

Universidad Nacional del Nordeste

Facultad de Ingeniería

Carrera:

Maestría en Ciencias de la Ingeniería



**Modelado Numérico de Interacción Fluido-Estructura
con Metodología de Mallas Solapadas**

T E S I S

Para obtener el título de:

Magíster en Ciencias de la Ingeniería

Presenta:

Ing. Marcelo Italo Adotti

Director de tesis:

Dr. Ing. Hugo Guillermo Castro

Co-director de tesis:

Dr. Ing. Mario Eduardo De Bortoli

Resistencia, Chaco Argentina

2022



**Modelado Numérico de Interacción
Fluido-Estructura con Metodología de Mallas
Solapadas**

Tesis de Maestría en Ciencias de la Ingeniería.

Tesista: Ing. Marcelo Italo Adotti
Director: Dr. Ing. Hugo Guillermo Castro
Co-director: Dr. Ing. Mario Eduardo De Bortoli

Diciembre de 2022

Miembros del comité evaluador

Dr. Ing. Horacio Javier Aguerre (CIMEC-CONICET-UNL)

MSc. Ing. Gerardo Franck (UNL)

Dr. Ing. Juan Manuel Podestá (UNNE-FI)

A mi querida Vanesa.
A mis padres.

Agradecimientos

En estas letras quiero agradecer la constante ayuda y consejos para mejorar mi formación, a todos los integrantes del Laboratorio de Aerodinámica y el Laboratorio de Mecánica Computacional de la Facultad de Ingeniería de la UNNE. A mis directores Guillermo y Mario que han sido actores fundamentales para que la presente tesis fuera posible, compartiendo tiempo y conocimiento valioso, no solo en estas instancias, sino durante varios años de trabajo compartido previo.

A mis compañeros de cursado Juan Manuel, Nelson y Julián, con los cuales he compartido muchas horas y vivencias, han sido una fuente incansable de charlas enriquecedoras y aliento.

A Vanesa que con cariño y amor en todo momento me acompañó, entendió e impulsó a continuar para alcanzar el objetivo final. A mis padres que me han enseñado la importancia de la formación y el trabajo siempre.

Y a todos mis amigos con los que he compartido charlas y momentos de la vida durante todo este período.

Resumen

El fenómeno de interacción fluido-estructura (FSI, por Fluid Structure Interaction) se encuentra presente en vastos campos de la naturaleza y desarrollos tecnológicos creados por el ser humano. El modelado numérico de las ecuaciones que describen al fenómeno multifísico resulta ser una herramienta cada vez más útil debido al incremento de potencia de cálculo de los ordenadores. Constituyendo así una técnica de análisis complementaria a los ensayos experimentales.

Existen varios métodos numéricos para modelar el fenómeno, algunos de ellos son: mallas deformables dinámicas descritas en un marco referencial Arbitrario Lagrangiano Euleriano (ALE, por Arbitrary Lagrangian - Eulerian), Método de Frontera Inmersa (IBM, por Immersed Boundary Method), Método de Volumen Fluido (VoF) y el Método de ajuste de nivel (LSM, por Level Set Method). Cada uno de estos posee virtudes y debilidades según las características particulares del fenómeno FSI que se desee simular.

Otro método es el de mallas solapadas (OM, por Overset Mesh) el cual posee ventajas respecto a los antes citados, para simular grandes desplazamientos e inclusive trayectorias completas, sin requerir la deformación de sus mallas, evitando así en casos extremos costosos procesos de remallado.

El objetivo de esta tesis es estudiar la aplicación OM disponible en el entorno de manejo y manipulación de campos denominado OpenFOAM, evaluando distintos casos ingenieriles de flujo externo. Se analizan dominios bidimensionales y tridimensionales, con flujos que escurren a distintos números de Reynolds. Además, se estudia el algoritmo con el cual es implementado el solver para OM y se le introducen modificaciones para medir los tiempos insumidos por las principales tareas, evaluando también métricas de paralelismo. Finalmente es evaluado el comportamiento de las funciones de interpolación que utiliza OM y la conservación global del flujo.

Los resultados obtenidos permiten concluir de que OM es un método muy versátil para el modelado de la FSI, debiéndose considerar siempre su implementación con una óptima calidad de mallado.

Abstract

The phenomenon of Fluid-Structure Interaction (FSI) is present in vast fields of nature and technological developments created by human beings. Numerical modeling of the equations that describe the multi-physical phenomenon turns out to be an increasingly useful tool due to the development in computing power of computers. Thus constituting a complementary analysis technique to the experimental tests.

There are several numerical methods to model the phenomenon, some of them are: dynamic deformable meshes described in an Arbitrary Lagrangian Eulerian frame of reference (ALE), Immersed Boundary Method (IBM), Fluid Volume (VoF) and Level Set Method (LSM). Each one of these has strengths and weaknesses depending on FSI phenomenon particular characteristics to be simulated.

Another method is that of overlapping meshes (OM) which has advantages over the aforementioned, to simulate large displacements and even complete trajectories, without requiring the deformation of their meshes, thus avoiding costly re-meshing processes in extreme cases.

The aim of this thesis is to study the OM application implemented in the OpenFOAM suite, evaluating different engineering cases of external flow. Two-dimensional and three-dimensional domains are analysed, with flows at different Reynolds numbers. In addition, OM's solver algorithm is studied, also modifications are introduced to measure the time consumed by the main tasks, evaluating parallelism metrics. Finally, the behaviour of the interpolation functions used by OM and the global conservation of the flow are evaluated.

The results obtained allow us to conclude that OM is a very versatile method for FSI modelling, and its implementation should always be considered with an optimal mesh quality.

Índice

1. Introducción	13
1.1. Antecedentes y Motivación	13
1.2. Objetivos generales y particulares	14
2. Interacción fluido-estructura FSI	16
2.1. Descripción del método ALE	17
3. Problema matemático - acople y solución	21
3.1. Malla móvil	29
4. Flujo Turbulento y modelo de cierre adoptado	31
5. Mallas solapadas OM	37
5.1. Formulación matemática de mallas solapadas	38
5.1.1. cellVolumeWeight	42
5.1.2. inverseDistance	43
5.1.3. leastSquares	44
5.2. Precisión y conservación de propiedades	45
5.3. Extensión a problemas de frontera móvil	46
6. Análisis de sensibilidad y configuración del solver	47
7. Modelos analizados y técnicas de mallado	50
7.1. Cilindro 2D	50
7.1.1. Malla de fondo	50
7.1.2. Malla solapada	50
7.2. Perfil Alar 2D	53
7.2.1. Malla de fondo	53
7.2.2. Malla solapada	53
7.3. BARC	56
7.3.1. Malla de fondo	56
7.3.2. Malla solapada	57
7.4. Cuerpo de Ahmed	60
7.4.1. Malla de fondo	61
7.4.2. Malla solapada	63

8. Configuración y Resultados	65
8.1. Cilindro 2D	65
8.1.1. Análisis de resultados	66
8.1.2. Coeficientes aerodinámicos	67
8.1.3. Frecuencia de vórtices - Número de Strouhal	68
8.1.4. Campos de velocidad en la zona de solape de mallas	68
8.2. Perfil alar 2D	69
8.2.1. Análisis de funciones de interpolación	69
8.2.2. Métricas del solver	71
8.2.3. Campos de velocidad	74
8.2.4. Análisis de resultados	75
8.3. BARC	76
8.3.1. Coeficientes aerodinámicos	76
8.3.2. yPlus - Balance de flujo	78
8.3.3. Métricas del solver	82
8.3.4. Análisis de resultados	83
8.4. Cuerpo de Ahmed	84
8.4.1. Coeficientes aerodinámicos	84
8.4.2. yPlus	86
8.4.3. Análisis de residuos, perfil del solver y conservación global	87
8.4.4. Iso superficies de vorticidad y Campo de velocidad en la estela	88
8.4.5. Análisis de resultados	91
9. Conclusión	93
10. Referencias	94
11. Anexo	98
11.1. Computación de Alto Rendimiento HPC	98
11.1.1. Escalabilidad fuerte	98
11.1.2. Métricas y el OM	98
11.2. overPimpleDyMFoamTimed	99

Índice de figuras

1. Diagrama de Collar	16
2. Diagrama de la caracterización y clasificación de solvers y formas de acople utilizado en el problema FSI.	17
3. Descripción esquemática del problema de dominio	20
4. Diagrama de flujo, del solver utilizado overPimpleDyMFoam.	24
5. Diagrama de flujo del solver de acople de presión-velocidad PIMPLE.	25
6. Espectro de velocidad y términos de descomposición.	32
7. Flujo cortante 2D	32
8. Espectro de energía turbulenta $E(\kappa)$ y número de onda κ . Rangos de modelado y resolución de la turbulencia para diferentes técnicas. Líneas horizontales con guiones: Modelado; Líneas horizontales negras: Resolución (Arvidson, 2017).	33
9. Vista esquemática del dominio definido en zonas.	37
10. Vista esquemática de los dominios mallados.	37
11. Campo de velocidad obtenido.	38
12. Dominio \mathcal{D} del problema	39
13. Mallas solapadas $\mathbb{M} = \mathcal{M}_1 + \mathcal{M}_2$	39
14. Detalle de región solapada y definición de plantillas	40
15. Ejemplo de esquema de interpolación llevado a cabo en el dominio \mathcal{C}	41
16. Ejemplo en 1D de malla solapada con interpolación implícita y explícita.	42
17. Esquema implementación CVWI. Trazo negro: Celdas donantes. Trazo rojo: Celda receptora	43
18. Esquema implementación IDW. Trazo negro: Celdas donantes. Trazo rojo: Celda receptora	44
19. Geometría 2D caso analizado	47
20. Configuración sin relajación ni control de errores residuales	48
21. Configuración mixta	48
22. Configuración con relajación y control de errores residuales	48
23. Campo de velocidades 2D obtenida con la configuración especificada en Fig. (22b)	49
24. Condiciones de borde impuestas en la malla de fondo	51
25. Distribución de celdas utilizadas en la malla de fondo	51
26. Malla solapada con condiciones de borde y refinamiento de mallado	52
27. Distribución de celdas utilizadas en malla solapada	52
28. Detalle región de solape	52
29. Perfil NACA0012	53

30. Condiciones de borde impuestas en la malla de fondo	54
31. Distribución de celdas utilizadas en la malla de fondo	54
32. Malla solapada con condiciones borde impuestas	55
33. Distribución de celdas utilizadas en malla solapada	55
34. Detalle de la zona solapada	55
35. Geometría del BARC. Relación de aspecto 5:1	56
36. Dominio computacional malla de fondo	57
37. Densidades malla de fondo	57
38. Tipos de celdas en la malla de fondo	57
39. Dimensiones dominio malla solapada, discretización y condiciones de borde	58
40. Configuración de celdas en la capa límite	58
41. Distribución de celdas OM en malla solapada	59
42. Detalle región solapada entre mallas	59
43. Cuerpo de Ahmed - slant 35°	60
44. Dimensiones, regiones de refinamiento y condiciones de borde	61
45. Detalle regiones de mallado en varios cortes	62
46. Celdas utilizadas en la plantilla OM	63
47. Condiciones de borde impuestas	63
48. Celdas utilizadas en la plantilla OM y densificación de malla en capa límite	64
49. Zona de solape entre mallas	64
50. Serie temporal de coeficientes aerodinámicos para cilindro fijo	67
51. Serie temporal de coeficientes aerodinámicos para frecuencia de excitación $\omega = 1,04 \text{ rad/s}$	67
52. Espectros de potencia para ambos casos de excitación	68
53. En las figuras a) y b) se presentan los campos de velocidad resultantes para ambas estrategias de mallado Alletto (2022) (circular) y este trabajo (rectangular).	68
54. Caso estático. Contraste de coeficientes aerodinámicos según diferentes métodos de interpolación y referencia bibliográfica.	70
55. Caso móvil. Contraste de coeficientes aerodinámicos según diferentes méto- dos de interpolación.	70
56. Métricas de escalabilidad clúster IMIT	72
57. Métricas de escalabilidad clúster Pirayú - CIMEC	73
58. Estrategias distribución de celdas	74
59. Campos de velocidad para distintos pasos de tiempo	75
60. Función senoidal de movimiento prescripto	77
61. Historia temporal de coeficientes aerodinámicos	78

62. Historia temporal del coeficiente y^+ de las primeras celdas de la capa límite sobre el BARC.	78
63. Historia temporal del balance de flujo ϕ .	79
64. Campos de velocidad para $t = 2,5$ seg.	79
65. Campos de velocidad en la región de solape	80
66. Campo de velocidades para los distintos ángulos de ataque barridos α .	81
67. Distribución de celdas por procesador.	82
68. Residuos finales y porcentaje de tiempo insumido en tareas por el solver overPimpleDyMFoamTimed	82
69. Coeficientes Aerodinámicos: a) arrastre C_d , b) sustentación C_l , c) lateral C_s , d) rolido CM_{roll} , e) cabeceo CM_{pitch} y f) guiñada CM_{yaw} .	86
70. Coeficientes Aerodinámicos: C_d y C_l vs. ángulo de guiñada β . Meile et al. (2016) Adotti et al. (2017)	86
71. Distribución de y^+ en el cuerpo de Ahmed para 0° , $12,5^\circ$ y 25° .	87
72. Historia temporal del coeficiente medio yPlus de las primeras celdas de la capa límite del cuerpo de Ahmed.	87
73. Residuos finales y porcentaje de tiempo insumido en tareas por el solver overPimpleDyMFoamTimed	88
74. Error porcentual del flujo ϕ entre entrada y salida del dominio computacional.	88
75. Contornos de iso-superficie para distintos ángulos de guiñada β en el tiempo.	90
76. $Q = 500$ flujo cruzado $\beta = -12,5^\circ$, $t = 0,785$ seg y esquema de vórtices dominantes reportado por Meile et al. (2016).	90
77. a) Campo de velocidad normalizado en la estela y b) Q-criterion 500 para $\beta = 0^\circ$	91
78. a) Campo de velocidad normalizado en la estela y b) Q-criterion 500 para $\beta = -12,5^\circ$	91

Índice de tablas

1. Condiciones de borde e iniciales	66
2. Comparación del coeficiente de arrastre medio $C_{d,medio}$ y el máximo coeficiente de sustentación $C_{l,max}$, para diferentes frecuencias de oscilación ω .	66
3. Condiciones de borde e iniciales	69
4. Tiempos insumidos en las pruebas para los distintos casos y funciones de interpolación.	70
5. Métricas clúster IMIT	72
6. Perfil de procesos overPimpleDyMFoamTimed clúster IMIT	72
7. Métricas clúster Pirayú - CIMEC	73

8. Perfil de procesos overPimpleDyMFoamTimed cluster Pirayú - CIMEC . . .	73
9. Condiciones de borde e iniciales	76
10. Resultados de coeficientes aerodinámicos	77
11. Condiciones de borde e iniciales.	84

1. Introducción

1.1. Antecedentes y Motivación

El estudio del comportamiento de estructuras de grandes y pequeñas dimensiones sometidas a la acción del viento de Capa Límite Atmosférica (CLA), requieren de un análisis multifísico para evaluar el fenómeno de interacción fluido - estructura (FSI, por Fluid Structure Interaction). En la Ingeniería Aeronáutica este fenómeno ha sido extensamente analizado y luego del incidente del puente Tacoma Narrows fue incorporado como proceso de análisis en el área de la Ingeniería Civil. La construcción de cuerpos pequeños cada vez más livianos y esbeltos, medios de transporte motriz, tendidos de cables para energía eléctrica, entre otros, también deben considerarse como estructuras sensibles a estos fenómenos de interacción. Los códigos de cálculo estructural abordan las cargas dinámicas usualmente con fórmulas empíricas, verificando que los parámetros relevantes no excedan valores extremos. En caso de no ser posible aplicar esa metodología de cálculo, recomiendan evaluar el problema en forma particular a fin de mejorar la descripción del fenómeno.

En la búsqueda del conocimiento que permita evaluar con mayor exactitud el fenómeno, la comunidad científica abordó en las últimas décadas el estudio de los fenómenos de FSI, generando profusas publicaciones que analizan el comportamiento dinámico del multisisistema, sus causas y origen estableciendo una serie de conjeturas tendientes a mitigar los efectos dañinos sobre las estructuras.

En la actualidad una rama importante de la ciencia se ocupa de modelar la FSI utilizando herramientas numéricas mediante el uso de códigos propios, códigos abiertos o bajo licencia de Dinámica de Fluidos Computacional (CFD, por Computational Fluid Dynamics). El avance de las prestaciones de los equipos computacionales, permitió abordar problemas de ingeniería con mayor detalle y aumentando la exactitud, en contraparte el costo computacional es elevado y los tiempos de cálculo extensos. Para mitigar esa problemática, muchas veces se requieren aplicar técnicas de Computación de Alto Rendimiento (HPC, por High Performance Computing) para resolver los grandes sistemas de matrices, característicos del problema matemático que surge del modelado del fenómeno multifísico.

En líneas generales el problema físico se reduce a modelar el flujo cuando escurre sobre elementos de condiciones de borde móviles como hélices, turbinas, cuerpos flexibles o geometrías complejas móviles (Laws et al., 2019). Una de las técnicas más utilizadas para el modelado del fenómeno FSI es el de mallas deformables-dinámicas descritas en un marco referencial Arbitrario Lagrangiano Euleriano (ALE, por Arbitrary Lagrangian - Eulerian) (Trépanier et al., 1991). En casos extremos esta técnica pone en compromiso la calidad del mallado al deformar una parte de ella para permitir el movimiento de la condición de borde móvil que puede ser un sólido rígido o flexible, debiendo recurrir a limitaciones de movimiento, zonas de deformación controlada o remallado para evitar efectos difusivos por baja calidad de sus elementos. Es por ello que su aplicación presenta una desventaja cuando la frontera móvil posee grandes desplazamientos o trayectorias que evaluar dentro de un dominio.

Por otra parte existen métodos opcionales que pueden procesar grandes desplazamientos de frontera móvil sin necesidad de remallado, algunos de éstos son: el Método de Frontera Inmersa (IBM, por Immersed Boundary Method) (Peskin, 2002) el cual posee

algunos problemas asociados a imposición de la condición de borde sobre cuerpos con geometría compleja, lo que deriva en problemas de mallados de la capa límite rondando en espesores 50 % mayores a métodos tradicionales (Jasak and Tuković, 2015); Método de Volumen de Fluido (VoF) (Hirt and Nichols, 1981) y el Método de ajuste de nivel (LSM, por Level Set Method) (Adalsteinsson and Sethian, 1995). Estos métodos son utilizados para modelado de superficies libres o interfaces, logrando representar fronteras móviles pero no específicamente para el movimiento de sólidos con geometría compleja inmersos en un único fluido.

Otro método que logra superar las limitaciones mencionadas, es el denominado Método de Mallas Solapadas (OM, por Overset Mesh) o técnica de mallado Chimera, aplicada a condiciones de bordes móviles (Meakin, 1993; Prewitt et al., 2000; Blades and Marcum, 2005; Wolf, 2011). Una ventaja del método es el mallado de forma independiente del dominio de fluido de fondo (estático) y el de la zona móvil adherida al sólido, el cual puede poseer un movimiento prescripto, o responder a las cargas inducidas por el flujo de fluido. Las mallas son solapadas, y se comunican entre ellas para acoplar la solución por cada paso de tiempo. La calidad de mallado es asegurada para ambas, manteniendo siempre una capa límite adecuada para el cuerpo y zonas de refinamiento local. Como punto débil, el proceso de interpolación entre mallas puede ser costoso para el uso de cálculo paralelo con memorias distribuidas, característica común de los equipos de HPC basados en el protocolo de Interfaz de Paso de Mensaje (MPI, por Message Passing Interface) (Houzeaux et al., 2014).

No obstante que Chimera fue introducida a mediados de 1980, la potencial superioridad para manejar problemas de frontera móvil utilizando la metodología de mallas solapadas no son tenidas en cuenta normalmente en el ambiente de los usuarios de CFD, lo que sugiere que la adaptación de la comunidad es lenta, quizás por el grado de complejidad que posee. En los últimos años se han reportado algunos códigos propios académicos, códigos abiertos y otros comerciales han sugerido tener éxito en el desarrollo de las mallas solapadas (STAR CCM+ 2012, ANSYS Fluent 2016, ESI OpenFOAM 2017, LS-DYNA 2020) (Chandar et al., 2018).

1.2. **Objetivos generales y particulares**

La presente tesis de maestría alcanza los siguientes objetivos generales:

- Comprender la metodología de Mallas Solapadas, para el modelado de objetos móviles rígidos sometidos a la acción de viento CLA.
- Proponer y analizar modelos bidimensionales y tridimensionales, con aplicación ingenieril.
- Estudiar los fundamentos matemáticos de las Mallas Solapadas y emplear un código abierto que posibilite modelar la FSI sobre cuerpos propuestos.
- Implementar estrategias de mallado que permitan el correcto acople entre las mallas.
- Aplicar estrategias de HPC para el cálculo en paralelo y analizar rendimiento.

Los objetivos particulares alcanzados son los siguientes:

- ✓ Estudiar los fundamentos matemáticos de CFD y el Método de Volúmenes Finitos (FVM, por Finite Volume Method) y su relación con las mallas solapadas.
- ✓ Estudiar las principales estrategias de interpolación utilizadas en el acople de las mallas.
- ✓ Estudiar estrategias de mallado capaces de funcionar correctamente con las mallas solapadas y mantener una buena calidad de mallado en la zona de la capa límite.
- ✓ Calibrar el modelo numérico con resultados experimentales y numéricos obtenidos de la bibliografía.
- ✓ Estudiar el rendimiento computacional para cálculo paralelo:
 1. Definir estrategia de medición de tiempos detallada (Profiling).
 2. Estudiar escalabilidad, eficiencia y speed-up.
 3. Definir una estrategia de descomposición de dominio que resulte óptima para cada problema analizado.

En el presente trabajo se aborda el estudio de la interacción fluido-estructura en cuerpos rígidos mediante el modelado numérico CFD y la metodología de mallas solapadas OM, imponiéndoles un movimiento prescripto y sometidos a flujos con diferente número de Reynolds. Las simulaciones fueron llevadas a cabo con el software de código abierto y libre OpenFOAM. El algoritmo utilizado para resolver el problema FSI recibe la clasificación de “escalonado o staggered”, al resolver el fluido y la estructura en conjunto con su interfaz de comunicación con distintos “solvers”, el acople es clasificado débil al resolver en forma secuencial primero la cinemática del sólido y luego el campo de fluido en una sola iteración por cada paso de tiempo, sin control de convergencia entre los campos (Bogaers, 2015).

En el Cap. (2) de esta tesis se revelan conceptos básicos de FSI y formulaciones referentes al marco referencial utilizado. En el Cap. (3) se presenta y discute el problema matemático, el acople y solución detallando características de implementación del software usado. El Cap. (5) presenta una descripción general de la formulación matemática de las mallas solapadas en conjunto con las funciones de interpolación. El Cap. (6) esboza la importancia de la configuración del “solver” para problemas transitorios, mediante un ejemplo sencillo. El Cap. (7) presenta el interés de estudio en los modelos analizados, condiciones de borde y técnicas de mallado utilizadas. La estrategia de modelado del viento CLA utilizado en los casos de análisis donde el flujo es turbulento se presenta en el Cap. (4). Los resultados obtenidos en los casos evaluados se presentan en el Cap. (8) siendo contrastados con literatura de referencia específica sobre ensayos experimentales y numéricos.

2. Interacción fluido-estructura FSI

La interacción fluido-estructura, es un fenómeno presente en vastos campos de la naturaleza y desarrollos tecnológicos creados por el ser humano. Desde la deformación de árboles causada por la incidencia del viento, el torrente de flujo sanguíneo dentro del sistema de circulación elástico del cuerpo humano, hasta construcciones civiles de toda índole sometidas a la acción del viento, generadores eólicos, hidráulicos o lubricación de componentes de máquinas entre otros, siendo la aeronáutica una de las ramas precursoras en el estudio del tema. Hace ya tiempo atrás [Collar \(1946\)](#) elaboró un diagrama de la expansión del dominio multifísico que abarca el estudio de la aeroelasticidad, ver Fig. [\(1\)](#).

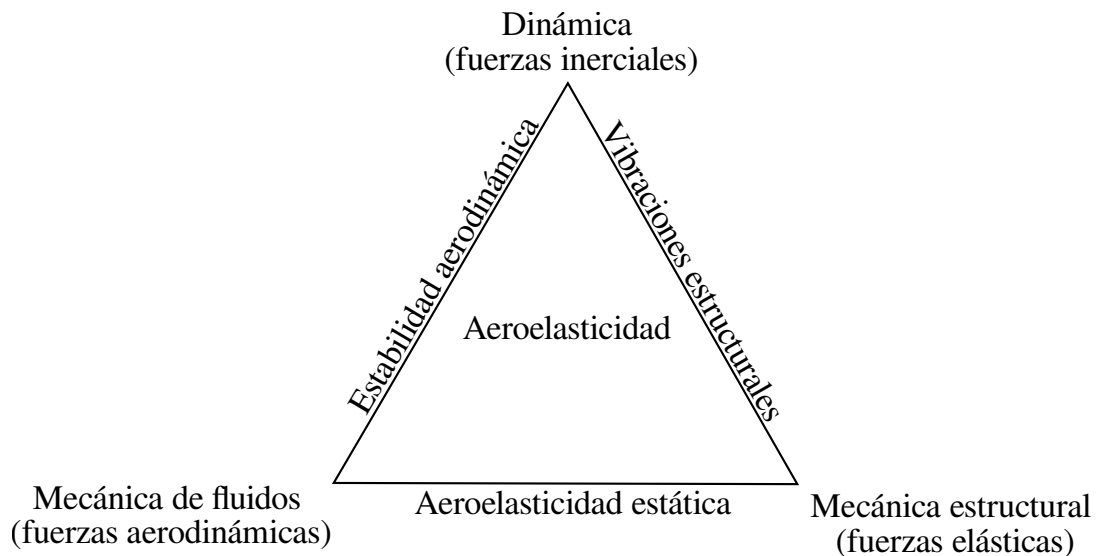


Figura 1: Diagrama de Collar

La simulación de este tipo de fenómenos acarrea una gran distorsión del continuo en las superficies de interfaz entre fluido-sólido, fluido-fluido, o fluido-estructura. Las metodologías clásicas para la descripción de movimiento utilizadas en la mecánica del continuo (Lagrangiana, Euleriana) por sí solas no resultan ser una herramienta eficiente para evaluar grandes desplazamientos en casos donde el dominio de análisis se compone de un sistema sólido-fluido. El método ALE ofrece una solución, pero la implementación en casos extremos con mallas dinámicas implican costosos procesos de remallado.

El acople y la solución del sistema de ecuaciones que representan los dominios en la Fig. [\(1\)](#) solo resulta posible formularlo con una única ecuación en casos de FSI sencillos para pocos grados de libertad ([Dowell et al., 2015](#)). Esta estrategia de acople se define como monolítica y es fuertemente acoplada, al resolver con una única formulación el campo fluido y estructural al mismo tiempo. En problemas complejos la formulación matemática y su implementación puede resultar una tarea difícil.

Otra estrategia es recurrir a métodos de solución particionados, que pueden ser de acople débil o fuerte, dependiendo de las iteraciones que realiza el algoritmo numérico para resolver el problema de acople entre el campo sólido y fluido por cada paso de tiempo. En este trabajo la imposición de movimiento del sólido y la consideración de rígido indeformable, permiten resolver el problema mediante un algoritmo particionado con acople débil donde el flujo se ve afectado por el cambio de posición de la estructura acorde a la función de movimiento impuesta, no admitiendo que las fuerzas generadas por el flujo sobre la estructura afecten la función de movimiento.

La Fig. (2) presenta un esquema que acompaña al párrafo anterior y ayuda a comprender la clasificación de los algoritmos numéricos utilizados en la resolución de problemas FSI y sus formas de acople.

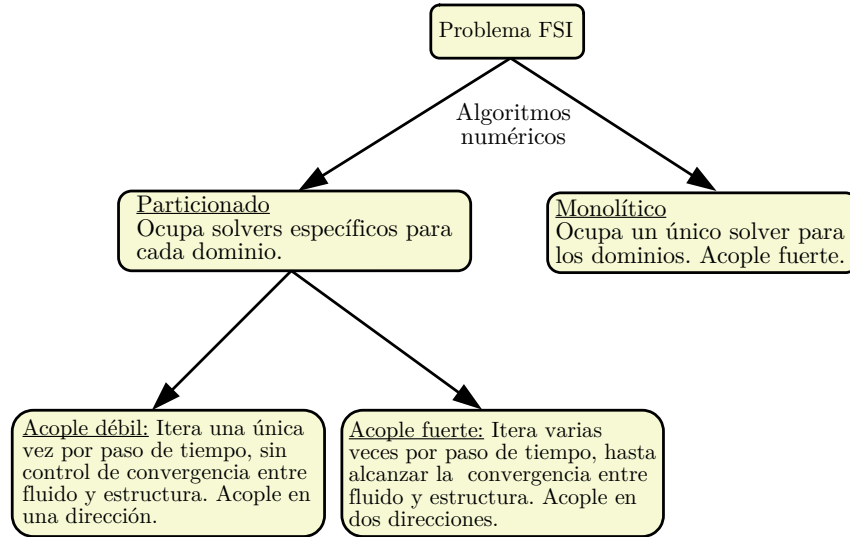


Figura 2: Diagrama de la caracterización y clasificación de solvers y formas de acople utilizado en el problema FSI.

2.1. Descripción del método ALE

El método ALE provee la formulación de las ecuaciones para problemas multifísicos, donde el sólido se desplaza dentro del fluido, en un marco temporal. El modelado de tal fenómeno se realiza mediante la descripción del flujo de fluido, a través de una malla móvil. La formulación material (Lagrangiana) y espacial (Euleriana) describen el movimiento de los conjuntos que componen el problema.

La descripción Lagrangiana acompaña con la malla el movimiento del continuo de cada punto material. Esto puede expresarse mediante una función φ que relaciona la coordenada material \vec{X} , con la coordenada espacial \vec{x} , en el tiempo, la cual define una ley de movimiento.

$$\varphi : (\vec{X}, t) \rightarrow \varphi(\vec{X}, t) = (\vec{x}, t) \quad (1a)$$

$$\vec{x} = \vec{x}(\vec{X}, t) \quad (1b)$$

El gradiente de la función toma la forma:

$$\frac{\partial \varphi}{\partial (\vec{X}, t)} = \begin{pmatrix} \frac{\partial \vec{x}}{\partial \vec{X}} & \vec{V} \\ \vec{0}^T & 1 \end{pmatrix} \quad (2)$$

$\vec{0}^T$ es un vector fila nulo y \vec{V} la velocidad en coordenadas materiales, asumiendo fija la coordenada material.

$$\vec{V}(\vec{X}, t) = \frac{\partial \vec{x}}{\partial t} \Big|_{\vec{x}} \quad (3)$$

Esta transformación debe poseer unicidad, es decir, que se corresponda un único punto para cada coordenada en el marco temporal y que su camino inverso permita obtener la misma coordenada. Para ello el Jacobiano debe verificar:

$$J = \det \left(\frac{\partial \vec{x}}{\partial \vec{X}} \right) > 0 \rightarrow \varphi^{-1}(\vec{x}, t) = (\vec{X}, t) \quad (4)$$

La descripción Lagrangiana resulta ideal para la aplicación entre interfaces o superficies, y es ampliamente aplicada en la mecánica estructural. Sin embargo, no puede lidiar con grandes distorsiones del dominio computacional sin operaciones de mallado intensivo (Donea et al., 2004). Es poco usual utilizar esta descripción en problemas de dinámica de fluidos caracterizados por fuertes gradientes de velocidad de deformación presentes en flujos turbulentos.

En la descripción Euleriana la malla computacional permanece fija y el fluido se mueve a través de ella, permitiendo sí grandes deformaciones características del movimiento del fluido. La velocidad material \vec{V} asociada a un nodo o centro de celda coincide para un paso de tiempo con la componente material del medio continuo, por lo tanto \vec{V} puede ser expresada en coordenadas espaciales mediante la Ec. (5). Esta formulación resulta ideal para la simulación de flujos turbulentos, pero posee problemas en la detección de interfaces, características en la FSI (Castro, 2013).

$$\vec{V} = \vec{V}(\vec{x}, t) \quad (5)$$

El método ALE combina las bondades de ambas descripciones, los nodos de la malla se mueven a una velocidad material distinta de las partículas de fluido, debido a esta libertad la malla soporta grandes desplazamientos y deformaciones, respecto de la formulación Lagrangiana, y con mejor resolución que la formulación Euleriana.

(Donea et al., 2004) presentan la formulación con tres dominios: material $R_{\vec{x}}$, espacial $R_{\vec{x}}$ y referencial $R_{\vec{\chi}}$, el cual es mapeado en los dominios material y espacial, mediante las funciones Φ y Ψ .

Así, el movimiento de la malla en el dominio espacial está definido por:

$$\Phi : (\vec{\chi}, t) \rightarrow \Phi(\vec{\chi}, t) = (\vec{x}, t) \quad (6a)$$

$$\frac{\partial \Phi}{\partial (\vec{\chi}, t)} = \begin{pmatrix} \frac{\partial \vec{x}}{\partial \vec{\chi}} & \vec{u}_M \\ \vec{0}^T & 1 \end{pmatrix} \quad (6b)$$

$$\vec{u}_M = \frac{\partial \vec{x}}{\partial t} \Big|_{\vec{x}} \quad (6c)$$

La Ec. (6c) representa la velocidad de malla, respecto del sistema referencial $\vec{\chi}$ considerado fijo.

La velocidad material, Ec. (3), y la velocidad de malla, Ec. (6c), se corresponden a derivadas temporales de posición material y de malla.

$$\Psi^{-1} : (\vec{X}, t) \rightarrow \Psi^{-1}(\vec{X}, t) = (\vec{\chi}, t) \quad (7a)$$

$$\frac{\partial \Psi^{-1}}{\partial (\vec{X}, t)} = \begin{pmatrix} \frac{\partial \vec{\chi}}{\partial \vec{X}} & \vec{w} \\ \vec{0}^T & 1 \end{pmatrix} \quad (7b)$$

$$\vec{w} = \frac{\partial \vec{\chi}}{\partial t} \Big|_{\vec{\chi}} \quad (7c)$$

La Ec. (7c) es la velocidad de la partícula en el dominio referencial, mide la variación temporal de $\vec{\chi}$ en el tiempo fijando \vec{X} .

La relación entre las velocidades \vec{V} , u_M y \vec{w} , viene dada por:

$$\frac{\partial \varphi}{\partial (\vec{X}, t)}(\vec{X}, t) = \frac{\partial \Phi}{\partial (\vec{\chi}, t)}(\Psi^{-1}(\vec{X}, t)) \frac{\partial \Psi^{-1}}{\partial (\vec{X}, t)}(\vec{X}, t) = \frac{\partial \Phi}{\partial (\vec{\chi}, t)}(\vec{\chi}, t) \frac{\partial \Psi^{-1}}{\partial (\vec{X}, t)}(\vec{X}, t) \quad (8)$$

$$\begin{pmatrix} \frac{\partial \vec{x}}{\partial \vec{X}} & \vec{V} \\ \vec{0}^T & 1 \end{pmatrix} = \begin{pmatrix} \frac{\partial \vec{x}}{\partial \vec{\chi}} & u_M \\ \vec{0}^T & 1 \end{pmatrix} \begin{pmatrix} \frac{\partial \vec{\chi}}{\partial \vec{X}} & \vec{w} \\ \vec{0}^T & 1 \end{pmatrix} \quad (9)$$

$$\vec{V} = u_M + \frac{\partial \vec{x}}{\partial \vec{\chi}} \cdot \vec{w} \quad (10a)$$

$$\vec{c} = \vec{V} - u_M = \frac{\partial \vec{x}}{\partial \vec{\chi}} \cdot \vec{w} \quad (10b)$$

La Ec. (10b) representa la velocidad relativa entre la coordenada material y la malla. Su inclusión en las ecuaciones de conservación de masa Ec. (11a) y momento Ec. (11b) de un sistema sólido-fluido, permite evaluar un problema de FSI mediante la metodología ALE.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho(\vec{V} - u_M)] = 0 \quad (11a)$$

$$\rho \frac{\partial \vec{V}}{\partial t} + \rho [(\vec{V} - u_M) \cdot \nabla] \vec{V} = \nabla \cdot \sigma + \rho \vec{b} \quad (11b)$$

ρ es la densidad de masa, σ es el tensor de tensiones de Cauchy y \vec{b} el vector de fuerzas por unidad de masa, estas ecuaciones pueden representar al sólido o fluido según la relación constitutiva impuesta.

La implementación ALE además, debe cumplir una restricción respecto de su velocidad de malla u_M , la cual debe satisfacer la denominada ley de conservación geométrica (GCL, por Geometric Conservation Law):

$$\frac{\partial V^{ce}}{\partial t} + \nabla \cdot u_M = 0 \quad (12)$$

En la Ec. (12) V^{ce} es el volumen de control de un elemento o celda, GCL requiere que el cambio del volumen entre pasos de tiempo, sea similar al volumen barrido por la cara de

una celda móvil en el mismo paso de tiempo, debido a la velocidad impuesta en la malla $u_M^{\vec{}}$. Las limitaciones de la metodología de acople ALE se dan debido a esta restricción, en casos donde el movimiento impuesto al sólido no sea un gran desplazamiento, sino un trayectoria completa descrita en el tiempo, la necesidad de mantener cierta calidad de elementos en la zona deformable de la malla, obliga a incluir al proceso de resolución del problema tareas de remallado, generando así un elevado coste computacional.

La metodología OM que se utilizará en este trabajo, ofrece una opción al estudio de la FSI, evitando tal proceso de remallado. La implementación utilizada define a $u_M^{\vec{}}$ como la velocidad de referencia impuesta al dominio móvil. Además, al considerar el cuerpo como un sólido rígido que acompaña el movimiento de la malla móvil permite asegurar $\nabla \cdot u_M^{\vec{}} = 0$, al igualar sus velocidades de frontera. Es posible afirmar entonces que las mallas no se deforman, mantienen su geometría inicial durante toda la simulación, asegurando la conservación del GCL en todo paso de tiempo.

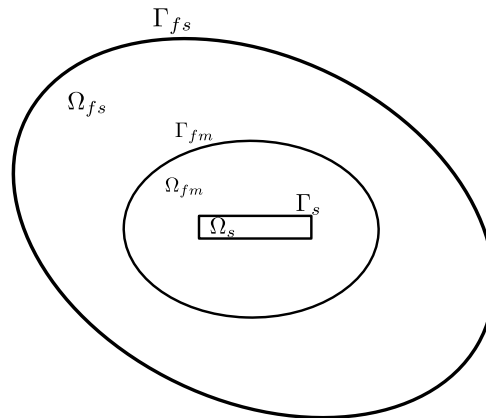


Figura 3: Descripción esquemática del problema de dominio

En la Fig. (3) se presentan los dominios y condiciones de borde esquemáticas para evaluar un problema de FSI, mediante la aplicación de la metodología OM. Ω_{fs} representa el dominio de fluido de fondo o denominado estático, Ω_{fm} el dominio de fluido móvil adherido a la estructura o sólido rígido indeformable Ω_s , Γ_{fs} la frontera del dominio de fluido estático o de fondo, Γ_{fm} frontera del dominio de fluido móvil y Γ_s la frontera de la estructura.

Aquí la Γ_{fm} posee la misma velocidad que Γ_s al asignar $u_M^{\vec{}}$ a toda la región, produciendo que el acople sea en coordenadas Eulerianas respecto de un sistema referencial móvil (Gamnitzer and Wall, 2006). Luego el acople entre los dominios Ω_{fs} y Ω_{fm} se produce por la interpolación de resultados entre ellas. Una descripción más detallada del OM se realizará en el Capítulo (5).

3. Problema matemático - acople y solución

El problema de flujo tridimensional, incompresible y transitorio, para un fluido newtoniano, es gobernado por las ecuaciones Navier-Stokes (NS), Ec. (13) y Ec. (14). Para la solución de ellas se requiere resolver el acople presión-velocidad, mediante algunos algoritmos, los cuales se presentarán y discutirán a continuación.

$$\nabla \cdot \vec{V} = 0 \quad (13)$$

$$\frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\vec{V}\vec{V}) - \nabla \cdot ((\nu + \nu_t)\nabla\vec{V}) = -\nabla p \quad (14)$$

donde el vector $\vec{V} = (u\vec{i}, v\vec{j}, w\vec{k})$ es la velocidad del campo, $p = P/\rho$ presión, ρ densidad del fluido, ν es la viscosidad cinemática y ν_t la viscosidad turbulenta la cual es calculada por el modelo de turbulencia adoptado. Las Ec. (13) y Ec. (14), son la denominadas ecuaciones de continuidad y conservación de momento. Tanto \vec{V} como p se encuentran acopladas en la ecuación de momento, las cuales pueden ser resueltas por los algoritmos clásicos Semi-Implicit Method for Pressure Linked Equations (SIMPLE) (Patankar and Spalding, 1972) y Pressure-Implicit with Splitting of Operators (PISO) (Issa et al., 1986). En general estos algoritmos constan de una primer parte denominada **predictor**, la cual solicita el valor de p^* de una condición inicial, o define un valor a partir de una iteración previa. Luego con el valor de p^* se discretiza la ecuación de momento y resuelve implícitamente, prediciendo así \vec{V}^* . Los campos de velocidad \vec{V}^* y presión p^* , satisfacen la ecuación de momento, pero al ser p^* incorrecta, arrastra el error sobre \vec{V}^* y hace que no satisfaga la ecuación de continuidad, debiendo corregir la velocidad. Para ello se re-ensambla la ecuación de presión derivándola (Ec. Poisson para la presión), para obtener p^{**} corregida, y luego explícitamente calcular \vec{V}^{**} , este último paso denominado **corrector**. El algoritmo descrito funciona iterativamente buscando que los campos de velocidad y presión satisfagan las ecuaciones gobernantes en forma simultánea.

El algoritmo de acople de presión-velocidad utilizado en esta tesis es el denominado (PIMPLE), el cual resulta de una combinación de PISO y SIMPLE. Posee algunas ventajas, permitiendo relajar la condición de Courant-Friedrichs-Lewy (CFL) Ec. (15), eliminando así la restricción de trabajar con pasos de tiempo pequeños $\Delta t \approx 0$, para cumplir con la condición $CFL \leq 1$, dado un tamaño de malla Δx y una velocidad de flujo predefinida U (Holzmann, 2019).

$$CFL = \frac{U\Delta t}{\Delta x} \quad (15)$$

El uso de PIMPLE asegura cierta estabilidad en la solución ante condiciones donde el $CFL \geq 1$ y con pasos de tiempo mayores a los requeridos por PISO, mediante la relajación implícita de la ecuación de momento como se plantea en el algoritmo SIMPLE.

El sistema de ecuaciones antes descrito puede ser resuelto por diferentes métodos numéricos, tales como: Elementos Finitos (FEM, por Finite Element Method), Diferencias Finitas (FDM, Finite Difference Method), Volúmenes Finitos (FVM, por Finite Volume Method). En este trabajo se utilizará el software `OpenFOAM`, el cual utiliza FVM, así divide al dominio computacional en volúmenes de control “e”. De esta manera el sistema de ecuaciones en derivadas parciales (PDE, por Partial Differential Equations) compuesto

por las ecuaciones Ec. (13) y Ec. (14), toman la siguiente forma matricial (Versteeg and Malalasekera, 1995; Jasak, 1996; Ferziger and Perić, 2020; Moukalled et al., 2016):

$$\sum_{f \sim nb(e)} S_f \vec{n}_f \cdot \vec{U}_f^{n+1} = 0 \quad (16)$$

$$\frac{\Delta \vec{U}}{\Delta t} + a_e \vec{U}_e^{n+1} + \sum_{v \sim nb(e)} a_v \vec{U}_v^{n+1} = \vec{b}_e - V_e (\nabla p^{n+1})_e \quad (17)$$

donde f representa la cara de e y v el volumen que rodea a e . S_f es el valor numérico del área de f y \vec{n}_f es el vector normal a f y V_e es el volumen de e , los coeficientes a_e , a_v y b_e , dependen de la velocidad del paso anterior, y pueden ser calculados explícitamente. La Ec. (17) usualmente es llamada **momentumpredictor** donde a partir de p^* , se resuelve el sistema de ecuaciones y obtiene U^* , la cual no satisface la ecuación de continuidad discretizada Ec. (16), y posteriormente U^* debe ser corregida. El algoritmo PIMPLE, a diferencia de PISO introduce un coeficiente de relajación (URF, por Under Relaxation Factor) α_u al término diagonal de la Ec. (17). La inclusión del URF expresada en la Ec. (18) permite esclarecer su formulación y qué términos resultan afectados, sin embargo la ecuación no será utilizada posteriormente en el desarrollo a fines de mantener una notación algebraica sencilla.

$$\frac{1}{\alpha_u} a_e \vec{U}_e^{n+1} + \sum_{v \sim nb(e)} a_v \vec{U}_v^{n+1} = \vec{b}_e - V_e (\nabla p^{n+1})_e + \frac{1 - \alpha_u}{\alpha_u} a_e \vec{U}_e^{n+1} \quad (18)$$

La necesidad de corrección y la restricción que plantea la ecuación de continuidad para flujos incompresibles al no contener términos de presión, hace que el sistema de ecuaciones no esté bien condicionado, por ello se requiere que sea reescrita en forma de la Ecuación de Poisson para la presión. Evaluando para un paso de tiempo el segundo término de la Ec. (17) puede ser rescrito como:

$$\vec{U}_e^{**} = \frac{1}{a_e} \left(- \sum_{v \sim nb(e)} a_v \vec{U}_v^* + \vec{b}_e - V_e (\nabla p^{**})_e \right) \quad (19)$$

Reemplazando la Ec. (19) en la Ec. (16), se obtiene el siguiente sistema de ecuaciones lineales, para todo los volúmenes de control:

$$\sum_{f \sim nb(e)} S_f \vec{n}_f \cdot \vec{U}_e^{**} = 0 \quad (20a)$$

$$\sum_{f \sim nb(e)} S_f \vec{n}_f \cdot \left[\frac{1}{a_e} \left(- \sum_{v \sim nb(e)} a_v \vec{U}_v^* + \vec{b}_e - V_e (\nabla p^{**})_e \right) \right] = 0 \quad (20b)$$

$$\sum_{f \sim nb(e)} S_f \vec{n}_f \cdot \frac{1}{a_e} (b_e - V_e (\nabla p^{**})) = \sum_{f \sim nb(e)} S_f \vec{n}_f \cdot \sum_{v \sim nb(e)} \frac{a_v}{a_e} \vec{U}_v^* \quad (20c)$$

$$A p^{**} = b \quad (21)$$

La Ec. (21) es denominada ecuación de presión, a partir de la cual se calcula p^{**} , en función de los coeficientes A y b . Luego es posible conocer \vec{U}^{**} con Ec. (19) para cada volumen de control. Estos últimos cálculos asociados a corregir la presión y luego la velocidad se denomina **pressurecorrector**.

Las tareas **momentumpredictor** y **pressurecorrector** pueden ser aplicadas en el algoritmo PIMPLE cuantas veces se desee mediante el control de repetición del loop. En OpenFOAM el acople del solver de movimiento de mallas solapadas denominado **overset** al PIMPLE, se denomina **overPimpleDyMFoam**, en la Fig. (4) se detalla el diagrama de flujo del mismo.

El software adopta una estrategia para el cálculo de la velocidad \vec{U}_f^{n+1} de la Ec. (16) mediante la interpolación de Rie-Chow, la cual puede ser considerada una corrección proporcional de la diferencia entre el gradiente de presión en la cara de un volumen y el valor interpolado a la cara desde el centro.

La estructura básica de archivos con la que se encuentra definida el solver en OpenFOAM, es mediante el uso de headers. En particular el header **UEqn.H** discretiza la Ec. (14). Al observar la sintaxis resulta algo diferente al incluir el término de movimiento relativo de malla, el cual será detallado más adelante.

```

1 // Solve the Momentum equation
2
3 MRF.correctBoundaryVelocity(U);
4
5 tmp<fvVectorMatrix> tUEqn
6 (
7     fvm::ddt(U) + fvm::div(phi, U)
8     + MRF.DDt(U)
9     + turbulence->divDevReff(U)
10    ==
11     fvOptions(U)
12 );
13
14 fvVectorMatrix& UEqn = tUEqn.ref();
15
16 UEqn.relax();
17
18 fvOptions.constrain(UEqn);
19
20 if (pimple.momentumPredictor())
21 {
22     solve(UEqn == -cellMask*fvc::grad(p));
23
24     fvOptions.correct(U);
25 }

```

Código 1: Header Ueqn.H, OF2006

En el Cód. (1) aparece un campo **phi** ϕ en lugar de una velocidad \vec{V} en el término convectivo, ϕ es la velocidad de flujo en cada cara del volumen evaluado, se expresa así debido a la aplicación del teorema de la divergencia, que transforma una integral de volumen en una de superficie, como se observa en la Ec. (22a).

$$\int_V \nabla \cdot (\vec{U}\vec{V})dV = \int_S (\vec{U}\vec{V})_f \cdot \vec{n}dS \approx \sum_i \vec{U}_{f,i} \vec{V}_{f,i} \cdot \vec{S}_f = \sum_i \vec{U}_{f,i} \phi_i \quad (22a)$$

$$\phi = \vec{V}_f \cdot \vec{S}_f \quad (22b)$$

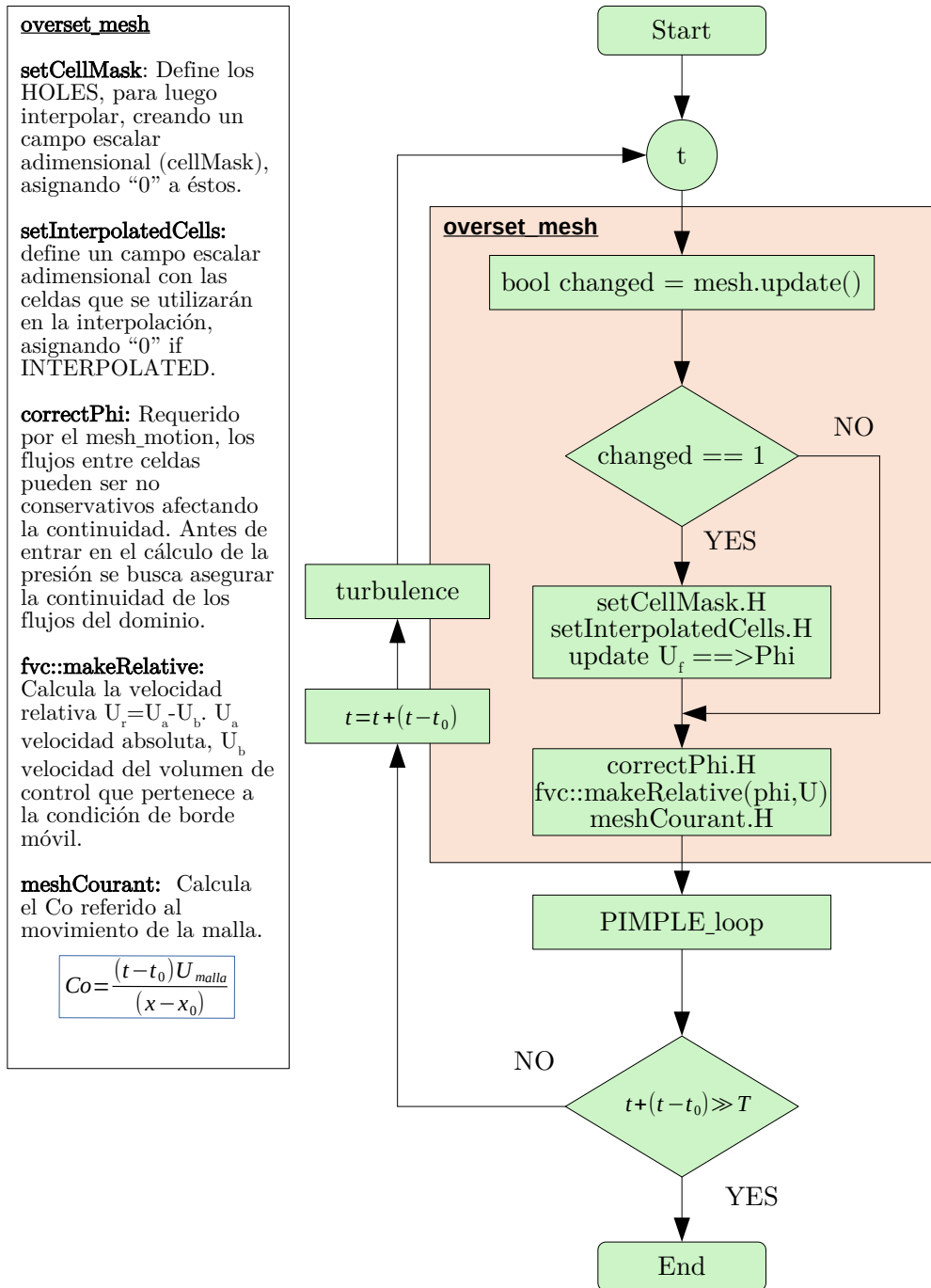


Figura 4: Diagrama de flujo, del solver utilizado overPimpleDyMFoam.

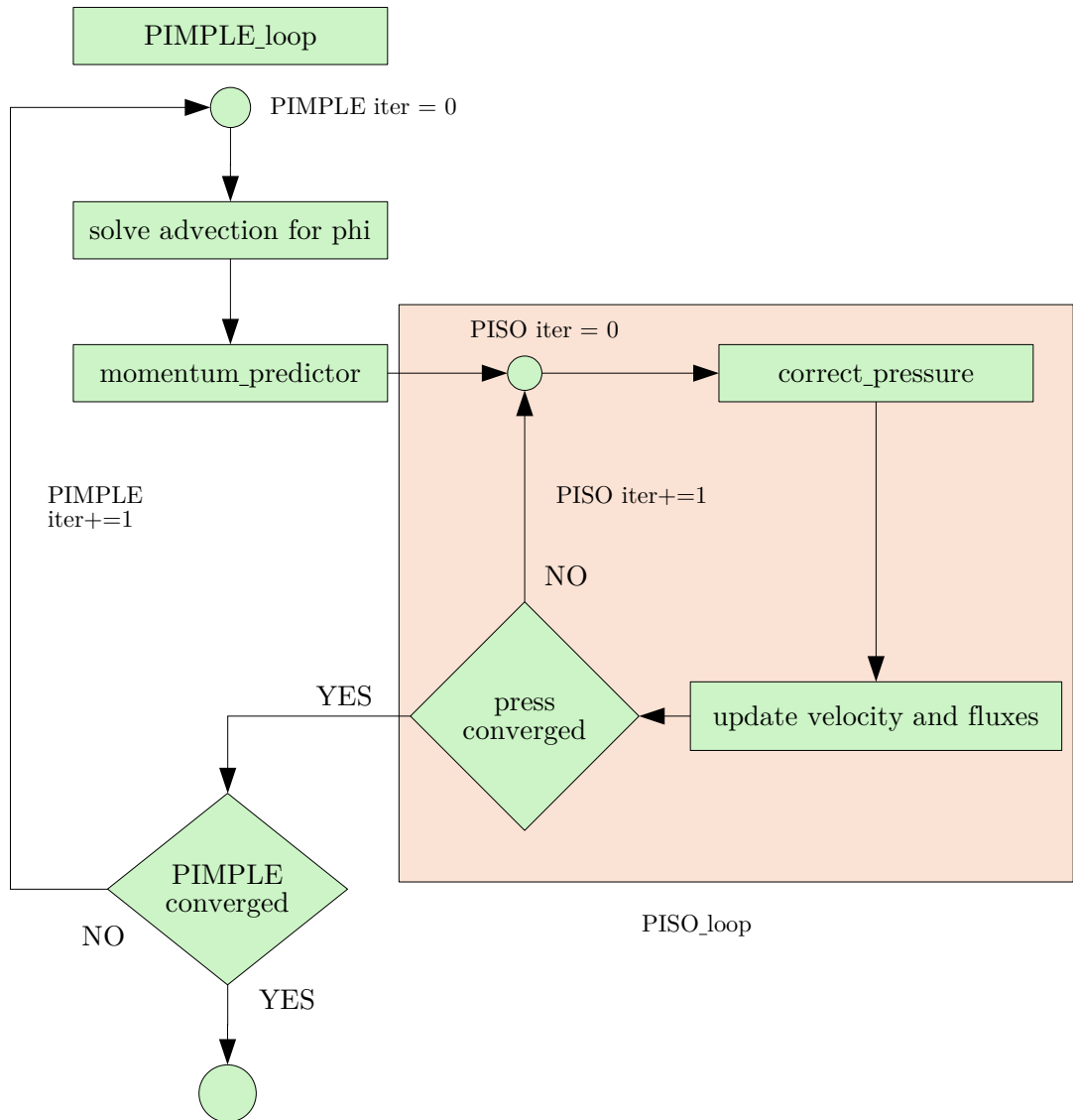


Figura 5: Diagrama de flujo del solver de acople de presión-velocidad PIMPLE.

$$\int_V \nabla \cdot (\vec{V}\vec{U})dV \rightarrow fvm :: div(phi,U) \quad (22c)$$

Evaluando la Ec. (22c), el vector \vec{V} proveniente del paso anterior del algoritmo planteado, expresado en la cara de la celda permanece constante cuando la ecuación para la presión se resuelva. Mientras que el vector \vec{U} es despejado de ella.

El Cód. (1) no incluye explícitamente el gradiente de presión, en cambio incluye el efecto de éste con la velocidad, discretizando la ecuación de momento para un paso de tiempo como un sistema matricial, el cual a partir de Ec. (17) se obtiene (Jasak, 1996; Uroić, 2019; Ferziger and Perić, 2020):

$$H[\vec{U}] = \vec{b}_e - \sum_{v \sim nb(e)} a_v \vec{U}_v^{n+1} \quad (23)$$

$$A[\vec{U}] = H - \nabla[p] \quad (24)$$

En la Ec. (24) A y H respectivamente representan a la descomposición diagonal y no diagonal de la matriz original resultante de la discretización mediante el FVM a la ecuación de momento. Son creadas por `UEqn.A()` y `UEqn.H()` denominadas así dentro del entorno de OpenFOAM.

$$UEqn.A() = H - \nabla[p] \quad (25a)$$

$$UEqn.H() = A[\vec{U}] + \nabla[p] \quad (25b)$$

La Ec. (24) es denominada `momentumpredictor` de $[\vec{U}]$, a partir del cual y con datos de presión del paso de tiempo anterior, o de una condición inicial se calcula $[\vec{U}]$ con la sentencia de la línea 22, del Cód. (1).

$$\vec{U}^* = \frac{UEqn.H()}{UEqn.A()} = 1 + \frac{\nabla[p]}{H - \nabla[p]} \quad (26)$$

La velocidad de Ec. (26) no satisface la continuidad. Es necesario corregir la presión, reescribiendo la ecuación de continuidad como la Ecuación de Poisson, mediante la divergencia de la Ec. (24), habiendo despejado la velocidad.

$$\nabla \cdot [\vec{U}] = \nabla \cdot \left[\frac{H}{A} \right] - \nabla \cdot \left[\frac{\nabla[p]}{A} \right] \quad (27)$$

El lado izquierdo de la Ec. (27) es nulo al suponer flujo incompresible $\nabla \cdot [\vec{U}] = 0$.

$$\nabla \cdot [\vec{U}^*] = \nabla \cdot \left[\frac{\nabla[p]}{A} \right] \quad (28)$$

Dejando constante la velocidad, se calcula la presión mediante la Ec. (28).

Así esta serie de cálculos desarrollados en las ecuaciones anteriores forman parte del algoritmo indicado en el diagrama de flujo de la Fig. (5), los cuales pueden ser

repetidos cuantas veces la estrategia de resolución requiera recorrer el bucle externo `momentum_predictor`, como interno `pressure_corrector`, teniendo en cuenta el coste computacional asociado a este proceso iterativo.

En OpenFOAM se reemplaza a \vec{U}^* , por la velocidad de flujo ϕ , evaluándola así en las caras y no en el centro de celda. En el Cód. (2) entre las líneas 4 a 31 se realiza la interpolación hacia las caras desde el centro de celda, tanto para la $\text{raU} \rightarrow \text{raUf}$, como para $\text{HbyA} = \text{U} \rightarrow \text{phiHbyA} = \phi$.

```

1 // Pressure corrector
2
3 // Option 1: interpolate rAU, do not block out rAU on blocked cells
4 volScalarField rAU("rAU", 1.0/UEqn.A());
5 mesh.interpolate(rAU);
6
7 surfaceScalarField rAUf("rAUf", fvc::interpolate(rAU));
8 volVectorField H("H", UEqn.H());
9
10 volVectorField HbyA("HbyA", U);
11 HbyA = constrainHbyA(rAU*H, U, p);
12
13 if (massFluxInterpolation)
14 {
15 #include "interpolatedFaces.H"
16 }
17
18 if (runTime.outputTime())
19 {
20 H.write();
21 rAU.write();
22 HbyA.write();
23 }
24
25 if (pimple.nCorrPISO() <= 1)
26 {
27 tUEqn.clear();
28 }
29
30 phiHbyA = fvc::flux(HbyA);
31
32 if (ddtCorr)
33 {
34 surfaceScalarField faceMaskOld
35 (
36 localMin<scalar>(mesh).interpolate(cellMask.oldTime())
37 );
38 phiHbyA += rAUf*faceMaskOld*fvc::ddtCorr(U, Uf);
39 }
40
41 MRF.makeRelative(phiHbyA);
42
43 // WIP
44 if (p.needReference())
45 {
46 fvc::makeRelative(phiHbyA, U);
47 adjustPhi(phiHbyA, U, p);
48 fvc::makeAbsolute(phiHbyA, U);
49 }
50
51
52 if (adjustFringe)
53 {
54 fvc::makeRelative(phiHbyA, U);
55 oversetAdjustPhi(phiHbyA, U);
56 fvc::makeAbsolute(phiHbyA, U);
57 }
58
59 while (pimple.correctNonOrthogonal())
60 {
61 fvScalarMatrix pEqn

```



```

62 (
63 fvm::laplacian(rAUf, p) == fvc::div(phiHbyA)
64 );
65
66 pEqn.setReference(pRefCell, pRefValue);
67
68 pEqn.solve(mesh.solver(p.select(pimple.finalInnerIter())));
69
70 if (pimple.finalNonOrthogonalIter())
71 {
72 phi = phiHbyA - pEqn.flux();
73 // option 2:
74 // rAUf*fvc::snGrad(p)*mesh.magSf();
75 }
76 }
77
78
79 #include "continuityErrs.H"
80
81 // Explicitly relax pressure for momentum corrector
82 p.relax();
83 volVectorField gradP(fvc::grad(p));
84
85 // Option 2: zero out velocity on blocked out cells
86 //U = HbyA - rAU*cellMask*gradP;
87 // Option 3: zero out velocity on blocked out cells
88 // This is needed for the scalar Eq (k,epsilon, etc)
89 // which can use U as source term
90 U = cellMask*(HbyA - rAU*gradP);
91 U.correctBoundaryConditions();
92
93 fvOptions.correct(U);
94
95 {
96 Uf = fvc::interpolate(U);
97 surfaceVectorField n(mesh.Sf()/mesh.magSf());
98 Uf += n*(phi/mesh.magSf() - (n & Uf));
99 }
100
101 // Make the fluxes relative to the mesh motion
102 fvc::makeRelative(phi, U);
103
104 surfaceScalarField faceMask
105 (
106 localMin<scalar>(mesh).interpolate(cellMask)
107 );
108 phi *= faceMask;
109
110 }

```

Código 2: Header Peqn.H, OF2006

En la línea 63 se calcula el gradiente de presión aplicando el Teorema de Gauss, se puede calcular una primer derivada en lugar del Laplaciano, al evaluar el flujo ϕ y la matriz diagonal inversa \mathbf{raUf} ya interpolados en las caras.

Entre las líneas 79 a 93 del Cód. (2), se calcula la velocidad en el centro de las celdas corregida con el gradiente de presión corregido, éste es evaluado usando el Teorema de Gauss, siendo solo necesario de dato la presión en las caras de la celda en cuestión. Línea 90 del Cód. (2), para más detalles sobre la implementación del algoritmo PIMPLE referirse a (Holzinger, 2020).

Hasta ahora solo se analizó el problema de acople presión-velocidad mediante el algoritmo citado, pero no se tuvo en cuenta el término de aceleración asociado al movimiento de la malla presente en el Cód. (1), $\text{MRF.DDt}(U)$, este se denomina Multiple Reference Fra-

me (MRF) y es utilizado cuando existen zonas móviles de una misma malla o de mallas solapadas, conectadas por una interfaz con una zona estática o background.

La Ec. (14) se modifica para incluir los efectos de movimiento relativo, adhiriendo una serie de términos nuevos. Se utilizarán las nomenclaturas I para el marco inercial o absoluto y R para el marco relativo, en la Ec. (29), que al ser evaluada para un paso de tiempo se suprimen los términos derivadas temporales.

$$\frac{\partial \vec{U}_R}{\partial t} + \frac{d\vec{\Omega}}{dt} \times \vec{r} + \vec{\Omega} \times \vec{U}_I + \nabla \cdot (\vec{U}_I \vec{U}_R) - \nabla \cdot ((\nu + \nu_t) \nabla \vec{U}_I) = -\nabla p \quad (29)$$

Como el término convectivo se discretiza aplicando el Teorema de Gauss, Ec. (22a), se puede asumir que $\vec{V} = \vec{U}_R$. Es posible así expresar la ecuación Ec. (29), solo en relación de la velocidad absoluta o inercial (Wimshurst, 2020).

$$\vec{U}_R = \vec{U}_I - (\vec{\Omega} \times \vec{r}) \quad (30)$$

A partir de esta sustitución aparece un término asociado a la corrección del flujo, de reemplazar Ec. (30) en Ec. (22a), expresado por la ecuación Ec. (32)

$$\sum_i \vec{U}_{f,i} \vec{V}_{f,i} \cdot \vec{S}_{f,i} = \sum_i \vec{U}_I \vec{U}_R \cdot \vec{S}_f \quad (31a)$$

$$\sum_i \vec{U}_{f,i} (\vec{U}_{f,i} - \vec{\Omega} \times \vec{r}) \vec{S}_{f,i} \quad (31b)$$

$$\sum_i \vec{U}_{f,i} \vec{U}_{f,i} \cdot \vec{S}_{f,i} - \sum_i \vec{U}_{f,i} (\vec{\Omega} \times \vec{r}) \cdot \vec{S}_{f,i} \quad (31c)$$

$$\sum_i \vec{U}_{f,i} (\vec{\Omega} \times \vec{r}) \cdot \vec{S}_{f,i} \quad (32)$$

El tratamiento de las mallas móviles y solapadas será tratado con mayor profundidad los capítulos siguientes.

3.1. Malla móvil

El problema físico de Interacción Fluido-Estructura se reduce a predecir el flujo cuando escurre sobre elementos de condiciones de borde móviles como hélices, turbinas, cuerpos flexibles o geometrías complejas móviles. En esta tesis la movilidad es asignada a los diferentes cuerpos analizados mediante una función prescripta en el tiempo.

Para ello el solver `overPimpleDyMFoam`, incluye a la librería de mallas dinámicas `dynamicFvMesh.H` y mediante la sentencia `mesh.update()` presentada en el diagrama de flujo de la Fig. (4), se calcula la nueva posición de los puntos asociados a la zona móvil. Es posible asignar diferentes funciones de movimiento a cuerpos o zonas de la malla, ellas pueden ser de traslación lineal, rotación u oscilantes o su forma más avanzada de movimiento prescripto en seis grados de libertad asociados a movimientos tridimensionales.

La imposición de movimiento prescripto al sólido rígido indeformable mediante las rutinas de mallas móviles y posterior cálculo del campo de flujo para un mismo paso de tiempo definen al **solver** como particionado y de acople débil. Mediante este esquema se conoce la serie temporal de las cargas incidentes sobre el cuerpo analizado.

4. Flujo Turbulento y modelo de cierre adoptado

Los flujos de fluidos en aplicaciones ingenieriles de distinta índole, son en su gran mayoría clasificados como turbulentos, siendo la turbulencia la principal causa de los fenómenos FSI. Debido a la complejidad que conlleva realizar una definición única y precisa de la turbulencia, se opta por describir éstos flujos mediante una serie de propiedades, que pueden ser ampliados en base al libro de [Kundu et al. \(2012\)](#) u bibliografía específica ([Tennekes and Lumley, 1978](#)).

El número adimensional de Reynolds permite clasificar los tipos de flujos (laminar, transición o turbulento), especificando la relación entre las fuerzas de inercia y viscosas, Ec. [\(33\)](#).

$$Re = \frac{\rho UL}{\mu} \quad (33)$$

donde ρ densidad del fluido, U velocidad de referencia, L longitud característica y μ viscosidad dinámica del fluido.

En un flujo turbulento, el campo de velocidades y la información que transporta éste varían de forma irregular en tiempo y espacio. Una estrategia para abordar el problema, es describir la variabilidad del fenómeno mediante el análisis estadístico, buscando una equivalencia del proceso aleatorio generado por las leyes de probabilidades, para caracterizar así el comportamiento del flujo mediante modelos no determinísticos.

La ecuación de NS expresada en la Ec. [\(34\)](#) describe a todos los flujos incompresibles, sin embargo a pesar de no haberse reportado una solución que incluya escurrimiento turbulento con condiciones de borde complejas, se recurren a estrategias. Una de ellas es la descomposición del campo de velocidades instantáneo propuesta por Reynolds Ec. [\(35a\)](#), equivalente a la suma de una componente de velocidad media solo función del espacio Ec. [\(35b\)](#) y la adición de un término fluctuante dependiente del espacio y tiempo.

$$\frac{\partial \vec{V}}{\partial t} + \nabla \cdot (\vec{V}\vec{V}) - \nabla \cdot (\nu \nabla \vec{V}) = -\nabla p \quad (34)$$

$$\vec{V}(\vec{x}, t) = \langle \vec{U}(\vec{x}) \rangle + \vec{u}'(\vec{x}, t) \quad (35a)$$

$$\langle \vec{U}(\vec{x}) \rangle = \lim_{T \rightarrow \infty} \int_0^T \vec{V}(\vec{x}, t) dt \quad (35b)$$

Al introducirse la descomposición propuesta por Reynolds Ec. [\(35a\)](#) en la Ec. [\(34\)](#), luego de una serie de arreglos, distribución de factores y aplicación de promedios temporales a términos fluctuantes, se obtiene una ecuación función solo de las variables promediadas del campo de velocidades, la cual es llamada Ecuación de Navier-Stokes promediada en el tiempo RANS, por simplicidad es expresada en notación indicial de Einstein Ec. [\(36\)](#), donde $\vec{U}(\vec{x})$ es reemplazado por U_i .

$$\frac{(\rho U_i)}{\partial t} + \frac{\partial}{\partial x_j} (\rho U_i U_j) = -\frac{\partial \langle P \rangle}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) - \rho \langle u'_i u'_j \rangle \right] \quad (36)$$

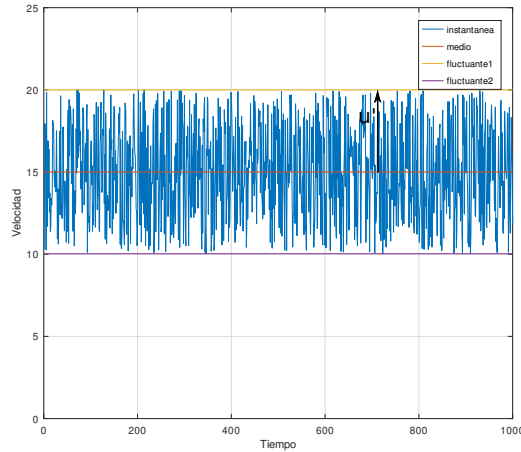


Figura 6: Espectro de velocidad y términos de descomposición.

El término $-\rho \langle u'_i u'_j \rangle$ se denomina tensor de Reynolds, representa el producto de las velocidades fluctuantes promediadas en el tiempo. Esta operación introduce seis nuevas componentes al sistema de ecuaciones de NS mal condicionándolo.

El tensor resultante es simétrico, tal propiedad se fundamenta al realizar un análisis temporal sobre las variables fluctuantes promediadas, el cual demuestra que el orden en la multiplicación entre componentes no altera el producto.

$$-\langle \rho u'_i u'_j \rangle = \tau_{ij} = \begin{bmatrix} \langle u'u' \rangle & \langle u'v' \rangle & \langle u'w' \rangle \\ \langle v'u' \rangle & \langle v'v' \rangle & \langle v'w' \rangle \\ \langle w'u' \rangle & \langle w'v' \rangle & \langle w'w' \rangle \end{bmatrix} \quad (37)$$

Acorde a lecturas desarrolladas por [Wimshurst \(2021\)](#) se describirá el problema de cierre en los párrafos siguientes.

Al aplicar modelos de turbulencia RANS no se calcula directamente $\langle u'_i u'_j \rangle$, se sustituye al tensor por una función del flujo medio. Los Modelos de Viscosidad Turbulenta son empleados para ello.

Los esfuerzos cortantes para un flujo en 2D, Fig. [\(7\)](#), están compuestos por la contribución del gradiente de velocidad del flujo medio (esfuerzo cortante viscoso) Ec. [\(38a\)](#) y por los torbellinos turbulentos Ec. [\(38b\)](#). Con $\langle \vec{U}(\vec{x}) \rangle = (U\vec{i}, V\vec{j}, W\vec{k})$.

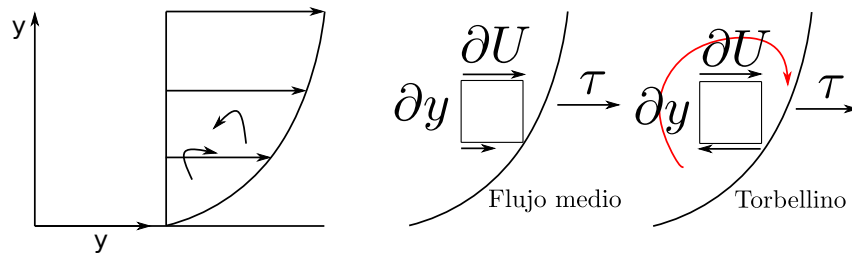


Figura 7: Flujo cortante 2D

$$\tau = \mu \frac{\partial U}{\partial y} \quad (38a)$$

$$\tau = - \langle u'_i u'_j \rangle \quad (38b)$$

El flujo turbulento posee múltiples escalas, las técnicas RANS modelan una única escala, bajo la hipótesis de que las fluctuaciones turbulentas son análogas al movimiento Browniano de un gas, caracterizado por partículas que chocan entre sí, intercambiando energía y cantidad de movimiento de forma aleatoria. En realidad las fluctuaciones de las partículas del fluido no son aleatorias, estas se mueven formando escalas de torbellinos que en ciertas condiciones pueden estar correlacionadas. Los modelos de viscosidad turbulenta lineal se basan en el supuesto de aleatoriedad, perdiendo así información de las escalas mayores de turbulencia, respecto de otras técnicas como la Simulación de grandes Vórtices LES o Simulación Directa (DNS, por Direct Numerical Simulation). La permanencia del término transitorio en las RANS permite que éstas capturen las escalas más grandes del flujo turbulento (uRANS, por Unsteady Reynolds Average Navier-Stokes), ver Fig. (8).

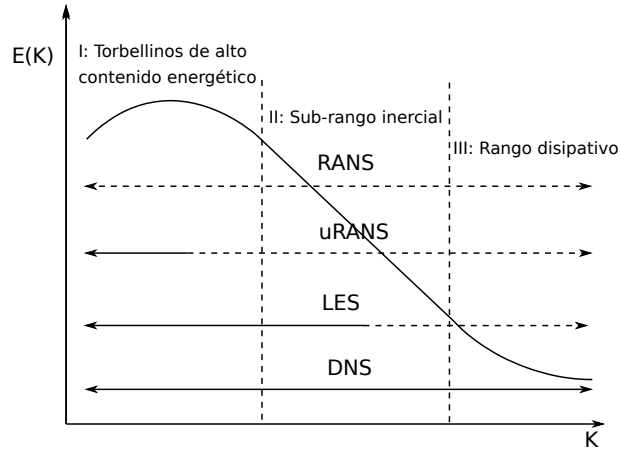


Figura 8: Espectro de energía turbulenta $E(\kappa)$ y número de onda κ . Rangos de modelado y resolución de la turbulencia para diferentes técnicas. Líneas horizontales con guiones: Modelado; Líneas horizontales negras: Resolución (Arvidson 2017).

La relación entre las fluctuaciones y el gradiente del campo de velocidades medias es proporcional a una variable definida por Boussinesq (1877) μ_t presente en la Ec. (39), como viscosidad turbulenta, expresión que varía para cada modelo turbulento. Esta viscosidad turbulenta es un parámetro artificial definido por el flujo, que representa la fuerza de difusión entre las capas que intercambian cantidad de movimiento y energía.

$$- \langle \rho u'_i u'_j \rangle = \mu_t \frac{\partial \langle \vec{U}(\vec{x}) \rangle}{\partial y} \quad (39)$$

Considerando los términos tangenciales la expresión anterior toma la forma:

$$- \langle \rho u'v' \rangle = \mu_t \left(\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right) \quad (40)$$

Y el análisis de las componentes normales presentan en el término izquierdo de la ecuación la inclusión de la expresión para la energía cinética del flujo turbulento:

$$- \langle \rho u'u' \rangle = \mu_t \left(\frac{\partial U}{\partial x} + \frac{\partial U}{\partial x} \right) = 2\mu_t \frac{\partial U}{\partial x} \quad (41)$$

La Energía Cinética Turbulenta se define, con la siguiente expresión:

$$k = \frac{1}{2}(\langle u'u' \rangle + \langle v'v' \rangle + \langle w'w' \rangle) \quad (42)$$

La suma de las componentes normales del tensor de Reynolds:

$$-\rho(\langle u'u' \rangle + \langle v'v' \rangle + \langle w'w' \rangle) = -2\rho k \quad (43)$$

El término derecho de la ecuación Ec. (44) es la divergencia del vector velocidad, el cual para flujos incompresibles es nulo $\vec{\nabla} \cdot \vec{V} = 0$, planteándose así un absurdo porque la energía cinética turbulenta debería ser nula y no lo es.

$$-\rho(\langle u'u' \rangle + \langle v'v' \rangle + \langle w'w' \rangle) = 2\mu_t \left(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W}{\partial z} \right) \quad (44)$$

Se corrige el error del análisis de las componentes normales, mediante la extracción de 1/3 de cada componente normal. Sería equivalente a sustraer del tensor de Reynolds (simétrico), la componente esférica, resultando de la sustracción la componente deviatoríca.

$$-\langle \rho u'u' \rangle = 2\mu_t \left(\frac{\partial U}{\partial x} - \frac{1}{3} \vec{\nabla} \cdot \vec{V} \right) - \frac{1}{3}(2\rho k) \quad (45a)$$

$$-\langle \rho v'v' \rangle = 2\mu_t \left(\frac{\partial V}{\partial y} - \frac{1}{3} \vec{\nabla} \cdot \vec{V} \right) - \frac{1}{3}(2\rho k) \quad (45b)$$

$$-\langle \rho w'w' \rangle = 2\mu_t \left(\frac{\partial W}{\partial z} - \frac{1}{3} \vec{\nabla} \cdot \vec{V} \right) - \frac{1}{3}(2\rho k) \quad (45c)$$

Sumando las componentes modificadas el primer término de la derecha de la Ec. (46) se anula quedando la igualdad planteada en la Ec. (42):

$$-\rho(\langle u'u' \rangle + \langle v'v' \rangle + \langle w'w' \rangle) = 2\mu_t(\vec{\nabla} \cdot \vec{V} - \vec{\nabla} \cdot \vec{V}) - 2\rho k \quad (46)$$

El análisis realizado de las componentes normales y tangenciales de la ecuación de Boussinesq, pueden ser reescritas de forma tensorial o vectorial.

$$-\rho \langle u'_i u'_j \rangle = \mu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} - \frac{2}{3} \frac{\partial U_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho k \delta_{ij} \quad (47)$$

$$-\rho \langle u'u' \rangle = \mu_t \left((\vec{\nabla} \vec{V}) + (\vec{\nabla} \vec{V})^T - \frac{2}{3} (\vec{\nabla} \vec{V}) \bar{I} \right) - \frac{2}{3} k \bar{I} \quad (48)$$

Es posible expresar la misma ecuación aún de forma más compacta mediante el uso del Tensor Velocidad de Deformación.

Tensor simétrico

$$S_{ij} = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \quad (49)$$

Tensor deviatórico

$$S_{ij}^* = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} - \frac{1}{3} \frac{\partial U_k}{\partial x_k} \delta_{ij} \right) \quad (50)$$

La Ec. (47) es reescrita:

$$-\rho \langle u'_i u'_j \rangle = 2\mu_t S_{ij}^* - \frac{2}{3} \rho k \delta_{ij} \quad (51)$$

Así se obtiene la relación entre las variables fluctuantes y las velocidades medias del flujo turbulento, el desafío resta en calcular la μ_t la cual varía según el tipo de modelo de turbulencia seleccionado.

En este trabajo los tipos de flujos turbulentos producto de las condiciones de borde impuestas y formas geométricas estudiadas, requieren de un modelo de turbulencia que posea la capacidad de modelar el flujo en la capa límite y corriente libre aguas abajo del cuerpo de forma acertada.

El modelo de turbulencia adoptado es $k\omega_{SST}$ propuesto por [Menter \(1992\)](#), el cual posee la capacidad de representar de forma correcta gradientes de presión negativos y separaciones de flujo de capa límite, presentes en los distintos modelos estudiados. El modelo en sí, es una combinación de las mejores características de los modelos de turbulencia $k-\epsilon$ y $k-\omega$. El modelo $k-\omega_{SST}$ incluye un término de $k-\epsilon$ controlado por una función de mezcla $0 \leq F_1 \leq 1$, en regiones cercanas a las paredes $F_1 \approx 1$ el modelo se comporta como $(k-\omega)$ y en regiones de flujo libre $F_1 \approx 0$ lo hace como $(k-\epsilon)$. De ese modo se logra aplicar un modelo de turbulencia óptimo en cada región de trabajo, pudiendo funcionar como un híbrido en zonas intermedias.

Además se incluye otra función F_2 , limitadora del valor de viscosidad turbulenta ν_t , F_2 aumenta en sectores cercanos a las paredes y la viscosidad turbulenta es reducida.

En `OpenFOAM` la ecuación Ec. (51) está definida en el archivo `eddyViscosity.C`, con el signo ya despejado.

$$((2,0/3,0) * I) * \text{tk}() - (\text{nut}_) * \text{dev}(\text{twoSymm}(\text{fvc} :: \text{grad}(\text{this} - > U_))$$

El término `tk()` representa la formulación de energía cinética y `nut_` viscosidad cinemática turbulenta, ambos parámetros específicos para el modelo de turbulencia seleccionado.

Así `dev(twoSymm(fvc :: grad(this - > U_)))` equivale a dos veces la parte deviatórica del tensor simétrico velocidad de deformación.

$$\text{twoSymm}(T) = T + T^T = \begin{bmatrix} 2T_{11} & T_{21} + T_{12} & T_{31} + T_{13} \\ T_{12} + T_{21} & 2T_{22} & T_{32} + T_{23} \\ T_{13} + T_{31} & T_{23} + T_{32} & 2T_{33} \end{bmatrix} \quad (52)$$

El software `OpenFOAM` posee implementada la versión de [Menter et al. \(2003\)](#), la cual debe ser inicializada con los parámetros de (k) energía cinética turbulenta y (ω) tasa de disipación turbulenta específica, ver Ec. (53a) y Ec. (53b), para calcular su expresión de μ_t .

$$k = \frac{3}{2} (I |\vec{u}_{ref}|)^2 \quad (53a)$$

$$\omega = \frac{k^{0,5}}{C_\mu^{0,25} L} \quad (53b)$$

donde I representa la intensidad de turbulencia, \vec{u}_{ref} velocidad de referencia, C_μ constante adimensional 0,09 y L es la longitud de referencia.

5. Mallas solapadas OM

En líneas generales el problema de FSI estudiado en este trabajo, puede ser resuelto mediante la estrategia de mallas solapadas. Así se definen dos dominios computacionales, uno móvil que acompaña pegado al cuerpo con su movimiento prescrito, y el otro dominio estático de fondo, esquematizados de forma genérica en la Fig. (9). En ellos se resuelven los campos de las variables analizadas e interpolan los resultados en cada paso de tiempo, para obtener así una solución única del problema planteado, como el de la Fig. (11).

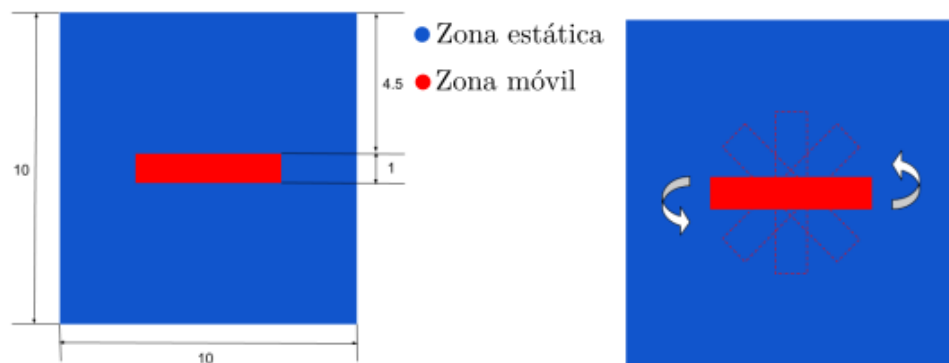


Figura 9: Vista esquemática del dominio definido en zonas.

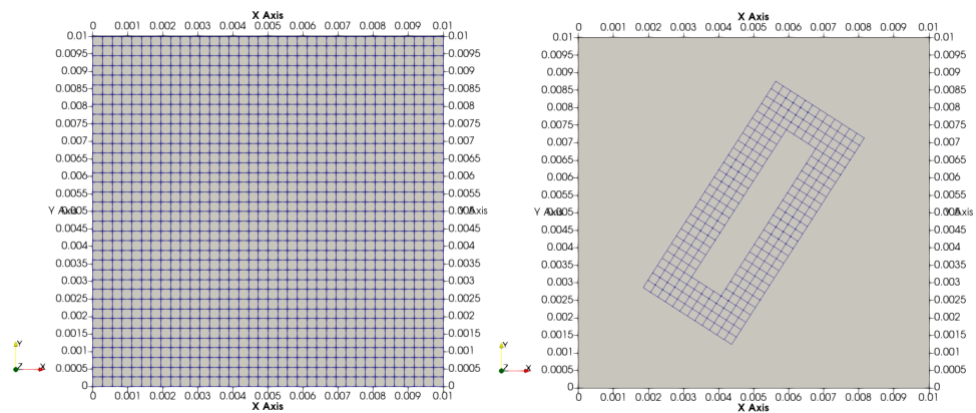


Figura 10: Vista esquemática de los dominios mallados.

Algunas de las ventajas que plantea la estrategia es que ambos dominios pueden ser mallados independientemente, por la no necesidad de alineación espacial alguna entre sus elementos, como es posible notar en la Fig. (10). Permitiendo el movimiento relativo en seis grados de libertad (6DoF) entre sus componentes o configuraciones multicuerpos.

Además el proceso de actualización de la malla para cada paso de tiempo solo es requerido en las variables definidas en la interpolación. Es así que el acople solo afecta las interfaces y si estas se encuentran alejadas del cuerpo, el impacto producido por el acople será mínimo (Houzeaux et al., 2014).

La metodología permite trabajar con mallas estructuradas y no estructuradas de alta calidad, muchas veces requeridas en geometrías complejas. La discretización independiente entre las partes, otorga una gran flexibilidad en procesos de diseño industrial y optimización, al no necesitar remallar todo el dominio (Storti, 2020). Resultando ideal para

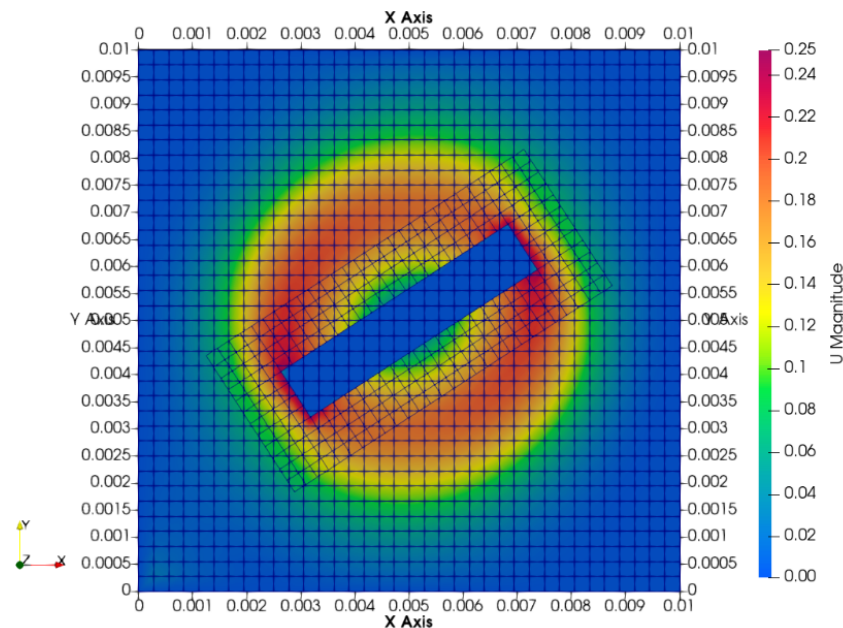


Figura 11: Campo de velocidad obtenido.

simular condiciones donde cuerpos inmersos en fluidos poseen grandes desplazamientos, y en donde otros métodos generan deformación en la malla a niveles restrictivos, requiriendo un remallado completo (Hadzic, 2006).

El proceso de mallado del dominio móvil y de fondo, la unión, definición de parches “overset” y zonas es llevado a cabo en forma manual. Estas acciones serán detalladas en la presente tesis en la sección de descripción del mallado para los modelos evaluados.

Con las mallas definidas el paso siguiente es la intercomunicación entre ellas, el solver se encarga de definir una plantilla de celdas definidas como:

- Discretización del problema: Donde se resuelven las variables de campo del dominio computacional mediante el FVM.
- Interpolación entre mallas: En ellas se realiza el traspaso de solución de las mallas mediante el uso de funciones de interpolación.
- Huecos o sin uso: Celdas definidas por la intersección del sólido de la malla móvil con las celdas de la malla de fondo. En ellas no se calcula ni se interpola ninguna variable.

La plantilla permite obtener una solución única en todo el dominio y por paso de tiempo. Esta se actualiza en función al movimiento del cuerpo, es decir que las celdas que la componen no necesariamente son las mismas durante el tiempo analizado.

5.1. Formulación matemática de mallas solapadas

Si se desea resolver un sistema de ecuaciones diferenciales en derivadas parciales PDE para un dominio \mathcal{D} , con frontera $\partial\mathcal{D}$ y $\mathbb{N} = 1, 2$ o 3 dimensiones, Fig. (12), utilizando mallas solapadas, es necesario definir un conjunto de mallas \mathbb{M} que cubran todo el dominio \mathcal{D} . El conjunto \mathbb{M} está formado por \mathcal{N} mallas estructuradas \mathcal{M}_m (Guerrero, 2009).

$$\mathbb{M} = \{\mathcal{M}_m\} \quad m = 1, 2, \dots, \mathcal{N}$$

Este conjunto de mallas \mathbb{M} , cubren por completo el dominio \mathcal{D} , superponiéndose entre ellas para lograrlo. Ver figura Fig. (13).

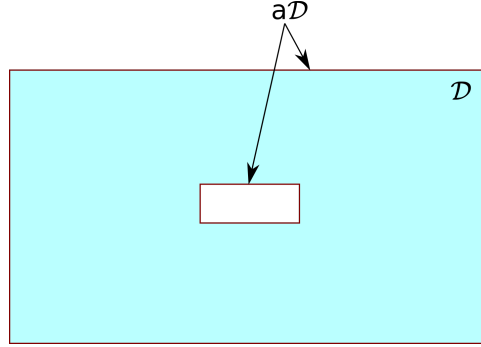


Figura 12: Dominio \mathcal{D} del problema

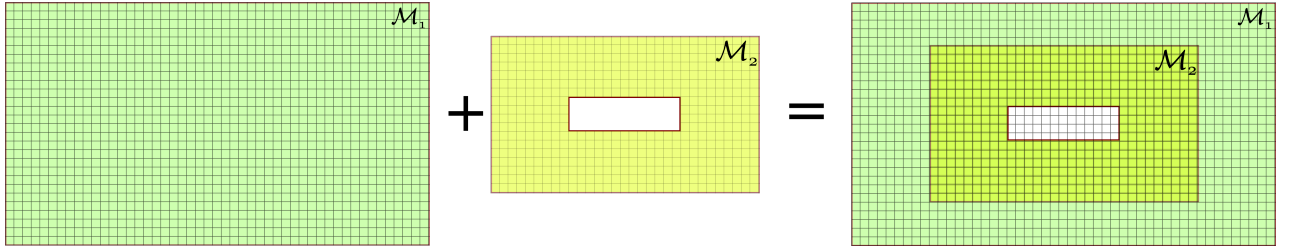


Figura 13: Mallas solapadas $\mathbb{M} = \mathcal{M}_1 + \mathcal{M}_2$

Cada componente de \mathbb{M} es una malla estructurada de \mathbb{N} dimensiones, generada desde el espacio computacional $\mathcal{C} = \mathcal{C}(\xi, \eta, \zeta, \tau)$ a través de un mapeo conforme hacia el espacio físico $\mathcal{F} = \mathcal{F}(x, y, z, t)$.

$$\mathcal{F} = \mathbf{J}_{\mathcal{M}}(\mathcal{C}), \quad \mathcal{C} \in [0, 1]^{\mathbb{N}}, \quad \mathcal{F} \in \mathfrak{R}^{\mathbb{N}}$$

Asimilando a \mathcal{F} como $\vec{x} = (x, y, z)$ con $\mathbb{N} = 3$, este vector contiene las coordenadas físicas, y \mathcal{C} es $\vec{r} = (\xi, \eta, \zeta)$ con $\mathbb{N} = 3$ las coordenadas en el espacio computacional.

Las variables de las componentes del conjunto \mathbb{M} , tales como vértices son guardadas en vectores, donde a su vez estos vértices poseen las información de sus coordenadas.

$$\mathcal{F}_{\mathbf{i}}^{\mathcal{M}}, \quad \mathbf{i} = (i_1, \dots, i_{\mathcal{N}}), \quad i_{\alpha} = \mathbb{N}$$

En el vector $\mathcal{F}_{\mathbf{i}}^{\mathcal{M}}$, las coordenadas de los vértices asociados a la malla móvil irán actualizándose para cada paso de tiempo.

En la Fig. (14) se muestra en detalle un cuarto de región solapada, similar al resto por simetría del ejemplo. En la vista superior se muestran las mallas estructuradas y solapadas en el espacio físico \mathcal{F} y en la vista inferior el espacio computacional \mathcal{C} de cada una de ellas. La condición de borde física impuesta a la malla solapada \mathcal{M}_2 denotada en rojo corta un hueco en la malla de fondo \mathcal{M}_1 , generando los denominados puntos huecos o sin uso en el dominio computacional \mathcal{C} de la \mathcal{M}_1 , este proceso obedece a una jerarquía impuesta manualmente en zonas durante el mallado, identificando a la malla de

fondo con un número de menor orden respecto de los otros componentes de la misma. Los restantes puntos se utilizan para conectar los dominios de las mallas interpolando sus soluciones en ambos sentidos, o simplemente para la discretización del problema. La clasificación y actualización de estos puntos es llevada a cabo por el “solver” para cada paso de tiempo, para mayor detalle de las funciones de la librería `overset` que utiliza `OpenFOAM` el lector puede recurrir a [Petra \(2019\)](#). Las condiciones de bordes físicas impuestas son consideradas como puntos de discretización en conjunto con todos los puntos no definidos como interpolación o huecos, es decir son todos aquellos puntos donde la solución es calculada por el método escogido para discretizar el problema. Existe una condición de borde ficticia generada manualmente durante el proceso de mallado que encierra la malla móvil \mathcal{M}_2 o de menor jerarquía, denominada `oversetPatch`.

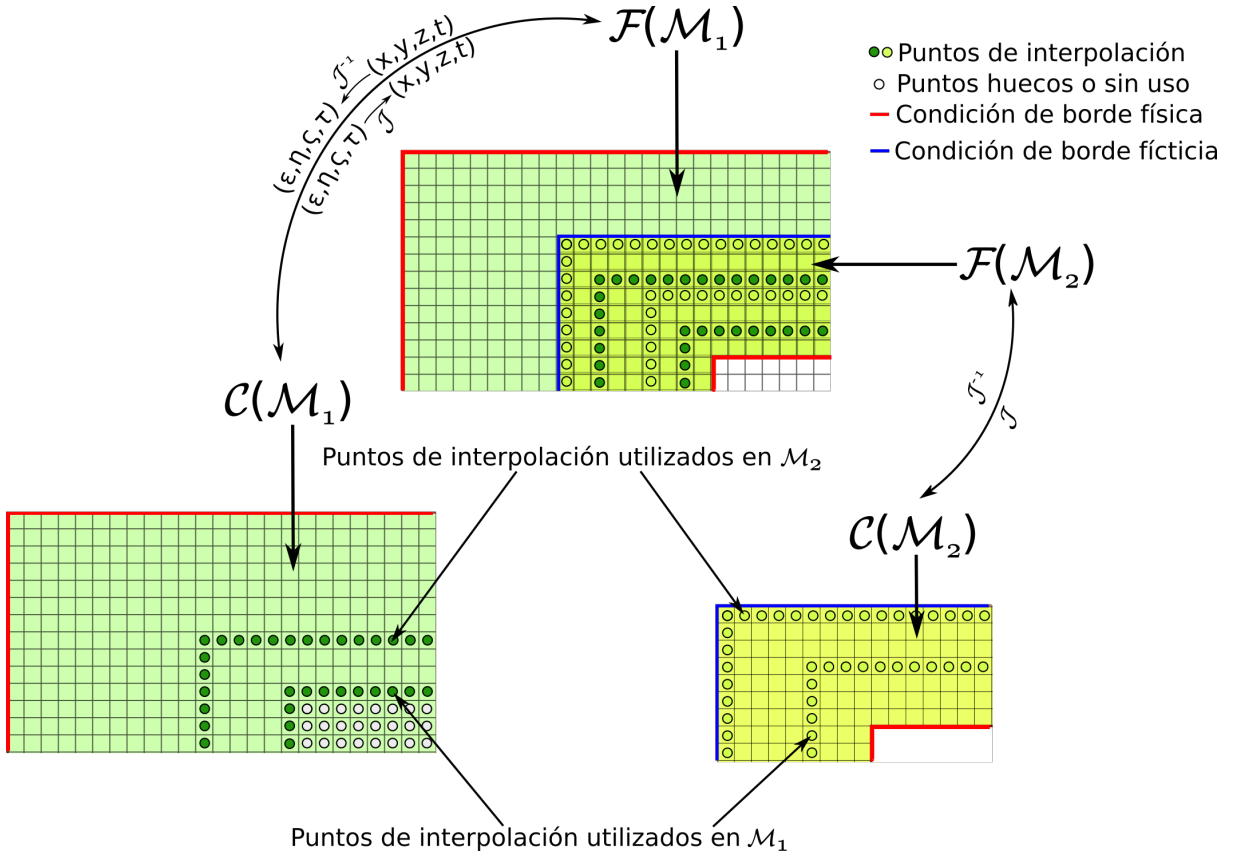


Figura 14: Detalle de región solapada y definición de plantillas

En forma general el proceso de interpolación es llevado a cabo en el dominio computacional \mathcal{C} , la localización de las celdas de interpolación depende de un algoritmo automatizado de búsqueda. Esto hace que la plantilla de interpolación posea una forma y extensión acorde al grado de la función utilizada. La intercomunicación de mallas mediante funciones de interpolación introduce errores de continuidad entre los campos interpolados, los cuales deben ser contemplados en la selección de los esquemas y metodologías de mallado, buscando evitar altos gradientes de forma y campos entre las celdas de interpolación.

En la Fig. [\(15\)](#) se muestra un caso genérico de interpolación para un punto que pertenece a la malla \mathcal{M}_2 y debe ser interpolado a partir de \mathcal{M}_1 . El “solver” detecta las coordenadas computacionales $\mathcal{C}^{\mathcal{M}_1}$ del punto $\mathcal{F}_i^{\mathcal{M}_2}$, mediante el mapeo $\mathcal{C}_i^{\mathcal{M}_1} = \mathbf{J}_{\mathcal{M}_1}^{-1}(\mathcal{F}_i^{\mathcal{M}_2})$, para así luego definir una plantilla de interpolación que encierre al punto analizado, asegurando que los puntos donantes de \mathcal{C}_1 contengan al punto anfitrión $\mathcal{C}_{\mathcal{F}_i^{\mathcal{M}_2}}^{\mathcal{M}_1}$, a fin de evitar

coeficientes de peso negativos usados luego en la función de interpolación seleccionada.

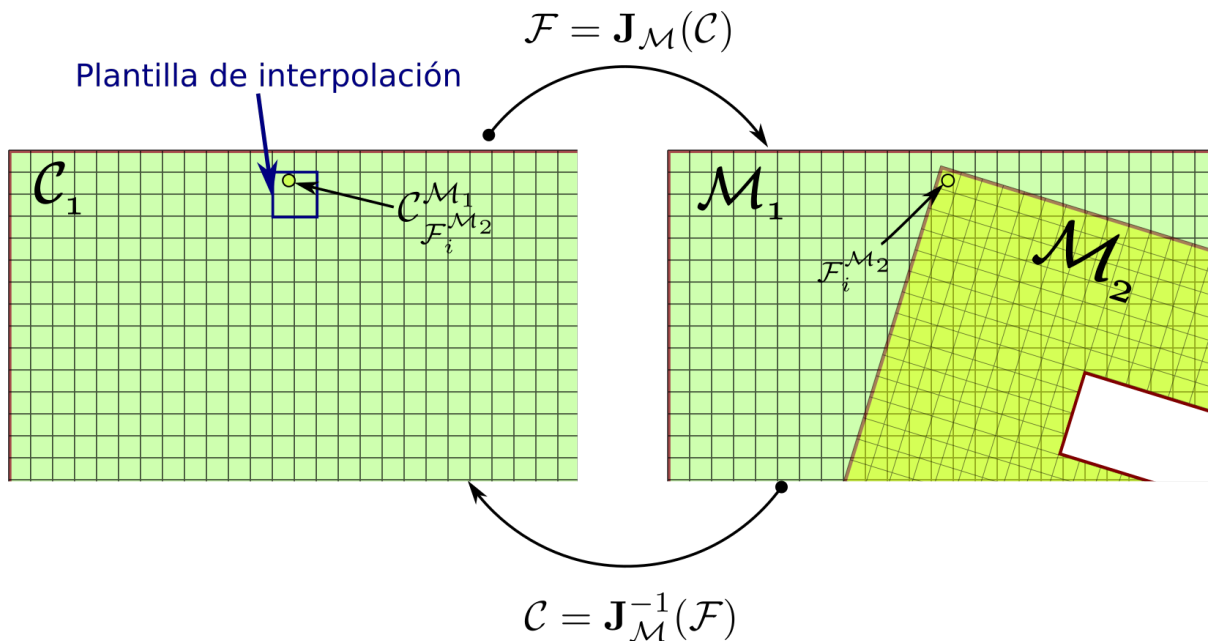


Figura 15: Ejemplo de esquema de interpolación llevado a cabo en el dominio \mathcal{C}

La construcción de la plantilla se realiza en función de las celdas vecinas más próximas en \mathcal{C}_1 al punto anfitrión, estas celdas son denominadas “donantes”. Es evidente y vale la pena citar nuevamente que dependiendo de la estrategia de interpolación adoptada la plantilla adaptará sus formas y extensión, algunas funciones de interpolación de mayor orden requieren de un mayor de puntos donantes.

En este sentido existen dos formas de interpolar mallas solapadas (Petersson, 1999). Implícita y explícita. La primera es cuando los puntos de interpolación de la malla solapada se acoplan con la solución empleando puntos de discretización e interpolación de la malla donante. Esta forma requiere menos solape entre las mallas respecto de la forma explícita, debido a que ésta solo permite que los puntos de interpolación sean donantes o anfitriones. Con la forma implícita se resuelve un pequeño sistema de ecuaciones para actualizar la solución antes de cada paso de tiempo del sistema de ecuaciones diferenciales en derivadas parciales PDEs. Presenta la ventaja de requerir menos puntos de solape. Lo cual en ciertos casos resulta una ventaja, cuando los cuerpos móviles se encuentran cercanos a condiciones de borde de la malla de fondo, obligando a aumentar la densidad del mallado de ésta última con el objetivo de asegurar formar una correcta plantilla de interpolación. Por otro lado la forma explícita simplifica el proceso de solución, pero con un aumento en la cantidad de elementos requeridos en la zona solapada (Guerrero, 2009). El OM en OpenFOAM está implementado implícitamente (Petra, 2019).

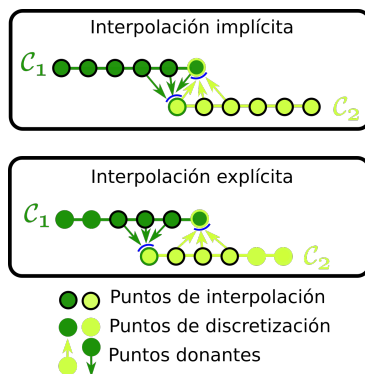


Figura 16: Ejemplo en 1D de malla solapada con interpolación implícita y explícita.

Con los puntos de interpolación utilizados en cada malla, se generan las celdas de interfase o comunicación. `OpenFOAM` cuenta con varias funciones interpolación:

- `cellVolumeWeigth`
- `inverseDistance`
- `leastSquares`

A continuación se detallaran someramente sus fundamentos y expresiones matemáticas para alguna de ellas.

5.1.1. `cellVolumeWeight`

El método `cellVolumeWeight` (CVWI) posee la habilidad de calcular el volumen común de dos celdas solapadas.

El CVWI se basa en el principio que la fracción de volumen solapado entre celdas puede ser utilizado para interpolar o mapear variables en forma conservativa entre mallas solapadas. El volumen solapado entre celdas es usado para determinar la contribución de la celda de base en la mapeada (Coetzee, 2005). La función de interpolación viene dada por la Ec. (54):

$$\phi_{mapeada} = \sum_{n=1}^N (\alpha\phi)_n \quad (54)$$

ϕ es la variable donante para la interpolación (masa, cantidad de movimiento), α la fracción de volumen de la celda de base n y N el número total de celdas solapadas. Se presenta a continuación un ejemplo de demostración simplificado en 2D de profundidad unitaria, en la Fig. (17) la celda roja es la receptora de las cuatro celdas de base donantes. Se suponen conocidos los volúmenes de todas ellas, y el porcentaje que ocupa la celda solapada respecto de cada celda de base.

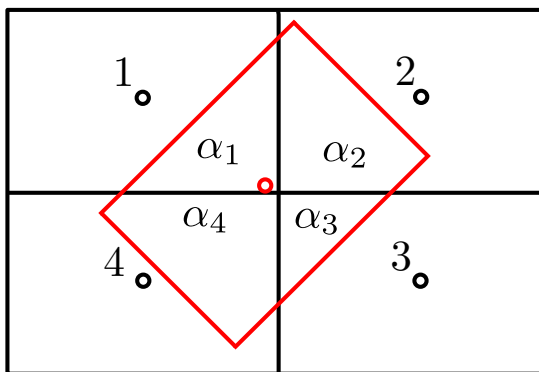


Figura 17: Esquema implementación CVWI. Trazo negro: Celdas donantes. Trazo rojo: Celda receptora

$$\begin{aligned} \phi_1 &= 0,5718255, \phi_2 = 0,3795712, \phi_3 = 0,4298572, \text{ y } \phi_4 = 0,0022126 \\ \alpha_1 &= 0,26, \alpha_2 = 0,25, \alpha_3 = 0,22 \text{ y } \alpha_4 = 0,27 \\ \phi_{mapeada} &= \alpha_1\phi_1 + \alpha_2\phi_2 + \alpha_3\phi_3 + \alpha_4\phi_4 = \boxed{0,34452} \end{aligned}$$

5.1.2. inverseDistance

El método de interpolación `inverseDistance` (IDW) basa su principio de cálculo en las distancias inversas entre la celda receptora y las celdas vecinas inmediatas a ella. La función de pesos es determinada por medio de la distancia inversa desde los centros de celda entre la plantilla receptor-donante.

En la Fig. (18) se presenta un ejemplo 2D para un receptor. La expresión matemática de la función se presenta en la Ec. (55), donde ϕ es la variable donante, d_n distancia entre la celda receptora y la celda donante de base n .

$$S = \sum_{n=1}^N \frac{1}{|d_n|} \quad (55a)$$

$$w_n = \frac{1/|d_n|}{S} \quad (55b)$$

$$\phi_{mapeada} = \sum_{n=1}^N (w\phi)_n \quad (55c)$$

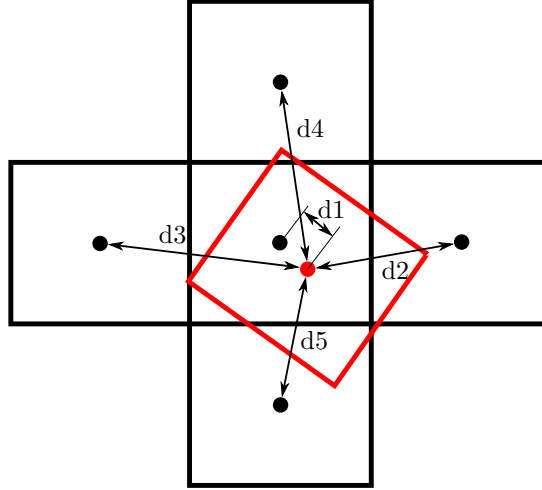


Figura 18: Esquema implementación IDW. Trazo negro: Celdas donantes. Trazo rojo: Celda receptora

$$\begin{aligned}
 d_1 &= 0,22361, d_2 = 0,80623, d_3 = 0,92195, d_4 = 1,2042 \text{ y } d_5 = 1,1180 \\
 S &= 0,23397 \\
 w_1 &= 1,04636, w_2 = 0,29021, w_3 = 0,25378, w_4 = 0,19430, w_5 = 0,20927 \\
 \phi_1 &= 20 \quad \phi_2 = 500 \quad \phi_3 = 30 \quad \phi_4 = -50 \quad \phi_5 = 2 \\
 \phi_{mapeada} &= w_1\phi_1 + w_2\phi_2 + w_3\phi_3 + w_4\phi_4 + w_5\phi_5 = \boxed{164,35}
 \end{aligned}$$

5.1.3. leastSquares

El método **leastSquares** (LS), basa su principio en el cálculo del error mínimo cuadrático de la variable a interpolar entre la celda receptora y las celdas donantes vecinas inmediatas a ella.

Suponiendo un problema 2D, conocidas las distancias d_i entre los centros de celdas receptoras-donantes(a partir de las coordenadas globales) y los valores de las variables donantes ϕ_i , es posible interpolar una variable cualquiera ϕ'_i mediante el análisis de mínimos cuadrados (SSE, por Squared Estimate Errors). En este método se calculan los coeficientes correspondientes y luego interpolan linealmente para obtener el valor de la variable receptora.

Sea \vec{X}_r el vector de coordenadas de la celda receptora que compone la plantilla de interpolación analizada, y $\phi_{mapeada}$ el valor de la variable mapeada respecto de las celdas donantes. El problema puede ser descrito por la Ec. (56).

$$\phi_{mapeada} = \vec{X}_{r1 \times 3} \cdot \vec{\beta}_{3 \times 1} = [1r_xr_y] \cdot [\beta_0\beta_1\beta_2]^T = \beta_0 + r_x\beta_1 + r_y\beta_2 \quad (56)$$

$\vec{\beta}$ es el vector de coeficientes solución el cual es calculado a partir de las distancias entre las celdas receptora-donantes y sus correspondientes valores de $\vec{\phi}$, formando el siguiente sistema de ecuaciones Ec. (57).

$$\vec{\phi} = [d]\vec{\beta} \quad (57a)$$

$$SSE = \sum_{i=1}^n (\vec{\phi} - \vec{\phi}')^2 = (\vec{\phi} - [d]\vec{\beta})^T (\vec{\phi} - [d]\vec{\beta}) \quad (57b)$$

$$\min_{\vec{\beta}} \rightarrow SSE = 0 \rightarrow \frac{\partial SSE}{\partial \vec{\beta}} = 0 \quad (57c)$$

$$[d]^T [d] \vec{\beta} = [d]^T \vec{\phi} \quad (57d)$$

$$\vec{\beta} = ([d]^T [d])^{-1} [d]^T \vec{\phi} \quad (57e)$$

$\vec{\phi}$ es el vector columna de variables aproximadas, $[d]$ la matriz de distancias entre celdas receptora-donantes.

Evaluando el mismo problema de la Fig. (18), la matriz de distancias de celdas $[d]$ formada a partir de las proyecciones sobre los ejes de referencia y el vector de variables de las celdas donantes $\vec{\phi}$ insertos y operados en la Ec. (57e) permiten calcular el vector coeficiente solución $\vec{\beta}$, que luego es utilizado para obtener la variable mapeada con la Ec. (56).

$$[d] = \begin{bmatrix} 1 & d_{1x} & d_{1y} \\ 1 & d_{2x} & d_{2y} \\ 1 & d_{3x} & d_{3y} \\ 1 & d_{4x} & d_{4y} \\ 1 & d_{5x} & d_{5y} \end{bmatrix} = \begin{bmatrix} 1 & 0,2 & 0,1 \\ 1 & 0,8 & 0,1 \\ 1 & 0,2 & 0,9 \\ 1 & 1,2 & 0,1 \\ 1 & 0,2 & 1,1 \end{bmatrix}$$

$$\vec{\phi} = [\phi_1 \phi_2 \phi_3 \phi_4 \phi_5]^T = [200, 500, 30, -50, 2]^T$$

$$\vec{\beta} = [\beta_0 \beta_1 \beta_2] = [172,87688 \quad -0,92965 \quad -156,50754]$$

Las coordenadas de la celda receptora son $\vec{X}_r = [r_x r_y] = [0,2 \quad 0,1]$.

$$\phi_{mapeada} = 172,87688 - 0,2 \cdot 0,92965 - 0,1 \cdot 156,50754 = 157,04$$

5.2. Precisión y conservación de propiedades

La comunicación de resultados entre las mallas solapadas, que es llevada a cabo en la interfaz o plantilla, con la función de interpolación seleccionada, es muy sensible a introducir errores por falta de conservación en los flujos de los campos interpolados. Sin embargo esta no conservación puede ser limitada a valores útiles que no degraden la solución. Para ello las interfaces de comunicación de las mallas solapadas se deben situar en zonas donde la solución sea suave, es decir alejadas de pronunciados gradientes en los campos a interpolar (Part-Enander and Sjogreen, 1994). Existen numerosos trabajos que reportan éxito en la aplicación de una interpolación no conservativa, recopilados por Hadzic (2006). En particular Kasper et al. (2015) presenta una análisis de conservación del valor medio cuadrático para el flujo másico de un caso de mallas solapadas, demostrando buenos resultados. Meakin (1994) sugirió que la precisión espacial y temporal de los métodos de mallas solapados para cuerpos móviles, se ve afectado en realidad por problemas de calidad de malla.

Como se estableció anteriormente la precisión de la solución en las mallas solapadas es afectada por el orden de las fórmulas de interpolación (Cheshire and Henshaw, 1990). Un objetivo en la presente tesis es seleccionar una función de interpolación que asegure una relación de compromiso adecuada entre la simplicidad, precisión y costo computacional, para cada caso estudiado.

5.3. Extensión a problemas de frontera móvil

El fenómeno de interacción fluido estructura FSI matemáticamente pueden ser definido como un problema de frontera móvil y de movimiento relativo entre los dominios analizados. El uso de mallas solapadas para la simulación de FSI, resulta una poderosa herramienta al poder evaluar grandes desplazamientos del sólido o zona móvil, por medio de la actualización automática llevada a cabo por el algoritmo de los puntos de interpolación y los huecos, cuando existe un movimiento entre las mallas.

Cuando las mallas poseen movimiento las derivadas temporales deben ser transformadas. La derivada temporal de una función o cantidad física cualquiera dependiente del tiempo $\phi = \phi(x, y, z, t)$, en un dominio dependiente del tiempo $(x, y, z) = x(t), y(t), z(t)$, para un punto fijo en el espacio físico \mathcal{F} es relacionada con la derivada temporal de un punto fijo en el espacio computacional \mathcal{C} por la regla de la cadena (Wimmer, 1993).

$$\frac{\partial \phi}{\partial t}|_{\mathcal{C}} = \frac{\partial \phi}{\partial t}|_{\mathcal{F}} + \nabla \phi \cdot \frac{\partial \mathcal{F}_i^{\mathcal{M}}}{\partial t}|_{\mathcal{C}} \quad (58)$$

El término izquierdo de la Ec. (58) es el punto fijo en el espacio computacional \mathcal{C} , el primer término del lado derecho de la ecuación es la derivada temporal de ϕ en el espacio físico \mathcal{F} . Notar que este término es el que gobierna las ecuaciones del sistema, y el último término de la derecha está formado por el gradiente de ϕ multiplicado escalarmente por la velocidad de la grilla en el espacio computacional \mathcal{C} . Este último término posee la información de movimiento de una condición de borde, por ejemplo la pared del sólido. En este trabajo el movimiento es asignado al sólido por una función prescripta, denominada `dynamicMeshDict` dentro del entorno `OpenFOAM`, la cual es presentada como una condición de borde en la Ec. (59).

$$\vec{U}(\mathcal{F}_i^{\mathcal{M}}|_{pared,t}) \text{ con } \mathcal{F}_i^{\mathcal{M}}|_{pared} \in \partial\mathcal{D}(t) \quad (59)$$

El movimiento impuesto en la función prescripta puede ser del tipo lineal, circular o trayectorias en seis grados de libertad.

6. Análisis de sensibilidad y configuración del solver

En este capítulo se intenta esbozar la problemática de la resolución de las ecuaciones de Navier-Stokes en flujos transitorios de forma sencilla y genérica, el uso de los algoritmos de acople y estrategias. Los lineamientos aquí expuestos, en búsqueda de una solución óptima, son utilizados en las simulaciones que se llevaron a cabo en los modelos evaluados del presente trabajo.

La resolución de problemas transitorios en CFD, aún para casos sencillos con dimensiones simplificadas y dominios estáticos, resultan en elevados tiempos de cálculo computacional cuando no se utilizan métodos de resolución en paralelo. El costo se incrementa con la cantidad de celdas utilizadas para discretizar el dominio, la inclusión de modelos de turbulencia (RANS, por Reynolds Averaged Navier-Stokes), resolución directa de los torbellinos turbulentos de mayor escala (LES, Large Eddy Simulations) o como en este trabajo la inclusión de mallas móviles solapadas OM.

Para ejemplificar el párrafo anterior se utilizará el caso de testeo denominado `pitzDaily`, con el fin de mostrar las variaciones en cada configuración de acople presión-velocidad del solver `pimpleFoam`, el cual forma parte de `overPimpleDyMFoam`, utilizado en este trabajo Fig. (5)

El dominio está discretizado en 12225 celdas hexaédricas. La geometría utilizada se presenta en la Fig. (19). Se trata de una boquilla donde ingresa el flujo por la sección izquierda y sale por la derecha luego de sufrir una expansión y contracción en sus paredes.

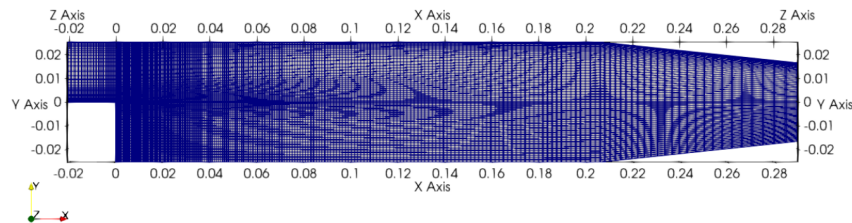
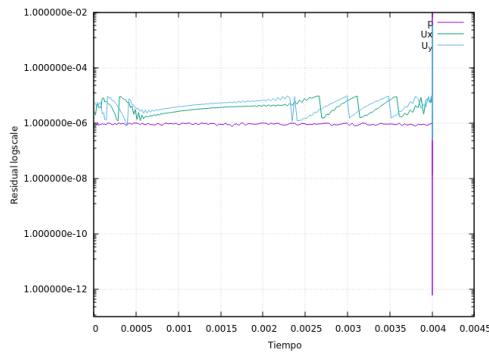


Figura 19: Geometría 2D caso analizado

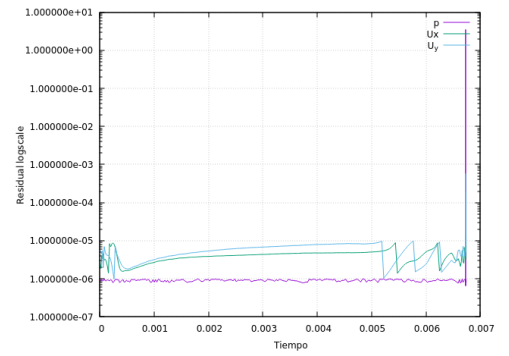
Las condiciones de borde impuestas al caso son: una velocidad de entrada de 10 m/s y una presión relativa de salida $P_{rel} = 0$, con una condición en las paredes de velocidad nula.

Es común evaluar flujos durante tiempos prolongados en busca de determinados comportamientos estacionarios del flujo, coeficientes aerodinámicos, o en búsqueda de valores estadísticos asociados a la movilidad del cuerpo inmerso en el fluido. La demora de la resolución se verá afectada por el tiempo computacional necesario por iteración, pudiendo llegar a tardar horas o días. Es por ello que existe una necesidad de configurar correctamente para cada caso el solver en conjunto con la resolución del problema en paralelo, con el objetivo de disminuir los tiempos de cálculo requeridos.

A continuación se presentan gráficas de errores residuales de corridas en serie del solver para el caso analizado con distintas configuraciones en un marco de tiempo pequeño de 0,05 a modo ejemplo. Se ha utilizado la función control de paso de tiempo ajustable automática `adjustTimeStep`, para mantener un valor de CFL máximo admitido.

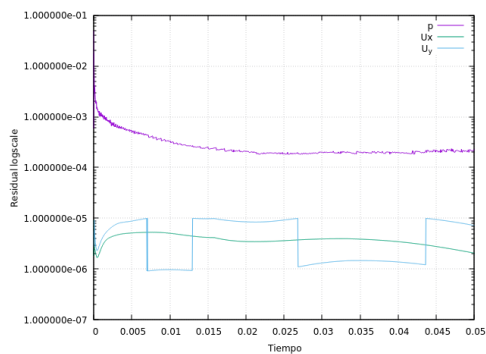


(a) Momentum-predictor=1; pressure-corrector=1; $p=1$, $CFL_{max} = 1$; diverge

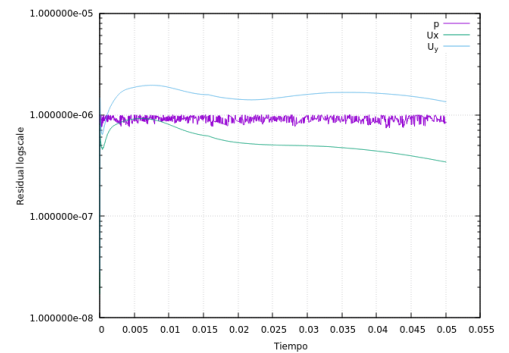


(b) Momentum-predictor=2; pressure-corrector=1; $p=1$, $CFL_{max} = 1$; diverge

Figura 20: Configuración sin relajación ni control de errores residuales

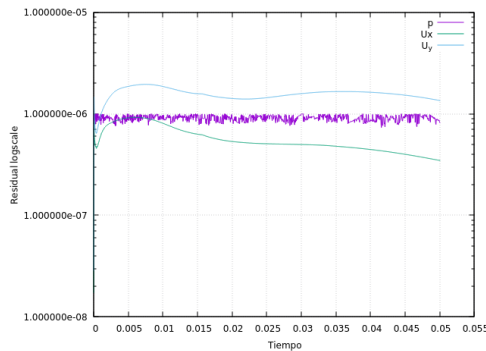


(a) Momentum-predictor=1; pressure-corrector=2; $p=1$; $CFL_{max} = 1$; $t=71s$

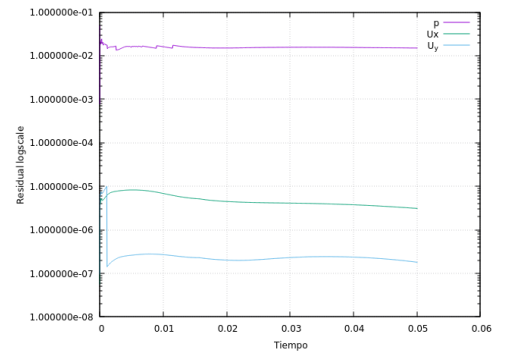


(b) Momentum-predictor=50; pressure-corrector=1; $p=1$; $CFL_{max} = 1$; $URF(p,U)=0.3$; $t=1797s$

Figura 21: Configuración mixta



(a) Momentum-predictor=50; pressure-corrector=1; $p=1$; $CFL_{max} = 1$; $URF(p,U) = 0,3$; $rControl(p,U) = 5e^{-4}, 1e^{-5}$; $t=1089s$



(b) Momentum-predictor=50; pressure-corrector=2; $p=1$; $CFL_{max} = 2$; $URF(p,U) = 0,4$; $rControl(p,U) = 5e^{-4}, 1e^{-5}$; $t=604s$

Figura 22: Configuración con relajación y control de errores residuales

En la Fig. (20a) y Fig. (20b) el solver no pudo resolver el problema con la configuración detallada y luego de algunas iteraciones, la solución diverge. El CFL se dispara bruscamente en ambos casos. En la Fig. (21a) PIMPLE funciona como PISO al operar 2 veces en el bucle corrección de presión, p posee convergencia a valores definidos. En cambio U_y demuestra un comportamiento errático. Estas tres figuras analizadas corresponden al caso donde la configuración del solver es básica, no se utilizan coeficientes de relajación URF, ni control de residuos para las variables calculadas.

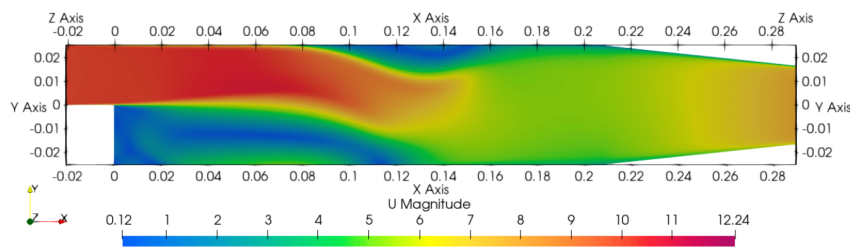


Figura 23: Campo de velocidades 2D obtenida con la configuración especificada en Fig. (22b)

En la Fig. (21b) se utilizan factores de relajación y se aumentan la cantidad de veces que se realiza el loop sobre el **momentum – predictor**. El uso de los URF permite suavizar el avance de la solución, al reducir la propagación del error en caso de que la relación entre los coeficientes del sistema de matrices sea muy abrupta, a costas de un aumento del tiempo insumido.

La Fig. (22a), adhiere a la configuración anterior el control de residuos para las variables, logrando así reducir el tiempo de cálculo necesario.

Mediante el análisis realizado hasta aquí a través de las figuras citadas es evidente la sensibilidad del solver para desarrollar una solución adecuada en calidad y tiempo. Existen casos donde ni aún con la configuración acorde del solver y la paralelización del problema, se logran resultados en tiempos razonables capaces de ser útiles en procesos de diseño o ingeniería. La limitación está definida por la capacidad de cálculo disponible, en conjunto con la eficiencia del solver en paralelizar el problema y comunicar la solución entre los procesadores y nodos de cálculo.

En casos complejos donde se deban acoplar al sistema de ecuaciones clásico a resolver del método FVM, modelos de turbulencia, mallas dinámicas o como en este trabajo mallas solapadas OM, los requerimientos computacionales resultan elevados y se busca reducir el tiempo inferido. Una estrategia es relajar la condición de CFL máximo, a un valor mayor a 1, y así modificar el paso de tiempo, requiriendo menos tiempo de cálculo para la resolución del problema a costas de perder algo de exactitud en alguno de los campos de interés, como sucede con la presión en la Fig. (22b). Esta técnica también puede ser aplicada en problemas paralelizados, buscando controlar que el valor medio del CFL afecte lo menos posible a las zonas de interés donde se busque conocer con detalle algunas variables. Es posible profundizar la información sobre la configuración del esquema de acople presión-velocidad y métodos de relajación, mediante el trabajo de [Holzmann \(2019\)](#).

7. Modelos analizados y técnicas de mallado

Se analizarán casos donde se aplica la metodología OM, para la simulación de la FSI mediante el FVM. Estos son ejemplos clásicos en la ingeniería asociados a flujos externos, los cuales resultan de utilidad para demostrar las virtudes y limitaciones de la aplicación, en casos bidimensionales y tridimensionales. Evaluando el comportamiento ante flujos laminares y turbulentos, sometiendo a los cuerpos a movimientos prescritos, definiendo un acople débil entre fluido y estructura. En todos los casos se considerará a los cuerpos como rígidos indeformables.

También se detallarán metodologías de mallado y buenas prácticas, que aseguren evitar los errores de conservación generados por la interpolación seleccionada del método, compensándolo con mallas más refinadas.

En el presente trabajo se han utilizado dos herramientas de mallado `blockMesh` y `snappyHexMesh` propias de `OpenFOAM`.

7.1. Cilindro 2D

Se analiza el caso de un cilindro bidimensional de diámetro unitario en condiciones de flujo laminar propuesto por Alletto (2022), a fines de contrastar resultados de coeficientes aerodinámicos y frecuencia de desprendimiento de vórtices, que permitan evaluar el comportamiento y solución de la metodología OM implementada en conjunto con la técnica de mallado utilizada.

La incidencia del flujo sobre el cilindro se evalúa en dos etapas, una primera con el cuerpo fijo, y la segunda imponiendo al mismo un movimiento prescrito, definido por una función senoidal de frecuencia y amplitud conocida. El análisis de las dos configuraciones permite especificar la sensibilidad de la metodología OM y la capacidad de resolver los campos de velocidad y presión entre las mallas solapadas.

7.1.1. Malla de fondo

La malla de fondo consiste en un dominio bi-dimensional rectangular sencillo, donde el cilindro se encuentra posicionado en la primer mitad de dominio próxima a la condición de borde de entrada de flujo. No se han definido zonas de mayor densidad de mallado, el tamaño del elemento de base escogido permite capturar el fenómeno eficientemente. En la Fig. (24) se observa el mallado hexaédrico realizado con la herramienta `blockMesh`, en conjunto a las condiciones de borde impuestas al dominio computacional. Estas son las mismas declaradas en la referencia bibliográfica, y aunque existentes las condiciones del tipo `empty` de frente y fondo de la malla no son representadas. La malla cuenta con 28210 celdas, de tamaño base de 15 mm y es definida como zona 0. En la Fig. (25) se presenta la plantilla de celdas utilizadas en la interpolación de campos, la composición por colores es: azules (cálculo), rojas (hueca) y blancas (interpolación).

7.1.2. Malla solapada

Es un recinto bidimensional que contiene al cilindro en el plano “xz”. En este se definieron la condiciones de borde `oversetPatch` y `wall` sobre el cuerpo, dejando un

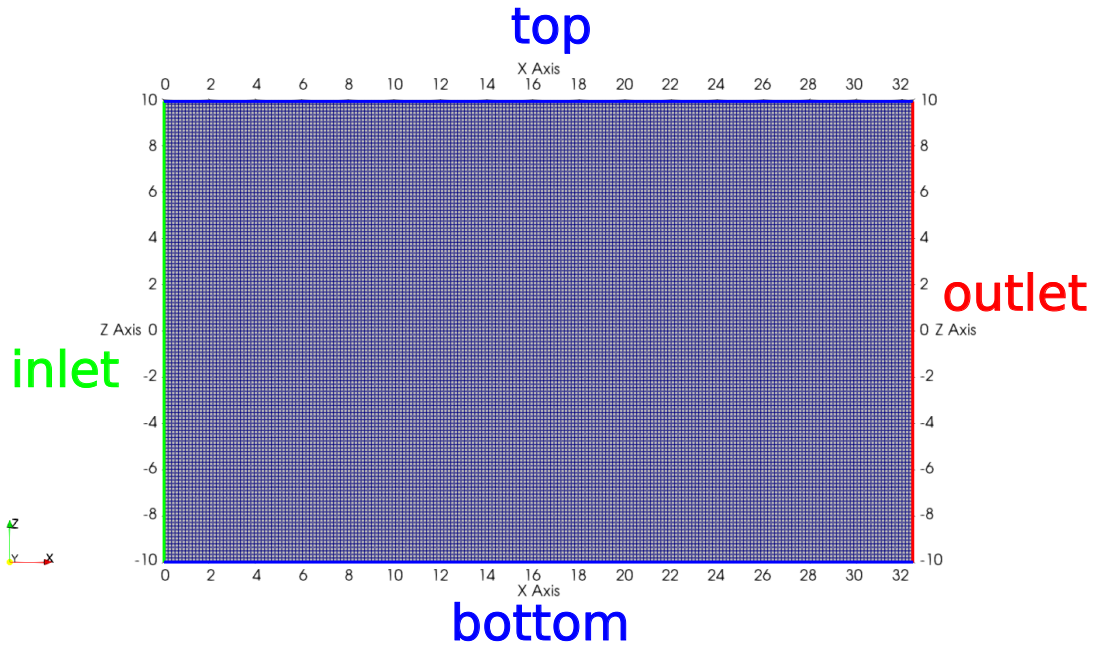


Figura 24: Condiciones de borde impuestas en la malla de fondo

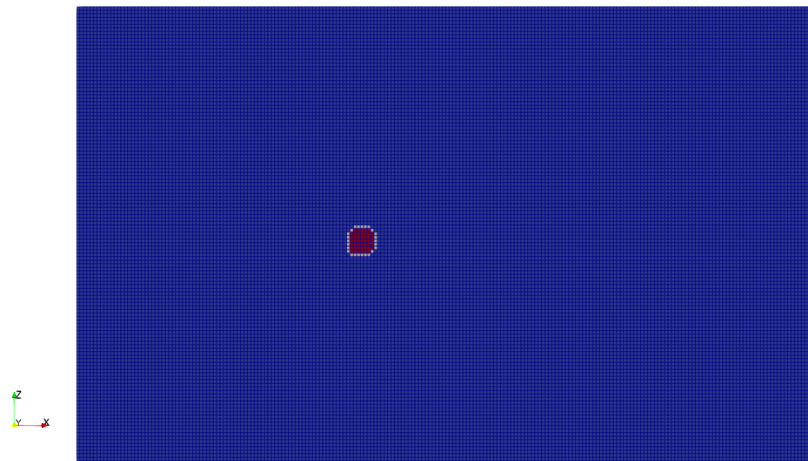


Figura 25: Distribución de celdas utilizadas en la malla de fondo

huelgo de celdas entre ellas a fin de contribuir con la correcta formación de la plantilla de interpolación. Posee un refinado de malla en las cercanías al cuerpo 16 veces respecto de las dimensiones de las celdas de base de 15mm, posee un total de 9625 celdas, ver Fig. (26). Se define a esta malla como zona 1, posee celdas de cálculo azules e interpolación blancas, ver Fig. (27).

El solape entre las mallas se puede observar en la Fig. (28). De esta forma se presentan las mallas y las zonas que componen el método de mallado OM, el gradiente de tamaño de las celdas de interpolación de la zona 1, respecto de las celdas de la zona 0 en la región de solape debe ser el menor posible (igual dimensión de celdas).

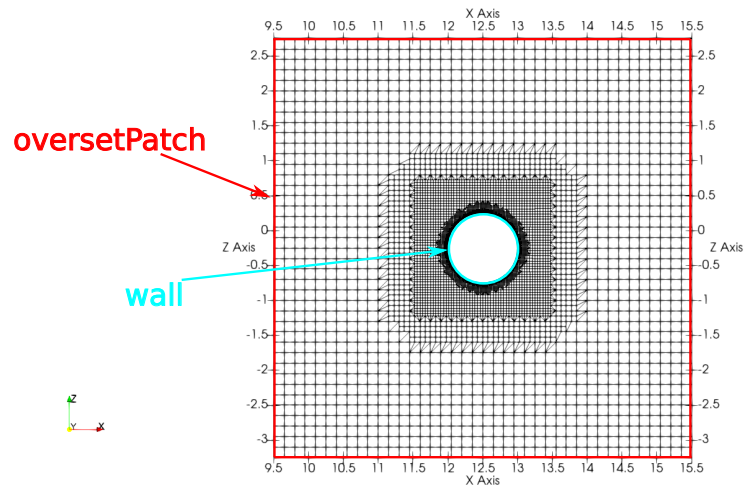


Figura 26: Malla solapada con condiciones de borde y refinamiento de malla

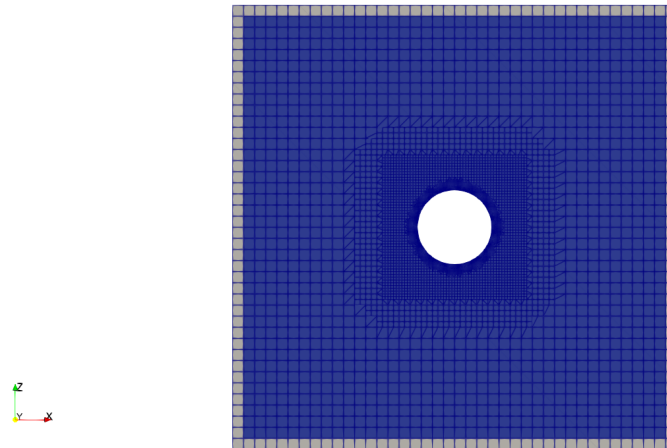


Figura 27: Distribución de celdas utilizadas en malla solapada

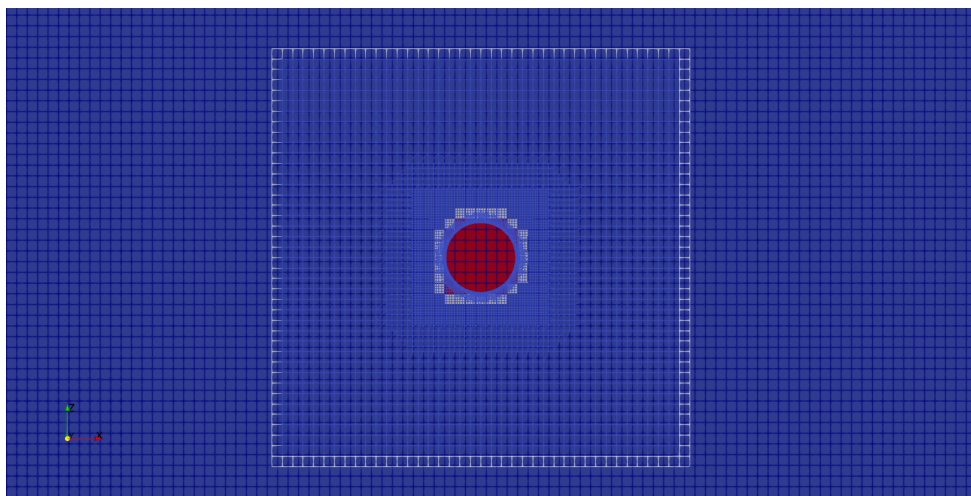


Figura 28: Detalle región de solape

7.2. Perfil Alar 2D

El caso analizado es un tutorial presentado en la [Fourth Midwest OpenFOAM User Group Meeting, \(2019\)](#)

El modelo es un perfil alar de la serie NACA0012 Fig. (29). Esta denominación es ampliamente utilizada a lo largo de la historia aeronáutica por [múltiples aviones](#) y [helicópteros](#).

El perfil alar será evaluado en un dominio bidimensional. El caso es utilizado como un punto de referencia, para demostrar las bondades del método OM, haciendo aletear el modelo dentro del dominio, con un movimiento prescrito en función del tiempo. Es un caso sencillo que permite visualizar el comportamiento ante los grandes desplazamientos del método y evaluar inconvenientes generados a partir del proceso de interpolación de resultados entre las mallas de fondo y solapada.

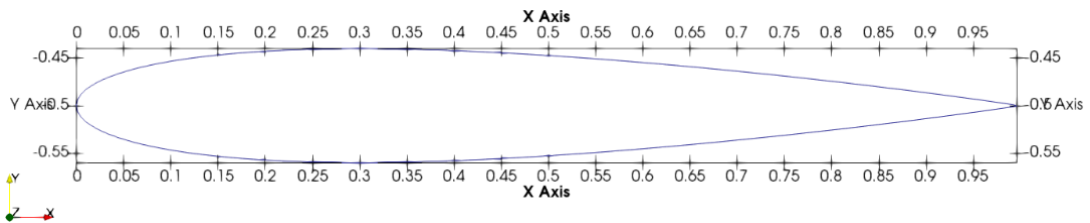


Figura 29: Perfil NACA0012

El primer paso es generar el modelo `.stl`, a partir de las coordenadas características del [perfil](#), se pueden utilizar diferentes generadores de CAD, en esta tesis se ha utilizado [OnShape](#), es una herramienta on-line de uso libre disponible, el cual posee un entorno amigable a partir del cual se pueden exportar los modelos creados en el formato requerido. Además posee una serie de aplicaciones integradas que facilitan el proceso de diseño.

7.2.1. Malla de fondo

La malla de fondo posee las dimensiones del dominio completo donde se desplaza el perfil alar y se produce la entrada y salida de flujo. Las dimensiones del mismo deben asegurar que durante la simulación no se produzcan efectos de bloqueo dinámico, que afecten a los campos de velocidad y presión. Se recomiendan relaciones de secciones frontales entre el dominio de fondo y el cuerpo, menores al 5%.

En la Fig. (30) se aprecia el mallado estructurado, y las condiciones de borde impuestas. Vale aclarar que no se presentan las condiciones de frente y fondo en la figura analizada. Esta malla o zona, posee un total de 24000 celdas, con un tamaño de celda de 10 mm. Se define a esta zona la identificación de zona 0.

En la Fig. (31) se observa la plantilla de interpolación en la malla de fondo celdas blancas, en conjunto con las celdas desactivadas rojas y las celdas para el cálculo azules. Cabe recordar que esta plantilla es actualizada por el algoritmo automático del solver, acorde al movimiento impuesto a la malla solapada en cada paso de tiempo.

7.2.2. Malla solapada

Esta contiene el perfil alar NACA0012. En el mallado se genera una brecha de celdas entre la condición de borde denominada `oversetPatch` y el perfil `wall`. La brecha asegura

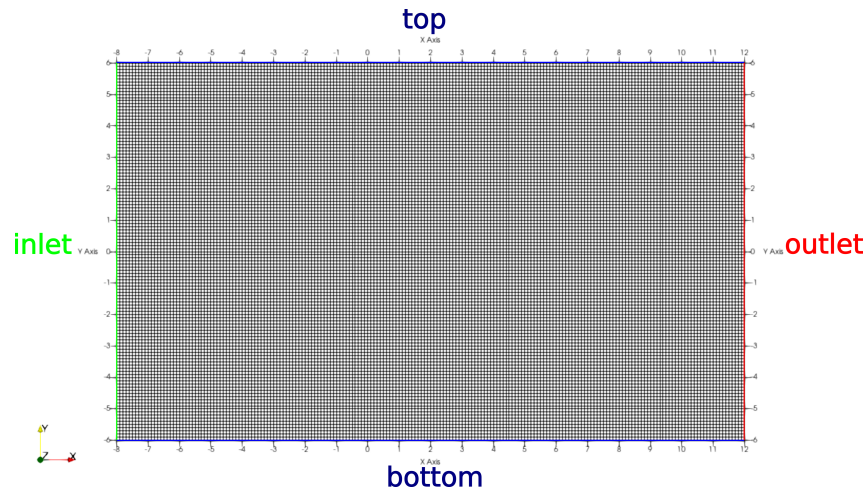


Figura 30: Condiciones de borde impuestas en la malla de fondo

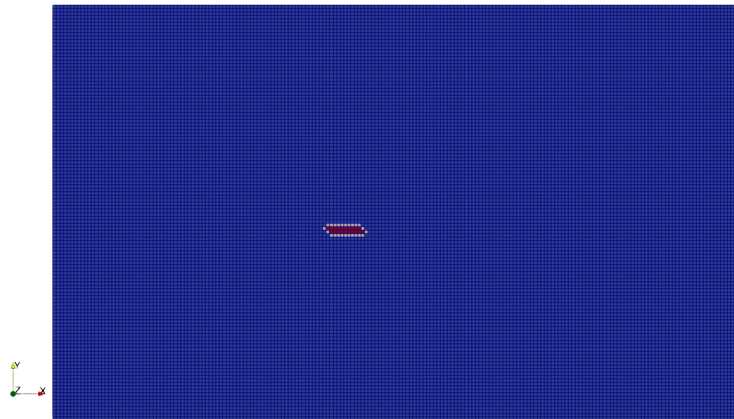


Figura 31: Distribución de celdas utilizadas en la malla de fondo

una cantidad mínima de celdas que permitan realizar una correcta definición de los puntos huecos y de interpolación (celdas) (Guerrero, 2019). Además se cumplen los requerimientos de refinado de malla, a fin de captar con mayor detalle los gradientes producidos en los campos por la presencia del cuerpo en el flujo. La malla solapada se identifica como zona 1 y posee 3024 celdas, ver Fig. (32).

En la Fig. (33) se presenta la distribución de celdas utilizada en el OM, para la malla solapada. En esta zona solo se encuentran presente celdas de cálculo (azules) y de interpolación (blancas) en el `oversetPatch`. El solape producido entre las mallas de las zonas 0 y 1 se presenta en la Fig. (34), obsérvese como las celdas inactivas de la malla de fondo contienen todo el contorno del cuerpo de la malla solapada. El proceso de mallado para OM, muchas veces resulta una arista dentro del análisis que puede desembocar en diversos problemas. Entre ellos la convergencia de residuos, propagación de errores por difusión numérica. Los problemas de no conservación, producidos por efectos de la interpolación realizada en la plantilla, deben ser evaluados cuidadosamente. La sensibilidad de éstos problemas dependen del grado de la función de interpolación utilizada y el gradiente de tamaño entre las celdas de las zonas.

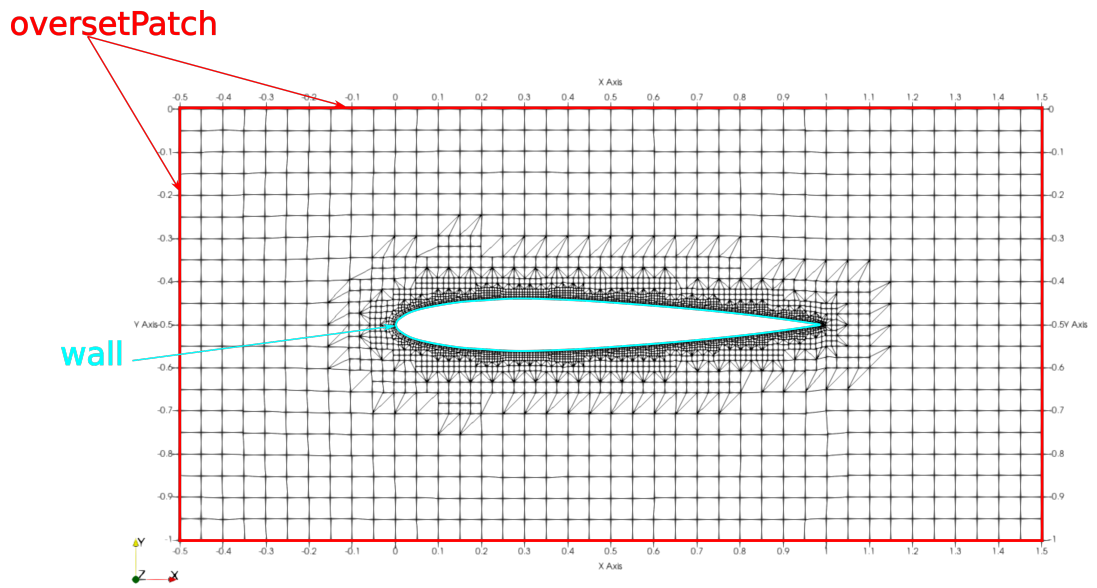


Figura 32: Malla solapada con condiciones borde impuestas

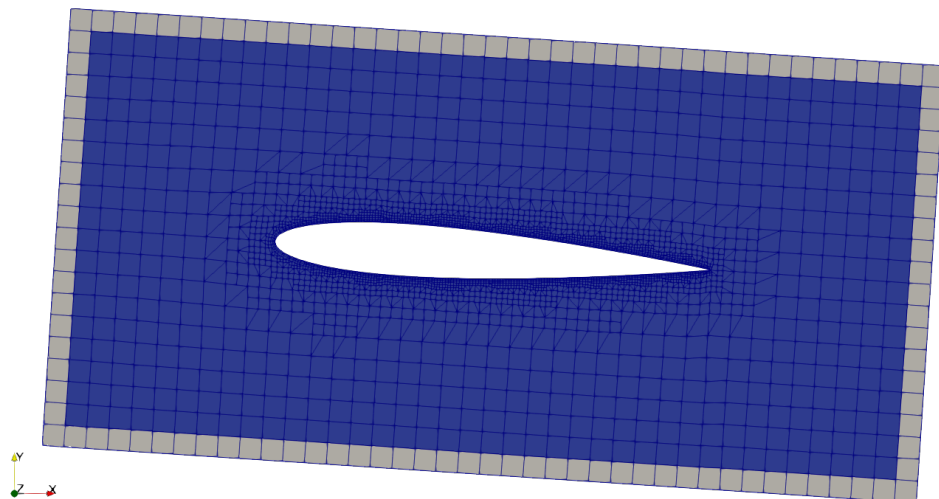


Figura 33: Distribución de celdas utilizadas en malla solapada

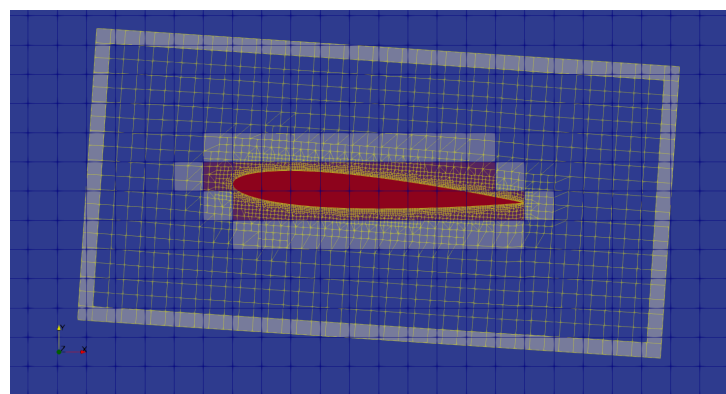


Figura 34: Detalle de la zona solapada

7.3. BARC

El BARC es un modelo utilizado como referencia para contribuir en el análisis del flujo turbulento separado de un cilindro rectangular fijo de relación cuerda profundidad igual a 5. En Ingeniería Civil sirve para proporcionar información aerodinámica de un amplio rango de cuerpos romos, tales como tableros de puente esbeltos, edificios de gran altura y otros (Bartoli et al., 2008; Bruno et al., 2014).

El modelo será analizado en un dominio bidimensional, al suponer simetría de flujo en dirección de la profundidad. Las dimensiones del cuerpo se presentan en la Fig. (35) acorde al trabajo de Wu et al. (2020). El análisis consta de dos etapas, una primera sin imposición de movimiento a la malla solapada con el fin contrastar los coeficientes aerodinámicos, y la segunda imponiendo una velocidad angular alternativa senoidal en el centro de gravedad del cuerpo, sobre uno de los ejes. El objetivo es conocer el comportamiento de la metodología OM, los resultantes campos de presión y velocidad, en conjunto con la variación de los coeficientes aerodinámicos en el tiempo, para un caso de flujo turbulento.

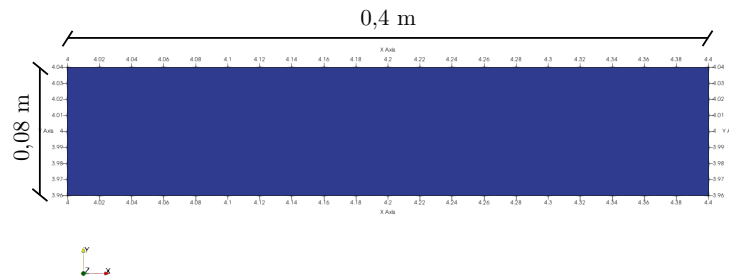


Figura 35: Geometría del BARC. Relación de aspecto 5:1

7.3.1. Malla de fondo

Posee las dimensiones definidas acorde a Wu et al. (2020), en la Fig. (36) se observa la densificación de mallado adoptada para este trabajo, se definieron dos regiones. El tamaño de las celdas de fondo es de 0,25 m, el de la región `refinementBox_a` es 32 veces menor y el de `refinementBox_b` es 64 veces menor a las celdas de fondo. Las condiciones de borde impuestas aseguran un flujo sin restricciones como sería el caso de un tablero de puente suspendido, el flujo tiene sentido izquierda a derecha, con una velocidad de entrada definida y una presión de salida, tanto el techo como en el piso se utilizan condiciones de borde del tipo simétricas, en estas el vector velocidad mantiene su magnitud y dirección, en cambio su componente normal es nula, esto resulta en la ausencia del efecto de corte en la dirección normal a la condición de borde (Moukalled et al., 2016). A pesar de no especificarse en la gráfica, existen condiciones de borde del tipo vacías `empty`, para la parte anterior y posterior del dominio presentado, condición necesaria en `OpenFOAM` para simular flujos bidimensionales, al menos tener una profundidad de una celda. La malla está discretizada con un total de 131436 celdas. Se la identifica como zona 0.

Los objetivos de la estrategia de mallado adoptada Fig. (37), son buscar el menor gradiente en las dimensiones entre los elementos de las celdas de ésta malla y la solapada a fin de evitar problemas en el proceso de interpolación, capturar el desprendimiento producido en las aristas vivas del modelo y modelar con buena definición el callejón de vórtices de von Kàrmàn que se forma en la estela del cuerpo.

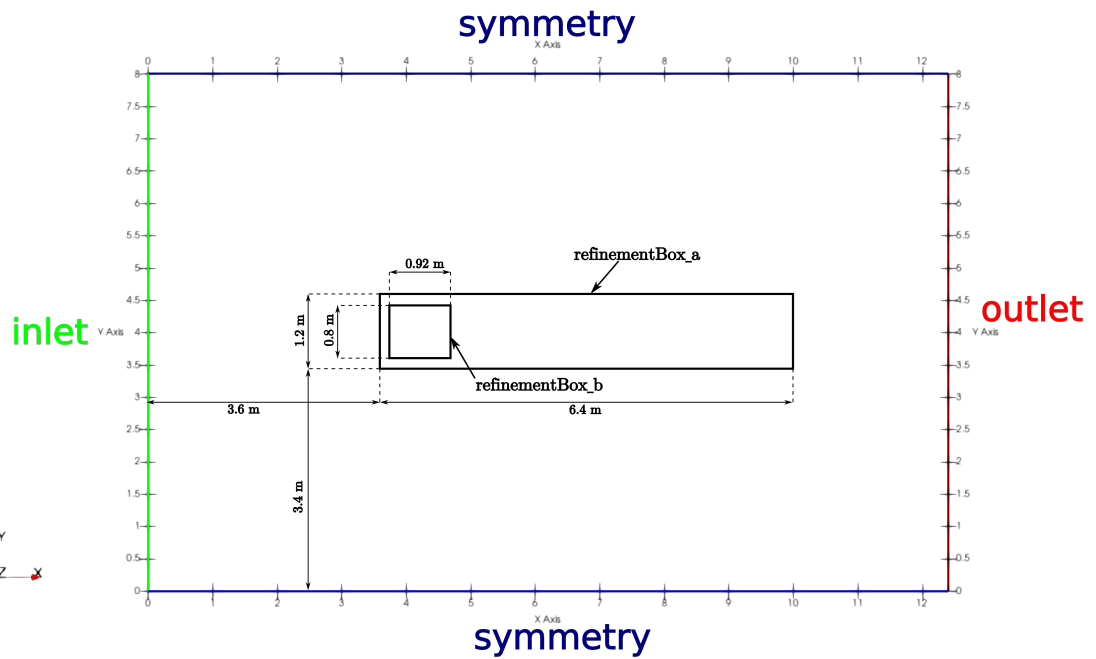
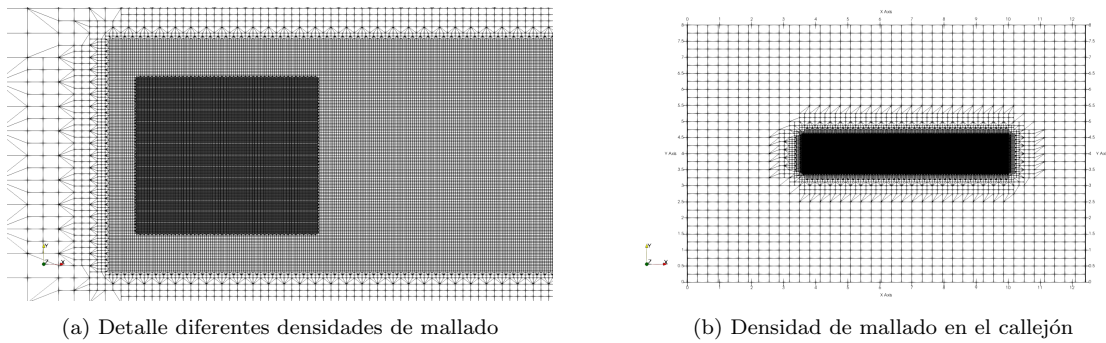


Figura 36: Dominio computacional malla de fondo

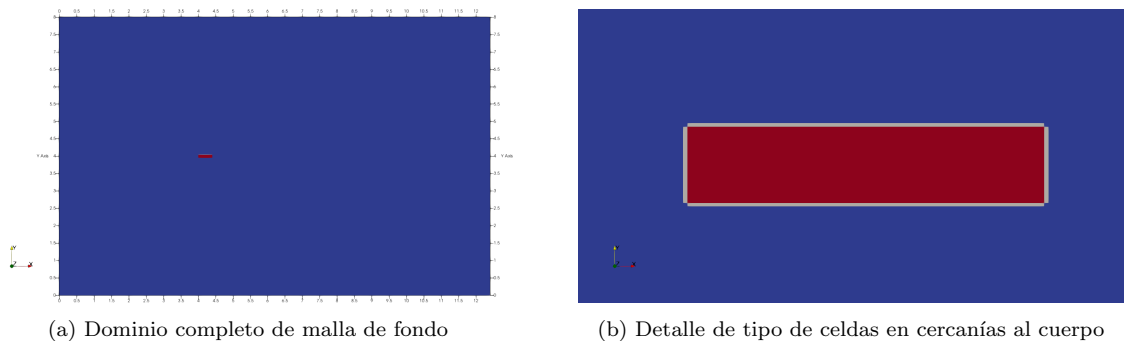


(a) Detalle diferentes densidades de mallado

(b) Densidad de mallado en el callejón

Figura 37: Densidades malla de fondo

Los tipos de celdas al aplicar OM en la malla de fondo se presentan en la Fig. (38), las celdas de cálculo azules, rojas inactivas y blancas de interpolación.



(a) Dominio completo de malla de fondo

(b) Detalle de tipo de celdas en cercanías al cuerpo

Figura 38: Tipos de celdas en la malla de fondo

7.3.2. Malla solapada

Esta malla luego será solapada con la de fondo para formar una sola, definiendo en ella un movimiento prescrito que acompañe al cuerpo denominado BARC. El tamaño de las

celdas en la región más lejana al cuerpo es de 3,98 mm. El refinamiento tanto superficial como de los bordes del cuerpo es 8 veces respecto de las celdas lejanas, se adoptan para el cambio entre niveles de densidad 2 celdas. En la Fig. (39), se observa la discretización y las condiciones de borde impuestas: el `oversetPatch` en el contorno del dominio y `wall` sobre el BARC, así como la densificación de malla debido a la capa límite. Se asigna como zona 1 a esta malla, posee 28484 celdas hexaédricas.

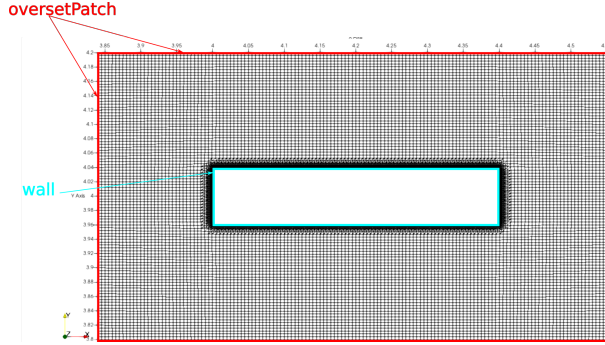


Figura 39: Dimensiones dominio malla solapada, discretización y condiciones de borde

La discretización de la capa límite se realiza en 6 celdas, con una altura total de 1,4 mm y una relación de crecimiento desde el cuerpo a la última celda de la capa límite de 1,2. El tamaño de la celda adherida al cuerpo es un detalle a cuidar en el proceso de malla. El modelado del perfil de velocidad sobre la superficie del cuerpo depende de la primera celda. Con el objetivo de lograr resultados adecuados y evitar un proceso de ensayo error en la definición de este tamaño, se define y analiza la variable y^+ (Liu, 2016), ver Fig. (40a).

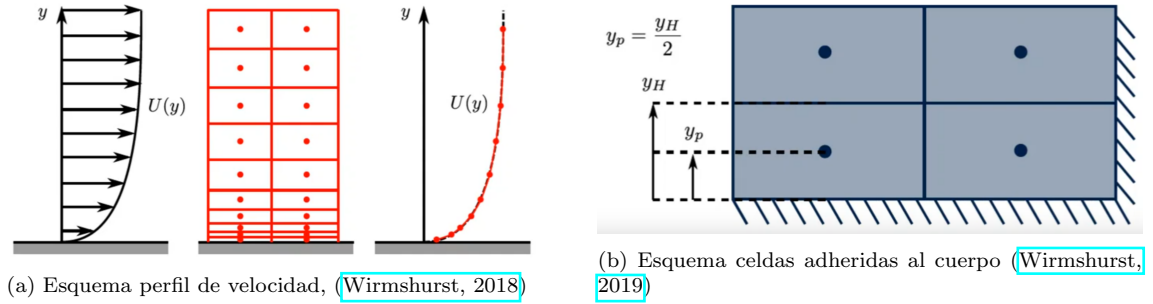


Figura 40: Configuración de celdas en la capa límite

$$y^+ = \frac{y_H u_t}{\nu} \quad (60)$$

y_H es la altura de la celda evaluada desde la pared, u_t velocidad de fricción y ν viscosidad cinemática.

El valor de y^+ de la primera celda depende de las condiciones de flujo Re , y la relación $y_H = 2yp$, donde y_H es la altura de la primera celda y yp la altura de su centro. Es posible determinar esta dimensión bajo la suposición de análisis que la capa límite se asemeja a aquella desarrollada en un flujo sobre un plato plano (Schlichting and Gersten, 2017).

En el trabajo de Wu et al. (2020) la velocidad de entrada es de 10m/s, la cuerda del BARC $L = 0,4m$ y el flujo es aire en CNTF. El flujo se caracteriza por un $Re = 3,872 \times 10^5$, aplicando el análisis antes mencionado para una $y^+ = 3$, se obtiene $y_H = 0,00013134m$

objetivo para la primer celda. La malla actual utilizada posee un valor promedio en sus primeras celdas de $0,00014m$, cercano al calculado. Inicialmente el valor de y^+ no se conoce por lo tanto se recurre al supuesto aquí utilizado, luego en la simulación se verificará el valor que adquiera con las iteraciones, verificando que cumpla los requerimientos en función a la estrategia utilizada, normalmente se desean valores inferiores a 5, o de entre 30 a 300, dependiendo del modelo adoptado de función de pared, el cual será detallado más adelante.

Esta zona posee una caracterización de celdas utilizadas en el OM, en la Fig. (41) se presentan las celdas de cálculo azules y de interpolación blancas.

El detalle de solape entre mallas es presentado en la Fig. (42), para este modelo se respetó la recomendación de igualdad de tamaño entre las celdas de la zona 0 y zona 1 en la región del `oversetPatch`. Buscando evitar problemas de difusión numérica por la interpolación de resultados entre las mallas en los pasos de tiempo (Guerrero, 2019). Las celdas de la zona 1 están representadas con líneas amarillas, para un mejor contraste.

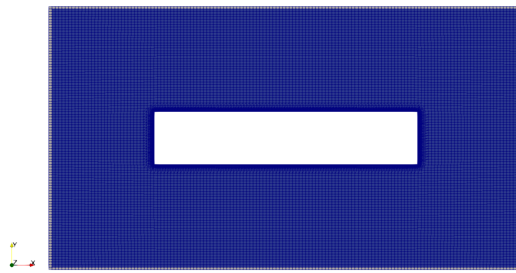
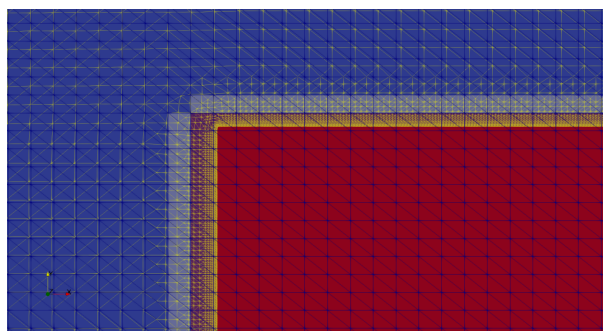
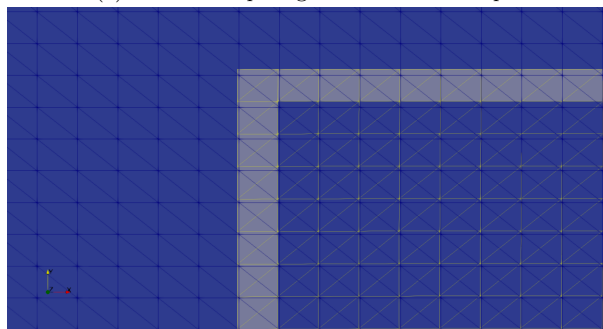


Figura 41: Distribución de celdas OM en malla solapada



(a) Detalle solape región cercana al cuerpo.



(b) Detalle solape región `oversetPatch`

Figura 42: Detalle región solapada entre mallas

7.4. Cuerpo de Ahmed

El cuerpo de [Ahmed et al. \(1984\)](#) es un modelo estándar propuesto para el estudio de vehículos terrestres, sirviendo como punto de referencia para calibrar túneles de viento, códigos computacionales, y para extrapolar resultados a fines de diseño o industriales.

El modelo aún al estar formado por líneas simples, reproduce el flujo turbulento de la estela, separaciones de capa límite y re-ingresos de flujo, similares a los encontrados en vehículos terrestres.

La motivación para el análisis del modelo con un ángulo de inclinación de cola de 35° , se debe a que la configuración de la estela turbulenta que se produce aguas abajo es bastante más compleja que la obtenida por otros ángulos estudiados, por efecto del desprendimiento temprano de flujo producido en la cola.

Con el estudio en condiciones de viento alineado y cruzado, se busca entender con mayor detalle la estela de flujo turbulento producida, manteniendo la idea de que ésta posee características similares a las estelas turbulentas de una amplia gama de vehículos terrestres.

La tendencia en el diseño de automóviles modernos es primar la baja resistencia aerodinámica en pos de la eficiencia, para la reducción de consumo de combustible y definir un estilo aero-estético. Es así que las superficies son más sensibles a la condición de viento cruzado, generando un aumento brusco del coeficiente de sustentación en parantes traseros ([Rossitto et al. \(2016\)](#)). La variación en la distribución del coeficiente de sustentación (Cl) entre la parte anterior y posterior del vehículo reducen su estabilidad. Un efecto similar se produce en el arreglo turbulento al realizar maniobras de alce o slalom, representadas por un cambio en el ángulo de guiñada en el tiempo.

En este trabajo se busca reproducir la condición de flujo antes mencionada con la técnica OM, imponiendo un movimiento prescrito senoidal alternativo al cuerpo Ahmed y malla solapada adherida a él, con el fin de simular la configuración turbulenta del flujo ante una variación de ángulo de guiñada en el tiempo, con las mismas condiciones de viento incidente especificadas según [Meile et al. \(2016\)](#) para luego contrastar resultados.

Para este trabajo las patas del cuerpo de Ahmed son removidas, a fin de garantizar el correcto funcionamiento de OM, asegurando así la cantidad mínima de celdas requeridas para una correcta interpolación de resultados. Además, evitando que las mallas compartan la condición de borde impuesta al suelo. En la Fig. (43) se presenta el modelo completo a fin de aclarar la disposición del centro de fuerzas utilizado para el cálculo de los coeficientes aerodinámicos y el centro de rotación impuesto a 0,169 m por encima del sistema graficado.

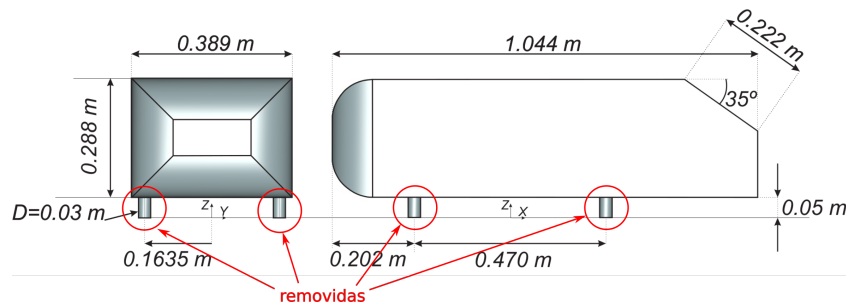


Figura 43: Cuerpo de Ahmed - slant 35°

La discretización del dominio se ha llevado con la herramienta denominada **snappyHexMesh** disponible en **OpenFOAM**. La sección transversal mayor del dominio computacional corresponde a la del Túnel de Viento Aerodinámico de baja velocidad perteneciente al "ISW at Graz University of Technology", del tipo Göttingen, de sección 3/4 abierta. Para mayores detalles referirse al trabajo de [Gretler and Meile \(1993\)](#).

En el proceso de mallado se tienen en cuenta metodologías y buenas prácticas, que aseguren evitar los errores de conservación generados por la función de interpolación seleccionada del método, buscando compensarlo con mallas más refinadas ([Verma and Hemmati, 2020](#); [Chandar et al., 2018](#)). A continuación se especificarán dividiendo al mallado de fondo y solapado.

7.4.1. Malla de fondo

La discretización más gruesa se realiza con celdas hexaédricas de dimensión promedio $L/17$ donde L es la longitud del cuerpo de Ahmed. Se define una densificación de mallado en la dirección del eje "z" con el propósito de aumentar la cantidad de celdas en la zona de venturi generada entre el suelo del túnel y el del Ahmed, a fin de lograr mayor precisión en la determinación de los posibles gradientes en el campo de la aceleración y los efectos de la interacción de las capas límites que se generan en esta región de flujo.

A fines de disminuir efectos difusivos producidos en la interpolación entre mallas por la aplicación de la técnica OM, se generan dos regiones con mayor densidad de mallado etiquetadas como I y II, con el objetivo de asegurar la confluencia entre las mallas con el menor gradiente de tamaño posible utilizándose 3 celdas entre regiones para difuminar los niveles de densidad. Las relaciones de refinamiento son definidas en forma escalonada, en la región I las celdas varían su tamaño entre $L/138 \leq L \leq L/69$, y en la región II se definen celdas de tamaño iguala $L/35$. La extensión de estas regiones son definidas a partir del giro máximo esperado de la malla solapada de tal forma que esta última quede cubierta siempre dentro de la región I, ver Fig. (44).

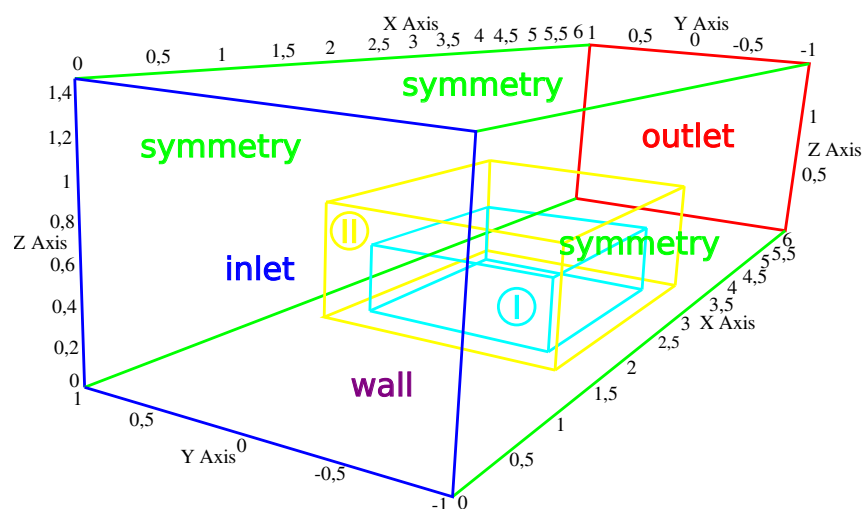


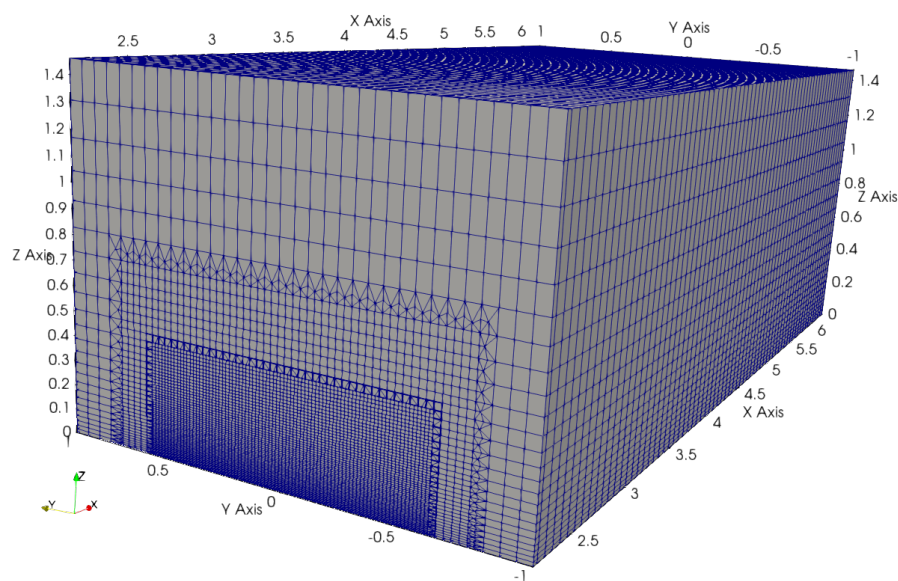
Figura 44: Dimensiones, regiones de refinamiento y condiciones de borde

Las condiciones de borde impuestas permiten simular el efecto de un túnel 3/4 abierto. La condición de borde denominada simetría permite que el vector velocidad en ella, mantenga su magnitud y dirección, definiendo a la componente normal nula.

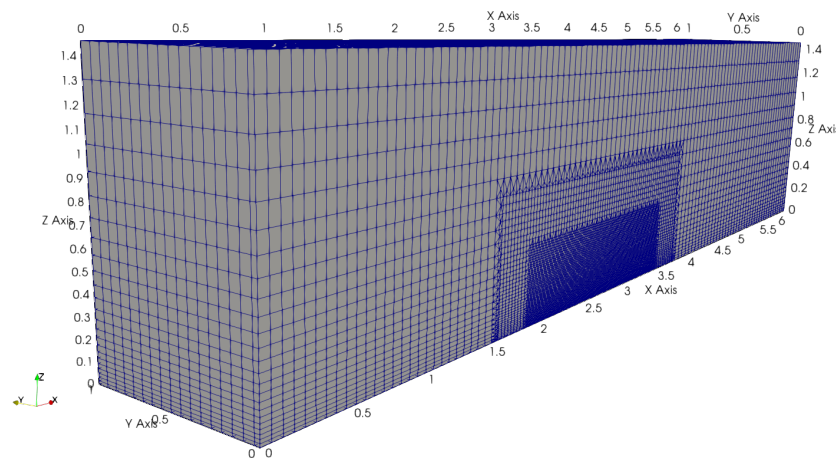
Esto resulta en la ausencia del efecto de corte en la dirección normal a la condición de borde (Moukalled et al., 2016), logrando simular el efecto de los límites inexistentes de la cámara de ensayo ocupados por aire.

La malla cuenta con un total de 619128 celdas hexaédricas, es definida como zona 0 en la aplicación de la técnica OM.

Se presentan en la Fig. (45) vistas de cortes Fig. (45a) y Fig. (45a), de la malla de fondo. El bloqueo aerodinámico varía entre 3.8 y 7.5%, en función a la variación de la sección frontal para los distintos ángulos de guiñada.



(a) Detalle mallado plano "yz"



(b) Detalle región mallado plano "xz"

Figura 45: Detalle regiones de mallado en varios cortes

La metodología OM trabaja para este caso con una plantilla de celdas tridimensionales, el algoritmo las define automáticamente en función de las condiciones de borde y asignación de zonas en las mallas, en la Fig. (46) se observan éstas celdas, azules de cálculo, blancas de interpolación y rojas inactivas. Las celdas rojas contienen todo el cuerpo de Ahmed dentro de ellas, y se irán actualizando acorde al movimiento impuesto en cada paso de tiempo. El número de tipos de celdas irán variando conforme avanza la simulación, se debe evitar una variación brusca en éstas.



Figura 46: Celdas utilizadas en la plantilla OM

7.4.2. Malla solapada

En esta región se encuentra embebido el cuerpo de Ahmed, con un tamaño de celdas hexaédricas de base $L/104$, el refinamiento sobre los bordes y superficie del Ahmed es la mitad estas. La capa límite es discretizada en 6 niveles, con una relación de expansión de 1,2, el espesor total es de 1,397 mm, y el tamaño de la primer celda es de 0,141 mm, algo mayor al valor 0,090060mm, resultante del cálculo teórico (Schlichting and Gersten, 2017), el cual permite determinar la altura yH estimativa de la primer celda para un $y^+ \leq 5$, valor considerado ideal.

La malla solapada cuenta con un total de 565134 celdas hexaédricas, distribuidas dentro de la denominada zona 1 por la metodología OM.

Las condiciones de borde impuestas en esta malla son dos: tipo `wall` en el cuerpo de Ahmed y `oversetPatch` a la caja que contiene al cuerpo Fig. (47). En ésta última es donde se realiza el proceso de interpolación con la malla de fondo, el algoritmo del método OM define la plantilla de celdas de cálculo azules e interpolación, identificadas en la Fig. (48).

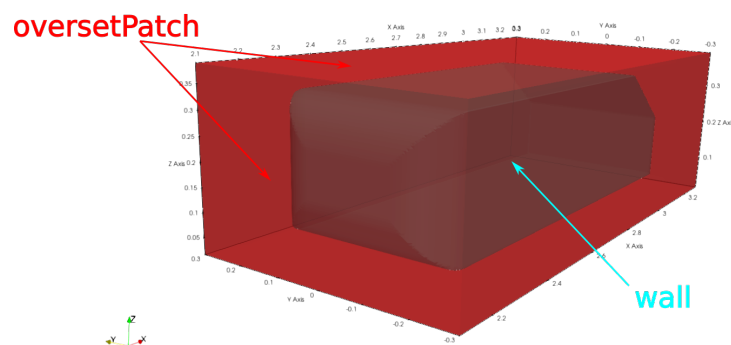


Figura 47: Condiciones de borde impuestas

Un reto del mallado 3D como el del modelo presentado, es aplicar la técnica OM en huecos pequeños, como el existente entre el suelo del túnel y el piso del cuerpo de Ahmed. En esta región coexisten las celdas de interpolación de ambas zonas, forzando

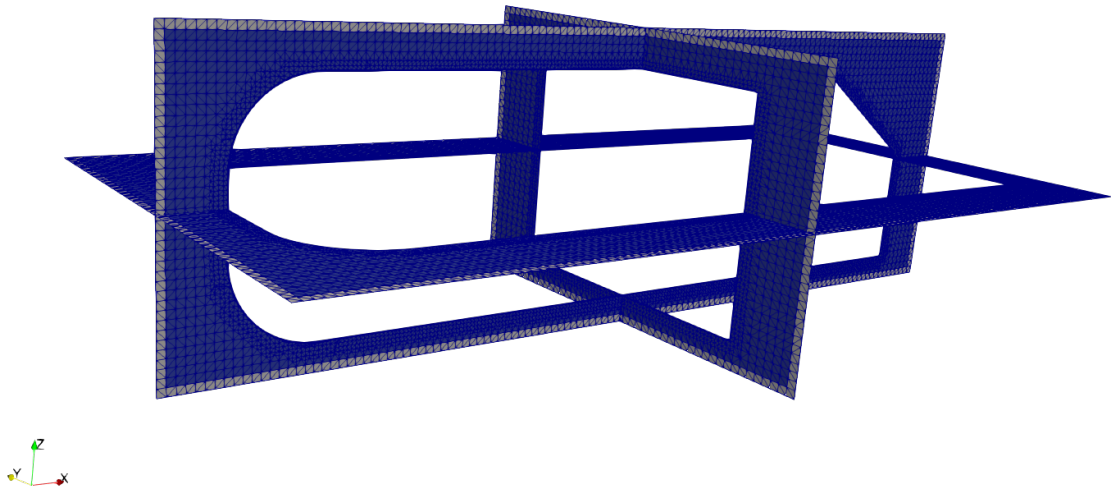
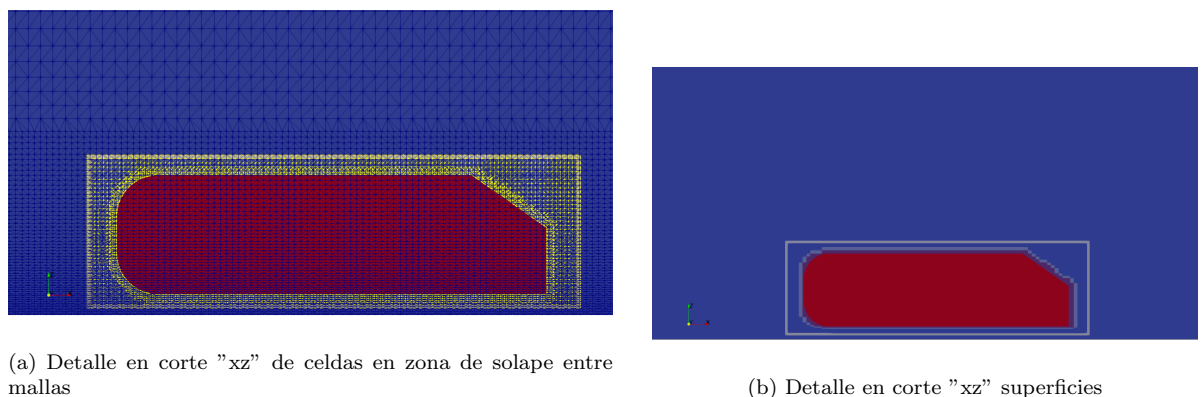


Figura 48: Celdas utilizadas en la plantilla OM y densificación de malla en capa límite



(a) Detalle en corte "xz" de celdas en zona de solape entre mallas

(b) Detalle en corte "xz" superficies

Figura 49: Zona de solape entre mallas

a la disminución del gradiente de tamaño entre éstas. Es por ello que se debe aplicar aquí especial cuidado en la etapa de mallado para disminuir errores de interpolación. La distancia entre el suelo del túnel y el piso del cuerpo de Ahmed debe estar cubierta por al menos 5 celdas con el menor gradiente de tamaño entre ellas (Guerrero, 2019).

En la malla de fondo celdas blancas de interpolación encierran al cuerpo de Ahmed, y en la malla solapada éstas celdas encierran al `oversetPatch`, esto se representa en la Fig. (49). Están presente los tres colores que identifican las celdas de cálculo, interpolación e inactivas. Durante el movimiento impuesto a la zona 1, esta plantilla va actualizándose, variando ubicación y cantidad de cada tipo.

8. Configuración y Resultados

8.1. Cilindro 2D

El análisis de flujo cruzado sobre el cilindro 2D constituye un patrón utilizado para testear el método OM, como fue explicado en el Capítulo (7.1). En el segundo caso de análisis se impone un movimiento prescrito en la dirección “x” a la zona 1, mediante la función Ec. (61), con $\omega = 2\pi f$, a través de la librería `dynamicMeshDict`, desde el instante inicial de cálculo.

$$x = K \sin(\omega t) \quad (61)$$

donde K representa la amplitud y ω la frecuencia de oscilación.

En la Tab. (1) se especifican las condiciones de borde impuestas al dominio computacional, al igual que las condiciones iniciales. Se define a U como velocidad de entrada, p presión de salida. La simulación se realizó imponiendo condición de flujo laminar, evaluado en el tiempo, seleccionando un paso de tiempo inicial $\Delta t = 0,0025s$, definiendo un $Co_{max} = 0,75$, con una función de paso de tiempo ajustable. El tiempo total de simulación evaluado es de 200 segundos. A fin de realizar el contraste con el trabajo de Alletto (2022) se selecciona la misma función de interpolación `inverseDistance`, y se impone el mismo número de Reynolds.

Se analiza la serie temporal de los coeficientes aerodinámicos de arrastre C_d , Ec. (62a), y sustentación C_l , Ec. (62b), calculando valores medios y máximos. Además se calcula y compara la frecuencia de desprendimiento de vórtices con la bibliografía de referencia mediante el número de Strouhal para el caso del cilindro fijo y móvil Ec. (62c).

$$C_d = \frac{2F_x}{\rho U^2 A} \quad (62a)$$

$$C_l = \frac{2F_z}{\rho U^2 A} \quad (62b)$$

$$S_t = \frac{f_s D}{U} \quad (62c)$$

Con F_x fuerza sobre el eje “x” y F_z fuerza sobre el eje “z”, $\rho = 1kg/m^3$ densidad unitaria, U velocidad de referencia, coincidente con la impuesta en la condición de borde `BC_inlet`, y el área $A = LD$, donde $L = 1m$ es el espesor de malla en la dirección “y”, $D = 1m$ el diámetro unitario del cilindro y f_s frecuencia de desprendimiento de vórtices en la estela, la cual se calcula a partir de la frecuencia característica de la serie temporal del coeficiente de sustentación C_l .

La forma geométrica con la que se discretizó la malla solapada Fig. (26) (cuadrada) difiere de la propuesta por Alletto (2022) (circular), con el objetivo de probar otra estrategia de mallado y verificar la influencia de la metodología OM sobre los resultados ante la igualdad en la definición de condiciones de borde y de flujo incidente al definir un mismo valor de Reynolds $Re = 100$.

BC_inlet	$1m/s(U)$
BC_outlet	$0Pa$
BC_back & front	empty
BC_cyl	noSlip
BC_topandbottom	ZeroGradient
Amplitude & frequency translation	$(-0,140\ 0,0\ 0,0)m - 1,04rad\ s^{-1}$
Reynolds	100

Tabla 1: Condiciones de borde e iniciales

8.1.1. Análisis de resultados

Los valores medios del coeficiente de arrastre $C_{d,medio}$ y máximos del coeficiente de sustentación $C_{l,max}$ de las series temporales, se presentan en la Tab. (2).

Trabajo	ω	$C_{d,medio}$	$C_{l,max}$	S_t
Alletto	0	1,33	0,33	0,164
Adotti	0	1,34	0,29	0,163
Alletto	1,04	1,37	0,46	0,159
Adotti	1,04	1,38	0,41	0,159

Tabla 2: Comparación del coeficiente de arrastre medio $C_{d,medio}$ y el máximo coeficiente de sustentación $C_{l,max}$, para diferentes frecuencias de oscilación ω .

Las discrepancias en los resultados presentados pueden estar asociadas a las diferentes estrategias de mallado y discretización en las cercanías al cilindro. La cantidad de celdas utilizadas en este trabajo es de 34274, respecto de las 17520 contabilizadas en la referencia, además el tamaño de la primera celda es del orden del 40 % más pequeña. También pueden ser causadas por el esquema de discretización temporal empleado en este trabajo “Euler”, diferente al utilizado por la referencia “Backward”.

En la Fig. (50) se presentan las series temporales de los coeficientes aerodinámicos para el caso de excitación nula. La metodología OM y el mallado propuesto captura correctamente el inicio de los vórtices de von Kármán, observables en la Fig. (50b) a partir de los 50 segundos aproximadamente, logrando un comportamiento estacionario luego de los 100 segundos. Evaluando el régimen estacionario se observa una diferencia menor al 1 % en los valores medios de coeficientes aerodinámicos de arrastre $C_{d,medio}$, Fig. (50a), sin embargo existe una diferencia en la amplitud del C_l de alrededor del 12 %, como se observa en la Fig. (50b).

En la Fig. (51) se presentan las series temporales de los coeficientes aerodinámicos para el caso con frecuencia de excitación no nula, en éstas los valores medios para ambos coeficientes, son coincidentes. En la Fig. (51a) se evidencia una amplitud menor del C_d en la zona estacionaria, lo cual se observa con mayor detalle en la ampliación inserta. En la Fig. (51b) se observa un corrimiento en la amplitud del C_l y un 11 % de diferencia en valores máximos, también el coeficiente oscila con similar período pero en contra fase.

Las frecuencias características del desprendimiento de vórtices en ambos casos son coincidentes como se observa en la Fig. (52), aún para el caso excitado con movimiento lineal oscilante en el eje “x” Fig. (52b). La imposición cinemática afecta levemente la frecuencia natural de desprendimiento de vórtices característica del caso no excitado, ver Fig. (52a).

En la Fig. (53) se presentan los campos de velocidad para las mallas solapadas de la literatura citada y este trabajo con una resolución en la renderización de 30 niveles, ambos casos con excitación no nula. La estrategia de mallado en el presente trabajo opta por una configuración cuadrada de la zona 1 o malla solapada, buscando que la similitud entre las celdas de interpolación sea la mayor posible, como se puede observar en el sector derecho de la Fig. (53b). En cambio en la literatura de referencia, la interpolación entre mallas se realiza entre celdas de coordenadas rectangulares (malla de fondo) y radiales (malla solapada).

8.1.2. Coeficientes aerodinámicos

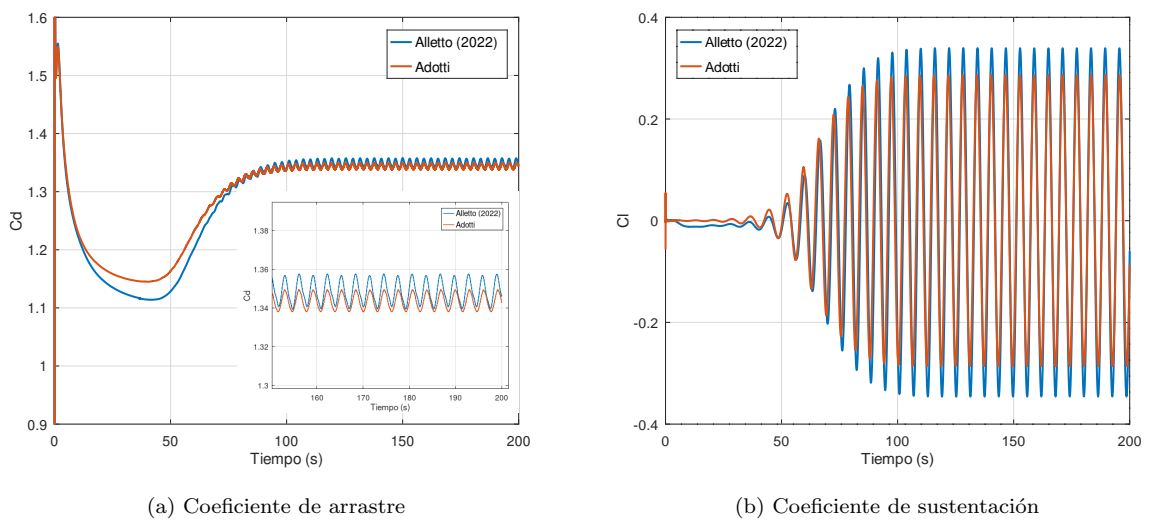


Figura 50: Serie temporal de coeficientes aerodinámicos para cilindro fijo

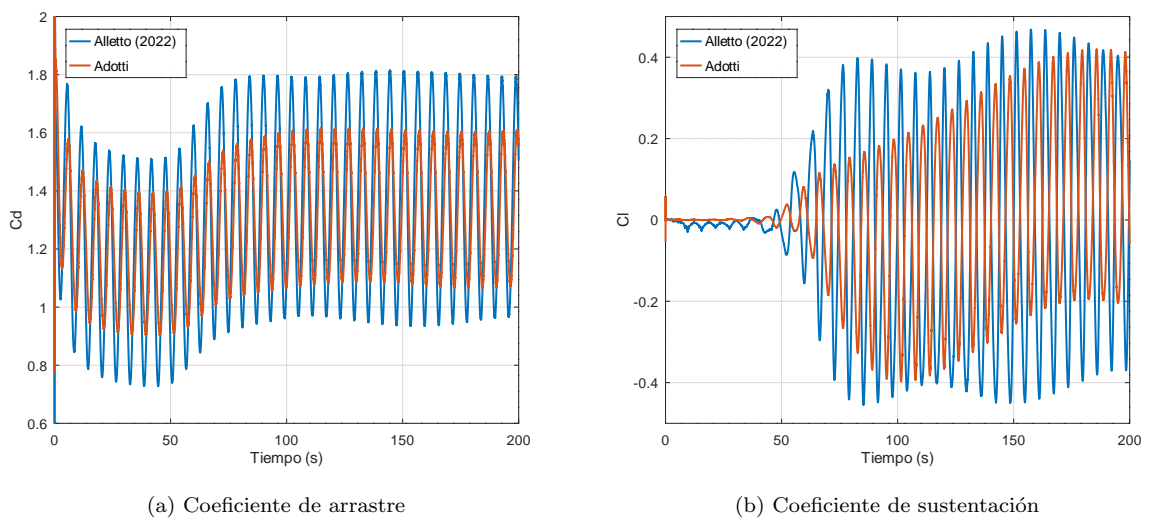


Figura 51: Serie temporal de coeficientes aerodinámicos para frecuencia de excitación $\omega = 1,04 \text{ rad/s}$

8.1.3. Frecuencia de vórtices - Número de Strouhal

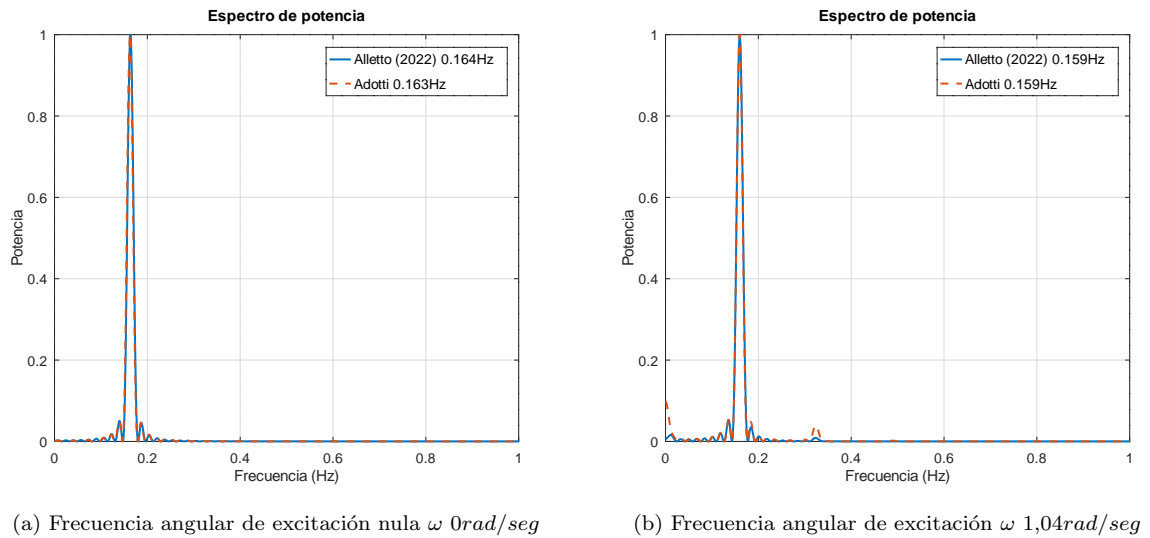


Figura 52: Espectros de potencia para ambos casos de excitación

8.1.4. Campos de velocidad en la zona de solape de mallas

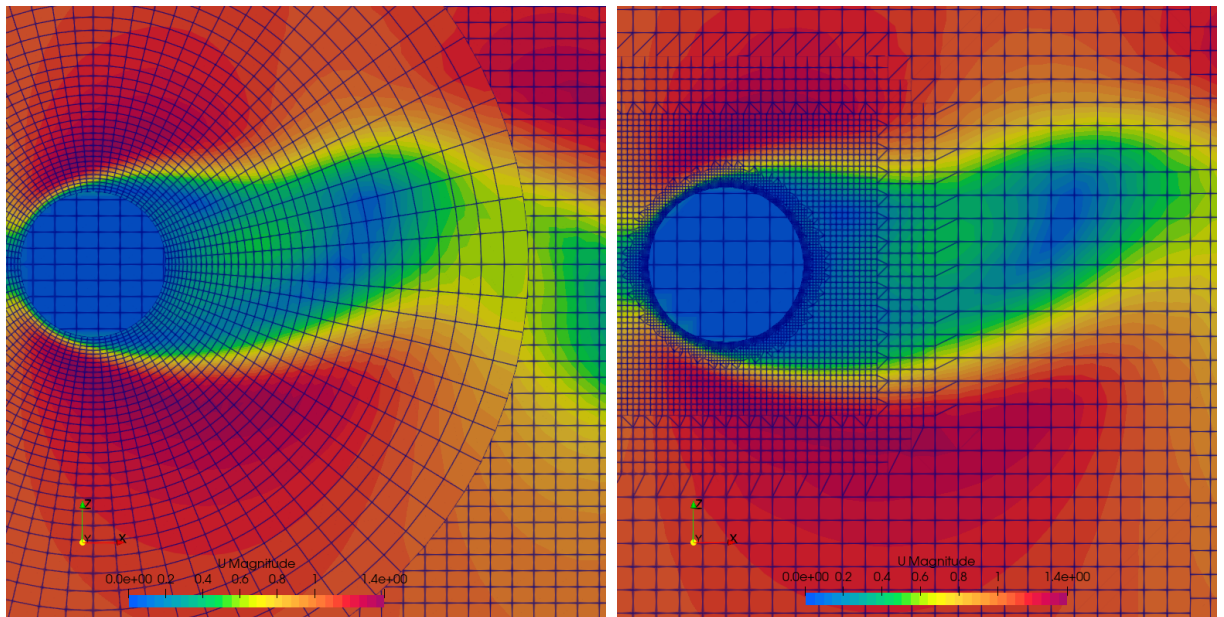
(a) Campo de velocidad entre mallas solapas, excitación $\omega = 1,04$ rad/seg Alletto (2022).(b) Campo de velocidad entre mallas solapas, excitación $\omega = 1,04$ rad/seg, este trabajo.

Figura 53: En las figuras a) y b) se presentan los campos de velocidad resultantes para ambas estrategias de mado Alletto (2022) (circular) y este trabajo (rectangular).

8.2. Perfil alar 2D

El perfil alar es también un modelo muy utilizado para testear el método OM. En él se impone un movimiento de gran desplazamiento a la zona 1 mediante el uso de la librería `dynamicMeshDict`. Ésta permite detallar desplazamientos en el eje “y” ángulos barridos en el eje “z” que describirá el perfil al estar adherido a esta zona en movimiento. Los movimientos son pre-calculados para cada paso de tiempo. Las funciones utilizadas para definir los movimientos son senoidales. La trayectoria pre-calculada es definida mediante un archivo de entrada `1`.

En la Tab. (3) se detallan las condiciones de borde impuestas al dominio en conjunto con sus condiciones iniciales, tanto para la malla de fondo como la solapada. U es la velocidad de entrada, k energía cinética turbulenta, ω tasa de disipación turbulenta. La simulación se realizó con la técnica uRANS, escogiendo el modelo de turbulencia kw_{SST} , adoptando un paso de tiempo inicial de $\Delta t = 1 \times 10^{-5}s$, permitiendo un $Co_{max} = 4$ con una función de paso de tiempo ajustable. Se evaluó un marco temporal de 20 segundos.

En primera instancia se analizan los coeficientes aerodinámicos y campos de velocidad, obtenidos con las distintas funciones de interpolación disponibles en `OpenFOAM`. Una vez determinada la función óptima, se realiza un análisis del rendimiento del solver y se discuten estrategias para el particionado de dominio.

BC_inlet	$1m/s(U) - 0,01m^2s^{-2}(k) - 15,0 s^{-1}(\omega)$
BC_outlet	$0Pa - 0,01m^2s^{-2}(k) - 15,0 s^{-1}(\omega)$
BC_back & front	empty
BC_wing	movingWallVelocity
BC_topandbottom	ZeroGradient
Amplitude & frequency translation	$(0,0 \pm 0,5 0,0)m - 0,628rad s^{-1}$
Amplitude & frequency rotation	$(0 0 30)degree - 0,628rad s^{-1}$
Reynolds	1000

Tabla 3: Condiciones de borde e iniciales

8.2.1. Análisis de funciones de interpolación

Se ha estudiado el comportamiento de las funciones de interpolación listadas en el Cap. (5), evaluando dos casos, uno donde la zona 1 permanece estática con el perfil alar a cero grados de ángulo de ataque y en el otro caso se impone un movimiento prescrito a la zona 1. Se realizan para ambos un contraste de los coeficientes aerodinámicos obtenidos mediante las distintas funciones de interpolación, y el trabajo de Kurtulus (2015), presentados en las Fig. (54) y Fig. (55).

Complementariamente se presentan en la Tab. (4), el tiempo total insumido para cada corrida realizada.

¹6DoF.dat

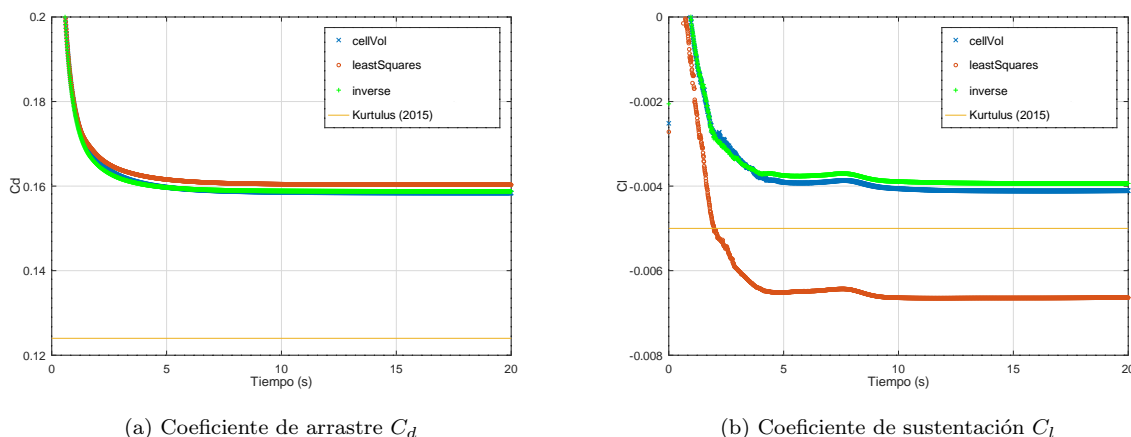


Figura 54: Caso estático. Contraste de coeficientes aerodinámicos según diferentes métodos de interpolación y referencia bibliográfica.

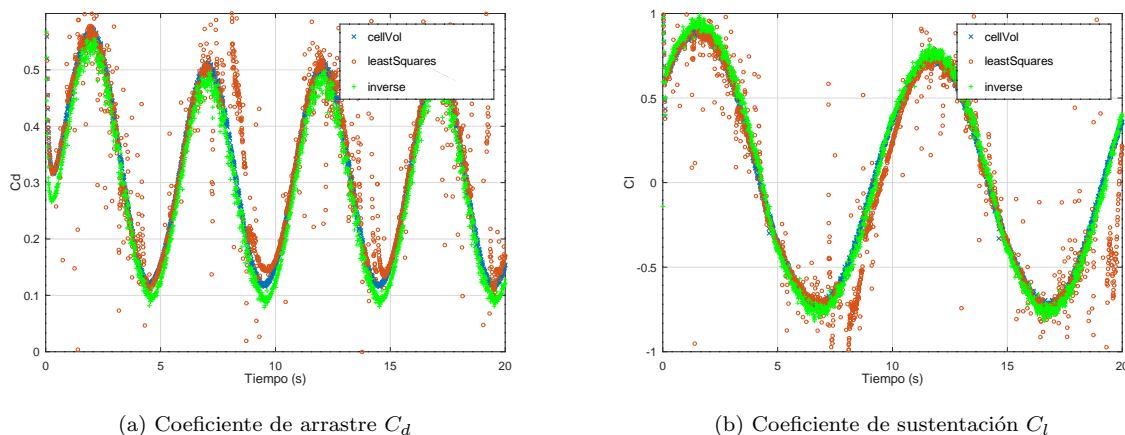


Figura 55: Caso móvil. Contraste de coeficientes aerodinámicos según diferentes métodos de interpolación.

Tiempo (s)	leastSquares	inverseDistance	cellVolumeWeight
Estático	878	922	1118
Móvil	1534	1696	1678

Tabla 4: Tiempos insumidos en las pruebas para los distintos casos y funciones de interpolación.

Los resultados obtenidos permiten evaluar el comportamiento de las funciones de interpolación para la condición estática y móvil de la zona 1. La Fig. (54) muestra la serie temporal de los coeficientes aerodinámicos para la condición estática. En la Fig. (54a), se observa una convergencia del resultado para las tres funciones de interpolación. En la Fig. (54b) es evidente un comportamiento y resultado similar de las funciones de interpolación `inverseDistance` y `cellVolumeWeight`, pero la función `leastSquares` difiere del valor de convergencia de las anteriores. La malla utilizada en este caso de validación para el OM, no posee discretizada la capa límite ni zonas de refinamiento, por lo cual puede ser un posible factor que explique la diferencia entre los resultados de la literatura citada y los coeficientes aerodinámicos obtenidos en este trabajo.

El análisis de tiempo insumido para el caso estático se puede ver en la Tab. (4), la función `leastSquare` resulta ser la más rápida, le sigue la función `inverseDistance` y

última `cellVolumeWeight`. Verma and Hemmati (2020) en su trabajo encuentra que la función `inverseDistance` es la de menor consumo, y al igual que en el presente trabajo la función `cellVolumeWeight` la de mayor consumo.

La Fig. (55) muestra la serie temporal de los coeficientes aerodinámicos para la condición móvil. En la Fig. (55a) se evidencia la respuesta del coeficiente C_d al movimiento senoidal impuesto de traslación sobre el eje “y” en conjunto con la rotación sobre el eje “z”. Las funciones `inverseDistance` y `cellVolumeWeight` no presentan discrepancia significativa a lo largo del tiempo simulado, en cambio la función `leastSquare` copia en cierta forma los resultados obtenidos por las otras funciones, pero con alta dispersión inclusive algunos valores por fuera del rango utilizado en el eje de abscisas, que no fueron representados. En la Fig. (55b) el coeficiente C_l presenta discrepancia de resultados entre la función `leastSquares` y las restantes. Esta dispersión de resultados para la función `leastSquares` se puede deber por efectos de la interpolación entre las mallas en la región de la estela donde existen fuertes gradientes de velocidad. Cada malla o zona posee una solución del campo de velocidad, los cuales se deben unir por medio de las celdas de interpolación. En este proceso pueden existir diferencias de velocidades notables en las celdas donantes de ambas zonas. Es útil recordar como se expuso en el Cap. (5) que a partir de éstas se calculan los coeficientes solución de la función de interpolación, los cuales responden a un orden cuadrático, magnificando aún más los posibles saltos o discontinuidades en la unión de las soluciones, respecto de los coeficientes basados en pesos lineales de las otras funciones de interpolación.

Luego de realizar la evaluación de las funciones en condiciones estáticas y móviles, se opta por utilizar la función `inverseDistance` para el desarrollo de los ejemplos posteriores. Esta resulta ser la opción más versátil y recomendada por la bibliografía especializada, y como se observa en el análisis presentado anteriormente no posee problemas de dispersión excesiva como la función `leastSquares`, y su desempeño en condición móvil es un 1% más lenta que la función `cellVolumeWeight`. Un factor más que influye en la selección de la función `inverseDistance`, es que el proceso de cálculo de pesos en mallas 3D para la función `cellVolumeWeight` consumirá mayores recursos que en este caso 2D.

8.2.2. Métricas del solver

La estrategia de testeo adoptada consiste en realizar una modificación al solver denominado `overPimpleDyMFoam` de tal forma que realice un perfil del tiempo insumido en cada una de las tareas especificadas en Fig. (4) y Fig. (5), sin que la medición interfiera con el funcionamiento general del mismo. La modificación implementada se denomina `overPimpleDyMFoamTimed` y es presentada en el Anexo (11).

La metodología OM produce normalmente un elevado consumo computacional, no solo por la cantidad de celdas con la que se discretizan los cuerpos móviles, sino por las ecuaciones que deben ser resueltas para cada paso de tiempo e interpolaciones de resultados entre celdas de mallas de fondo y solapada, requiriendo para cualquier aplicación medianamente realista un elevado tiempo de cálculo aún aplicando metodologías de cálculo paralelo. Es importante tener en cuenta que uno de los principales desafíos en el cómputo de altas prestaciones es conseguir aplicaciones paralelas eficientes, las cuales sean capaces de aprovechar de la mejor manera posible los recursos que ofrecen los sistemas. Por ello es de interés en conocer las métricas e índices que indiquen si la aplicación está haciendo un uso eficiente del sistema. Entre ellas se pueden citar el tiempo de ejecución de la

aplicación, el speed up, la eficiencia, escalamiento fuerte (Panadero, 2015), las ecuaciones son detalladas en el Anexo (11).

Así se evalúan las métricas antes mencionadas para el caso móvil, en corridas con distinto número de procesadores y en distintos equipos de cálculo paralelo a fines de conocer la escalabilidad del solver empleado. Para la evaluación se han utilizado equipos del Sistema Nacional de Cálculo Intensivo, pertenecientes al IMIT CONICET-UNNE y el Centro de Cómputos (C3) del CIMEC CONICET-UNL.

cells	N(procs)	time(sec)	Strong Scaling
27024	1	1653	1,000
27024	2	2298	0,719
27024	4	1155	1,432
27024	16	620	2,666

Tabla 5: Métricas clúster IMIT

cells	N (procs)	Total Time (s)	Mesh Update Time (%)	PIMPLE Time (%)	U_Eq (%)	P_Eq (%)	Turb Mod (%)
27024	2	2298	33,895	61,819	4,981	52	4,451
27024	4	1155	41,618	54,679	3,8	47,215	3,628
27024	16	620	39,184	56,745	3,629	49,011	4,043

Tabla 6: Perfil de procesos overPimpleDyMFoamTimed clúster IMIT

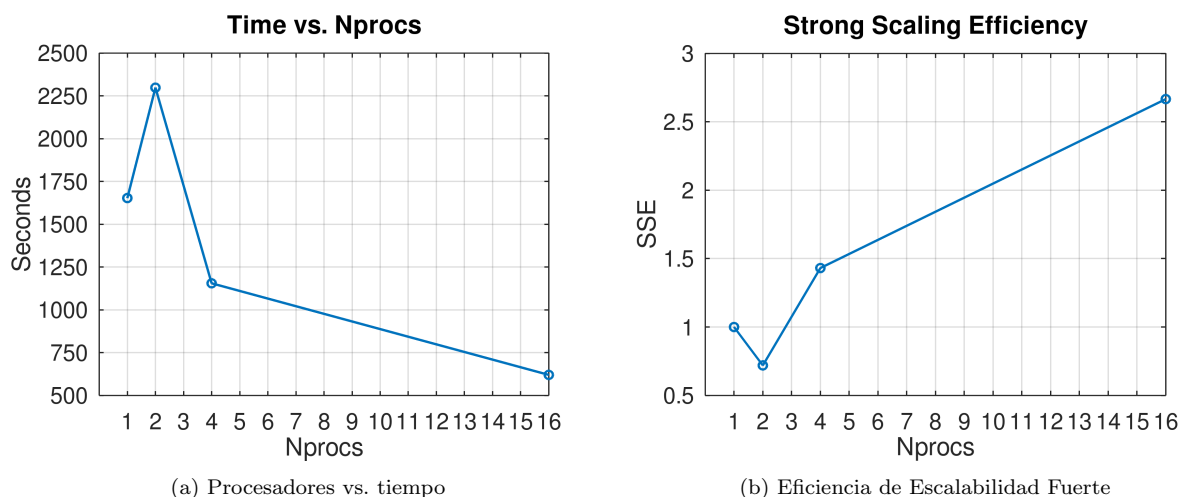


Figura 56: Métricas de escalabilidad clúster IMIT

Tanto en la Tab. (5) como en la Tab. (7) se presentan las métricas de los cálculos llevados en los diferentes clusters. En ambas se observa que la implementación en paralelo del solver overPimpleDyMFoam mejora con el aumento de número de procesadores. En particular solo se pudo evaluar en el cluster Pirayú - CIMEC, el comportamiento del caso con 2 nodos y 32 procesadores donde se evidencia una caída de la escalabilidad, el incremento de tiempo se asocia al proceso adicional de comunicación entre los nodos, marcando un límite o punto óptimo de la descomposición del dominio computacional, hecho que se observa en Tab. (7) y Fig. (??).

En todos los casos analizados se ha utilizado el método de descomposición de dominio **simple** en configuraciones (2 1 1), (2 2 1), (4 4 1) y (8, 4, 1) asumiendo direcciones "xyz". El método se encarga de balancear la cantidad de celdas entre los procesadores de una forma óptima. La selección del método de descomposición presenta mejores resultados que sus opcionales dentro de **OpenFOAM** (Windt et al., 2018).

cells	N(procs)	time(sec)	Strong Scaling
27024	1	1653	1,000
27024	2	1205	1,372
27024	4	853	1,938
27024	16	539	3,067
27024	32	1217	1,358

Tabla 7: Métricas clúster Pirayú - CIMEC

cells	N(procs)	Total Time (s)	Mesh Update Time (%)	PIMPLE Time (%)	U_Eq (%)	P_Eq (%)	Turb Mod (%)
27024	2	1205	35,165	60,848	4,756	51,56	4,507
27024	4	853	45,120	51,328	3,68	43,922	3,708
27024	16	539	51,068	44,505	2,755	38,592	3,127
27024	32	1217	21,16	76,892	1,442	73,726	1,699

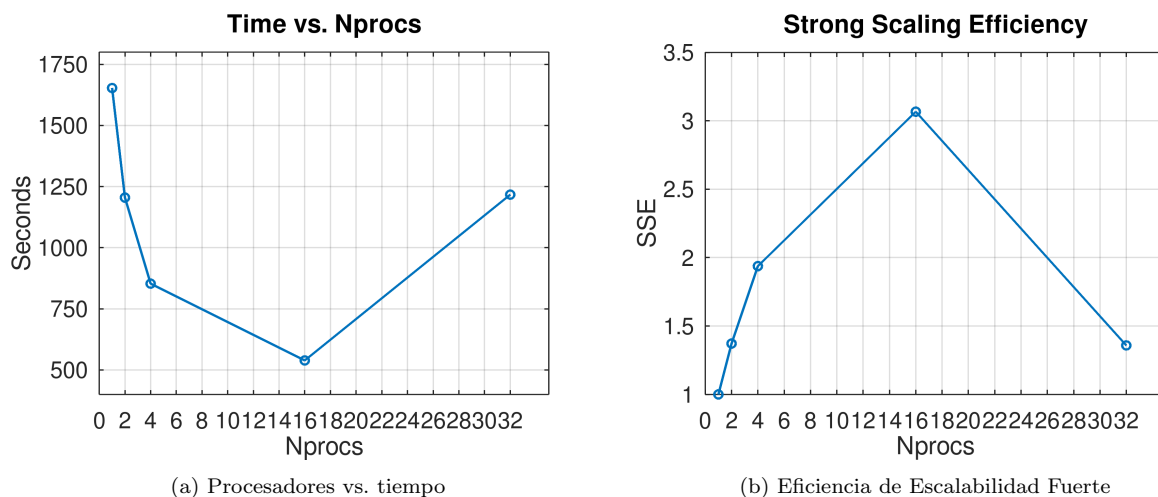
Tabla 8: Perfil de procesos `overPimpleDyMFoamTimed` cluster Pirayú - CIMEC

Figura 57: Métricas de escalabilidad clúster Pirayú - CIMEC

En este análisis se ha observado una sensibilidad en la eficiencia del cálculo paralelo, en función de la distribución asignada en la descomposición del dominio y la división entre procesadores del `oversetPatch`. En los resultados presentados en las Tab. (6) y Tab. (8) es evidente que el proceso de actualización de malla consume una parte importante del tiempo total, en pruebas adicionales dejando estático el perfil alar, es decir donde no es actualizada la malla se ha reportado una mejora del 34%. Por ello se opta además por realizar un análisis de tiempos a una metodología de descomposición de dominio manual. El objetivo de esta es asegurar que la región donde se desplazará la zona 1 quede contenida dentro de un mismo procesador. Así se logra evitar que el `oversetPatch` sea compartido

por procesadores. Reduciendo así los tiempos requeridos en el proceso de actualización de malla al no existir transferencia de datos entre distintos procesadores.

En la Fig. (58) se presentan dos estrategias de distribución de dominio una simple (4 1 1) donde el algoritmo balancea la carga entre los procesadores en función a la división requerida y otra llevada a cabo en forma manual. En esta última se escogen los límites de distribución de celdas para cada procesador también (4 1 1), pero con el `oversetPatch` contenido dentro de un procesador durante en todo su historial de movimiento.

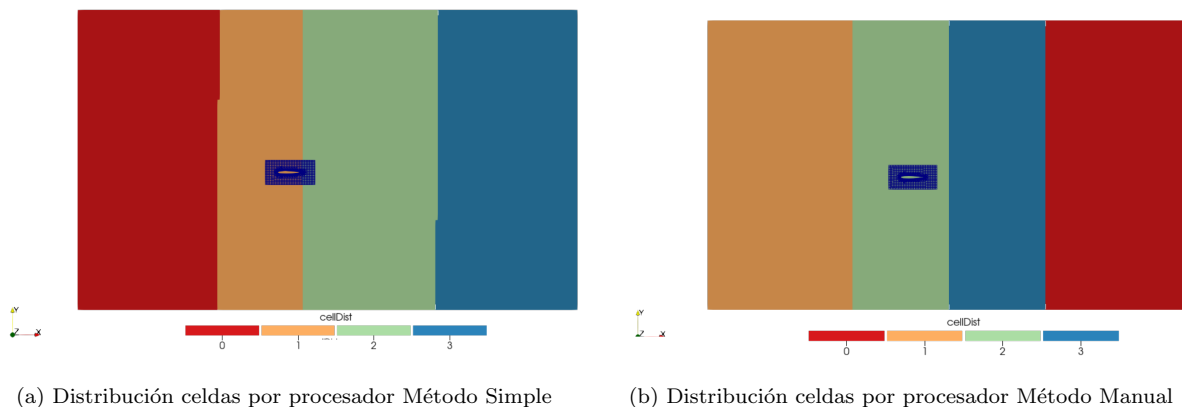


Figura 58: Estrategias distribución de celdas

La estrategia planteada mejora un 28,7% para el caso analizado en una Desktop, y de un 15% en el clúster IMIT. El desbalance máximo de distribución de celdas por procesador al aplicar la metodología manual es del 15,8% respecto del valor medio. Una posterior prueba con 16 procesadores repitiendo estrategia de asignar un solo procesador a la zona 1, ya no resulta efectiva al ser un 60% más lenta respecto del método `simple`, la pérdida viene asociada al extremo desbalance entre los procesadores igual a 292,18% respecto del valor medio.

Por lo tanto se concluye que la estrategia manual de descomposición de dominio resulta útil solo en casos donde el desbalance de celdas por procesador no sea excesivo. Hay reportes que destacan esta mejora al evitar la división del `oversetPatch` entre procesadores, pero aún así el cálculo paralelo parece introducir errores (Windt et al., 2018).

Un problema crucial de la metodología OM es la conservación de flujo, el cual se ve afectado por la introducción de errores producidos por la función de interpolación utilizada para transferir resultados entre las mallas de fondo y solapada. En esta simulación se evaluó el flujo total ϕ en las condiciones de borde `inlet` y `outlet` para cada paso de tiempo y cada corrida, obteniendo diferencias en promedio menores al 0,35%, con lo cual se prueba así de forma general la conservación.

8.2.3. Campos de velocidad

Los campos son presentados en la Fig. (59), para los distintos pasos de tiempo. En particular, en la Fig. (59d), se observa una diferencia en la continuidad de los contornos de velocidad en la zona de interpolación entre la malla de fondo y solapada especificada en la Fig. (34). El error resulta evidente pero no es del todo grosero, para ser una malla simple utilizada para realizar evaluaciones rápidas del método.

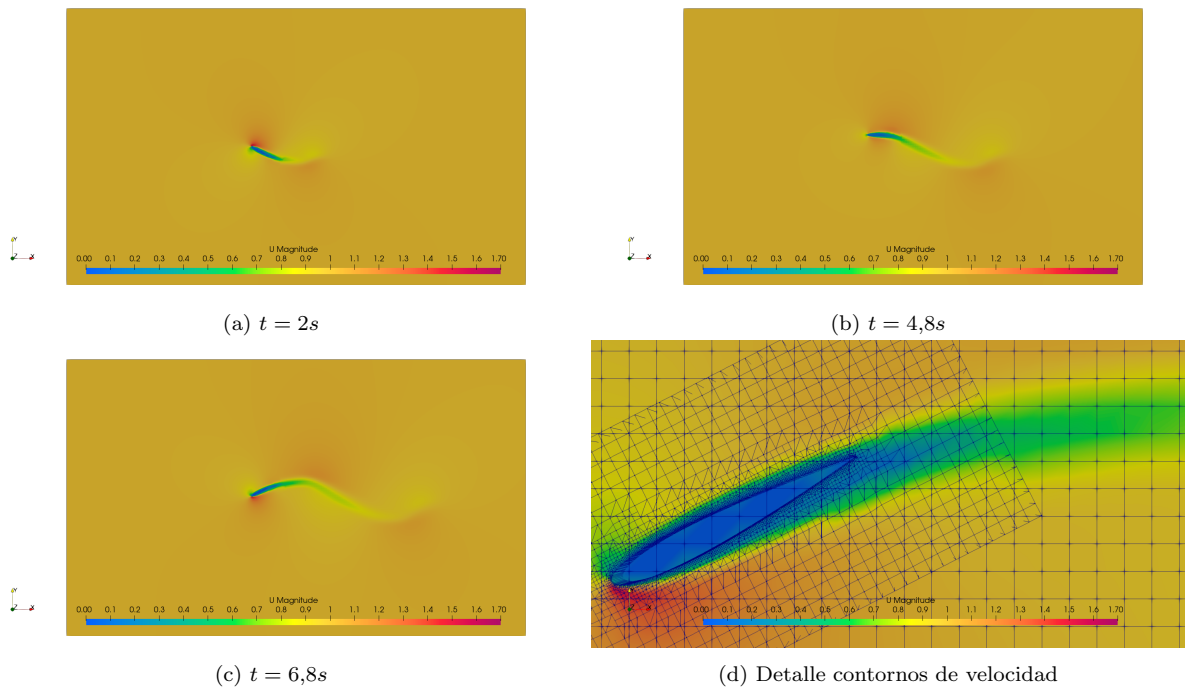


Figura 59: Campos de velocidad para distintos pasos de tiempo

8.2.4. Análisis de resultados

En las secciones anteriores se detalló y analizó el comportamiento del método OM para el caso evaluado en cuestión, mediante un caso sencillo específicamente utilizado para testeo. En el análisis de las funciones de interpolación se logró escoger una función candidata `inverseDistance` y someterla a prueba en dos sistemas de cálculo paralelo, concluyendo que el número máximo de procesadores recomendable es aquel que asegure no realizar la comunicación entre nodos.

8.3. BARC

El BARC es otro modelo 2D utilizado en este trabajo para evaluar el comportamiento del OM, a diferencia del perfil alar 2D es un cuerpo romo inmerso en un flujo netamente turbulento. En este modelo se ha impuesto un movimiento prescrito oscilatorio por medio de la librería `dynamicMeshDict`, haciendo variar el ángulo de ataque acorde a una función senoidal con una velocidad angular y amplitud definida.

En la Tab. (9) se detallan las condiciones de borde impuestas al dominio en conjunto con sus condiciones iniciales, tanto para la malla de fondo como la solapada. U es la velocidad de entrada, k energía cinética turbulenta, ω tasa de disipación turbulenta. La simulación fue llevada a cabo mediante la técnica uRANS con el modelo kw_{SST} (Menter et al., 2003), con un paso de tiempo inicial de $\Delta t = 2,5 \times 10^{-5}s$, permitiendo un $Co_{max} = 2$ con una función de paso de tiempo ajustable. Se utiliza una media móvil de 50 para el análisis estadístico de los coeficientes. Se utiliza la función de interpolación `inverseDistance`, acorde a los fundamentos presentados en la sección 8.2.1

BC_inlet	$10m/s(U) - 0,00375m^2s^{-2}(k) - 0,153 s^{-1}(\omega)$
BC_outlet	$0Pa$
BC.back & front	empty
BC_BARC	noSlip
BC.roof & floor	symmetry
Amplitude (α) & frequency rotation (ω)	$(0\ 0\ 6) - (-)0,25rad\ s^{-1}$
Reynolds	53600

Tabla 9: Condiciones de borde e iniciales

8.3.1. Coeficientes aerodinámicos

La simulación fue realizada en dos etapas, una primera en el intervalo temporal 0 a 2,5 segundos manteniendo el ángulo de ataque fijo a 0° . Las estadísticas de los coeficientes aerodinámicos se calculan para los últimos 0,5 segundos, asegurando que el flujo turbulento se haya desarrollado y se hayan formado los vórtices aguas abajo del cuerpo estudiado. Los resultados obtenidos en este trabajo son contrastados con la publicación de Wu et al. (2020).

En la segunda etapa con movimiento impuesto representado en la Fig. (60), se evalúan los coeficientes aerodinámicos desde el principio en un rango de tiempo de 6,1 segundos. Las estadísticas de los coeficientes obtenidos son contrastados con la misma bibliografía de referencia, ver Tab. (10).

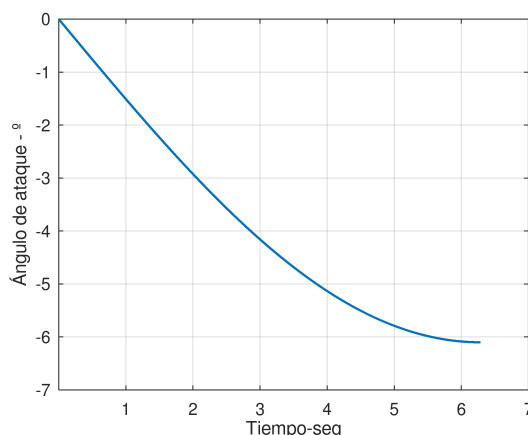


Figura 60: Función senoidal de movimiento prescripto

ángulo	Este trabajo		Bo Wu 3D LES	
	$\overline{C_d}$	$\overline{C_l}$	$\overline{C_d}$	$\overline{C_l}$
0°	0,981	-0,012	1,020	-0,006
1°	1,001	0,251	1,056	0,189
2°	1,048	0,552	1,110	0,352
3°	1,140	0,887	1,200	0,402
4°	1,286	1,316	1,306	0,411
5°	1,421	1,730	1,430	0,431
6°	1,545	2,106	1,554	0,457

Tabla 10: Resultados de coeficientes aerodinámicos

Existe una discrepancia en el C_l , entre la referencia bibliográfica y los resultados obtenidos en el presente trabajo. Se evidencia la sensibilidad del modelo ante cargas verticales, en condiciones de movimiento impuesto al mismo. Sin embargo ha de contemplarse que los coeficientes declarados en la referencia son con el cuerpo fijo, y los presentados aquí poseen condición móvil. Además ha de tenerse en cuenta la imposibilidad de modelar la transición de capa límite, por la versión del modelo de turbulencia utilizado $\kappa\omega_{SST}$. Sin embargo el cambio continuo de posición del modelo presenta una limitación aún para modelos RANS que contemplen tal efecto, los cuales no podrían replicar la transición real por efectos de la actualización continua en la longitud de desarrollo de la capa límite frente al flujo medio incidente.

La historia temporal de los coeficientes aerodinámicos son presentados en la Fig. (61), en ella se observa el crecimiento tanto del C_d como del C_l acorde al aumento del ángulo de ataque. En línea de punto negra se indica la sucesión de ángulos de ataque expuestos en la Tab. (10) de menor a mayor.

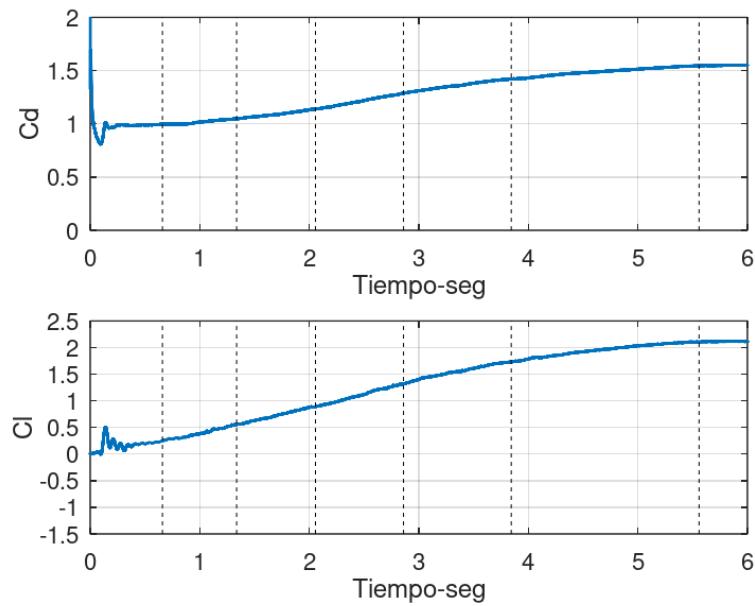
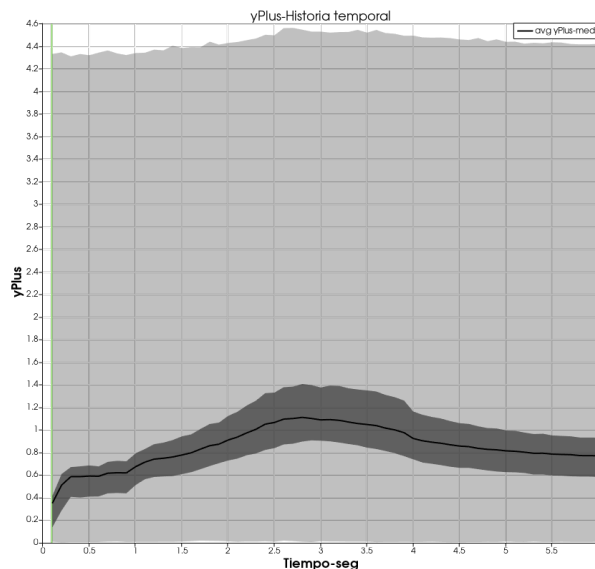


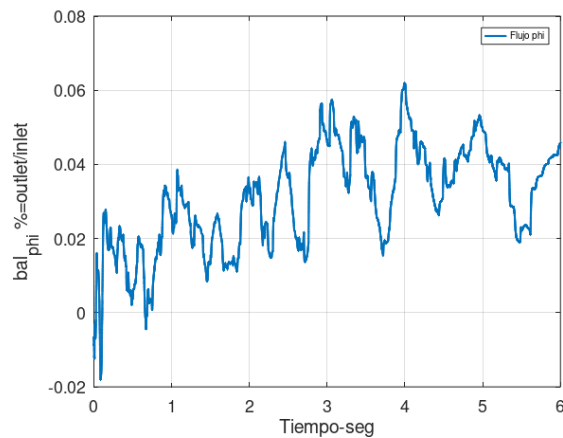
Figura 61: Historia temporal de coeficientes aerodinámicos

8.3.2. y^+ Plus - Balance de flujo

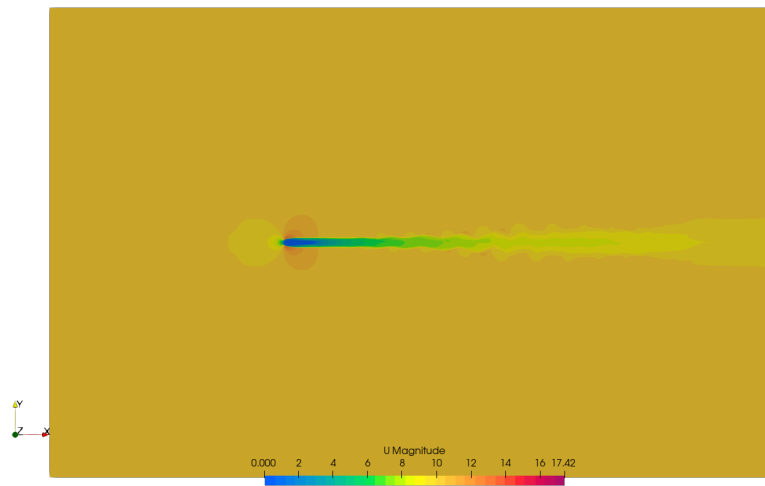
La simulación se ha llevado a cabo monitoreando temporalmente el coeficiente y^+ sobre la superficie del BARC. En la Fig. (62) se presentan los resultados, evidenciando la correcta discretización de la capa límite al poseer valores inferiores a 5 durante toda la corrida.

Figura 62: Historia temporal del coeficiente y^+ de las primeras celdas de la capa límite sobre el BARC.

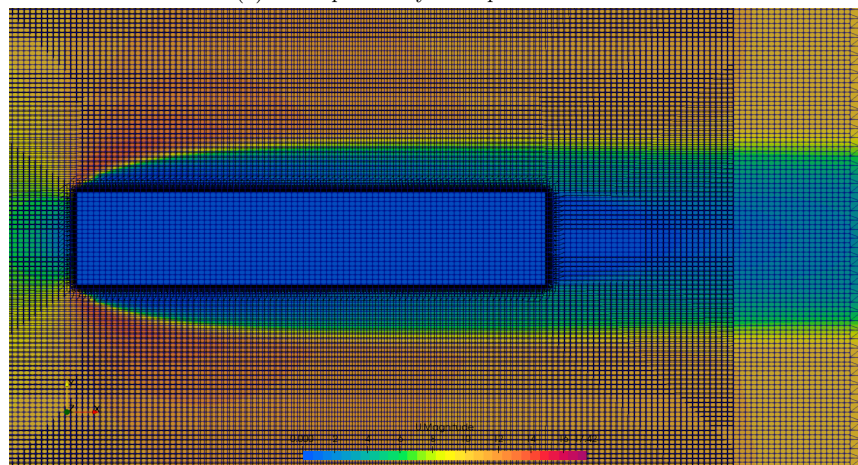
El balance de flujo total ϕ entre las condiciones de borde *inlet* y *outlet* para cada paso de tiempo, permiten evaluar la conservación de masa en forma general de la simulación llevada a cabo con la implementación OM, el error medio igual a 0,03 %, ver Fig. (63).

Figura 63: Historia temporal del balance de flujo ϕ

Se presenta en la Fig. (64) el campo de velocidad en la condición de ángulo de ataque fijo. En la Fig. (64a), se observa la formación de los vórtices de von Kàrmàn aguas abajo del cuerpo. No se aprecia un gradiente en la continuidad de los contornos del campo de velocidad en la región de solape, ver Fig. (64b).



(a) Vista plano "xy" campo de velocidad



(b) Detalle campo de velocidad en la región de solape

Figura 64: Campos de velocidad para $t = 2,5$ seg

En la Fig. (66) se presenta el campo de velocidad para la condición de movimiento impuesto, barriendo la sucesión de ángulos de ataque $\alpha = \{1, \dots, 6\}$, cada uno correspondiente a un instante de tiempo. Se aprecia la formación de vórtices aguas abajo del modelo, hasta la Fig. (66b). Luego por aumento del α se abandona la simetría respecto del flujo incidente y los vórtices de von Kàrmàn desaparecen formando una estela turbulenta que aumenta su espesor en forma proporcional con el ángulo de ataque.

La concordancia entre los contornos de velocidad en la región de solape definida por el `oversetPatch` es precisa. No se evidencian fuertes gradientes entre los campos de velocidad de las mallas, ver Fig. (65). Con lo cual ante los resultados obtenidos, la implementación de la malla, configuración y condiciones de borde parecen no generar problemas en la continuidad de la solución en forma global.

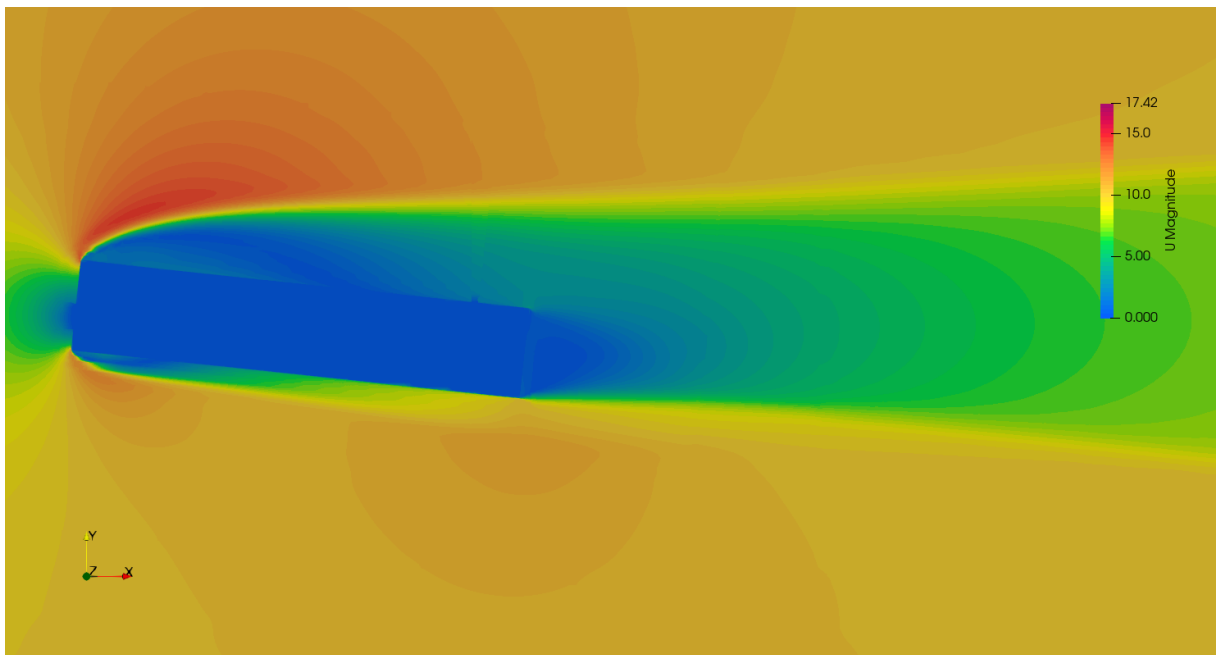


Figura 65: Campos de velocidad en la región de solape

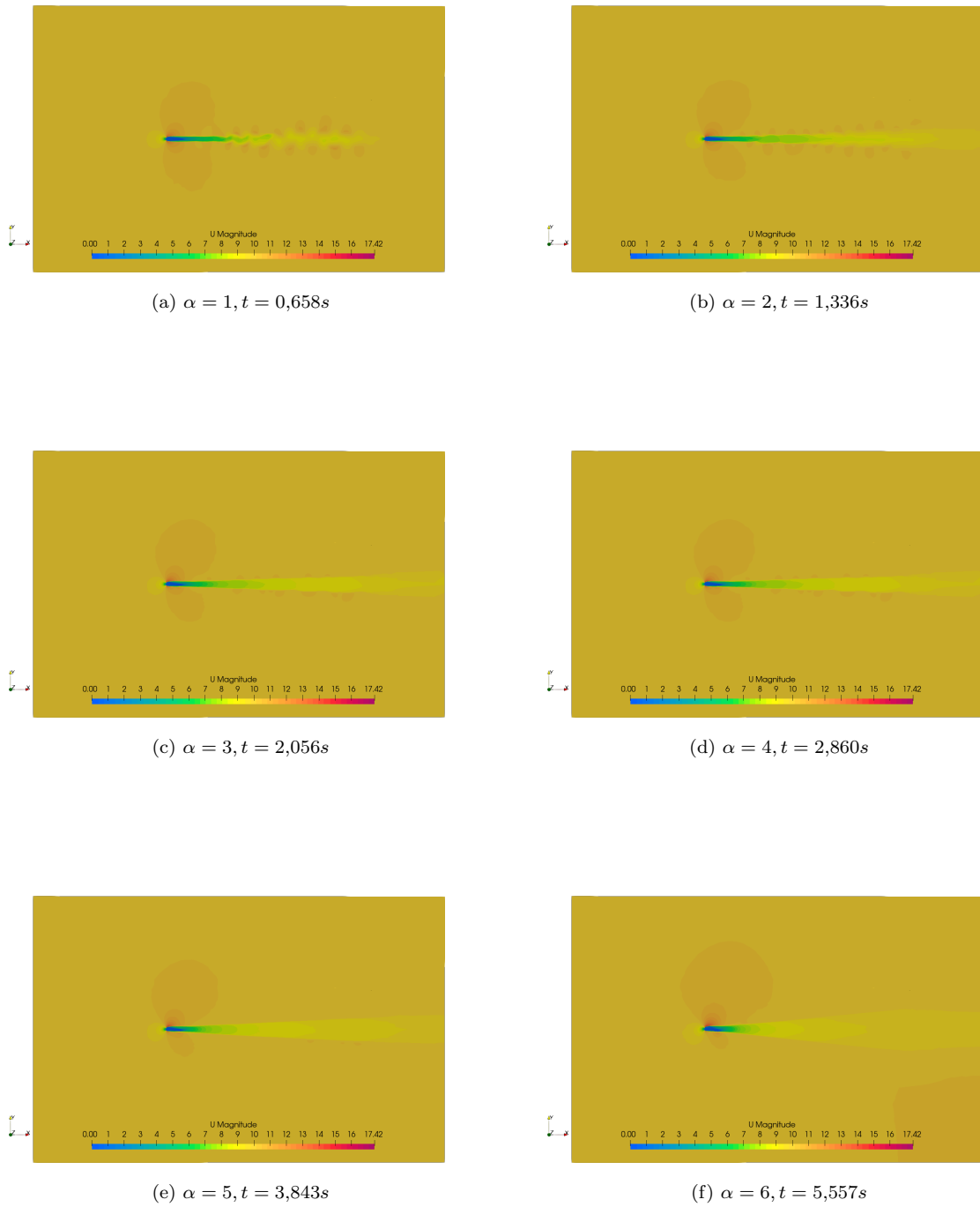


Figura 66: Campo de velocidades para los distintos ángulos de ataques α .

8.3.3. Métricas del solver

Durante la simulación se realizó la medición de los tiempos inferidos por el solver `overPimpleDyMFoamTimed` antes descrito. El tiempo total insumido 749566 segundos $\approx 8,7$ días. El dominio computacional ha sido descompuesto en 12 procesadores con la metodología "simple" con una distribución en xyz (3, 4, 1), ver Fig. (67). La zona 1 o región móvil que contiene el `oversetPatch` en su máximo ángulo comparte 8 procesadores. Luego de algunas pruebas se concluyó que esta disposición resulta ser la más eficiente. A pesar de poseer varios procesadores intercomunicando resultados en la zona móvil, lo cual como ya se especificó anteriormente no resulta óptimo, sí es la configuración que menos tiempo insume y menor desbalance entre procesadores produce evitando aumentar los tiempos de sincronización.

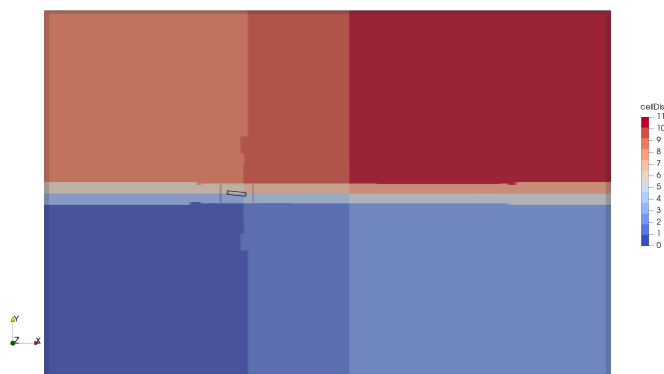


Figura 67: Distribución de celdas por procesador.

Las métricas obtenidas al analizar el solver, evidencian un cuello de botella generado por la tarea de actualización de malla inducida por el movimiento del cuerpo. Ésta se encarga de modificar las matrices en cada paso de tiempo, al renovar la plantilla de interpolación. Este proceso consiste en la re-identificación de celdas cálculo, interpolación y huecas en todo el dominio computacional. Por otra parte los residuos finales de velocidad y presión poseen un comportamiento estable durante toda la corrida, ver Fig. (68).

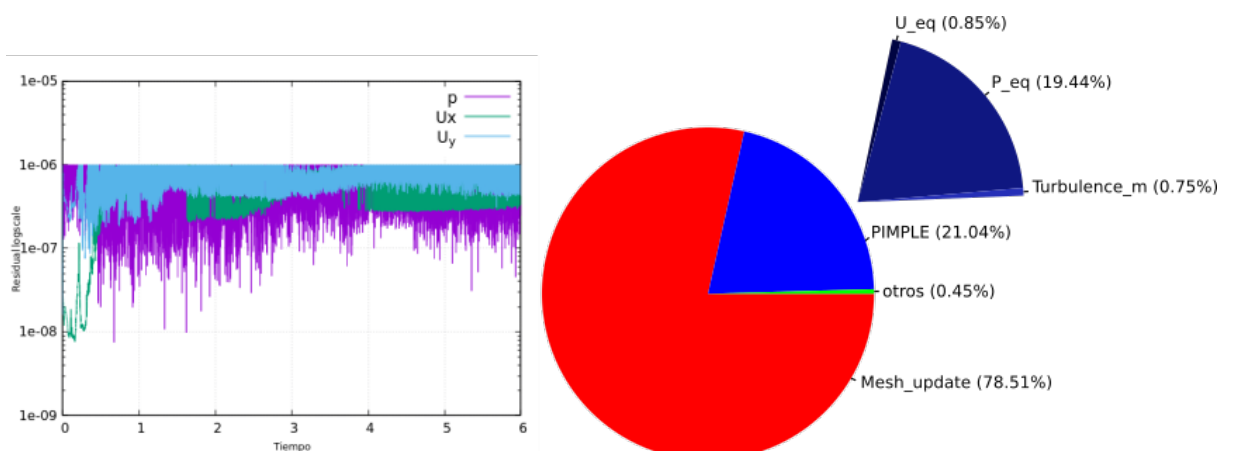


Figura 68: Residuos finales y porcentaje de tiempo insumido en tareas por el solver `overPimpleDyMFoamTimed`

8.3.4. Análisis de resultados

Los resultados obtenidos resultan prometedores, entendiendo que en sí la implementación OM dentro de `OpenFOAM`, aún es reciente y que el caso simulado es un cuerpo romo inmerso en un flujo turbulento, con lo cual el desafío de modelado de la capa límite fue resuelta eficientemente como se observa en la Fig. (62). En gran parte la literatura referida a OM aún trata con modelos sencillos en condiciones de escurrimiento laminar. Por el contrario el caso estudiado combina el modelado de la turbulencia, mallas dinámicas y OM. Resulta evidente la necesidad a futuro de realizar una mejora sobre el proceso de actualización de mallado, observando la porción de tiempo insumido por ella, ver Fig. (68).

El análisis de conservación de flujo global realizado entre las condiciones de borde `inlet` y `outlet` para el tiempo simulado, presentado en la Fig. (63) evidencia una leve variación porcentual. Volkner et al. (2017) explica en su trabajo que esa leve diferencia de conservación de masa en casos OM móviles produce fluctuaciones en el campo de presiones a causa del proceso de interpolación y por ende en el campo de velocidades en función al tipo de formulación de acople que se ha utilizado. La estrategia más eficiente para reducir esas fluctuaciones de presión indeseadas, es asegurando que las mallas en la región de solape posean la misma dimensión. Aún así resulta necesario implementar a futuro correcciones sobre el balance de masa y funciones de interpolación de mayor orden a las evaluadas, en conjunto con la utilización de un modelo de turbulencia que resuelva la transición de la capa límite, para una posterior evaluación de coeficientes aerodinámicos y variables de interés.

8.4. Cuerpo de Ahmed

En la Tab. (11) se detallan las condiciones iniciales impuestas a la malla de fondo y solapada. Siendo U la velocidad de entrada, κ energía cinética turbulenta, ω tasa de disipación turbulenta. La simulación fue llevada a cabo promediando para cada paso de tiempo las ecuaciones de Navier-Stokes (uRANS). Se ha utilizado el modelo de turbulencia $\kappa\omega_{SST}$ (Menter et al., 2003), el cual es capaz de modelar correctamente flujos con gradientes negativos de presión, separación o re ingreso, característicos de la configuración turbulenta del modelo estudiado. Esta versión de $\kappa\omega_{SST}$ no modela los efectos de transición de la capa límite (laminar a turbulenta), pudiendo introducir así errores en los resultados de los coeficientes aerodinámicos. Sin embargo el cambio continuo de posición del modelo presenta una limitación aún para modelos RANS que contemplen tal efecto, los cuales no podrían replicar la transición real por efectos de la actualización continua en la longitud de desarrollo de la capa límite frente al flujo medio incidente.

Se ha configurado un paso de tiempo ajustable limitado por un $Co_{max} = 4$. Se evaluó un marco temporal total de 6,5s. En los primeros 0,2s, se busca estabilizar la capa límite turbulenta en el cuerpo de Ahmed estático alineado al flujo y luego se impone un movimiento al `oversetPatch` de oscilación rotante sobre el eje “z”, definiendo una amplitud y velocidad pequeña de rotación, que permita representar una maniobra de slalom con grandes ángulos barridos (usualmente los ángulos reales barridos son menores y con mayor velocidad angular), que no resulte agresiva para el proceso de actualización de malla y plantillas de interpolación, evitando incrementos en los tiempos de resolución o disparo de errores de conservación y posterior inestabilidad de la simulación, y así calibrar el método.

BC_inlet	$30m/s(U) - 0,0054m^2s^{-2}(k) - 1,055s^{-1}(\omega)$
BC_outlet	0Pa
oversetPatch	$\pm 25\text{grados}, 1rad\ s^{-1}$
Reynolds	2020645

Tabla 11: Condiciones de borde e iniciales.

El acople de presión-velocidad, en conjunto con el movimiento de la malla solapada para OM, se realiza con el solver `overPimpleDyMFoam`. Este fue levemente modificado para medir tiempos inferidos en cada tarea desarrollada. Se usó una configuración (PIMPLE) de 50 iteraciones máximas para el predictor de momento y 2 iteraciones para el corrector de presión, en conjunto con coeficientes de relajación de 0,8 para todas las variables (Holzmann, 2019). La función de interpolación es la misma que en los modelos anteriores `inverseDistance`, por las razones ya discutidas previamente.

Con esta configuración se obtuvieron resultados de coeficientes aerodinámicos, valores de y^+ sobre el cuerpo de Ahmed, residuos de variables, también se evaluó la conservación de masa global mediante balance de flujos entre las condiciones de borde `inlet`, `outlet` y campos de velocidad normalizada en la estela e iso-superficies de vorticidad “Q-criterion”.

8.4.1. Coeficientes aerodinámicos

Todos los coeficientes aerodinámicos fueron calculados para cada paso de tiempo y son filtrados con una media móvil de 50. Los coeficientes de fuerza y momento se presentan en las Ec. (63).

$$C_d = \frac{2F_x}{\rho U^2 A_f} \quad C_{d(f/r)} = 0,5C_d(\pm CM_{Roll}) \quad (63a)$$

$$C_l = \frac{2F_z}{\rho U^2 A_f} \quad C_{l(f/r)} = 0,5C_l(\pm CM_{Pitch}) \quad (63b)$$

$$C_s = \frac{2F_y}{\rho U^2 A_f} \quad (63c)$$

$$CM_{Roll} = \frac{2M_x}{\rho U^2 A_f b} \quad CM_{Pitch} = \frac{2M_y}{\rho U^2 A_f L} \quad CM_{Yaw} = \frac{2M_z}{\rho U^2 A_f b} \quad (63d)$$

La fuerza global $\vec{F} = (F_x\vec{i}, F_y\vec{j}, F_z\vec{k})$ se encuentra representada por sus componentes, al igual que el momento global $\vec{M} = (M_x\vec{i}, M_y\vec{j}, M_z\vec{k})$, $\rho = 1,205kg/m^3$ la densidad del aire, $A_f = 0,112m^2$ el área frontal del cuerpo de Ahmed, b el ancho, L el largo y U la velocidad de fondo definida en la condición de borde `inlet`.

En la Fig. (69)a) se presentan el coeficiente de arrastre global C_d , y sus componentes frontal C_{df} y trasera C_{dr} , Fig. (69)b) el de sustentación global C_l , y sendas componentes frontal C_{lf} y trasera C_{lr} . Además, son graficadas con línea de puntos las posiciones de alineación del cuerpo con el flujo incidente a $0 \pm 12,5$ y ± 25 de ángulo de guiñada.

En primera instancia se simulan 0,2 segundos de forma estática con flujo alineado al cuerpo obteniendo valores cercanos a los publicados por Meile et al. (2016) mediante ensayo experimental y Adotti et al. (2017) mediante simulación RANS, ambos trabajos analizan el ángulo de guiñada de forma estática. Luego se impone movimiento al cuerpo con las características antes descritas, alcanzando mayor amplitud a los 1,5707 y 4,7123 segundos. Los resultados de los coeficientes C_d y C_l con guiñada dinámica, resultan mayores respecto de los trabajos citados con posicionamiento estático utilizados como referencia. La discrepancia puede estar asociada al cambio de posición continua del cuerpo respecto del flujo incidente, provocando que el flujo no se termine de desarrollar, por la complejidad de la geometría en condiciones de viento cruzado y el cambio continuo de los puntos de desprendimiento de capa límite sobre el cuerpo, además de las limitaciones del modelo de turbulencia utilizado en este trabajo, ver Fig. (70).

También en la Fig. (69)c) se expone el coeficiente lateral C_s , en la Fig. (69)d) el coeficiente de rolido CM_{roll} y Fig. (69)f) el coeficiente de guiñada CM_{Yaw} , los cuales presentan simetría durante el tiempo simulado. El coeficiente de cabeceo CM_{Pitch} Fig. (69)e) presenta un comportamiento negativo en casi todo su desarrollo para cuando el cuerpo no está alineado con el flujo, en coincidencia con la magnitud mayor del coeficiente de sustentación trasero C_{lr} , respecto del delantero C_{lf} .

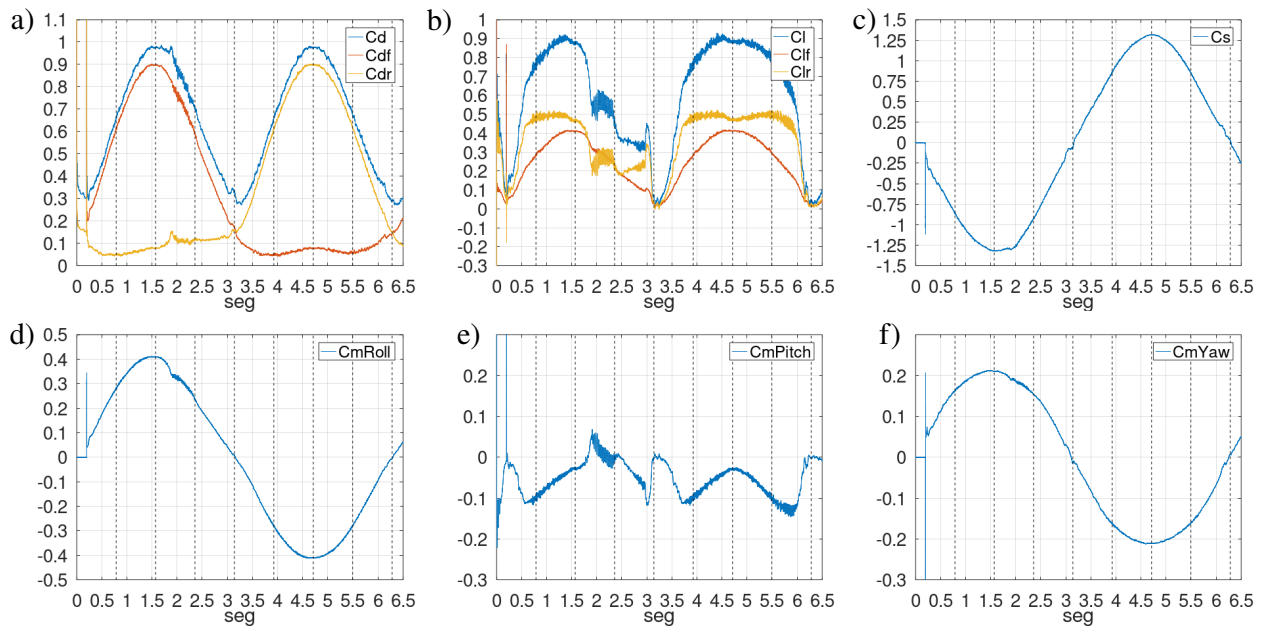


Figura 69: Coeficientes Aerodinámicos: a) arrastre C_d , b) sustentación C_l , c) lateral C_s , d) rolido CM_{roll} , e) cabeceo CM_{pitch} y f) guiñada CM_{yaw}

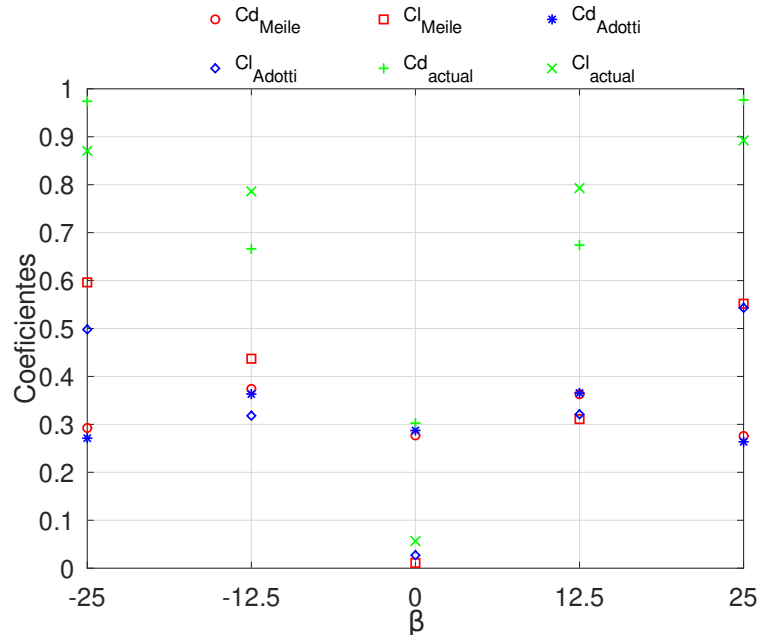


Figura 70: Coeficientes Aerodinámicos: C_d y C_l vs. ángulo de guiñada β . Meile et al. (2016) Adotti et al. (2017)

8.4.2. y^+ Plus

En la Fig. (71) se presentan en histogramas valores de y^+ para las celdas adyacentes al cuerpo de Ahmed. La mayor parte de ellas se encuentran dentro de la sub-capa viscosa $0 < y^+ \leq 5$ y el resto se reparte dentro de la capa de transición $5 \leq y^+ \leq 30$, y una ínfima parte en la región logarítmica. El aumento de celdas en la zona de transición sucede cuando el cuerpo de Ahmed barre el ángulo de guiñada, produciendo un cambio en las componentes de velocidad, difiriendo de la configuración inicial con la que se definió la primer celda de la capa límite, ver Fig. (72). En ésta es posible observar que el valor medio

de y^+ de las celdas de la capa límite en contacto con el cuerpo levemente pertenecen a la región de transición no capturada por el modelo de turbulencia utilizado.

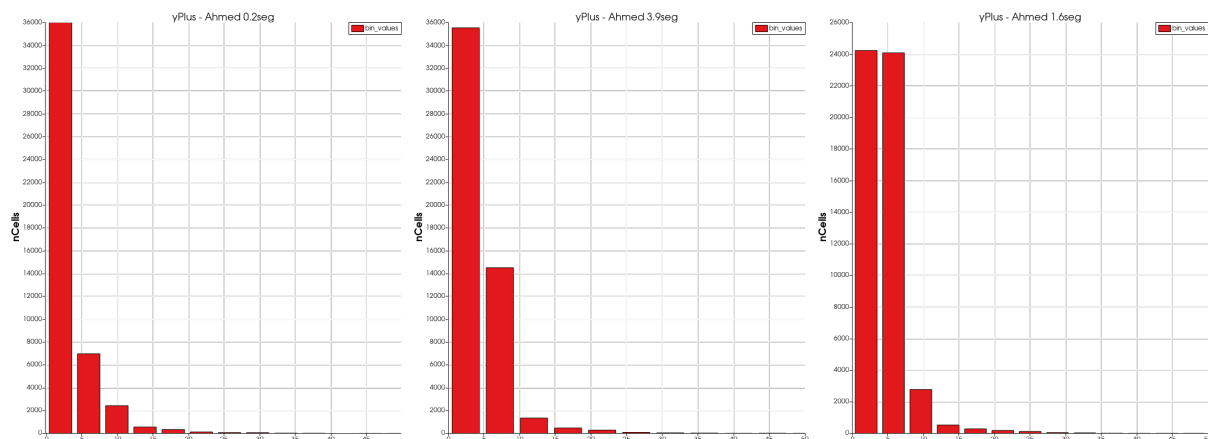


Figura 71: Distribución de y^+ en el cuerpo de Ahmed para 0° , $12,5^\circ$ y 25° .

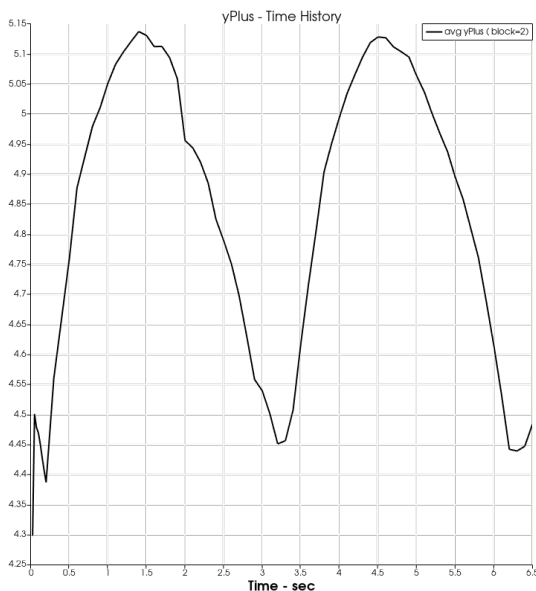


Figura 72: Historia temporal del coeficiente medio $yPlus$ de las primeras celdas de la capa límite del cuerpo de Ahmed.

8.4.3. Análisis de residuos, perfil del solver y conservación global

En la Fig. (73) se presentan los residuos para U_x , U_y , U_z y p . Se observa un comportamiento estable en los residuos finales de las variables en cuestión. Se alcanza la convergencia del loop (PIMPLE) entre 2 y 3 iteraciones a lo largo del tiempo simulado. A su vez también se presenta la distribución porcentual del tiempo insumido en las tareas realizadas por el solver `overPimpleDyMFoamTimed`, sobre un total de 811008 seg $\approx 9,38$ días. La actualización de malla produce un cuello de botella, al realizar las tareas de actualización de matrices con la nueva plantilla de celdas (cálculo, interpolación y huecas) derivadas del movimiento de la malla y posterior interpolación entre los campos de las mallas solapadas, la implementación es detallada por Petra (2019).

Los cálculos se realizaron en el clúster Pirayú (<https://cimec.org.ar/c3/pirayu/index.php>). El dominio fue descompuesto en 16 partes con el método “simple” sobre el

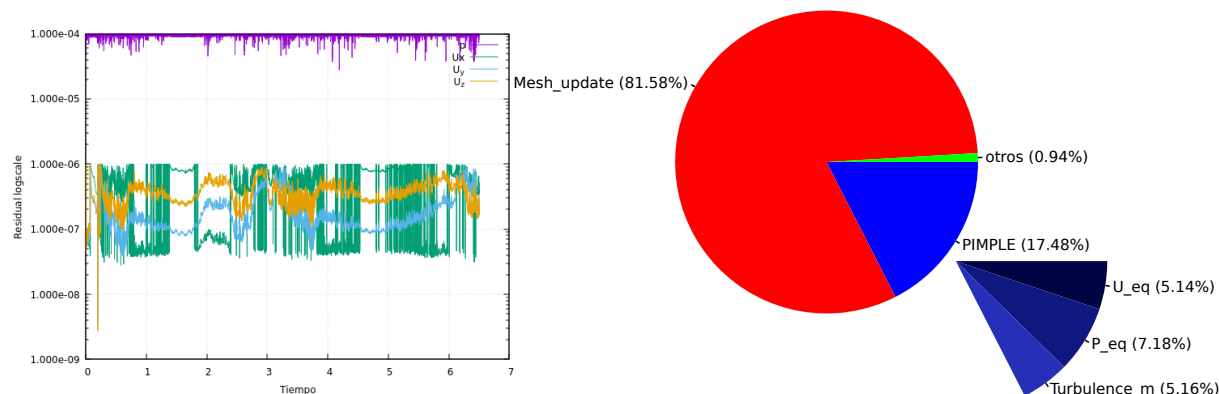


Figura 73: Residuos finales y porcentaje de tiempo insumido en tareas por el solver overPimpleDyMFoamTimed

eje “z”. Es recomendable no dividir el `oversetPatch` para evitar el aumento en tiempo de comunicación entre ellos (Windt et al., 2018). Para este caso eso no resulta posible porque el desbalance de carga al alocar la zona móvil a un solo procesador, produce un aumento en los tiempos de sincronización siendo aún más desfavorable que los tiempos inferidos por la comunicación de procesadores dentro del `oversetPatch` o zona móvil, teniendo en cuenta que ésta contiene el 48 % del total de las celdas del dominio.

Uno de los problemas asociados al OM es la pérdida de conservación de masa por efecto de la interpolación entre los campos de las mallas (Houzeaux et al., 2014). En este trabajo se evalúa la conservación mediante un análisis del error porcentual resultante de la diferencia de flujo ϕ entre las condiciones de borde `inlet` y `outlet` para cada paso de tiempo, el error medio durante toda la simulación es 0,13 % ver Fig. (74).

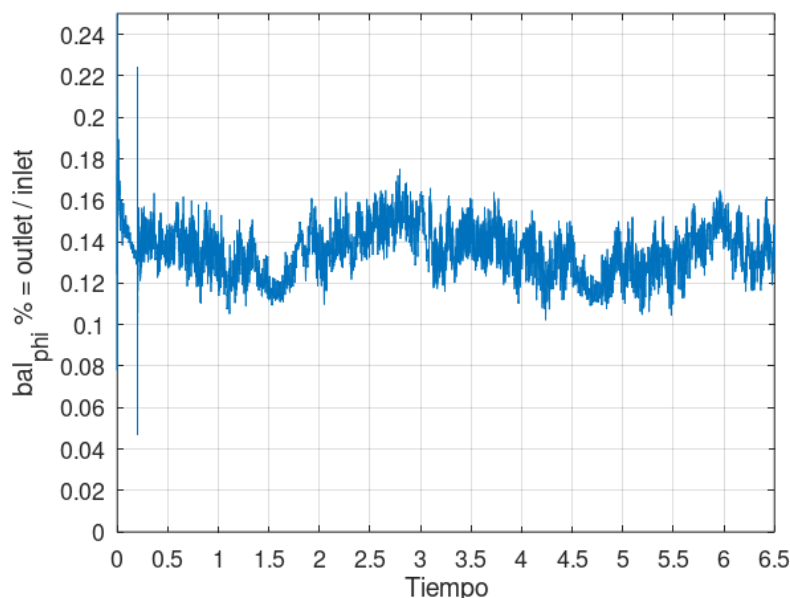


Figura 74: Error porcentual del flujo ϕ entre entrada y salida del dominio computacional.

8.4.4. Iso superficies de vorticidad y Campo de velocidad en la estela

Existen regiones de vórtices coherentes, donde la concentración de vorticidad es suficientemente elevada para inducir un torbellino en el flujo, manteniendo una forma carac-

terística por un tiempo prolongado, mayor al tiempo de rotación local (Lesieur, 2008).

Es posible capturar estas estructuras mediante iso-superficies de alto módulo de vorticidad, o componentes de vorticidad, o baja presión. El método denominando Q-criterion propuesto por Hunt et al. (1988) es bastante eficaz para determinar estas iso-superficies en flujos incompresibles, identificando vórtices en determinadas zonas del fluido donde el valor del segundo tensor invariante de ∇u es positivo.

El método se basa en la descomposición del tensor gradiente velocidad de deformación ∇u en dos partes, una denominada anti-simétrica Ω_{ij} Ec. (64b) y la otra simétrica S_{ij} Ec. (64c), el primer término representa la energía cinética turbulenta y el segundo la tasa de disipación de energía cinética turbulenta.

$$Q = \frac{1}{2}(\Omega_{ij}\Omega_{ij} - S_{ij}S_{ij}) \quad (64a)$$

$$\Omega_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right) \quad (64b)$$

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (64c)$$

Cuando $Q > 0$, el método representa regiones donde la vorticidad es mayor que la deformación del fluido debido a los esfuerzos cortantes producto de la adhesión viscosa, por lo tanto se pueden observar las estructuras coherentes y distinguir vórtices en ellas, donde las partículas del fluido rotan como un sólido sin deformarse, y lo hacen en conjunto formando vórtices dentro del dominio analizado.

En la Fig. (75) se presentan las iso-superficies de vorticidad para condición de flujo alineado y para el cuerpo de Ahmed en posición de guiñado, mostrando concordancia con los esquemas de estados dominantes de vorticidad reportados por Meile et al. (2016), ver Fig. (76).²

Se evalúa el campo de velocidad normalizado en cinco planos “x-z” en dirección del eje “y” calculado a partir de la Ec. (65), encontrando longitudes de estela similar a la referencia para ángulos de guiñada $\beta = 0^\circ$ y $\beta = -12,5^\circ$, presentes en las Fig. (77) y Fig. (78).

$$uw_{norm} = \frac{\sqrt{(u^2 + w^2)}}{U} \quad (65)$$

²Figura extraída de la fuente citada.

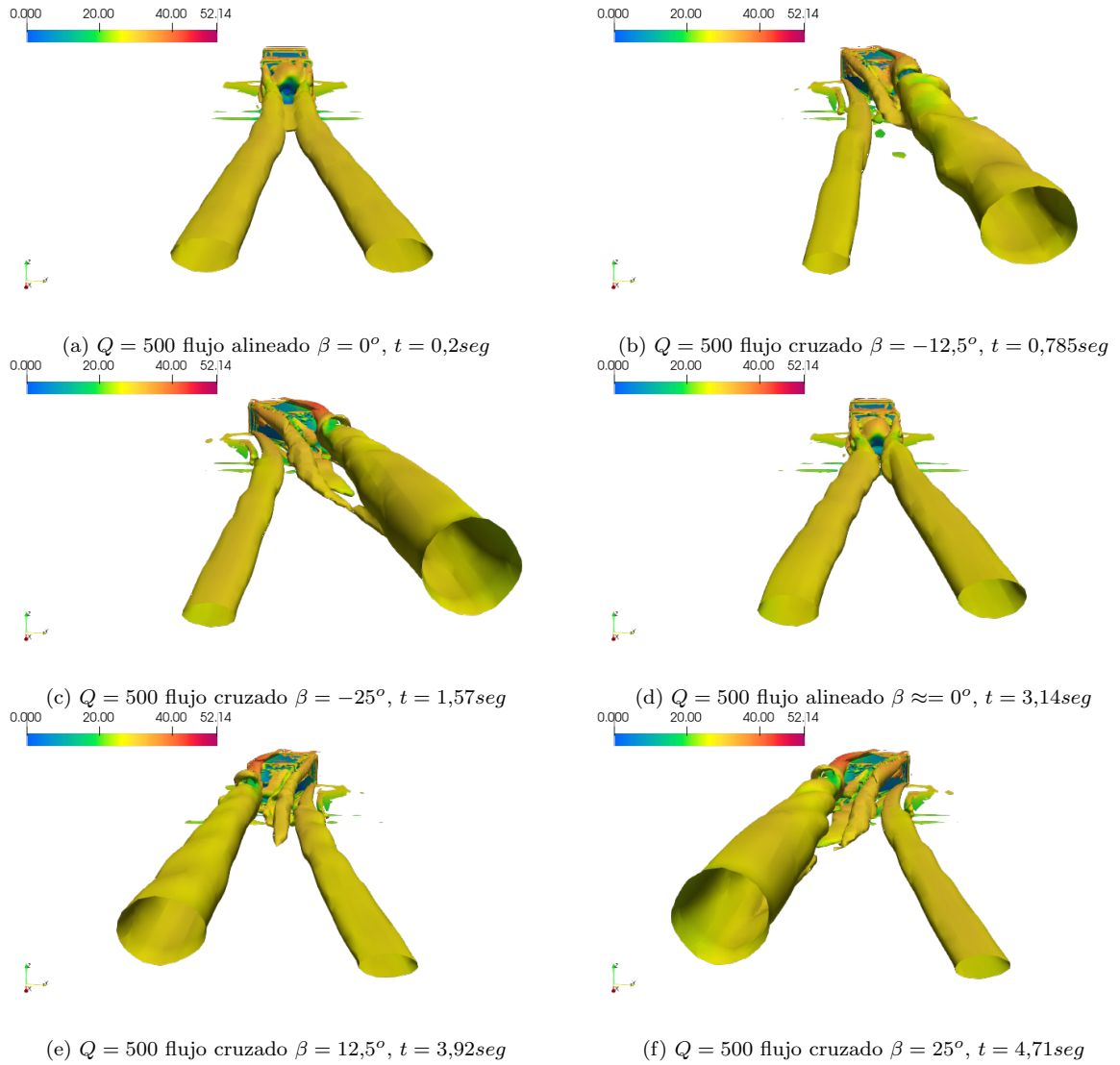


Figura 75: Contornos de iso-superficie para distintos ángulos de guiñada β en el tiempo.

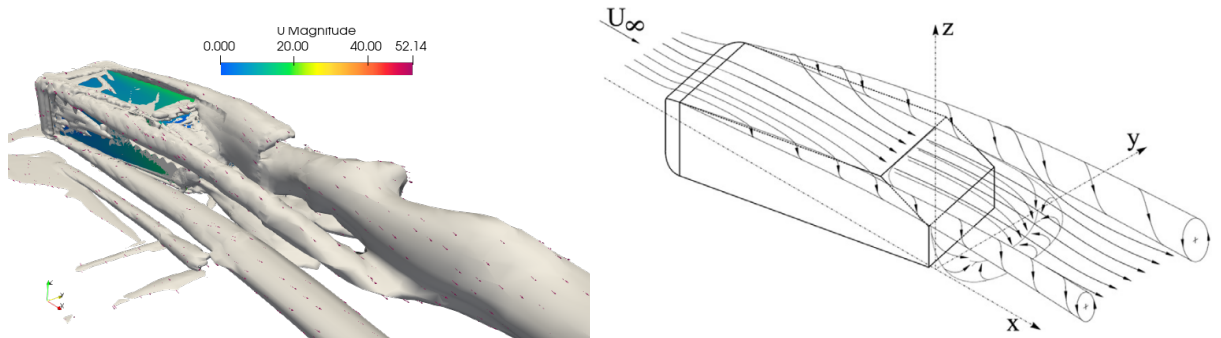


Figura 76: $Q = 500$ flujo cruzado $\beta = -12,5^\circ$, $t = 0,785\text{seg}$ y esquema de vórtices dominantes reportado por Meile et al. (2016).

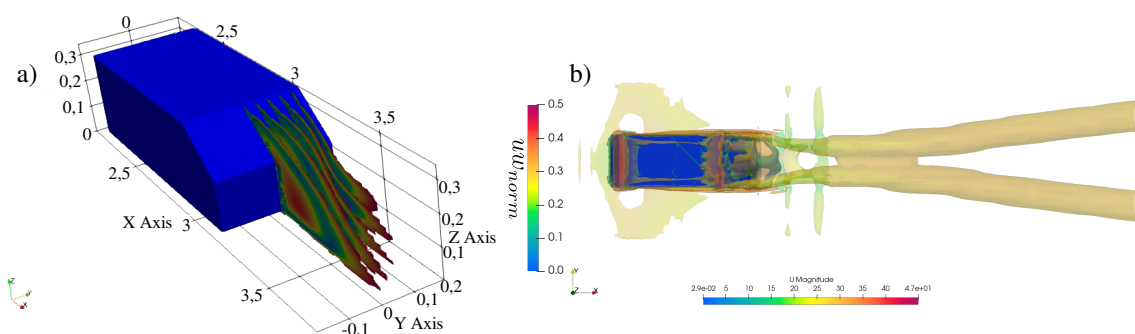


Figura 77: a) Campo de velocidad normalizado en la estela y b) Q-criterio 500 para $\beta = 0^\circ$

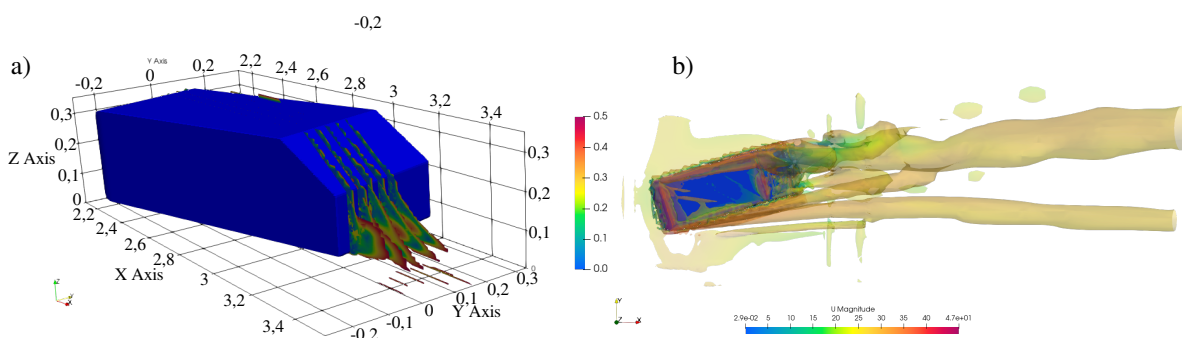


Figura 78: a) Campo de velocidad normalizado en la estela y b) Q-criterio 500 para $\beta = -12,5^\circ$

8.4.5. Análisis de resultados

El análisis tridimensional del flujo sobre el cuerpo de Ahmed presenta varios desafíos aún en casos de simulaciones con metodologías tradicionales, considerando como tales condiciones estáticas. Algunas de las características que lo tornan complejo son: el flujo en la estela turbulenta, los desprendimientos de capa límite sobre la luneta y la combinación de capas límites en la zona del piso. El estudio llevado a cabo en el presente trabajo aumenta el grado de complejidad al insertar el análisis móvil del ángulo de guiñada en el tiempo, el cual es simulado con la metodología de mallas solapadas OM. Actualmente la metodología utilizada se encuentra en vías de desarrollo en **OpenFOAM** y otras plataformas de simulación, como es fundamentado en la bibliografía citada durante el desarrollo de la tesis, por lo tanto analizar al cuerpo de Ahmed en condiciones móviles de flujo turbulento somete a la metodología OM a un campo de pruebas exigente.

Los resultados obtenidos en la evaluación de los coeficientes aerodinámicos son prueba de ello. Las condiciones de viento cruzado y con el cuerpo de Ahmed en movimiento pueden contener una componente de error algo superior a la deseada provocada por los efectos de no conservación de la masa, donde aún en variaciones inferiores al 1 % del flujo ϕ , puede ocasionar fluctuaciones indeseadas sobre el campo de presión desencadenando un incremento no deseado en el campo de velocidades (Volkner et al., 2017). Ha de tenerse en cuenta también la limitación del modelo de turbulencia $k\omega_{SST}$ que no reproduce la transición de la capa límite y aún más la limitación de los modelos RANS asociada al cambio continuo de la longitud de desarrollo de la capa límite. Un paso siguiente como se propuso sería utilizar un modelo RANS que contemple la transición mencionada. Por otra parte el uso de otra estrategia de modelado turbulento como ser LES podría capturar el

cambio constante de longitud de desarrollo de la capa límite en las condiciones del cuerpo móvil, a costas de un conocido incremento notable de recursos y tiempo computacional insumido al resolver las escalas de torbellinos más grandes y una parte del rango subinercial.

El OM en `OpenFOAM`, aún se encuentra en etapas iniciales de optimización para el cálculo paralelo, el aumento en cantidad de celdas resulta restrictivo contemplando el tiempo inferido en la malla utilizada en este trabajo (9,38 días), con el equipamiento utilizado.

9. Conclusión

En la presente tesis se han evaluado casos con distintos niveles de complejidad y se ha demostrado que OM resulta ser una herramienta muy versátil para simular no solo grandes desplazamientos, sino trayectorias completas en el tiempo manteniendo alta calidad de mallado.

La complejidad que representa la configuración del mallado y solver para los casos analizados mediante OM, es compensada con la seguridad de no trabajar con mallas deformables que puedan inferir en errores por violación de la GCL. No obstante fue posible realizar múltiples modificaciones a la malla de base y solapada sin mayores problemas una vez bien desarrollados y entendidos los procesos requeridos para un mallado correcto.

Las funciones de interpolación evaluadas demuestran ser suficientes para casos donde no exista movilidad, pero al aportar movimiento al `oversetPatch` se generan inestabilidades sobre el campo de velocidades en la zona de acople de mallas. En todos los casos evaluados se busca mitigar tales efectos indeseados mediante una correcta estrategia de mallado en la zona de solape de mallas, lo cual resulta ser el factor de mayor peso respecto de la sensibilidad de resultados obtenidos. No obstante es necesario en trabajos futuros mejorar el comportamiento de las funciones de interpolación para evaluar casos móviles.

El paralelismo para el OM es evaluado especificándose recomendaciones para disminuir los tiempos de cálculo y conservar buenos resultados en las métricas de rendimiento en función a los recursos disponibles. Siempre que sea posible la zona 1 (móvil) debe estar alocada en la región computacional correspondiente a un único procesador para disminuir los tiempos de comunicación. Siendo no siempre posible implementar tal estrategia de paralelismo mencionada por problemas de desbalance de carga entre procesadores. La cantidad de procesadores óptimos también es evaluada. En caso de requerir más de un nodo se evidencia un aumento considerable en los tiempos de cálculo producidos por la comunicación ya no a velocidad de bus de placa madre sino, por medio de conexiones tipo InfiniBand o Ethernet, proceso que se ve magnificado por el constante cambio de posición del `oversetPatch` en simulaciones con mallas móviles. La implementación en paralelo aún necesita desarrollo, considerando que el lanzamiento fue realizado en la versión `OpenFOAMv1706` en el año 2017, aún se encuentra en etapas iniciales, en la `OpenFOAMv1906` presentan una mejora para el caso de las mallas móviles las cuales en versiones anteriores no aseguran que la celda donante pertenezca al mismo procesador que la celda receptor. En esta tesis se han utilizado versiones superiores a la última citada.

El método OM resulta una herramienta poderosa para la simulación de objetos móviles con formas superficiales complejas y respectivas capas límites, a lo largo de la presente tesis se han especificado las ventajas, desafíos e inconvenientes asociados a su utilización.

10. Referencias

Referencias

- Adalsteinsson, D. and Sethian, J. A. (1995). A Fast Level Set Method for Propagating Interfaces. *Journal of Computational Physics*, 118(2):269,277.
- Adotti, M. I., Castro, H. G., Paz, R. R., and Bortoli, M. D. (2017). Simulación Computacional del Cuerpo de Ahmed bajo Distintos ángulos de Incidencia de Viento. *Asociación Argentina de Mecánica Computacional. AMCA*, 37.
- Ahmed, S., Ramm, G., and Faltin, G. (1984). Some Salient Features of The Time-Averaged Ground Vehicle Wake. *SAE International Congress and Exposition*, page 34.
- Alletto, M. (2022). Comparison of Overset Mesh with Morphing Mesh: Flow Over a Forced Oscillating and Freely Oscillating 2D Cylinder. *OpenFOAM Journal*, 2(1):13,30.
- Arvidson, S. (2017). *Methodologies for RANS-LES interfaces in turbulence-resolving simulations*. PhD thesis, CHALMERS UNIVERSITY OF TECHNOLOGY. Gothenburg, Sweden.
- Bartoli, G., Bruno, L., Buresti, G., Ricciardelli, F., M.V., S., and A., Z. (2008). BARC Overview Document. Available online. <http://www.aniv-iawe.org/barc>.
- Blades, E. and Marcum, D. (2005). A Sliding Interface Method for Unsteady Unstructured Flow Simulations. *International Journal for Numerical Methods in Fluids*, 53:507,529.
- Bogaers, A. (2015). *Efficient and robust partitioned solution schemes for fluid-structure interactions*. PhD thesis, Faculty of Engineering and the Built Environment, University of Cape Town. South Africa.
- Bruno, L., Salvetti, M. V., and Ricciardelli, F. (2014). Benchmark on the Aerodynamics of a Rectangular 5:1 Cylinder: An overview after the first four years of activity. *Journal of Wind Engineering and Industrial Aerodynamics*, 126:87–106.
- Castro, H. G. (2013). *Inlet Turbulence Generation for Large Eddy Simulation: Applications to Fluid-Structure Interaction Problems*. PhD thesis, Universidad Nacional del Litoral. Facultad de Ingeniería y Ciencias Hídricas; Argentina.
- Chandar, D., Boppana, B., and Kumar, V. (2018). A Comparative Study of Different Overset Grid Solvers Between OpenFOAM, StarCCM+ and Ansys-Fluent. In *2018 AIAA Aerospace Sciences Meeting*.
- Chesshire, G. and Henshaw, W. (1990). Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics*, 90(1):1–64.
- Coetzee, R. V. (2005). *Volume weighted interpolation for unstructured meshes in the Finite Volume Method*. PhD thesis, North-West University, Potchefstroom Campus.
- Collar, A. R. (1946). The expanding domain of aeroelasticity. *The Royal Aeronautical Society*, pages 613–636.
- Donea, J., Huerta, A., Ponthot, J., and Rodríguez-Ferran, A. (2004). *Arbitrary Lagrangian-Eulerian Methods*, chapter 14, pages 413–437. Wiley Online Library.
- Dowell, E., E., C., H., C., D., P., R., S., and F., S. (2015). *A Modern Course in Aeroelasticity*. Solid Mechanics and its Applications. Springer.
- Ferziger, J. H. and Perić, M. (2020). *Computational Methods for Fluid Dynamics. 4th Edition*. Springer International Publishing.
- Gamnitzer, P. and Wall, W. A. (2006). An ALE-Chimera Method for Large Deformation Fluid Structure Interaction. In *ECCOMAS CFD 2006 - European Conference on Computational Fluid Dynamics*.
- Gretler, W. and Meile, W. (1993). Der 2m windkanal am institut fur stromungslehre und gasdynamik der technischen universitat graz. *Osterreichische Ingenieur und Architekten Zeitschrift*.

- Guerrero, J. (2009). *Numerical simulation of the unsteady aerodynamics of flapping flight*. PhD thesis, University of Genoa, Italy.
- Guerrero, J. (2019). A comparative assessment and benchmarking study of OpenFOAM overset meshes capabilities. Wolf Dynamics in 7th ESI OpenFOAM Conference, Germany.
- Hadzic, H. (2006). *Development and Application of Finite Volume Method for the Computation of Flows Around Moving Bodies on Unstructured, Overlapping Grids*. PhD thesis, Technische Universitat Hamburg.
- Hirt, C. and Nichols, B. (1981). Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201,225.
- Holzinger, G. (2020). *OpenFOAM - a little user manual*.
- Holzmann, T. (2019). *Mathematics, Numerics, Derivations and OpenFOAM*. Holzmann CFD.
- Houzeaux, G., Eguzkitza, B., Aubry, R., Owen, H., and Vázquez, M. (2014). A Chimera method for the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 75(3):155,183.
- Hunt, J. C. R., Wray, A. A., and Moin, P. (1988). Eddies, streams, and convergence zones in turbulent flows. *Technical Memorandum (TM), NTRS - NASA Technical Reports Server*.
- Issa, R., Gosman, A., and Watkins, A. (1986). The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme. *Journal of Computational Physics*, 62(1):66–82.
- Jasak, H. (1996). *Error Analysis and Estimation for the Finite Volume Method With Applications to Fluid Flows*. PhD thesis, Imperial College London.
- Jasak, H. and Tuković, Z. (2015). Immersed Boundary Method in FOAM Theory, Implementation and Use.
- Kasper, J., Mohamed cherif, M., Stoevesandt, B., and Peinke, J. (2015). Towards an implementation of the overlapping grid method in OpenFOAM.
- Kundu, P., Cohen, I., and Dowling, D. (2012). *Fluid Mechanics*. Academic Press.
- Kurtulus, D. F. (2015). On the Unsteady Behavior of the Flow around NACA 0012 Airfoil with Steady External Conditions at Re=1000. *International Journal of Micro Air Vehicles*, 7(3):301–326.
- Laws, P., Saini, J. S., and Kumar, A. (2019). A study on OpenFOAM's Overset Mesh Support using flow past NACA 0018 airfoil. *Preprints*, pages 1–18.
- Lesieur, M. (2008). *Turbulence in Fluids*. Springer.
- Liu, F. (2016). A Thorough Description Of How Wall Functions Are Implemented In OpenFOAM. In *Proceedings of CFD with OpenSource Software, Edited by Nilsson. H.*
- Meakin, R. (1993). *Moving body overset grid methods for complete aircraft tiltrotor simulations*, pages 576,588.
- Meakin, R. (1994). On the spatial and temporal accuracy of overset grid methods for moving body problems. *12th AIAA Applied Aerodynamics Conference, AIAA-94-1925*.
- Meile, W., Ladinek, T., Brenn, G., Reppenhausen, A., and Fuchs, A. (2016). Non-symmetric bi-stable flow around the Ahmed body. *International Journal of Heat and Fluid Flow*, 57:34,47.
- Menter, F., Kuntz, M., and Langtry, R. (2003). Ten years of industrial experience with the SST turbulence model. *Heat and Mass Transfer*, 4.
- Menter, F. R. (1992). Improved two-equation $\kappa - \omega$ turbulence models for aerodynamic flows. NASA Ames Research Center Moffett Field, CA, United States. *Technical Memorandum (TM)*.

- Moukalled, F., Mangani, L., and Darwish, M. (2016). *The Finite Volume Method in Computational Fluid Dynamics An Advanced Introduction with OpenFOAM and Matlab*.
- Panadero, J. (2015). *Predicción del rendimiento: Análisis de la escalabilidad de aplicaciones paralelas*. PhD thesis, Universitat Autònoma de Barcelona.
- Part-Enander, E. and Sjogreen, B. (1994). Conservative and non-conservative interpolation between overlapping grids for Finite Volume solutions of hyperbolic problems. *Computers & Fluids*, 23(3):551–574.
- Patankar, S. and Spalding, D. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15(10):1787–1806.
- Peskin, C. (2002). The Immersed Boundary Method. *Cambridge University Press*, 11:479,517.
- Petersson, N. (1999). An algorithm for assembling overlapping grid systems. *SIAM J. Sci. Comput.*, 20:1995–2022.
- Petra, T. (2019). Description of the overset mesh approach in esi version of openfoam. In *Proceedings of CFD with OpenSource Software, Edited by Nilsson H*.
- Prewitt, N. C., Belk, D. M., and Shyy, W. (2000). Parallel computing of overset grids for aerodynamic problems with moving objects. *Progress in Aerospace Sciences*, 36(2):117,172.
- Rossitto, G., Sicot, C., Ferrand, V., Jacques, B., and Harambat, F. (2016). Drag and crosswind sensitivity of rounded fastback vehicle.
- Schlichting, H. and Gersten, K. (2017). *Boundary-Layer Theory*.
- Storti, B. (2020). *Desarrollo de un método de composición de dominios tipo chimera en mecánica computacional*. PhD thesis, Universidad Nacional del Litoral. Facultad de Ingeniería y Ciencias Hídricas; Argentina.
- Tennekes, H. and Lumley, J. (1978). *A First Course in Turbulence*. MIT Press.
- Trépanier, J., Reggio, M., Zhang, H., and Camarero, R. (1991). A Finite-Volume Method for the Euler equations on Arbitrary Lagrangian-Eulerian grids. *Computers and Fluids*, 20(4):399,409.
- Uroić, T. (2019). *Implicitly Coupled Finite Volume Algorithms*. PhD thesis, University of Zagreb Faculty of Mechanical Engineering and Naval Architecture.
- Verma, S. and Hemmati, A. (2020). Performance of overset mesh in modeling the wake of sharp-edge bodies. *Computation*, 8(3).
- Versteeg, H. K. and Malalasekera, W. (1995). *An Introduction to Computational Fluid Dynamics - the Finite Volume Method*. Addison-Wesley-Longman.
- Volkner, S., Brunswig, J., and Rung, T. (2017). Analysis of non-conservative interpolation techniques in overset grid Finite-Volume Methods. *Computers & Fluids*, 148:39–55.
- Wimmer, K. (1993). *Two-Dimensional Nonplanar Process Simulation*. PhD thesis, Institute for Microelectronics Technical University Vienna. Austria.
- Wimshurst, A. (2020). Multiple Reference Frame (MRF) Approach for Turbomachinery. https://www.youtube.com/watch?v=oa-xcE0_0UY&ab_channel=FluidMechanics101. [Online; posted 24-May-2020].
- Wimshurst, A. (2021). Eddy viscosity models for RANS and LES. https://www.youtube.com/watch?v=SVYXNICeNWA&t=2080s&ab_channel=FluidMechanics101. [Online; posted 24-Feb-2021].
- Windt, C., Davidson, J., Akram, B., and Ringwood, J. (2018). Performance Assessment of the Overset Grid Method for Numerical Wave Tank Experiments in the OpenFOAM Environment.
- Wolf, C. (2011). *A Chimera Simulation Method and Detached Eddy Simulation for Vortex-Airfoil Interactions*. PhD thesis, Georg-August-Universität Gottingen.

- Wu, B., Li, S., Li, K., and Zhang, L. (2020). Numerical and experimental studies on the aerodynamics of a 5:1 rectangular cylinder at angles of attack. *Journal of Wind Engineering and Industrial Aerodynamics*, 199:104097.
- Young, L., Dominguez, M., Allan, B., and Malpica, C. (2021). NASA Aeronautics Contributions to Ingenuity Mars Helicopter. *Seminar*, pages 1–39.

11. Anexo

11.1. Computación de Alto Rendimiento HPC

La HPC se focaliza en el desarrollo de supercomputadoras, programas de procesamiento paralelo y software relacionado como `MPI` o (`PETSc`, por Portable, Extensible Toolkit for Scientific Computation), `OpenFOAM` corre en paralelo problemas bajo la implementación de dominio público OpenMPI.

El método de cálculo paralelo usado por el software es la descomposición de dominios, donde la geometría y los campos asociados son divididos en piezas y alocaos en procesadores separados para la solución. El proceso de cálculo computacional paralelo incluye: descomposición de la malla y campos, correr la aplicación en paralelo y post-procesar los resultados divididos.

Una tarea común en HPC es medir la escalabilidad de una aplicación. Esta medición indica cuán eficiente es una aplicación cuando se incrementan el números de elementos de procesamiento paralelo. (CPUs / núcleos / procesos / hilos).

Existen dos formas básicas de medir la eficiencia paralela de una aplicación cualquiera. Éstas son la Escalabilidad Fuerte (limitada por PC) y Débil (limitada por RAM).

11.1.1. Escalabilidad fuerte

En este caso el tamaño del problema es fijo y el número de los elementos de procesamiento son incrementados. Esta forma es utilizada para determinar en corridas en serie que tardan mucho tiempo, un punto óptimo de paralelismo que logre realizar el cálculo en una cantidad de tiempo razonable y que no genere sobrecarga paralela (pérdida de tiempo en comunicación-sincronización entre los procesos). Una aplicación se considera que escala linealmente si la aceleración es igual al número de elementos procesamiento paralelo usados, es decir que posea un comportamiento lineal. En general no es fácil conseguir escalabilidad por la sobrecarga de comunicación que se incrementa en proporción a los elementos de proceso.

El cálculo de eficiencia contempla el tiempo total demandado por un elemento de procesamiento t_1 en realizar un trabajo dado, y el tiempo total en completar el mismo trabajo con N elementos de proceso t_N , la eficiencia de escalabilidad fuerte expresada como un porcentaje lineal se expresa:

$$\frac{t_1}{t_N} 100 \% \quad (66)$$

11.1.2. Métricas y el OM

Resulta de interés conocer el comportamiento de la implementación de cálculo paralelo para el método OM dentro del software `OpenFOAM`. Los requerimientos solicitados por el método normalmente son elevados, producto de mallas densas implementadas en las zonas móviles o solapadas con el objetivo de obtener buenos resultados, en combinación con la resolución temporal promediada de las ecuaciones de Navier-Stokes (`uRANS`) para flujo incompresible, implementando algún algoritmo de acople presión-velocidad, y un modelo


```

62     {
63     argList::addNote
64     (
65     "Transient_solver_for_incompressible,_turbulent_flow"
66     "_on_a_moving_mesh."
67     );
68
69     #include "postProcess.H"
70
71     #include "setRootCaseLists.H"
72     #include "createTime.H"
73     #include "createDynamicFvMesh.H"
74     #include "initContinuityErrs.H"
75
76     pimpleControl pimple(mesh);
77
78     #include "createFields.H"
79     #include "createUf.H"
80     #include "createMRF.H"
81     #include "createFvOptions.H"
82     #include "createControls.H"
83     #include "CourantNo.H"
84     #include "setInitialDeltaT.H"
85
86     turbulence->validate();
87     /*Simply saying: To access members of a structure, use the dot operator.
88     To access members of a structure through a pointer, ...
89     use the -> arrow operator.*/
90
91     // * * * * *
92
93     Info<< "\nStarting_time_loop\n" << endl;
94
95     //Added code
96     #include "StopWatch.H"
97     StopWatch totalTime;
98     StopWatch mainLoopTime;
99     StopWatch readControlsTime;
100    StopWatch CourantNoTime;
101    StopWatch setDeltaTTime;
102    StopWatch infoTime;
103    StopWatch meshUpdateTime;
104    StopWatch setCellMaskTime;
105    StopWatch setInterpolatedCellsTime;
106    StopWatch interpolationsTime;
107    StopWatch correctPhiTime;
108    StopWatch relativeFluxTime;
109    StopWatch meshCourantNoTime;
110    StopWatch pimpleTime;
111    StopWatch UEqnTime;
112    StopWatch pEqnTime;
113    StopWatch turbEqnTime;
114    StopWatch writeTime;
115    /*StopWatch absoluteFluxTime;*/
116    //end
117
118    //Added code
119    totalTime.start();//1
120    //end
121
122    while (runTime.run())
123    {
124
125    mainLoopTime.start(); //Added code 2
126
127    readControlsTime.start(); //Added code 3
128    #include "readControls.H"
129    readControlsTime.stop(); //Added code 3
130
131    CourantNoTime.start(); //Added code 4
132    #include "CourantNo.H"
133    CourantNoTime.stop(); //Added code 4
134
135    setDeltaTTime.start(); //Added code 5

```

```

136     #include "setDeltaT.H"
137     setDeltaTTime.stop(); //Added code 5
138
139
140     ++runTime;
141
142     infoTime.start(); //Added code 6
143     Info<< "Time=_" << runTime.timeName() << nl << endl;
144     infoTime.stop(); //Added code 6
145
146     meshUpdateTime.start(); //Added code 7
147     bool changed = mesh.update();
148     meshUpdateTime.stop(); //Added code 7
149
150
151     interpolationsTime.start(); //Added code 10
152     if (changed)
153     {
154         setCellMaskTime.start(); //Added code 8
155         #include "setCellMask.H"
156         setCellMaskTime.stop(); //Added code 8
157
158         setInterpolatedCellsTime.start(); //Added code 9
159         #include "setInterpolatedCells.H"
160         setInterpolatedCellsTime.stop(); //Added code 9
161
162         surfaceScalarField faceMaskOld
163         (
164             localMin<scalar>(mesh).interpolate(cellMask.oldTime())
165         );
166
167         // Zero Uf on old faceMask (H-I)
168         Uf *= faceMaskOld;
169         // Update Uf and phi on new C-I faces
170         Uf += (1-faceMaskOld)*fvc::interpolate(U);
171         phi = mesh.Sf() & Uf;
172
173         // Zero phi on current H-I
174         surfaceScalarField faceMask
175         (
176             localMin<scalar>(mesh).interpolate(cellMask)
177         );
178         phi *= faceMask;
179     }
180     interpolationsTime.stop(); //Added code 10
181
182     correctPhiTime.start(); //Added code 11
183     if (mesh.changing() && correctPhi)
184     {
185         // Calculate absolute flux from the mapped surface velocity
186         #include "correctPhi.H"
187     }
188     correctPhiTime.stop(); //Added code 11
189
190     relativeFluxTime.start(); //Added code 12
191     // Make the flux relative to the mesh motion
192     fvc::makeRelative(phi, U); //https://www.cfd-online.com/Forums/...
193     openfoam-programming-development/122415-dynamic-mesh-adustment-phi.html
194     relativeFluxTime.stop(); //Added code 12
195
196     meshCourantNoTime.start(); //Added code 13
197     if (mesh.changing() && checkMeshCourantNo)
198     {
199         #include "meshCourantNo.H"
200     }
201     meshCourantNoTime.stop(); //Added code 13
202
203     pimpleTime.start(); //Added code 14
204
205     // --- Pressure-velocity PIMPLE corrector loop
206     while (pimple.loop())
207     {
208         UEqnTime.start(); //Added code 15
209         #include "UEqn.H"

```

```

210     UEqnTime.stop(); //Added code 15
211
212     pEqnTime.start(); //Added code 16
213     // --- Pressure corrector loop
214     while (pimple.correct())
215     {
216     #include "pEqn.H"
217     }
218     pEqnTime.stop(); //Added code 16
219
220
221     turbEqnTime.start(); //Added code 17
222     if (pimple.turbCorr())
223     {
224     laminarTransport.correct();
225     turbulence->correct();
226     }
227     turbEqnTime.stop(); //Added code 17
228     }
229     pimpleTime.stop(); //Added code 14
230
231
232     writeTime.start(); //Added code 18
233     runTime.write();
234     writeTime.stop(); //Added code 18
235
236     infoTime.start();
237     runTime.printExecutionTime(Info);
238     infoTime.stop();
239
240     mainLoopTime.stop(); //Added code 2
241     }
242
243
244     //Added code
245     totalTime.stop(); //1
246
247     Info<<"Time_Profile:_"
248     <<"\n\tTotal_Time_(s):_" << totalTime.getTotalTime()
249     <<"\n\tMain_Loop_Time_(%):_" << mainLoopTime.getTotalTime()/
250     totalTime.getTotalTime()*100.0
251     <<"\n\tRead_Controls_Time_(%):_" << readControlsTime.getTotalTime()/
252     totalTime.getTotalTime()*100.0
253     <<"\n\tCourant_Number_Time_(%):_" << CourantNoTime.getTotalTime()/
254     totalTime.getTotalTime()*100.0
255     <<"\n\tSet_Delta_T_Time_(%):_" << setDeltaTTime.getTotalTime()/
256     totalTime.getTotalTime()*100.0
257     <<"\n\tInfo_Time_(%):_" << infoTime.getTotalTime()/
258     totalTime.getTotalTime()*100.0
259     <<"\n\tMesh_Update_Time_(%):_" << meshUpdateTime.getTotalTime()/
260     totalTime.getTotalTime()*100.0
261     <<"\n\tSet_Cell_Mask_Time_(%):_" << setCellMaskTime.getTotalTime()/
262     totalTime.getTotalTime()*100.0
263     <<"\n\tSet_Interpolated_Cells_Time_(%):_" << setInterpolatedCellsTime.
264     getTotalTime()/totalTime.getTotalTime()*100.0
265     <<"\n\tinterpolations_Time_(%):_" << interpolationsTime.getTotalTime()
266     /totalTime.getTotalTime()*100.0
267     <<"\n\tCorrect_Phi_Time_(%):_" << correctPhiTime.
268     getTotalTime()/totalTime.getTotalTime()*100.0
269     <<"\n\tRelative_Flux_Time_(%):_" << relativeFluxTime.getTotalTime()/
270     totalTime.getTotalTime()*100.0
271     <<"\n\tMesh_Courant_Number_Time_(%):_" << meshCourantNoTime.
272     getTotalTime()/totalTime.getTotalTime()*100.0
273     <<"\n\tPIMPLE_Time_(%):_" << pimpleTime.getTotalTime()/
274     totalTime.getTotalTime()*100.0
275     <<"\n\t\tU_Equation_Time_(%):_" << UEqnTime.getTotalTime()/
276     totalTime.getTotalTime()*100.0
277     <<"\n\t\ttp_Equation_Time_(%):_" << pEqnTime.getTotalTime()/
278     totalTime.getTotalTime()*100.0
279     <<"\n\t\tTurbulence_Model_Equation_Time_(%):_" << turbEqnTime.
280     getTotalTime()/totalTime.getTotalTime()*100.0
281     <<"\n\tWrite_Time_(%):_" << writeTime.getTotalTime()/
282     totalTime.getTotalTime()*100.0
283

```

```
284     <<endl;
285     //end
286
287     Info<< "End\n" << endl;
288
289     return 0;
290 }
291
292
293     // *****
294 }
```

Código 3: *overPimpleDyMFoamTimed.C*, OF2006