para usar como condiciones iniciales para la simulación RANS.

### 4.2.2. simpleFoam

simpleFoam es un *solver* que resuelve las ecuaciones RANS (ver sección 2.1.1) para un flujo incompresible, estacionario y turbulento. Al resolver las ecuaciones RANS, necesitamos un modelo de turbulencia para poder resolver el tensor de Reynolds. Para ello, en este estudio se ha utilizado el modelo SST k- $\omega$ .

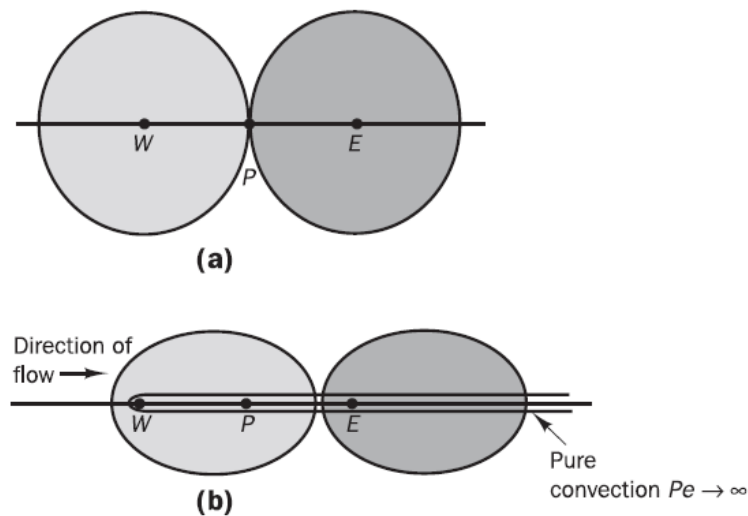
## 4.3. Esquemas Numéricos

En este apartado se van a introducir los principales esquemas numéricos usados. OpenFOAM pide especificar los diferentes esquemas que se van a usar para resolver cada uno de los operadores. En la mayoría se han probado varios esquemas para analizar las diferencias. El objetivo al final es poder fijar todos los esquemas de discretización con esquemas numéricos de segundo orden o superior.

Un buen esquema numérico tiene que cumplir unas características determinadas (Versteeg & Malalasekera, 1995). Las más importantes son:

- Debe ser **conservativo** (*Conservativeness*). Esto implica que el flujo de una variable que entra por una cara debe ser igual al flujo que sale de la cara adyacente de la celda contigua (Versteeg & Malalasekera, 1995).
- Debe estar delimitado (*Boundedness*). Al tener un volumen discretizado el cual se quiere resolver, se tienen una serie de ecuaciones algebraicas en cada celda del dominio. La resolución de estos sistemas empieza con una suposición inicial o resultados previos de un problema similar y, mediante un proceso iterativo, se van resolviendo las ecuaciones hasta que la solución converge.
- La transportividad (*Transportiveness*). Los esquemas numéricos deben considerar la dirección del flujo y las características que tiene para el transporte por convección de

la variable. Para analizar este fenómeno se utiliza el número de Peclet ( $Pe$ ), el cual da la relación entre la convección y la difusión ( $Pe \rightarrow 0$ , difusión pura).



**Figura 4.2 Distribución de  $\phi$  cerca de dos fuentes a diferentes número de Peclet. (a) Difusión pura i (b) Difusión i convección ((Versteeg & Malalasekera, 1995) ,Fig 5.9 - p144)**

En el caso de difusión pura, el fluido está en reposo, y, por tanto, esta propiedad es menos importante, ya que la variable se suele expandir igual en todas las direcciones (Versteeg & Malalasekera, 1995). Cuando el fluido no se encuentra en reposo y, por tanto, tenemos transporte por convección hay que tener en cuenta este fenómeno. Por ejemplo, si nos fijamos en la Figura 4.2 (b) vemos como el punto P no se ve afectado por la fuente en E, solo por la fuente en W. Esto conlleva a que los esquemas numéricos lleven integrados la relación entre la influencia de la dirección, la dirección del flujo y el número de  $Pe$  para cumplir con la esta condición (Versteeg & Malalasekera, 1995).

- **Precisión.** Con esto nos referimos al orden del error del esquema numérico. Lo deseado es que sean como mínimo de segundo orden, ya que los de primer orden son esquemas difusivos y conllevan un mayor error (Versteeg & Malalasekera, 1995).

Los esquemas numéricos usados en este trabajo se resumen en la siguiente tabla. Además, en los anexos se puede encontrar una descripción detallada de los más usados en este trabajo.

**Tabla 1. Esquemas numéricos usados.**

Tipo	Esquema
gradScheme	cellLimited Gauss Linear $\psi$
	Gauss Linear
	leastSquares
	cellLimited leastSquares $\psi$
divScheme	bounded Gauss upwind
	bounded Gauss linearUpwindV
	bounded Gauss linearUpwind
	bounded Gauss vanLeer
laplacianSchemes	Gauss linear limited 0.5
	Gauss linear corrected
interpolationScheme	linear
snGradScheme	limited 0.5
	corrected

#### 4.4. Modelo de turbulencia

Como se ha mencionado anteriormente, para resolver el tensor de Reynolds en las ecuaciones RANS es necesario incluir un modelo de turbulencia. El modelo turbulento que se ha escogido es el kOmegaSST. El modelo de turbulencia SST k- $\omega$  es un modelo de dos ecuaciones para la energía cinética turbulenta (k) (Ecuación 4-2) y la tasa de disipación turbulenta ( $\omega$ ) (Ecuación 4-1).

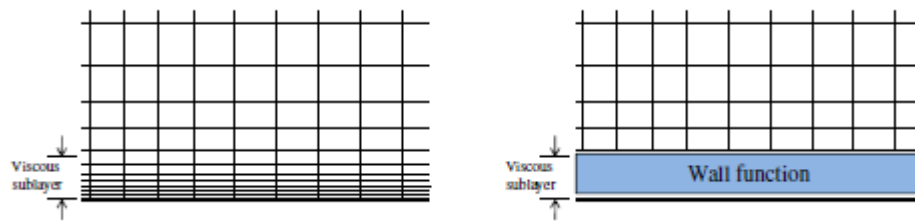
Este modelo de turbulencia es una combinación de dos modelos de turbulencia ya existentes, el k- $\epsilon$  y el k- $\omega$ . El modelo k- $\epsilon$ , el cual no es capaz de capturar correctamente la capa límite turbulenta hasta su separación y es poco fiable en las zonas de separación, pero ofrece mejores resultados en la región externa del flujo (Wilcox, 2006). En cambio, el modelo k- $\omega$  es capaz de reproducir la zona cercana a la pared (Menter et al., 2003). El modelo SST k- $\omega$  se basa en una función de mezclado que une ambos modelos dependiendo de la distancia a la pared (Ecuación 4-3). El modelo usado por OpenFOAM está definido por las siguientes ecuaciones basadas en (Christopher Rumsey, 2021; Menter et al., 2003). Para encontrar las ecuaciones completas, se puede consultar el Anexo B.b.

$$\frac{D\rho\omega}{Dt} = \alpha\rho S^2 - \rho\beta\omega^2 + \frac{\partial}{\partial x_i} \left[ (\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_i} \right] + 2(1 - F_1)\rho\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \quad \text{Ecuación 4-1}$$

$$\frac{D\rho k}{Dt} = \tilde{P}_k - \rho\beta^*\omega k + \frac{\partial}{\partial x_i} \left[ (\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_i} \right] \quad \text{Ecuación 4-2}$$

$$\alpha = F_1\alpha_1 + (1 - F_1)\alpha_2 \quad \text{Ecuación 4-3}$$

El modelo de turbulencia planteado por Wilcox (k- $\omega$ ) es capaz de resolver correctamente la zona cercana a la pared. Por tanto, para resolverlo deberemos crear una malla con  $y^+ < 5$ , idealmente con valores  $y^+ \approx 1$ . Por lo que, una vez realizada la simulación, se verificará que el modelo tiene un  $y^+$  adecuado para el modelo de turbulencia escogido.



**Figura 4.3** Malla cerca de la pared para un flujo resuelto cerca de la pared (dcha.) y usando funciones de pared (izqda) ((Bastian Nebenführ, 2010) , p14)

Sin embargo, también se han decidido lanzar simulaciones con  $y^+ > 30$  usando funciones de pared. De esta forma se ha podido realizar simulaciones menos costosas (menor tiempo y menor potencia de cálculo) para verificar el comportamiento de la herramienta bajo estas condiciones. Igualmente, al acabar la simulación se deberá verificar que el  $y^+$  que se tiene está por encima de 30 y no hemos caído en la zona de transición. Las funciones de pared que se han usado son:

**Tabla 2. Funciones de pared usadas**

Variable	Función de pared
k	kqRWallFunction
$\omega$	omegaWallFunction
$\nu_t$	nutkWallFunction
	nutUSpaldingWallFunction

Siendo tanto *omegaWallFunction* y *nutUSpaldingWallFunction* validas como condición de contorno para simulaciones con  $y^+ < 5$  debido a su formulación (OpenCFD Ltd., 2022). Ambas se han usado cuando se han hecho las simulaciones resolviendo la subcapa viscosa fijando el valor de k en la pared a 0.

## 5. Introducción de los Modelos

Para este trabajo se han planteado dos modelos diferentes para analizar y simular. El primero es un modelo simplificado del cual se tienen resultados experimentales de  $C_L$  y  $C_D$  (Lanuza, 2016). Estos ensayos se realizaron en el túnel de viento de la universidad de Cranfield, para analizar el efecto que tenía usar generadores de vórtices en el difusor de un coche.

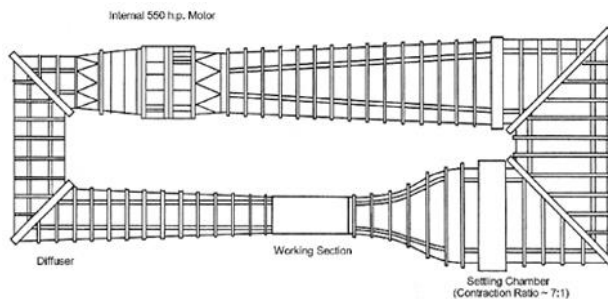


Figura 5.1 Layout del Túnel de viento de 2.4 x 1.8 m de baja velocidad (Fuente: <https://www.cranfield.ac.uk>)



Figura 5.2 Sección de test con el modelo durante los ensayos

El segundo tipo analizado son los modelos de DrivAer creados por la universidad de Múnich (TUM). Estos modelos se han establecido como referencia en el sector para estudios de benchmark (University of Oxford, 2021) y existen una gran cantidad de resultados, tanto de simulaciones como experimentales. En el artículo (Varney et al., 2020) se recogen una gran variedad de ensayos realizados por otros autores previamente.

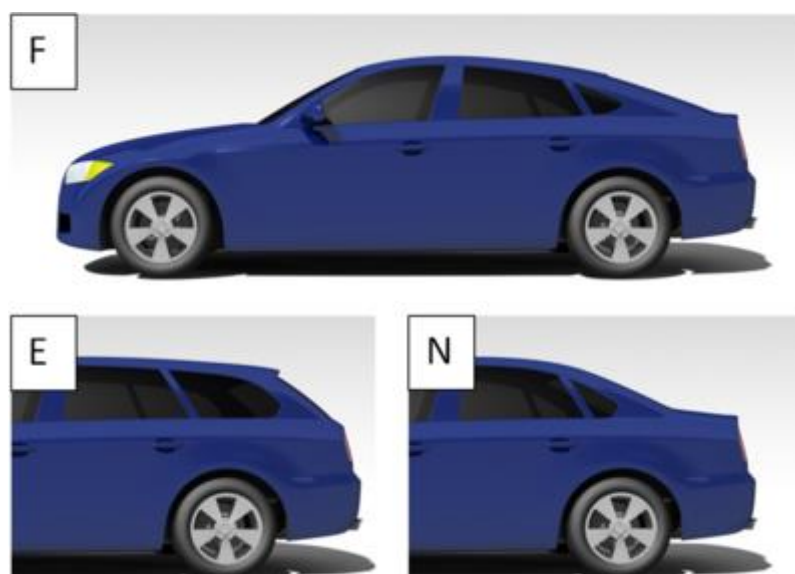


Figura 5.3 Modelos DrivAer (Fuente: (TUM School of Engineering and Design, n.d.))

## 6. Modelo Simplificado

### 6.1. Modelo de turbulencia

El modelo que se ha usado es un cuerpo como similar a un coche con difusor, pero con muchas simplificaciones (Figura 5.2). Tiene varias configuraciones disponibles, con ruedas traseras o sin, difusor simple o separado en tres. Esto lo convierte en un modelo ideal para ser usado como herramienta de investigación.

Aunque no se tiene un modelo 3D exacto del coche, durante la investigación se tomaron medidas y se dibujó un modelo 3D aproximado. El modelo tiene un área frontal de  $0.10625 \text{ m}^2$  con una altura de 250 mm y una anchura de 425 mm. La longitud del coche es de 1070 mm con una longitud de difusor de 190 mm (Figura 6.1). El difusor tiene un rango de 0 a  $11.5^\circ$ . Finalmente, el coche se ensayó a diferentes alturas respecto al suelo y así analizar el comportamiento de los generadores de vórtices conforme se reducía la distancia al mismo.

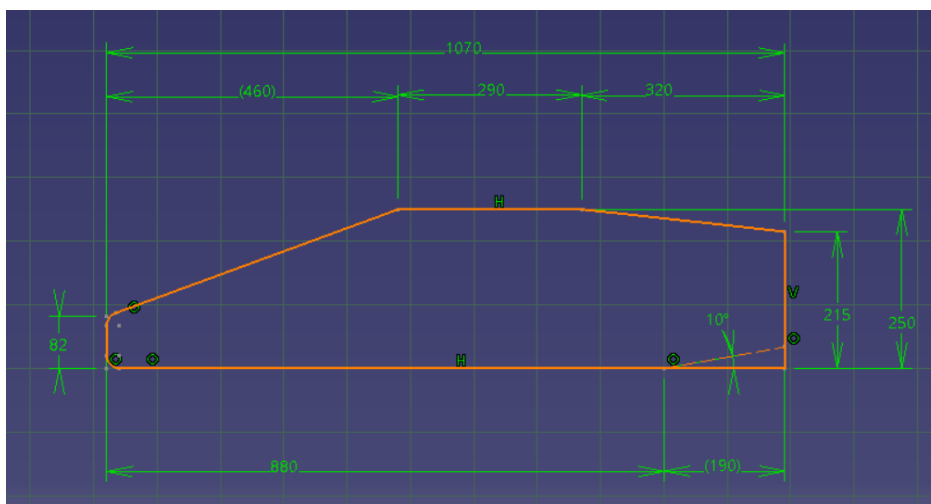


Figura 6.1 Dibujo esquemático del modelo simplificado ((Lanuza, 2016), Fig4.3, p26)

Para este trabajo se han simulado este coche con dos posiciones de difusor y unas alturas del suelo de 40, 60, 80 y 100 mm.

Para las simulaciones, no se ha intentado reproducir el set up del túnel de viento, se ha asumido las condiciones como si el coche estuviera en una configuración de carretera.

El dominio usado se ha creado basándonos en las recomendaciones de Siemens para STAR-CCM (Siemens, 2020) (Figura 6.2). Una vez con el dominio creado, se continua con las condiciones de la simulación y el mallado.

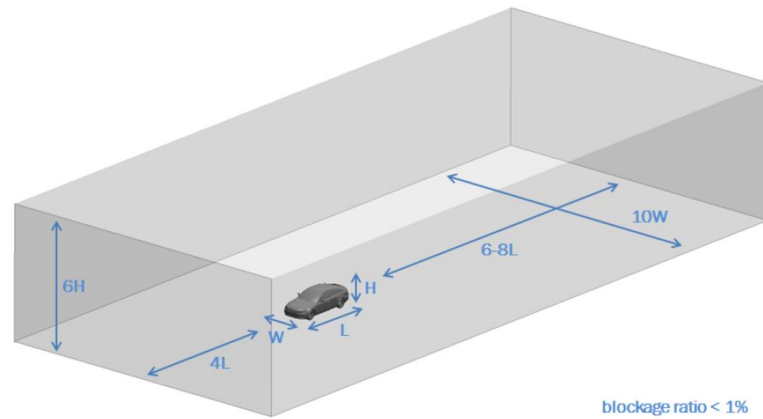


Figura 6.2 Dimensiones del dominio (Siemens, 2020)

### 6.1. Modelo de turbulencia

En este caso, se ha tratado de reproducir las condiciones del ensayo en el túnel de viento. La velocidad en los ensayos fue de 35m/s, con el suelo moviéndose a la misma velocidad y una intensidad turbulenta del 1%. La densidad era de  $1.16 \text{ kg/m}^3$  y la viscosidad ( $\mu$ ) era de  $1.851 \cdot 10^{-5} \text{ kg}\cdot\text{m}\cdot\text{s}^{-1}$ . Con todo esto, se obtiene un Re de  $2.34 \cdot 10^6$ .

Como condiciones de contorno (Figura 6.3), se ha impuesto la velocidad a la entrada y se ha dado la condición de presión de salida 0 Pa en el outlet. En el suelo se ha impuesto una velocidad igual a la de la entrada. El resto de paredes se han fijado como *symmetry*, como se recomienda en (Siemens, 2020).

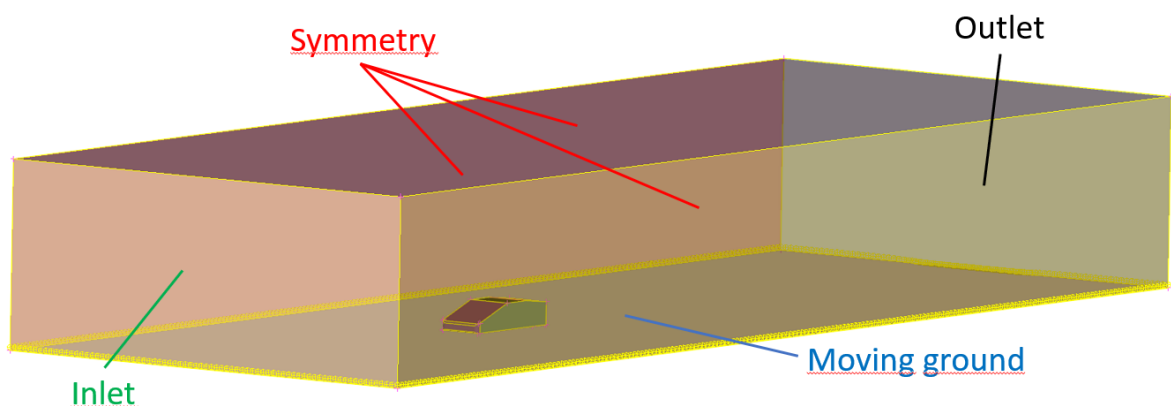


Figura 6.3 Condiciones de contorno del dominio

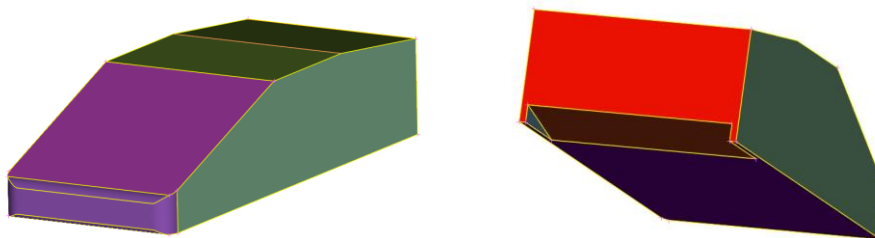
## 6.2.Malla

Para la malla se ha usado el software comercial ANSA. Se ha decidido utilizar este programa, porque, además de ofrecer una gran variedad de opciones y técnicas para el mallado, es uno de los programas que se tienen actualmente en la empresa. Por tanto, no implicaría ningún incremento de coste, ya que la preparación de los modelos para introducirlos a softwares comerciales se hace con esta herramienta.

Para el modelo simplificado se ha decidido usar una malla híbrida, dominante hexaédrica con elementos tetraédricos entre la capa límite y la malla exterior de hexaedros. Se han utilizado entre 3 y 5 celdas de transición entre la capa límite y el dominio estructurado.

Antes de entrar en el mallado, el proceso se ha realizado con un procedimiento de mallado *Batch Meshing*, que permite automatizarlo y asegurar mallas repetitivas de un modelo a otro (BETA CAE, 2022).

Para este modelo se ha decidido no usar funciones de pared por lo que se ha buscado un  $y^+ \approx 1$ . Para calcular la altura de la primera celda, se ha usado una de las diferentes calculadoras de  $y^+$  disponibles en internet. Con un  $Re$  de  $2.34 \cdot 10^6$ , una longitud característica de 1.07m y las propiedades del aire a las condiciones del ensayo, tenemos una altura aproximada de la primera celda de 0.03mm.



**Figura 6.4 Geometría modelo simplificado**

La capa límite consta de 19 capas con un *growth ratio* variable entre 1.2 y 1.5 similar a (Skaperdas, 2021). Las superficies del modelo se han dividido en varias zonas para asegurar un mallado correcto, este paso tendrá mucha más relevancia en el modelo complejo.

Se ha mallado superficialmente con tamaños de entre 1 y 5mm, refinando los bordes, las zonas con curvatura (parte frontal) y zonas como las paredes laterales del difusor. El tipo de celdas utilizadas para mallar la superficie del coche son triángulos.

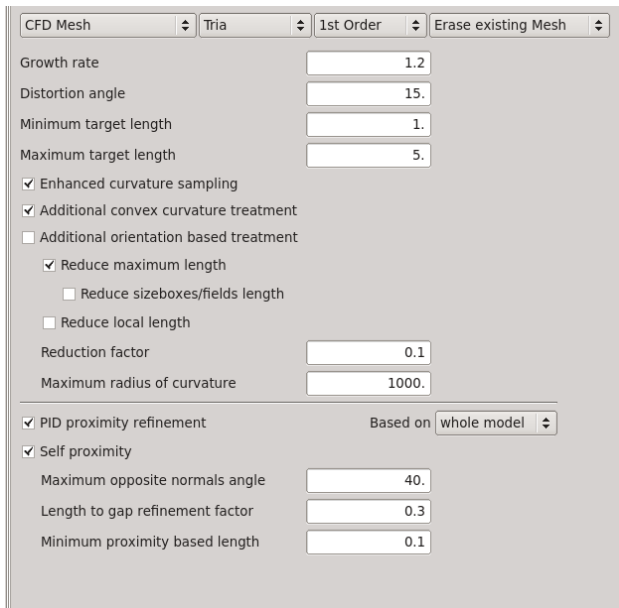


Figura 6.5 Set Up malla superficial

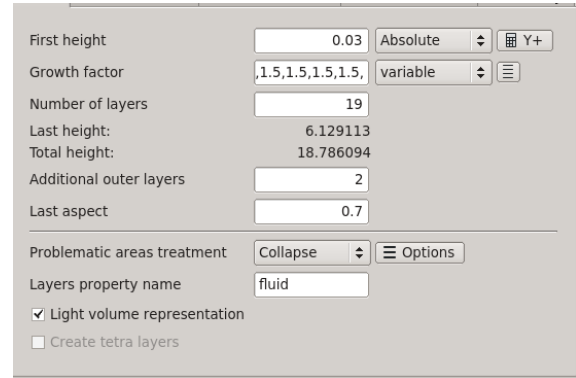


Figura 6.6 Set Up Malla capa límite

El dominio se ha mallado con un tamaño máximo de 80mm, mientras que el suelo se ha fijado a 60mm. Como recomienda (Siemens, 2020; Skaperdas, 2021), se han definido volúmenes de control alrededor del coche para un mayor refinamiento de las zonas de interés a partir de offset de la superficie.

La primera zona de refinamiento se ha hecho a una distancia de 100 mm del coche, extruyéndose detrás de éste 600mm. Esta zona se ha fijado con un tamaño máximo de 12mm.

El segundo volumen se ha hecho a 350mm del coche y se ha extendido hasta el final del dominio para asegurar una buena resolución de la estela del coche. En esta zona se ha fijado un tamaño máximo de malla de 24mm.

Finalmente, se ha refinado la zona del difusor (volumen formado por las líneas azules) con un tamaño máximo de 8mm.

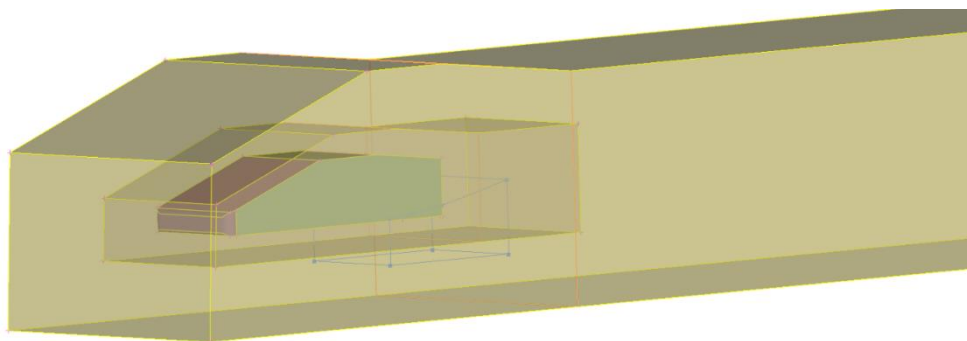


Figura 6.7 Volúmenes de refinamiento

Una vez definidas las condiciones del mallado. Se procede a mallar el dominio obteniendo la malla de la Figura 6.8. Podemos ver las zonas de refinamiento alrededor del coche. La malla tiene un total de 6.6 millones de celdas.

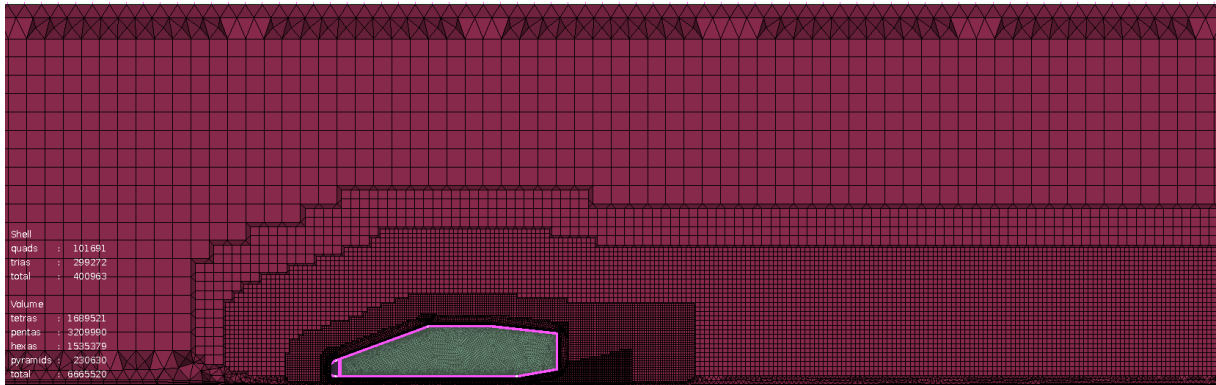


Figura 6.8 Malla Modelo Simplificado – Difusor 4º y RH=40mm

Si nos fijamos en algunos de los parámetros de calidad de la malla vemos los siguientes gráficos:

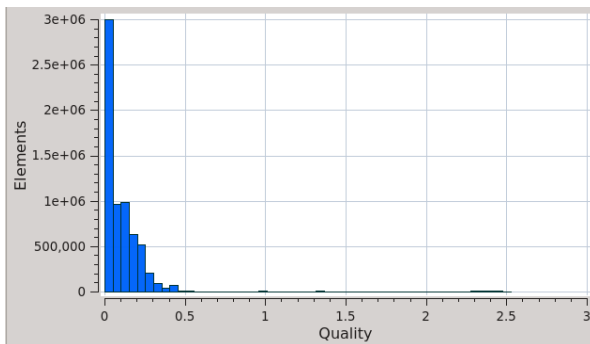


Figura 6.9 Skewness

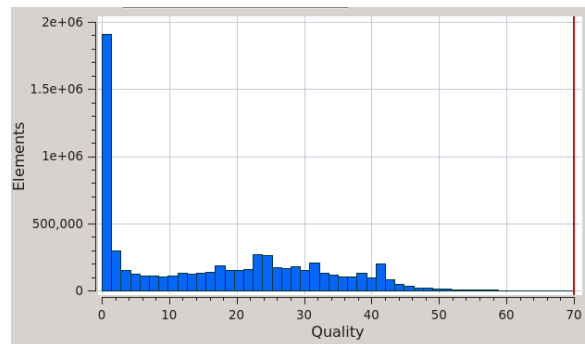


Figura 6.10 Non-orthogonality

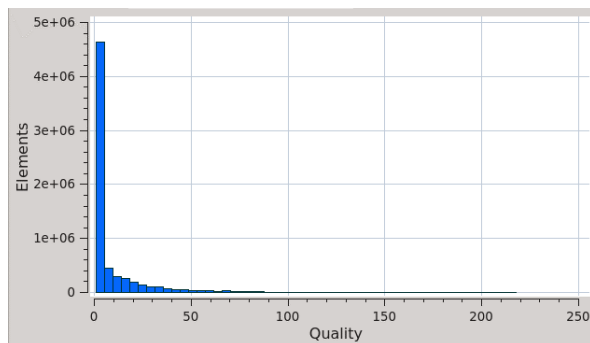


Figura 6.11 Aspect Ratio

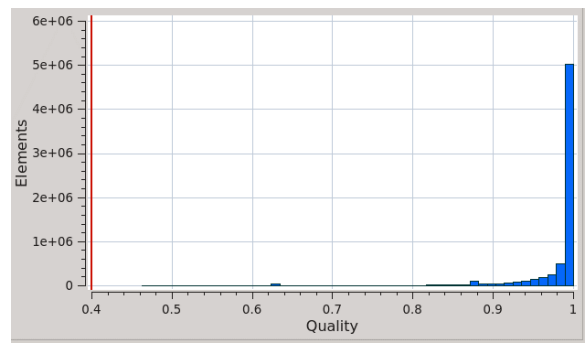


Figura 6.12 Jacobian

Vemos como todos ellos cumplen los requisitos de mallado de OpenFOAM. La non-ortogonalidad está por debajo de 70 y la *skewness* por debajo de 3. Si nos fijamos en el *aspect ratio*, vemos como nos salen celdas con AR mayores de 200. Estas celdas se

encuentran en la zona de la capa límite, y al tener  $y^+$  de uno, se entiende que aparezcan estos valores elevados. Finalmente, el jacobiano está por encima de 0.4, que es otro de los criterios de OpenFOAM.

Siendo una geometría simple, podría mejorarse la malla. Sin embargo, se decide continuar con esta malla ya que, cuando compliquemos la geometría, se ira haciendo más difícil conseguir mejores mallas que la que tenemos ahora.

Se ha utilizado el mismo proceso de mallado para el resto de configuraciones, obteniendo mallas con calidades y especificaciones similares.

### 6.3. SetUp

Para el set up, se han fijado las condiciones de contorno mencionadas en apartados previos. Se ha fijado una velocidad de entrada de 35 m/s y una presión de salida de 0 Pa.

En *internalField*, para este modelo, se ha fijado velocidad a (0, 0, 0) para inicializarla y la presión se ha fijado a 0 Pa. Para inicializar las variables de turbulencia se han usado las siguientes fórmulas ((ANSYS, 2021; Greenshields, 2011; OpenCFD Ltd., 2022)):

$$k = \frac{3}{2}(I \cdot U)^2 \quad \text{Ecuación 6-1}$$

$$\omega = \rho \frac{k}{\mu} \left(\frac{\mu_T}{\mu}\right)^{-1} \quad \text{Ecuación 6-2}$$

Sabiendo, por las propiedades del túnel, que la intensidad es menor que 1% y que  $\left(\frac{\mu_T}{\mu} = 10\right)$  es el ratio de viscosidad, obtenemos:

$$k = \frac{3}{2}(0.01 \cdot 35)^2 = 0.184 \quad \text{Ecuación 6-3}$$

$$\omega = \rho \frac{k}{\mu} \left(\frac{\mu_T}{\mu}\right)^{-1} = 1151.54 \quad \text{Ecuación 6-4}$$

En las paredes, las variables se han inicializado como:

**Tabla 3. Condiciones de contorno en las paredes – Modelo simplificado**

Variable	Tipo	Valor
U	fixedValue	(0, 0, 0)
p	zeroGradient	-
k	fixedValue	0
$\omega$	omegaWallFunction	1151.54
$\nu_t$	nutUSpaldingWallFunction;	0

El modelo de turbulencia usado es el SST k- $\omega$ . Queda por especificar los esquemas y los *solvers* lineales usados:

```

ddtSchemes
{
  default      steadyState;
}
gradSchemes
{
  default      Gauss linear;
}
divSchemes
{
  default      none;
  div(phi,U)   bounded Gauss linearUpwindV grad(U);
  div(phi,k)   bounded Gauss upwind;
  div(phi,omega) bounded Gauss upwind;
  div((nuEff*dev2(T(grad(U)))) Gauss linear;;
}
laplacianSchemes
{
  default      Gauss linear corrected;
}
interpolationSchemes
{
  default      linear;
}
snGradSchemes
{
  default      corrected;
}
    
```

**Figura 6.13 fvSchemes**

```

solvers
{
  "p.*"
  {
    solver          GAMG;
    tolerance        1e-10;
    relTol           0.05;
    smoother         GaussSeidel;
    nPreSweeps       0;
    nPostSweeps      2;
    cacheAgglomeration on;
    agglomerator     faceAreaPair;
    nCellsInCoarsestLevel 10;
    mergeLevels      1;
    maxIter          15;
  }
  "U.*"
  {
    solver          smoothSolver;
    smoother         GaussSeidel;
    tolerance        1e-10;
    relTol           0.1;
    nSweeps          1;
  }
  Phi
  {
    $p;
  }
  "(k|omega)"
  {
    solver          smoothSolver;
    smoother         GaussSeidel;
    tolerance        1e-10;
    relTol           0.1;
    nSweeps          1;
  }
}
SIMPLE
{
  nNonOrthogonalCorrectors 0;

  residualControl
  {
    p                1e-7;
    U                 1e-6;
    k                 1e-6;
    omega             1e-6;
    nuTilda           1e-6;
  }
}
    
```

**Figura 6.14 fvSolution**

Como podemos ver, en este caso no se han incluido correctores non-ortogonales. Finalmente, como factores de relajación, se han usado 0.3 para la presión y 0.7 para el resto. Al incluir estos factores, se consigue que las simulaciones sean más estables.

En el Anexo B.a, se puede encontrar una tabla resumen de todas las simulaciones realizadas para este modelo simplificado.

## 6.4. Resultados

Antes de analizar los resultados, se revisa que la simulación haya convergido correctamente. Si analizamos algunos historiales de convergencia, vemos los residuos más elevados son del orden de  $10^{-5}$ .

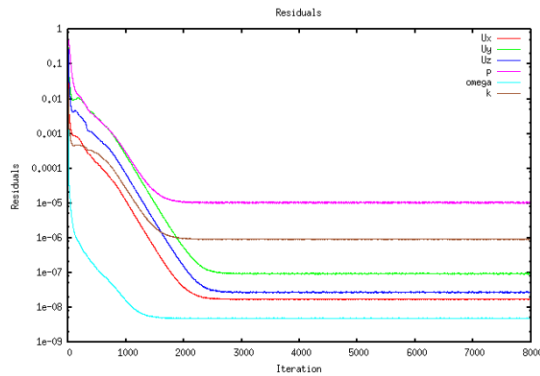


Figura 6.15 Residuales – Modelo Difusor 4º, RH=80mm y *linearUpwindV*

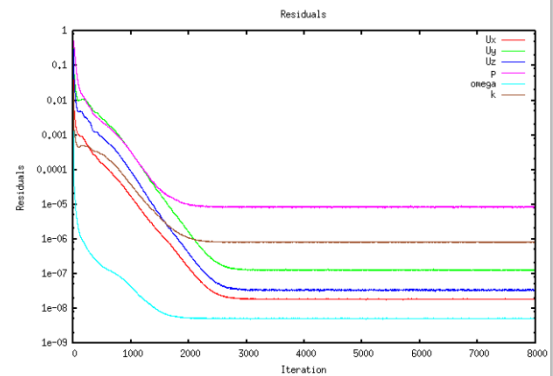


Figura 6.16 Residuales – Modelo Difusor 4º, RH=60mm y *linearUpwind*

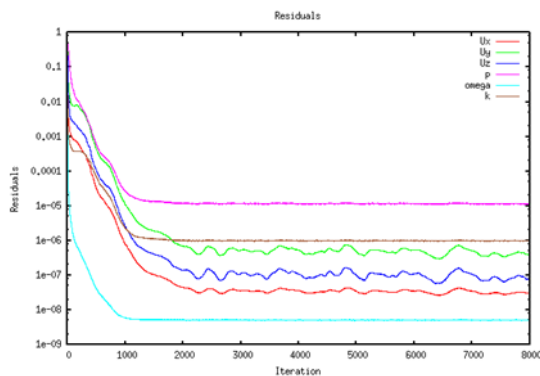


Figura 6.17 Residuales – Modelo Difusor 10º, RH=40mm y *linearUpwindV*

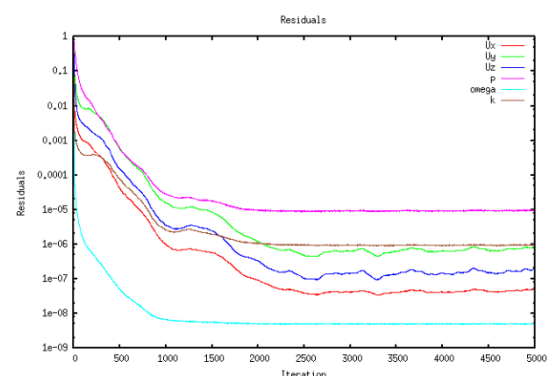


Figura 6.18 Residuales – Modelo Difusor 10º, RH=40mm y *linearUpwind*

Además, si nos fijamos en las variables de control ( $C_L$  y  $C_D$ ), vemos como esta también se ha estabilizado. Las pequeñas variaciones que se ven son menores al tercer decimal (Figura 9.1 y Figura 9.2). Por tanto, también se da por válido.

Finalmente, si graficamos el valor de  $y^+$  alrededor del coche, vemos como este tiene valores de alrededor de  $y^+=1$ . Como podemos ver, se observan valores máximos de  $y^+=3$ , pero la mayoría del coche tiene valores alrededor de 1. Incluso para los máximos, la malla de capa límite es suficientemente fina para resolverla correctamente con el modelo de turbulencia usado.

Habiendo verificado todo esto, procedemos a hacer un análisis de los resultados obtenidos.

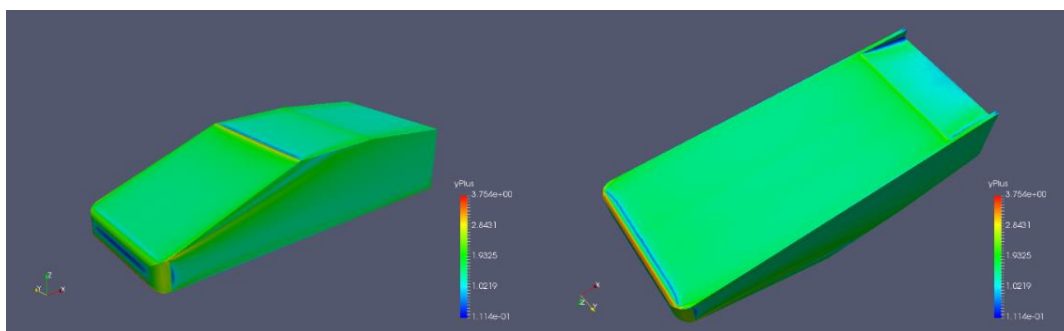


Figura 6.19 Contorno de  $y^+$  - Modelo Simplificado – Difusor 10° y RH=60mm

#### 6.4.1. Comparativa entre mallas y esquemas

El objetivo de este apartado era intuir que tipo de malla es más adecuada para resolver con OpenFOAM. Sabemos que los softwares comerciales son capaces de lidiar con mallas de peor calidad y con mallas de todo tipo (poliédricas, hexaédricas, tetraédricas). Por tanto, queríamos analizar el comportamiento del historial de convergencia para varias mallas diferentes. Hay que tener en cuenta que se han utilizado los mismos set ups para ambas mallas, que esto es solo para tener una primer análisis y que se necesitarían una mayor cantidad de simulaciones para concluir en cualquier dirección.

Tras analizar los resultados (de Figura 6.20 a Figura 6.25), vemos como con la malla hexaédrica parece que tenemos una mejor convergencia y constancia de un modelo a otro. Esto concuerda con el set up definido. Por definición, las mallas tetraédricas tendrán una mayor non-ortogonalidad. Esto implica, la necesidad de incluir correctores non-ortogonales. Además, se recomienda usar *leastsquares* para resolver los gradientes en estos casos (Hvroje Jasak, 2016).

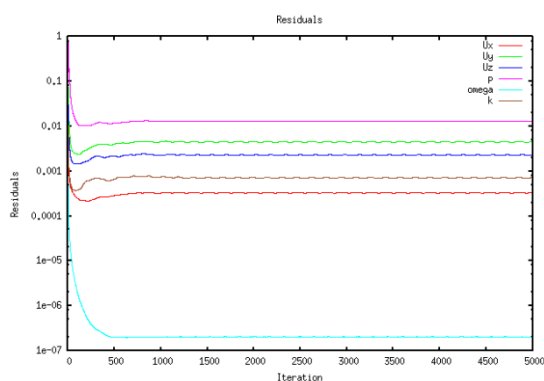


Figura 6.20 Residuales – Modelo Difusor 10°, RH=40mm, malla tetra, gradScheme(cellLimited leastsquares)

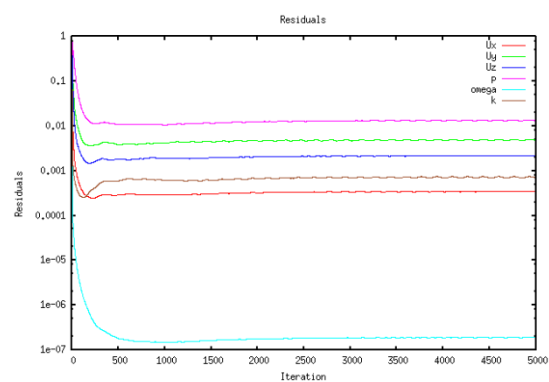
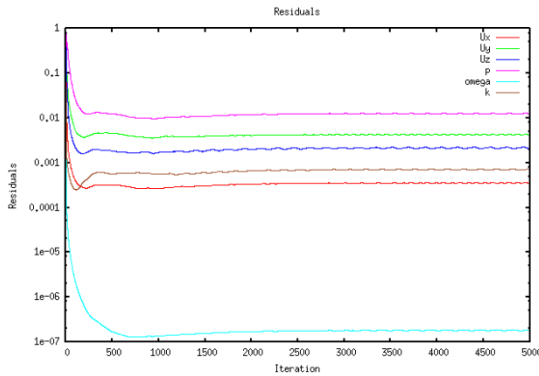
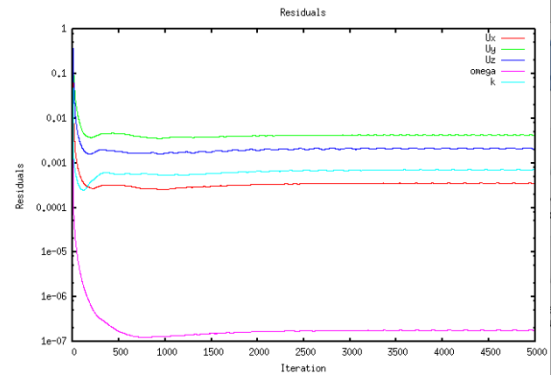


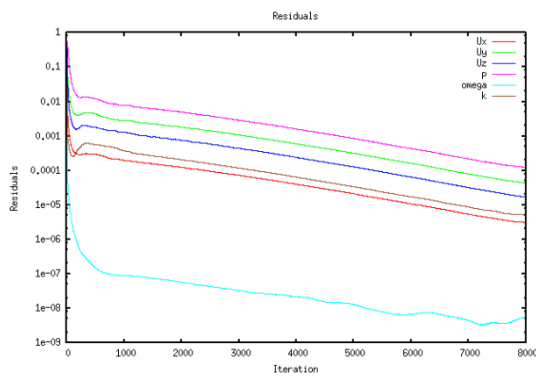
Figura 6.21 Residuales – Modelo Difusor 10°, RH=40mm, malla tetra, gradScheme(Gauss Linear)



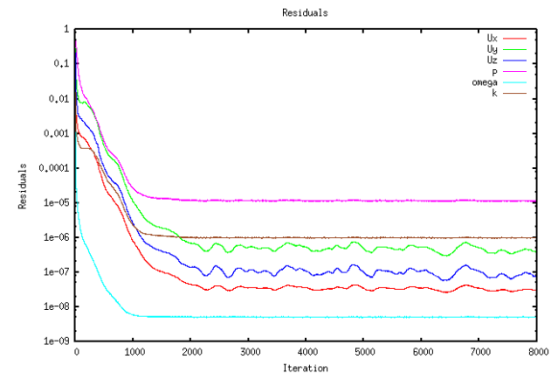
**Figura 6.22 Residuales – Modelo Difusor 10º, RH=60mm, malla tetra, gradScheme(Gauss Linear)**



**Figura 6.23 Residuales – Modelo Difusor 10º, RH=60mm, malla tetra, gradScheme(Gauss Linear), con 2 correctores de non-ortogonalidad**



**Figura 6.24 Residuales – Modelo Difusor 10º, RH=100mm, malla tetra, gradScheme(Gauss Linear)**



**Figura 6.25 Residuales – Modelo Difusor 10º, RH=100mm, malla Hexa, gradScheme(Gauss Linear)**

Comparando los resultados con la simulación que usa el esquema de leastsquares, no se ha apreciado ninguna diferencia significativa entre ambos historiales (Figura 5.23 y 5.24). También se ha simulado la misma malla con correctores de non-ortogonalidad y tampoco no se han observado diferencias significativas para este caso (Figura 6.22 y Figura 6.23).

Si nos fijamos en el caso del difusor de 10º, los resultados tienen una mejor tendencia que en los otros dos casos. Por tanto, analizaremos los valores de  $C_L$  y  $C_D$ . En ambos casos, los valores se han estabilizado hasta su cuarto decimal obteniendo unos valores de -0.417 y 0.269 para el caso de la malla tetraédrica, y de -0.411 y 0.255 para el caso de la malla hexaédrica. Ambos resultan en valores que entran dentro de tolerancias vistas en las diferentes pruebas realizadas.

Aunque se ha visto que, en términos de resultados, pueden dar valores similares, se decide continuar con una malla hexaédrica dominante para el caso complejo, ya que el proceso parece más robusto.

Finalmente, usamos las simulaciones convergidas como caso inicial y modificamos los esquemas de  $k$  y  $\omega$  a esquemas de mayor orden. Para ello se ha usado el caso con el difusor a  $10^\circ$  y una RH de 100mm. Se han usado dos esquemas, *vanLeer* y *linearUpwind*. Si primero analizamos la convergencia, vemos como, sobre todo los residuos de la energía cinética turbulenta ( $k$ ), los residuos han subido algún orden.

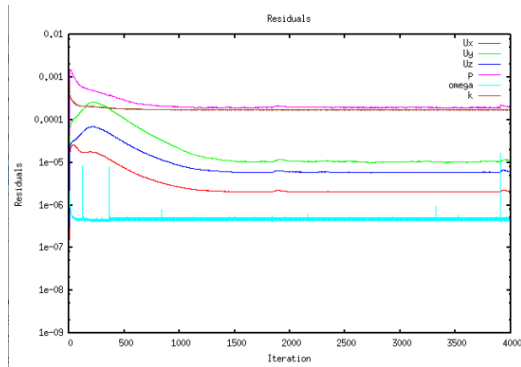


Figura 6.26 Residuals – Modelo Difusor  $10^\circ$ ,  
 RH=100mm, divScheme (bounded Gauss vanLeer)

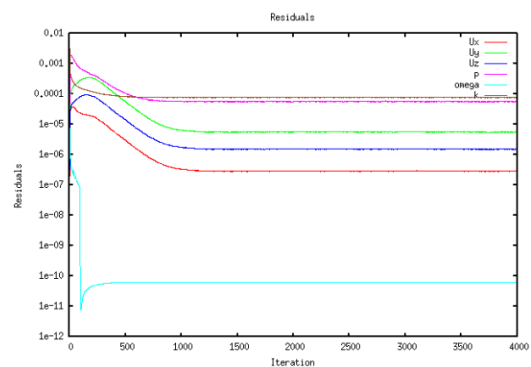


Figura 6.27 Residuals – Modelo Difusor  $10^\circ$ , RH=100mm,  
 divScheme(bounded Gauss linearUpwind grad(k/omega))

Tabla 4. Casos simulados – Modelo simplificado

Coeficiente	<i>Upwind</i>	<i>linearUpwind</i>	<i>vanLeer</i>
$C_L$	-0.411	-0.411	-0.412
$C_D$	0.255	0.250	0.252

Comparando los valores de los coeficientes de sustentación y de resistencia obtenidos (Tabla 4), vemos como las diferencias entre ellos son mínimas. Por tanto, para reducir tiempo de computación, se decide continuar este análisis con el esquema *upwind* para las variables  $k$  y  $\omega$ . Sin embargo, habrá que tener en cuenta esto y que estamos introduciendo un error de discretización mayor.

#### 6.4.2. Comparativa resultados experimentales

Como se ha comentado previamente, se tienen resultados experimentales de este modelo. Importante mencionar que, aunque los valores de  $C_L$  si fueron repetitivos entre diferentes ensayos realizados, dentro del margen de error de la medida (Figura 6.28), se obtienen diferencias significativas en el  $C_D$  (Figura 6.29). Además, se puede observar cómo, en el año 2016, el desprendimiento del difusor sucede significativamente antes de lo que se observa en el año anterior.

Con esto en mente, vamos a comparar los resultados obtenidos.

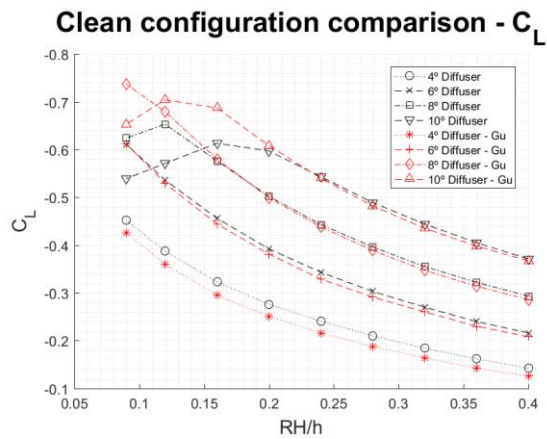


Figura 6.28 Comparativa resultados  $C_L$  2015 vs 2016  
 ((Lanuza, 2016) , Fig 5.3 – p38)

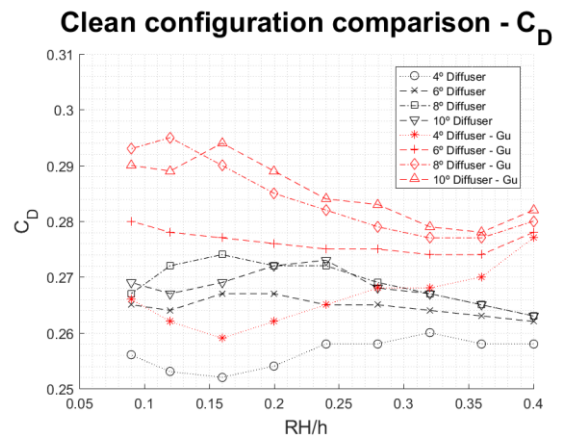


Figura 6.29 Comparativa resultados  $C_D$  2015 vs 2016  
 ((Lanuza, 2016) , Fig 5.4 – p39)

Vemos como en los gráficos de  $C_L$  se obtiene una tendencia similar a la vista en los ensayos, con resultados cercanos a los experimentales. En el caso del difusor inclinado  $4^\circ$ , vemos como la pendiente en la que crece parece menor que la que se observa en los ensayos. En el caso del difusor de  $10^\circ$ , el punto de desprendimiento no se captura correctamente. De todas formas, se asemeja a los resultados vistos en el año 2015, donde el difusor no se había desprendido y se tenía un valor de alrededor de -0.7.

Comparando el drag, los resultados no se asemejan a los experimentales de ninguno de los dos años. Sin embargo, éstos están dentro del rango definido por los resultados obtenidos en ambos años.

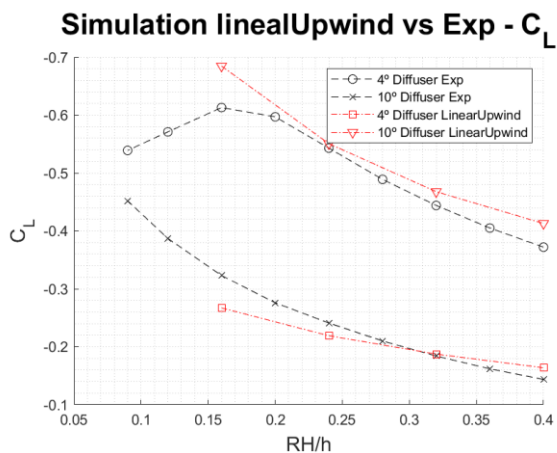


Figura 6.30 Comparativa simulación con resultados experimentales -  $C_L$

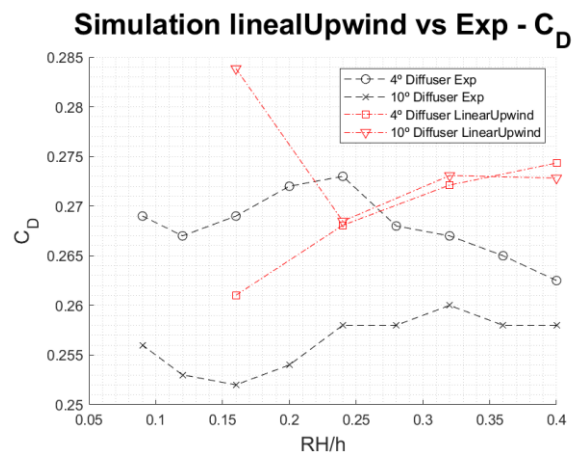
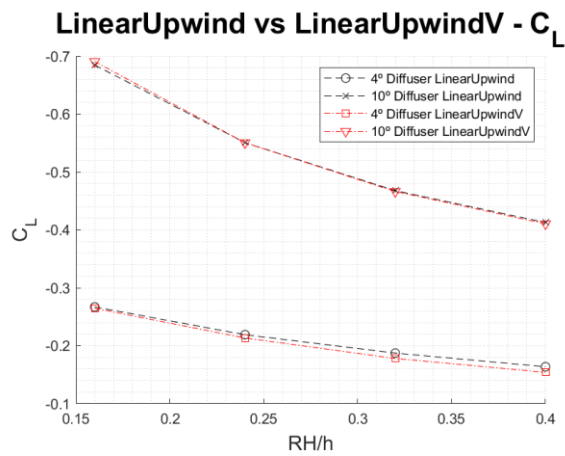


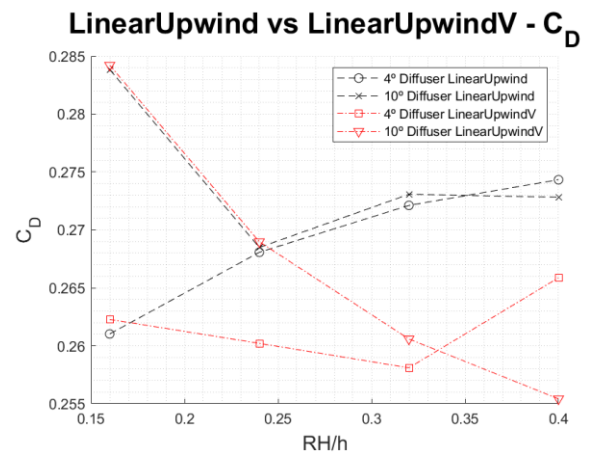
Figura 6.31 Comparativa simulación con resultados experimentales -  $C_D$

Si ahora comparamos los resultados obtenidos usando los dos esquemas diferentes de *linearUpwind*, para el  $C_L$ , se obtienen curvas casi idénticas (Figura 6.32). Sin embargo, al

comparar el  $C_D$ , vemos como tenemos diferencias significativas en los resultados, teniendo diferencias de hasta 15 puntos.

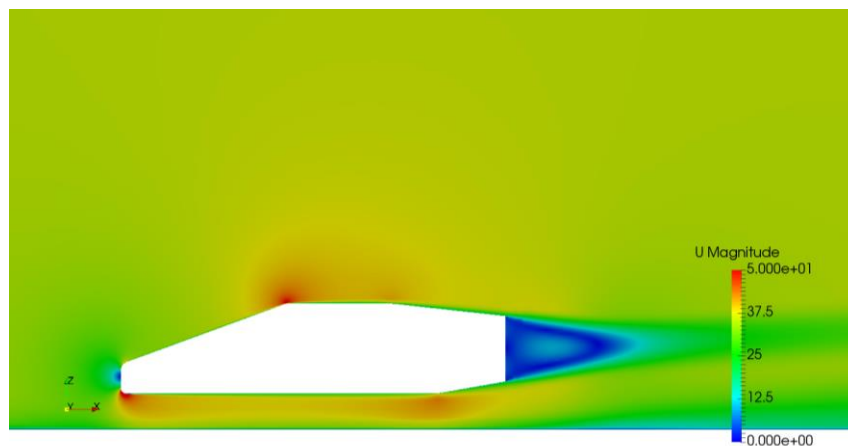


**Figura 6.32** Comparativa linearUpwind vs linearUpwindV -  $C_L$



**Figura 6.33** Comparativa linearUpwind vs linearUpwindV -  $C_D$

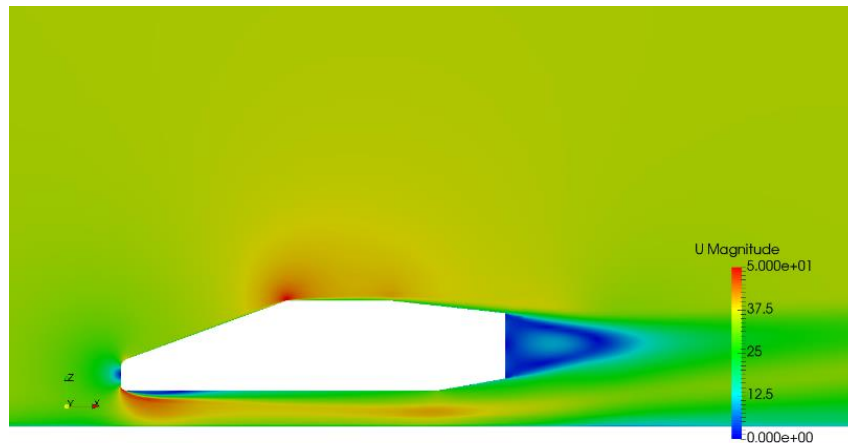
Estas diferencias pueden deberse a varias razones. La primera, y que seguro tiene un efecto, es que la malla no está suficientemente refinada en la estela. Como ya hemos comentado previamente, la mayoría de la resistencia que experimenta el coche se debe a la estela que se forma detrás de él. Por tanto, si no somos capaces de resolverla correctamente, puede haber diferencias. Como se menciona en Anexo B.b.ii., el esquema linearUpwindV usa el limitador más restrictivo, que se obtiene en la dirección en que la variable cambia más rápido, para limitar el resto de componentes.



**Figura 6.34** Contorno de velocidades en  $y=0$  – linearUpwindV

Si analizamos el contorno de velocidades en el plano central (Figura 6.34 y Figura 6.35), se aprecian diferencias en la estela que pueden afectar al  $C_D$  obtenido. Sin embargo, la mayor diferencia la vemos en la Figura 6.35 con la aparición de un desprendimiento en la parte

delantera del vehículo, la cual tiene un efecto también en la estela. La presencia de esta diferencia entre ambos resultados, puede ser un indicativo de la necesidad de un refinamiento en esa zona de la malla para capturar correctamente su comportamiento.

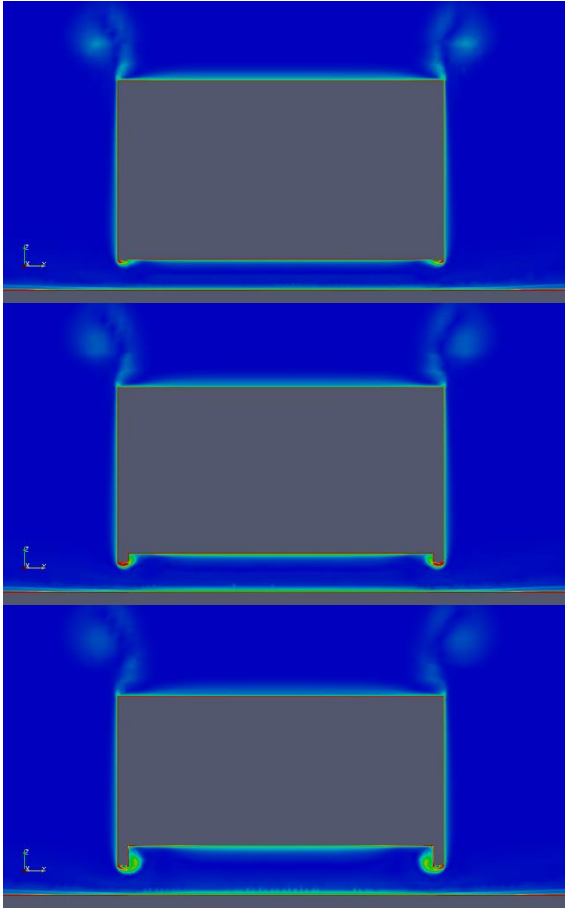


**Figura 6.35 Contorno de velocidades en  $y=0$  - linearUpwind**

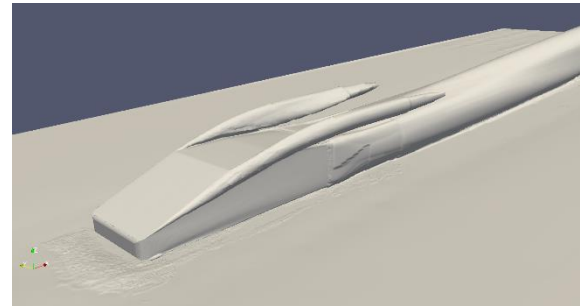
Si ahora analizamos el flujo en el difusor para uno de los casos, podemos ver como se forman dos vórtices contra rotantes en los laterales (Figura 6.36). Este fenómeno se ha observado experimentalmente (Zhang et al., 2006), y ayuda a que el flujo de las esquinas a adherirse de nuevo. En las secciones más atrasadas del modelo vemos como el centro comienza a mostrar un ligero incremento de la vorticidad, pudiendo indicar un principio de separación del flujo. Sin embargo, vemos como en las esquinas, en la zona próxima a la pared del difusor, se ve el contorno similar al del comienzo. En este estado, el difusor sigue ganando carga aerodinámica conforme reducimos la altura con respecto al suelo (RH).

Otro fenómeno común, son los vórtices que se generan en la zona del pilar. Este fenómeno se ha analizado ampliamente (Larry Deaton et al., 2018; Nakamura et al., 2020), y es de gran relevancia a la hora de diseñar un coche. Esto se debe a la influencia de estos vórtices en la generación de drag, y debido al ruido que se genera al interactuar con las ventanillas bajadas cuando se circula a velocidades de entre 50-120 km/h (Larry Deaton et al., 2018).

Habiendo definido un primer modelo que funciona en diferentes configuraciones, que ofrece resultados del mismo orden que los experimentos y en el que se observan patrones relevantes para la aerodinámica del vehículo, se decide continuar el estudio con el modelo más complejo y completo.



**Figura 6.36** Contorno de vorticidad en  $x=0.89\text{m}$ ,  
 $x=0.95\text{m}$  y  $x=1.07\text{m}$

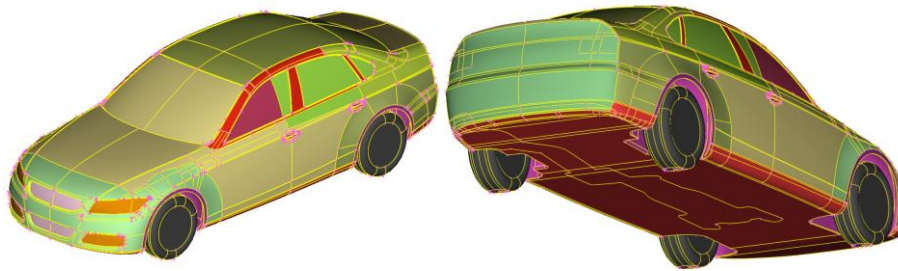


**Figura 6.37** Isosuperficie de la energía cinética  
turbulenta ( $k$ )

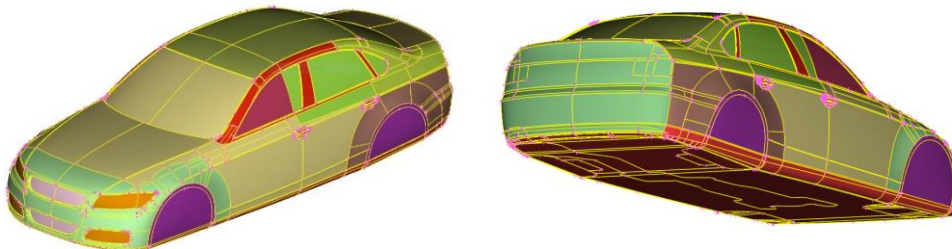
## 7. Modelo de DrivAer

### 7.1. Geometría

La geometría del vehículo utilizada se ha descargado de (TUM School of Engineering and Design, n.d.). La geometría que se ha decidido ensayar es el caso con el suelo del coche sin detalles (plano), sin las entradas de refrigeración ni retrovisores. Se han ensayado el modelo Notchback (Figura 5.3 - N, Figura 7.1 y Figura 7.2). Se han usado las ruedas simplificadas y no se han incluido los retrovisores. También se han incluido simulaciones sin ruedas.

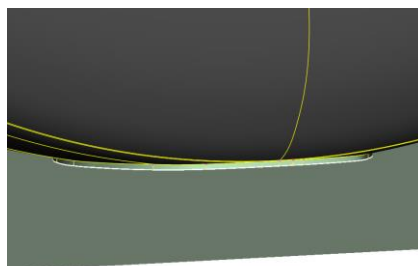


**Figura 7.1 Modelo DrivAer NotchBack – Con ruedas**



**Figura 7.2 Modelo DrivAer NotchBack – Sin ruedas**

Una zona particular a la hora de simular vehículos es la zona de contacto entre el neumático y el asfalto. Se ha establecido como mejores prácticas añadir una extrusión de 2-10 mm para poder conseguir un mejor mallado en la unión del asfalto con la rueda (Figura 7.3). En nuestro caso, se ha usado una extrusión de 2mm.



**Figura 7.3 Zona unión neumático-asfalto**

El dominio para la simulación, de nuevo, se ha definido como en la Figura 6.2.

## 7.2. Condiciones de la simulación

Para la simulación, se han usado las condiciones de ensayo de (James et al., 2018). Primero, es importante posicionar el coche a la altura del ensayo y con la misma inclinación. En el ensayo, el vehículo se posicionó con una altura de 683 mm en la parte superior del paso de rueda frontal y a 679 mm del paso de rueda trasero.

Una vez posicionado el coche, hay que fijar las condiciones de ensayo. En el ensayo se realizó con ruedas y suelo estático. Esto genera la indefinición de que no conocemos el perfil de la capa límite en la sección de ensayo. Durante el proceso de simulación, también se harán pruebas fijando velocidad suelo.

La velocidad impuesta en el ensayo es de 140km/h con un  $Re \approx 11.8 \cdot 10^6$  y una densidad de  $1.217 \text{ kg/m}^3$  (James et al., 2018; Varney et al., 2020). Como condiciones de contorno, se hace un ejercicio similar al de la Figura 6.3. La presión a la salida se fija a 0 Pa, los laterales y la parte superior del dominio se fijan como *symmetry*.

El área frontal de vehículo es de  $2.12 \text{ m}^2$  y la longitud característica es de 4.61m.

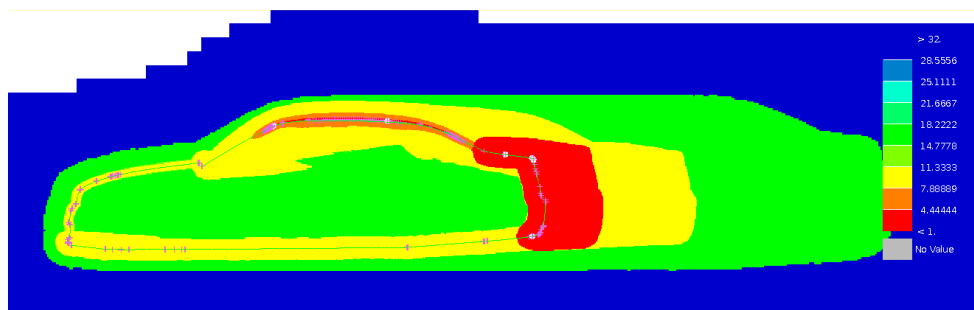
## 7.3. Proceso de mallado

Para el proceso de mallado, se ha decidido utilizar una malla hexaédrica. La diferencia con el modelo simplificado es que se va a usar la herramienta HeXtreme de ANSA. Se ha decidido usar este tipo de malla ya que, el hecho de usar una zona de transición con mallas tetraédricas puede dar problemas en los resultados y de convergencia en esa zona (Skaperdas, 2021; University of Oxford, 2021). El proceso de mallado consiste en partir de una malla de todo el dominio con el valor definido. A continuación, empieza a dividir cada hexaedro en hexaedros más pequeños, siempre dividiéndolo por 8 hexaedros de dimensión  $x/2$ , siendo  $x$  la dimensión del hexaedro del que se parte. Cuando nos acercamos a las paredes del vehículo, se cortan estos hexaedros adaptándolos a la superficie según los parámetros definidos. Este tipo de mallado sigue un proceso similar, aunque con una interfase más amigable para el usuario, que *snappyHexMesh* (mallador en OpenFOAM).

Para esta simulación, se definen características superficiales dependiendo de la zona del coche. Se presta especial atención a la malla superficial alrededor de los pilares y la zona lateral posterior. Si nos fijamos en los diferentes colores de la Figura 7.1 y la Figura 7.2, cada

uno de ellos hace referencia a un PID diferente, con unas determinadas condiciones de malla. Los tamaños de malla usado para definir la superficie oscilan entre 1mm y 16mm dependiendo del caso y la malla.

A diferencia que en el caso simplificado, se han creado volúmenes de control a partir de un offset virtual hecho por el mallador (Figura 7.4). Se han creado volúmenes alrededor de todo el coche a una distancia de 500 y 200 mm que se extienden en la estela del coche. Además, se han añadido el mismo tipo de volúmenes en las ruedas, en la zona del pilar y en la parte trasera del coche, extendiéndose en la dirección principal del flujo. Particular atención se ha tenido en la zona de la malla entre las ruedas y el asfalto, llegando a valores de malla de 0.25mm.



**Figura 7.4 Volúmenes virtuales – Mallado con offset de las superficies**

Estos volúmenes han variado ligeramente conforme evolucionaba el proyecto, refinando algunas zonas con respecto a simulaciones previas, o variando la asignación de PIDs.

Respecto a la capa límite, se ha trabajado tanto con mallas con  $y^+ \approx 1$ , como con mallas con  $y^+ > 30$ . El grosor de la malla de capa límite en ambos casos ha sido de unos 14mm.

Para la malla con  $y^+ \approx 1$ , se han utilizado 19 capas, con una ratio de crecimiento entre 1.2 y 1.6 para obtener un grosor suficiente. El valor de la primera celda ha sido de 0.02mm. Se ha asegurado que no hubiera un salto muy grande entre la transición de la última celda de la capa límite y la primera celda fuera de ésta (Figura 7.6).

Como podemos ver en la Figura 7.5, se obtiene un valor de  $y^+$  adecuado a lo largo de todo el vehículo para el modelo de turbulencia sin funciones de pared escogido.

Para la malla usada con funciones de pared, se han usado 6-7 capas, con una ratio de crecimiento de 1.2 y una altura de la primera celda de 1.05mm. De nuevo, se ha verificado que no se ven diferencias muy grandes en la zona de transición (Figura 7.8).

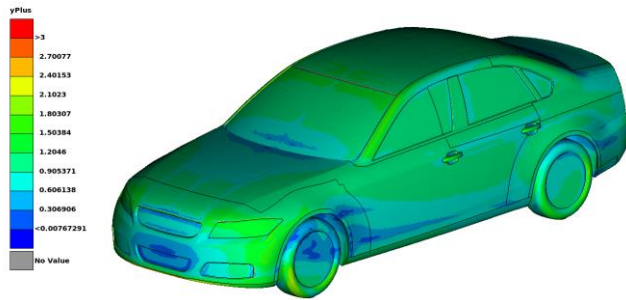


Figura 7.5 Contorno de  $y^+$  malla  $y^+ \approx 1$

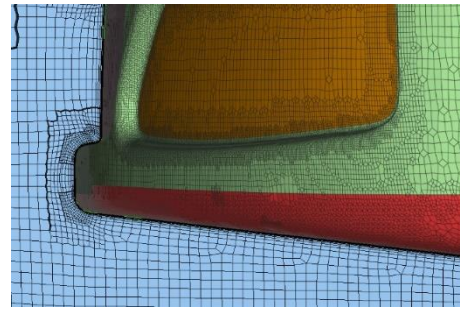


Figura 7.6 Malla zona capa límite  $y^+ \approx 1$

Si vemos el contorno de  $y^+$ , se observa que gran parte del vehículo, excepto la zona de estancamiento del flujo, donde las velocidades son muy bajas, el valor de  $y^+$  esta fuera de la zona de transición. Por tanto, se obtiene una malla adecuada para resolver la zona de la pared correctamente.

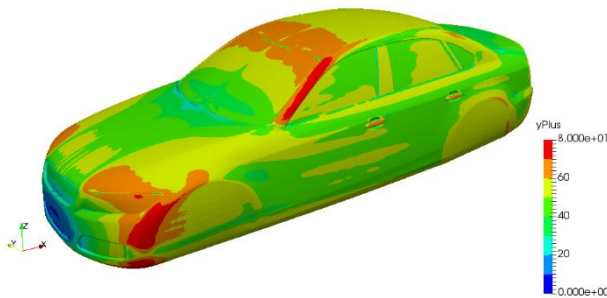


Figura 7.7 Contorno de  $y^+$  malla  $y^+ \approx 30$

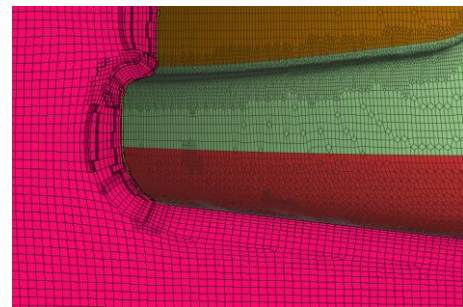
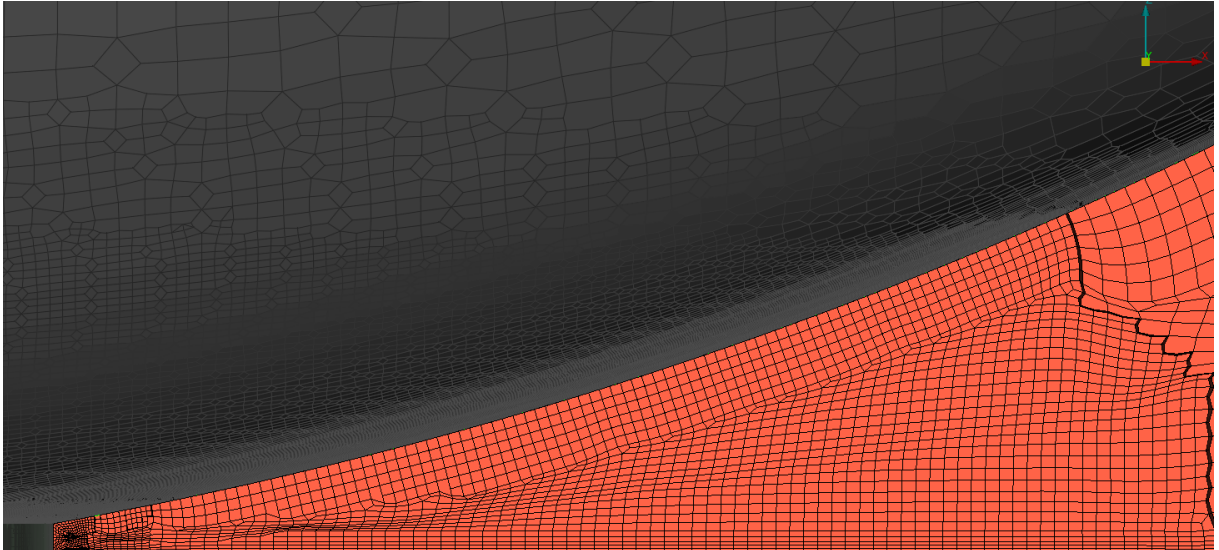
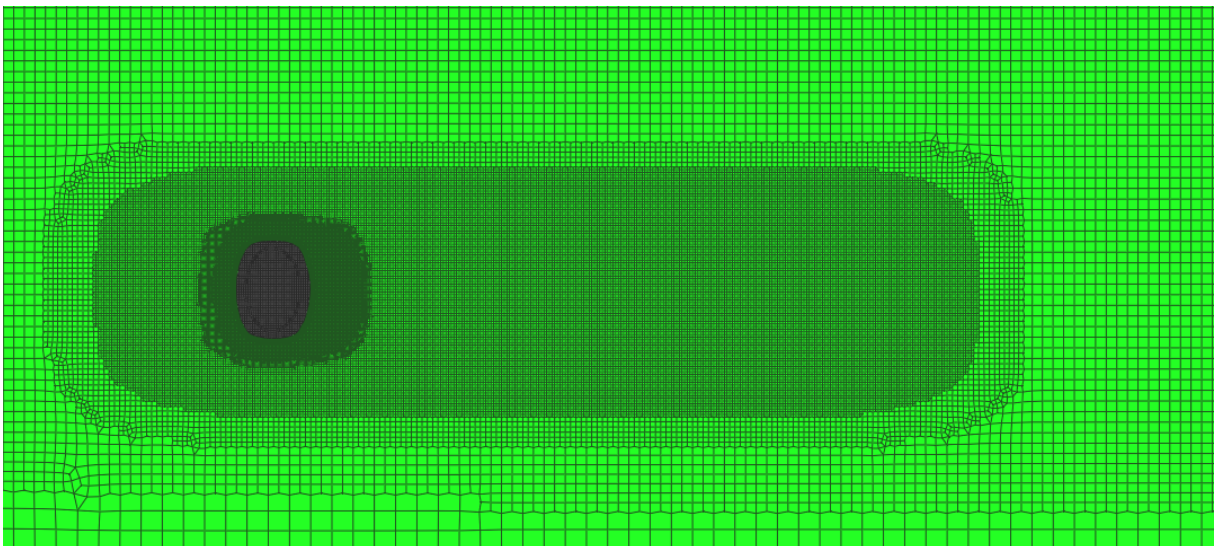


Figura 7.8 Malla zona capa límite  $y^+ \approx 30$

Finalmente, se ha prestado una especial atención a la malla en la zona de la unión de la rueda con el asfalto (Figura 7.9). Para mallar esta zona se ha activado la opción de poder reducir el número de capas en zonas conflictivas, para así poder adaptar la malla. Esto conlleva a que no se obtengan calidades de celdas especialmente malas en esta zona. El tamaño de malla usado ha sido de 0.25mm, y se ha creado una extrusión virtual para generar una malla apropiada en la estela (Figura 7.10).



**Figura 7.9 Malla – Zoom zona de contacto neumático y asfalto**



**Figura 7.10 Refinado de la malla en la zona de contacto y estela de la rueda.**

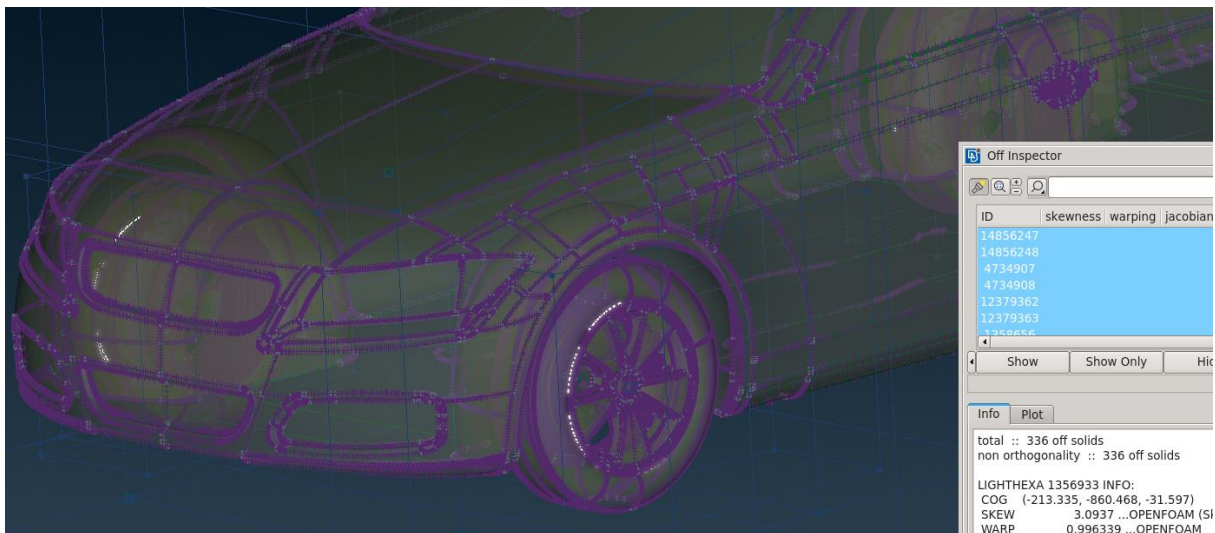
#### 7.4.SetUp

El set up de inicio ha sido el mismo que el descrito en el modelo simplificado. Se han actualizado los valores de  $k$ ,  $\omega$  y la velocidad a la entrada para adaptarlo a este caso. Además, para estos casos, se han usado entre 1 y 4 correctores non-ortogonales.

## 7.5.Resultados

En este apartado, haremos un recorrido por las diferentes configuraciones y modificaciones que se han ido haciendo, tanto en el mallado como en el set up, hasta el estado final.

Se empezó trabajando con el modelo con llantas, igual que en el estudio experimental que se usa como referencia. Sin embargo, tras varios intentos de mallado, aparecían valores de non-ortogonalidad por encima de 80 en la zona de unión entre el neumático y la llanta (Figura 7.11). Como la non-ortogonalidad es uno de los parámetros que más influye en OpenFOAM, se decidió simplificar la geometría para mejorar la calidad de malla en esa zona. Importante notar, que esta malla fue antes de implementar los mallados por superficies con offset virtual existente en ANSA. Existe la posibilidad que, usando este método, se pudiera mejorar la malla en esa zona.



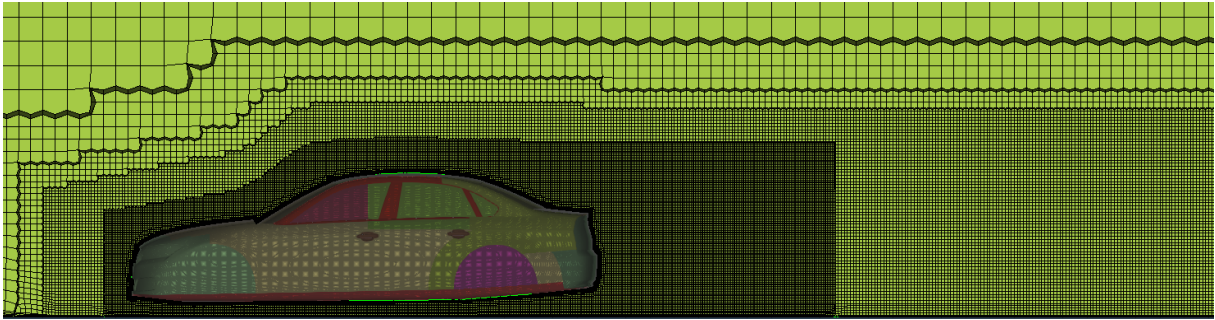
**Figura 7.11 Malla de baja calidad unión llanta-neumático**

Tras esto, se trabaja para solventar estas zonas con celdas de baja calidad. Para ello, se toman dos caminos paralelos. El primero es trabajar con un modelo sin ruedas y el otro es omitir la llanta, simplificando la rueda.

### 7.5.1. Modelo de DrivAer sin ruedas

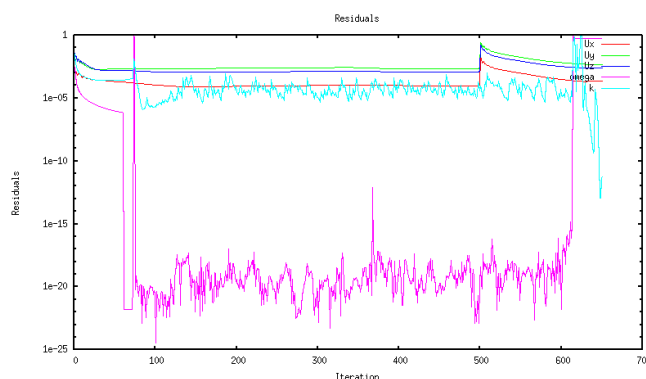
En este apartado se resumirán las diferentes simulaciones que se han hecho. Aunque se ha dividido en dos bloques, para diferenciar y comparar resultados, se han ido realizando modificaciones en paralelo en ambas configuraciones.

La primera malla que se ha ensayado es una malla de 31 millones de celdas, construida fijando tamaños de malla superficial por PIDs y mediante dos volúmenes de control del tipo “size box”. Los tamaños de malla usados se seleccionaron a partir de (Siemens, 2020). Una vez construida, se ha usado la opción de reparar la calidad. Tras una reunión con Beta CAE, se nos explica que no está recomendado usar esta opción con las mallas construidas con HeXtreme. Por ello, se descarta para futuras mallas.



**Figura 7.12 Malla Modelo 1 - DrivAer sin ruedas**

Con esta malla y fijando los esquemas de divergencia como linearUpwindV para la velocidad y linearUpwind para  $k$  y  $\omega$ , se procede a realizar la simulación. Además, se fija como esquema laplaciano un *Gauss linear corrected* y *snGradSchemes* se ha usado también *corrected*. Tras 800 iteraciones, la simulación diverge. Se acaban obteniendo valores de omega del orden de  $7 \cdot 10^{148}$ . Para el próximo caso, se inicializará la simulación con esquemas de primer orden para tratar de tener un primer resultado, antes de usar esquemas de orden superior.



**Figura 7.13 Historial de convergencia Modelo 1 - DrivAer sin ruedas**

El siguiente paso fue, siguiendo las recomendaciones de Beta CAE, modificar el proceso de mallado para evitar el uso de la herramienta de reparar la calidad de la malla. Por esta razón, y ya que permite un mejor control, se cambia la metodología y se usan offset virtuales de las

propias superficies del vehículo para el mallado. Además, tras la prueba, diferentes aspectos vistos en el modelo con ruedas y el incremento de celdas debido a refinamientos, se decide continuar con una malla con  $y^+$  mayor que 30.

Esta nueva malla tiene 30 millones de celdas, con una calidad que satisface los criterios recomendados de OpenFOAM. Se han creado 3 volúmenes de control principales, los cuales se extienden 3m, 8m y hasta el final del dominio con un tamaño de malla de 16, 32 y 128mm. Los primeros dos volúmenes tienen un offset de 200 y 500 y se han creado mediante un offset virtual. El tercero se ha hecho mediante una *size box* (Figura 9.7) y se distancia, aproximadamente, tres metros de la parte frontal del vehículo y 0.8 metros de la parte superior. Además, a parte de un control superficial, se ha añadido un volumen de refinamiento en la parte inferior del coche, con un tamaño de malla de 8mm.

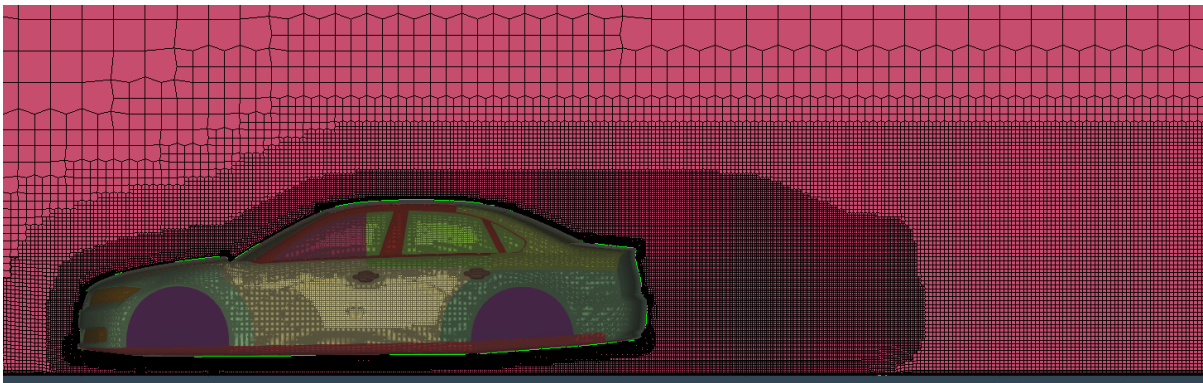


Figura 7.14 Malla Modelo 2 - DrivAer sin ruedas

Con esta malla, se modifican los esquemas numéricos a upwind debido a los problemas vistos en el caso anterior. Además, se cambia el esquema laplaciano a *Gauss linear limited 0.5*, el *snGradSchemes* a *limited 0.5* y el *gradSchemes* a *cellLimited Gauss Linear 1*. Se ha decidido empezar así y asegurar una buena convergencia antes de usar esquemas de mayor orden.

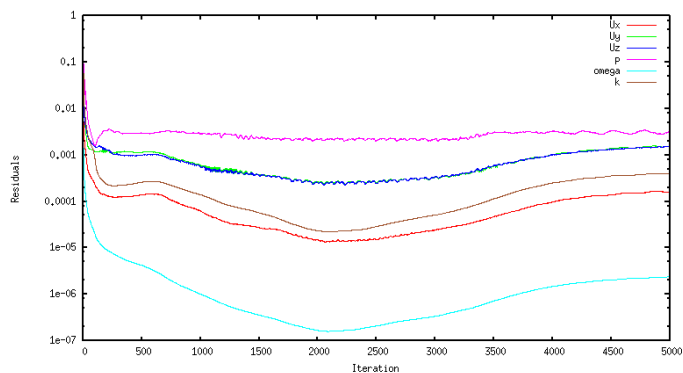
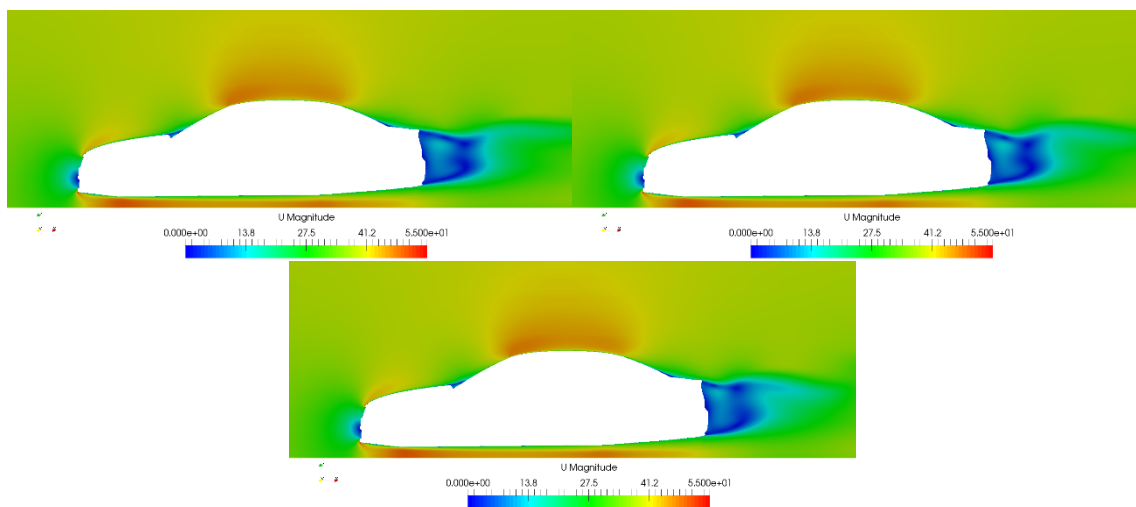


Figura 7.15 Historial de convergencia Modelo 2 - DrivAer sin ruedas

Si analizamos los contornos de velocidad, vemos fluctuaciones en la estela. Estas fluctuaciones pueden afectar a los residuos y ser una de las causas de que no sigan bajando a valores menores. Sabemos que el flujo que estamos analizando es transitorio y fluctúa con el tiempo. Al haber hecho la simplificación de flujo estacionario, tratamos de encontrar una solución promedio. Sin embargo, dependiendo de diferentes criterios, entre ellos la malla, puede ser que no se acabe encontrando esta solución estacionaria en el tiempo y nos fluctúen los resultados, siendo necesario la selección de un modelo transitorio. Además, también se ha podido observar fluctuaciones que van en aumento en el coeficiente de sustentación (Figura 9.3).



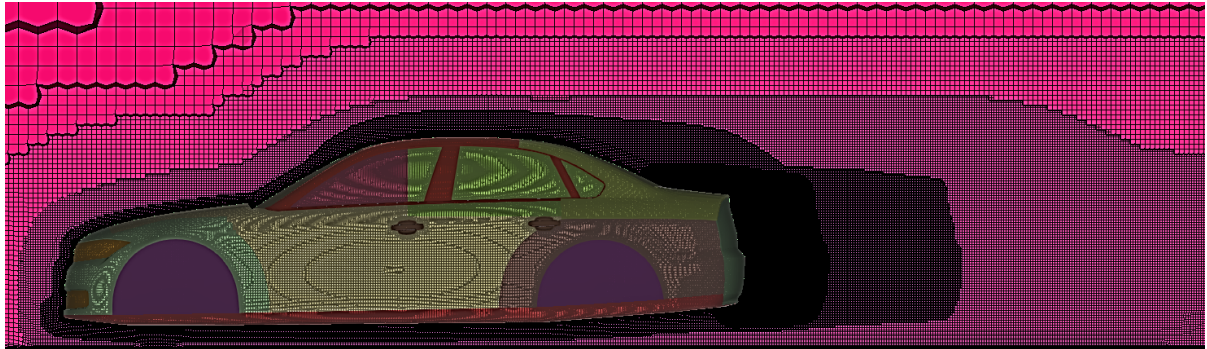
**Figura 7.16 Contorno de velocidad Modelo 2 en la iteración 4000, 4500 y 5000 - DrivAer sin ruedas**

Viendo estas fluctuaciones en la estela, se trata de refinar más la malla. A parte de refinar otras zonas, se presta una mayor atención a la estela, generando hasta dos nuevos volúmenes para un mayor control del tamaño.

**Tabla 5. Valores de malla modelo 3 - Modelo DrivAer sin ruedas.**

Zona	Offset (mm)	Extrusión (mm)	Tamaño malla superficial (mm)	Tamaño malla volumétrica (mm)
Todo el vehículo	30	0	8	8
Pilares y zona trasera lateral	150	1000	4	8
Underbody	100	800	8	8
Zona trasera (estela 1)	150	400	4	4
Zona trasera (estela 2)	150	1300	8	8
Todo el vehículo	200	3000	16	16
Todo el vehículo	200	8000	32	32
Size Box 1	-	-	128	128

Por tanto, se han creado varios volúmenes adicionales para la generación de la malla. En la tabla 5 se puede ver un resumen de los valores y volúmenes usados. Este nuevo set up fija el tamaño máximo de celda alrededor del coche a 8mm. Con estas modificaciones, el número de celdas se incrementa hasta 51 millones de celdas con una calidad que cumple con los requisitos de OpenFOAM.



**Figura 7.17 Malla Modelo 3 - DrivAer sin ruedas**

Con esta malla, y usando el mismo set up que en el caso anterior, se mejora la convergencia obtenida. Tras analizar los resultados de los coeficientes de sustentación y resistencia, se observa como estos siguen variando (Figura 9.5 y Figura 9.6). Sin embargo, vista la mejora, se decide hacer una iteración adicional con la malla y no seguir corriendo esta simulación.

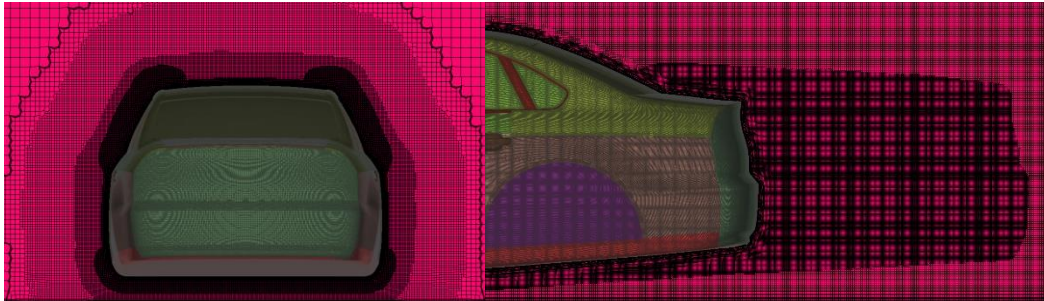
Finalmente, se realiza una malla similar al caso anterior con los parámetros mostrados en la tabla 6. En este caso, la misma caja de refinamiento (size box) usada para la malla anterior, se ha dividido en dos (Figura 9.7), separando la estela más lejana de los alrededores más cercanos del coche.

**Tabla 6. Valores de malla modelo 4 - Modelo DrivAer sin ruedas.**

Zona	Offset (mm)	Extrusión (mm)	Tamaño malla superficial (mm)	Tamaño malla volumétrica (mm)
Todo el vehículo	30	0	4	4
Pilares y zona trasera lateral	150	1000	4	4
Underbody	100	800	4	4
Zona trasera (estela 2)	150	1300	4	4
Todo el vehículo	200	3000	8	8
Todo el vehículo	200	7000	16	16
Size Box 1	-	-	64	64
Size Box 2			128	128

La malla obtenida finalmente consta de 134 millones de celdas con una calidad que entra dentro de los parámetros de OpenFOAM (Figura 9.8 y Figura 9.9). Esta malla es mejorable en

términos de eficiencia. Quizás no es necesario mallar con tanto detalle algunas zonas y es posible que alguna zona pueda necesitar de un refinamiento adicional. Sin embargo, este trabajo queda pendiente para seguir realizando en el futuro.



**Figura 7.18 Malla Modelo 4 - DrivAer sin ruedas**

Analizando los residuales (Figura 9.12, p84), vemos como los de la velocidad han bajado a valores de  $2 \cdot 10^{-4}$ , mientras que los de presión han bajado ligeramente a  $2 \cdot 10^{-3}$ . Se necesita seguir trabajando para mejorar estos valores, sobre todo los residuales de p.

De este modelo se han realizado ensayos en un túnel de viento con una maqueta a escala 1:2.5 por la universidad de Múnich (TUM) (Heft et al., 2012b; Mack et al., 2012; Miao et al., 2015). Las condiciones de ensayo fueron con  $Re$  de  $5.2 \cdot 10^6$  y con el suelo a la misma velocidad que el flujo. Los resultados obtenidos para el modelo Notchback fueron un  $C_D$  de 0.129 y un  $C_L$  de -0.37 (Mack et al., 2012). Aunque no son las mismas condiciones de ensayo, se puede usar esta referencia para tener un orden de magnitud de los valores. Al haber empezado con el modelo con ruedas, por practicidad, se decidió continuar con ese mismo set up. Esto nos ha llevado a no poder comparar directamente los valores de los coeficientes de drag obtenidos, pero si tener una idea del orden de magnitud de éstos.

Los valores de  $C_D$  y  $C_L$  obtenidos promediando las últimas 1000 iteraciones son de 0.143 y -0.442 respectivamente. Aunque es complicado comparar estos resultados como ya se ha expuesto, vemos como los valores obtenidos son del mismo orden que los vistos en los ensayos. De todas formas, si analizamos los gráficos de las últimas 2000 iteraciones de los coeficientes (Figura 9.10 y Figura 9.11, p83), vemos que los valores de  $C_D$  siguen decreciendo, 3 puntos en las últimas 1000 iteraciones. Los valores de  $C_L$  parecen haberse estabilizado sobre un valor promedio con oscilaciones de unos 10 puntos.

Al no conseguirse unos valores de convergencia suficiente, no se ha podido continuar cambiando a esquemas numéricos de mayor orden.

### 7.5.2. Modelo de DrivAer con ruedas

Con este modelo se han ido probando diferentes configuraciones de set up y malla. Según (James et al., 2018), el  $C_D$  es de 0.233 y el  $C_L$  es de -0.031. Estos valores deberán ser usados con cautela ya que se han modificado las ruedas. En la simulación, como se ha comentado, no se han usado llantas.

La configuración usada para la primera simulación, tras divergir los resultados con el modelo sin ruedas (Figura 7.13), se muestra en la siguiente tabla:

**Tabla 7. Esquemas numéricos simulación 1 - Modelo DrivAer con ruedas.**

Tipo	Variable	Esquema
gradScheme	default	cellLimited Gauss Linear 0.5
divScheme	U, p, k, $\omega$	bounded Gauss upwind
laplacianSchemes	default	Gauss linear limited 0.5
interpolationScheme	default	linear
snGradScheme	default	limited 0.5

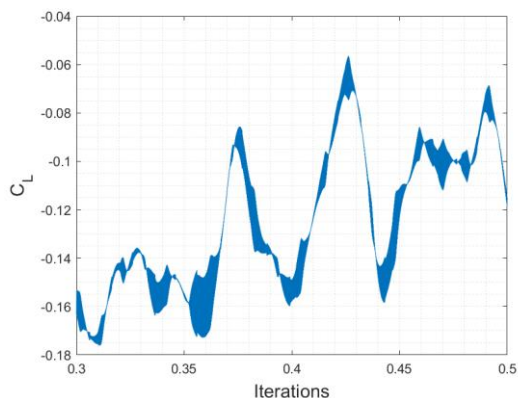
La primera simulación muestra valores de los residuales de  $U_y$  y  $U_z$  de  $5 \cdot 10^{-2}$ . Al considerarse unos valores insuficientes, se decide realizar diferentes actuaciones:

1. Se reducen las tolerancias relativas de los solvers lineales, para conseguir que el residuo final de las iteraciones internas decrezca más.
2. Se reducen los criterios de convergencia a  $10^{-10}$  para la velocidad, k y omega, y a  $10^{-8}$  para la p. Se ha leído en foros que openFOAM usa estos valores para parar de iterar algunas variables. Por tanto, si por ejemplo, se tiene un valor de los residuos de p elevado, y se necesita que decrezcan los de U para seguir bajando, pero estos ya han llegado al valor, esto nos limitará también los valores de p.
3. Su usan entre 1-2 correctores de non-orthogonalidad.
4. Para mejorar la robustez y la convergencia, se recomienda inicializar y definir un valor pequeño ( $\approx 10^{-4}$ ) para la viscosidad turbulenta. Por tanto, se modifica el fichero en la carpeta 0 de nut, introduciendo un valor de 0.0001.
5. Se divide el suelo en dos zonas para poder realizar una mejor malla en la zona central, donde se encuentra el vehículo.
6. Para asegurar una mayor robustez, se han reducido los factores de relajación de 0.5 para la presión y 0.7 para el resto, a 0.3 y 0.5 respetivamente. Los valores de 0.5 y 0.7 son bastante usados en la industria.

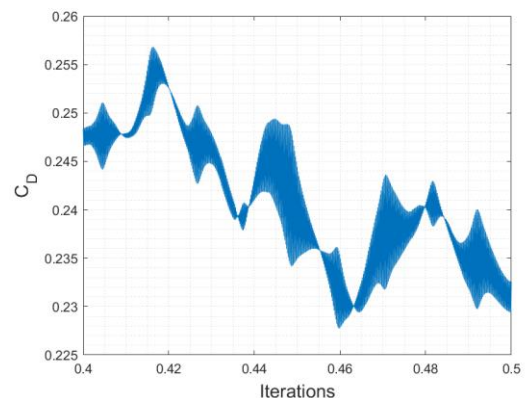
7. Se modifica el esquema del gradiente a ***cellLimited Gauss Linear 1***. De esta forma no permitimos ningún valor mayor que los extremos. Puede afectar a la precisión, pero ofrece una mayor robustez.
8. Se usan funciones de pared para reducir el número de celdas al implementar los nuevos volúmenes de control (similar a los últimos modelos sin ruedas).

Con estos cambios, e implementando una malla del mismo estilo que el modelo 3 sin ruedas (Figura 9.14, Figura 9.15, Figura 9.16 y Tabla 8 para ver especificaciones), se realiza la simulación. Se ha mejorado ligeramente los residuos obteniendo valores para  $U_y$  y  $U_z$  de  $10^{-3}$ . Sin embargo, no se ha conseguido mejorar los residuos para la presión, similar a los que se pudo ver en el modelo sin ruedas.

De todas formas, comparemos los resultados que se obtienen con los resultados experimentales. Vemos como los valores de  $C_L$  y  $C_D$  no han acabado de estabilizarse, pero nos ofrecen una idea del orden en el que oscilamos (Figura 7.19 y Figura 7.20).



**Figura 7.19** Gráfico  $C_L$  Modelo final – DrivAer con ruedas – últimas 2000 iteraciones



**Figura 7.20** Gráfico  $C_D$  Modelo final – DrivAer con ruedas – últimas 1000 iteraciones

En esta última simulación se ha fijado la velocidad en el suelo. Esto tendrá un efecto sobre los coeficientes de resistencia y sustentación. De todas formas, se comparan los resultados para tener una idea del orden de magnitud de los obtenidos.

Se usa el promedio en las últimas 1000 iteraciones para obtener un valor. Se obtiene un valor de  $C_D$  de 0.236 y un valor de  $C_L$  de -0.097, valores muy cercanos, sobre todo el  $C_D$ . Sin embargo, notar que ambos coeficientes tienen tendencias descendentes y ascendentes respectivamente (Figura 7.19 y Figura 7.20).

Si ahora comparamos el flujo en el plano central, la estela no se ha estabilizado y sigue oscilando, lo que explica las oscilaciones de los coeficientes. Esto nos hace pensar en dos posibles soluciones, o directamente pasar a un modelo transitorio, ya sea un modelo híbrido o URANS, o seguir trabajando en la malla, esquemas y *solvers* lineales para tratar de solucionar estas oscilaciones.

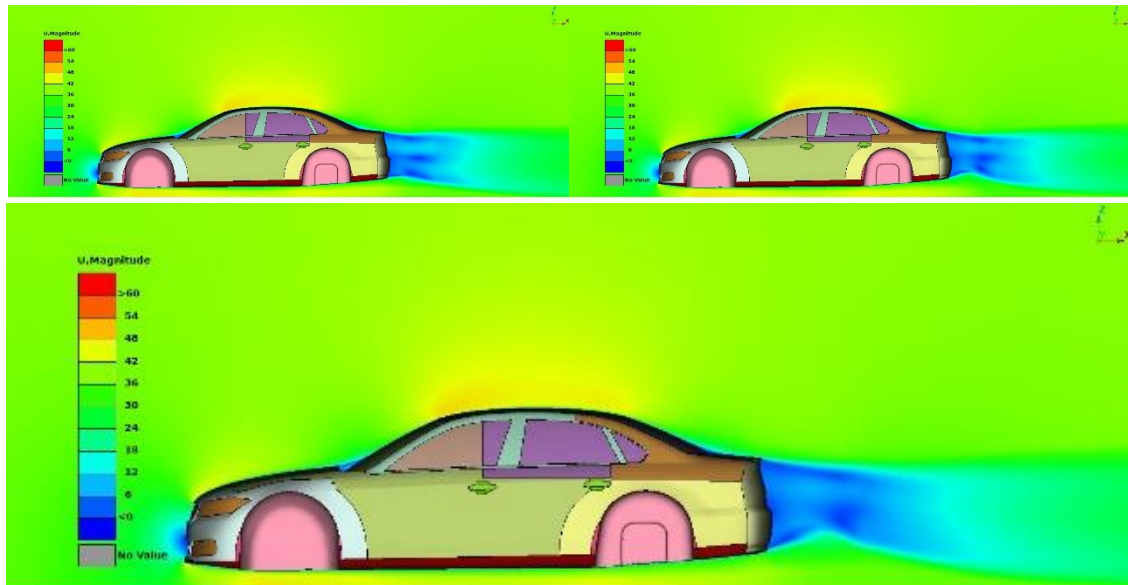


Figura 7.21 Contorno de velocidades Modelo Final (iteración 4000, 4500 y 5000) – DrivAer con ruedas

Al igual que se observó en los resultados recogidos en (Hupertz et al., 2021), se observa una ligera zona de desprendimiento al final de la luneta trasera (Figura 7.22). Esta zona es bastante complicada de predecir. Como muestra (Hupertz et al., 2021), hay una gran variabilidad en las predicciones en esta zona. Los resultados mostraron que, simulaciones con los mismos modelos para resolver la turbulencia (modelos híbridos, RANS...), mostraban diferencias en la severidad e, incluso, en la aparición de esta zona de desprendimiento.

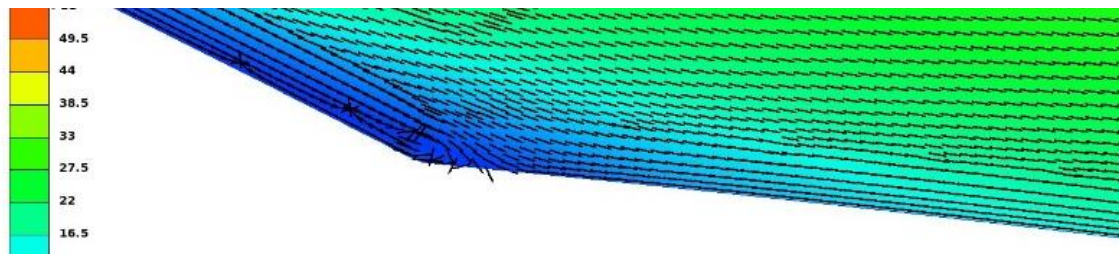
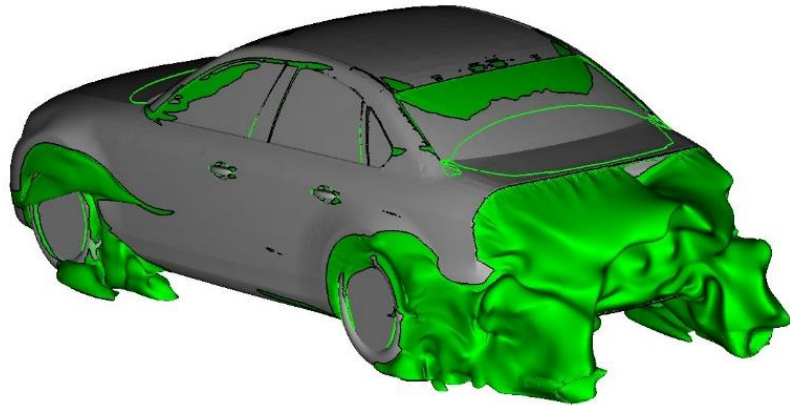


Figura 7.22 Contorno de velocidades Modelo Final (zoom final luneta trasera) – DrivAer con ruedas

Finalmente, se muestra la estela generada tanto por el cuerpo del vehículo, como en las ruedas. Se aprecia la estela generada por las ruedas y como se extiende hacia la zona posterior. Tras revisar esto, se considera que se debería aumentar la zona de refinamiento

creada para las ruedas, ya que se observan zonas donde una mayor resolución de la malla puede implicar una mejor resolución de los efectos turbulentos en esa zona. Este aumento también se debería aplicar a las ruedas posteriores, siendo necesario extruir este volumen de control una mayor distancia en X para cubrir la propia estela y la interacción de ésta con la del cuerpo del coche.



**Figura 7.23** Isosuperficie del coeficiente de presión total 0 ( $C_{pT}=0$ ) – DrivAer con ruedas

### 7.5.3. Modelo segunda conferencia AutoCFD

Como caso adicional, se decide simular la malla usada en el segundo Workshop de AutoCFD (University of Oxford, 2021). Este modelo es el modelo de NotchBack de DrivAer con el suelo detallado y con retrovisores. Esta malla se ha descargado de la página del evento, la cual está disponible en formato OpenFOAM.

Debido al tamaño de las mallas, se decide usar la malla para Reynolds altos (*high Reynolds mesh*), donde necesitaremos definir funciones de pared. Esta malla tiene alrededor de 135 millones de celdas.

El set up usado es similar a los casos anteriores. Las condiciones de contorno en el asfalto vienen dadas por el organizador para tener un correcto perfil de velocidades en la zona de ensayo. Se fija la velocidad en la entrada, la presión a la salida (0 Pa) y el resto de contornos se marcan como *symmetry*.

Todos los resultados presentados en este estudio se resumen en las siguientes figuras con los coeficientes de resistencia y sustentación (Hupertz et al., 2021).

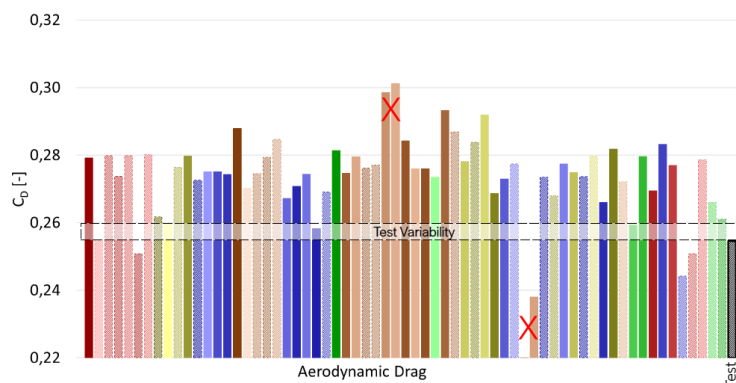


Figura 7.24 Resumen resultados Workshop AutoCFD -  $C_D$  (Hupertz et al., 2021)

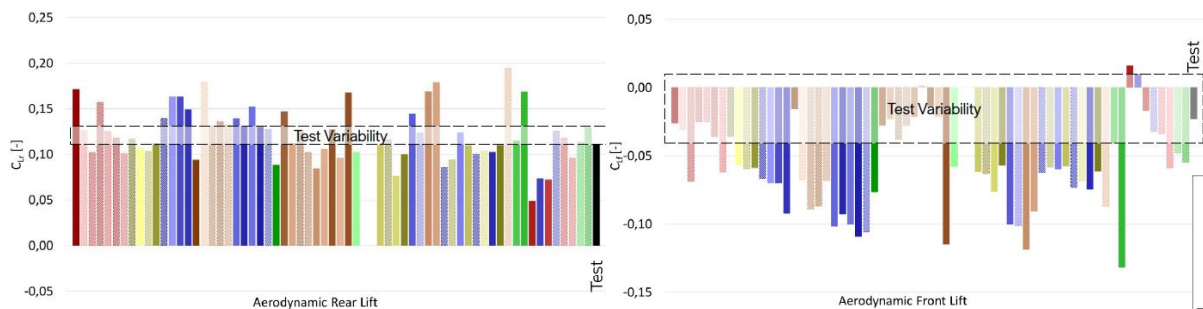


Figura 7.25 Resumen resultados Workshop AutoCFD -  $C_{Lr}$  y  $C_{Lf}$  (Hupertz et al., 2021)

Para resolver esta simulación, se han fijado primero los esquemas como en el último caso del modelo de DrivAer sin ruedas (7.5.1). Tras correr 10000 iteraciones, se cambian los esquemas de divergencia a *linearUpwindV* para la velocidad y a *vanLeer* para  $k$  y  $\omega$ , y se corre durante 2500 iteraciones más.

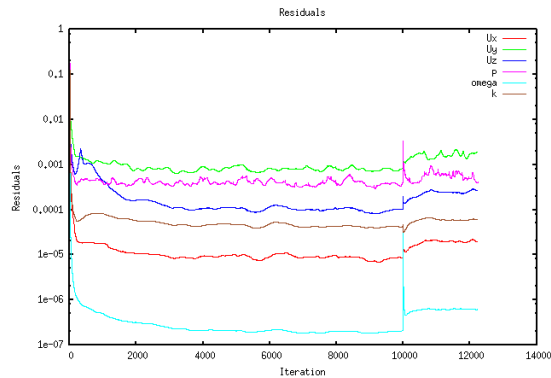


Figura 7.26 Historial de convergencia modelo AutoCFD

Al finalizar, si analizamos los valores obtenidos de coeficiente de resistencia y de sustentación (Figura 7.27), vemos como se obtienen valores de  $C_D$  de 0.270,  $C_{L_f}$  de -0.045 y de  $C_{L_r}$  de 0.150. Notar que los resultados aún no se han estabilizado y siguen oscilando. Aún así, vemos como los valores obtenidos se encuentran dentro de la dispersión propia que se observa en la Figura 7.24 y Figura 7.25.

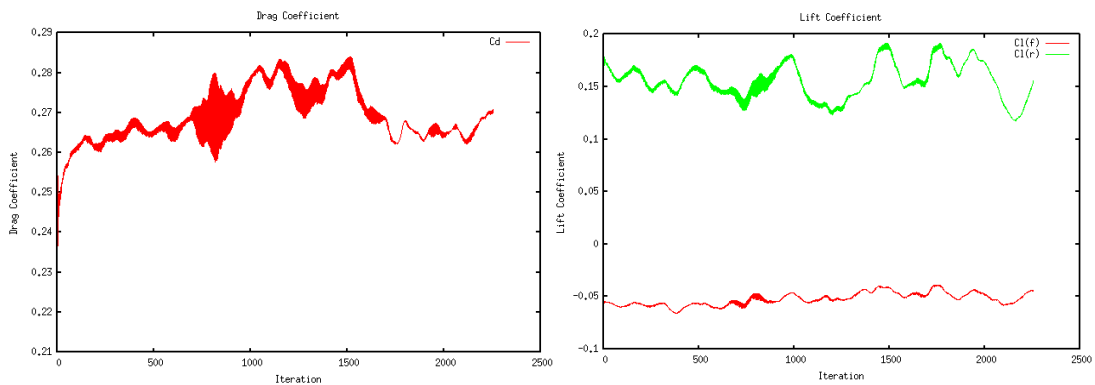


Figura 7.27 Valor de los coeficientes últimas 2300 iteraciones

## 8. Conclusiones y Trabajo futuro

No se ha conseguido el objetivo final de obtener el proceso completo de simulación, pero, con este trabajo, se ha dado un primer paso para la introducción de este software de código libre, como una herramienta más de simulación dentro de la empresa. Esto, dado el incremento de carga de trabajo que hay en el sector, puede implicar ahorros de entre 60000 y 100000€ anuales en la compra de nuevas licencias de softwares comerciales.

Sin embargo, sí que se han cumplido objetivos intermedios. Se ha aprendido a usar el mallador de ANSA generando mallas de buena calidad con el mallador HeXtreme. Además, se ha obtenido un mayor conocimiento de OpenFOAM, lo que facilitará los próximos pasos.

Aún teniendo trabajo por mejorar, se han obtenido resultados del mismo orden que los resultados experimentales. Se ha visto como, con el set up definido en este TFM, se obtienen resultados de  $C_D$  y  $C_L$  del mismo orden que los presentados en el estudio de benchmark. Esto parece indicar que se ha obtenido un set up robusto, aunque queda trabajo en cuanto a la definición de la malla. Todo parece indicar que, con OpenFOAM, se necesita una malla más fina para obtener buenos resultados, y esto nos lleva al siguiente punto.

Durante el desarrollo de este estudio se ha planteado comparar los resultados y la robustez de las simulaciones entre OpenFOAM y softwares comerciales. Se ha preparado un modelo con la última malla realizada con ruedas, y en cuanto se tenga disponibilidad en el cluster, se realizará la simulación, en la que también se comparará el tiempo para realizar un número determinado de iteraciones. Además, se añadirán más simulaciones en el transcurso de futuros estudios. De esta forma, se podrá validar lo expuesto en el punto anterior.

En cuanto al trabajo con OpenFOAM, se debe continuar con el estudio. Primero se debe optimizar la última malla en el caso sin ruedas y adaptarla al caso con ruedas. Se propone continuar con el refinado de la malla en la zona del neumático (zona contacto rueda-asfalto y el paso de rueda). Se sugiere centrarse en esta zona ya que, como se ha visto, al incluir las ruedas en la simulación, la convergencia ha empeorado.

Finalmente, se propone introducir un modelo realista de vehículo. Se habla de un modelo más realista ya que, normalmente, las geometrías que se tienen son de modelos en fase de diseño, y las superficies no suelen ser tan limpias como el modelo usado en este trabajo.

## REFERENCIAS BIBLIOGRÁFICAS

- Agarwal, R. K. (2013). INTERNATIONAL JOURNAL OF ENERGY AND ENVIRONMENT Sustainable ground transportation-review of technologies, challenges and opportunities. In *Journal homepage: www.IJEE.IEEFoundation.org ISSN* (Vol. 4, Issue 6). Online. [www.IJEE.IEEFoundation.org](http://www.IJEE.IEEFoundation.org)
- Anderson, J. (2017). *Fundamentals of Aerodynamics* (Sixth). Mc Graw Hill Education.
- ANSYS. (2021). *ANSYS Fluent Theory Guide*. <http://www.ansys.com>
- Ashton, N., & Revell, A. (2015). Comparison of RANS and DES methods for the DrivAer automotive body. *SAE Technical Papers, 2015-April(April)*. <https://doi.org/10.4271/2015-01-1538>
- Ashton, N., West, A., Lardeau, S., & Revell, A. (2016). Assessment of RANS and DES methods for realistic automotive models. *Computers and Fluids, 128*, 1–15. <https://doi.org/10.1016/j.compfluid.2016.01.008>
- Aultman, M., Wang, Z., & Duan, L. (2021). Effect of time-step size on flow around generic car models. *Journal of Wind Engineering and Industrial Aerodynamics, 219*. <https://doi.org/10.1016/j.jweia.2021.104764>
- Bastian Nebenführ. (2010). *OpenFOAM: A tool for predicting automotive relevant flow fields* [CHALMERS UNIVERSITY OF TECHNOLOGY]. [http://www.tfd.chalmers.se/~lada/postscript\\_files/Bastian-Nebenfuhr-OpenFOAM\\_A\\_tool\\_for\\_predicting\\_automotive\\_flow\\_fields.pdf](http://www.tfd.chalmers.se/~lada/postscript_files/Bastian-Nebenfuhr-OpenFOAM_A_tool_for_predicting_automotive_flow_fields.pdf)
- BETA CAE. (2022). *ANSA v22.0.1 ANSA for CFD Brief User's Guide*. <http://www.beta-cae.com>
- Bezavada, S., & Kodali, S. P. (2012). Numerical simulation of air flow over a passenger car and the Influence of rear spoiler using CFD. In *International Journal of Advanced Transport Phenomena* (Vol. 01, Issue 01). <https://www.researchgate.net/publication/336589740>
- Christopher Rumsey. (2021, November 8). *The Menter Shear Stress Transport Turbulence Model*. NASA - Langley Research Center. <https://turbmodels.larc.nasa.gov/sst.html>

- Collin, C., Mack, S., Indinger, T., & Mueller, J. (2016). A Numerical and Experimental Evaluation of Open Jet Wind Tunnel Interferences using the DrivAer Reference Model. *SAE International Journal of Passenger Cars - Mechanical Systems*, 9(2), 657–679. <https://doi.org/10.4271/2016-01-1597>
- Ekman, P., Wieser, D., Virdung, T., & Karlsson, M. (2020). Assessment of hybrid RANS-LES methods for accurate automotive aerodynamic simulations. *Journal of Wind Engineering and Industrial Aerodynamics*, 206. <https://doi.org/10.1016/j.jweia.2020.104301>
- Ferziger, J. H., Perić, M., & Street, R. L. (2002). *Computational Methods for Fluid Dynamics* (4th ed.). Springer. <https://doi.org/10.1007/978-3-319-99693-6>
- Frank M. White. (2006). *Viscous Fluid Flow* (3th ed.). McGraw-Hil.
- Frank M. White. (2008). *Fluid Mechanics* (6th ed.). McGraw-Hill Companies, Inc.
- Greenshields, C. (2011). *OpenFOAM The OpenFOAM Foundation User Guide*. <https://openfoam.org>
- Heft, A. I., Indinger, T., & Adams, N. A. (2012a). *Experimental and Numerical Investigation of the DrivAer Model*. <http://www.aer.mw.tum.de/en/research-groups/automotive/drivaer>
- Heft, A. I., Indinger, T., & Adams, N. A. (2012b). EXPERIMENTAL AND NUMERICAL INVESTIGATION OF THE DRIVAER MODEL. *ASME 2012 Fluids Engineering Summer Meeting*. <http://www.aer.mw.tum.de/en/research-groups/automotive/drivaer>
- Hupertz, B., Kuthada, T., Ashton, N., Duan, L., & Lewington, N. (2021). *AutoCFD2 Case2 - Results Overview*. <https://autocfd.eng.ox.ac.uk/#2nd-workshop>
- Hvroje Jasak. (2016). *Finite Volume Discretisation in OpenFOAM. Best Practice Guidelines*.
- Islam, A., & Thornber, B. (2017). High-order detached-eddy simulation of external aerodynamics over an SAE notchback model. *Aeronautical Journal*, 121(1243), 1342–1367. <https://doi.org/10.1017/aer.2017.61>
- James, T., Krueger, L., Lentzen, M., Woodiga, S., Chalupa, K., Hupertz, B., & Lewington, N. (2018). Development and Initial Testing of a Full-Scale DrivAer Generic Realistic Wind

- Tunnel Correlation and Calibration Model. *SAE International Journal of Passenger Cars - Mechanical Systems*, 11(5), 353–367. <https://doi.org/10.4271/2018-01-0731>
- Lanuza, J. (2016). *Investigation into the Effect of Vortex Generators on a Car Diffuser Performance*. Cranfield University.
- Larry Deaton, Mohan Rao, & Wei-Zen Shih. (2018). *Root Cause Identification and Methods of Reducing Rear Window Buffeting Noise*.
- Leschziner, M. (2015). *Statistical turbulence modelling for fluid dynamics, demystified : an introductory text for graduate engineering students*.
- Lin, J. C. (1992). *CONTROL OF LOW-SPEED TURBULENT SEPARATED FLOW OVER A BACKWARD-FACING RAMP - NASA-TM-109740*.
- Mack, S., Indinger, T., Adams, N. A., Blume, S., & Unterlechner, P. (2012). THE INTERIOR DESIGN OF A 40% SCALED DRIVAER BODY AND FIRST EXPERIMENTAL RESULTS. *ASME 2012 Fluids Engineering Division SummerMeeting*. <http://www.asme.org/about-asme/terms-of-use>
- Menter, F. R., Kuntz, M., & Langtry, R. (2003). Ten Years of Industrial Experience with the SST Turbulence Model. *Proceedings of the Fourth International Symposium on Turbulence, Heat and Mass Transfer*, 625–632. <http://aac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/>
- Miao, L., Mack, S., & Indinger, T. (2015). EXPERIMENTAL AND NUMERICAL INVESTIGATION OF AUTOMOTIVE AERODYNAMICS USING DRIVAER MODEL. *ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. <http://www.asme.org/about-asme/terms-of-use>
- Nakamura, Y., Nakashima, T., Hiraoka, T., Shimizu, K., Nouzawa, T., Doi, Y., & Mutsuda, H. (2020). Identification of the vortex around a vehicle by considering the pressure minimum. *Journal of Visualization*, 23(5), 793–804. <https://doi.org/10.1007/s12650-020-00665-8>
- OpenCFD Ltd. (2022). *OpenFOAM: User Guide v2112*. <https://www.openfoam.com/documentation/guides/latest/doc/>

- Osborne Reynolds. (1895). On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion. *Philosophical Transactions of the Royal Society of London, 186*, 123–164. <https://doi.org/10.1098/rsta.1895.0004>
- Oweis, G. F., Winkel, E. S., Cutbrith, J. M., Ceccio, S. L., Perlin, M., & Dowling, D. R. (2010). The mean velocity profile of a smooth-flat-plate turbulent boundary layer at high Reynolds number. *Journal of Fluid Mechanics, 665*, 357–381. <https://doi.org/10.1017/S0022112010003952>
- Pope, S. B. (2000). *Turbulent Flows*. Cambridge University Press.
- Siemens. (2020). *External Aerodynamics with Simcenter STAR-CCM+. Best Practices Guidelines (2020.2)*.
- Skaperdas, V. (2021). *Committee grids for Case 2 2 nd Automotive CFD workshop*. [www.beta-cae.com](http://www.beta-cae.com)
- Soares, R. F., Garry, K. P., & Holt, J. (2017). Comparison of the Far-Field Aerodynamic Wake Development for Three DrivAer Model Configurations using a Cost-Effective RANS Simulation. *SAE Technical Papers, 01(1514)*. <https://doi.org/10.4271/2017-01-1514>
- Soares, R. F., & José De Souza, F. (2018). *Influence of CFD Setup and Brief Analysis of Flow Over a 3D Realistic Car Model*.
- Soares, R., Souza, F., Soares, R. F., & de Souza, F. J. (2015). Influence of numerics and analysis of external flow over a three-dimensional realistic car model. *SAE Technical Paper*. <https://doi.org/10.13140/RG.2.1.1068.0408>
- Spalart, P. R. (2001). *Young-Person's Guide Simulation Grids to Detached-Eddy*. <http://www.sti.nasa.gov>
- TUM School of Engineering and Design. (n.d.). *DrivAer Model*. Retrieved May 14, 2022, from <https://www.epc.ed.tum.de/en/aer/research-groups/automotive/drivaer/>
- University of Oxford. (2021). *Automotive CFD Prediction Workshop (AutoCFD)*. <https://autocfd.eng.ox.ac.uk/>
- Varney, M., Passmore, M., Wittmeier, F., & Kuthada, T. (2020). Experimental data for the validation of numerical methods: DrivAer model. *Fluids, 5(4)*. <https://doi.org/10.3390/fluids5040236>

Versteeg, H. K., & Malalasekera, W. (1995). *An Introduction to Computational Fluid Dynamics Second Edition*. [www.pearsoned.co.uk/versteeg](http://www.pearsoned.co.uk/versteeg)

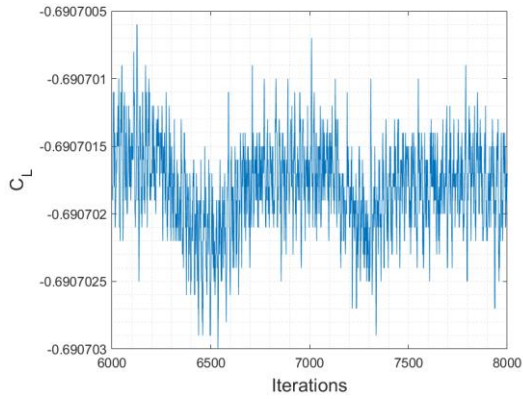
Wilcox, D. C. (2006). *Turbulence modeling for CFD*. DCW Industries.

Wolf Dynamics. (n.d.). *Supplement Tips and tricks*. Retrieved June 26, 2022, from [http://www.wolfdynamics.com/training/introOF8/supplement\\_tipsandtricks.pdf](http://www.wolfdynamics.com/training/introOF8/supplement_tipsandtricks.pdf)

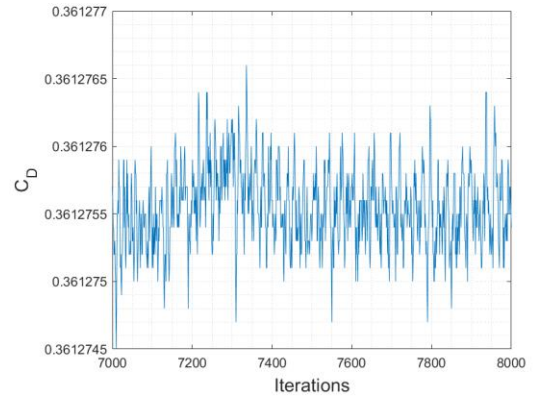
Zhang, X., Toet, W., & Zerihan, J. (2006). Ground effect aerodynamics of race cars. In *Applied Mechanics Reviews* (Vol. 59, Issues 1–6, pp. 33–48). <https://doi.org/10.1115/1.2110263>

## 9. ANEXOS

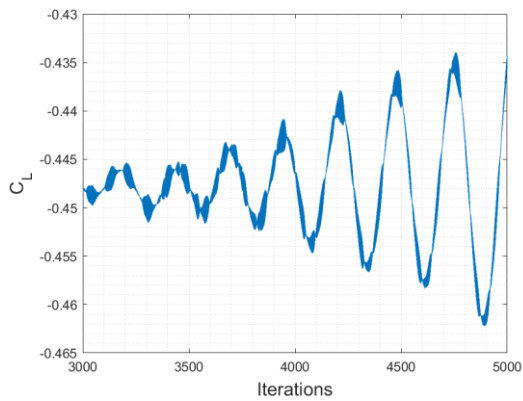
### Anexo A. Figuras adicionales



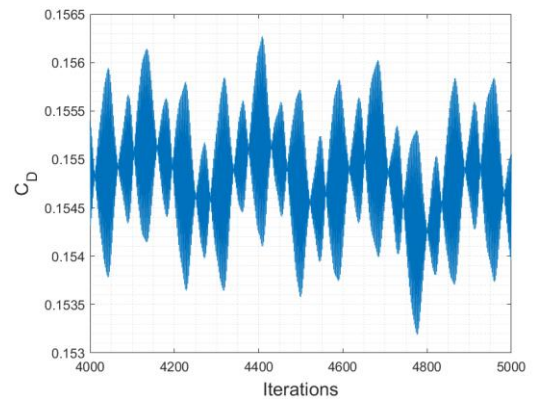
**Figura 9.1 Gráfico  $C_L$  Modelo Simplificado – Difusor  
4º y RH=40mm – últimas 2000 iteraciones**



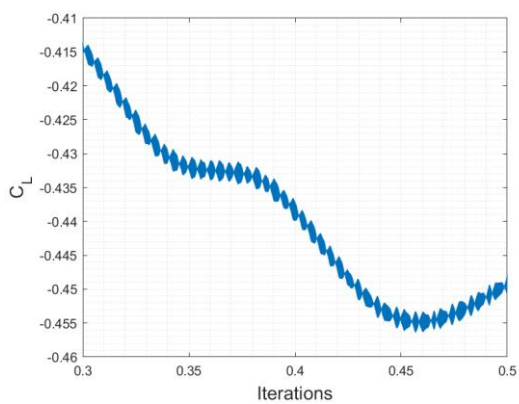
**Figura 9.2 Gráfico  $C_D$  Modelo Simplificado – Difusor  
4º y RH=40mm – últimas 1000 iteraciones**



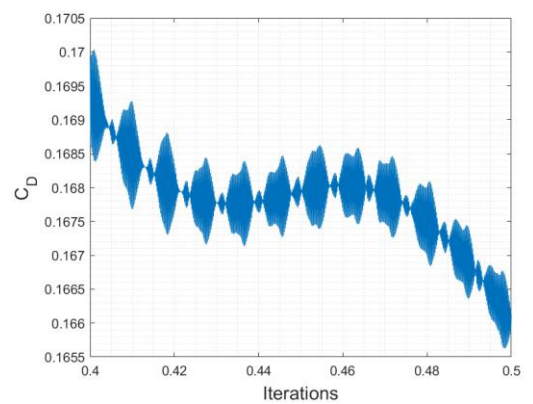
**Figura 9.3 Gráfico  $C_L$  Modelo 2 – DrivAer sin ruedas –  
últimas 2000 iteraciones**



**Figura 9.4 Gráfico  $C_D$  Modelo 2 – DrivAer sin ruedas –  
últimas 1000 iteraciones**



**Figura 9.5 Gráfico  $C_L$  Modelo 3 – DrivAer sin ruedas –  
últimas 2000 iteraciones**



**Figura 9.6 Gráfico  $C_D$  Modelo 3 – DrivAer sin ruedas –  
últimas 1000 iteraciones**

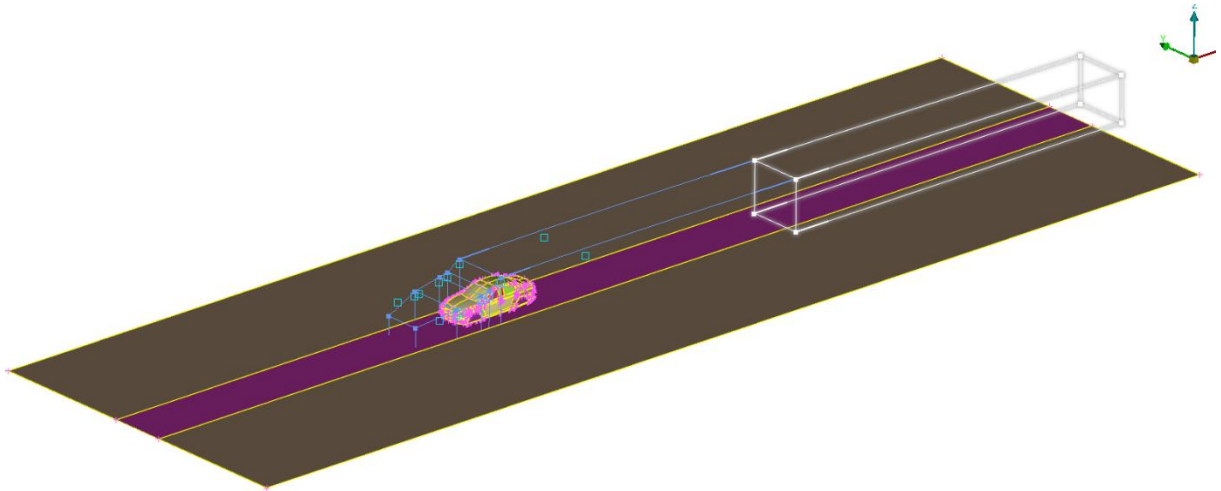


Figura 9.7 Detalle caja de refinamiento – Resultado Size box 2

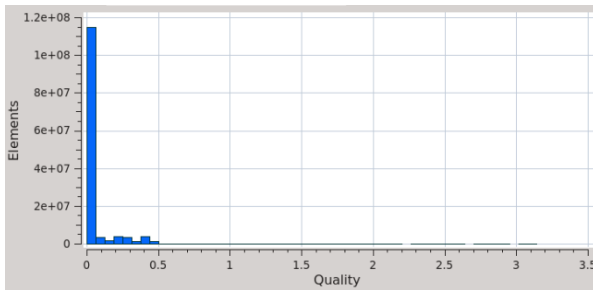


Figura 9.8 Skewness Modelo 4 – Modelo DrivAer sin  
ruedas

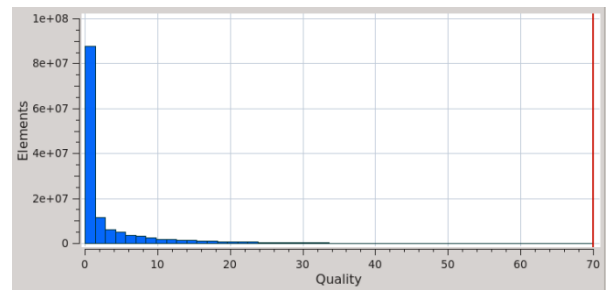


Figura 9.9 Non-orthogonality Modelo 4 – Modelo  
DrivAer sin ruedas

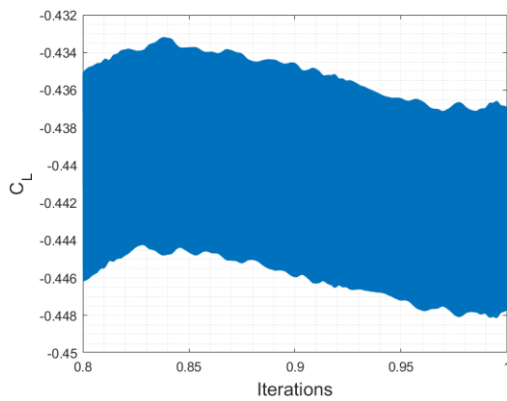


Figura 9.10 Gráfico  $C_L$  Modelo 4 – DrivAer sin ruedas  
– últimas 2000 iteraciones

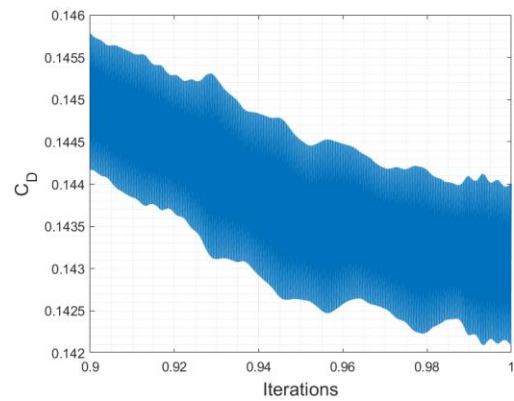
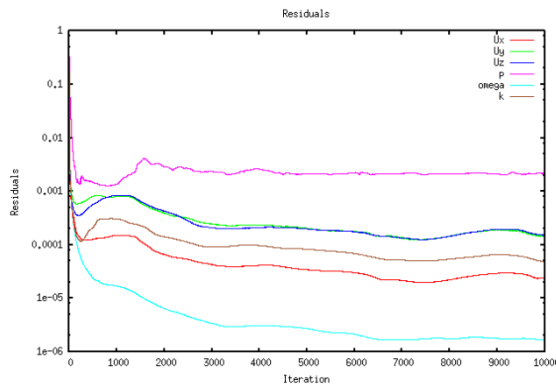
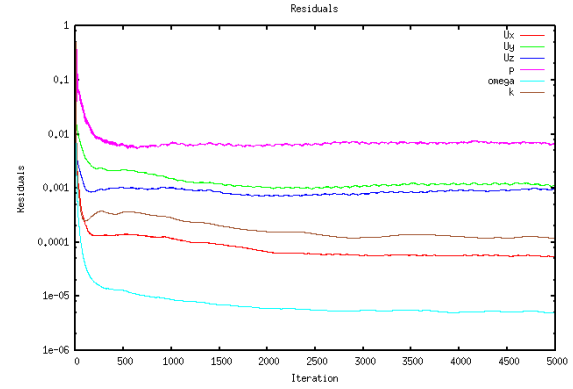


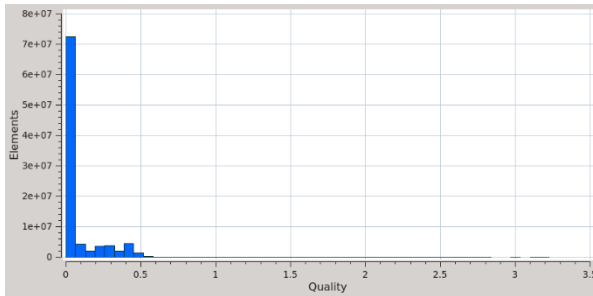
Figura 9.11 Gráfico  $C_D$  Modelo 4 – DrivAer sin ruedas  
– últimas 1000 iteraciones



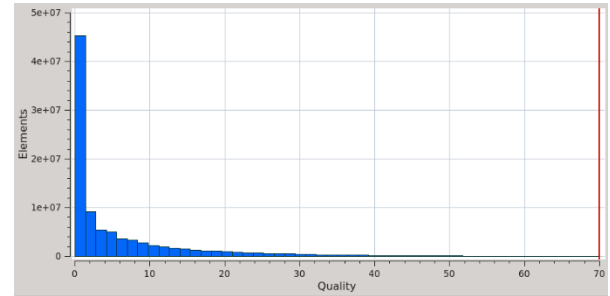
**Figura 9.12** Historial de convergencia modelo 4 –  
 DrivAer sin ruedas



**Figura 9.13** Historial de convergencia modelo final –  
 DrivAer con ruedas



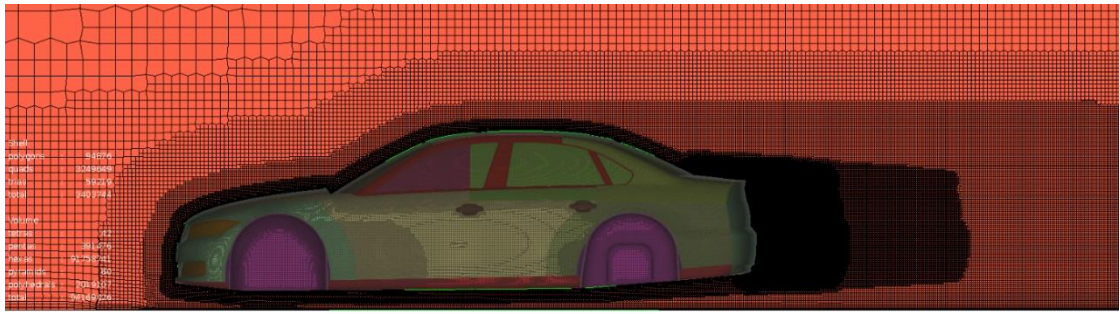
**Figura 9.14** Skewness modelo final – DrivAer con  
 ruedas



**Figura 9.15** Non-orthogonality modelo final – DrivAer  
 con ruedas

**Tabla 8.** Valores de malla modelo final - DrivAer con ruedas.

Zona	Offset (mm)	Extrusión (mm)	Tamaño malla superficial (mm)	Tamaño malla volumétrica (mm)
Pilares y zona trasera lateral	150	1000	4	4
Paso rueda (solventar mala calidad malla)	12	0	2	4
Estela ruedas	70	800	4	4
Contacto rueda-asfalto	30	60	0.25 - 1	0.25 - 1
Underbody	100	800	8	8
Zona trasera (estela 1)	150	600	4	4
Zona trasera (estela 2)	150	1600	8	8
Todo el vehículo	80	0	8	8
Todo el vehículo	200	4000	16	16
Todo el vehículo	200	7500	32	32
Size Box 1	-	-	64	64
Size Box 2			128	128



**Figura 9.16 Sección malla Y0 modelo final – DrivAer con ruedas**

## Anexo B. Información Adicional

### a. Estructura carpetas OpenFOAM

Repetimos la estructura para facilitar la siguiente explicación.

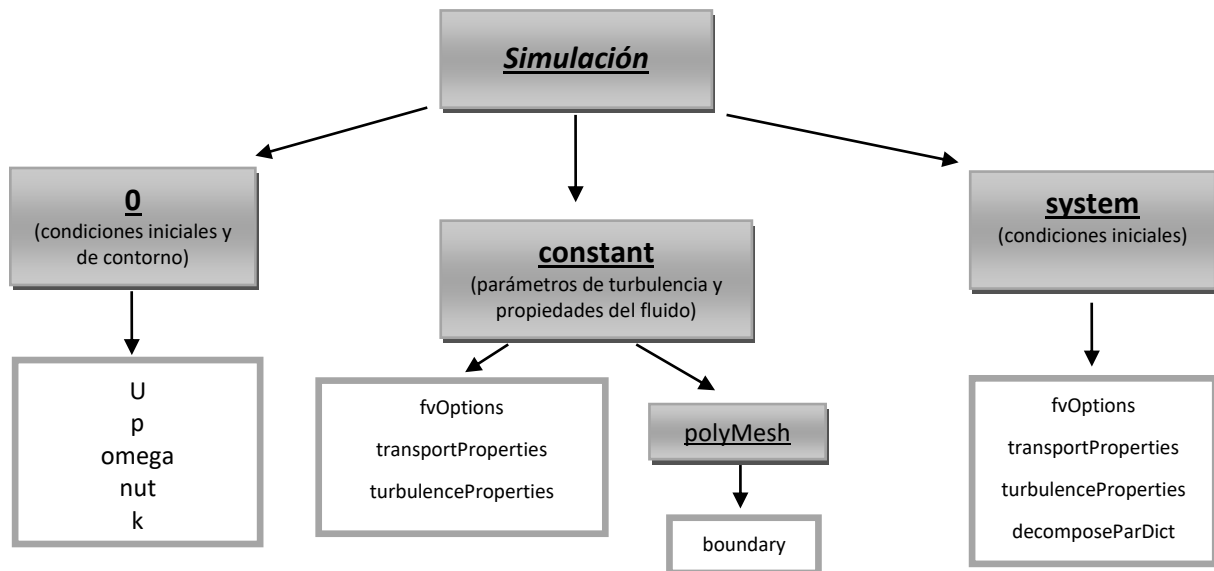


Figura 9.17 Estructura de carpetas en OpenFOAM (inspirado en (Bastian Nebenführ, 2010), Fig 2.6 – p16)

En la carpeta *0* se deben indicar las condiciones de contorno y las iniciales para cada variable (dependiendo del modelo, pueden variar las variables a introducir) en cada uno de los contornos. Si analizamos el fichero de U (Anexo C.a), vemos que hay tres campos principales a entrar:

1. *dimensions*: vemos que se introducen en un vector. Dependiendo de la posición, corresponde a un tipo u otro.

Tabla 9. Unidades ficheros condiciones iniciales (Ej. U)

[	0	1	-1	0	0	0	0]
Masa (kg)	Longitud (m)	Tiempo (s)	Temperatura (K)	(kgmol)	Corriente (A)	Intensidad lumínica (cd)	

2. *internalField*: Valor para inicializar el interior. En el caso de ser un vector, como por ejemplo U, hay que introducir las tres componentes. Si fuera un escalar, solo el valor de éste.

3. *boundaryField*: en este apartado, hay que introducir las condiciones de contorno (y sus valores iniciales si es necesario) para cada uno de los *patches* (contornos) definidos en la geometría/malla.

En la carpeta *constant*, se especifican los parámetros de turbulencia y las propiedades del fluido. En *turbulenceProperties* podemos elegir el modelo de turbulencia, en nuestro caso el *komegaSST*. En el fichero *transportProperties* podemos cambiar el valor de la viscosidad y, si se necesita para el tipo de simulación, introducir la densidad del fluido. Finalmente, en nuestro caso, se ha añadido un fichero opcional *fvOptions*. En este fichero el usuario puede especificar, como en nuestro caso, una restricción. En nuestro caso, limitamos la velocidad con un valor máximo. En el caso de ser mayor, lo limita al valor especificado en este fichero. Es importante definir este valor con cuidado para asegurarse que no afecta a los resultados, solo ayuda a la convergencia. Al acabar la simulación, se debe verificar que el valor máximo está por debajo de este valor. En la subcarpeta de *polyMesh*, en el fichero *boundary*, se definen el tipo de los contornos (*patches*, *wall* o *symmetry* en nuestro caso).

En *system* se introducen los parámetros de la simulación. En el fichero *controlDict*, podemos modificar el número de iteraciones, el paso de tiempo, el intervalo con el que guarda resultados y otros parámetros. En el fichero *decomposeParDict*, se especifica el tipo de partición que se hará y el número de procesadores que se usarán para resolver la solución. En *fvSchemes* se especifican los diferentes esquemas de discretización. En *fvSolution* podemos elegir con que *solver* lineal se resolverán las ecuaciones, correctores de *non-orthogonality* y factores de relajación.

## b. Ampliación esquemas numéricos usados

En este apartado se hace una introducción teórica de los esquemas numéricos usados en este trabajo final de máster.

### i. gradSchemes

Primero empezaremos por analizar como discretizaremos el gradiente de una variable:

$$\nabla \mathbf{u} = \frac{\partial u_i}{\partial x_i} \quad \text{Ecuación 9-1}$$

El esquema más comúnmente usado es un esquema Gauss linear. Asumamos un modelo 2D con la siguiente discretización de malla:

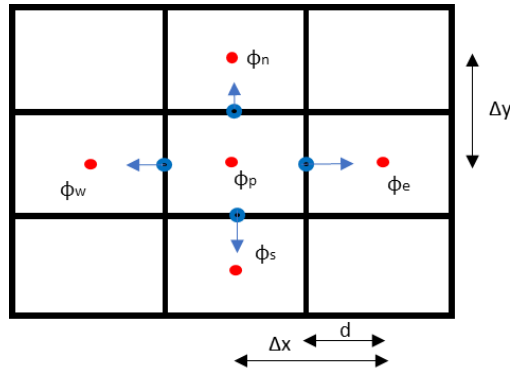


Figura 9.18 Discretizado 2D – Malla estructurada y alineada con el flujo

Para calcular el gradiente en el centro de la celda, lo que haremos será:

$$\nabla \phi_p = \frac{1}{V_p} \sum_n \phi_f \hat{\mathbf{n}}_f A_f \quad \text{Ecuación 9-2}$$

Siendo  $n$  el número de caras (en nuestro caso 4),  $\phi_f$  el valor de la variable en el centro de la cara,  $A_f$  el área de esa cara,  $\hat{\mathbf{n}}_f$  el vector normal en el centro de la cara y  $V_p$  el volumen de la celda.

La pregunta ahora es cómo se calcula el valor de la variable en el centro de la cara (*interpolationScheme*). Para calcularlo podemos usar una interpolación lineal de la forma:

$$\phi_f = 0.5(\phi_p + \phi_d) \quad \text{Ecuación 9-3}$$

Otro método muy usado es el esquema *leastSquares*. Con este método no es necesario calcular el valor en el centro de las caras, aunque trae asociado un error, el cual trata de minimizarse. Supongamos que tenemos una malla no estructurada como en la Figura 9.19. Este método parte de la Ecuación 9-4, en la que el valor de la variable en el punto i es igual al valor en p más el gradiente por el vector entre p e i.

$$\phi_i = \phi_p + \mathbf{d}_{pi} \cdot (\nabla\phi)_p \quad \text{Ecuación 9-4}$$

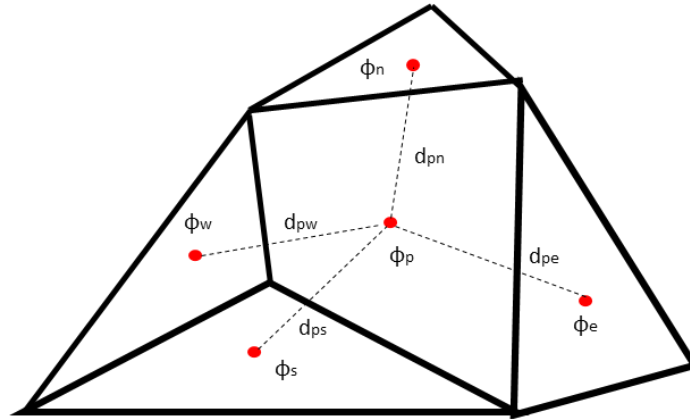


Figura 9.19 Discretizado 2D – Malla no estructurada 2D

En esta ecuación, se conoce el valor de la variable en todos los centros de las celdas i también conocemos el vector que une estos centros. Por tanto, la incógnita es el gradiente  $(\nabla\phi)_p$ . Si reescribimos la ecuación, nos queda que:

$$\mathbf{d}_{pi} \cdot (\nabla\phi)_p = \phi_i - \phi_p \quad \text{Ecuación 9-5}$$

Si aplicamos esta ecuación para nuestro ejemplo, pero en una configuración 3D, nos quedaría la siguiente matriz:

$$\begin{bmatrix} d_{pn_x} & d_{pn_y} & d_{pn_z} \\ d_{ps_x} & d_{ps_y} & d_{ps_z} \\ d_{pe_x} & d_{pe_y} & d_{pe_z} \\ d_{pw_x} & d_{pw_y} & d_{pw_z} \end{bmatrix} \begin{bmatrix} \left(\frac{\partial\phi}{\partial x}\right)_p \\ \left(\frac{\partial\phi}{\partial y}\right)_p \\ \left(\frac{\partial\phi}{\partial z}\right)_p \end{bmatrix} = \begin{bmatrix} \phi_n - \phi_p \\ \phi_s - \phi_p \\ \phi_e - \phi_p \\ \phi_w - \phi_p \end{bmatrix} \quad \text{Ecuación 9-6}$$

Esto nos acaba generando un sistema matricial de la forma  $\mathbf{Ax}=\mathbf{B}$ . Para resolverlo se plantea lo siguiente:

$$(\nabla\phi)_p = (\mathbf{d}_{pi}^T \mathbf{d}_{pi})^{-1} \mathbf{d}_{pi}^T (\phi_i - \phi_p) \quad \text{Ecuación 9-7}$$

Al ser la malla conocida y fija, excepto para casos con “*sliding mesh*”, el término  $(\mathbf{d}_{pi}^T \mathbf{d}_{pi})^{-1} \mathbf{d}_{pi}^T$  solo deberá calcularse una vez, al principio de la simulación para cada celda (Versteeg & Malalasekera, 1995). Este método puede ser poco preciso en celdas muy alargadas, como por ejemplo en la capa límite, ya que da más peso a las componentes donde la distancia es más larga (Versteeg & Malalasekera, 1995). Esta ponderación va en contra de lo que realmente pasa en la capa límite, donde los gradientes en la dirección normal a la pared, teniendo la distancia entre centro de celdas menor, son mayores que en la dirección tangencial a la pared. Este problema se puede solventar añadiendo una función de ponderación.

Éstos son los métodos que se han utilizado para calcular el gradiente en el centro de las celdas. Para incrementar la estabilidad de los esquemas, existen limitadores que limitan la solución obtenida dependiendo del mínimo y máximo en los alrededores. El problema de introducir estos limitadores es que reducen la precisión de los resultados. En este trabajo, además de simular sin limitador, se han usado los limitadores *cellLimited* y *cellMDLimited*, siendo este último menos difusivo (Wolf Dynamics, n.d.). Estos limitadores llevan un coeficiente al final de su definición ( $\psi$ ), siendo 1 cuando se limita completamente, 0 cuando no se limita y cualquier valor entre medias implica una ponderación teniendo en cuenta ambos valores de celda (OpenCFD Ltd., 2022).

## ii. divSchemes

Se han utilizado varios esquemas, tanto de primer como de mayor orden, siempre usando esquemas que tengan en cuenta la dirección del flujo. La forma genérica que toman es:

$$\text{div}(\phi, U) \quad \mathbf{Gauss} <\text{interpolation scheme}>$$

Por tanto, debemos indicar el esquema de interpolación. Se ha usado un esquema de interpolación *upwind* de primer orden para obtener un primer resultado y alimentar, como condición inicial, a la simulación con esquemas de mayor orden. Los esquemas que se han usado se pueden generalizar de la siguiente forma (Versteeg & Malalasekera, 1995):

$$\phi_e = \phi_p + \frac{1}{2}\Psi(r)(\phi_e - \phi_p) \quad \text{Ecuación 9-8}$$

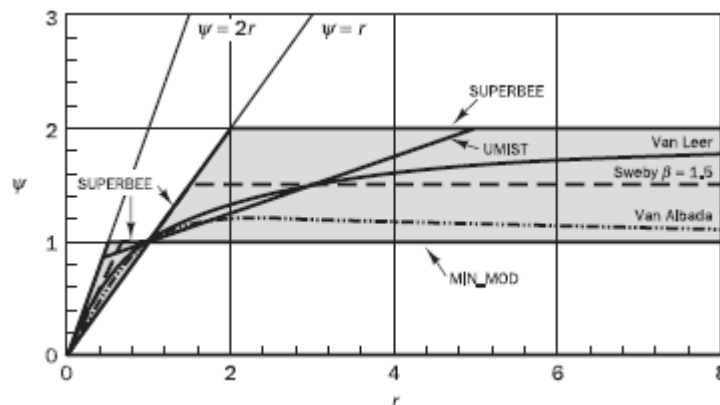
$$r = \frac{\phi_p - \phi_w}{\phi_e - \phi_p} \quad \text{Ecuación 9-9}$$

Dependiendo del valor de este limitador  $\Psi(r)$  tendremos un esquema u otro. Los que se han usado son:

**Tabla 10. Esquemas usados**

Esquema	Función limitadora $\Psi(r)$
Upwind	0
LinearUpwind	r
VanLeer	$\frac{r +  r }{1 + r}$

Existen otros esquemas dependiendo de la función  $\Psi(r)$ . Podemos verlos gráficamente en la siguiente figura.



**Figura 9.20 Esquemas TVD – Representación diferentes funciones limitadoras  $\Psi(r)$  ((Versteeg & Malalasekera, 1995) , Fig 5.25 - p171)**

Otro de los esquemas usados es el *linearUpwindV*. Este esquema está adaptado para campos vectoriales (la velocidad) y es similar al *linearUpwind*. La diferencia es que solo usa un limitador para las tres componentes, a diferencia de su versión más general, que calcula un limitador diferente para cada componente. El limitador único se calcula teniendo en cuenta la dirección en la que más rápido cambia el gradiente (OpenCFD Ltd., 2022). Este método es más estable, pero algo menos preciso.

En simulaciones estacionarias, se incluye un término adicional para garantizar una mayor estabilidad en la simulación (OpenCFD Ltd., 2022). Esto se consigue añadiendo el término

*bounded* cuando se introducen los esquemas numéricos a usar (Anexo C.c). Quedando de la forma:

$$\text{div}(\text{phi}, U) \quad \text{bounded Gauss linearUpwindV grad}(U)$$

Lo que hace esta opción es añadir un término en la derivada material de las ecuaciones (Greenshields, 2011) tal que:

$$\frac{D\phi}{Dt} = \frac{\partial\phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = \frac{\partial\phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) - (\nabla \cdot \mathbf{u})\phi \quad \text{Ecuación 9-10}$$

Sabemos que una vez la solución numérica ha convergido, para flujos incompresibles, el término  $\nabla \cdot \mathbf{u} = 0$ , esto implica que este término adicional es 0 y no tiene influencia en el resultado. Sin embargo, antes de que la solución converja  $\nabla \cdot \mathbf{u} \neq 0$ . Esto puede ayudar a mantener *boundedness* de la solución, sobre todo para simulaciones estacionarias como se ha comentado previamente (Greenshields, 2011).

### iii. laplacianSchemes

En este caso, como se menciona en los manuales, todas las opciones se basan en el teorema de Gauss (Greenshields, 2011; OpenCFD Ltd., 2022). Lo que debemos escoger es un esquema de interpolación para obtener los valores en las caras a partir de los valores en el centro de las celdas (*interpolationScheme* en OpenFOAM) y un esquema para calcular el gradiente normal a las superficies (*snGradSchemes* en OpenFOAM). Para calcular el gradiente en el centro de la cara usaremos un método Green-Gauss basado en la celda (Green-Gauss Cell Based):

$$\nabla_f^i \phi = \alpha \frac{\phi_p - \phi_w}{|\mathbf{d}|} \quad \text{Ecuación 9-11}$$

$$\nabla_f^i \phi = \alpha \frac{\phi_p - \phi_w}{|\mathbf{d}|} + (\hat{\mathbf{n}} - \alpha \hat{\mathbf{d}}) \cdot (\nabla \phi)_f \quad \text{Ecuación 9-12}$$

En la mayoría de las simulaciones en este trabajo se ha usado un esquema *limited 0.5*. Esta opción combina el esquema *corrected* (Ecuación 9-12) con el esquema *uncorrected* (Ecuación 9-11), no permitiendo que la contribución no ortogonal exceda la parte ortogonal (OpenCFD Ltd., 2022). Además, incluye un corrector por el ángulo ( $\theta$ ) que forman el vector normal y el vector que une el punto P y el W (en nuestro ejemplo de la Figura 9.18 sería 0 y el corrector  $\alpha = 1/\cos(\theta) = 1$ ).

### c. Ecuaciones URANS

Para el caso transitorio y flujo incompresible, las ecuaciones quedan:

$$\frac{\partial \tilde{u}_i}{\partial x_i} = 0 \quad \text{Ecuación 9-13}$$

$$\frac{\partial \tilde{u}_i}{\partial t} + \frac{\partial (\tilde{u}_i \tilde{u}_j)}{\partial x_j} = -\frac{1}{\rho} \frac{d\bar{p}}{dx_i} + \nu \frac{\partial^2 \tilde{u}_i}{\partial x_j^2} - \frac{\partial (\tilde{u}_i \tilde{u}'_j)}{\partial x_j} \quad \text{Ecuación 9-14}$$

$$\tilde{u}_i \tilde{u}'_j = \nu_t \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} k \quad \text{Ecuación 9-15}$$

Donde  $\tilde{u}_i$  indica que la integral sobre el tiempo se ha hecho con un valor de t suficientemente más grande que las fluctuaciones turbulentas pero menor que las estructuras coherentes y deterministas existentes en el flujo.

## d. Modelo de Turbulencia k- $\omega$ SST

En este anexo se introducen las ecuaciones que rigen el modelo y las constantes usadas.

$$\frac{D\rho\omega}{Dt} = \alpha\rho S^2 - \rho\beta\omega^2 + \frac{\partial}{\partial x_i} \left[ (\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_i} \right] + 2(1 - F_1)\rho\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i} \quad \text{Ecuación 9-16}$$

$$\frac{D\rho k}{Dt} = \tilde{\rho}k - \rho\beta^*\omega k + \frac{\partial}{\partial x_i} \left[ (\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_i} \right] \quad \text{Ecuación 9-17}$$

$$\tilde{\rho}k = \min \left( \tau_{ij} \frac{\partial u_i}{\partial x_j}, 20\beta^*\omega K \right); \quad \tau_{ij} = \mu_t \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij} \quad \text{Ecuación 9-18}$$

$$\mu_t = a_1 \frac{\rho k}{\max(a_1 \omega, SF_2)} \quad \text{Ecuación 9-19}$$

$$\alpha = F_1 \alpha_1 + (1 - F_1) \alpha_2 \quad \text{Ecuación 9-20}$$

$$F_1 = \tanh \left( \left[ \min \left( \max \left( \frac{\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right), \frac{4\rho\sigma_{\omega 2} k}{CD_{k\omega} y^2} \right) \right]^4 \right) \quad \text{Ecuación 9-21}$$

$$F_2 = \tanh \left( \left[ \max \left( \frac{2\sqrt{k}}{\beta^* \omega y}, \frac{500\nu}{y^2 \omega} \right) \right]^2 \right) \quad \text{Ecuación 9-22}$$

$$CD_{k\omega} = \max \left( 2\rho\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}, 10^{-10} \right) \quad \text{Ecuación 9-23}$$

**Tabla 11. Constantes del modelo SST k- $\omega$**

$\sigma_{k1}$	$\sigma_{k2}$	$\sigma_{\omega 1}$	$\sigma_{\omega 2}$	$\beta_1$	$\beta_2$
0.85	1	0.5	0.856	0.075	0.0828
$\alpha_1$	$\alpha_2$	$\beta^*$	$a_1$	$b_1$	$c_1$
5/9	0.44	0.09	0.31	1	10

## e. Tabla resumen simulaciones modelo simplificado

Para mayor claridad, en este apartado se muestran las diferentes combinaciones simuladas:

**Tabla 12. Casos simulados – Modelo simplificado**

Angle Difusor (°)	RH (mm)	Malla	Esquema numérico div (phi,U)	Esquema numérico div (phi,k/omega)
10	100	Hexa	<i>linearUpwind</i>	<i>upwind</i>
		Hexa	<i>linearUpwindV</i>	<i>upwind</i>
		Hexa	<i>linearUpwindV</i>	<i>VanLeer</i>
		Hexa	<i>linearUpwindV</i>	<i>linearUpwind</i>
		Tetra	<i>linearUpwindV</i>	<i>upwind</i>
10	80	Hexa	<i>linearUpwind</i>	<i>upwind</i>
		Hexa	<i>linearUpwindV</i>	<i>upwind</i>
10	60	Hexa	<i>linearUpwind</i>	<i>upwind</i>
		Hexa	<i>linearUpwindV</i>	<i>upwind</i>
		Tetra	<i>linearUpwindV</i>	<i>upwind</i>
10	40	Hexa	<i>linearUpwind</i>	<i>upwind</i>
		Hexa	<i>linearUpwindV</i>	<i>upwind</i>
		Tetra	<i>linearUpwindV</i>	<i>upwind</i>
4	100	Hexa	<i>linearUpwind</i>	<i>upwind</i>
		Hexa	<i>linearUpwindV</i>	<i>upwind</i>
4	80	Hexa	<i>linearUpwind</i>	<i>upwind</i>
		Hexa	<i>linearUpwindV</i>	<i>upwind</i>
4	60	Hexa	<i>linearUpwind</i>	<i>upwind</i>
		Hexa	<i>linearUpwindV</i>	<i>upwind</i>
4	40	Hexa	<i>linearUpwind</i>	<i>upwind</i>
		Hexa	<i>linearUpwindV</i>	<i>upwind</i>

Las mallas tetraédricas se han hecho con las mismas condiciones que las hexaédricas. Con esto, se quiere analizar las diferencias entre un tipo de malla u otro. No se podrán hacer conclusiones generales debido a que solo se han simulado tres casos.

No se han añadido más casos, simulando todo con esquemas de mayor orden a *upwind* debido a que se decidió continuar con el modelo más complejo.

## Anexo C. Ficheros OpenFOAM

Solo se ha incluido el último set up que ha ofrecido mejores resultados. En el caso del modelo con ruedas, se han incluido otras fronteras, pero con las mismas condiciones de contorno usadas en el caso sin ruedas.

### a. 0

#### K

```
FoamFile
{
  version 2.0;
  format  ascii;
  class   volScalarField;
  location "0";
  object  k;
}
/*-----*/
/*-----*/
dimensions [0 2 -2 0 0 0 0];

internalField uniform 0.227;

boundaryField
{
  ref3_vidresTipusCar
  {
    type kqRWallFunction;
    value uniform 0.227;
  }

  ref1_pilars
  {
    type kqRWallFunction;
    value uniform 0.227;
  }

  ref2_Letrals_sostre
  {
    type kqRWallFunction;
    value uniform 0.227;
  }

  ref2_manecPorta
  {
    type kqRWallFunction;
    value uniform 0.227;
  }

  ref2_rearTop
  {
    type kqRWallFunction;
    value uniform 0.227;
  }

  ref2_rearCar
  {
    type kqRWallFunction;
    value uniform 0.227;
  }

  ref2_underbody
  {
    type kqRWallFunction;
    value uniform 0.227;
  }

  ref2_frontCar
  {
    type kqRWallFunction;
    value uniform 0.227;
  }

  ref4_car
  {
    type kqRWallFunction;
    value uniform 0.227;
  }

  ref3_vidresLatFront
  {
    type kqRWallFunction;
    value uniform 0.227;
  }

  road_far
  {
    type kqRWallFunction;
    value uniform 0.227;
  }

  sym_right
  {
    type kqRWallFunction;
    value uniform 0.227;
  }
}
```

```

    {
        type symmetry;
    }

    sym_left
    {
        type symmetry;
    }

    sym_top
    {
        type symmetry;
    }

    inlet
    {
        type fixedValue;
        value uniform 0.227;
    }

    outlet
    {
        type inletOutlet;
        value uniform 0.227;
        inletValue uniform 0.227;
    }

    ref1_rearquarter
    {
        type kqRWallFunction;
        value uniform 0.227;
    }
    
```

```

    road_near
    {
        type kqRWallFunction;
        value uniform 0.227;
    }

    ref2_faros
    {
        type kqRWallFunction;
        value uniform 0.227;
    }

    ref2_reixes
    {
        type kqRWallFunction;
        value uniform 0.227;
    }

    ref1_pasRodaZonaCrit
    {
        type kqRWallFunction;
        value uniform 0.227;
    }

    ref2_areaRoda
    {
        type kqRWallFunction;
        value uniform 0.227;
    }
    }
    
```

### nut

```

FoamFile
{
    version 2.0;
    format ascii;
    class volScalarField;
    location "0";
    object nut;
}
/*-----*/
/*-----*/
/*-----*/

dimensions [0 2 -1 0 0 0];

internalField uniform 0.0001;

boundaryField
{
    
```

```

    ref3_vidresTipusCar
    {
        type nutkWallFunction;
        Cmu 0.09;
        kappa 0.41;
        epsilon 9.8;
        value uniform 0.0001;
    }

    ref1_pilars
    {
        type nutkWallFunction;
        Cmu 0.09;
        kappa 0.41;
        epsilon 9.8;
        value uniform 0.0001;
    }

    ref2_Letrals_sostre
    {
        type nutkWallFunction;
    }
    
```

```
        Cmu 0.09;
        kappa 0.41;
        epsilon 9.8;
        value uniform 0.0001;
    }

    ref2_manecPorta
    {
        type nutkWallFunction;
        Cmu 0.09;
        kappa 0.41;
        epsilon 9.8;
        value uniform 0.0001;
    }

    ref2_rearTop
    {
        type nutkWallFunction;
        Cmu 0.09;
        kappa 0.41;
        epsilon 9.8;
        value uniform 0.0001;
    }

    ref2_rearCar
    {
        type nutkWallFunction;
        Cmu 0.09;
        kappa 0.41;
        epsilon 9.8;
        value uniform 0.0001;
    }

    ref2_underbody
    {
        type nutkWallFunction;
        Cmu 0.09;
        kappa 0.41;
        epsilon 9.8;
        value uniform 0.0001;
    }

    ref2_frontCar
    {
        type nutkWallFunction;
        Cmu 0.09;
        kappa 0.41;
        epsilon 9.8;
        value uniform 0.0001;
    }

    ref4_car
    {
        type nutkWallFunction;
        Cmu 0.09;
        kappa 0.41;
        epsilon 9.8;
        value uniform 0.0001;
    }

    }

    ref3_vidresLatFront
    {
        type nutkWallFunction;
        Cmu 0.09;
        kappa 0.41;
        epsilon 9.8;
        value uniform 0.0001;
    }

    road_far
    {
        type nutkWallFunction;
        Cmu 0.09;
        kappa 0.41;
        epsilon 9.8;
        value uniform 0.0001;
    }

    sym_right
    {
        type symmetry;
    }

    sym_left
    {
        type symmetry;
    }

    sym_top
    {
        type symmetry;
    }

    inlet
    {
        type calculated;
        value uniform 0.;
    }

    outlet
    {
        type calculated;
        value uniform 0.;
    }

    ref1_rearquarter
    {
        type nutkWallFunction;
        Cmu 0.09;
        kappa 0.41;
        epsilon 9.8;
        value uniform 0.0001;
    }
}
```

```

road_near
{
    type nutkWallFunction;
    Cmu 0.09;
    kappa 0.41;
    epsilon 9.8;
    value uniform 0.0001;
}

ref2_faros
{
    type nutkWallFunction;
    Cmu 0.09;
    kappa 0.41;
    epsilon 9.8;
    value uniform 0.0001;
}

ref2_reixes
{
    type nutkWallFunction;
    Cmu 0.09;
    kappa 0.41;
    epsilon 9.8;
    value uniform 0.0001;
}

ref1_pasRodaZonaCrit
{
    type nutkWallFunction;
    Cmu 0.09;
    kappa 0.41;
    epsilon 9.8;
    value uniform 0.0001;
}

ref2_areaRoda
{
    type nutkWallFunction;
    Cmu 0.09;
    kappa 0.41;
    epsilon 9.8;
    value uniform 0.0001;
}
    
```

### **omega**

```

FoamFile
{
    version 2.0;
    format ascii;
    class volScalarField;
    object omega;
}
/*-----*/
/*-----*/

dimensions [0 0 -1 0 0 0];

internalField uniform 1493.133;

boundaryField
{
    ref3_vidresTipusCar
    {
        type omegaWallFunction;
        value uniform 1493.133;
    }

    ref1_pilars
    {
        type omegaWallFunction;
        value uniform 1493.133;
    }

    ref2_Letrals_sostre
    {
        type omegaWallFunction;
        value uniform 1493.133;
    }

    ref2_manecPorta
    {
        type omegaWallFunction;
        value uniform 1493.133;
    }

    ref2_rearTop
    {
        type omegaWallFunction;
        value uniform 1493.133;
    }

    ref2_rearCar
    {
        type omegaWallFunction;
        value uniform 1493.133;
    }

    ref2_underbody
    {
        type omegaWallFunction;
        value uniform 1493.133;
    }

    ref2_frontCar
    {
        type omegaWallFunction;
        value uniform 1493.133;
    }
}
    
```

```

{
    type omegaWallFunction;
    value uniform 1493.133;
}

ref4_car
{
    type omegaWallFunction;
    value uniform 1493.133;
}

ref3_vidresLatFront
{
    type omegaWallFunction;
    value uniform 1493.133;
}

road_far
{
    type omegaWallFunction;
    value uniform 1493.133;
}

sym_right
{
    type symmetry;
}

sym_left
{
    type symmetry;
}

sym_top
{
    type symmetry;
}

inlet
{
    type fixedValue;
    value uniform 1493.133;
}

outlet
{
    type inletOutlet;
    value uniform 1493.133;
    inletValue uniform 1493.133;
}

ref1_rearquarter
{
    type omegaWallFunction;
    value uniform 1493.133;
}

road_near
{
    type omegaWallFunction;
    value uniform 1493.133;
}

ref2_faros
{
    type omegaWallFunction;
    value uniform 1493.133;
}

ref2_reixes
{
    type omegaWallFunction;
    value uniform 1493.133;
}

ref1_pasRodaZonaCrit
{
    type omegaWallFunction;
    value uniform 1493.133;
}

ref2_areaRoda
{
    type omegaWallFunction;
    value uniform 1493.133;
}
}
    
```

## U

```

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

/*-----*/
/*-----*/

dimensions [0 1 -1 0 0 0];

internalField uniform ( 0. 0. 0. );
    
```

```
boundaryField
{
    ref3_vidresTipusCar
    {
        type fixedValue;
        value uniform (0. 0. 0.);
    }

    ref1_pilars
    {
        type fixedValue;
        value uniform (0. 0. 0.);
    }

    ref2_Letrals_sostre
    {
        type fixedValue;
        value uniform (0. 0. 0.);
    }

    ref2_manecPorta
    {
        type fixedValue;
        value uniform (0. 0. 0.);
    }

    ref2_rearTop
    {
        type fixedValue;
        value uniform (0. 0. 0.);
    }

    ref2_rearCar
    {
        type fixedValue;
        value uniform (0. 0. 0.);
    }

    ref2_underbody
    {
        type fixedValue;
        value uniform (0. 0. 0.);
    }

    ref2_frontCar
    {
        type fixedValue;
        value uniform (0. 0. 0.);
    }

    ref4_car
    {
        type fixedValue;
        value uniform (0. 0. 0.);
    }
}
```

```
ref3_vidresLatFront
{
    type fixedValue;
    value uniform (0. 0. 0.);
}

road_far
{
    type fixedValue;
    value uniform (38.8889 0. 0.);
}

sym_right
{
    type symmetry;
}

sym_left
{
    type symmetry;
}

sym_top
{
    type symmetry;
}

inlet
{
    type fixedValue;
    value uniform (38.8889 0 0);
}

outlet
{
    type inletOutlet;
    inletValue uniform (0. 0. 0.);
    value uniform (38.8889 0. 0.);
}

ref1_rearquarter
{
    type fixedValue;
    value uniform (0. 0. 0.);
}

road_near
{
    type fixedValue;
    value uniform (38.8889 0. 0.);
}

ref2_faros
{
    type fixedValue;
    value uniform (0. 0. 0.);
}
```

<pre> ref2_reixes {     type fixedValue;     value    uniform (0. 0. 0.); }  ref1_pasRodaZonaCrit {     type fixedValue;         </pre>	<pre>         value    uniform (0. 0. 0.);     }      ref2_areaRoda     {         type fixedValue;         value    uniform (0. 0. 0.);     } }         </pre>
---	--

**p**

<pre> FoamFile {     version 2.0;     format  ascii;     class  volScalarField;     location "0";     object  p; } /*-----*/ /*-----*/  dimensions [0 2 -2 0 0 0];  internalField uniform 0.;  boundaryField {     ref3_vidresTipusCar     {         type zeroGradient;     }      ref1_pilars     {         type zeroGradient;     }      ref2_Letrals_sostre     {         type zeroGradient;     }      ref2_manecPorta     {         type zeroGradient;     }      ref2_rearTop     {         type zeroGradient;     } }         </pre>	<pre> ref2_rearCar {     type zeroGradient; }  ref2_underbody {     type zeroGradient; }  ref2_frontCar {     type zeroGradient; }  ref4_car {     type zeroGradient; }  ref3_vidresLatFront {     type zeroGradient; }  road_far {     type zeroGradient; }  sym_right {     type symmetry; }  sym_left {     type symmetry; }  sym_top {         </pre>
---	---

```

        type symmetry;
    }

    inlet
    {
        type zeroGradient;
    }

    outlet
    {
        type fixedValue;
        value uniform 0.;
    }

    ref1_rearquarter
    {
        type zeroGradient;
    }

    road_near
    {
        type zeroGradient;
    }
}

}

ref2_faros
{
    type zeroGradient;
}

ref2_reixes
{
    type zeroGradient;
}

ref1_pasRodaZonaCrit
{
    type zeroGradient;
}

ref2_areaRoda
{
    type zeroGradient;
}
}
}

```

## b. constant

### fvOptions

```

FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    location "";
    object fvOptions;
}
/*-----*/
/*-----*/

Velocity_Limiter
{
    type limitVelocity;
    active yes;
    selectionMode all;
    max 130;
}

```

### turbulenceProperties

```

FoamFile
{
    version 2;
    format ascii;
    class dictionary;
    location "constant";
    object turbulenceProperties;
}

simulationType RAS;

RAS
{
    RASModel kOmegaSST;
    turbulence on;
    printCoeffs on;
}

// ***** //

```

## c. system

### fvSchemes

```
FoamFile
{
    version 2.0;
    format  ascii;
    class   dictionary;
    object  fvSchemes;
}
/*-----*/
/*-----*/
ddtSchemes
{
    default  steadyState;
}
gradSchemes
{
    default  cellLimited Gauss linear 1;
}
divSchemes
{
    default  none;
    div(phi,U)  bounded Gauss upwind;
    div(phi,k)  bounded Gauss upwind;
    div(phi,omega)  bounded Gauss upwind;
    div(phi,epsilon)  bounded Gauss upwind;
    div(phi,nuTilda)  bounded Gauss upwind;
    div((nuEff*dev2(T(grad(U))))  Gauss linear;;
}
}
```

```
laplacianSchemes
{
    default  Gauss linear limited 0.5;
}
interpolationSchemes
{
    default  linear;
}
snGradSchemes
{
    default  limited 0.5;
}
fluxRequired
{
    default  no;
    p;
}
wallDist
{
    method  meshWave;
}
```

### fvSolution

```
FoamFile
{
    version 2.0;
    format  ascii;
    class   dictionary;
    object  fvSolution;
}
/*-----*/
/*-----*/
solvers
{
    "p.*"
    {
        solver      GAMG;
        tolerance   1e-10;
        relTol      0.05;
        smoother    GaussSeidel;
        nPreSweeps  0;
        nPostSweeps 2;
    }
}
```

```
cacheAgglomeration on;
agglomerator  faceAreaPair;
nCellsInCoarsestLevel 10;
mergeLevels  1;
maxIter      15;
}
"U.*"
{
    solver      smoothSolver;
    smoother    GaussSeidel;
    tolerance   1e-10;
    relTol      0.1;
    nSweeps     1;
}
Phi
{
    $p;
}
```

```
"(k|omega|)"
{
    solver      smoothSolver;
    smoother    GaussSeidel;
    tolerance    1e-10;
    relTol      0.1;
    nSweeps     1;
}
}
SIMPLE
{
    nNonOrthogonalCorrectors 1;
```

```
residualControl
{
    p      1e-8;
    U      1e-8;
    k      1e-10;
    omega  1e-10;
}
}
potentialFlow
{
    nNonOrthogonalCorrectors 50;
}
```

### controlDict

```
FoamFile
{
    version 2.0;
    format binary;
    class dictionary;
    location "";
    object controlDict;
}
/*-----*/
/*-----*/

libs
(
);

application simpleFoam;

startFrom startTime;

startTime      0.5;

stopAt endTime;

endTime      1;

deltaT 0.0001;

writeControl timeStep;

writeInterval 2500;

purgeWrite 2;

writeFormat binary;
```

```
writePrecision 6;

writeCompression off;

timeFormat general;

timePrecision 6;

graphFormat raw;

runTimeModifiable true;

functions {

    force_coefs
    {
        type forceCoeffs;
        libs ("libforces.so");
        writeControl timeStep;
        writeInterval 1;
        patches
        (
            ref1_pasRodaZonaCrit
            ref2_areaRoda
            ref1_rearquarter
            ref2_faros
            ref2_reixes
            ref2_rearTop
            ref2_rearCar
            ref2_underbody
            ref2_frontCar
            ref4_car
            ref3_vidresLatFront
            ref3_vidresTipusCar
            ref1_pilars
            ref2_Letrals_sostre
            ref2_manecPorta
        );
    }
}
```

```
log true;
UName U;
rho rhoInf;
pRef 0.;
rhoInf 1.217;
CofR (0. 0. 0.);
liftDir (0. 0. 1.);
dragDir (1. 0. 0.);
pitchAxis (0. 1. 0.);
magUInf 38.8889;
lRef 4.61;
Aref 2.12;
}
min_max_velocity
{
    type fieldMinMax;
    libs ( "libfieldFunctionObjects.so"

writeControl timeStep;
writeInterval 1;
log true;

);

mode magnitude;
fields
(
    U
);
}
min_max_pressure
{
    type fieldMinMax;
    libs ( "libfieldFunctionObjects.so"

);

writeControl timeStep;
writeInterval 1;
log true;
mode magnitude;
fields
(
    p
);
}
}
```