



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



INSTITUTO UNIVERSITARIO DE MATEMÁTICA PURA Y APLICADA

Estudio CFD del ascenso del cohete VEGA



Máster Universitario en Ingeniería Aeronáutica

Autor Carlos García Melero

Directores Sergio Hoyas Calvo
Lluís M. García Raffi
Mara Salut Escartí Guillem

Valencia, 15 de septiembre de 2020

Resumen

Estudio CFD del ascenso del cohete VEGA

Uno de los puntos más críticos durante el lanzamiento de un vehículo espacial se encuentra en los primeros instantes. En este periodo en el que se produce una gran aceleración, esta genera una gran cantidad de ruido y vibraciones que pueden afectar en gran medida a la carga de pago y a la estructura del objeto en cuestión. Este proyecto es parte de una investigación de la Universitat Politècnica de València en colaboración con la Agencia Espacial Europea. El trabajo trata de simular el despegue y el ascenso de un vehículo espacial mediante el uso de la mecánica de fluidos computacional y, con ello, analizar tanto el impacto de dichas vibraciones como la influencia que tiene el movimiento del vehículo en la generación y la propagación de ruido. Por otra parte, también se ha buscado la optimización de la malla en uso con la finalidad de simular con mayor precisión el comportamiento del flujo que se eyecta por la tobera de la aeronave. Mediante la visualización de las fotografías de Schlieren y el SPL (Sound Pressure Level), se observa cómo el movimiento del cohete tiene un gran impacto sobre la generación de ruido en el fairing del cohete.

Dichas simulaciones se realizan mediante la herramienta OpenFOAM que utiliza un software gratuito y de código abierto.

Palabras clave: CFD, cohetes, VEGA, generación de ruido, Overset, despegue

Abstract

CFD study of the ascent of the VEGA rocket

One of the critical points during a spacecraft launching happens in the first moments of this process. In this period, a huge acceleration is produced and it causes a big amount of noise and vibrations that can largely affect the payload and the structure of the space vehicle. This project is part of an investigation from “Universitat Politècnica de València” in cooperation with the European Space Agency (ESA). The goal of this task is to simulate the stages of take-off and initial climb from a spacecraft using Computational Fluid Dynamics (CFD) and, thus, to analyse the impact of these vibrations and the influence that the movement of the vehicle has in the noise generation and propagation. Otherwise, also the optimisation of the mesh use has been sought in order to simulate, with a better precision, the flow behaviour by the nozzle. Thanks to the pictures of Schlieren and the SPL (Sound Pressure Level), it is observed that the rocket’s movement has a big influence in the noise generation in the rocket’s fairing.

These simulations are performed by the tool OpenFOAM that uses free software and open source.

Keywords: CFD, rockets, VEGA, noise generation, Overset, take-off

Resum

Estudi CFD de l'ascens del coet VEGA

Un dels punts crítics durant el llançament d'un vehicle espacial es troba en els primers instants. En aquest període es produeix una gran acceleració, aquesta genera una gran quantitat de soroll y de vibracions que poden afectar en gran manera la càrrega de pagament i l'estructura del objecte. Aquest projecte forma part d'una investigació amb la Universitat Politècnica de València en col·laboració amb l'Agència Espacial Europea (ESA). Aquest estudi tracta de simular l'enlairament i l'ascens d'un vehicle espacial mitjançant l'ús de la mecànica de fluids computacional i, amb això, analitzar tant l'impacte d'aquestes vibracions com la influència que té el moviment del vehicle en la generació i la propagació de soroll. D'altra banda, també s'ha buscat l'optimització de la malla utilitzada amb la finalitat de simular amb una major precisió, el comportament del flux que se ejecta per la tovera de l'aeronau. Mitjançant la visualització de les fotografies de Schlieren i el SPL, s'observa com el moviment del coet té un gran influx sobre la generació de soroll en el *fairing* del coet.

Aquestes simulacions es realitzen per mitjà de l'eina OpenFOAM que utilitza un software gratuït i de codi obert.

Paraules clau: CFD, coets, VEGA, generació de soroll, Overset, enlairament.

Índice general

1. Introducción	1
1.1. Introducción	1
1.2. Estudios previos	3
1.2.1. Historia del cohete VEGA	3
1.3. Motivación del estudio	7
1.4. Fases del proyecto	9
2. Estructura de la configuración de un modelo en CFD	11
2.1. Introducción al CFD	11
2.2. Estructura del caso CFD	13
2.3. Creación de la malla	16
2.4. Modelos matemáticos	18
2.4.1. Equilibrio de ecuaciones	18
2.4.2. Modelo de turbulencia $k - \omega$ SST	19
2.5. Métodos numéricos	23
2.5.1. Método de volúmenes finitos	23
2.5.2. Algoritmos de solución del acoplamiento presión - velocidad	23
3. Modelado de la capa límite	26
3.1. Herramientas para capas límite	27

3.1.1.	Cálculo del tamaño de la capa límite	31
3.2.	Discusión de los resultados	32
3.2.1.	Refinamiento de la malla	32
3.2.2.	Comparación de los resultados	37
4.	Modelado de la malla dinámica	41
4.1.	Introducción	41
4.1.1.	<i>Mesh Adaption-Dynamic Remeshing</i>	42
4.1.2.	Overset	43
4.2.	Implementación del Método Overset	46
4.3.	Caracterización del movimiento del cohete VEGA	50
4.3.1.	Propulsión en cohetes	51
4.4.	Discusión de los resultados	56
5.	Conclusiones	67
5.1.	Proyectos futuros	69
6.	Pliego de condiciones	70
6.1.	Condiciones del puesto de trabajo	70
6.1.1.	Lugares de trabajo	71
6.1.2.	Medidas de emergencia: vías y salidas de evacuación	71
6.1.3.	Condiciones de protección contra incendios	71
6.1.4.	Instalación eléctrica	71
6.1.5.	Condiciones termohigrométricas	72
6.1.6.	Iluminación	72
6.1.7.	Ergonomía: diseño del puesto de trabajo	72

6.1.8. Ruido	73
6.2. Condiciones de los recursos informáticos	74
6.2.1. Introducción	74
6.2.2. Condiciones de hardware	74
6.2.3. Condiciones del software	75
6.2.4. <i>Software</i> de simulación numérica	75
7. Presupuesto	76
7.1. Introducción	76
7.2. Tareas realizadas	76
7.3. Recursos	77
7.4. Presupuestos	78
7.4.1. Personal técnico	78
7.4.2. Materiales técnicos: equipo informático y material de oficina	79
7.4.3. Software	80
7.5. Presupuesto final	80
Bibliografía	82

Índice de figuras

1.1.	Ascenso del cohete VEGA [19].	1
1.2.	<i>Fairing</i> del cohete VEGA [18].	2
1.3.	Elementos del cohete VEGA [18].	5
1.4.	1ª etapa : motor P80 [18].	5
1.5.	Adaptador entre las etapas P80 y Zefiro-Z23 [18].	6
1.6.	2ª etapa: motor ZEFIRO 23 [18].	6
1.7.	3ª etapa: motor ZEFIRO 9 [18].	7
1.8.	4ª etapa: módulo AVUM [18].	7
2.1.	Estructura simplificada de un caso CFD en OpenFOAM [1].	14
2.2.	Inicio del mallado de una geometría con la herramienta <i>snappyHexMesh</i> [4].	17
2.3.	Final del mallado de una geometría con la herramienta <i>snappyHexMesh</i> [4].	17
2.4.	Distribución de la velocidad en la proximidad de una pared.	22
2.5.	Dominio discretizado en volúmenes de control [3].	23
3.1.	Adición de la capa límite mediante la herramienta <i>snappyHexMesh</i> [4].	27
3.2.	Representación del parámetro <i>expansionRatio</i> [5].	28
3.3.	Representación de <i>featureAngle</i> = 45° [5].	29
3.4.	Representación de <i>featureAngle</i> = 180° [5].	29
3.5.	Representación de <i>maxNonOrtho</i> [5].	31

3.6.	Representación del refinamiento central donde se va a propagar el chorro.	33
3.7.	Representación del refinamiento esférico y gradual.	34
3.8.	Representación de la capa límite en comparación con el refinamiento del chorro.	34
3.9.	Visualización de las capas de celdas de la capa límite.	34
3.10.	Representación del mallado realizado.	35
3.11.	Representación del mallado de las zonas más relevantes del caso.	36
3.12.	Representación del mallado total.	36
3.13.	Representación de las plataformas de lanzamiento implementadas.	36
3.14.	Representación de las plataformas de lanzamiento implementadas.	37
3.15.	Representación del campo de la presión en el dominio del deflector.	39
3.16.	Representación del campo de la presión en el dominio del deflector.	40
4.1.	Dos mallas superpuestas (de color gris y roja) y una malla <i>background</i> [8].	43
4.2.	Malla Overset. El color azul indica las celdas calculadas, el rojo el agujero y el gris las celdas interpoladoras [8].	44
4.3.	Mapeado de campos inconsistentes [4].	50
4.4.	Representación de la conservación del momento [23].	52
4.5.	Cambio de marco de referencia [23].	52
4.6.	Cambio de marco de referencia [23].	53
4.7.	Curva de velocidad de ascenso de VEGA [22] [21] [24].	55
4.8.	Representación de las mallas utilizadas.	56
4.9.	Representación de las mallas utilizadas.	58
4.10.	Representación de las mallas utilizadas.	60
4.11.	Representación del fairing.	62
4.12.	Representación de la velocidad en la zona inferior del cohete.	63
4.13.	Fotografías de Schlieren en el dominio de cálculo.	64

4.14. Fotografías de Schlieren para la visualización de las ondas de choque.	65
4.15. Nivel de presión sonora para la malla Overset.	66

Índice de cuadros

2.1. Tabla de coeficientes del modelo de turbulencia $k - \omega$ SST	20
4.1. Datos del despegue del cohete VEGA mediante vídeo	54
4.2. Datos del despegue del cohete VEGA mediante vídeo	55
4.3. Mallas calculadas	61
7.1. Presupuesto del personal	79
7.2. Presupuesto del material técnico	80
7.3. Presupuesto Final	81

Capítulo 1

Introducción

1.1. Introducción

En la actualidad, la investigación espacial se encuentra en un punto álgido y se ha desarrollado en gran medida hasta tratar de averiguar si existe otro lugar habitable o con posibilidades de ello y tratar de colonizarlo. Para ello se requiere una gran cantidad de recursos tecnológicos, lo que supone una gran inversión económica. Los recursos necesarios en dicha investigación tienen un alto valor económico, por lo tanto, se necesita un gran capital monetario para poder hacer frente a los costos que se requieren.



Figura 1.1: Ascenso del cohete VEGA [19].

Este trabajo se desarrolla con la intención de obtener un modelo capaz de simular el comportamiento durante el despegue y ascenso de un vehículo espacial. Con ello se pretende plasmar los

fenómenos que suceden en las inmediaciones de la nave y así poder estimar cuales son sus repercusiones tanto en la plataforma de despegue como en la propia aeronave. Así pues, se podría estudiar qué cargas soportan los materiales de los que se compone la aeronave y de este modo se podría abaratar en cierto modo esa inversión, ya que se sabría con una gran exactitud cuales son las necesidades que requiere la nave para poder realizar su misión.

En todas las aeronaves, la carga de mayor valor se ubica en el *fairing*, situado en la punta superior de la nave. En este lugar es donde se encuentra la carga de pago, es decir, todos los instrumentos o dispositivos que serán necesarios para realizar la función principal de la misión. Estas tareas pueden variar desde reabastecer a otro vehículo espacial hasta transportar material de investigación a otra aeronave o poner satélites en órbita.



Figura 1.2: *Fairing* del cohete VEGA [18].

Durante el despegue, que suele durar alrededor de dos minutos, es donde suceden los momentos más críticos de la misión. En ese tiempo, se produce una aceleración muy brusca que provoca unas vibraciones muy elevadas que pueden afectar a la estructura de la nave así como a la carga de pago. Los efectos que se originan pueden incluso llegar a provocar la rotura de las estructuras. Esto es debido a que durante el despegue el motor emite un chorro supersónico que crea una onda de presión que se propaga. Dicha onda es la culpable de la producción de dichas vibraciones.

En el caso de que se pueda conocer cómo avanza dicha onda de presión sobre la estructura y sobre la nave, se podrá estudiar qué impacto tendrá sobre la carga de pago y si esta podrá soportarlo o si habrá que diseñar un dispositivo que se encargue de amortiguar dichas perturbaciones.

1.2. Estudios previos

Hasta el momento, la Agencia Espacial Europea (ESA) ha investigado los fenómenos que suceden durante el encendido del motor y el ascenso de sus naves, con la finalidad de poder conocer las incidencias y poder reducir el ruido producido sobre las plataformas desde donde se lanza la aeronave así como sobre el mismo cohete.

Esta investigación parte de un proyecto previo realizado en la Universitat Politècnica de València (UPV). El proyecto es “*Estudio aeroacústico del cohete Vega*”, cuya autora es Mara Salut Escartí Guillem. Dicho proyecto consistió en la estimación del campo acústico durante el lanzamiento de un cohete. Para su realización se definió una malla en la que se encontraban las geometrías del cohete y las plataformas de lanzamiento. En esta malla, no se incluyó la capa límite sobre las superficies en las que impactaría el chorro supersónico. Además, el análisis que se realizó fue con un método de malla estática, en el que no se llegó a implementar el movimiento del cohete durante su lanzamiento. Ambos factores (capa límite y movimiento del cohete), como se verá más adelante, tienen influencia en el problema de la generación del ruido.

1.2.1. Historia del cohete VEGA

El cohete a estudiar ha sido de los primeros en utilizarse por la ESA. Este está diseñado en gran medida para lanzar cargas pequeñas que vienen a ser pequeños satélites con misiones científicas o de observación de la superficie terrestre.

El programa VEGA (*Vettore Europeo di Generazione Avanzata*) comenzó a desarrollarse en 1998 como fruto de la colaboración de la Agencia Espacial Italiana (ASI) y la ESA. Según informes de la ESA, consta que se invirtieron unos 710 millones de euros, a los que se deben de sumar los 76 millones aportados por la compañía Avio SpA, promotora del programa, para el desarrollo del motor cohete P80 [13].

Por lo tanto, como se puede observar, este proyecto es de ámbito únicamente Europeo, el cual encabeza Italia seguido de Francia, España, Bélgica, Holanda, Suiza y Suecia. Pese a que el primer lanzamiento estaba previsto para 2009, se tuvo que retrasar hasta el 2012 debido a una serie de imprevistos [13].

La primera misión que realizó fue la puesta en órbita de los satélites italianos LARES y ALMASat-1, además de siete CubeSats, nanosatélites de investigación, entre los que cabe remarcar uno de ellos que llevaba la marca España: el Xatcobeo. Este cohete lanzador se sumó al Ariane 5 y a los Soyuz 2, que son parientes de tamaño medio, a complementar las misiones de carga.

Esta primera misión fue clave para el programa de desarrollo de este lanzador ligero. El objetivo del primer vuelo fue el de cualificar el sistema de lanzamiento, incluyendo el vehículo, su infraestructura de lanzamiento y las operaciones que se debieran realizar. Dichas operaciones van desde el despegue del cohete hasta la separación de la carga útil y el desechado del módulo superior [13].

Las misiones que puede realizar VEGA son principalmente la puesta en órbita de satélites que necesitan de una una órbita baja, entre 250 y 1000 km de apogeo, trabajos de investigación y observación de la Tierra y programas de estudio del espacio como puede ser el demostrador de reentrada atmosférica IXV de la ESA. Hasta el momento, se han realizado 16 lanzamientos y para este año 2020 está programado uno más pero con un cohete VEGA con nuevos motores. El último ha sido lanzado el día 6 de septiembre de 2020 para poner en órbita el microsátélite ESAIL [14].

El cohete VEGA puede llevar cargas desde 300 hasta 2500 kg, dependiendo del tipo y de la altura de la órbita que se requiera por los clientes. El punto de referencia es para 1500 kg en una órbita polar a 700 km de altura. Cabe destacar que VEGA no lleva ningún tipo de tripulación y que la ubicación del lanzamiento es el Centro Espacial Europeo de Kourou, en la Guayana Francesa [14].

VEGA está compuesto de varias etapas como se puede ver en la Figura 1.3.





Figura 1.3: Elementos del cohete VEGA [18].

La primera etapa inicialmente era una versión de dos segmentos del Ariane-5 , pero posteriormente se modificó por un motor cohete de la familia del P80 y cuyas características fundamentales se verán en el proyecto ya que en la simulación solo se estudia la fase del despegue, donde únicamente actúa la primera etapa del cohete [14].

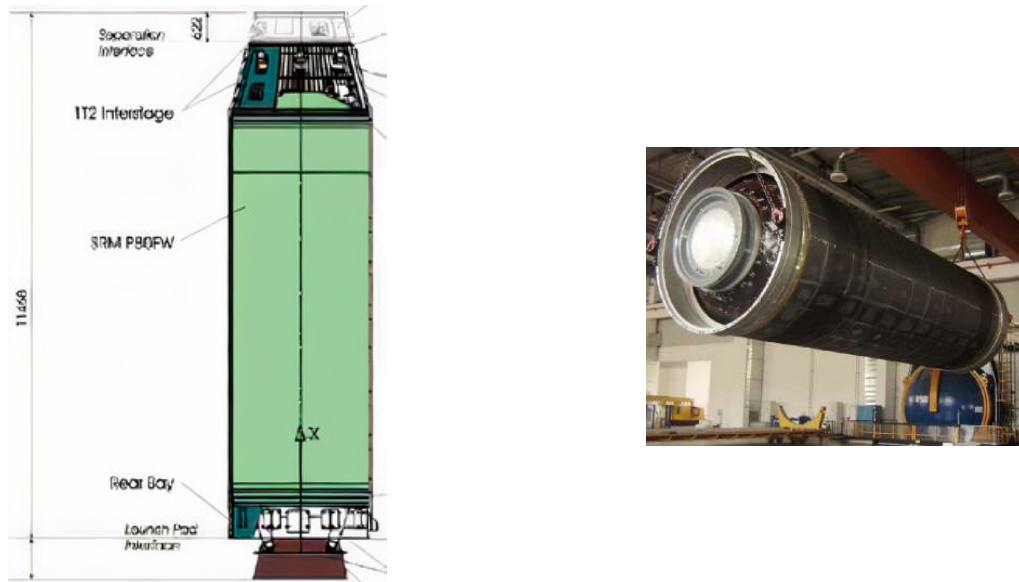


Figura 1.4: 1ª etapa : motor P80 [18].

Además de esta primera etapa las siguientes constan respectivamente del motor cohete ZEFIRO 23, ZEFIRO 9 y AVUM, donde se encuentra la carga de pago. [14]



Figura 1.5: Adaptador entre las etapas P80 y Zefiro-Z23 [18].

La segunda y la tercera etapa son versiones del motor italiano Zefiro, llamadas Z23 y Z9.

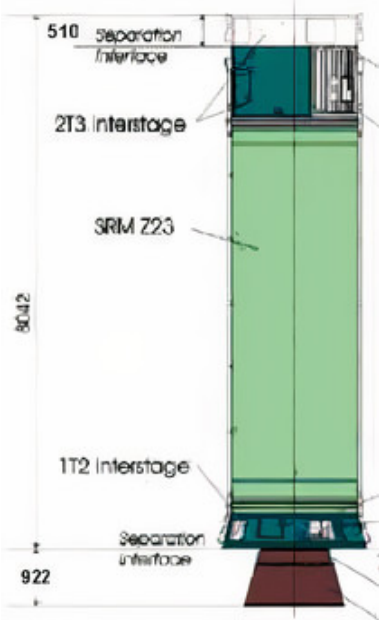


Figura 1.6: 2ª etapa: motor ZEFIRO 23 [18].

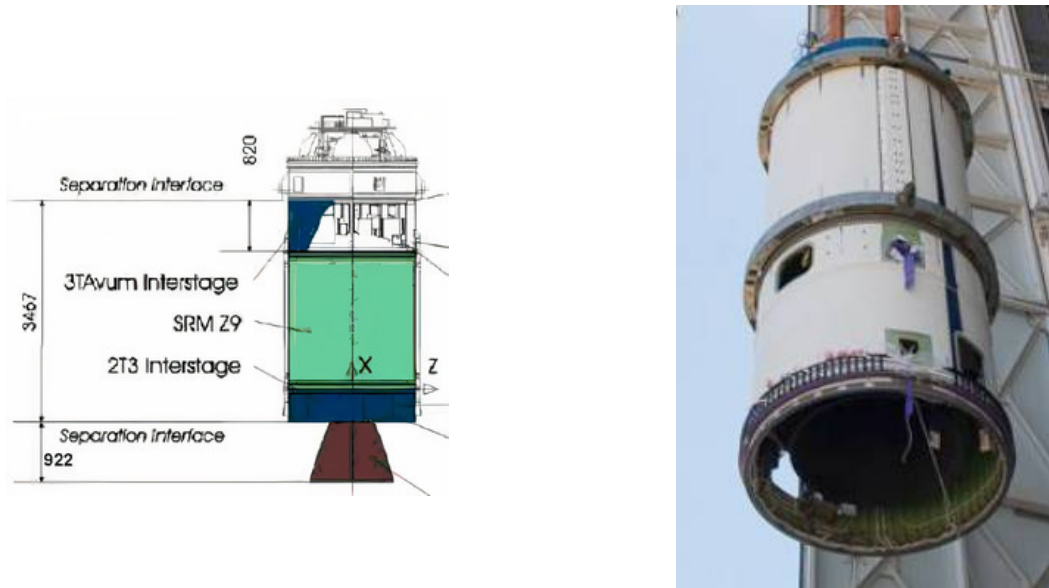


Figura 1.7: 3ª etapa: motor ZEFIRO 9 [18].

Por último, la 4ª etapa está formada por un módulo de propulsión de combustible líquido llamado AVUM, propulsado por el motor ruso RD-869. Debido a que la 4ª etapa es reconfigurable, el cohete VEGA es capaz de situar múltiples cargas de pago en órbita. En concreto, este cohete ofrece diversas configuraciones capaces de manejar desde un solo satélite hasta un satélite principal además de seis micro satélites [14].

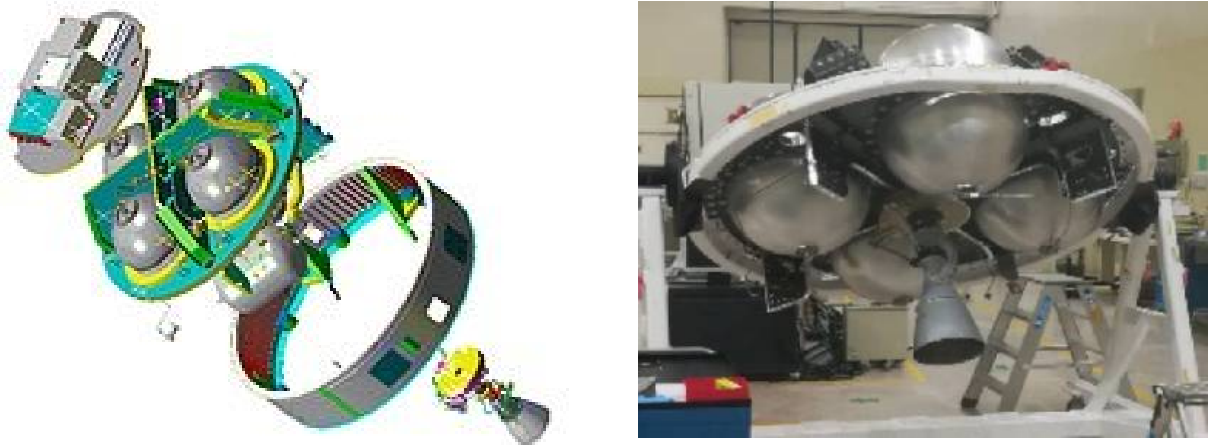


Figura 1.8: 4ª etapa: módulo AVUM [18].

1.3. Motivación del estudio

En el pasado, la investigación espacial era muy limitada debido a que requería mucho capital, el cual pocos países o empresas poseían. Los costes eran tan elevados dado que en ese entonces los avances tecnológicos se basaban en el método de ensayo y error. Sin embargo,

en la actualidad, se poseen herramientas más potentes para poder alcanzar esos objetivos suponiendo un coste mucho menor como pueden ser los métodos computacionales, en el que la inversión se realiza sobre un grupo de ingenieros, hardware y software. Este enfoque resulta más económico y seguro a largo plazo.

Por ello, la obtención de un método en el que se pueda simular el análisis del despegue de un cohete teniendo la oportunidad de cambiar los parámetros del cohete así como los de la plataforma de despegue, todo de forma computacional, facilita la entrada a nuevas empresas que se quieran sumar a la competición por la conquista del espacio.

El punto de partida de esta investigación será en un primer momento incluir la capa límite sobre las superficies que sean oportunas, para poder comparar el efecto del chorro sobre estas teniendo o no la capa límite. Gracias a ello se puede estudiar con mayor precisión los puntos de parada del flujo y la producción de ondas de choque.

Por otro lado, teniendo en cuenta que el despegue implica que el cohete tenga cierto movimiento; se implementará un modelo de malla móvil. El campo acústico generado varía completamente a partir del momento en el que el vehículo inicia su despegue. La fuente de ruido principal, que en este caso es el motor y los gases de escape que provienen de este, modifican su posición. En cuanto a las fuentes de ruido secundarias, debidas al impacto del chorro con las superficies del canal de evacuación de los gases de escape reducen su intensidad mientras la capa límite que se genera alrededor del cohete, especialmente en el *fairing*, surge como nueva fuente de ruido. Además, la propagación de las ondas de presión y su ascenso alrededor del cohete varían en el instante en el que se inicia el ascenso del vehículo.

Este modelo tendrá las mismas geometrías que el modelo estático, así como el chorro supersónico tendrá las mismas características. Para acelerar la simulación y, ya que se tiene la solución para el caso estático, se asume que a partir de un determinado instante el cohete comienza a moverse. De este modo, el dominio no parte desde el instante inicial, lo que supondría el comienzo del despegue del cohete y de la salida de los gases por la tobera, sino que parte desde un instante en el que la pluma ya se encuentra desarrollada y ha incidido sobre el deflector.

El método que se va a utilizar para analizar el impacto y los diferentes experimentos numéricos es un análisis mediante la dinámica de fluidos computacional (CFD). La practicidad del análisis CFD es que si se quiere obtener la información de manera experimental conllevaría un gasto de recursos y de instrumentos muy elevado. De igual manera, la instalación de dichos instrumentos y la obtención de datos de los mismos no sería del todo precisa debido a que si se quieren sacar datos realistas estos instrumentos deberían tener mejores sensores y encontrarse en gran cantidad alrededor del cohete, lo que resulta imposible. Si se pone un ejemplo, en la pluma del cohete es imposible sacar cualquier información veraz ya que la colocación de los dispositivos es inviable.

Por las razones comentadas anteriormente, se elige realizar un estudio CFD que, además, resulta mucho más económico y práctico ya que solo se necesitan ciertos parámetros característicos de la aeronave así como de las superficies cercanas que pueden influir en el análisis.

La herramienta que ha sido elegida para el proyecto es el software OpenFOAM construido como un conjunto de módulos de C++. Esta se encuentra de manera gratuita y de código abierto a diferencia de otros software que se dedican al análisis CFD. La ventaja que aporta frente a la competencia es que se puede modificar el código fuente para adaptarlo a diferentes condiciones y necesidades. Esto también puede acarrear algún inconveniente si no se conoce el lenguaje, pero para aquellos investigadores que conozcan dicho código o les sea fácil de interpretar, es una herramienta muy útil ya que pueden desarrollar, optimizar y probar diferentes subrutinas a medida.

Por otra parte, se han documentado muchos artículos para la industria aeronáutica utilizando dicha herramienta, aunque más enfocada al estudio de la aerodinámica o del análisis de ciertos motores.

1.4. Fases del proyecto

El objetivo principal del proyecto es la mejora del modelo CFD para la simulación del despegue del cohete para poder aproximarlos a las condiciones reales. Asimismo, se puede dividir en dos subobjetivos. En primer lugar, mejorar el mallado mediante la inclusión de una capa límite capaz de representar con mayor precisión los fenómenos cerca de la pared. En segundo lugar, implementar un modelo de malla móvil para representar el movimiento del cohete VEGA durante la fase de ascenso. Para poder cubrir todos estos puntos será necesario seguir los siguientes pasos:

- Comprender los mecanismos y las herramientas del software OpenFOAM para poder realizar simulaciones CFD.
- Estudiar las nuevas herramientas de OpenFOAM para poder definir mallas móviles (dinámicas).
- Generar las capas límite en las zonas adyacentes a las superficies de despegue como el deflector.
- Lanzar la simulación con los elementos del estudio previo y monitorizar el cálculo.
- Comparar y analizar la influencia de la adición de capa límite respecto al estudio previo.

- Estructurar un mallado óptimo para el caso dinámico en función de las herramientas que se pueden usar y decidir cuál utilizar.
- Averiguar mediante una serie de cálculos e investigación el movimiento que debe seguir el cohete durante la fase de despegue o ascenso.
- Generar la malla Overset, malla dinámica, sobre la que se realizará la simulación.
- Desarrollar un modelo de simulación para el caso Overset con todos los elementos apropiados, como los sistemas de discretización, el número de iteraciones, tolerancias, etc...
- Lanzar varias simulaciones CFD con la malla dinámica para poder hacer una comparación entre la estática y la dinámica.
- Obtener un análisis con el que poder identificar la influencia del aporte de movimiento al cohete.

Capítulo 2

Estructura de la configuración de un modelo en CFD

En esta sección del trabajo se definirán las bases del modelado CFD en relación al software OpenFOAM. Cabe mencionar que se han utilizado dos versiones de este software para este proyecto, las versiones v6 y v1912 de OpenFOAM, esta última posee la herramienta Overset que será la que se utilizará en el modelado de la malla dinámica; en esta última versión se encuentran todos los modelos descritos de la versión v6. A continuación se explicarán cuáles son los pasos y las herramientas necesarias para modelar un caso en CFD con dicho software.

2.1. Introducción al CFD

El acrónimo CFD proviene de *Computational Fluid Dynamics* o Dinámica de Fluidos Computacional. Se trata de una herramienta que trata un área de conocimiento como es la simulación numérica de flujos fluidos, transferencia de calor y otros fenómenos relacionados. Esta herramienta surgió de la fusión de la mecánica de fluidos con el cálculo numérico. Las ecuaciones que rigen estos flujos vienen dadas por la mecánica de los fluidos, las ecuaciones de Navier-Stokes, y son resueltas mediante métodos numéricos.

Hoy en día es una herramienta vastamente utilizada ya que permite obtener resultados con una precisión razonable y fiables con un coste operacional asequible. Una de las ventajas de los modelos CFD es que puede optimizar el proceso de diseño ya que se pueden incluir modelos físicos relevantes para el objeto en cuestión. Asimismo, también se puede utilizar para simular un caso experimental y comprender cuál ha sido el fallo que se ha podido producir y diseñar un mecanismo para evitar que vuelva a suceder.

En este caso, se podrán realizar análisis para comprender qué sucede en las inmediaciones del cohete y qué está ocurriendo sobre el mismo. Cabe destacar que la ESA no ha proporcionado todos los datos necesarios para poder recrear la física del chorro. Esta falta de datos puede

suponer en un futuro una posible fuente de error.

Existen tres etapas principales en los modelos CFD: el pre-procesado, los cálculos y el post-procesado. Durante el pre-procesado se define y se crea la geometría y la malla. Una vez acabada dicha etapa se define la parte más importante del modelo: se especifica la física del modelo y se realizan los cálculos pertinentes computacionalmente. Por último, en el post-proceso se visualizan los resultados obtenidos durante el cálculo mediante herramientas de post-procesado.

En la etapa del pre-procesado se define la geometría a analizar y se crea la malla en varias fases:

- Primeramente, se define la geometría de la región de interés así como el dominio total sobre el que se evaluará el análisis. Una vez definida las regiones de interés, se crean las regiones de fluido, las regiones sólidas y los nombres divisorios de cada superficie (aunque este último paso se haga previamente, al definir la geometría).
- Como segundo paso, se hacen los ajustes pertinentes a la creación de la malla como cuántos elementos globales queremos como máximo, los requerimientos de calidad de la malla, los refinamientos de la malla en las superficies de la geometría e incluso los refinamientos en capas límite. Este paso es esencial ya que los volúmenes de control en los que se dividirá el dominio serán sobre los que se realice el cálculo posterior.
- Por último, durante esta fase, también se deben seleccionar tanto el modelo físico que se va a utilizar como las propiedades del fluido. También se deben indicar las condiciones iniciales que tiene el flujo y en qué regiones se encuentran dichas condiciones iniciales. Cuanto mejor definidas estén las condiciones de contorno más se acercará el modelo al caso real.

Posteriormente, durante la fase de cálculo, el problema se resuelve mediante las ecuaciones en derivadas parciales que son integradas en los volúmenes de control, de ahí que se llame el método de los volúmenes finitos utilizando esquemas de discretización temporales y espaciales. Dichas ecuaciones se convierten a un sistema de ecuaciones algebraicas que se resuelven de manera iterativa.

Se considera que la solución ha convergido si:

- Se cumplen las ecuaciones de conservación en forma discreta en todas las celdas.
- La solución no varía con más iteraciones adicionales
- Se cumplen los balances de masa, energía y momento en todas las celdas.

Un último método para afirmar que ha convergido la solución es observar los residuales. Este término es una medida del error en las ecuaciones de conservación y es descrito como la diferencia del valor de una variable en una iteración y en la anterior. Normalmente, se puede afirmar que ha convergido el caso si los residuales se encuentran en torno al orden de 10^{-6} . Finalmente, para asegurar la convergencia se deben mirar los campos físicos más relevantes para demostrar también que la solución ha convergido, comparándolos con una posible solución teórica.

Por último, está la etapa del post-procesado, donde mediante una herramienta se puede analizar, visualizar y representar los resultados. Del mismo modo, la solución que se ha obtenido se comparará con los resultados experimentales, en el caso de que los haya, para ver cómo de realista es el modelo para poder hacer estudios posteriores sobre otros campos relevantes en el análisis a realizar. En este proyecto, cabe mencionar que no se puede realizar ningún tipo de validación experimental por falta absoluta de datos sobre el cohete.

2.2. Estructura del caso CFD

El software OpenFOAM carece de interfaz gráfica, por lo que el caso íntegro está formado por carpetas en las que se encuentran los ficheros con toda la información necesaria para poder analizarlo. Un caso se compone principalmente de tres carpetas donde se almacena prácticamente toda la información: *0*, *system* y *constant*. En estas carpetas se encuentran los archivos que explican las condiciones iniciales, de contorno, las propiedades de la malla, la geometría, las propiedades de la turbulencia, térmicas y de transporte, los esquemas de discretización, el control de simulación y el movimiento de la región de la malla indicada.

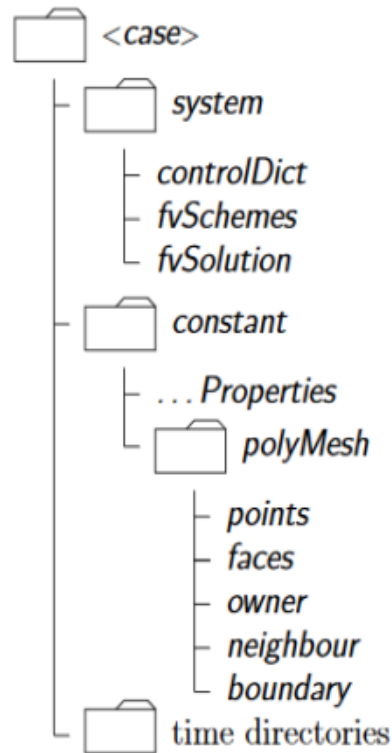


Figura 2.1: Estructura simplificada de un caso CFD en OpenFOAM [1].

Las carpetas que se van a desglosar a continuación son las que se encuentran en el modelo de la malla móvil, por lo que existirán un mayor número de archivos.

En la carpeta 0 es donde se encuentra almacenada la información de cada una de las variables. Se encuentran tanto los valores de las condiciones iniciales del campo fluido como las condiciones de contorno definidas en cada frontera de la malla. También se encuentran los ficheros necesarios para definir la malla Overset. En este caso se tiene una carpeta con nombre $0.orig$ que es donde se obtienen todos estos valores una vez se reinicie la malla, de este modo siempre se tiene guardado el valor inicial del caso [1].

En la carpeta *constant* se deposita la información correspondiente a la malla dentro de una subcarpeta llamada *polyMesh*. En esta subcarpeta se encuentran todos los datos correspondientes a los puntos, vértices, caras y fronteras del dominio; esta carpeta se crea una vez se ejecuta la malla. En el fichero *dynamicMeshDict* se encuentran los parámetros y el diccionario que se utilizarán posteriormente para el movimiento de la malla o de una de las regiones de la misma [1].

También se encuentra el diccionario *turbulenceProperties* que especifica el modelo de turbulencia a usar, así como sus parámetros. Otros diccionarios que se encuentran son el *thermophysicalProperties* donde se encuentran propiedades físicas necesarias para el caso, como son las ecuaciones de estado a utilizar, la viscosidad del fluido o los parámetros de transferencia de

calor. En función del *solver* que se vaya a utilizar en el caso, los parámetros pueden cambiar ya que puede ser que sean necesarios o que deban utilizarse otro tipo de parámetros.

En último lugar, en la carpeta *system*, se ubican los archivos más relevantes de la malla y de la simulación del caso. Los archivos que se encargan de la generación de la malla son *blockMeshDict* y *snappyHexMeshDict*. El primer fichero se encarga principalmente de definir el dominio global del caso, así como de los vértices y caras que lo conforman. El segundo, en cambio, necesita una malla inicial para poder ejecutarse, por ello se necesita siempre que se haya ejecutado primero el fichero *blockMeshDict* y posteriormente el fichero *snappyHexMeshDict*. Este último se caracteriza principalmente porque se pueden incluir geometrías diseñadas previamente y externas al programa. Asimismo, se definen las regiones de refinamiento que se van a dar en el dominio y ,por último, si se desea añadir capas límite en las superficies de las geometrías insertadas [1].

Por otro lado, se encuentran los ficheros que controlan la simulación del caso:

- En el fichero *controlDict* se encuentran los parámetros generales del caso, por ejemplo: se encuentra el tiempo de inicio y final de la simulación, la precisión de escritura de los campos, como el del tiempo, también se especifica el paso temporal así como el tiempo de escritura de los campos.
- En cuanto al fichero *fvSchemes* se especifican cuáles son los esquemas de discretización a utilizar en el caso.
- En el de *fvSolution* se definen ciertos parámetros relacionados con los algoritmos de la simulación, los factores de relajación utilizados y las tolerancias relativas y totales del caso.

A parte de estos ficheros se encuentran otros como:

- *fvOptions*, en el que se definen ciertos parámetros como por ejemplo limitadores para los campos que se van a estudiar.
- *mapFieldsDict*, que se encarga de indicar a partir de un fichero, las regiones del archivo origen y las que equivaldrían del archivo destino y hace una copia de los campos del archivo origen al de destino haciendo una interpolación en la malla destino.
- *setFieldsDict*, se encarga de especificar a qué zona pertenece cada malla, se usa para cuando haya diferentes mallas identificar cuál es cuál por si cada una tiene una función diferente a la otra, por ejemplo en este caso una de las mallas es móvil.

2.3. Creación de la malla

La generación de la malla depende de la geometría que se desea estudiar y al dominio total que se analiza. En cuanto a la geometría, se suele diseñar mediante programas de CAD ya que en este caso el software OpenFOAM no dispone de diseñador CAD en sus herramientas, por lo que hay que importar la geometría de una fuente externa. Antes de importar las geometrías hay que diseñar la malla base o, en otras palabras, el dominio total del sistema que será la base del análisis. Para generar la malla, el software OpenFOAM posee dos herramientas que se explicarán a continuación (*blockMesh* y *snappyHexMesh*):

- *blockMesh*

La herramienta *blockMesh* crea mallas mediante parámetros y se genera con el fichero mencionado previamente *blockMeshDict*. Una vez se ejecuta la malla, se guarda en la carpeta *constant* en la subcarpeta *polyMesh*, especificando vértices, puntos, caras y celdas de la malla.

Se indican los bloques en los que se quiere dividir el dominio y para ello cada bloque debe estar formado por 8 vértices que constituyen un hexaedro. La enumeración de los vértices para formar cada bloque tiene que ser realizada con meticulosidad, ya que en función del sentido en el que se definan puede generar un error y es que la normal a dicha cara no sea la adecuada.

Seguidamente, se realiza una división del dominio en celdas hexaédricas en función de lo que se le ha especificado. La distribución de celdas en el bloque puede ser uniforme o puede ser definida por una serie de parámetros como, por ejemplo, que aumente en una dirección en función de un ratio que se le predetermina.

Por último, se enumeran los vértices que van a formar una cara y al conjunto de estas se le asigna un nombre [4].

- *snappyHexMesh*

La función que ofrece *snappyHexMesh* es la de generar mallas hexaédricas de carácter tridimensional. Además, dicha malla se puede formar en base a una geometría reconocida mediante un formato de estereolitografía (STL) aunque puede leer cualquier geometría en formato triangulado. Dicha malla inicial que se ha creado con la herramienta *blockMesh* se refina en torno a la geometría que se le ha importado mediante un procedimiento iterativo hasta que consigue reproducir la geometría.

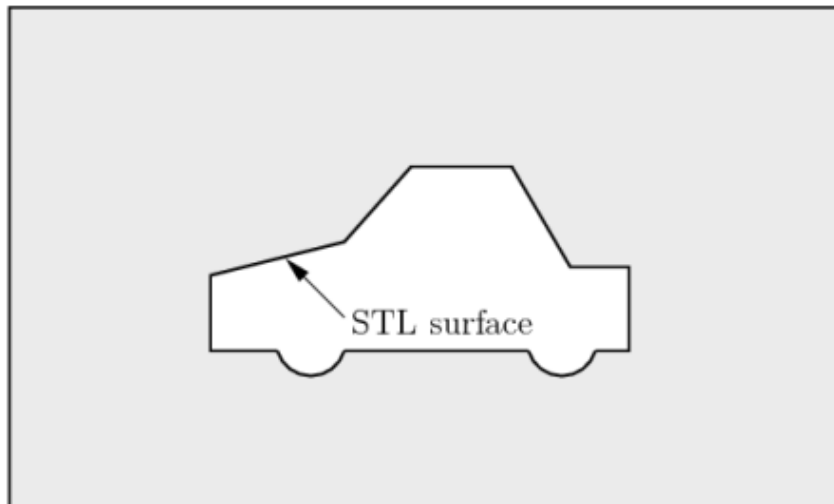


Figura 2.2: Inicio del mallado de una geometría con la herramienta *snappyHexMesh* [4].

El archivo posee internamente ciertas geometrías simples para el refinamiento, puede generar capas prismáticas en las cercanías de las geometrías y puede trabajar con superficies “sucias”, superficies irregulares de difícil diseño. Se trata de una herramienta muy robusta y con una gran manejabilidad de forma que es flexible a muchas modificaciones del usuario en base a sus requerimientos.

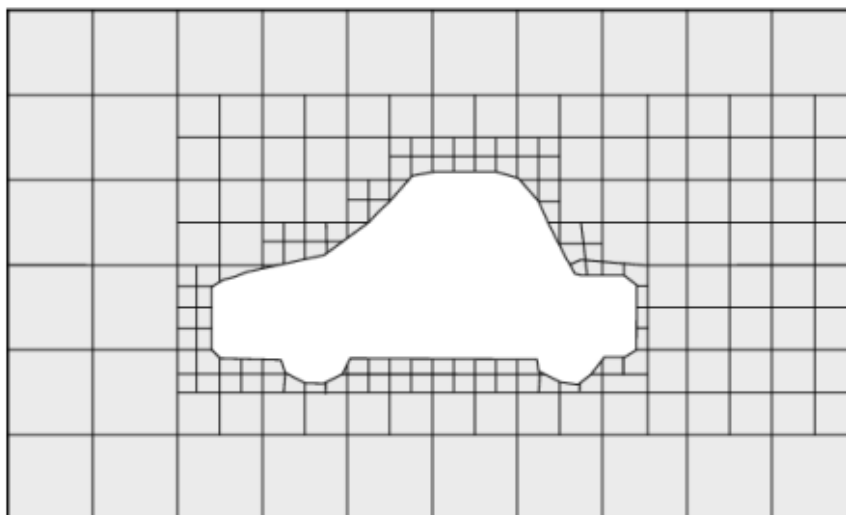


Figura 2.3: Final del mallado de una geometría con la herramienta *snappyHexMesh* [4].

La configuración donde se encuentra toda la información sobre el mallado se denomina *snappyHexMeshDict* como se ha descrito previamente y tiene cinco secciones principales [4]:

- *geometry*: definición de las superficies que se quieren introducir.

- *castellatedMeshControls*: se encarga de que desde cualquier malla hexaédrica se realiza un refinamiento y en el caso de que se esté realizando en paralelo, se hace un balance de carga de las celdas a crear en los procesadores.
- *snapControls*: garantiza la calidad de la malla mediante una adaptación de la malla a la geometría introducida.
- *addLayersControls*: se introducen capas límite prismáticas encogiendo la malla existente y creando el relleno que conformarían dichas capas, siempre respetando las restricciones de la calidad de la malla.
- *meshQualityControls*: configuración de la calidad de la malla que se impone durante las fases existentes en el *snappyHexMesh*.

2.4. Modelos matemáticos

2.4.1. Equilibrio de ecuaciones

La dinámica de fluidos computacional está basada en el balance de las ecuaciones fundamentales. Las tres ecuaciones elementales de conservación son la ecuación de continuidad de la masa, la ecuación de la conservación del momento y la ecuación de la conservación de la energía.

Asumiendo que el fluido es homogéneo y Newtoniano el equilibrio de las ecuaciones se puede expresar de la siguiente manera:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \quad (2.1)$$

$$\left(\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_j u_i}{\partial x_j} \right) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho f_i \quad (2.2)$$

$$\left(\frac{\partial \rho e}{\partial t} + \frac{\partial \rho u_j e}{\partial x_j} \right) = -\frac{\partial p u_j}{\partial x_j} + \frac{\partial u_i \tau_{ji}}{\partial x_j} + \frac{\partial}{\partial x_j} \left(\lambda \frac{\partial T}{\partial x_j} \right) + S_E \quad (2.3)$$

donde ρ es la densidad, t es el tiempo, x es la coordenada espacial, u es la velocidad del flujo, p es la presión, f_i es la fuerza del cuerpo por unidad de masa que se ejerce sobre el fluido, S_E es el término de energía y el tensor de estrés viscoso es:

$$\tau_{ij} = 2\mu S_{ij} - \frac{2}{3}\mu \frac{\partial u_k}{\partial x_k} \delta_{ij} \quad (2.4)$$

donde el tensor de velocidad de deformación es:

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.5)$$

2.4.2. Modelo de turbulencia $k - \omega$ SST

El término más conflictivo de las ecuaciones previas resulta ser el término convectivo que, además, es el que representa la turbulencia. Si se estuviera simulando con un flujo laminar todo el campo de la velocidad sería estable, en cambio, al sobrepasar cierto número de Reynolds se suceden pequeñas perturbaciones en el flujo que deja paso a un flujo desordenado.

El flujo turbulento se caracteriza básicamente por fluctuaciones caóticas y la representación de la turbulencia aún presenta un problema importante. Se puede simular la turbulencia mediante un sistema DNS (*Direct Numerical Simulation*) en el que no se realizan hipótesis de ningún tipo y se resuelven todas las escalas del flujo pero supone un coste computacional demasiado elevado.

Por otro lado, también se pueden simular escalas de las fluctuaciones con LES (*Large Eddy Simulation*) en el que se realiza un filtrado espacial en el subintervalo inercial. Se reduce dicho coste computacional al ignorar las escalas pequeñas mediante un filtrado de paso bajo en las ecuaciones de Navier-Stokes. Este filtrado se puede ver como una media espacio-temporal que elimina la información de las pequeñas escalas de la solución numérica.

De todos modos, ambos métodos son computacionalmente muy costosos y no son prácticos para la industria. Existe otro modelo turbulento llamado *Reynolds Averaged Navier-Stokes* (RANS) en el que se necesitan muchas menos celdas. En este modelo, se asume que la dependencia temporal en las fluctuaciones de la velocidad turbulenta se puede separar de la velocidad media del flujo. De esta manera, se introduce una serie de incógnitas llamadas esfuerzos de Reynolds que son función de la fluctuación de la velocidad y que requieren un modelo turbulento para poder cerrar el sistema de ecuaciones y que sea resoluble [20].

Existen diversos modelos de turbulencia como son: el $k - \omega$, el $k - \epsilon$ y el $k - \omega$ SST. En este caso se va a hacer uso del modelo $k - \omega$ SST debido a que combina las mejores características del modelo $k - \omega$ y $k - \epsilon$. Este último se aplicará para altos números de Reynolds .

El modelo $k - \epsilon$ se usa en la región del flujo medio, ya que el modelo $k - \omega$ es más sensible en las regiones de mezcla y externas a la capa límite. Este modelo consiste en la resolución de dos

ecuaciones: para el transporte de energía cinética turbulenta k y para la tasa de disipación de la energía cinética turbulenta ϵ . En cuanto al flujo cerca de las paredes, se modela mediante el uso de funciones de pared, por lo tanto, no es simulado [17].

En cambio, el modelo $k - \omega$ de Wilcox [20] es implementado en la subcapa viscosa y en la capa logarítmica y no necesita funciones de pared aunque se pueden implementar. Resulta ser mucho más preciso y robusto en la capa viscosa que el modelo $k - \epsilon$ al igual que en la zona logarítmica. También es mejor para los gradientes de presión, para flujos compresibles y para flujos cerca de paredes sólidas [17].

En cuanto al modelo $k - \omega$ SST, se implementa una función para la viscosidad, la cual mejora la precisión de la resolución de los flujos con fuertes gradientes de presión adversa. Las ecuaciones de transporte para el modelo $k - \omega$ SST para k y ω son [7]:

$$\frac{\partial \rho k}{\partial t} + \frac{\partial \rho k u_i}{\partial x_i} = P_k - \beta^* \rho k \omega + \frac{\partial}{\partial x_i} \left[(\mu + \mu_t \sigma_k) \frac{\partial k}{\partial x_i} \right], \quad (2.6)$$

$$\begin{aligned} \frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho \omega u_i}{\partial x_i} = & \alpha_{SST} \frac{\rho P_k}{\mu_t} - \beta \rho \omega^2 + \frac{\partial}{\partial x_i} \left[(\mu + \mu_t \sigma_{\omega 1}) \frac{\partial \omega}{\partial x_i} \right] \\ & + 2(1 - F_1) \frac{\rho \sigma_{\omega 2}}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}, \end{aligned} \quad (2.7)$$

donde k es la energía cinética turbulenta, ω es el ratio de disipación, F_1 es una primera función de mezcla y $\alpha_{SST} = \alpha_1 F_1 + \alpha_2 (1 - F_1)$. La viscosidad turbulenta de eddy se define como:

$$\mu_t = \frac{\rho a_1 k}{\max(a_1 \omega, S F_2)}, \quad (2.8)$$

donde F_2 es la segunda función de mezcla y S es el invariante de deformación. Los coeficientes empíricos del modelo definidos por Menter [20] son los siguientes:

α_1	α_2	$\alpha_{\omega 1}$	$\alpha_{\omega 2}$	β_1	β_2	β^*	a_1
0.85	1.0	0.5	0.856	0.075	0.0828	0.09	0.31

Tabla 2.1: Tabla de coeficientes del modelo de turbulencia $k - \omega$ SST

Ley de pared y capa límite

Para números de Reynolds bajos el modelo $k-\omega$ es necesario que la primera celda tras la pared se encuentre en la subcapa viscosa, ya que en esta subcapa es donde los esfuerzos viscosos dominan el comportamiento turbulento. La distancia a la que tiene que encontrarse se define como una distancia adimensional desde la pared, que se puede calcular de la siguiente forma:

$$y^+ = \frac{yu_\tau}{\nu} \quad (2.9)$$

donde y es la distancia desde la pared u_τ es la velocidad de cizalladura y ν es la viscosidad cinemática.

En la subcapa viscosa el perfil de velocidad se muestra a través de una relación lineal con la distancia a la pared, por lo que la primera celda en esta capa debe satisfacer el requisito mínimo de $y^+ < 5$ y preferiblemente $y^+ \approx 1$.

Para satisfacer este requisito, es muy probable que se necesite un gran número de celdas. Usualmente se suele optar por situar esta primera celda directamente en la región logarítmica donde el perfil de velocidad muestra una relación logarítmica con la distancia a la pared. En esta capa la distancia adimensional debe encontrarse entre $100 < y^+ < 0.2 \delta_{99}$. Existen ciertas funciones de pared que incorporan los efectos viscosos de la subcapa viscosa. Dichas funciones de pared se aplican en la capa logarítmica [7].

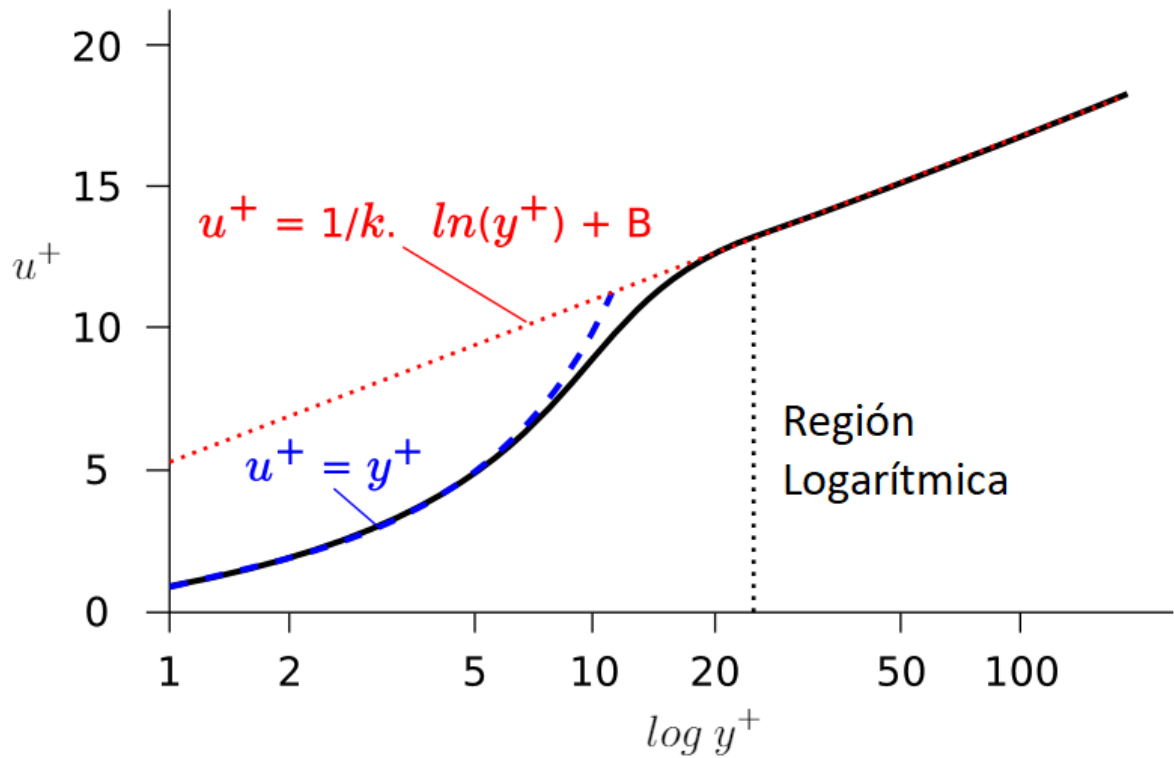


Figura 2.4: Distribución de la velocidad en la proximidad de una pared.

Además de lo explicado previamente, una buena elección donde no se puede mantener constante el y^+ en la capa logarítmica es utilizar la función de pared de Launder-Spalding que es continua y funciona tanto en la subcapa viscosa como en la logarítmica donde el y^+ varía entre 5 y 30. Dicha función de pared se muestra de la siguiente forma:

$$y^+ = u^+ + \frac{1}{E} \left[e^{\kappa u^+} - 1 - \kappa u^+ - 0.5(\kappa u^+)^2 - \frac{1}{6}(\kappa u^+)^3 \right] \quad (2.10)$$

donde u^+ es la velocidad adimensional y κ es la constante de von Karman.

2.5. Métodos numéricos

2.5.1. Método de volúmenes finitos

El método de volúmenes finitos (MVF) es el método numérico mayormente utilizado en la dinámica de fluidos computacional a la hora de resolver las ecuaciones diferenciales que se plantearon en los modelos numéricos.

Consiste en la división del dominio de cálculo en un número finito de volúmenes de control que no se solapan, por lo que cada nodo se encuentra rodeado de uno de estos volúmenes de control y se resuelve la integral de la ecuación diferencial sobre estos. El resultado de dicha integral es la ecuación discretizada con los valores de una variable dependiente para los nodos de la malla [2].

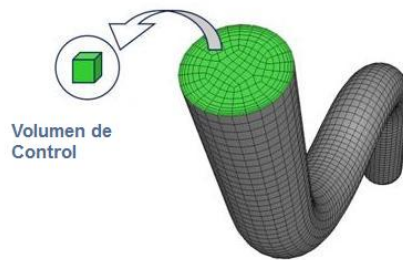


Figura 2.5: Dominio discretizado en volúmenes de control [3].

Es un método comúnmente utilizado en aplicaciones CFD ya que la solución que se obtiene asegura de manera implícita la conservación de las ecuaciones de masa, movimiento y energía para todo volumen de control y, por lo tanto, para todo el dominio en conjunto.

Para obtener los valores en los nodos de la malla se tienen que resolver las ecuaciones discretizadas. En el método de los volúmenes finitos solo se considera como solución aquellos valores de la variable dependiente de los nodos que se han fijado, sin tener en cuenta los valores de esta misma variable para otros nodos. Únicamente se toman métodos de interpolación como funciones auxiliares para poder evaluar otras integrales dentro de las ecuaciones estipuladas [2].

2.5.2. Algoritmos de solución del acoplamiento presión - velocidad

Al tratar de resolver las ecuaciones de Navier-Stokes se observa que existe un término de presión en cada una de las tres ecuaciones en las que se tiene que realizar el equilibrio, por lo

tanto, no existe una ecuación independiente para la presión.

Para ello, el campo de la presión se define indirectamente en la ecuación de la continuidad, de manera que cuando el campo de la presión sea el correcto y se sustituya en la ecuación de cantidad de movimiento, el campo de velocidad que resulte será el correcto para cumplir la ecuación de la continuidad [2].

Para que dichos resultados sean correctos y al carecer de un campo de presiones conocido, se debe realizar un procedimiento iterativo en el que mediante las ecuaciones de la continuidad y del movimiento vayan convergiendo hacia la solución real. Este acoplamiento de presión-velocidad se puede resolver mediante ciertos algoritmos como el SIMPLE, PISO o PIMPLE.

SIMPLE (*Semi-Implicit-Method-of-Pressure-Linked-Equations*) es el algoritmo presión-momento más antiguo y se usa para análisis estacionarios. La derivación en el tiempo no está implementada y no es consistente debido a la falta de un término de la presión. Debido a las características del algoritmo y, a que no existe un paso temporal físico, no hay un limitador para la solución, lo que significa que las ecuaciones tienen que tener un factor de relajación. Este factor representa la medida en que se ve afectada la variable obtenida con respecto a la que se utiliza en la siguiente iteración, y en este caso tiene que ser menor que uno, lo que indica una subrelajación para no llegar a una solución inestable. La subrelajación hace que la convergencia se ralentice y, por consiguiente, una buena estimación es importante para una convergencia y estabilidad óptima [16] [7].

PISO (*Pressure-Implicit-of-Split-Operations*) es el algoritmo que se usa en simulaciones transitorias donde la precisión temporal es muy importante. La derivación en el tiempo se encuentra incluida y el acoplamiento presión-momento es consistente. Se introduce el número de *Courant-Friedrichs-Levy*, que es un parámetro adimensional que contiene información de lo veloz que viaja la información a través de las celdas. El número de *Courant* se define de la siguiente forma:

$$Co = \frac{U \Delta t}{\Delta x} \quad (2.11)$$

donde U es la velocidad local en la celda, Δt es el paso temporal y Δx es la distancia entre los centros de las celdas [7].

El algoritmo PIMPLE es una mezcla del PISO y del SIMPLE. Todos estos algoritmos son iterativos, el PIMPLE y el PISO se usan para casos transitorios y el SIMPLE para casos estacionarios. La mejor de entender el algoritmo PIMPLE es imaginarlo como si fuera un algoritmo SIMPLE para cada paso temporal, donde los correctores externos son las iteraciones.

Una vez haya convergido este paso se pasará al siguiente paso temporal hasta que la solución sea correcta. Una mejor estabilidad se consigue en el PIMPLE gracias al PISO, especialmente cuando se tienen grandes pasos temporales, donde el máximo número de *Courant* es mayor que 1 o cuando la solución es inestable. Además de la estabilidad, se puede definir los correctores externos y los internos [9].

Correctores Externos

El número de correctores externos en el algoritmo PIMPLE, como se ha mencionado anteriormente, especifica cuantas iteraciones externas se deben realizar. Esto significa cuantas veces el sistema de ecuaciones es calculado antes de que se fuerce a pasar al siguiente paso temporal, independientemente de si el paso temporal actual ha convergido o no [9].

Este número no significa que se tengan que realizar todas las iteraciones, sino que simplemente iterará hasta que la convergencia en ese paso temporal se cumpla. Esto hace que el algoritmo sea más rápido cuando sea estable pero permite que realice un mayor número de iteraciones cuando existan inestabilidades. El criterio para que se cumpla la convergencia en el paso temporal se define como la tolerancia absoluta del *solver*.

Otro aspecto a remarcar es que se realizarán los cálculos con factores de relajación hasta que se llegue a la última iteración. Esto significa que si se han definido 5 iteraciones, se realizarán 4 con factores de relajación y la última sin ellos.

Correctores Internos

El número de los correctores internos es el número de veces que la ecuación de la presión es corregida dentro de una iteración. Un número mayor de correctores no mejora en gran medida los resultados, por lo que se recomienda como máximo utilizar 3 iteraciones [9].

Estabilidad

La estabilidad de este algoritmo puede mejorarse reduciendo los factores de relajación. Con este paso la convergencia tardará más en alcanzarse y, por lo tanto, se deben añadir correctores externos. Puede ser beneficioso para evitar que la solución se encuentre con inestabilidades, donde el máximo número de *Courant* es probable que aumente [9].

Capítulo 3

Modelado de la capa límite

El punto de partida de este trabajo, como se ha comentado previamente, es un proyecto que se realizó anteriormente como trabajo de fin de máster en el que se estudiaba el análisis de las fuentes de ruido originadas en el lanzamiento del cohete VEGA. El estudio se realizó con la herramienta OpenFOAM que es la misma que se usará en este proyecto.

Del proyecto previo se utilizará tanto la geometría base del cohete y de las estructuras adyacentes como la malla sobre la que se hizo el análisis. Del mismo modo, se estudiará si las condiciones de contorno y los valores iniciales son los adecuados para este caso y se hará un estudio profundo para comprender el caso previo con el objetivo de avanzar, utilizándolo como punto de partida, usando la misma metodología.

En resumen, la geometría sobre la que se basaba era el cohete VEGA y la plataforma de lanzamiento que se compone por un deflector en forma de cuña y unos conductos que se utilizan para la evacuación de los gases.

En cuanto a condiciones iniciales, métodos numéricos y esquemas de discretización, cabe resaltar que el chorro de flujo que se origina en el motor cohete posee condiciones supersónicas. De igual modo, son relevantes y se utilizarán las mismas condiciones iniciales y métodos numéricos del estudio previo, ya que se pretende hacer una comparación con este. En cuanto a los refinamientos de la malla, se eligen unas geometrías de refinamiento diferentes a las del estudio previo. Se eligen estos refinamientos con el objetivo de que el número de celdas refinadas sea menor en el conjunto total pero mayor en las zonas de mayor interés del caso.

En este capítulo se verá uno de los subobjetivos comentado previamente, cómo se ha hecho posible la inclusión de la capa límite en la malla del estudio previo. Del mismo modo, tras especificar las herramientas necesarias y, una vez generada la capa límite en la malla, se iniciará la simulación del caso. Una vez finalizada la simulación, se podrán comparar los resultados obtenidos al haber introducido en el modelo esta capa límite con los del estudio previo.

3.1. Herramientas para capas límite

La última etapa a la hora de realizar un mallado es la adición de capa límite, donde una capa de celdas se añaden en la zona más cercana a una serie de superficie definidas. Esta etapa se controla mediante unos comandos que se encuentran en el diccionario denominado *addLayersControl*.

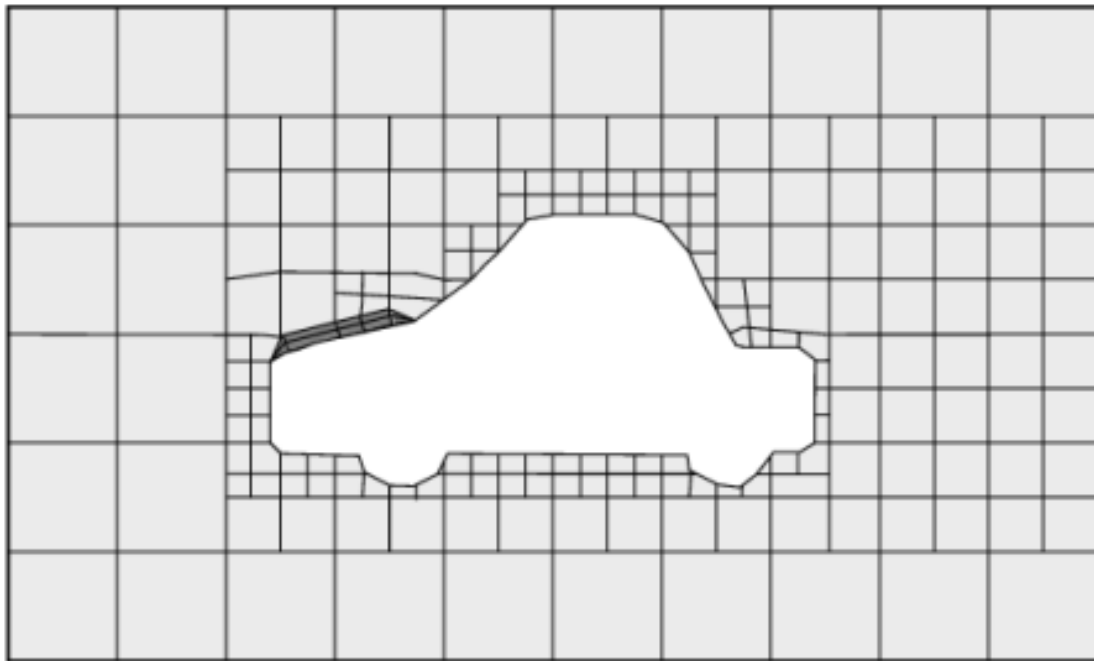


Figura 3.1: Adición de la capa límite mediante la herramienta *snappyHexMesh* [4].

Como se puede ver en la zona sombreada de la Figura.3.7 se han añadido una serie de capas de celdas perpendiculares a la superficie sobre la que se quiere realizar.

El proceso de adición de la capa en la malla implica reducir la malla existente en la zona del contorno de la superficie e insertar capas de celdas siguiendo el siguiente procedimiento:

1. La malla es proyectada hacia atrás desde la superficie por un espesor específico en la dirección normal a dicha superficie.
2. Se resuelve la relajación de la malla interna con los últimos vértices proyectados del contorno.
3. Se comprueba si los criterios de validación se satisfacen, en el caso de que no sea así, se reduce el espesor proyectado y se vuelve al paso anterior. En el caso de que la validación no pueda ser satisfecha para ningún espesor, no se insertan las capas.
4. Si los criterios de validación se cumplen, se insertan las capas.

5. En el último paso se comprueba de nuevo la malla; en el caso de que la validación falle se borran las capas y se vuelve al paso 2.

El procedimiento de adición de capas usa la configuración que se encuentra en el subdiccionario *addLayersControls* dentro del *snappyHexMeshdict*. El usuario tiene la opción de seleccionar entre cuatro parámetros diferentes de espesor de la capa: *expansionRatio*, *finalLayerThickness*, *firstLayerThickness* y *thickness*. De estos parámetros solo se deben definir dos, ya que si se definen más el caso se encuentra sobre definido y se produciría un error a la hora de realizar el proceso de mallado.

Las entradas del subdiccionario más relevantes a la hora de definir las capas y sus definiciones son las siguientes [5]:

- *layers*: este subdiccionario contiene diversas entradas para cada *patch*, que así es como se le llama a cada división de la geometría que conforma la totalidad de la misma, en el que las capas van a ser agregadas y se requiere conocer el número de capas a introducir. Se necesita conocer el nombre del *patch* en vez de la superficie de la región ya que la adición de capas se realiza en la malla existente y no en la superficie de la geometría.
- *relativeSizes*: se puede elegir entre que los espesores de la capa sean relativos al tamaño de celda sin distorsiones fuera de la capa o que sean absolutos.
- *expansionRatio*: se trata del factor de expansión para la capa que se va a mallar. Este factor es el ratio entre el tamaño de una celda y otra de la siguiente capa. Por ejemplo:

$$\frac{\Delta_2}{\Delta_1} = \frac{\Delta_3}{\Delta_2} = \frac{\Delta_4}{\Delta_3} = \frac{\Delta_5}{\Delta_4}$$

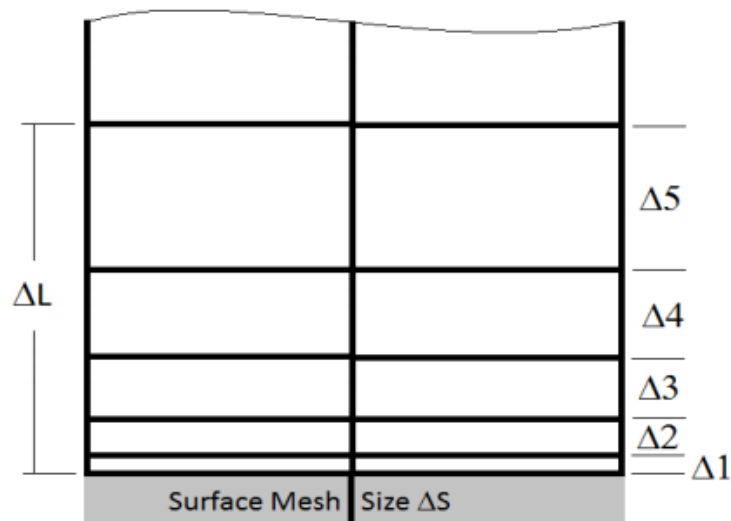


Figura 3.2: Representación del parámetro *expansionRatio* [5].

- *finalLayerThickness*: espesor de la capa más alejada a la pared. De normal se utiliza cuando la entrada *relativesSizes* es relativa.
- *firstLayerThickness*: a diferencia de la anterior, es el espesor de la capa más cercana a la pared. Por lo general, se utiliza cuando la entrada *relativesSizes* es absoluta.
- *thickness*: espesor total de todas las celdas de la capa. Normalmente se suele usar con tamaños absolutos en la entrada de *relativeSizes*.
- *minThickness*: mínimo espesor de las celdas de la capa, independientemente de la entrada *relativeSizes*. Por debajo de este espesor las capas se colapsarán automáticamente.
- *nGrow*: número de capas con caras conectadas que no crecen si los puntos no son extruidos. Este parámetro facilita la convergencia de la adición de las capas cerca de las figuras.

Otras entradas de menor importancia y sus valores típicos (aunque el usuario puede cambiarlos en función de la necesidad de su problema) son [5]:

- *featureAngle*: se define como el ángulo sobre el que las capas colapsan automáticamente. Cuanto mayor sea el valor, mejor se reproducirán las capas.

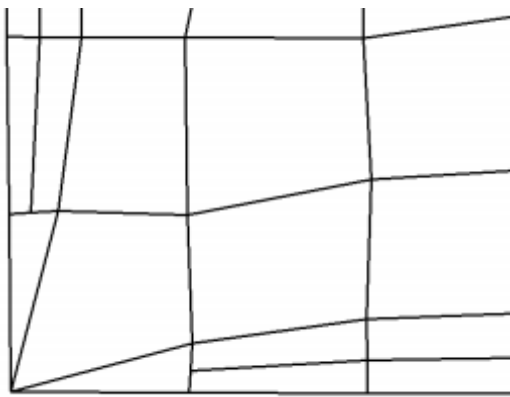


Figura 3.3: Representación de *featureAngle* = 45° [5].

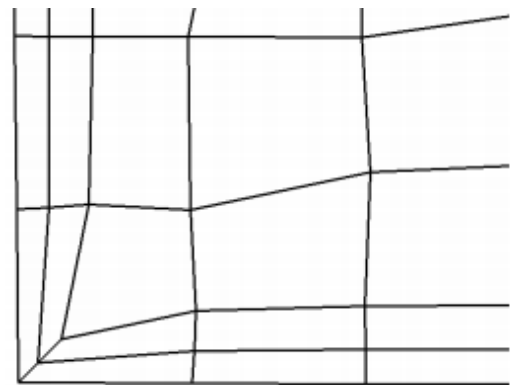


Figura 3.4: Representación de *featureAngle* = 180° [5].

- *nSmoothSurfaceNormals*, *nSmoothNormals*, *nSmoothThickness*: El suavizamiento se puede dar en las normales de los puntos de la superficie (*nSmoothSurfaceNormals*), en el espesor de la capa (*nSmoothThickness*) y en el campo interno de desplazamiento (*nSmoothNormals*). Usualmente se usan 1, 10 y 3 iteraciones respectivamente.

- *minMedianAxisAngle*: se utiliza este ángulo para captar puntos del eje medial a la hora de mover la malla fuera de la superficie. Un valor típico es 90 °.
- *maxThicknessToMedialRatio*: reduce el crecimiento de la capa donde el ratio del espesor a la distancia medial es grande. Generalmente se usa un valor de 0.3.
- *maxFaceThicknessRatio*: se usa para identificar caras deformadas y capas acabadas en esas caras. Habitualmente se usa un valor de 0.5.
- *nLayerIter*: si la iteración de las capas no ha convergido después de un cierto número de iteraciones, se sale del bucle de agregación de capa con la malla que se haya generado hasta dicha iteración. De normal se suelen utilizar 50 iteraciones, pero este es un valor que se puede modificar con facilidad si el usuario considera que con menos iteraciones cumple sus requisitos para la malla.
- *nRelaxedIter*: máximo número de iteraciones después de que los controles en el subdiccionario *relaxed* del *meshQuality* son utilizados. El valor típico para este número de iteraciones es 20.
- *nRelaxIter*: controla el número de reducción de iteraciones durante la etapa de reducción de errores. Típicamente se usan 5 iteraciones.

Además de especificar las entradas en el subdiccionario de *addLayersControl* la definición de la calidad de la malla es un paso prioritario tanto para el mallado de la geometría como para el de la capa límite. Si no se tienen los parámetros bien definidos es posible que las iteraciones a la hora de hacer el mallado no sean las adecuadas y la malla resultante sea de mala calidad, lo que resultaría nefasto para el cálculo del problema.

Existen bastantes parámetros a definir para validar la calidad de la malla, pero solo se van a definir aquellos que si no se cumplen pueden hacer que el cálculo tenga problemas de estabilidad y llevarlo a la divergencia [5].

- *maxNonOrtho*: la ortogonalidad de las caras se calcula como el producto escalar normalizado del vector del área de la cara con un vector desde el centroide de la cara hasta el centroide de la cara adjunta.

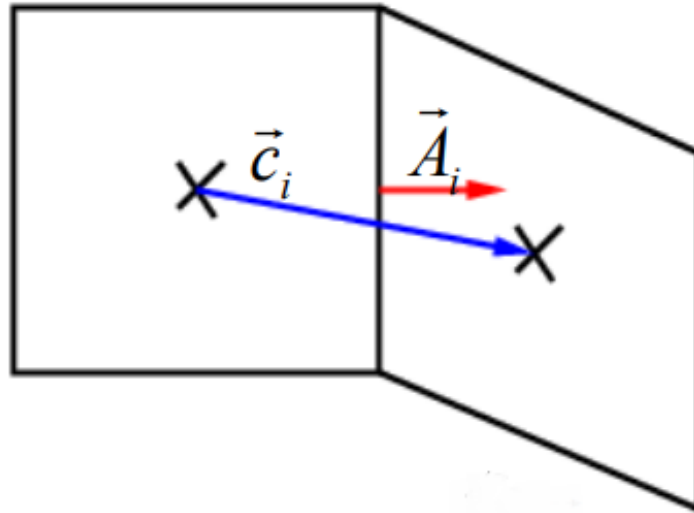


Figura 3.5: Representación de *maxNonOrtho* [5].

- *minVol*: evalúa el mínimo volumen de cara de las celdas.
- *minDeterminant*: el determinante de la celda se calcula tomando el determinante del tensor calculado por los vectores del área de las caras.
- *minFaceWeight*: el peso de las caras se calcula como el mínimo valor entre la distancia proyectada del centro de la celda a la cara y la distancia del centro de la celda vecina a la cara común, dividido entre la suma de ambas distancias.
- *minVolRatio*: el mínimo ratio del volumen se calcula como el ratio entre el mínimo volumen de la propia celda o de la vecina y el máximo volumen de las dos.

3.1.1. Cálculo del tamaño de la capa límite

Para calcular la primera celda en la pared en nuestro caso, se debe conocer primeramente el tensor cortante en la pared.

$$\tau = \frac{1}{2}C_f\rho U^2 = 5.02 \times 10^3 \quad (3.1)$$

donde C_f es un coeficiente de fricción, ρ es la densidad del aire y U es la velocidad del chorro que se establece como 2744 m/s. El coeficiente de fricción C_f se obtiene empíricamente, pero para flujos externos se puede calcular mediante el número de Reynolds mediante la siguiente ecuación:

$$C_f = 0,058Re^{-0.2} = 1.13 \times 10^{-3} \quad (3.2)$$

donde el Re es de 3.544×10^8 mediante la ecuación $Re = \frac{\rho UD}{\mu}$

Una vez obtenidos los parámetros necesarios se pueden obtener la velocidad de fricción haciendo uso del tensor de cortante y de la densidad de la siguiente forma:

$$u_\tau = \sqrt{\frac{\tau}{\rho}} = 64.52 \text{ m/s} \quad (3.3)$$

De esta forma y mediante la Ecuación 2.9 se obtienen los valores entre los que tiene que medir la primera celda de la capa límite. El tamaño debe estar entre 2.37×10^{-5} y 7.1×10^{-6} m.

Debido a que el refinamiento cerca de la pared no es elevado y a que se quiere mantener la malla del estudio previo, el tamaño máximo de primera celda introducido es de $4.63E-4$ m. Este es el tamaño máximo de celda que permite cumplir los requisitos de calidad de la malla y poder generar la capa límite al completo.

3.2. Discusión de los resultados

A continuación, una vez ya definidos todos los parámetros que intervienen en la creación de la última etapa del mallado, se observará cuál ha sido el resultado final en la malla. Se procederá a observar la mejora de la malla en cuanto a refinamientos y a la adición de la capa límite. Por último, se realizará una comparación con el estudio previo para observar las mejoras del caso.

3.2.1. Refinamiento de la malla

El refinamiento de la malla es un paso importante, ya que en función de los tamaños de las celdas la resolución del caso se realizará con mayor o con menor precisión.

Para el caso que se está estudiando es muy importante tener un tamaño de celdas adecuado cerca de las paredes, sobretodo en el deflector, ya que en este punto es donde el chorro que proviene del cohete interactúa con mayor intensidad.

En primer lugar, el refinamiento propuesto es una zona central que corresponde al lugar por donde el chorro se desplazará y cuyo tamaño de celda es muy reducido. Esta es una de las zonas más delicadas, ya que en esos puntos se producen muchos cambios bruscos de forma casi inmediata. En el caso de no refinar bien esta zona se podría perder información de alto valor y no representaría la solución real del caso. El tamaño de celda en esta zona es de 3.9 cm.

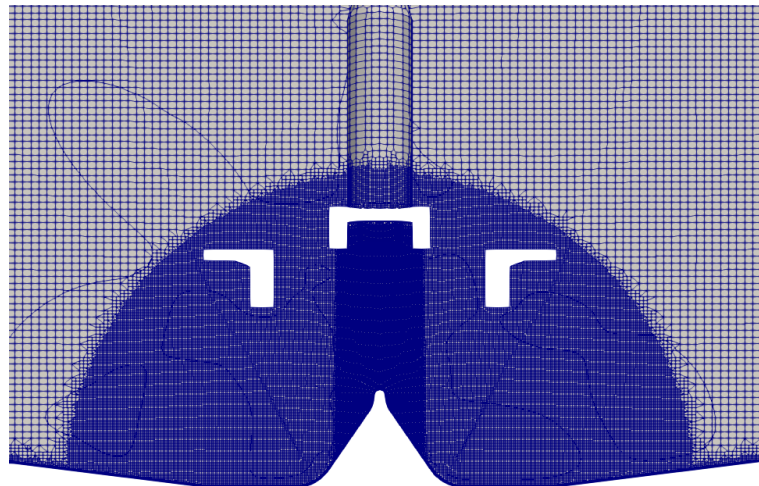


Figura 3.6: Representación del refinamiento central donde se va a propagar el chorro.

En comparación con el estudio previo, el refinamiento se ha realizado de manera diferente. En este caso, se ha optado por realizar un refinamiento de forma esférica, hasta el punto de captar los conductos de evacuación de los gases, y otra zona, hasta aproximadamente la zona central del cohete. Con este cambio se busca la optimización del mallado, de forma que se evita refinar ciertas celdas que no tienen relevancia en la solución pero sí deben formar parte del dominio.

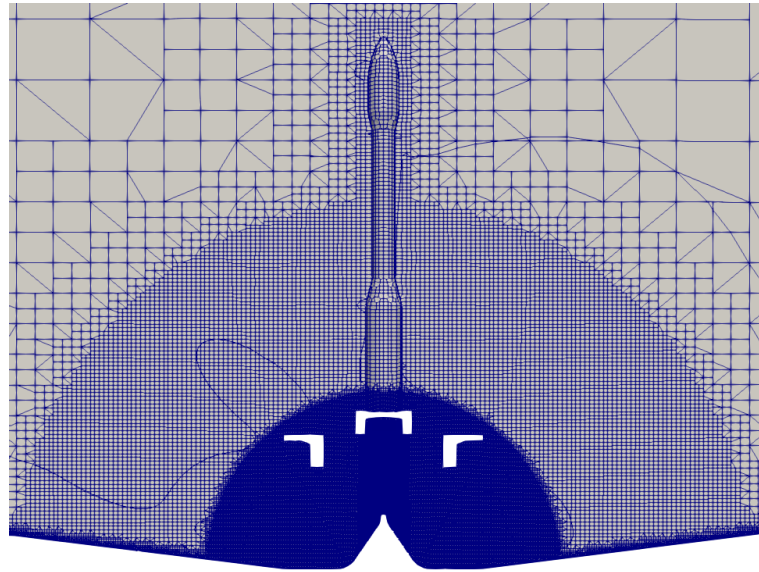


Figura 3.7: Representación del refinamiento esférico y gradual.

Por último, destacar la introducción de la capa límite en la superficie del deflector. Se han introducido 9 capas de celdas con un ExpansionRatio de 1.2. Además, para mantener la continuidad de las celdas, es decir, que la última celda de la capa tenga un tamaño similar a la celda de refinamiento siguiente, se ha definido que el tamaño de la primera capa de celdas sea de $4.63E-3$ m, cuyo cálculo se plantea en 3.1.1.

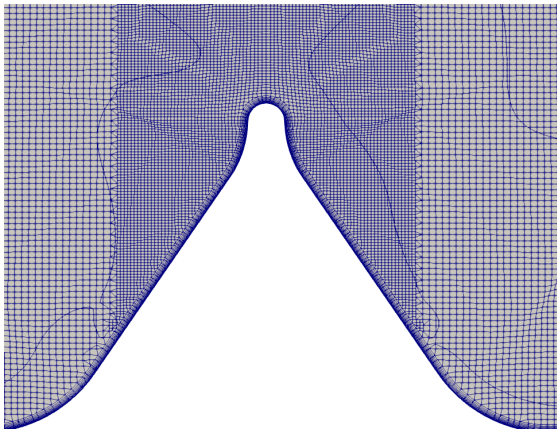


Figura 3.8: Representación de la capa límite en comparación con el refinamiento del chorro.

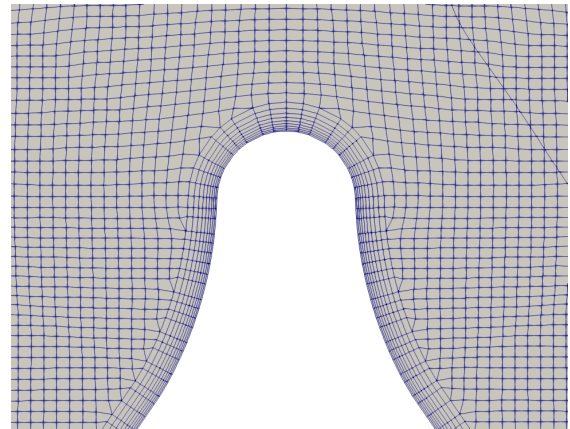


Figura 3.9: Visualización de las capas de celdas de la capa límite.

En la Figura 3.10 se puede ver el mallado final que se ha realizado. En la imagen superior izquierda se puede observar cómo el refinamiento del cohete en la parte superior y en la inferior, en la zona del inlet, es diferente. Este fenómeno se puede apreciar de manera ampliada en la imagen de la derecha. Este aumento de refinamiento en la zona inferior se debe a que es donde se producirá la emisión del chorro por la tobera y, por lo tanto, el refinamiento tiene que ser mayor para poder captar la máxima información posible.

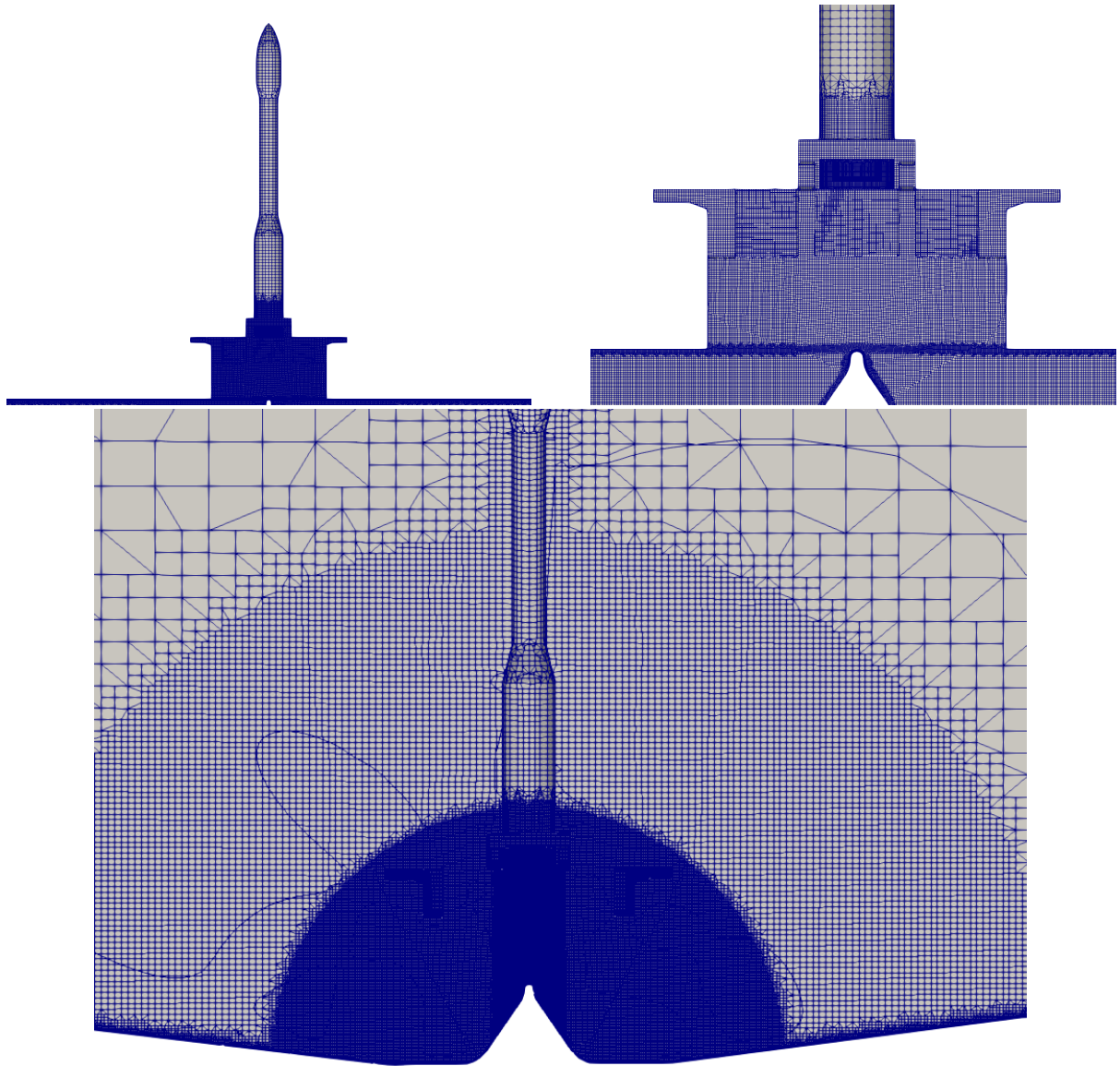
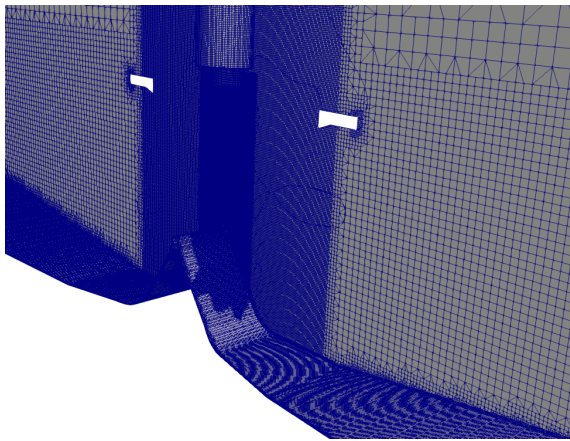
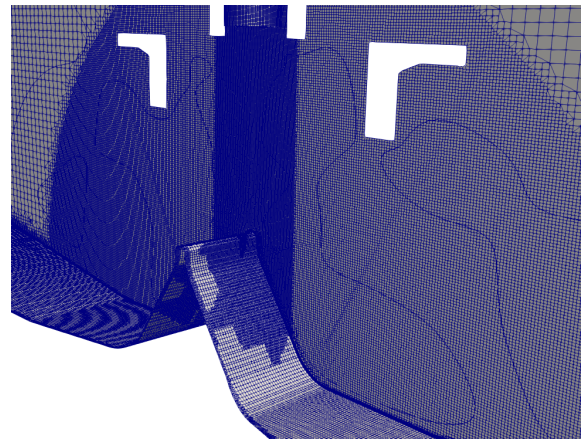


Figura 3.10: Representación del mallado realizado.

A continuación, se hará una comparación entre las mallas de este estudio y del estudio previo para ver la diferencia y la mejora del mallado.

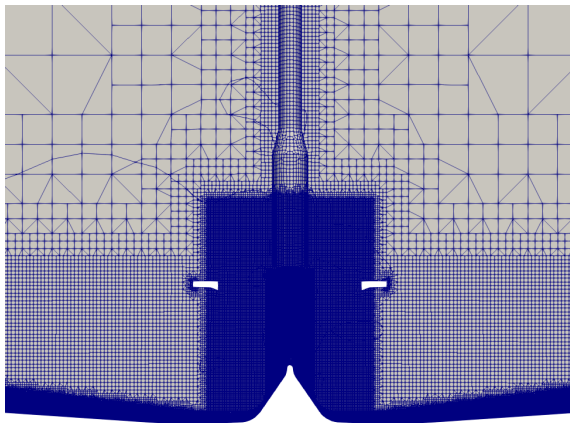


(a) Estudio previo.

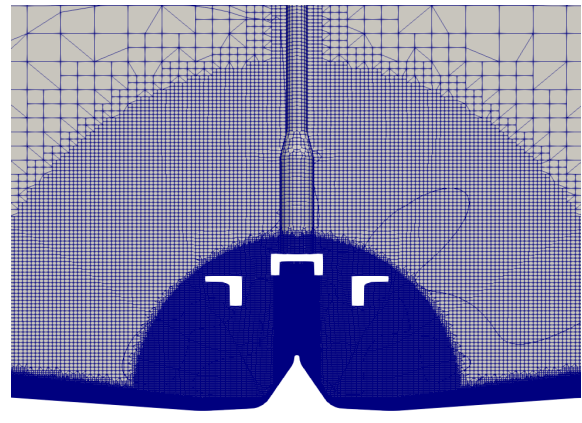


(b) Proyecto actual.

Figura 3.11: Representación del mallado de las zonas más relevantes del caso.

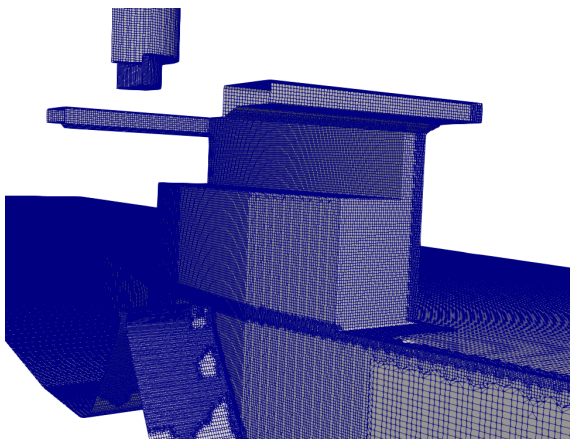


(a) Estudio previo.

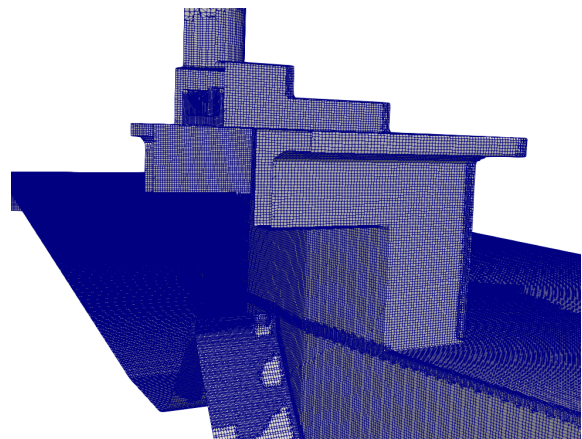


(b) Proyecto actual.

Figura 3.12: Representación del mallado total.



(a) Estudio previo.



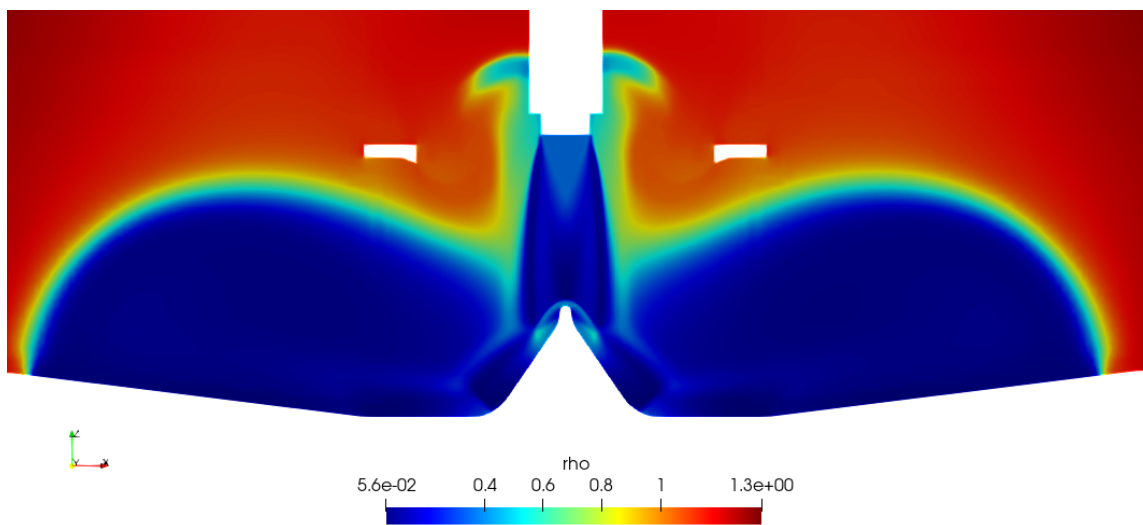
(b) Proyecto actual.

Figura 3.13: Representación de las plataformas de lanzamiento implementadas.

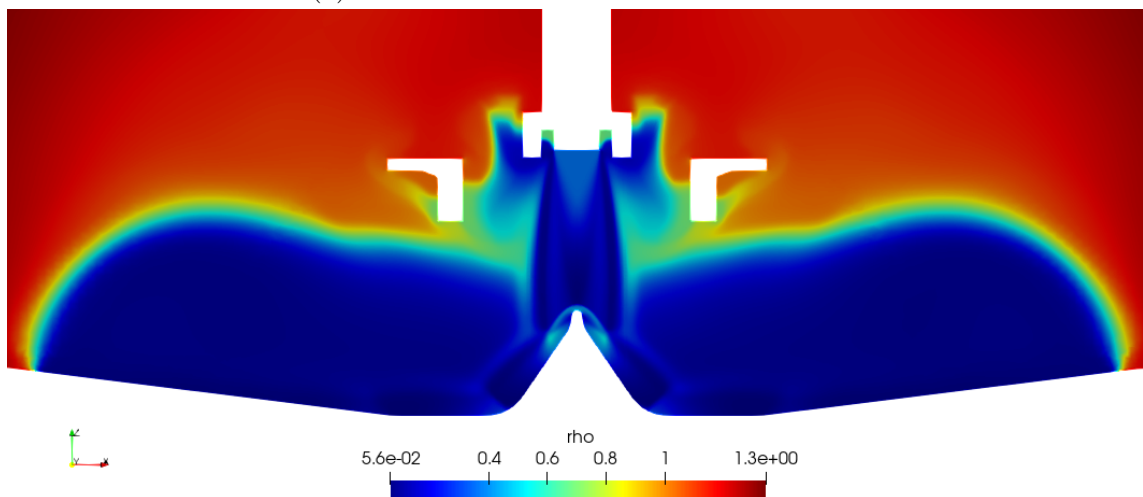
3.2.2. Comparación de los resultados

Influencia de la estructura de lanzamiento

Cabe analizar cuál es el efecto de la estructura de lanzamiento en el caso. Al no tener una geometría específica por la ESA, se ha decidido estudiar el impacto que podría llegar a tener la geometría de esta estructura. Esta geometría se ha conseguido acotar con mayor exactitud en este caso que en el estudio previo.



(a) Cohete sin estructura de lanzamiento.



(b) Cohete con estructura de lanzamiento.

Figura 3.14: Representación de las plataformas de lanzamiento implementadas.

Se puede observar cómo en estos campos de la densidad se aprecia una diferencia notable entre una geometría y otra. En el estudio previo se visualiza cómo se mueve una onda sobre el cohete que proviene del chorro, en cambio, en la figura de la derecha, la del caso actual, se observa como ese mismo flujo se encuentra con la plataforma de despegue, teniendo que

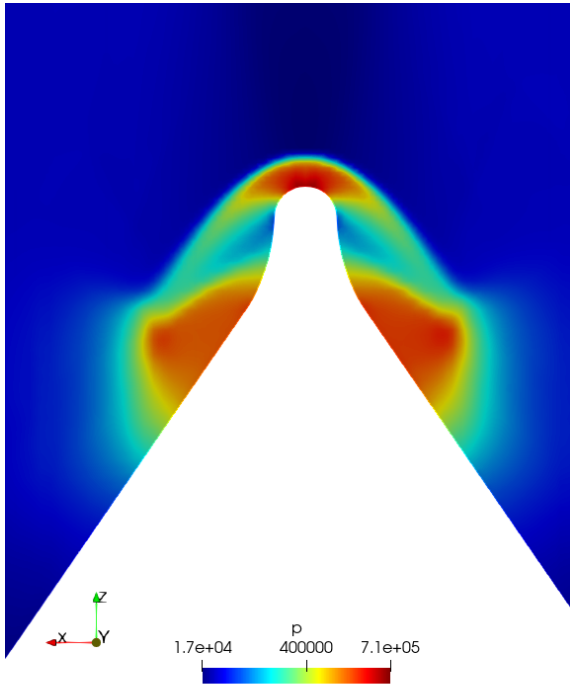
evitarla y generando otro tipo de perturbaciones. Estas también llegan a influir sobre el chorro en cierto modo.

Influencia de la capa límite

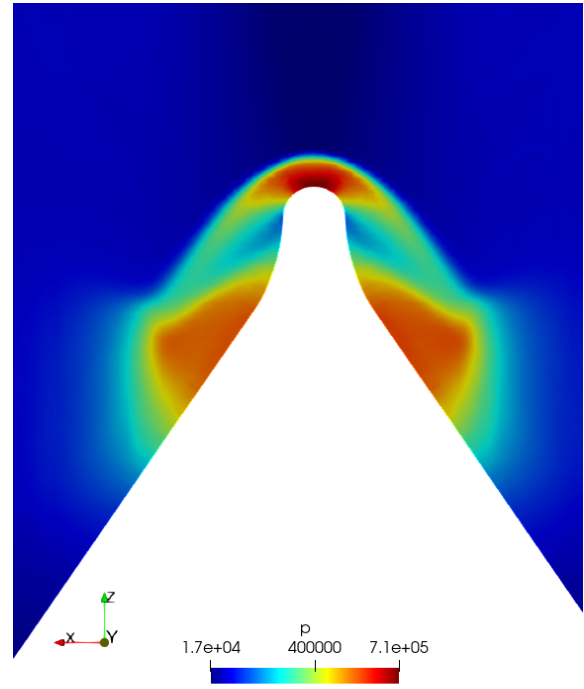
Para poder analizar la influencia de la capa límite sobre los resultados, se debe hacer una comparativa con el estudio previo. En dicho estudio se hacía referencia a dos mallas con diferentes refinamientos, una de 10 millones de celdas y otra de 60 millones. El resultado mostraba cómo la malla de mayor número de celdas representaba con mayor exactitud la realidad. Esta afirmación es debida a que las celdas que se encuentran en los bordes de las geometrías tienen un tamaño menor y, por lo tanto, se acercan más al tamaño de celda necesario para poder simular los efectos reales.

En este caso se agregó la capa límite, donde la primera celda tiene un tamaño de $4.63E-3$ m, que a diferencia de la malla de 60 millones de celdas, cuya celda más cercana a la geometría del deflector era de $1.95E-2$ m, puede plasmar con mayor exactitud el efecto real que ocurre en la pared del deflector. La malla de este caso consigue representar con mayor precisión el flujo en las paredes, ya que las funciones de pared son capaces de modelar de manera más efectiva la transición entre flujo laminar y flujo turbulento.

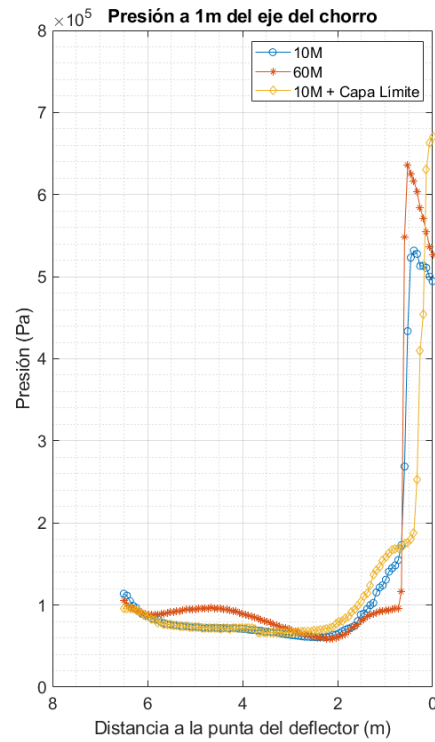
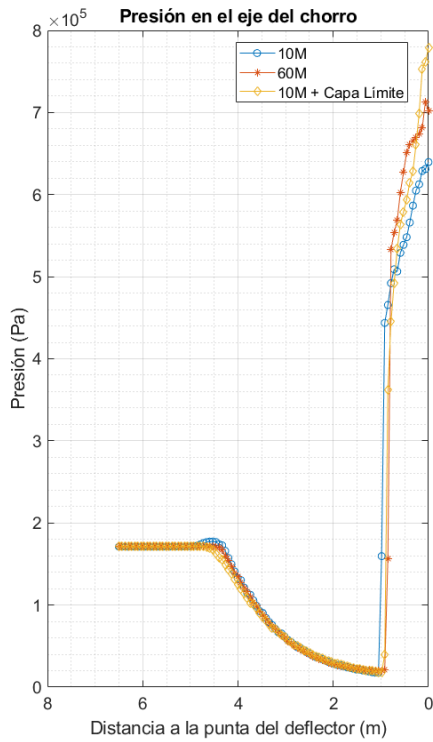
Como se observa en la Figura 3.15c el modelo de malla con 10M de celdas y la capa límite representa con mayor calidad la onda de choque producida en el deflector. Muestra una mejor representación del campo de la presión que la malla de 60M de celdas y, claramente mejor, que la de 10M. En conclusión, se puede afirmar que es más efectivo aplicar una capa límite sobre las superficies de pared que aumentar el número de celdas y, por consiguiente, reducir el tamaño de la celda. Con esta adición de la capa límite se ahorra mucho cálculo computacional al reducir el número de celdas necesarias para reproducir los campos lo más realista posible.



(a) Deflector sin capa límite.

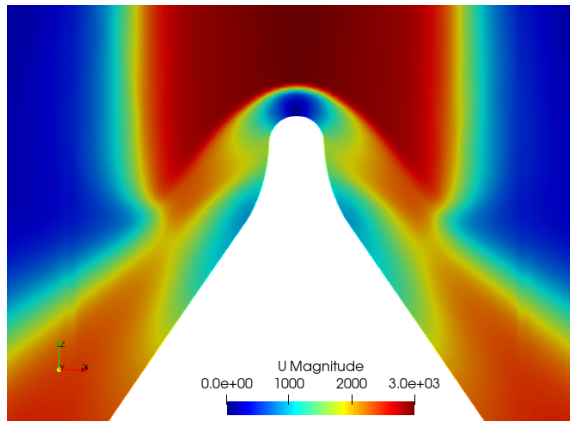


(b) Deflector con capa límite.

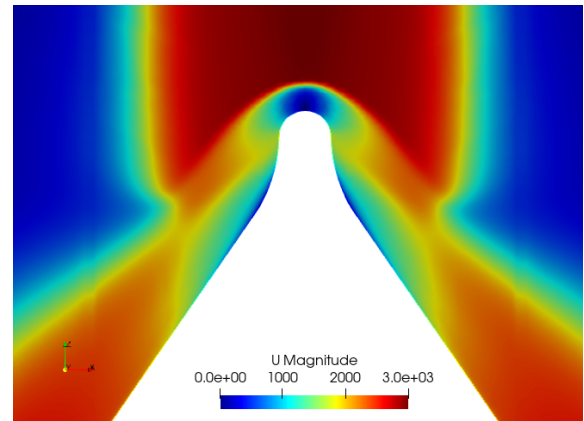


(c) Estudio de la presión a lo largo del eje del chorro y a 1 m entre la tobera y el deflector.

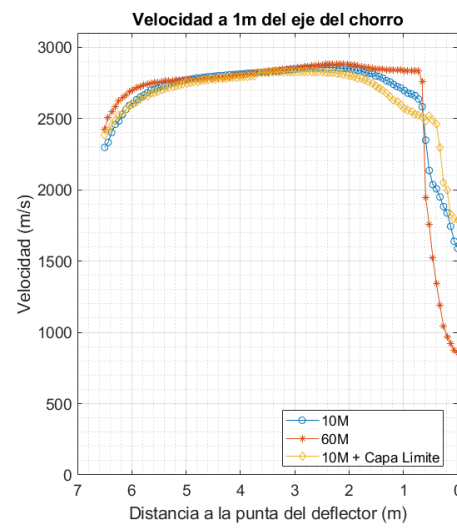
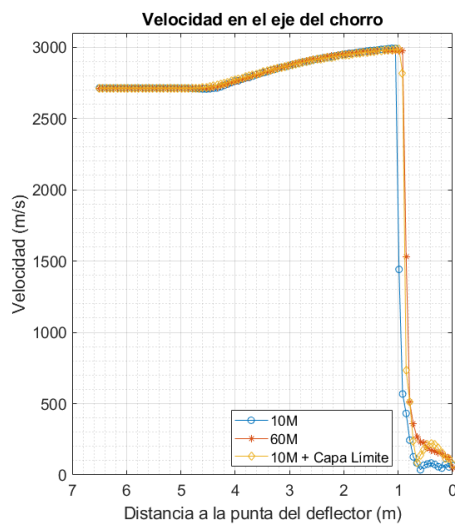
Figura 3.15: Representación del campo de la presión en el dominio del deflector.



(a) Deflector sin capa límite.



(b) Deflector con capa límite.



(c) Estudio de la velocidad a lo largo del eje del chorro y a 1 m entre la tobera y el deflector.

Figura 3.16: Representación del campo de la presión en el dominio del deflector.

Capítulo 4

Modelado de la malla dinámica

En este capítulo se va a tratar el caso en el que el cohete deja de ser estático y comienza a desplazarse. Este desplazamiento plantea un problema ya que la fuente del chorro se va alejando de la plataforma de lanzamiento. De este modo, desde el punto de vista del cálculo numérico, se tiene que considerar que la malla sigue a la fuente en movimiento.

Al definir una malla móvil y otra estática, se puede tener en cuenta cómo la fuente donde se produce el chorro se va alejando de la plataforma de lanzamiento. Para ello, se van a introducir ciertas herramientas nuevas que harán posible el movimiento de la malla para simular el ascenso del cohete en la primera etapa de la misión de VEGA.

4.1. Introducción

Como se ha descrito previamente, OpenFOAM proporciona un generador de malla muy eficiente, *snappyHexMesh*, que está diseñado para poder mallar geometrías complejas con un enfoque de recorte de celdas. La malla resultante es una malla hexaédrica, con celdas poliédricas cerca de las paredes.

A la hora de elegir qué malla utilizar primero hay que pensar en qué técnica de movimiento se va a escoger. Los diferentes métodos de movimientos de las mallas que existen para el software OpenFOAM se dividen en función de los requisitos de la deformación de las celdas: técnicas de movimiento de mallas rígidas o mediante un enfoque de deformación de la malla.

El método *Overset*, que pertenece a la primera categoría, se caracteriza por no necesitar ninguna deformación de la malla para simular el movimiento, sino que se actualiza el dominio de flujo de acuerdo a diferentes técnicas. Por otro lado, *Mesh Adaption-Dynamic Remeshing* se basa en la deformación de la malla, la operación que se realiza en el ordenador se evalúa

intrínsecamente en la actualización de las celdas cercanas a la pared. A continuación se va a describir brevemente en que constan dichos métodos.

4.1.1. *Mesh Adaption-Dynamic Remeshing*

El *Mesh Adaption-Dynamic Remeshing* o mallado adaptativo es un método de mallado numérico que incluye una deformación de la malla como consecuencia del movimiento de las celdas internas. En este método el procedimiento a seguir es el siguiente: relajamiento de la malla, reconexión de la malla y remallado de la malla.

El primer paso se caracteriza por no tener ningún cambio en la conectividad de las celdas. Este paso se encarga de tratar de retrasar el remallado manteniendo la calidad de la malla tan alta como se pueda, de hecho, la distorsión de la malla en los contornos se suaviza dentro del dominio. El modo en el que dicha relajación se produce es definida por el usuario mediante, por ejemplo, la resolución de la ecuación de Laplace con una difusividad impuesta por el usuario. Lo que se trata de evitar con este paso es posponer lo máximo posible el remallado del dominio ya que conlleva una gran cantidad tiempo y puede llevar a errores en la interpolación [6].

El segundo paso es el de la reconexión de la malla, que se realiza cuando la distorsión de la malla es excesiva y no puede suavizarse en el primer paso. Solo se produce dicho remallado en aquellas celdas que tengan una mala calidad, lo que limita los errores de interpolación que puedan haber y reduce el esfuerzo computacional. Además del remallado, también se puede realizar un refinamiento aplicado en función de la zona de la malla que más se va a distorsionar, es decir, las celdas más cercanas a aquellas que porten el movimiento [6].

El último paso del algoritmo es el mapeado de la solución de las celdas antiguas en las nuevas celdas o las celdas que han cambiado de posición en la nueva malla. El mapeado se realiza intersectando las celdas de la malla antigua con las de la nueva, en este punto el cálculo y limitación de los gradientes de la malla fuente entran en juego. Por último, se interpola la solución mediante el volumen y la distancia pesada en series de Taylor y, finalmente, tenemos la solución en la malla de destino [6].

Este método puede conllevar cierto tipo de deformación, pero es bastante rápido y efectivo ya que el remallado es local, a pesar de ello, tiene ciertas limitaciones como que solo soporta celdas muy simples, mallas tetraédricas y no garantiza el alineamiento de las paredes. Otras limitaciones están relacionadas con el trabajo en paralelo, que resulta en un método poco robusto.

4.1.2. Overset

El fundamento de la malla Overset está en la aplicación del Método de los Volúmenes Finitos (MVF) a mallas superpuestas. Una o más mallas Overset, que conforman los cuerpos móviles, son impresas sobre una malla estática o *background*, que se mantiene inmóvil, y conforman el dominio computacional del caso. Una de las ventajas de este método consiste en evitar la deformación de la malla durante el cálculo del dominio móvil. En el método Overset la información entre las celdas superpuestas se comparte mediante la interpolación por puntos o por centro de las celdas mientras que en las celdas únicas la información se transmite a través de las caras.

Este método simplifica el mallado y aumenta la calidad de la malla ya que se puede mallar por separado la malla *background* sobre la que se superpondrán las demás mallas. Cuando se considera el movimiento de los cuerpos, se puede definir el movimiento de cada malla independientemente, ventaja que ofrece sobre otros métodos. Por lo tanto, este método ofrece una mayor robustez ya que permite un mayor rango de movimientos dentro del dominio total [7].

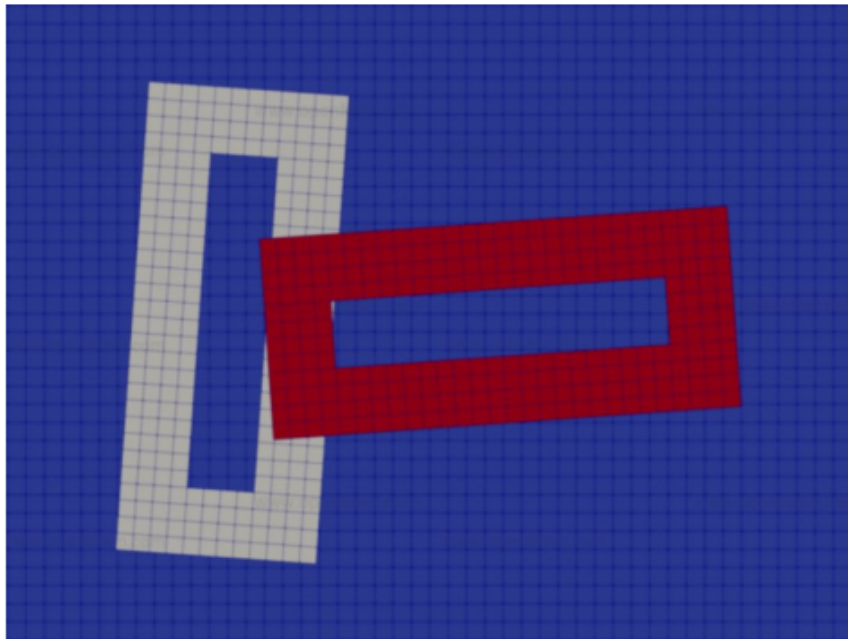


Figura 4.1: Dos mallas superpuestas (de color gris y roja) y una malla *background* [8].

Esta técnica Overset tiene como base el uso simultáneo de celdas activas, pasivas e interpoladoras. En las celdas activas, el cálculo se resuelve sin ninguna necesidad de interpolar nada. En cambio, en las celdas pasivas no se resuelve ningún tipo de ecuación ya que están temporal o permanentemente eliminadas del dominio de cálculo.

El procedimiento Overset comienza con la detección de las celdas que pertenecen a los volúmenes sólidos y por tanto no se resuelve el flujo en ellas. En OpenFOAM se denominan celdas

agujero o, con otras palabras, celdas pasivas. La resolución de todo el conjunto de ecuaciones debe darse en todas las mallas simultáneamente. Por ello, la solución en la malla *background* debe ser pareja a la obtenida por las mallas Overset, interpolando la solución entre las celdas receptoras y donantes.

Las celdas receptoras marcan el contorno de una malla sobre la otra superpuesta. Estas celdas actúan como unas celdas fantasma en la discretización de los volúmenes finitos, por lo que su centro solo actúa en las ecuaciones de discretización de sus celdas vecinas. Esto puede llevar a que haya un hueco entre el número de ecuaciones y el de variables. Este se llena con unas ecuaciones para interpolar los valores de las variables de los centros de las celdas de las receptoras. Las celdas donantes son un conjunto de celdas que se encargan de la creación de las ecuaciones interpoladoras, las cuales pueden estar basadas en diferentes métodos de interpolación [12]:

- ***cellVolumeWeight***: Este método es de precisión de primer orden. Es rápido, pero tiene problemas a la hora de detectar los agujeros.
- ***inverseDistance***: Es un método de segundo orden, pero es un método limitado y computacionalmente costoso.
- ***leastSquares***: Se trata de un método de segundo orden que puede llegar a estar limitado, también es computacionalmente costoso. Cabe destacar que tiene muy buena precisión y es el que se suele recomendar para las simulaciones Overset [11].

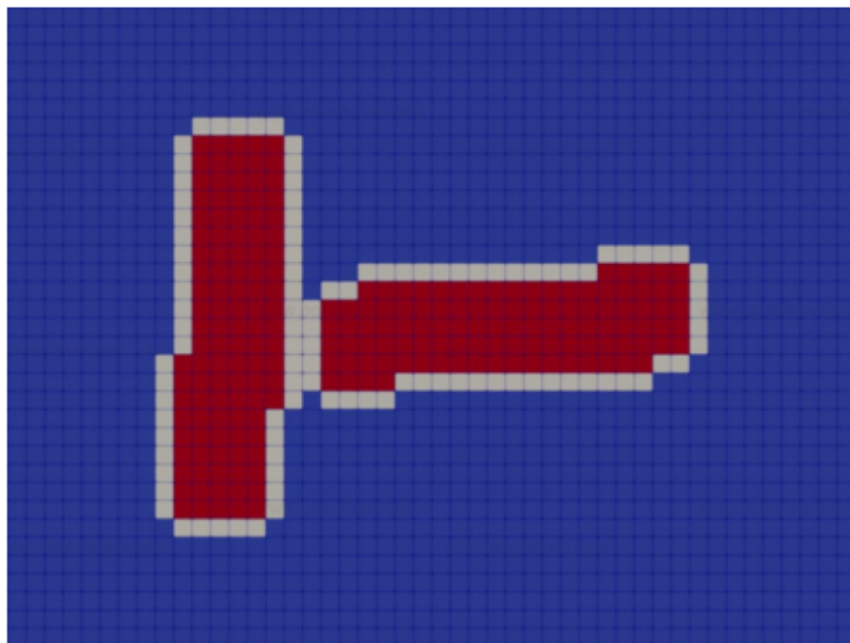


Figura 4.2: Malla Overset. El color azul indica las celdas calculadas, el rojo el agujero y el gris las celdas interpoladoras [8].

A continuación se procede a explicar con mayor detalle los pasos que sigue el método Overset internamente.

1. Identificación y corte de los agujeros

Las celdas agujero o *hole* son aquellas en las que no se calcula ningún tipo de ecuación. Estas celdas tratan de representar aproximadamente los contornos de las paredes de las geometrías en las mallas Overset. El corte se realiza mediante la intersección de las superficies sobre la malla *background* y, de este modo, todas las celdas dentro del corte de dicha superficie se denominan agujeros.

Tras haber marcado los contornos de pared, se aplica un algoritmo de rellenado por difusión dentro de las paredes para identificar que todas las celdas internas de esos contornos son agujeros. Para geometrías complejas es más fácil identificar las celdas que conforman los contornos mediante las celdas externas a la geometría que mediante las celdas internas.

La identificación de los agujeros puede fallar por diferentes razones: por el algoritmo de rellenado que se usa, por la compleja geometría de los contornos de las paredes Overset o por la pésima resolución de la malla. Por estos motivos, puede ser que las celdas agujero se identifiquen fuera en vez de dentro del dominio [7].

2. Búsqueda de las celdas donantes

El siguiente paso sería tratar de encontrar la pareja de celdas donante-receptor en el dominio. Primeramente se identifican todas las celdas donantes y receptoras del dominio. Las celdas candidatas para ser receptoras son todas aquellas que se encuentren a una distancia específica del contorno externo o de los contornos de los agujeros definidos en la fase previa. Por otra parte, las celdas candidatas para ser donantes son cualquiera que se ubiquen en la zona de superposición y que no sean agujeros.

Para determinar cada celda receptora para cada donante se realiza una búsqueda mediante un algoritmo tras el que cada receptor candidato informa si tiene una celda donante. Por último se le asigna la información al procesador que tiene dicha celda receptora.

Se pueden dar ciertas complicaciones a la hora de encontrar la pareja de celdas. Esto suele ocurrir cuando una posible celda donante no encuentra una celda receptora para la interpolación, lo que produce que se originen celdas huérfanas. Esto se puede llegar a dar cuando la zona de superposición entre mallas no es lo suficientemente grande o cuando los tamaños de celdas entre mallas superpuestas no son de un orden similar. Con el movimiento de la malla,

este fenómeno puede aumentar, aumentando el número de celdas huérfanas o cambiando sus posiciones a medida que se mueven. Se puede evitar este efecto aumentando la resolución de la malla y mejorando la zona de superposición de las mallas, asegurando que los tamaños de celdas de ambas mallas en la zona de superposición son del mismo orden [7].

3. Asignación de los tipos de celdas de la malla

Todas las celdas que todavía no se han identificado se marcan como celdas de cálculo o receptoras. A la hora de asignar el tipo a las celdas, lo primero que se hace es identificar a las celdas que se encuentran en las cercanías de los agujeros como celdas receptoras, que deben siempre interpolar la solución de las celdas de otra malla. Este paso asegura que el algoritmo de resolución de todas las celdas sean válidas en el sistema de discretización que gobiernan las ecuaciones diferenciales parciales.

Tras definir las celdas receptoras que se encuentran en las cercanías de los agujeros, las celdas de cálculo y las receptoras son designadas entre las celdas restantes que correspondan. La resolución de las celdas se compara con la de todas las celdas donantes candidatas. Si la resolución de la celda es la más alta, entonces esta se identifica como una celda de cálculo. En caso contrario, se identifica como una celda receptora y su celda donante se elige como aquella que tenga la mejor resolución. Como paso final, los procesadores comparten la información sobre las celdas donantes candidatas rechazadas y aceptadas [7].

4. Interpolación de las celdas

Tras la asignación a cada celda de su tipo de propiedad, la interpolación se realiza para cada pareja de celdas receptoras/donantes. Tras este proceso, la solución se interpola para la celda receptora mediante una interpolación del orden que se le indique. La solución interpoladora se calcula como la media pesada de la información de las celdas interpoladas [7].

4.2. Implementación del Método Overset

Para poder generar una malla Overset se deben de definir los siguientes parámetros que no son descritos para las mallas del caso estático:

- El caso debe tener como mínimo un “patch” de tipo Overset, preferiblemente que sea el primero en la lista del archivo *boundary*. La condición de contorno de este “patch” será de tipo Overset, mediante la inclusión del archivo set-up por defecto en la sección del campo de contornos o *boundaryField* de los archivos existentes en la carpeta θ . La

inclusión se realiza mediante la escritura del siguiente comando en los archivos especificados [7]:

```
#includeEtc "caseDicts/setConstraintTypes"
```

- La zona donde se superponen las dos mallas tienen que tener como mínimo cinco capas de celdas tanto en la malla *background* como en las *Overset*. Este requerimiento, puede ser una desventaja ya que el tamaño y la cantidad de celdas aumenta de gran manera pero, por otro lado, esta técnica permite mantener la morfología de la malla intacta, lo que favorece a que el mallado tenga una calidad excelente durante la simulación.
- Se crea un fichero en la carpeta *0* llamado *zoneID* donde se genera un *volScalarField* que indica a qué zona pertenece dicha malla. Este campo se puede generar mediante unas herramientas que proporciona OpenFOAM como son *topoSet* o *setFields* [7].
- Cuando las paredes de alguna geometría se vayan a mover debido al movimiento de la malla, se le debe asignar la condición de contorno de *movingWallVelocity*.
- En el diccionario *fvSchemes* se debe seleccionar el método de interpolación y qué tipo debe ser elegido. Del mismo modo se debe elegir que campos se tienen que interpolar [7].

```
oversetInterpolation
{
    method      inverseDistance;
}
```

```
oversetInterpolationRequired
{
    U;
    p;
    T;
    k;
    omega;
}
```

- En el caso de la discretización temporal, es preferible utilizar el método Euler antes que el esquema de CrankNicolson, ya que este puede introducir oscilaciones espurias en las mallas móviles.
- El método de cálculo de la distancia a la pared que existe por defecto, *meshWave*, ya no es válido para la interpolación *Overset*. Para este tipo de mallas se debe optar por un método continuo como *Poisson* o *advectionDiffusion* [7].

- Es necesario actualizar los *solvers* asimétricos en el diccionario *fvSolution*. Para las ecuaciones de transporte (U , k , etc) es aconsejable seleccionar el *solver smoothSolver* con *symGasussSeidel* y para las ecuaciones elípticas (p , $yPsi$) el *solver PBICGStab* con un preconditionador *DILU* [7].
- Para el movimiento de los cuerpos, se utiliza la opción *moveMeshOuterCorrector* en el diccionario *fvSolution* para ganar estabilidad. Este paso actualizará la malla cada iteración externa del bucle PIMPLE.
- Si se añade la entrada *checkMeshCourantNo* en el diccionario *fvSolution* se comprueba el número de *Courant* de la malla.
- Se debe definir el tipo de malla a *dynamicOversetMesh* en el archivo *dynamicMeshDict* que se encuentra en la carpeta *constant* [15].

```
dynamicFvMesh      dynamicOversetFvMesh;
solver             displacementLaplacian;
displacementLaplacianCoeffs
{
    diffusivity     uniform 1;
}
```

En este mismo archivo, e inmediatamente después, se deben escribir los parámetros de movimiento del cohete. En este caso, el usuario especifica el tipo de movimiento y, por lo tanto, la malla se mueve de acuerdo al movimiento que se ha especificado.

Para cargar el *solver* que incorpora los diferentes tipos de movimientos es necesario escribir los siguientes comandos:

```
dynamicMeshDict
{
    dynamicFvMesh solidBodyMotionFvMesh;
    solidBodyMotionFvMeshCoeffs
    {
        //Definición de los parámetros de movimiento
    }
    motionSolverLibs ( "libfvMotionSolvers.so");
}
```

Existen distintos tipos de movimientos que se pueden seleccionar, aunque en este caso se va a seleccionar un desplazamiento lineal. Se puede aplicar a todo el dominio o a una región específica como se hace en este caso en VEGA [15].

```
multiSolidBodyMotionSolverCoeffs
{
    VEGA
    {
```

```

        solidBodyMotionFunction linearMotion;
        linearMotionCoeffs
        {
            velocity (0 0 3.588);
        }
    }
}

```

La velocidad se debe introducir como un vector. La magnitud de dicho vector es la velocidad total, cuyas unidades en la simulación se definen, por lo general, en m/s.

- Se requiere hacer un mapeado de la solución ya que antes del despegue se realiza el encendido del motor. Los campos calculados para el último paso temporal del encendido serán los campos iniciales del caso Overset. La etapa de encendido del motor ha sido calculada con una malla estática en el estudio previo. De este modo, se evita tener que simular los instantes previos a que el cohete comience a moverse.

Al mapear el caso cabe definir que los campos no son consistentes. Esto se debe a que la malla y las geometrías del caso estático (origen) no son las mismas que las del caso Overset (destino).

Por ello, el fichero *mapFieldsDict* debe seguir las siguientes reglas:

- La información de los campos es mapeada del origen al destino de donde sea posible. Esto quiere decir que la información de los campos dentro de la geometría destino serán mapeados del origen, excepto aquellas regiones que no estén dentro de estas geometrías, las cuáles mantendrán el campo previo al mapeado.
- La información del campo de los “patch” se dejará inalterada, a no ser que se especifique de otra forma en el archivo *mapFieldsDict*.

El archivo *mapFieldsDict* se compone de dos listas que especifican la información de los “patch” a mapear. La primera lista es el *patchMap*, este especifica la información del mapeado entre parejas de “patch” del origen y destino que son geoméricamente coincidentes. La segunda lista es *cuttingPatches* que contiene los nombres de los “patch” del destino cuyos valores se tienen que mapear desde el campo interno del origen [4].

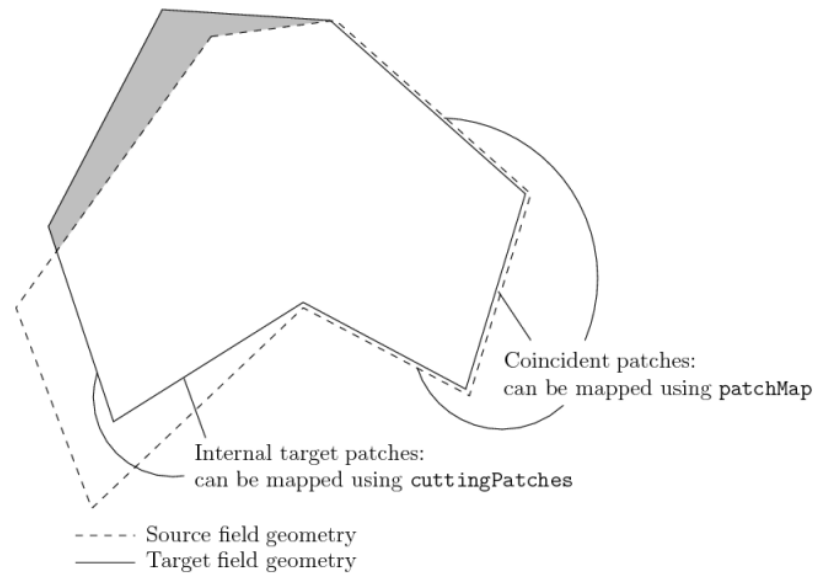


Figura 4.3: Mapeado de campos inconsistentes [4].

Una vez definidos todos los parámetros necesarios para poder calcular el caso, falta el dato más importante, el *solver* que se va a usar. El *solver* utilizado en este caso es el *overRhoPimpleDyMFoam*. Este se usa para casos transitorios donde el flujo es compresible y turbulento. Además, acepta cambios en el movimiento y la topología de la malla. No existen una gran cantidad de *solvers* disponibles para mallas Overset, pero este cumple con los requisitos de modelar flujos compresibles y turbulentos así como cambios en el movimiento de la malla.

Esta herramienta utiliza un esquema implícito y está diseñada para trabajar con flujos con número de Mach menores de 0.3 (subsónicos). Sin embargo, se puede introducir en el fichero *fvSolutions* en el apartado de los parámetros del PIMPLE, una entrada que diga *transonic true*. De este modo, el *solver* será capaz de modelar flujos supersónicos.

4.3. Caracterización del movimiento del cohete VEGA

Para poder conocer cuál es la velocidad del cohete VEGA, se ha hecho un análisis mediante las características que se pueden conocer como el empuje máximo, el tiempo de combustión o la masa del cohete. Aparte, con aquellos datos que no se conocen, se ha hecho una estimación mediante vídeos y fotogramas del ascenso del cohete.

Los datos que se conocen sobre el cohete son los siguientes:

- Empuje del motor de la 1ª etapa (P80): 3015 kN.

- Tiempo de combustión en la 1ª etapa: 110 s.
- Masa total del cohete en el despegue: 137000 kg.
- Masa del combustible en la 1ª etapa: 88000 kg.
- Impulso específico del motor P80: 280 s.

4.3.1. Propulsión en cohetes

La fuerza externa que se ejerce sobre un objeto se puede determinar como la rapidez en la que cambia el momento. Utilizando dicha referencia y aplicando la segunda Ley de Newton se obtiene la siguiente ecuación:

$$F = ma = \frac{m\Delta v}{\Delta t} = \frac{\Delta p}{\Delta t} \quad (4.1)$$

donde el momento se define como el producto de la masa por la velocidad $p = mv$.

Si se generaliza de modo que sea una ecuación para un tiempo instantáneo y para una masa variable, como es el caso del cohete, ya que va quemando combustible a medida que asciende. De esta manera, haciendo uso de la derivada del producto, se obtiene una ecuación que modela el empuje de un cohete a partir de la segunda Ley de Newton para una masa variable [23]:

$$F_{externa} = \frac{d(mv)}{dt} = m \frac{dv}{dt} + v \frac{dm}{dt} \quad (4.2)$$

De una forma similar se puede calcular la velocidad, sabiendo que el empuje de un cohete es debido a la eyección de gases a gran velocidad por la tobera del motor. Junto con la conservación del momento, se constata que como cierto material del cohete se va perdiendo, el momento del resto del cohete debe aumentar para cumplir la conservación.

Por lo que se define que el momento en el instante inicial es $P_i = Mv$ y debido a la conservación del momento, para un instante final $P_f = -\Delta MU + (M + \Delta M)(v + \Delta v)$. Donde U es la velocidad de los gases desde un marco de referencia inercial, ΔM es el cambio de masa debido a la pérdida de combustible, y v y Δv son la velocidad del cohete y la variación de dicha velocidad respectivamente [23].

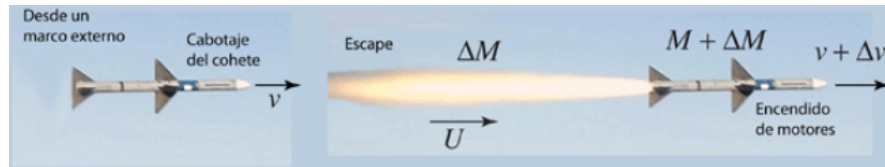


Figura 4.4: Representación de la conservación del momento [23].

Realizando un cambio de referencia al del propio cohete, en el que se define que la dirección de los gases es contraria a la velocidad del cohete, $u = v + \Delta v - U$ (la velocidad de los gases en el marco de referencia del cohete es igual a la velocidad del cohete con su variación menos la velocidad de los gases en el marco de referencia inercial), y resolviendo la conservación del momento $P_i = P_f$:

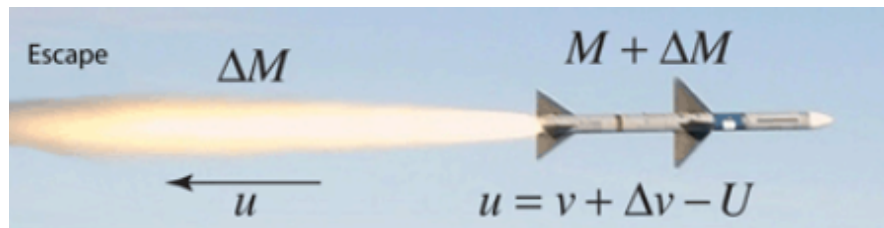


Figura 4.5: Cambio de marco de referencia [23].

$$\begin{aligned}
 P_i = P_f &\rightarrow Mv = -dM(-u + v + dv) + (M + dM)(v + dv) \\
 0 &= -Mv + udM - vdM + dvdM + Mv + Mdv + vdM + dMdv
 \end{aligned}
 \tag{4.3}$$

donde despreciamos el producto de las diferenciales $dMdv$ y dividiendo por dt se obtiene [23]:

$$-u \frac{dM}{dt} = M \frac{dv}{dt} = Ma = F_{externa}
 \tag{4.4}$$



Figura 4.6: Cambio de marco de referencia [23].

A partir de la segunda Ley de Newton y bajo la influencia de la gravedad se puede calcular la velocidad de la siguiente forma:

$$F = ma \rightarrow a = \frac{dv}{dt} = \frac{F}{m} - g \quad (4.5)$$

introduciendo la Ec. 4.4 en la Ec. 4.5 conseguimos la diferencial de la velocidad del siguiente modo:

$$\frac{dv}{dt} = -\frac{u}{M} \frac{dM}{dt} - g \rightarrow dv = -\frac{u}{M} dM - g dt \quad (4.6)$$

Integrando la Ec. 4.6 desde el inicio del despegue hasta un tiempo t , se puede obtener la velocidad para cualquier tiempo de la 1ª etapa de la siguiente manera:

$$v - v_0 = -u \ln \left(\frac{m}{m_0} \right) - gt \quad (4.7)$$

donde u es la velocidad de los gases de escape, v es la velocidad para el instante t , v_0 es la velocidad inicial del cohete, m y m_0 son la masa para el instante t y la masa inicial respectivamente, g es la gravedad y t es la variable tiempo para la que se quiera calcular la velocidad.

Esta ecuación es una variante de la ecuación de Tsiolkovsky para los cálculos de la propulsión a chorro. La velocidad de escape de los gases por la tobera se calcula mediante el impulso específico del cohete (I_{sp}) y la aceleración gravitatoria (g) mediante la ecuación $u = I_{sp}g$. Como se sabe que $I_{sp} = 280$ s, para el motor P80, se puede obtener que la velocidad de los gases de escape de la tobera es de 2744 m/s.

Por lo tanto, conociendo todos los datos necesarios, se puede averiguar la velocidad para el final del tiempo de combustión de la 1ª etapa. De este modo, se puede interpolar la velocidad para conocer esta en cada instante de tiempo. El resultado de la velocidad alcanzada para el tiempo de combustión, 110 s, es de 1743.27 m/s.

Para poder realizar una interpolación de mayor calidad se necesitan más datos intermedios. Puesto que no se han proporcionado ningún tipo de dato adicional, la forma de obtener estos valores consiste en la visualización de vídeos y fotogramas del despegue del cohete VEGA.

No existe una gran cantidad de vídeos al respecto, y los que se encuentran no son de muy larga duración, por lo que únicamente se ha podido estudiar los tres primeros segundos del despegue. Del mismo modo, se conoce que a los 30 s el cohete rompió la barrera del sonido a 4.7 km de altura. De esta manera, asumiendo que la barrera del sonido son 1234.8 km/h o 343 m/s, se conoce otro valor más a aportar en la interpolación.

Mediante los vídeos se pueden obtener los siguientes datos:

	Altura de ascenso (m)	
Tiempo de ascenso (s)	1	3.588
	2	11.305
	3	26.737

Tabla 4.1: Datos del despegue del cohete VEGA mediante vídeo

Realizando una interpolación lineal se obtienen los siguientes resultados para las velocidades:

Velocidad de ascenso (m/s)		
Tiempo de ascenso (s)	1	3.588
	2	6.685
	3	11.973

Tabla 4.2: Datos del despegue del cohete VEGA mediante vídeo

La tendencia que tiene la velocidad del cohete es de tipo exponencial cuando el marco de referencia sea el del cohete. De este modo, a estas velocidades se le tendrá que añadir la velocidad de los gases de escape que es constante. Conocidos todos estos valores, se puede observar en la Figura 4.7 cómo es la curva de la velocidad de ascenso hasta el fin del tiempo de combustión de la 1ª etapa.

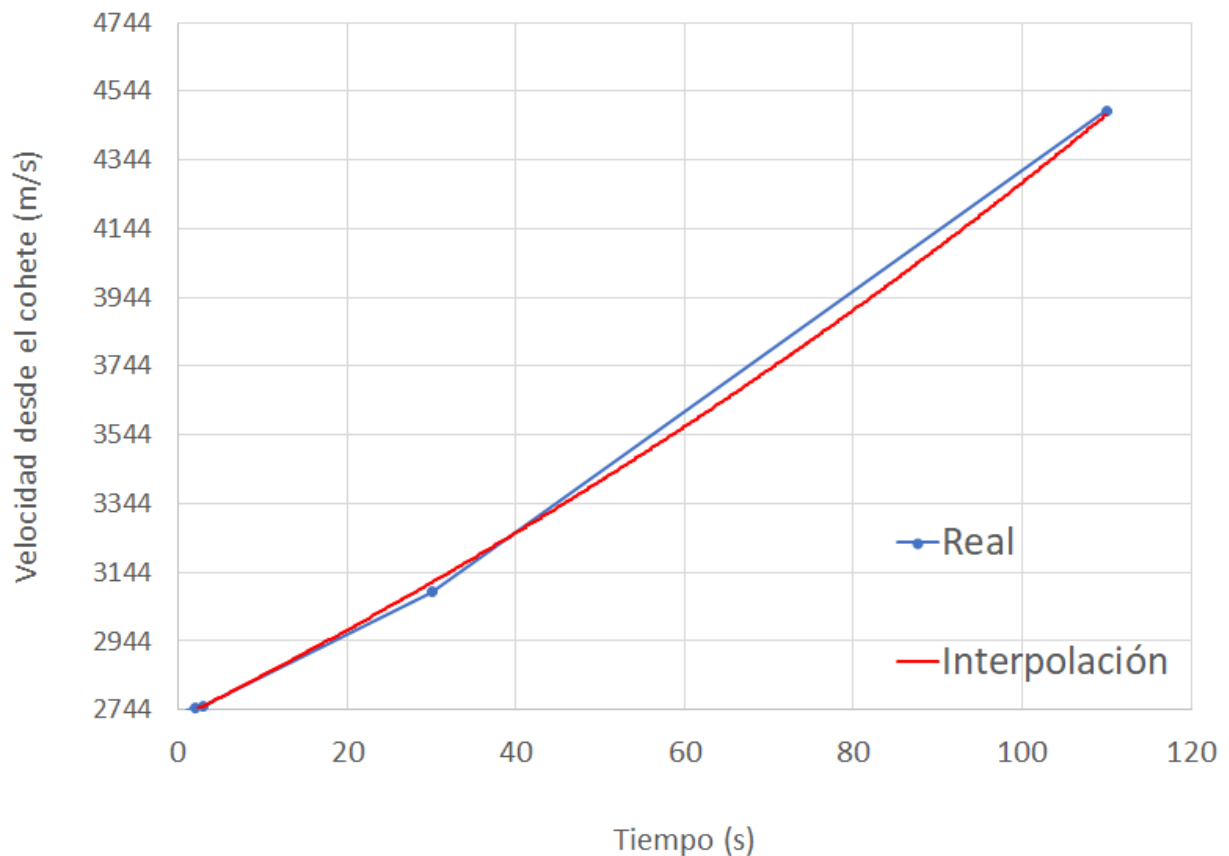


Figura 4.7: Curva de velocidad de ascenso de VEGA [22] [21] [24].

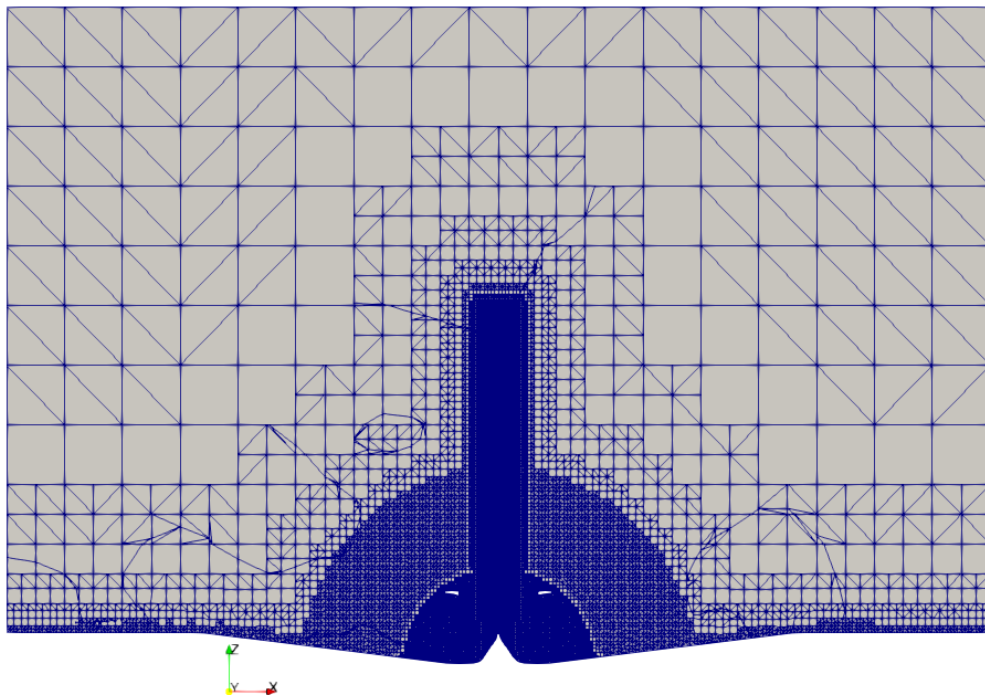
Como se puede observar la interpolación representa con gran exactitud los datos impuestos, por lo que se asume que la ecuación que representa la gráfica es la que tendrá el cohete real que es $v = 2718.3 e^{0.0045 t}$. El valor que se va a introducir en el *dynamicMeshDict* es el correspondiente a 1 s dado a que no se espera que la simulación que se va a calcular supere ese valor.

4.4. Discusión de los resultados

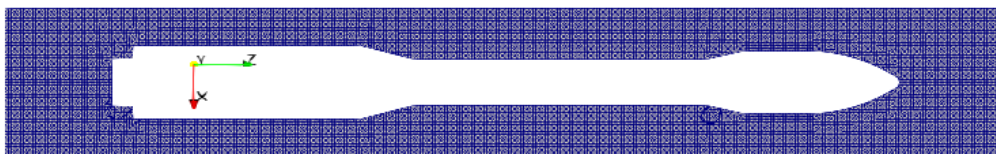
Relevancia del mallado

Uno de los aspectos más importante a la hora de modelar un caso con el método Overset es la elaboración de la malla. Para la consecución de esta se deben seguir una serie de pasos que se van a describir a continuación.

Se deben realizar tantas mallas como objetos móviles se requieran además de la malla donde se encontrarán los objetos estáticos. Como se puede ver en la Figura 4.8, en este caso se han realizado dos mallas: una donde se ubica el cohete, que se dotará con movimiento, y otra estática o *background* donde se encontrarán todas las geometrías que vayan a quedarse inmóviles durante la simulación.



(a) Malla *background* o estática.



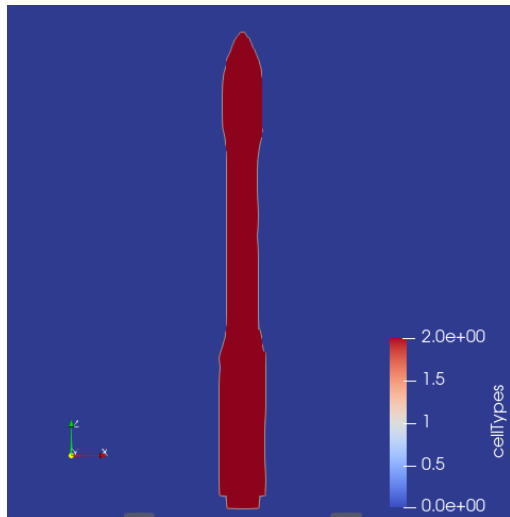
(b) Malla del cohete o móvil.

Figura 4.8: Representación de las mallas utilizadas.

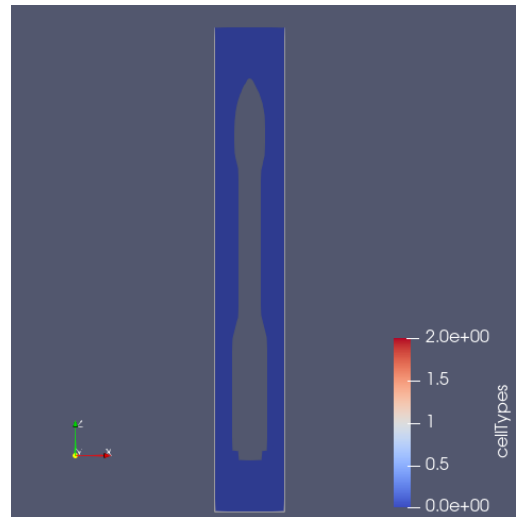
Cabe destacar que entre las superficies de las diferentes mallas debe haber un número mínimo de capas de celdas. Esta cantidad permite al *solver* poder realizar la interpolación. En el

caso de no tener suficiente número de capas, el *solver* no identificará correctamente las celdas receptoras y donantes y no plasmará con exactitud las celdas interpoladoras.

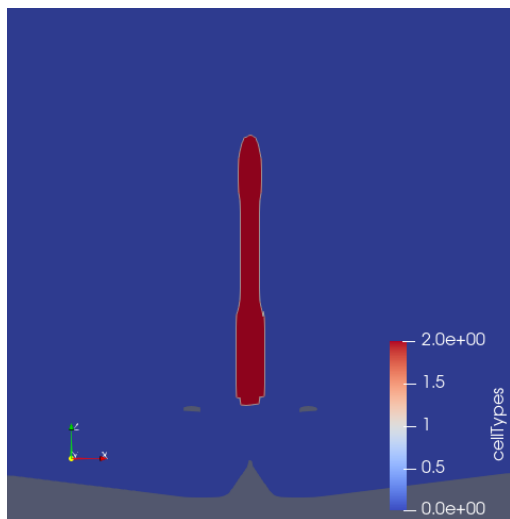
Para mostrar este efecto, se han mallado dos casos con elementos suficientes pero uno con mayor refinamiento que el otro. En la Figura 4.9 se puede observar cómo en las imágenes superiores al ser el refinamiento mayor, el tamaño de las celdas es menor para la zona que se va a interpolar, se reproduce el contorno del cohete en la malla *background* mediante las celdas interpoladoras con una mayor precisión. Esto facilitará la interpolación en nuestro caso ya que por una de esas zonas se tiene la fuente del chorro.



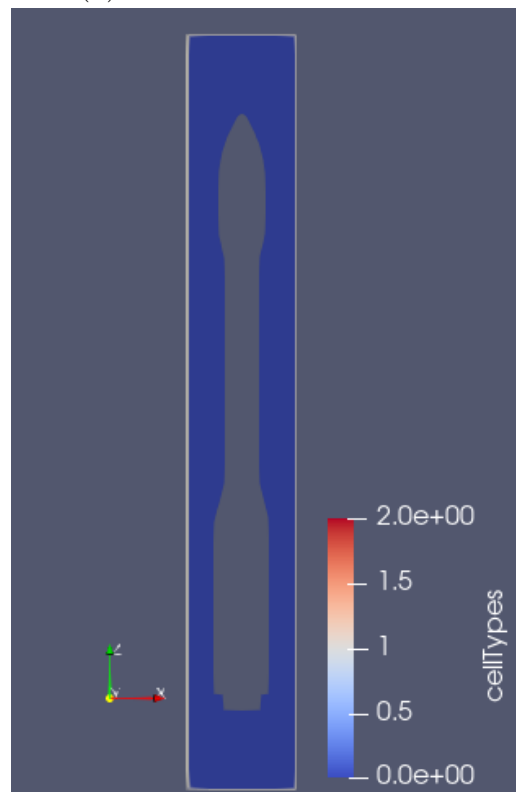
(a) Malla *background* o estática.



(b) Malla del cohete o móvil.



(c) Malla *background* o estática.



(d) Malla del cohete o móvil.

Figura 4.9: Representación de las mallas utilizadas.

Aunque como se ha visto en la Sección 3.2.2 la geometría de la plataforma de despegue tiene mucha influencia sobre los resultados, para ser capaces de aproximar correctamente la geometría se ha decidido elegir la plataforma de despegue del estudio previo, ya que en esta hay una mayor distancia entre la tobera del cohete (malla móvil) y la plataforma de despegue (malla estática). Para evitar el cambio de la geometría, se debe realizar un refinamiento tan alto como sea posible para conseguir que entre dichas superficies hayan el mínimo número de capas de celdas.

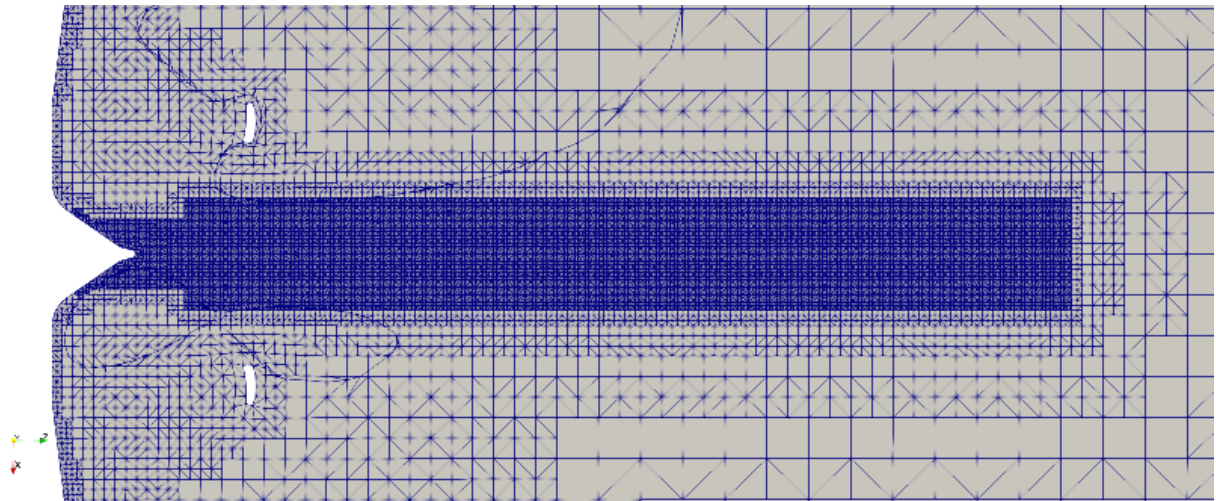
No se ha podido hacer un refinamiento parcial de la zona de la plataforma de despegue debido a otra característica de las mallas Overset. Las zonas de las mallas que se solapan, es decir, se interpolen, deben tener el mismo orden de tamaño de celda para que las celdas puedan encontrar los nodos comunes con exactitud. De esta manera, si se realiza un refinamiento local en una zona que se encuentre en este solapamiento, también se debe realizar el mismo sobre las demás mallas y en la misma zona de solapamiento, para mantener el mismo orden de tamaño de celda. Cabe destacar que este es uno de los factores más importantes ya que, como se ha visto durante el cálculo de los casos, hacía que el cálculo entrara en un tipo de divergencia.

El motivo de no realizar estos refinamientos locales en las mallas pertinentes se debe a que la malla móvil se va a ir desplazando. Por consiguiente, al paso de unos instantes temporales la zona que tenía el mismo refinamiento se habrá desplazado y se encontrará con una zona que no lo tenga y, por lo tanto, no se cumplirá la norma de que tengan el mismo orden.

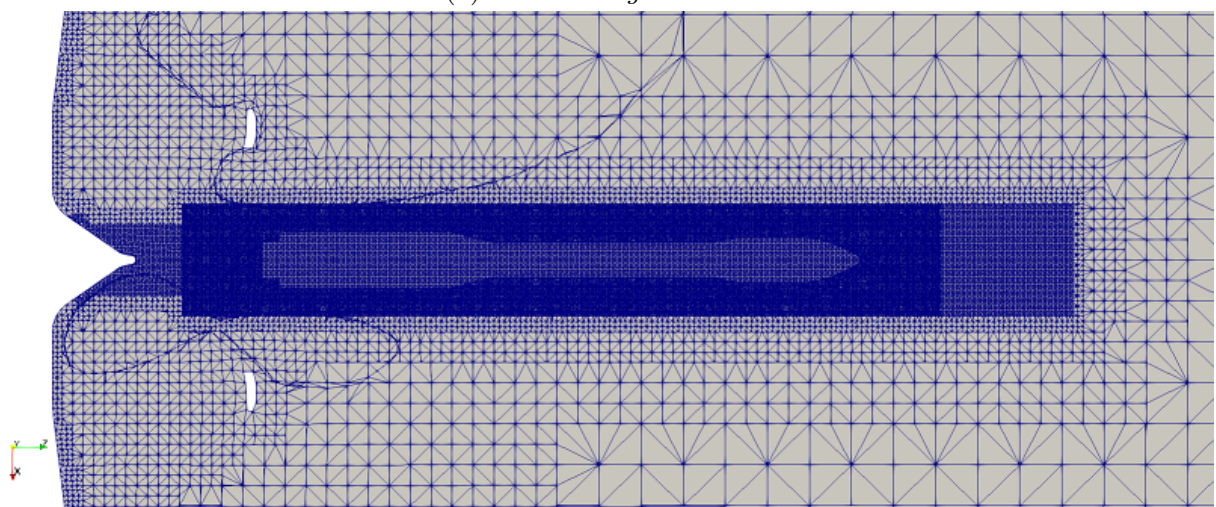
Una alternativa sería que conociendo el movimiento que va a tomar la malla móvil, refinar toda esa zona de la malla background hasta conseguir el mismo de tamaño de celda. No obstante, esto implica un gran aumento del coste computacional. Por ello, en este trabajo no se ha aplicado.

El siguiente paso tras haber realizado los mallados por separado es unir las mallas. Se superponen unas sobre las otras identificando siempre cuál es cuál. Esto se puede averiguar mediante el comando *checkMesh* y observando las regiones que existen.

Una vez unidas las mallas ya tenemos finalizado el mallado como se puede ver en la Figura 4.10. Se observa cómo en la zona del cohete existe un mayor número de celdas. Es un error de visualización, realmente es la superposición de la malla móvil y de la estática.



(a) Malla *background*.



(b) Malla *background* y malla cohete unidas.

Figura 4.10: Representación de las mallas utilizadas.

Coste computacional del caso

En este apartado se va a proceder a calcular cuál es el coste del cálculo de la malla Overset. El coste computacional de un caso va estrechamente relacionado con el número de celdas que se tengan que calcular y, particularmente en este caso, la cantidad de celdas que tiene que interpolar.

Cabe tener en cuenta que el mallado Overset lo que realiza es una superposición de mallas, por lo que se va a obtener un número de celdas bastante elevado. Existirán zonas donde habrán el doble de celdas debido a esta superposición pero en realidad son dos mallas independientes unidas por las celdas interpoladoras.

Que las zonas superpuestas tengan que tener el mismo tamaño de celda influye en gran medida

al número de celdas total. Por ejemplo, en el caso de que se quiera refinar una zona, se deberá refinar esa zona y la correspondiente a la malla superpuesta. Por lo tanto, si se quiere refinar todo ese dominio en vez de ser el doble de celdas en esa zona será cuatro veces el número de celdas.

La malla que se ha calculado tiene 12M de celdas de las cuáles 5.5M corresponden a la malla del cohete y 6.5M a la malla *background*. Como se puede observar influye en gran medida la existencia de dos mallas, ya que si únicamente tuviéramos la malla estática se necesitarían 6.5M y se podría refinar más la malla o el cálculo iría más rápido. Además, el proceso de cálculo también se ralentiza debido a que se tiene que realizar la interpolación de las celdas entre las mallas.

Esta malla se ha lanzado en el supercomputador de la UPV, Rigel, con 128 procesadores. Para ejecutar 8.4×10^{-4} s ha tardado 300 h, por lo que cada 1×10^{-4} s tardan 35.71 h en calcular. Para poder alcanzar 1 s de cálculo y poder analizar con detalle el análisis del despegue se necesitarían 357100 h, lo que equivaldría a 40.7 años. Como se puede apreciar, este tipo de simulación no es viable ya que requiere mucho coste de tiempo. Se necesitaría un computador más potente para poder ejecutar el cálculo con un mayor número de procesadores a pesar de necesitar una mayor comunicación entre ellos.

Por otro lado, se requeriría un mayor refinamiento de la malla, ya que ciertas zonas de esta no se han podido refinar como se quería, debido a que aumentaba demasiado el coste computacional.

De forma paralela, se han lanzado dos mallas similares pero de menor número de celdas totales. Esto supone que el refinamiento de la malla es peor y, por lo tanto, los resultados no son del todo exactos.

Estos casos paralelos se van a utilizar para realizar una aproximación de cuántos procesadores serían óptimos para lanzar el caso y optimizar el coste de tiempo de los procesadores. Las mallas que existen son las siguientes:

Número de celdas	Nº procesadores	Tiempo de cálculo del caso (s)	Tiempo de ejecución de la máquina (h)
2M	32	2.064e-2	540
6.6M	128	4.93e-3	645

Tabla 4.3: Mallas calculadas

De las dos mallas se va a utilizar, para hacer una comparación, la que se ha calculado también con 128 procesadores ya que la comunicación entre estos es la misma.

Se observa que para 1×10^{-4} s se tarda 13.08 h por lo que el tiempo de uso de la máquina

para llegar a 1s de cálculo es de 14.93 años, algo inviable. Un tamaño de celda relativamente óptimo en la malla requeriría unas 25M de celdas. Mediante una interpolación entre los datos de las mallas que se tienen se puede estimar que el tiempo de uso de la máquina para llegar a 1 s de cálculo con una malla de 25M de celdas es de 75.896 años, totalmente inviable. Debido a estos datos, se requiere una mayor potencia computacional y una elección óptima de los refinamientos de la malla para poder agilizar el cálculo.

Análisis de los campos

En este apartado se van a estudiar los campos más relevantes que han surgido debido al movimiento del cohete. Cabe mencionar que los resultados mostrados a continuación son los referentes al instante 8.4×10^{-4} s, ya que es el tiempo que se ha conseguido resolver hasta este momento.

Se puede observar en la Figura 4.11 cómo aparece en la punta del fairing un aumento de la presión debido al movimiento del cohete. Del mismo modo, se visualiza una depresión en el cambio de sección del cohete que equivale a la zona de unión entre la primera etapa y la segunda.

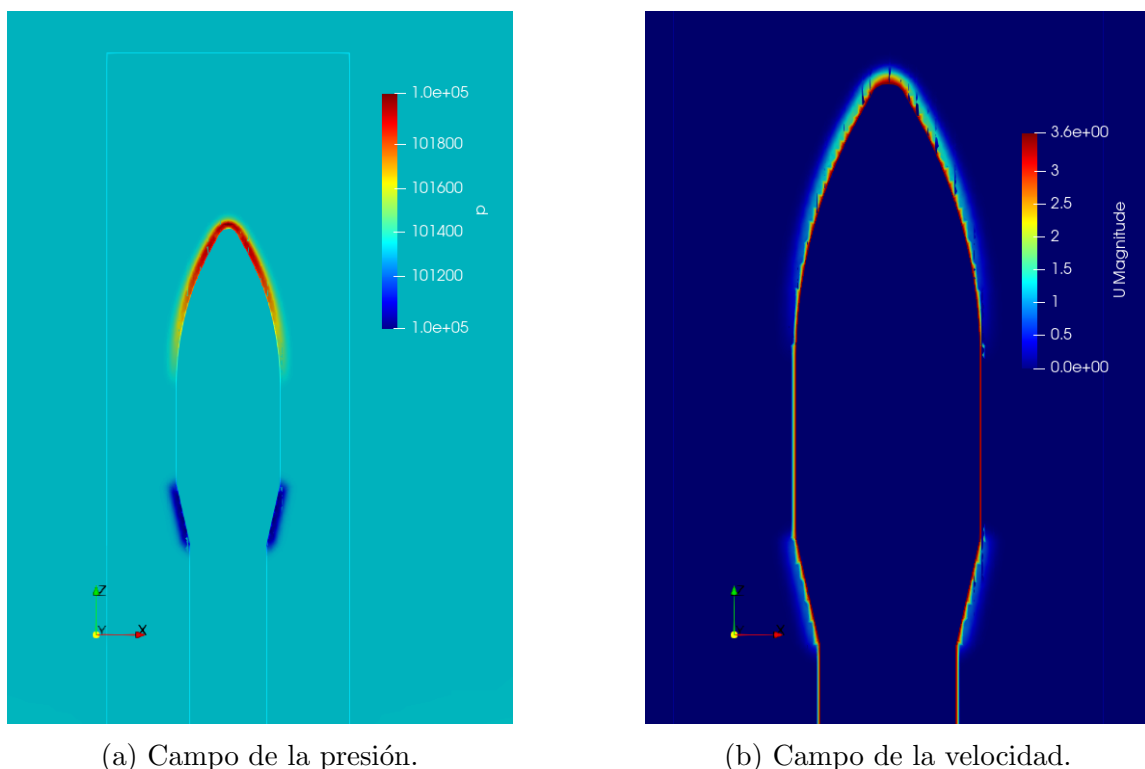


Figura 4.11: Representación del fairing.

Como se percibe en la Figura 4.12, en el interior del chorro aparecen una especie de perturbaciones. Estas perturbaciones se pueden deber a que lo primero que se encuentra el chorro

que sale de la tobera es una celda interpoladora, por lo que el valor que se obtiene no es un valor calculado sino interpolado. Este valor puede introducir cierto tipo de error que es lo que simula estas formas de capa.

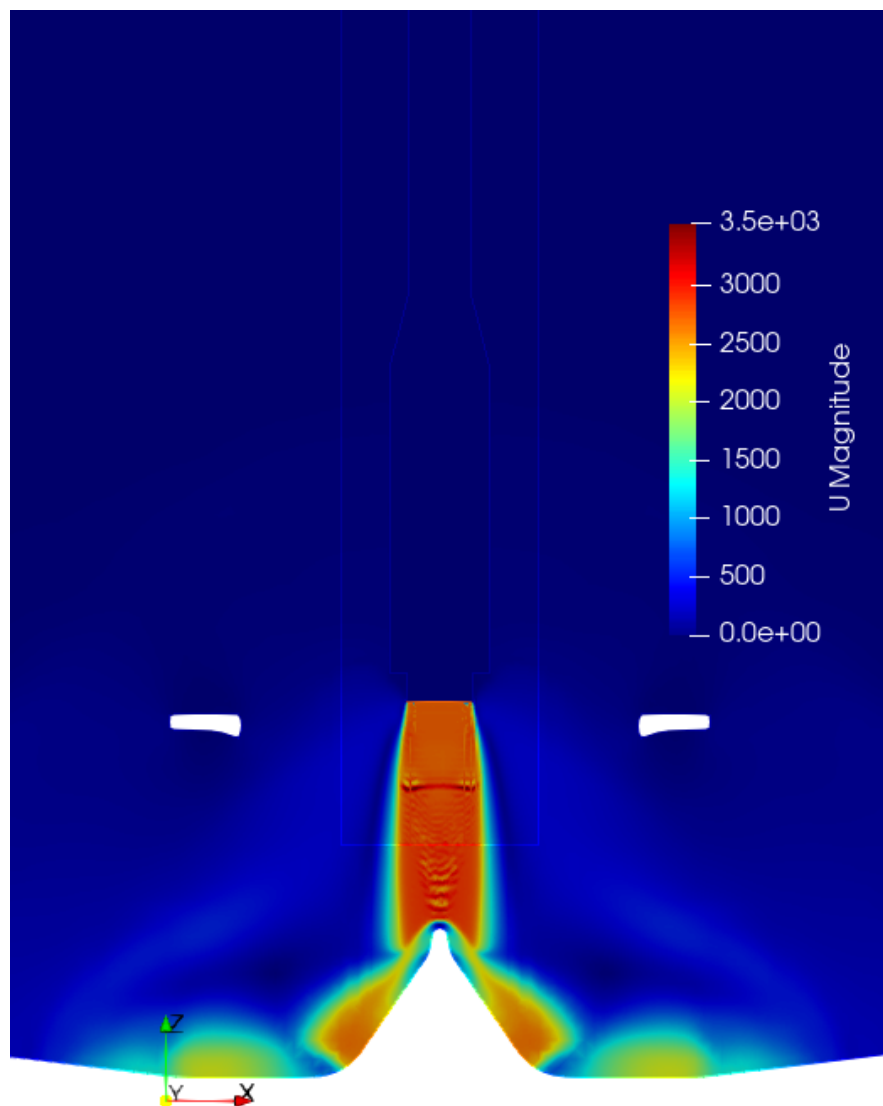


Figura 4.12: Representación de la velocidad en la zona inferior del cohete.

Por otra parte, se pueden identificar las fuentes sonoras que aparecen por el efecto del movimiento del cohete. Se pueden predecir las fuentes de generación de ruido mediante su visualización. Para representar las ondas de choque, producidas por la generación de ruido, se suele utilizar una técnica llamada las fotografías de Schlieren, que captan los gradientes de la densidad del fluido.

Por consiguiente, se han intentado reproducir estas fotografías con los datos de las simulaciones. Para ello, se ha representado la divergencia del gradiente de la densidad para el paso

temporal más avanzado que se ha conseguido.

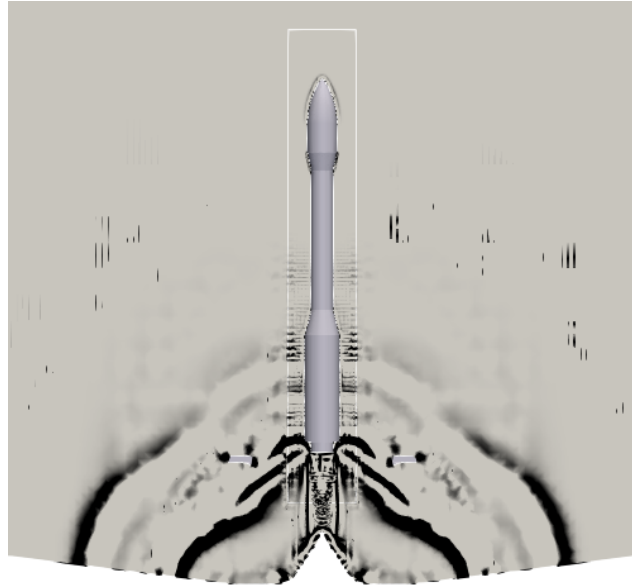
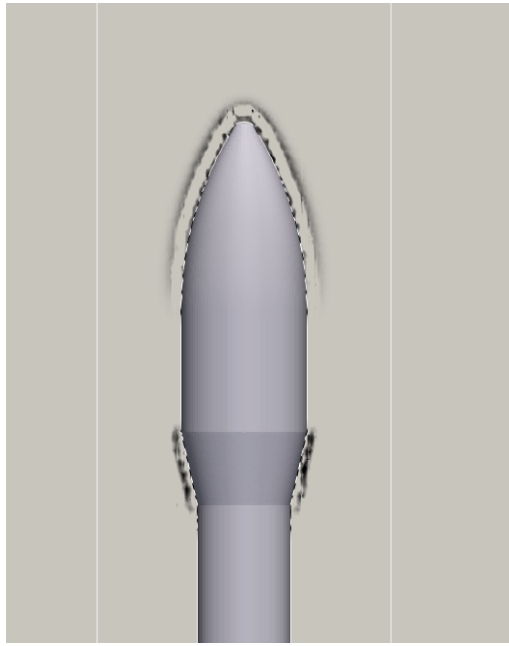
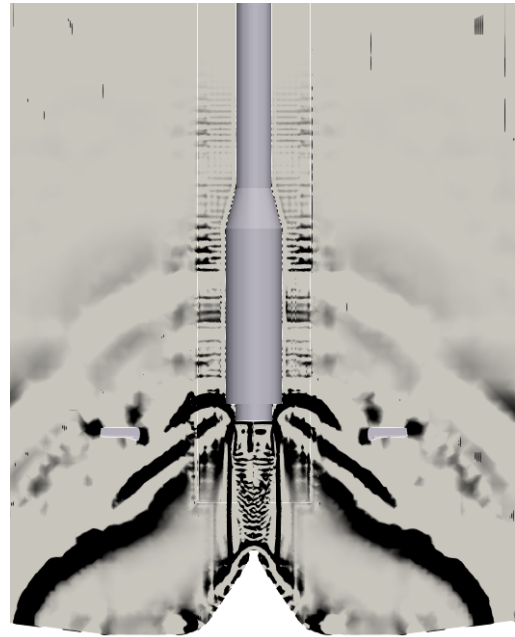


Figura 4.13: Fotografías de Schlieren en el dominio de cálculo.

En la Figura 4.13 se observan diversas ondas de choque dominantes. Una se debe a la frontera entre el chorro y el flujo en movimiento en la plataforma de despegue; otra a la onda de choque separada en la punta del deflector y, por último, la que se genera por el cambio de curvatura en la rampa del deflector. Además, se observa una onda en la zona exterior debida al avance del fluido en movimiento. Por consiguiente, las ondas de choque creadas por la pluma del cohete y su impacto con la plataforma de despegue son las principales fuentes generadoras de ruido que se han detectado.



(a) Fotografías de Schlieren en el fairing del cohete.



(b) Fotografías de Schlieren en la zona inferior del cohete y deflector.

Figura 4.14: Fotografías de Schlieren para la visualización de las ondas de choque.

En la Figura 4.14a se observa las ondas de choque que se están formando en el fairing, tanto en la punta del cohete como en la unión entre la etapa 1 y 2. Estas ondas de choque se deben al movimiento producido por el cohete. Por otro lado, cabe mencionar cómo en la Figura 4.14b se observa en la zona del chorro unas perturbaciones cuyo origen se ha explicado previamente.

Del mismo modo, se puede calcular la cantidad de energía por tiempo que producen dichas fuentes mediante la potencia sonora. Este término es básico en el campo de la aeroacústica ya que mide la energía acústica que puede producir una fuente. Con el fin de calcular la presión sonora generada por la intensidad del sonido, se utiliza la expresión del nivel de presión sonora (SPL) del siguiente modo:

$$SPL = 20 \log_{10} \left(\frac{p}{p_0} \right) \quad (4.8)$$

donde p_0 se define como la mínima presión acústica que se puede escuchar tomando un valor de 20×10^{-6} .

El término nivel de presión tiene como unidad de medida los decibelios y se utiliza para poder analizar dónde se encuentra la mayor intensidad de presión sonora.

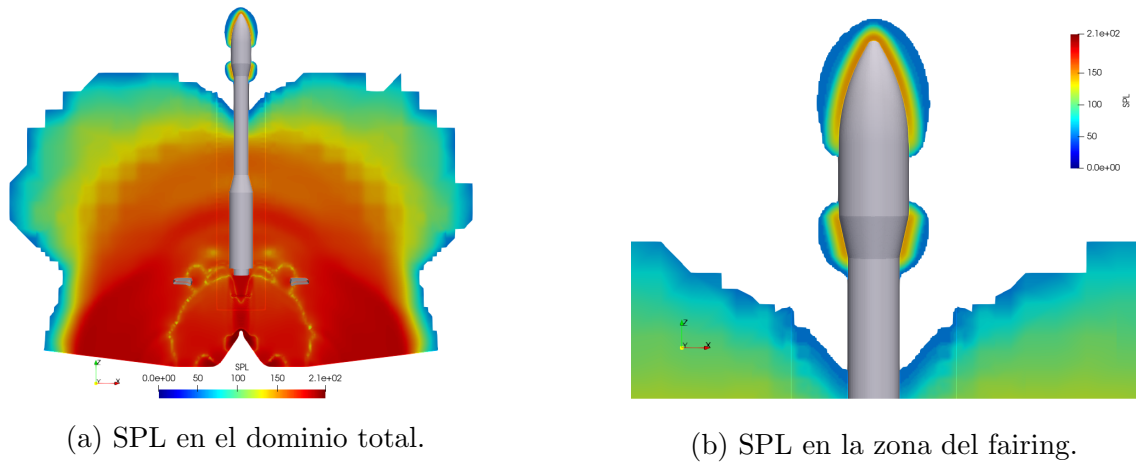


Figura 4.15: Nivel de presión sonora para la malla Overset.

Haciendo referencia al estudio previo del cual se ha mapeado la solución inicial, el SPL para el caso estático muestra como el nivel de presión cubre casi todo el cohete como se observa en Figura 4.15a.

Del mismo modo, en la Figura 4.15a se observa una diferenciación en el nivel de presión sonora entre el frente de la onda acústica y el frente originado por el movimiento del flujo que se propaga a velocidad supersónica. La zona alcanzada por el flujo representa un nivel de presión sonora cercano a 170 dB, mientras que la zona que no ha sido alcanzada por el flujo sino por la onda acústica tiene un nivel que ronda los 150 dB. Del mismo modo, se puede observar como en la zona del deflector y en el cambio de curvatura de la rampa se llegan a niveles de presión de 210 dB, siendo este valor el máximo en todo el dominio.

En la Figura 4.15b se puede observar como el efecto del movimiento del cohete influye en el nivel de presión sonora sobre el fairing. En este punto comienzan a propagarse unas ondas que encajan con las ondas que se producen en la Figura 4.14b. El nivel de presión sonora máximo en el fairing para el último paso calculado es de 100 dB. Se puede intuir como tras un cierto tiempo las ondas procedentes de la onda acústica y la alcanzada por el flujo interactuarán con estas nuevas, que surgen de la nariz del cohete y de la unión de las etapas 1 y 2. Del mismo modo se intuye que las ondas formadas en la punta del fairing se convertirán en ondas de choque.

Capítulo 5

Conclusiones

Este proyecto que se ha realizado forma parte de otro que está desarrollando un grupo de investigación de la UPV en colaboración con la Agencia Espacial Europea (ESA). Este proyecto surgió debido a la necesidad de conocer la generación y propagación del ruido durante la fase de despegue del cohete. Además, había una motivación adicional de poder reproducir unas condiciones más semejantes a las reales. Por lo tanto, este trabajo es parte del proceso de predicción de ruido en el despegue de VEGA.

El actual trabajo se puede dividir en tres bloques principales: análisis y comprensión del estudio previo y de las herramientas del software OpenFOAM, estudio sobre la influencia de la adición de la capa límite y, por último, desarrollo y análisis de un modelo CFD de una malla móvil con la herramienta OpenFOAM. Durante este proyecto se ha conseguido profundizar en el aprendizaje de esta herramienta. Cabe destacar que se ha hecho un gran esfuerzo para obtener unos resultados lo más realistas posibles teniendo en cuenta las limitaciones de cálculo computacional y escasez de referencias bibliográficas a día de hoy en cuanto a las mallas Overset.

En primer lugar, se ha investigado cuál es el efecto de añadir la capa límite a la estructura del deflector, donde impacta con mayor intensidad el chorro. Se ha podido comprobar cómo añadiendo este factor se puede estudiar con mayor detalle qué ocurre en las cercanías de las paredes de las geometrías. Un aspecto bastante importante a la hora de generar estas capas es el refinamiento del mallado. Si el refinamiento de la malla previo a incluir la capa límite no es óptimo, estas no se incluirán de forma completa en el mallado. Por ello, antes de realizar el mallado de las geometrías se debe estudiar qué tamaño de celda se buscará en la capa límite. De esta forma, se cumplirán los requisitos necesarios para generar la capa límite en su totalidad. Como ya se dijo en la Sección 4.15, la adición de la capa límite ahorra mucho cálculo computacional al reducir el número de celdas necesarias para reproducir los efectos en la pared.

El siguiente bloque trata del modelado de una malla móvil para poder simular el movimiento del cohete durante su despegue. En un primer momento, se barajaban dos opciones: una malla *MADR* (*Mesh Adaption-Dynamic Remeshing*) basada en el remallado del caso una vez

la distorsión de la malla, debida al movimiento del cohete, fuera muy alta y, por otro lado, una malla Overset basada en la superposición de dos mallas que se comunicarían mediante las celdas de interpolación.

Se eligió el segundo método ya que este permitía tener siempre una calidad de malla constante, por lo que no sería un motivo de preocupación un supuesto error en la solución debido a la calidad de esta. Sin embargo, se probó el primer método para comprobar cuál era más práctico. Gracias a esto, se llegó a la conclusión de que el primero, antes del remallado, empeoraba la calidad de la malla en gran medida debido a la distorsión y podía afectar a la solución del caso. Por otro lado, también se descartó debido a que el coste computacional del mallado, a pesar de ser un mallado local, era muy alto.

Durante el proceso de modelado de la malla Overset se observaron ciertas debilidades en este método para el caso actual que se está estudiando. El tamaño de las celdas de ambas mallas que se solapan deben ser del mismo orden, es decir, cuando la malla móvil se mueva dentro del dominio tiene que tener una concordancia de tamaño de celda con la malla estática en todo momento. Otro aspecto es que las celdas donde surge la fuente del flujo corresponde a un contorno de geometría, por lo que esta se identifica como celdas interpoladoras. Esto puede generar ciertas perturbaciones en el flujo que se emite desde la tobera y no presentar unos datos del todo reales. Pese a todo, este factor se podría llegar a mitigar con un mayor refinamiento de la malla.

En la etapa de cálculo se ha podido constatar cómo el cálculo de los pasos temporales era mayor que el de un caso estático, esto es debido a que se realizan dos cálculos. El primer cálculo se realiza en una malla y mediante las celdas interpoladoras se realiza el cálculo de la otra malla.

Pese a estos inconvenientes que surgieron en el proyecto, se han conseguido obtener ciertos resultados relevantes localizados en la punta del fairing. Estas soluciones muestran cómo se generan unas ondas de choque debido al movimiento del cohete tanto en la punta del fairing como en la sección entre la 1ª y 2ª etapa. Se espera que tras un mayor tiempo de computación los ondas en la punta del fairing se intensifiquen y se pueda analizar qué efectos originan sobre el cohete.

En conclusión, en este proyecto se ha conseguido modelar el movimiento del cohete VEGA durante la fase de despegue. Gracias a esta simulación se ha podido aportar nueva información al proyecto global de la investigación. Cabe recordar que el análisis CFD es fundamental ya que desde el punto de vista experimental es imposible obtener datos en ciertas regiones del dominio.

5.1. Proyectos futuros

Tras haber resuelto el caso de modelado de una malla móvil, se plantean nuevas ideas a desarrollar. En primer lugar, continuar la simulación con una nueva malla que tenga un mejor refinamiento y con una mayor potencia computacional. De este modo, se podrá analizar con mayor precisión los fenómenos que se originan en el fairing, como se ha comprobado en este proyecto, o en otra zona del dominio. Para ello, la simulación debe estar más avanzada y el cohete debe desplazarse una distancia mayor.

Por otro lado, se puede buscar la forma de introducir un fichero con las velocidades para cada tiempo de la simulación, ya que en este estudio se ha hecho una aproximación y se ha considerado que la velocidad era constante. Del mismo modo, se puede investigar si se puede introducir una ecuación en función del tiempo de simulación para evitar la interpolación entre dos valores temporales.

Otro aspecto interesante a estudiar sería la implementación del mallado *MADR* (*Mesh Adaption-Dynamic Remeshing*) con el objetivo de comparar las soluciones entre los dos modelos existentes de mallas móviles. De esta manera, se puede hacer una comparación con soluciones experimentales, en el caso de que se faciliten, y constatar qué modelo se ajusta más a la realidad.

Por último, un aspecto muy interesante a estudiar sería configurar un nuevo modelo de expulsión de los gases por la tobera, ya que en este modelo se asumen que se expulsa de forma constante. Asimismo, sería de gran interés estudiar si se generan nuevas fuentes de ruido debido a la producción de reacciones químicas originadas en la postcombustión de la combinación de la mezcla de gases a alta temperatura con el oxígeno del aire.

Capítulo 6

Pliego de condiciones

6.1. Condiciones del puesto de trabajo

Durante la realización de cualquier proyecto, el trabajador está sometido a una serie de condiciones que pueden intuir tanto en su salud como en su rendimiento. Debido a este hecho, es necesario regular estas condiciones para minimizar los riesgos laborales que puedan suponer. La normativa que se ajusta al tipo de actividad realizada en este caso es la recogida en el Real Decreto 488/1997 del 14 de abril, sobre disposiciones mínimas de seguridad y salud relativas al trabajo con equipos que incluyen pantallas de visualización (PVD). Para ser exactos en su definición, un puesto de trabajo con PVD's es .^{el} constituido por un equipo con pantalla de visualización provisto, en su caso, de un teclado o dispositivo de adquisición de datos, de un programa para la interconexión persona-máquina, de accesorios ofimáticos y de un asiento y mesa o superficie de trabajo”.

Se atenderá a cuatro variables para prever el tipo de riesgos a los que puede enfrentarse el trabajador:

- Tiempo de trabajo con la pantalla de visualización.
- Tiempo de atención requerida ante la pantalla, que a su vez puede ser continua o discontinua.
- Exigencia y grado de complejidad de la tarea realizada ante la pantalla.
- Necesidad de obtener una información de manera muy rápida.

Por otro lado, a partir de este tipo de actividades pueden surgir una serie de riesgos que se enumeran a continuación:

- Seguridad (contactos eléctricos). Higiene industrial.

- Iluminación.
- Ruido.
- Condiciones termohigrométricas.
- Ergonomía.
- Fatiga visual.
- Fatiga física
- Fatiga mental.

6.1.1. Lugares de trabajo

Los entornos donde se lleva a cabo la actividad deben atenerse a lo establecido en el Real Decreto 486/1997 del 14 de abril, sobre condiciones mínimas de seguridad y salud aplicables a los lugares de trabajo, además de a lo establecido específicamente en el Real Decreto 488/1997 ya reseñado sobre puestos de trabajo con PVD's.

6.1.2. Medidas de emergencia: vías y salidas de evacuación

La empresa debe haber adoptado medidas de emergencia en las que se incluyan las vías y salidas de evacuación en caso de que se declare una emergencia. Estas medidas deben darse a conocer a los trabajadores.

6.1.3. Condiciones de protección contra incendios

Todas las instalaciones contra incendios deben estar proyectadas, implantadas y mantenidas por empresas debidamente autorizadas por el organismo competente.

6.1.4. Instalación eléctrica

La instalación eléctrica debe estar proyectada, puesta en funcionamiento y mantenida por una empresa debidamente autorizada. Esta instalación debe evitar originar contactos con las personas, incendios y explosiones, ateniéndose para ello a lo establecido sobre tensiones y seguridad en los Reglamentos de Baja y Alta Tensión en vigor.

Según lo establecido para actividades con equipos que incluyen pantallas de visualización, se debe cumplir lo indicado a continuación:

- Reducir a niveles insignificantes toda la radiación electromagnética. Garantizar el adecuado mantenimiento de los cables y conexiones.

- Mantener separados los cables eléctricos de los telefónicos.
- Emplear longitudes de cable suficientes como para permitir futuros cambios.
- Facilitar el acceso y mantenimiento de los cables sin interrupción de las actividades del trabajo, evitando ubicarlos sobre superficies donde puedan ser pisados o sometidos a condiciones adversas.

6.1.5. Condiciones termohigrométricas

El Real Decreto 488/1997 sobre puestos de trabajo con equipos con pantallas de visualización establece las siguientes magnitudes para garantizar el bienestar y confort térmico de los trabajadores:

- La temperatura operativa de confort debe mantenerse en el rango de 23 a 26 ° C verano y de 20 a 24 ° C en invierno.
- La humedad relativa del aire debe mantenerse siempre entre el 45 % y el 65 %.

6.1.6. Iluminación

La iluminación puede ser natural o artificial, a pesar de que es recomendable que sea natural. Sin embargo, dado que su intensidad varía con la hora del día, con la meteorología y con las estaciones, generalmente debe complementarse con iluminación artificial. Esta suele ser general, debiendo complementarse con iluminación localizada. En cualquier caso, no deben producirse deslumbramientos, ni contrastes excesivos.

Los niveles de iluminación serán suficientes para el correcto desarrollo de las tareas que se realicen en el puesto de trabajo. Además, la ubicación y orientación del puesto y la pantalla deben evitar reflejos y deslumbramientos.

6.1.7. Ergonomía: diseño del puesto de trabajo

El diseño del puesto de trabajo está directamente relacionado con los problemas posturales. Si se tiene en cuenta que el trabajo con pantallas de visualización se caracteriza por posturas estáticas prolongadas, se puede deducir que los efectos de estas posturas se agravan cuando el diseño del puesto de trabajo es incorrecto.

Asiento La altura del asiento debe ser ajustable dentro del rango necesario para el conjunto de usuarios, así como su profundidad e inclinación. El respaldo debe tener una suave prominencia para dar apoyo a la zona lumbar. Se recomienda la utilización de sillas dotadas de ruedas, aunque la resistencia de las ruedas a iniciar el movimiento debe

evitar desplazamientos involuntarios en suelos lisos y con actividades de tecleo intensivo.

Mesa La superficie de la mesa de trabajo debe ser poco reflectante y de dimensiones suficientes como para permitir movimientos de trabajo y cambiar de postura. Además, debe permitir una colocación flexible de la pantalla, del teclado, de los documentos y del material accesorio.

Colocación de la pantalla La distancia óptima entre el usuario y la pantalla, en lo referente al confort visual, debe ser de 45 a 75 cm. La pantalla debe colocarse de manera que su área útil pueda ser vista bajo ángulos comprendidos entre la línea de visión horizontal y la trazada a 60 ° bajo la horizontal. En el plano horizontal, la pantalla debe estar colocada dentro de un ángulo de 70 °, dentro del campo de visión del usuario.

Pantalla La imagen de la pantalla debe ser estable, sin fenómenos de destellos. La luminosidad y el contraste debe ser fácilmente regulable por el usuario. Asimismo, la pantalla debe ser orientable e inclinable a voluntad.

Teclado El teclado debe estar inclinado y ser independiente de la pantalla, de modo que permita al trabajador adoptar una postura cómoda, que no provoque cansancio en brazos y manos.

Equipos de protección individual Es recomendable el uso de protectores de pantalla. Estos consisten en un cristal polarizado, sostenido por un marco, que se superpone al cristal de la pantalla. De este modo se consigue proteger la vista de los usuarios, además de mejorar las prestaciones ofrecidas por la pantalla, relacionadas con el contraste y la claridad.

6.1.8. Ruido

El Real Decreto 1316/1989 del 27 de octubre establece que el empresario debe evaluar la exposición de los trabajadores al ruido, con el objeto de determinar si se superan los límites establecidos en esta norma y en su caso aplicar las medidas procedentes.

Normalmente, en los lugares de trabajo donde se emplean PVD's no suelen existir altos niveles sonoros, sin embargo pueden existir niveles medios que pueden molestar y perturbar la atención de los trabajadores. Por lo tanto, el nivel sonoro en estos puestos de trabajo debe de ser tan bajo como sea posible. Para conseguirlo, se deben utilizar equipos con una emisión sonora mínima y optimizar la acústica del local.

La directiva establece que para tareas difíciles y complejas que requieren una especial concentración, el nivel sonoro equivalente (L_{aeq}) no debe exceder 55 dB(A).

6.2. Condiciones de los recursos informáticos

6.2.1. Introducción

Para llevar a cabo los cálculos requeridos por el presente proyecto, se hace necesario el uso de recursos informáticos de altas prestaciones. Estos recursos informáticos pueden ser divididos en dos grupos: hardware y software.

Hardware Dadas las exigencias de los cálculos a realizar, se requieren equipos informáticos con gran potencia de cálculo y memoria. Estos son estaciones de trabajo conectadas en red que permiten acceder a ellas de forma remota y trabajar de forma ininterrumpida, sin saturar el terminal de acceso. Estas máquinas requieren de un mantenimiento periódico para asegurar su perfecto funcionamiento de forma ininterrumpida y con un alto rendimiento.

Software Software Para la realización de las tareas requeridas es necesario el uso de programas informáticos de distintos tipos, como son programas CAD, malladores y programas de simulación numérico y postproceso. Dado que se ha optado por el uso de software comercial, con el fin de disponer de soporte técnico en caso de problemas, es necesario poseer las correspondientes licencias de uso en vigor.

6.2.2. Condiciones de hardware

Como hardware han sido utilizados un equipo portátil, un ordenador de sobremesa y un cluster de cálculo. El equipo portátil y el ordenador de sobremesa han servido como terminal de acceso al cluster de cálculo. El portátil se ha usado además para realizar tareas que no requiriesen altas prestaciones mientras que en el ordenador de sobremesa también se han realizados cálculos menos pesados. La interconexión de los equipos se ha llevado a cabo mediante una red Ethernet de alta velocidad. A continuación se describen las especificaciones del equipo portátil, del ordenador de sobremesa y del cluster de cálculo.

Ordenador Portátil:

- Procesador: Intel Core i7, 2.0 GHz. Número de procesadores: 2
- Memoria RAM: 8 GB
- Tarjeta gráfica: Intel(R) HD Graphics 4000 .
- Unidad de almacenamiento: Disco duro híbrido HDD-SSD de 400 GB.
- Sistema operativo: Windows 10 Professional N. Arquitectura: 64 bits.

Cluster de cálculo de la UPV

El cluster para proceso paralelo masivo de la UPV esta compuesto por

- 72 nodos tipo BX920S3 de Fujitsu. Se cuenta con un total de 4 chasis los cuales están compuestos por 18 nodos cada uno. Cada nodo del chasis esta compuesto por
 - 2 procesadores de 8 cores y 64 Gb de RAM DDR3.
- 56 nodos del modelo R424E4 de la marca BULL .
 - 48 nodos con 2 procesadores de 8 cores, 2,4 Ghz, 20 Mb de caché L3 y 64 Gb de memoria.
 - 8 nodos con 2 procesadores de 12 cores, 2,5 Ghz, 30 Mb de caché L3 y 128 Gb de memoria.
- 27 nodos de la marca Dell Power Edge R640 que aportan una potencia de cálculo de 3132 GFLOPS. Estos nodos están compuestos por diferentes unidades
 - 24 nodos con 2 procesadores de 18 cores, 3 Ghz, 25 Mb de caché L3 y 192 Gb de memoria.
 - 3 nodos con 2 procesadores de 18 cores, 3 Ghz, 35 Mb de caché L3 y 768 Gb de memoria.

6.2.3. Condiciones del software

A continuación se detalla el tipo de software empleado. Se trata de software libre.

Software de CAD

El software de diseño asistido por ordenador elegido ha sido *Autodesk*. Sus requerimientos son suficientemente bajos como para realizar todo el trabajo de CAD en el equipo portátil.

6.2.4. *Software de simulación numérica*

El código elegido para llevar a cabo las simulaciones numéricas es OpenFOAM v6 y v1912.

Capítulo 7

Presupuesto

7.1. Introducción

En este capítulo se va a especificar cual es el valor económico del trabajo invertido en este proyecto. Para esto, se tendrá en cuenta tanto el número de horas dedicadas por el personal que ha trabajado en la investigación como los recursos materiales que han sido necesarios. Para poder valorar el esfuerzo y dedicación de alguna manera, se ha decidido utilizar la hora de trabajo como sistema de esfuerzo. En cuanto al aspecto económico, se empleará el euro por hora de trabajo.

7.2. Tareas realizadas

A continuación, se enumerarán las tareas que se han realizado con el fin de realizar el proyecto:

Actividad 1: Recopilación y análisis de información y entendimiento del software

Investigación necesaria para poder realizar el proyecto así como búsqueda de nuevas funciones a implementar en el trabajo. También se realiza un estudio del software a utilizar y del proyecto previo sobre el que se basa.

Actividad 2: Generación de la capa límite del caso estático

Implementación de la agregación de la capa límite al deflector para poder estudiar con mejor precisión las ondas de choque y los puntos de parada del flujo. Se realizará la implementación sobre la malla del estudio previo.

Actividad 3: Configuración y ejecución del caso estático con la capa límite

Ejecución del cálculo del caso estático con los modelos físicos que más se ajustan a los fenómenos reales. Posteriormente, monitorización del avance del cálculo ante posibles problemas tales como la divergencia del caso.

Actividad 4: Generación de la malla con el método Overset

Confección de las diferentes mallas necesarias para el caso Overset. Es muy importante tener en cuenta que la zona común de ambas mallas tiene que tener el mismo tamaño de celda. Si no se cumple la regla previa, el cálculo falla debido a que no puede realizar la interpolación entre mallas.

Actividad 5: Configuración y ejecución del caso Overset

Ejecución del caso Overset , se utilizan los modelos físicos del caso estático ya que son los más se aproximan a la realidad. Posteriormente, monitorización del cálculo por si surge cualquier problema con la convergencia de la solución o por si hay que cambiar cualquier parámetro de control durante el cálculo.

Actividad 6: Postprocesado de los resultados

Análisis de los resultados obtenidos durante los cálculos de los casos estático y Overset. Del mismo modo, se realizan las comparaciones y análisis pertinentes para constatar primero, el efecto de la adición de la capa límite y segundo, el efecto que causa el movimiento del cohete.

Actividad 7: Elaboración de la memoria

Se realiza la redacción y confección de la memoria que se va a entregar a los tutores, los directores y el tribunal durante la defensa del proyecto.

7.3. Recursos

En este apartado se definen los recursos utilizados en el proyecto. Estos se van a dividir en dos grupos: personal y material. Se han separado de dicha manera ya que son los dos únicos

factores relevantes en el proyecto. Del mismo modo, estos recursos se pueden dividir en los siguientes:

Recursos	
Personal	Materiales
Ingeniero	Equipo informático portátil
Co-Autora	Equipo informático de mesa
Doctor	Lugar de trabajo
	Cluster de cálculo
	Material de oficina

Durante la investigación de este proyecto han supervisado y colaborado dos Doctores de la Universitat Politècnica de Valencia y una co-Autora, además de un técnico de investigación que es el que realiza el proyecto en su totalidad.

7.4. Presupuestos

7.4.1. Personal técnico

El presupuesto que se necesita en relación al personal técnico y para cada actividad se basa en el número de horas invertidas y en el coste de cada una de ellas. Teniendo en cuenta que el coste por hora de un ingeniero es de 15 /h y que el de un Doctor es de 30 /h, se pueden realizar los cálculos pertinentes en función del tiempo invertido en las actividades como se puede ver a continuación:

		Horas de trabajo	Coste (€/h)	Coste (€)
Ingeniero	Act. 1	60		900
	Act. 2	60		900
	Act. 3	240		3600
	Act. 4	300	15	4500
	Act. 5	200		3000
	Act. 6	60		900
	Act. 7	100		1500
Doctor	Act. 1	5		150
	Act. 2	15		450
	Act. 3	15		450
	Act. 4	40	30	1200
	Act. 5	40		1200
	Act. 6	30		900
	Act. 7	30		900
			Coste Total	20550 €

Tabla 7.1: Presupuesto del personal

7.4.2. Materiales técnicos: equipo informático y material de oficina

El equipo informático y el material de oficina que se ha utilizado durante el proyecto ha sido:

- **Ordenador de mesa:** El ordenador de mesa proporcionado por la Universitat Politècnica de Valencia (UPV) tuvo un precio de 6000 €. La amortización de dicha máquina es a 5 años, pero como el tiempo entre el que se adquirió y cuando se comenzó el proyecto es mayor se considera que ya está amortizado.
- **Ordenador portátil:** El ordenador portátil que se ha utilizado para poder trabajar en conexión remoto fue comprado en 2014 por el autor de este proyecto. Por lo tanto, el coste de este equipo también está amortizado.
- **Cluster de cálculo:** Han sido utilizados 128 procesadores durante 4 meses, 32 procesadores durante 2 meses y 256 durante medio mes. Considerando que el precio de uso es 1 cent/hora por procesador, se ha invertido un total de 5068.8 €.
- **Estación de trabajo:** El lugar de trabajo pertenece a la Universitat Politècnica de Valencia (UPV) y entra en los gastos indirectos del proyecto.
- **Tablet:** Se ha hecho uso de una Tablet Huawei T10 principalmente durante la búsqueda de información de datos y en la monitorización de la simulación del cálculo. Este

modelo de tablet tiene un precio de 200 €, se considera amortizada ya que se compró hace tiempo y su uso ha sido considerable previo al inicio del proyecto.

- **Material de oficina:** Se utilizaron dos libretas de texto, de coste unitario aproximado 2 €, 2 bolígrafos, de coste unitario 1.5 € y un pendrive de coste 6 €. La cantidad a la que asciende el coste del material de oficina durante el proyecto fue de 13 €.

	Horas de uso	Coste (€)
Ordenador de mesa	200	0
Ordenador portátil	1320	0
Cluster de cálculo	506880	5068.8
Estación de trabajo	200	0
Tablet	100	0
Material de oficina	-	13
Coste Total		5081.8 €

Tabla 7.2: Presupuesto del material técnico

7.4.3. Software

Los principales softwares utilizados y su coste han sido

- **OpenFOAM.** La licencia de ambas versiones es gratuita, de modo que no supone coste.
- **Autodesk.** La licencia es gratuita para estudiantes, de modo que no supone coste.
- **Paraview.** La licencia es gratuita, de modo que no supone coste.

7.5. Presupuesto final

En el presupuesto final se han añadido un 25 % de gastos indirectos que conlleva un trabajo de estas características, un 6 % de beneficio industrial y un 21 % de IVA.

		Coste (€)
Personal técnico		20550
Materiales técnicos		5081.8
Software		0
COSTE TOTAL NETO		25631.8
Gastos indirectos	25 %	6407.95
Beneficio industrial	6 %	1537.91
IVA	21 %	5382.68
TOTAL		38960.34 €

Tabla 7.3: Presupuesto Final

Por tanto, el coste total bruto del estudio de la adición de capa límite a un caso estático y el estudio dinámico del ascenso del cohete VEGA, se eleva a:

**# Treinta y ocho mil novecientos sesenta euros con treinta y cuatro céntimos
(38960.34 €) #**

Bibliografía

- [1] Hernández Alayeto, Iván. Aplicación del código OpenFOAM al control activo de capa límite, 2016.
- [2] Moreno Rodríguez, Cristobal Daniel y otros. Simulación numérica de flujos convectivos a altos números de rayleigh en sistemas pasivos de climatización, 2008.
- [3] Equipe ESS. Dinámica de Fluidos Computacional: ¿que es?, 2016.
- [4] OpenFOAM. OpenFOAM: User Guide v2006 The open source CFD toolbox, 2016.
- [5] engys. A Comprehensive Tour of snappyHexMesh, 2012.
- [6] Casari, N., Fadiga, E., Pinelli, M., Suman, A., Ziviani, D. CFD Simulations of Single-and Twin-Screw Machines with OpenFOAM. *Designs*, 4(1), 2, 2020
- [7] Autio, E. Dynamic overset CFD simulation of a pneumatic impact device (Master's thesis), 2018.
- [8] OpenFOAM, ESI group. OpenFOAM v1706: New and improved numerics, 2017.
- [9] SimScale Staff. CFD: PIMPLE Algorithm, 2018.
- [10] Jauregui Lorda, Matías, y Ana Elena Scarabino. “Estudio de pluma de motor cohete impactando sobre diferentes superficies deflectoras de gases”. *Anales AFA*. Vol. 23. Asociación Física Argentina, 2012.
- [11] Lemaire, Sébastien, Guilherme Vaz, and Stephen Turnock. “On the need for higher order interpolation with overset grid methods.”, (2019): 1-6.

- [12] Tisovska, P.: Description of the overset mesh approach in ESI version of OpenFOAM. In Proceedings of CFD with OpenSource Software, 2019.
- [13] Quintana, Diego. El cohete Vega dispara las misiones de la ESA. El cultural, 2012.
- [14] VEGA (P80 based). Gunter's Space Page, 2020.
- [15] Parameter Definitions - solidBodyMotionFvMesh. OpenFOAM Wiki, 2020.
- [16] Muñoz-Barranco, José. "Evaluación de la eficiencia de métodos acoplados en la resolución de problemas fluidomecánicos con OpenFOAM.", 2015.
- [17] Richmond Navarro, Gustavo. Modelos de turbulencia introductorio, s.f.
- [18] VEGA Photos. Recuperado de [VEGA_Gallery](#) .
- [19] VEGA Launch. Recuperado de [Launch of VEGA](#) .
- [20] D Wilcox. Turbulence Modeling for CFD. DWC Industries, 2010.
- [21] Ciencia y tecnología en Rtve.es. El cohete Vega de la ESA despega con éxito para cumplir su primera misión. RTVE, 2012.
- [22] Despega la nueva nave espacial Vega con el satélite Xatcobeo en su interior. EL IMPARCIAL, 2012.
- [23] M Olmo R Nave. Rocket Principles. HyperPhysics, sf.
- [24] Gowdy, R.H. The physics of perfect rockets. Am. J. Phys. 63 (3) March 1995, pp. 229-232