
Trabajo Fin de Máster

APLICACIÓN DEL MÉTODO DE LOS VOLÚMENES FINITOS AL CÁLCULO DEL PARÁMETRO DE MASA
AÑADIDA DE UN CILINDRO VIBRANTE EN EL SENO DE UN FLUIDO

LAS PALMAS DE GRAN CANARIA, JULIO DE 2015

ALUMNO

D. HÉCTOR RUBÉN DÍAZ OJEDA

TUTOR

DR. D. LUIS ALBERTO PADRÓN HERNÁNDEZ
Universidad de Las Palmas de Gran Canaria
España



SIANI

2014-2015

MÁSTER UNIVERSITARIO EN SISTEMAS INTELIGENTES Y APLICACIONES NUMÉRICAS
EN INGENIERÍA

Agradecimientos

No quería dejar pasar esta oportunidad para recordar y agradecer a todas aquellas personas que han hecho posible que haya llegado hasta aquí y que me instarán a continuar.

Entre ellas, cabe nombrar a mis padres y hermana por el apoyo y sacrificio que realizan diariamente para posibilitarme esta valiosa formación que he estado adquiriendo durante estos años y por ser fuente de apoyo tanto en las buenas como en las malas. A mi tutor, D. Luis Alberto Padrón Hernández, primero por a haberme permitido trabajar junto a él y segundo por la formación, consejos y ayuda que desde el sosiego me ha transmitido siendo de gran valor para mí.

Finalmente me gustaría agradecer a los compañeros del Máster SIANI por su comportamiento, amabilidad y unión durante este año. Aunque, especialmente señalar a D. Jorge Ignacio Pirez Mora por el gran apoyo y por la motivación que me transmite diariamente llegando a ser una de las personas que más apoyo me han dado para acabar este trabajo. Y no dejando escapar la oportunidad de agradecer enormemente también a los compañeros Ingenieros Navales Pedro Gallego Martínez y Tanausú Almeida Medina por su gran interés, ayuda y ganas de progresar que hacen que mi interés por continuar trabajando duro aumente.

Gracias.

Resumen

Enmarcado dentro del campo de la Mecánica de Fluidos y desarrollado en conjunto con la división de Mecánica de los Medios Continuos y Estructuras del Instituto Universitario SIANI, se desarrolla un trabajo fin de título correspondiente al máster en Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería.

En este trabajo académico se propone estudiar la interacción fluido-estructura para el caso de estructuras infinitamente rígidas (2D), vibrando en contacto con un flujo viscoso e incompresible. Para la discretización y resolución de las ecuaciones Navier–Stokes se hará uso del Método de Volúmenes Finitos que se encuentra implementado en la herramienta de código abierto OpenFoam. Más concretamente se realizará la comparativa del coeficiente de masa añadida y del coeficiente de arrastre para una estructura cilíndrica que vibra según un movimiento armónico en un fluido inicialmente en reposo con los resultados de problemas clásicos de los que existen diversas publicaciones. No obstante, para poder desarrollar y obtener unos resultados válidos, se realizará un tratamiento adecuado de la malla de estudio.

En conclusión, con los resultados que se obtienen en este trabajo, se estima conseguir la base metodológica para posteriormente pasar a calcular no solo geometrías más complicadas sino también geometrías en 3D.

Abstract

Belonging to the Fluid Mechanic field and developed together with the Continuum Mechanics and Structures group at the University Institute SIANI, a Master Thesis is developed corresponding to the Intelligent Systems and Numerical Applications in Engineering master.

In this academic work, we do intend to study the fluid – structure interaction for indefinitely rigid structures (2D), vibrating in contact with a non-compressible and viscous flow. For the discretization and the solution of Navier–Stokes' equations we will make use of the Finite Volume Method which is implemented in the open source tool OpenFoam.

Concretely, a comparative will be done between the added mass coefficient and drag coefficient for a cylinder which will be oscillating in vacuum under a harmonic movement with classic problems about which several research papers have been published. Nonetheless, in order to develop and get reliable results, we had better make an accuracy process to the mesh we are going to deal with.

To sum up, with the results that are supposed to be getting, it is deemed that we have already got the methodological basis to subsequently go in not only with more complicated geometries but also with 3D geometries.

Índice general

Lista de figuras	5
Lista de tablas	7
1. Introducción	9
1.1. Motivación	9
1.2. Antecedentes	10
2. Conceptos básicos de Dinámica de Fluidos	13
2.1. Introducción	13
2.2. Las ecuaciones de Navier - Stokes	14
2.2.1. Ecuación de conservación de masa	14
2.2.2. Ecuación de la conservación de momento	15
2.3. Modelos de flujo simplificados	15
2.3.1. Flujo incompresible	16
2.3.2. Flujo no viscoso	17
2.3.3. Flujo potencial y flujo ideal	17
2.4. Turbulencia	18
2.4.1. Aproximaciones numéricas para el tratamiento de la turbulencia . .	19
2.4.2. Modelo k-épsilon	20

2.5. Resolución y discretización de las ecuaciones de la dinámica de fluidos (CFD)	20
3. Metodología	23
3.1. Introducción	23
3.2. El Método de los Volúmenes Finitos	23
3.2.1. Conceptos generales	23
3.2.2. Discretización espacial	25
3.2.3. Definiciones generales de la metodología numérica	26
3.3. Herramienta numérica: OpenFoam	27
3.3.1. Definición de OpenFoam	27
3.3.2. Capacidades de OpenFoam	28
3.3.3. Casos	29
3.4. Simulación en OpenFoam	30
3.4.1. Generación de mallas mediante BlockMesh	30
3.4.2. Elección del solver	41
3.4.3. Post-proceso	42
3.5. Concepto de masa añadida	43
3.6. Ecuación de Morison	44
3.6.1. Método de Morison	45
3.6.2. Cálculo del ángulo δ mediante la transformada rápida de Fourier	46
4. Resultados	49
4.1. Introducción	49
4.2. Solución numérica del problema del cilindro oscilante	49
4.2.1. Estructura de la malla utilizada para los estudios numéricos	50
4.2.2. Condiciones de contorno	51
4.2.3. Mallados	52

4.3.	Números adimensionales	55
4.4.	Solución analítica para un cilindro en el seno de un fluido ideal	56
4.4.1.	Consideraciones iniciales	56
4.4.2.	Cilindro oscilante en el seno de un fluido inicialmente en reposo	60
4.4.3.	Cilindro rígido en reposo en el seno de un fluido oscilante	61
4.4.4.	Comentarios	62
4.5.	Solución numérica del cilindro oscilante	62
4.5.1.	Definición del problema	62
4.5.2.	Resultados	63
4.5.3.	Análisis de resultados	65
5.	Conclusiones y trabajo futuro	69
5.1.	Resumen	69
5.2.	Trabajo futuro	69
A.	Utilidades en OpenFoam	73
A.1.	Posición	73
A.2.	Velocidad	75
A.3.	Presión	76
A.4.	controlDict	78
A.5.	fvSchemes	80
A.6.	fvSolution	82
A.7.	dynamicMeshDict	85
A.8.	transportProperties	86
A.9.	turbulenceProperties	86

Índice de figuras

1.1. Ejemplo real de estructuras cilindricas en plataforma Offshore.	11
2.1. Pasos para resolver numéricamente un problema con CFD. Cardeza [12]. . .	21
3.1. Ejemplo de malla unidimensional.[5]	24
3.2. Ejemplo de malla bidimensional.[5]	25
3.3. Ejemplo de mallas con punto de control en el centro o en los vértices de los elementos.[18]	26
3.4. Esquema general de OpenFoam. [19]	27
3.5. Estructura carpetas de un caso en OpenFoam.[19]	29
3.6. Ejemplos de directorio antes y después de llevar a cabo una simulación . . .	30
3.7. Malla creada.	42
3.8. Resultados de la simulación.	43
3.9. Ejemplo de señales (velocidad, aceleración y fuerza) Massie and Journée [8].	45
4.1. Geometría del sistema.	50
4.2. Geometría del sistema.	51
4.3. Malla Gruesa.	53
4.4. Sección de Malla Normal.	53
4.5. Sección de Malla Fina.	54
4.6. Análisis de 6 mallas diferentes (desechables).	54

4.7. Análisis de las 3 mallas finales.	55
4.8. Comparativa de resultados de Cd para $\beta = 23200$	67
4.9. Comparativa de resultados de Cd para $\beta = 48600$	67

Índice de tablas

4.1. Parámetros para $\beta = 23200$	63
4.2. Parámetros para $\beta = 48600$	63
4.3. Coeficientes para régimen laminar y $\beta = 23200$	64
4.4. Coeficientes para régimen laminar y $\beta = 48600$	64
4.5. Coeficientes para régimen turbulento y $\beta = 23200$	64
4.6. Coeficientes para régimen turbulento y $\beta = 48600$	64
4.7. Coeficientes obtenidos por el Método Morison para régimen laminar y $\beta = 23200$	64
4.8. Coeficientes obtenidos por el Método Morison para régimen laminar y $\beta = 48600$	65
4.9. Coeficientes obtenidos por el Método Morison para régimen turbulento y $\beta = 23200$	65
4.10. Coeficientes obtenidos por el Método Morison para régimen turbulento y $\beta = 48600$	65

Capítulo 1

Introducción

1.1. Motivación

Dentro del área de la mecánica de fluidos, surgen numerosos problemas como la tarea de estimar las cargas tanto de corrientes como de oleaje en el diseño de las plataformas petrolíferas (Figura 1.1), molinos eólicos *Offshore* y otras estructuras relacionadas. Esto, en otras palabras, tiene como partida la determinación del coeficiente de masa añadida y del coeficiente de arrastre siendo esto en lo que se centrará el presente trabajo.

Dicho trabajo se ha dividido en seis capítulos que se exponen a continuación:

- Introducción.
- Conceptos básicos de Dinámica de Fluidos.
- Metodología.
- Resultados.
- Conclusiones.
- Referencias.

En los capítulos mencionados se comentará lo relativo al problema que se ha estudiado. Dicho problema consiste en la interacción fluido-estructura en el caso de estructuras infinitamente rígidas vibrando en contacto con un flujo viscoso e incompresible, donde validar una metodología basada en el Método de los Volúmenes Finitos para el cálculo de los parámetros

masa añadida y coeficiente de arrastre para el caso bidimensional será el principal objetivo. Esta validación se realizará a través de la comparación de resultados con un problema clásico para el que existen diversos resultados publicados y haciendo uso de la herramienta de código abierto OpenFoam. Esto, es de suma importancia ya que se deberá ser capaz de diseñar estructuras que puedan operar en zonas con condiciones adversas y que han sido de acceso imposible en las últimas décadas, además de tener la intención de aumentar la vida útil de los materiales no solo para disminuir el aspecto económico sino también para aumentar la seguridad de operación de dichas estructuras. Siendo aquí donde los parámetros comentados entran en juego ya que son entre otros los que permiten modelar la influencia del fluido en contacto con una estructura en vibración.

La consecución del objetivo arriba expuesto permitirá, en un paso posterior, utilizar la misma metodología para el cálculo de estos parámetros, primero en problemas de carácter tridimensional con geometrías sencillas (por ejemplo de cilindros de longitud reducida), y posteriormente de geometrías complicadas (como pueda ser el caso de estructuras tipo Jacket para estructuras Offshore).

1.2. Antecedentes

Indudablemente el uso y la importancia del CFD viene dado principalmente por el gran desarrollo que la computación ha experimentado desde 1950. Esto es así ya que, hasta que este desarrollo no se produjo, era inviable el uso de estas técnicas que requieren el cálculo de gran número de operaciones por unidad de tiempo. Estas técnicas permiten que se lleve a cabo en este trabajo final de máster necesario para la obtención del título de máster en *Sistemas Inteligentes y Aplicaciones Numéricas en la Ingeniería* una serie de simulaciones que indicarán los valores de los coeficientes comentados en el apartado anterior.

Por otro lado, en este trabajo se ha seguido una extensa y variada bibliografía que va desde textos elementales de Mecánica de Fluidos hasta textos más específicos relacionados con la materia que nos involucra (Massie and Journée [8], White [17].) Todo esto pasando por libros como los de Ferziger and Perić [3], Versteeg and Malalasekera [11] o Fernandez Oro [13] que hacen un recorrido desde los conceptos de la mecánica de fluidos hasta los métodos numéricos aplicados para resolver diferentes problemas como el que se ha propuesto en este trabajo fin de máster.

Sin lugar a dudas debemos destacar tres textos base en lo relacionado con la aplicación que será desarrollada en este trabajo: Chen [2], Wilson [4] y Sarpkaya [15]. Estos son de alguna manera los textos más conocidos y más referenciados por diversas publicaciones científicas, ya que básicamente contienen todo lo elemental sobre aplicaciones y problemas



Figura 1.1: Ejemplo real de estructuras cilíndricas en plataforma Offshore.

específicos del sector *Offshore*.

Finalmente, es de destacar que existe una amplia literatura e investigaciones sobre el estudio de la interacción fluido–estructura (Chaplin [1], Dütsch et al. [6], Gadelho et.al. [7], Bonfiglio et.al. [10] y Keulegan and Carpenter [9]), artículos que se centran en el cálculo de los coeficientes hidrodinámicos para diversas consideraciones de la estructura como por ejemplo para una estructura oscilatoria propuesta por Bonfiglio et.al. [10]. Aunque nuestro trabajo tendrá una estrecha relación con Sarpkaya [14] y sobre todo con Troesch and Kim [16]. Esta última publicación muestra unos resultados experimentales obtenidos para C_m y C_d los cuales serán de gran interés a la hora de validar la metodología que aquí se expone.

Capítulo 2

Conceptos básicos de Dinámica de Fluidos

2.1. Introducción

A la hora de analizar los problemas de mecánica, existen dos posibilidades:

1. El enfoque lagrangiano donde la solución buscada es la evolución temporal de las magnitudes de todas y cada una de las partículas de sistema que intervienen en el problema. Matemáticamente las magnitudes las consideraremos funciones de la posición inicial de la partícula y del tiempo (usado en mecánica de sólidos).
2. El enfoque euleriano en el que se calculan los valores las magnitudes de todas las partículas del sistema que en cada instante t están ocupando un volumen de control fijado previamente. Matemáticamente las magnitudes que se analizan son funciones de la posición en el volumen de control y en el tiempo (usado en mecánica de fluidos).

En este trabajo, se hace uso de la segunda posibilidad. Es por ello, que en este capítulo se introducirán de una forma somera las ecuaciones pertinentes de los fluidos con la única finalidad de poner en contexto y dar la base elemental al lector para que se pueda lograr un mayor entendimiento del trabajo que será desarrollado posteriormente.

2.2. Las ecuaciones de Navier - Stokes

Si consideramos una variable específica ϕ definida sobre un volumen de control (VC) y calculamos la variación temporal de la variable en dicho volumen de control, teniendo en cuenta que existen dos mecanismos fundamentales responsables de la generación de un flujo, como son la difusión, originado a nivel molecular, microscópico, y la convección, asociado al movimiento del fluido a nivel macroscópico, es posible llegar a la ecuación 2.1:

$$\underbrace{\frac{\partial(\rho\phi)}{\partial t}}_{\text{Temporal}} + \underbrace{\nabla \cdot (\rho\vec{v}\phi)}_{\text{Convectivo}} = \underbrace{\nabla \cdot (\Gamma\nabla\phi)}_{\text{Difusivo}} + \underbrace{S}_{\text{Fuente}} \quad (2.1)$$

Por tanto y como se define en la ecuación 2.1 aparecen cuatro términos:

- Término **temporal**, representa la variación local con el tiempo en el interior del volumen de control; es decir, la acumulación o disminución de ϕ .
- Término **convectivo**, que representa el transporte de la variable de un punto a otro del dominio por medio de la velocidad de flujo.
- Término **difusivo**, se corresponde con alguno de los fenómenos de transporte que ocurren a nivel molecular: la ley de Fourier para la difusión de calor; la ley de Fick para la difusión de masa o la ley de Newton para la difusión de cantidad de movimiento por efectos viscosos.
- Término **fuentes**, para tener en cuenta fuentes de generación o destrucción de la variable transportada.

Finalmente cabe señalar que las ecuaciones de gobierno para el flujo y la transferencia de calor y de masa, así como aquellas que rigen el transporte de otras variables se pueden enunciar de forma conservativa como en la ecuación 2.1. En cada caso, los coeficientes de transporte, así como la variable específica que sustituye a ϕ , adoptan formas diferentes.

2.2.1. Ecuación de conservación de masa

La ecuación de conservación de masa, o ecuación de continuidad, establece de forma general que el incremento de masa en el interior de un elemento fluido es consecuencia del flujo neto de masa hacia dicho elemento. Puesto que en general no puede crearse ni

destruirse masa (excepto en casos donde intervengan reacciones nucleares), la expresión no estacionaria, tridimensional, en un punto para un fluido general es:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \quad (2.2)$$

En el caso particular de flujo incompresible la densidad del fluido no varía temporal ni espacialmente en el dominio, por lo que la ecuación se reduce a que la divergencia de la velocidad debe ser nula.

$$\nabla \cdot \vec{v} = 0 \quad (2.3)$$

2.2.2. Ecuación de la conservación de momento

La expresión general de la ecuación de movimiento para un fluido newtoniano (por tanto, que presenta una relación lineal entre las tensiones cortantes y deformaciones resultantes según $\tau = \mu \frac{\partial u}{\partial y}$) e incompresible es (White [17]):

$$\rho \frac{\partial \vec{v}}{\partial t} + \rho(\vec{v} \cdot \nabla)\vec{v} = -\nabla p + \rho \vec{g} + \nabla(\mu \nabla \vec{v}) \quad (2.4)$$

Aprovechándose de las propiedades de los campos vectoriales, es relativamente fácil reformular el término convectivo según:

$$(\vec{v} \cdot \nabla)\vec{v} \equiv \vec{v} \cdot \nabla \vec{v} = \nabla \cdot (\vec{v} \cdot \vec{v}) - \vec{v} \cdot (\nabla \cdot \vec{v}) \quad (2.5)$$

Y teniendo en cuenta que al ser incompresible el flujo es adivergente, se obtiene directamente sustituyendo 2.5 en 2.4: que es la ecuación en su forma conservativa.

$$\frac{\partial(\rho \vec{v})}{\partial t} + \nabla(\rho \vec{v} \vec{v}) = \nabla(\mu \nabla \vec{v}) - \nabla p + \rho \vec{g} \quad (2.6)$$

Se obtendría una expresión similar para el caso compresible pero para el caso que se estudia aquí no procede enunciarla.

2.3. Modelos de flujo simplificados

Aunque no sea objeto de este trabajo analizar todas las posibles simplificaciones de las ecuaciones de gobierno, sí es conveniente repasarlas brevemente con el objeto de presentar las técnicas numéricas más apropiadas para resolver cada una de ellas.

En primer lugar, se puede decir que el comportamiento dinámico de fluidos está gobernado por las ecuaciones de conservación de la masa, la cantidad de movimiento y la energía.

En el caso más general, las ecuaciones completas de Navier-Stokes presentan las siguientes características:

- Son dependientes del tiempo.
- Tridimensionales.
- Incluye esfuerzos viscosos.
- Considera la compresibilidad del fluido.
- Son ecuaciones en derivadas parciales elípticas y altamente no lineales.
- Se encuentran acopladas entre sí. Sin embargo, en la mayoría de los casos es posible reducir el análisis, simplificando las ecuaciones mediante:
 - Imponiendo unas condiciones de contorno que no varíen con el tiempo (estacionarios).
 - La reducción de dimensiones (problemas estacionarios, bidimensionales o axisimétricos).
 - Aproximando el comportamiento del fluido (incompresible, no viscoso).
 - Simplificando las fuerzas que actúan sobre el fluido (hidrostático, potencial).
 - Aproximaciones basadas en valores medios (Reynolds medio, profundidad media (*depth-averaged*)).

2.3.1. Flujo incompresible

Los fluidos, en general, son todos compresibles en mayor o menor grado. Sin embargo, las variaciones de densidad motivadas por cambios de presión pueden ser despreciados si:

- El número de Mach, $M_a = \frac{u}{c} < 1$. Donde:
 - u = velocidad del fluido.
 - c = velocidad del sonido.
- Las variaciones de temperatura son pequeñas.

Cabe resaltar que el término incompresible, no significa que la densidad sea constante; de hecho, existen un gran número de casos donde la densidad es la responsable del movimiento de los fluidos; como por ejemplo: las variaciones de salinidad en los océanos o las variaciones de temperatura en la atmósfera. Las consecuencias numéricas de los flujos incompresibles son:

- La energía interna es irrelevante (las ecuaciones no están acopladas).
- La conservación de la masa lleva implícita una ecuación para la presión.
- Solución mediante un método basado en la presión.

Con lo dicho con anterioridad, podremos concluir diciendo que en condiciones de flujo isoterma, la hipótesis de incompresibilidad implica que la solución de las variables primitivas de flujo (presión y velocidad) se haga independiente del campo de temperaturas. El sistema de ecuaciones requerido queda reducido a la ecuación de continuidad adivergente y a la ecuación de cantidad de movimiento, que expresadas vectorialmente establecen:

$$\nabla \cdot \vec{v} = 0$$

$$\rho \frac{\partial \vec{v}}{\partial t} + \rho(\vec{v} \cdot \nabla)\vec{v} = \rho \vec{f} - \nabla p + \mu \nabla^2 \vec{v} \quad (2.7)$$

Nota: la ecuación 2.7 corresponde a un flujo incompresible viscoso.

2.3.2. Flujo no viscoso

Todos los fluidos reales poseen viscosidad. Sin embargo, muchas veces se comportan como si no la tuvieran. Por esta razón, resulta útil conocer la dinámica de un fluido ideal que se supone incompresible y carente de viscosidad. El análisis del movimiento de un fluido ideal es más simple que el de un flujo viscoso debido a que no existen esfuerzos cortantes. Los esfuerzos normales son los únicos que se deben considerar en dicho análisis. Esto deja la ecuación como la ecuación 2.8 como la correspondiente para este caso y que se conoce como la ecuación de Euler:

$$\rho \vec{g} - \vec{\nabla} p = \rho \frac{d\vec{v}}{dt} \quad (2.8)$$

2.3.3. Flujo potencial y flujo ideal

En el caso de flujos con números de Reynolds suficientemente altos, despreciar los efectos viscosos y de conducción resulta una aproximación bastante cómoda, pues elimina los términos difusivos de segundo orden en las ecuaciones, convirtiéndolas en ecuaciones de primer orden.

Manteniendo la condición de flujo subsónico, al igualar la viscosidad a cero, se obtienen las ecuaciones de Euler, en las que todo el transporte de cantidad de movimiento se debe a fenómenos macroscópicos. Esto es:

$$\nabla \cdot \vec{v} = 0$$

$$\rho \left(\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} \right) = \rho \vec{f} - \nabla p \quad (2.9)$$

En el flujo estacionario, se cumple que:

$$\nabla \left(\frac{v^2}{2} + \frac{p}{\rho} + gz \right) = 0 \quad (2.10)$$

por lo que H no varía a lo largo de una línea de corriente. Si el flujo es irrotacional se cumple además que $\nabla \times \vec{v} = 0$, por lo que H se hace constante en todo el dominio del flujo. En estas condiciones, denominadas flujo potencial, se puede definir un potencial de velocidad, $\vec{v} = \nabla \phi$, que introducido en la ecuación de continuidad proporciona la ecuación de Laplace. Esta ecuación se resuelve con la condición de componente normal nula en la superficie de un cuerpo sumergido y con la especificación de la velocidad en los puntos lejanos al cuerpo. Una vez obtenido el campo de velocidades, la ecuación de Euler proporciona el campo de presiones. La ecuación de Laplace es lineal y posee soluciones exactas simples que pueden superponerse para crear nuevas soluciones.

Cuando la velocidad no se puede expresar a partir de un potencial, el flujo ideal se suele resolver introduciendo la vorticidad ($\nabla \times \vec{v}$), que ya no es nula, en las ecuaciones de cantidad de movimiento y energía.

2.4. Turbulencia

La turbulencia es un estado caótico e irregular del movimiento de un fluido que se establece a partir de la aparición de irregularidades en las condiciones iniciales o de contorno de la corriente fluida. Estas inestabilidades se amplifican y se retroalimentan de forma cíclica, creando vórtices (*eddies*) turbulentos que se crean y se destruyen.

La turbulencia es una característica de los flujos, no de los fluidos como tales, sino que su aparición exige de la existencia de un fluido en movimiento, en el que los fenómenos de convección (inerciales) asociados a la velocidad sean varios órdenes de magnitud superiores a los efectos difusivos (disipativos) relacionados con la viscosidad del fluido. Esta relación es

el conocido número de Reynolds que establece la frontera (aproximada) entre las condiciones de flujo laminar y flujo turbulento.

Finalmente podremos resumir que las propiedades más destacadas de la turbulencia son:

- Aleatoriedad.
- Vorticidad.
- Difusividad (mayor transporte).
- Tridimensionalidad.
- Disipación.
- Altos números de Reynolds.

2.4.1. Aproximaciones numéricas para el tratamiento de la turbulencia

La solución numérica para flujos turbulentos puede abordarse desde distintos niveles de aproximación, proporcionando así descripciones del flujo con mayor o menor detalle. Esto se consigue en función del número de escalas de la turbulencia que se quieran resolver en la simulación.

En general, se distinguen tres aproximaciones diferentes:

- **DNS:** *Direct numerical simulation:* enfoque en el que todas las escalas deben ser resueltas según las ecuaciones de Navier-Stokes. Eso implica discretizar con celdas suficientemente pequeñas para captar las microescalas.
- **LES:** *Large-Eddy simulation:* son técnicas en las cuales los torbellinos y vórtices de tamaños grandes y medios se simulan directamente (con mallas adecuadas), mientras los más pequeños, donde tiene lugar la disipación de energía, se modelan de alguna forma.
- **RANS:** *Reynolds-Averaged Navier-Stokes:* se usa la idea de las ecuaciones de Navier – Stokes promediadas por Reynolds en la que todas las escalas se modelizan mediante el uso de modelos de turbulencia.

2.4.2. Modelo k-épsilon

El método simple más completo para simular la turbulencia y uno de los modelos RANS isotrópicos más utilizados a nivel industrial es el modelo k-épsilon.

El k-épsilon estándar es un modelo semiempírico que se basa en la modelización de dos ecuaciones de transporte: uno para la energía cinética turbulenta k y otro para su tasa de disipación ε . La ecuación de transporte para k se obtiene analíticamente a partir de las ecuaciones de Navier-Stokes, mientras que la de ε se obtiene a partir de razonamientos físicos y analogías con la ecuación de k .

El cálculo de las condiciones de contorno del mismo vienen dado por:

$$k = \frac{3}{2}(Tu\bar{U})^2 \quad (2.11)$$

$$\varepsilon = C_\mu^{(\frac{3}{2})} \frac{k^{(\frac{3}{2})}}{L} \quad (2.12)$$

Donde:

- $L \Rightarrow$ longitud característica del flujo $\sim 0,07l$. Donde l es la dimensión característica del problema que en este caso es el diámetro del cilindro.
- $Tu \Rightarrow$ intensidad turbulencia.
- \bar{U} velocidad media del flujo.
- $C_\mu \Rightarrow 0.09$.

2.5. Resolución y discretización de las ecuaciones de la dinámica de fluidos (CFD)

El punto de inicio de todo método numérico es el modelo matemático del fenómeno físico que se desea estudiar y que generalmente suele ser expresado en forma de ecuaciones diferenciales en derivadas parciales o ecuaciones integro-diferenciales junto con las condiciones de contorno. En el caso de la dinámica de fluidos computacional se utilizan las ecuaciones de Navier-Stokes o simplificaciones de las mismas dependiendo de la aplicación tal y como se ha comentado.

2.5. RESOLUCIÓN Y DISCRETIZACIÓN DE LAS ECUACIONES DE LA DINÁMICA DE FLUIDOS (CFD)21

Cabe señalar que el ordenador es una máquina finita y no puede manejar ecuaciones en derivadas parciales con variables continuas en el espacio y el tiempo. Por ello, una vez definido el problema matemático que se quiere resolver, se procede a realizar la discretización temporal y espacial, transformando las ecuaciones diferenciales en algebraicas. La solución que obtenemos no será continua sino que vendrá dada por una serie discreta de valores tanto en el espacio como en el tiempo.

Queda por tanto mencionar que, tal y como ya se comentó, existen numerosas metodologías que nos permiten discretizar nuestro problema como el método de los elementos finitos, método de los volúmenes finitos, etc.

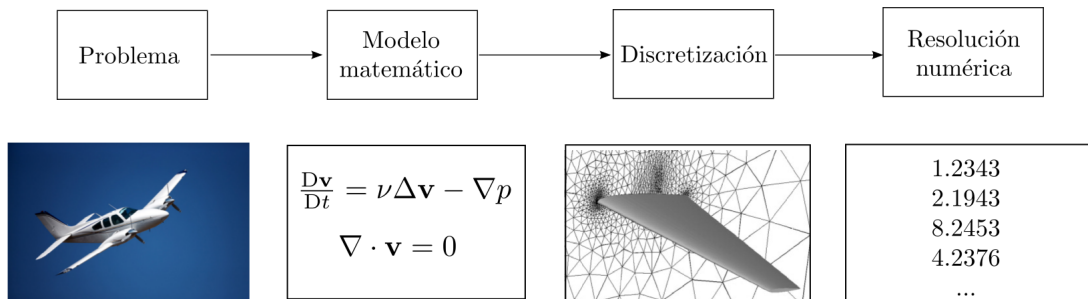


Figura 2.1: Pasos para resolver numéricamente un problema con CFD. Cardeza [12].

Capítulo 3

Metodología

3.1. Introducción

En este capítulo se describe el método de aproximación que será utilizado. Además también se introducirán los conocimientos básicos necesarios a cerca de la herramienta que nos permitirá resolver nuestro problema, OpenFoam.

3.2. El Método de los Volúmenes Finitos

3.2.1. Conceptos generales

El objetivo del método de los volúmenes finitos es desarrollar una metodología numérica para resolver la ecuación general de transporte. La idea de partida fundamental es el concepto de discretización: reemplazar una solución analítica en derivadas parciales que proporciona el valor de ϕ de forma continua en todos los puntos del espacio por una solución numérica aproximada que da el valor de ϕ únicamente en una serie de puntos discretos por la malla que define el dominio.

Este método, volúmenes finitos, propone una forma de llevar a cabo esa discretización. En particular, establece que los valores discretos de ϕ quedarán descritos por un conjunto de ecuaciones algebraicas que relacionan el valor de la variable en un punto con el valor en los puntos vecinos. La forma en que se transmite la información entre esos nodos requiere de algún tipo de aproximación, que en el caso de volúmenes finitos, es mediante esquemas conservativos que evalúan los flujos a través de superficies de control.

La transformación de las ecuaciones diferenciales en un conjunto de ecuaciones algebraicas precisa necesariamente de una discretización espacial. Esto se consigue generando una malla que permite dividir el dominio de interés en una serie de celdas a las cuales se les asocia el valor de la variable ϕ .

Para ver con mayor claridad este método se realizará una definición de los pasos que se tienen que dar para resolver por este método:

1. Construir una malla.

La malla puede ser en 1, 2 o 3 dimensiones, y se basa en situar al nodo de la discretización en el centro de un segmento (1 dimensión), en el centro de un rectángulo (2 dimensiones), o en el centro de un cubo (3 dimensiones). Este segmento, rectángulo o cubo, representa un volumen de control.

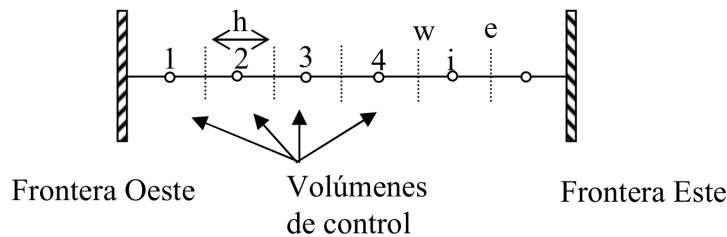


Figura 3.1: Ejemplo de malla unidimensional.[5]

En la figura 3.1 se observa que se han marcado una serie de puntos numerados correlativamente (nodos), y el volumen de control estaría constituido por el segmento que va desde el punto medio anterior al punto medio posterior. Se suele emplear la notación de los puntos cardinales para referirse a una frontera determinada del volumen de control. Así el volumen de control i tendría una frontera al oeste (a la izquierda del nodo i) y otra al este (a la derecha del nodo i).

También, se muestra el volumen de control para el punto (i,j) , y sus cuatro fronteras: Norte, sur, este y oeste.

2. Integración.

La idea fundamental, y que lo distingue de las diferencias finitas, es realizar una integración de la ecuación diferencial en el volumen de control.

3. Aplicación de un esquema de diferencias finitas.

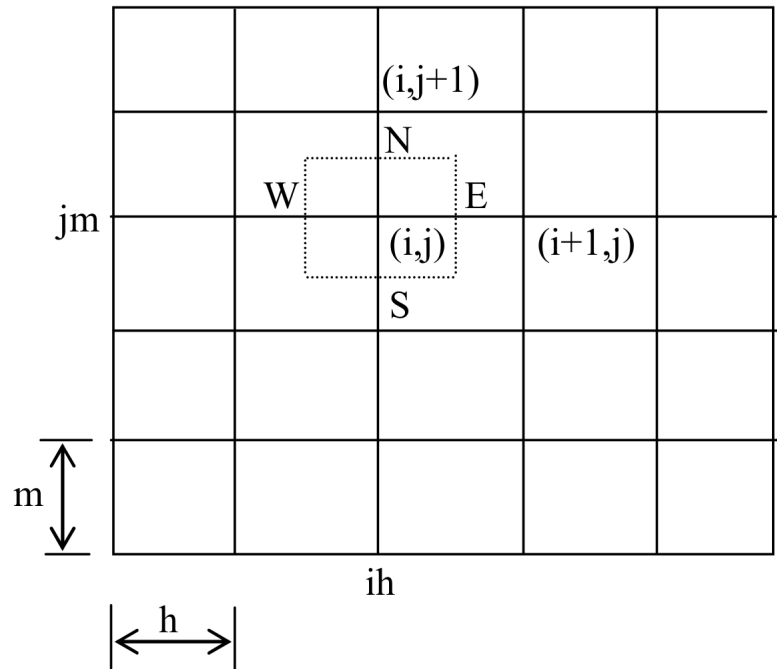


Figura 3.2: Ejemplo de malla bidimensional.[5]

Este esquema se puede aplicar antes o después de integrar. Si la integral es factible de hacer entonces primero se hace la integral, y luego se aplica el esquema de diferencias finitas. Si la integral no es resoluble, se hace el esquema de diferencias finitas dentro del integrando y luego se integra.

4. Construcción de una matriz global.

El procedimiento anterior conduce a un sistema de ecuaciones, que una vez resuelto proporciona la solución buscada.

3.2.2. Discretización espacial

Básicamente, existen dos tipos de mallado:

1. Mallados estructurados: cada punto de la malla está inequívocamente identificado por

los índices i, j, k , en coordenadas cartesianas. Las celdas de la malla son cuadriláteros en 2-D, y hexaedros en 3-D.

2. Mallados no estructurados: las celdas y los nodos de la malla no tienen un orden particular, es decir, los celdas o nodos cercanos a uno dado, no pueden identificarse directamente por sus índices. Los elementos de la malla, en este caso, son una mezcla de cuadriláteros y triángulos en 2-D y tetraedros y hexaedros en 3-D.

El tipo de discretización espacial a emplear, depende del tipo de discretización de las ecuaciones empleado, así como de la estructura interna de datos empleados para resolver el flujo. El método de las diferencias finitas, que discretiza las ecuaciones de Navier-Stokes en forma diferencial, requiere una malla estructurada de puntos en los que se guardan las variables de flujo. Por otro lado, en el método de los volúmenes finitos es necesario definir puntos de control en cada volumen generado. Precisamente es, en esos vértices, donde se guardan las variables, que dependiendo del método que se emplee pueden ser definidos en el centro del elemento (*cell-centred*) o en los vértices (*cell-vertex*) de las celdas.

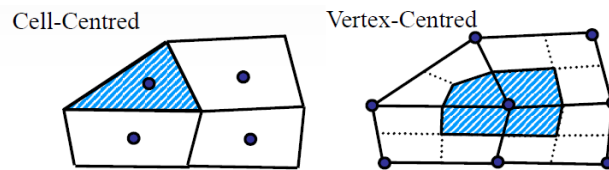


Figura 3.3: Ejemplo de mallas con punto de control en el centro o en los vértices de los elementos.[18]

3.2.3. Definiciones generales de la metodología numérica

La forma general de una ley de conservación para una cantidad escalar U , con fuentes volumétricas Q , sobre un volumen finito que incorpora flujos por las caras del volumen de control viene dada por:

$$\frac{\partial}{\partial t} \int_{\Omega} U d\Omega + \oint_S \vec{J} \cdot d\vec{S} = \int_{\Omega} Q d\Omega \quad (3.1)$$

Usando la ecuación general de transporte en su formulación integral que ya había sido definida

$$\frac{\partial}{\partial t} \int_V \rho \phi dV + \oint_A (\rho \vec{v} \phi - \Gamma \nabla \phi) \cdot d\vec{A} = \int_V S_{\phi} dV \quad (3.2)$$

Basta con identificar que $U = \rho \phi$, $\vec{J} = \rho \vec{v} \phi - \Gamma \nabla \phi$ y $Q = S_\phi$, así como que el volumen se representa por Ω en lugar de V y el área por S en vez de A , para entender que estamos ante la misma expresión. Esta nueva forma de expresar la ecuación de transporte es muy compacta e idéntica de forma matemática al sentido de los términos involucrados: término temporal (*unsteady*), término de flujo por las superficies de control (*fluxes*) y término fuente.

3.3. Herramienta numérica: OpenFoam

3.3.1. Definición de OpenFoam

OpenFoam, es un paquete de software de libre divulgación que utiliza diferentes métodos de discretización para resolver ecuaciones en derivadas parciales. Este es principalmente usado en dinámica de fluidos aunque abarca otros campos (se especificará mejor en el capítulo 3.3.2). Se trata de una librería C++ a partir de la cual se crean ejecutables, conocidos como aplicaciones.

Existen dos tipos de aplicaciones: *solvers* y utilidades. Los *solvers* se ocupan de resolver las ecuaciones que gobiernan un fenómeno físico. Las utilidades son aplicaciones que se utilizan antes o después de haber empleado el *solvers* y que básicamente son aplicaciones que nos permiten manipular datos y ajustarlos al problema que se está tratando y que será tratado por el *solver*. Un esquema de la organización global de OpenFoam se muestra en la figura.

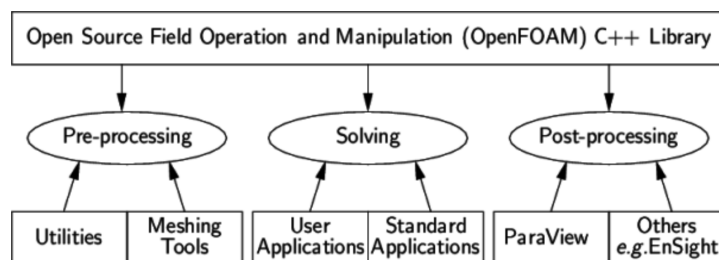


Figura 3.4: Esquema general de OpenFoam. [19]

Queda claro que los pasos para usar OpenFoam vienen dados por la definición del problema, pre-procesar haciendo uso de las utilidades, resolver en función del problema establecido y finalmente post procesar con las diversas utilidades que se puedan necesitar. No obstante,

no siempre se resolverán casos que estén implementados en Openfoam por lo que será necesario la modificación o creación mediante el lenguaje de programación C++ de *solvers* que puedan resolver dichos casos.

3.3.2. Capacidades de OpenFoam

Una de las principales características que desde el diseño inicial se buscado es que sea un sistema abierto. Esta característica junto con sus características de software libre ha posibilitado el desarrollo de numerosos solvers así como herramientas de pre y post procesado para adaptarse a un gran abanico de casos. Entre los distintos solvers pueden destacarse:

- **Fluidos incompresibles.**
 - *boundaryFoam*: Solver estacionario para flujos turbulentos de 1D.
 - *icoDyMFoam*: Flujo laminar no estacionario con movimientos de malla asociados.
 - *icoFoam*: Flujo laminar no estacionario.
 - *nonNewtonianIcoFoam*. Flujo laminar no estacionario para flujos no estacionarios.
 - *turbFoam*: Flujos turbulentos incompresibles no estacionarios.
- **Fluidos compresible.**
 - *coodles*: LES para flujos compresibles.
 - *rhoPimpleFoam*: Flujos turbulentos para ventilación y transferencia de calor.
- **Flujos multifase.**
 - *bubbleFoam*: Dos fluidos incompresibles con una fase dispersa.
 - *lesCavitatingFoam*: Procesos no estacionarios de cavitación con turbulencia mediante LES.
- **Combustiones.**
 - *dieselEngineFoam*: Análisis de procesos de combustión con inyección en diésel spray.
- **Transmisión de calor.**

- *buoyantFoam*: Flujos turbulento para flujos compresibles para fenómenos de ventilación y transmisión. de calor, los efectos gravitacionales son considerados.
 - *buoyantSimpleRadiationFoam*: Flujos turbulentos para fluidos compresibles con fenómenos de radiación.
- **Análisis de fuerzas.**
- *solidDisplacementFoam*: Solver no estacionario para pequeñas deformaciones elásticas, con efectos térmicos.

3.3.3. Casos

En un problema que se quiera resolver es necesario una serie de carpetas y archivos que definan el problema y también que definan como se va a resolver dicho problema. Dichas carpetas son tres carpetas: *constant*, *system* y una carpeta temporal inicial. Esta última no tiene porque corresponder con el tiempo 0, es decir, puede partir en un tiempo posterior al tiempo 0 ya que puede darse el caso que se tengan simulaciones previas y se necesite continuar la simulación.

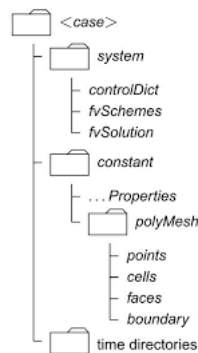


Figura 3.5: Estructura carpetas de un caso en OpenFoam.[19]

La carpeta *constant* contiene una carpeta llamada *polyMesh*, donde se encuentra toda la información acerca de como es la malla que ha creado, además de archivos que son necesarios para el *solver* ya que le proporcionan información sobre propiedades características del problema que se esté tratando.

En *system* están los archivos correspondientes a la resolución. Debe haber como mínimo tres: *controlDict*, *fvSchemes*, y *fvSolution*. El *controlDict* es donde se imponen parámetros que tienen que ver con la ejecución del problema que se esté tratando, esto es por ejemplo el tiempo de comienzo y de finalización, el paso de tiempo o cada cuántas iteraciones se desea guardar el resultado. El *fvSolution* permite determinar como afrontar y como solucionar las ecuaciones de nuestro trabajo. Este tiene algoritmos de resolución de esquemas numéricos como el *Gauss-Seidel*. Por otro lado, el archivo *fvSchemes* es el que se contiene los esquemas numéricos que se usarán para resolver las ecuaciones de nuestro problema.

Finalmente, para empezar la simulación se debe tener una carpeta temporal inicial en la que se establezcan las condiciones iniciales de las diferentes variables implicadas en el problema que se vaya a tratar.



Figura 3.6: Ejemplos de directorio antes y después de llevar a cabo una simulación

3.4. Simulación en OpenFoam

Una vez introducidas las características principales de OpenFoam, se hace necesario destacar los diferentes pasos que se necesitan para hacer una simulación de forma más específica.

3.4.1. Generación de mallas mediante BlockMesh

Lo primero que se hace es crear la geometría y proceder a mallarla. Dado que tratar de crear una malla discreta “a mano” es un trabajo largo y algo complicado, OpenFoam nos

presenta la utilidad **blockMeshDict** que permite realizar dicha tarea.

3.4.1.1. Cabecera

Lo primero que se encuentra al abrir el fichero *blockMesh* es la cabecera. En esta simplemente se observa que el fichero es un fichero OpenFoam y las unidades en las que se trabajarán.

```

/*-----*- C++ -*-----*\
| ===== |
| \\      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
| \\      / O p e r a t i o n | Version: 2.1.1 |
|  \\    / A n d           | Web:      www.OpenFOAM.org |
|   \\  / M a n i p u l a t i o n |
|-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    object       p;
}
// ***** //

convertToMeters 1;

```

3.4.1.2. Definición de vértices

A la hora de definir la geometría se necesitarán las coordenadas espaciales (x , y, z) de los diferentes puntos de la geometría que tienen importancia significativa, es decir, puntos que representan un extremo, un inicio de otra geometría, etc.

```

vertices
(
    //back up

```

```

(0.1015 0 -0.5) //Punto 0
(0.203 0 -0.5) //Punto 1
(0.812 0 -0.5) //Punto 2
(0.812 0.143543 -0.5) //Punto 3
(0.143543 0.143543 -0.5) //Punto 4
(0.071771 0.071771 -0.5) //Punto 5
(0.812 0.812 -0.5) //Punto 6
(0.143543 0.812 -0.5) //Punto 7
(0 0.812 -0.5) //Punto 8
(0 0.203 -0.5) //Punto 9
(0 0.1015 -0.5) //Punto 10
(-0.1015 0 -0.5) //Punto 11
(-0.203 0 -0.5) //Punto 12
(-0.812 0 -0.5) //Punto 13
(-0.812 0.143543 -0.5) //Punto 14
(-0.143543 0.143543 -0.5) //Punto 15
(-0.071771 0.071771 -0.5) //Punto 16
(-0.812 0.812 -0.5) //Punto 17
(-0.143543 0.812 -0.5) //Punto 18

.
.
.
.
.

//front down
(0.812 -0.143543 0.5) //Punto 51
(0.143543 -0.143543 0.5) //Punto 52
(0.071771 -0.071771 0.5) //Punto 53
(0.812 -0.812 0.5) //Punto 54
(0.143543 -0.812 0.5) //Punto 55
(0 -0.812 0.5) //Punto 56
(0 -0.203 0.5) //Punto 57
(0 -0.1015 0.5) //Punto 58
(-0.812 -0.143543 0.5) //Punto 59

```

```

(-0.143543 -0.143543 0.5) //Punto 60
(-0.071771 -0.071771 0.5) //Punto 61
(-0.812 -0.812 0.5) //Punto 62
(-0.143543 -0.812 0.5) //Punto 63

```

```
);
```

3.4.1.3. Definición de bordes

Para geometrías curvas se necesita definir los puntos que intervienen en ella, ya que de no definirse dichos puntos se unirán como si estuvieran en línea entre sí. Para ello definiremos que por esos puntos pasa una línea circular mediante la palabra *arc*, que básicamente lo que hace es indicarle al programa que esos puntos están unidos por una curva.

Para poder llevar a cabo esto necesitaremos tres puntos; el inicial, el final y el intermedio (son los que definen por donde va geometría). El inicial y el final ya se tienen definidos de cuando se definieron los vértices, por lo tanto queda definir el central, haciendo uso otra vez de las coordenadas espaciales.

```

edges
(
  //up
  arc 0 5 (0.093774 0.038842 -0.5)
  arc 5 10 (0.038842 0.093774 -0.5)
  arc 19 24 (0.093774 0.038842 0.5)
  arc 24 29 (0.038842 0.093774 0.5)
  arc 11 16 (-0.093774 0.038842 -0.5)
  arc 16 10 (-0.038842 0.093774 -0.5)
  arc 30 35 (-0.093774 0.038842 0.5)
  arc 35 29 (-0.038842 0.093774 0.5)

```

```

.
.
.
.
.

```



```

arc 1 39 (0.187548 -0.077685 -0.5)
arc 39 44 (0.077685 -0.187548 -0.5)
arc 20 52 (0.187548 -0.077685 0.5)
arc 52 57 (0.077685 -0.187548 0.5)
arc 47 44 (-0.077685 -0.187548 -0.5)
arc 12 47 (-0.187548 -0.077685 -0.5)
arc 31 60 (-0.187548 -0.077685 0.5)
arc 60 57 (-0.077685 -0.187548 0.5)
);

```

3.4.1.4. Definición de bloques

La definición de los bloques consiste en definir una lista de vértices que definen una geometría y un tipo específico de mallado. Para empezar a definir dichos bloques es necesario indicárselo al programa, por ello se usa la palabra *blocks*. Una vez hecho esto se definirán tres cosas:

1. Los vértices que componen el bloque (aunque estemos trabajando geometrías 2D siempre se definen en 3D los bloques).
2. Numero de celdas con la que se procederá a mallar según las coordenadas espaciales.
3. El ratio de expansión de las celdas que básicamente es su orientación en el bloque según las coordenadas espaciales.

```

blocks
//

hex (5 4 9 10 24 23 28 29) (38 24 1) simpleGrading (1 1 1) //0
hex (0 1 4 5 19 20 23 24) (38 24 1) simpleGrading (1 1 1) //1
hex (1 2 3 4 20 21 22 23) (144 24 1) simpleGrading (1 1 1) //2
hex (4 3 6 7 23 22 25 26) (144 144 1) simpleGrading (1 1 1) //3
hex (9 4 7 8 28 23 26 27) (24 144 1) simpleGrading (1 1 1) //4
hex (15 16 10 9 34 35 29 28) (38 24 1) simpleGrading (1 1 1) //5
hex (12 11 16 15 31 30 35 34) (38 24 1) simpleGrading (1 1 1) //6
hex (13 12 15 14 32 31 34 33) (144 24 1) simpleGrading (1 1 1) //7
hex (14 15 18 17 33 34 37 36) (144 144 1) simpleGrading (1 1 1) //8
hex (15 9 8 18 34 28 27 37) (24 144 1) simpleGrading (1 1 1) //9

```

```

//down
hex (40 45 44 39 53 58 57 52) (24 38 1) simpleGrading (1 1 1) //10
hex (0 40 39 1 19 53 52 20) (24 38 1) simpleGrading (1 1 1) //11
hex (1 39 38 2 20 52 51 21) (24 144 1) simpleGrading (1 1 1) //12
hex (39 42 41 38 52 55 54 51) (144 144 1) simpleGrading (1 1 1) //13
hex (44 43 42 39 57 56 55 52) (144 24 1) simpleGrading (1 1 1) //14
hex (47 44 45 48 60 57 58 61) (24 38 1) simpleGrading (1 1 1) //15
hex (12 47 48 11 31 60 61 30) (24 38 1) simpleGrading (1 1 1) //16
hex (13 46 47 12 32 59 60 31) (24 144 1) simpleGrading (1 1 1) //17
hex (46 49 50 47 59 62 63 60) (144 144 1) simpleGrading (1 1 1) //18
hex (47 50 43 44 60 63 56 57) (144 24 1) simpleGrading (1 1 1) //19

);

```

3.4.1.5. Definición del contorno

En las condiciones de contorno lo que se hará es definir las partes de la geometría que por ejemplo será considerada como una pared, una entrada o salida de flujo etc.

```

patches
(
    patch out
    (
        //up
        (2 3 22 21)
        (3 6 25 22)

        //down
        (38 2 21 51)
        (41 38 51 54)
    )
    patch Top
    (

```

```
(7 8 27 26)
(6 7 26 25)
(8 18 37 27)
(18 17 36 37)
)
```

```
patch Down
(
  (43 42 55 56)
  (42 41 54 55)
  (50 43 56 63)
  (49 50 63 62)
)
```

```
patch in
(
  //up
  (14 13 32 33)
  (17 14 33 36)

  //down
  (13 46 59 32)
  (46 49 62 59)
)
```

```
)
wall cylinder
(
  //up
  (10 5 24 29)
  (5 0 19 24)
  (16 10 29 35)
  (11 16 35 30)

  //down
  (40 45 58 53)
  (0 40 53 19)
  (45 48 61 58)
  (48 11 30 61)
)
```

```
)

empty back
(
```

3.4.1.6. Generar la malla ejecutando BlockMesh y corroborar mallado

Ahora quedará generar la malla. Para ello con la terminal del sistema operativo *Linux* dentro de la carpeta donde se encuentran los ficheros ya comentados se pondrá **blockMesh**.

```
hector@hector-G74Sx:~/Escritorio/Simulacionesfinal/
PrimeraRonda/Primera$ blockMesh
```

```
/*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 2.3.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n |
\*-----*/
Build : 2.3.1-bcfaaa7b8660
Exec : blockMesh
Date : Jul 08 2015
Time : 18:13:20
Host : "hector-G74Sx"
PID : 7144
Case : /home/hector/Escritorio/Simulacionesfinal/PrimeraRonda/Primera
nProcs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files
using timeStampMaster
allowSystemOperations : Allowing user-supplied system call operations

// * * * * * //
Create time

Creating block mesh from
```

```
"/home/hector/Escritorio/Simulacionesfinal/PrimeraRonda/  
Primera/constant/polyMesh/blockMeshDict"
```

```
Creating curved edges  
Creating topology blocks  
Creating topology patches
```

```
Reading patches section
```

```
Creating block mesh topology
```

```
Reading physicalType from existing boundary file
```

```
Default patch type set to empty
```

```
Check topology
```

```
Basic statistics  
Number of internal faces : 28  
Number of boundary faces : 64  
Number of defined boundary faces : 64  
Number of undefined boundary faces : 0  
Checking patch -> block consistency
```

```
Creating block offsets  
Creating merge list .
```

```
Creating polyMesh from blockMesh  
Creating patches  
Creating cells  
Creating points with scale 1
```

```
Block 0 cell size :  
  i : 0.00267108 .. 0.00267108  
  j : 0.00332143 .. 0.00332143  
  k : 1 .. 1
```

```
.  
. .  
. .
```

```

      .
      .

Block 19 cell size :
  i : 0.00464206 .. 0.00464206
  j : 0.00664288 .. 0.00664288
  k : 1 .. 1

Writing polyMesh
-----
Mesh Information
-----
  boundingBox: (-0.812 -0.812 -0.5) (0.812 0.812 0.5)
  nPoints: 237312
  nCells: 117888
  nFaces: 472320
  nInternalFaces: 235008
-----
Patches
-----
  patch 0 (start: 235008 size: 336) name: out
  patch 1 (start: 235344 size: 336) name: Top
  patch 2 (start: 235680 size: 336) name: Down
  patch 3 (start: 236016 size: 336) name: in
  patch 4 (start: 236352 size: 192) name: cylinder
  patch 5 (start: 236544 size: 117888) name: back
  patch 6 (start: 354432 size: 117888) name: front

End
```

Se comprueba que ha creado la malla sin errores con el comando **checkMesh**.

```
hector@hector-G74Sx:~/Escritorio/Simulacionesfinal/
PrimeraRonda/Primera$ checkMesh
```

.
.
.
.
.

Mesh stats

points:	237312
internal points:	0
faces:	472320
internal faces:	235008
cells:	117888
faces per cell:	6
boundary patches:	7
point zones:	0
face zones:	0
cell zones:	0

Overall number of cells of each type:

hexahedra:	117888
prisms:	0
wedges:	0
pyramids:	0
tet wedges:	0
tetrahedra:	0
polyhedra:	0

Checking topology...

Boundary definition OK.
Cell to face addressing OK.
Point usage OK.
Upper triangular ordering OK.
Face vertices OK.
Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...

Patch	Faces	Points	Surface topology
-------	-------	--------	------------------

out	336	674	ok (non-closed singly connected)
Top	336	674	ok (non-closed singly connected)
Down	336	674	ok (non-closed singly connected)
in	336	674	ok (non-closed singly connected)
cylinder	192	384	ok (non-closed singly connected)
back	117888	118656	ok (non-closed singly connected)
front	117888	118656	ok (non-closed singly connected)

Checking geometry...

```

Overall domain bounding box (-0.812 -0.812 -0.5) (0.812 0.812 0.5)
Mesh (non-empty, non-wedge) directions (1 1 0)
Mesh (non-empty) directions (1 1 0)
All edges aligned with or perpendicular to non-empty directions.
Boundary openness (9.49811e-17 -7.52451e-17 4.38594e-15) OK.
Max cell openness = 2.8836e-16 OK.
Max aspect ratio = 2.39071 OK.
Minimum face area = 8.98725e-06. Maximum face area = 0.00664288.
  Face area magnitudes OK.
Min volume = 8.98725e-06. Max volume = 2.80826e-05.
Total volume = 2.60502. Cell volumes OK.
Mesh non-orthogonality Max: 43.8649 average: 7.51961
Non-orthogonality check OK.
Face pyramids OK.
Max skewness = 0.488529 OK.
Coupled point location match (average 0) OK.

```

Mesh OK.

End

Finalmente, observamos el resultado en *ParaView* usando el comando **paraFoam**.

3.4.2. Elección del solver

Para poder elegir el *solver* es muy importante tener claro las características del problema en cuestión y como no, que se pretende calcular. Como se ha comentado en el apartado

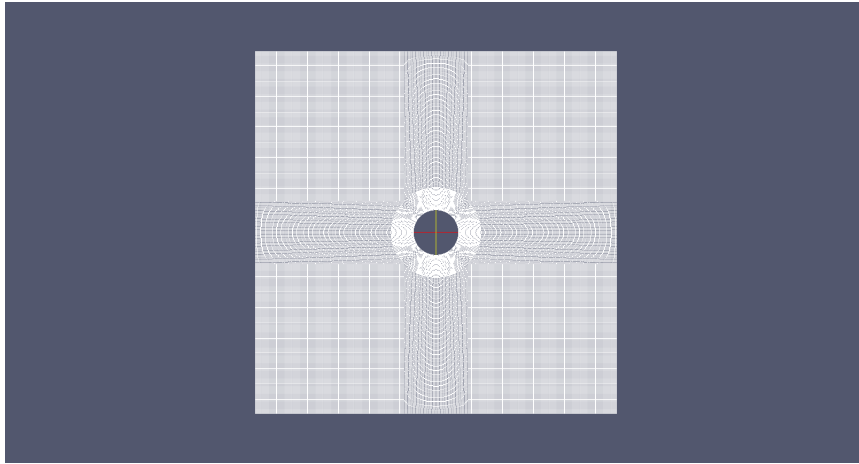


Figura 3.7: Malla creada.

3.3.2 existen numerosos *solvers*. Nosotros hemos elegido el *solver* **pimpleDyMFoam** ya que es un *solver* que permite analizar fluidos newtonianos, incompresibles para una malla en movimiento.

Simplemente se debe indicar que a la hora de utilizar este *solver*, se presenta la similitud a los otros en cuanto a la introducción de características propias del fluido como laminar o turbulento pero que difiere en que en este caso en concreto se necesita que se incluya un diccionario llamado **dynamicMeshDict** en el que se determinará la forma de tratar el movimiento de la malla por parte del *solver*.

3.4.3. Post-proceso

En este punto este punto solo quedará decir que una vez simulado el problema para las características específicas que se tengan, será el momento de analizar los resultados. Para ello se usan programas como **paraView** donde podemos hacer análisis a nuestro sistema.

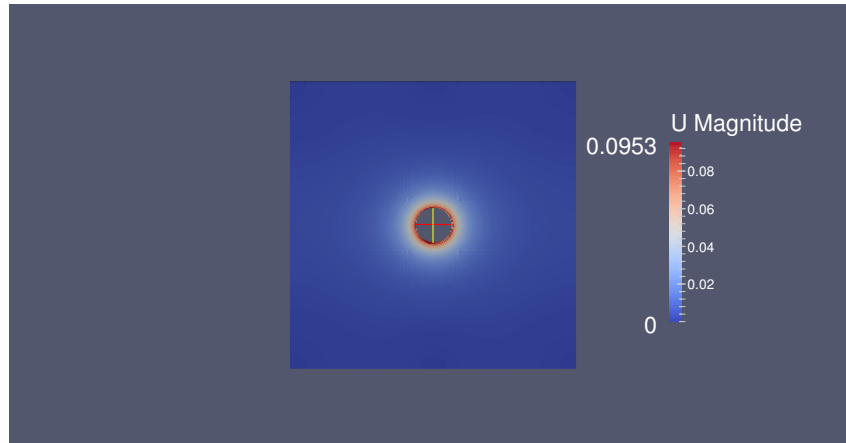


Figura 3.8: Resultados de la simulación.

3.5. Concepto de masa añadida

Entrando en más profundidad en el problema es conveniente introducir el concepto de *masa añadida*, aunque se estudiará con mayor detalle en apartados sucesivos.

Cuando un cuerpo se mueve a través de un fluido, éste debe empujar el fluido fuera de su camino para poder desplazarse. Si el cuerpo sufre una aceleración, el fluido circundante también debe acelerarse. El cuerpo se comporta entonces como si su masa fuera mayor y el incremento aparente de la masa se conoce como masa añadida del fluido.

$$\Sigma F = (m + m_a) \frac{\partial^2 u}{\partial t^2} \quad (3.3)$$

Donde $\partial^2 u / \partial t^2$ corresponde a la aceleración del cuerpo y m_a está referido a la masa añadida. Dicha masa es proporcional a la densidad del fluido ρ y al volumen del cuerpo V , quedando definida como:

$$m_a = \rho V C_m \quad (3.4)$$

Donde C_m es el coeficiente de masa añadida.

3.6. Ecuación de Morison

Experimentos mostrados en Massie and Journée [8] nos dicen que existe una fuerza de arrastre proporcional a la velocidad y al diámetro de la estructura, en este caso al diámetro del cilindro, D por la ecuación 3.5.

$$F_D(t) = \frac{1}{2} \rho C_D D u_0^2 \cdot u(t) |u(t)| \quad (3.5)$$

donde:

- C_D = coeficiente adimensional de arrastre.
- D = diámetro del cilindro.
- u_0 = Amplitud
- $u(t)$ velocidad dependiente del tiempo.

De la ecuación de arrastre 3.5 y de la ecuación de inercia 4.42 o 4.34 (según en el caso que nos encontremos) enunciamos una ecuación semi-empírica conocida como ecuación de **Morison** 3.6.

$$F(t) = \frac{\pi}{4} \rho C_m D^2 \frac{d^2 u}{dt^2} + \frac{1}{2} \rho C_d D u_0^2 \cdot u(t) |u(t)| \quad (3.6)$$

De donde:

$$F_0 e^{i(\omega t + \delta)} = i \frac{1}{2} \rho \omega^2 u_0^2 D C_d + i \frac{\pi}{4} \rho D^2 \omega^2 u_0 C_m \quad (3.7)$$

Obteniendo la expresiones para calcular los coeficientes de arrastre 3.9 y de inercia 3.8.

$$C_m = \frac{4 F_0 \sin \delta}{\pi \rho D^2 \omega^2 u_0} \quad (3.8)$$

$$C_D = \frac{2 F_0 \cos \delta}{\rho \omega^2 u_0^2 D} \quad (3.9)$$

3.6.1. Método de Morison

Una sugerencia hecha por Morison donde se proponía calcular los coeficientes C_d y C_m sin necesidad de usar ordenadores. Su acercamiento proponía lo que se ve en las ecuaciones 3.10 y 3.11.

$$C_d = \frac{2F}{\rho D u_0 |u_0|} \quad (3.10)$$

$$C_m = \frac{4F}{\pi \rho D^2 \omega u_0} \quad (3.11)$$

Para que las ecuaciones 3.11 y 3.10 puedan ser aplicadas, tal y como se indica en [8] se tiene que obtener el valor de la fuerza en función de las siguientes condiciones.

- Valor de F para C_d cuando la aceleración es de valor cero.
- Valor de F para C_m cuando la velocidad es de valor cero.

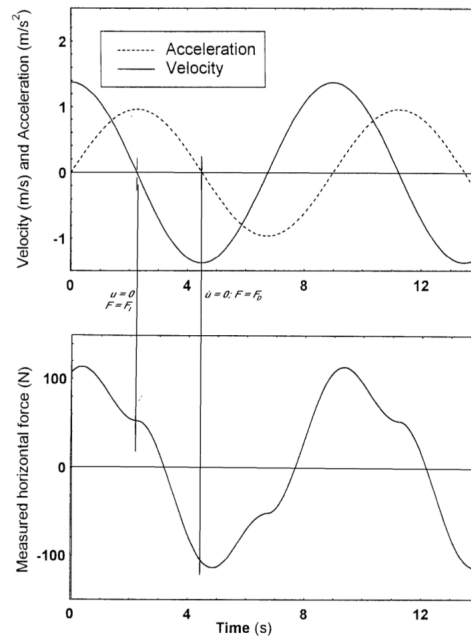


Figura 3.9: Ejemplo de señales (velocidad, aceleración y fuerza) Massie and Journée [8].

Aunque tal y como se indica en [8] este método no es el más preciso ya que pequeñas variaciones en la velocidad causan errores significativos principalmente en el cálculo de C_d . No obstante se calculan estos datos con la intención de un posterior análisis de dichos resultados.

3.6.2. Cálculo del ángulo δ mediante la transformada rápida de Fourier

Para aplicar las ecuaciones 3.8 y 3.9 con la finalidad de obtener dichos coeficientes se precisa despejar el valor de la incógnita δ para cada uno de los casos. Este ángulo de desfase entre fuerzas y desplazamientos se calculará aplicando una transformada rápida de Fourier implementada en el código de Matlab mostrado en 3.6.2.

```

1 clear all;
2 clc;
3 %% Carga de datos
4 load desplazamientos;
5 load fuerza;
6 %% Busca de posiciones en los vectores de datos
7 % Los valores son tomados para cada caso
8 % Fuerza
9 fi=max(find(fx<=0.8873)); % Valor inicial para el vector de
   fuerzas
10 ff=max(find(fx<=23.91)-1); % Valor final para el vector de
   fuerzas
11 dtf=fx(2)-fx(1); % Diferencial temporal para las fuerzas
12 % Desplazamiento
13 di=max(find(dx<=0.8873)); % Valor inicial para el vector de
   desplazamientos
14 df=max(find(dx<=23.91)-1); % Valor final para el vector de
   desplazamientos
15 dtd=dx(2)-dx(1); % Diferencial temporal para los
   desplazamientos
16 %% Vectores con los valores de estudio
17 fx=fx(fi:ff); % Vector de fuerzas eje X
18 fy=fy(fi:ff); % Vector de fuerzas eje Y
19 npf=length(fx); % Número de puntos de los vectores fuerza
20 dx=dx(di:df); % Vector de desplazamientos eje X
21 dy=dy(di:df); % Vector de desplazamientos eje Y
22 npd=length(dx); % Número de puntos de los vectores
   desplazamientos
23 %% Fast Fourier Transform
24 nyquistf=pi/dtf; % Nyquist para las fuerzas

```

```

25 nyquistd=pi/dtd; % Nyquist para los desplazamientos
26 Wf=linspace(-nyquistf,nyquistf,npf+1); % Vector de frecuencias
    de las
27     % fuerzas
28 Wd=linspace(-nyquistd,nyquistd,npd); % Vector de frecuencias de
    los
29     % desplazamientos
30 Yf=fft(fy); % FFT de las fuerzas
31 Yd=fft(dy); % FFT de los desplazamientos
32 %% Ordenar vectores
33 Yordf=zeros(1,npf+1);
34 for i=1:npf/2-1
35     Yordf(i)=conj(Yf((npf/2)+i)); % ordena la parte negativa de
        la
36         % transformada
37 end
38 for i=1:npf/2+1
39     Yordf((npf/2)+i)=conj(Yf(i)); % ordena la parte positiva de
        la
40         % transformada
41 end
42 Yordd=zeros(1,npd);
43 for i=1:(npd-1)/2
44     Yordd(i)=conj(Yd((npd+1)/2+i)); % ordena la parte negativa
        de la
45         % transformada
46 end
47 for i=1:(npd+1)/2
48     Yordd(((npd-1)/2+i))=conj(Yd(i)); % ordena la parte
        positiva de la
49         % transformada
50 end
51 %% Plot FFTs
52 % Fuerza
53 figure;
54 subplot(3,1,1);
55 plot(fx,fy);
56 title('Señal (fuerza)');
57 subplot(3,1,2);
58 plot(Wf/2/pi,real(Yordf));
59 title('Parte real de la frecuencia');
60 xlabel('Hz');
61 xlim([-0.8 0.8]); % Para mostrar el pico a la frecuencia de
    excitación
62 subplot(3,1,3);

```

```
63 plot(Wf/2/pi, imag(Yordf));
64 title('Parte imaginaria de la frecuencia');
65 xlabel('Hz');
66 xlim ([-0.8 0.8]); % Para mostrar el pico a la frecuencia de
    excitación
67 % Desplazamiento
68 figure;
69 subplot(3,1,1);
70 plot(fx,fy,'r');
71 title('Señal (desplazamiento)');
72 subplot (3,1,2);
73 plot(Wd/2/pi, real(Yordd), 'r');
74 title('Parte real de la frecuencia');
75 xlabel('Hz');
76 xlim ([-0.8 0.8]); % Para mostrar el pico a la frecuencia de
    excitación
77 subplot(3,1,3);
78 plot(Wd/2/pi, imag(Yordd), 'r');
79 title('Parte imaginaria de la frecuencia');
80 xlabel('Hz');
81 xlim ([-0.8 0.8]); % Para mostrar el pico a la frecuencia de
    excitación
```

Capítulo 4

Resultados

4.1. Introducción

Estructuras sumergidas o semi-sumergidas (parques eólicos, plataformas *Offshore*, estructuras flotantes, conductos de *oil and gas*, boyas) en corrientes marinas están habitualmente sometidas a fuertes vibraciones que presentan un serio daño para las partes mecánicas de las estructuras. Es por ello, que con la intención de analizar y eliminar esos problemas se propone estimar el valor de los coeficientes de arrastre y de inercia sobre una estructura cilíndrica. Para ello se considera un cuerpo bidimensional oscilando con una frecuencia y amplitud conocida en un fluido que inicialmente está en reposo y cuya principal característica es que es un fluido Newtoniano incompresible.

4.2. Solución numérica del problema del cilindro oscilante

Para la resolución del problema del cilindro oscilante en el seno de un fluido no ideal inicialmente en reposo se ha tomado las dimensiones y las características del problema ofrecido en Troesch and Kim [16]. Esto es, se toman las dimensiones del cilindro y características propias del mismo como son la amplitud, su frecuencia angular ya que así se podrá simular este problema y tener unos resultados de referencia con los que poder comparar. Además, interesa destacar que el fluido con el que se trabajará tiene una viscosidad cinemática “ ν ” a 20 °C de $1,003 \cdot 10^{-6} \frac{m^2}{s}$.

4.2.1. Estructura de la malla utilizada para los estudios numéricos

En un primer lugar, para poder llevar a cabo la resolución del problema comentado con Openfoam, se hace necesario definir y realizar una geometría. Está, presentada en la figura 4.1, tiene las siguientes características:

- Tiene la estructura cilíndrica centrada, $d = 0.203$ metros.
- Sobre la estructura cilíndrica centrada se establece una geometría cilíndrica de diámetro $D = 0.406$ con la que se pretende hacer un análisis más exhaustivo en esa zona.
- Se añaden dos franjas centrales sobre el cilindro, una transversal y otra perpendicular de las que también interesará un análisis diferente. Longitud de las mismas $l = 0.287086$ metros.
- Quedarán cuatro bloques uno por cada esquina de nuestra geometría dando lugar a una geometría de largo total de $L = 1.624$ metros y ancho total de $L = 1.624$ metros.

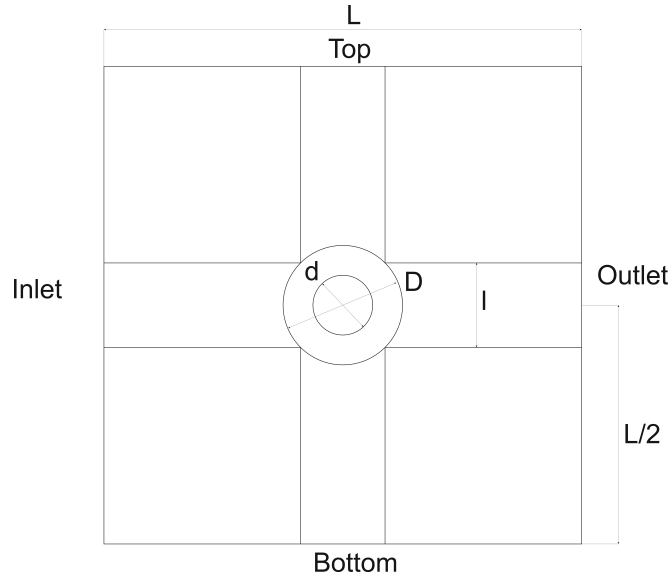


Figura 4.1: Geometría del sistema.

4.2.2. Condiciones de contorno

Tras estudiar la influencia de las condiciones de contorno en los límites exteriores del dominio y apreciar que la solución es prácticamente independiente de dichas condiciones se decide realizar las simulaciones con las condiciones de contorno especificadas en la figura 4.2 (Véase además las utilidades con las condiciones de contorno en el apéndice A). Dichas condiciones de contorno se dividen en dos grupos, por un lado las que corresponden al cilindro y por otro las que corresponden a los bordes de la geometría.

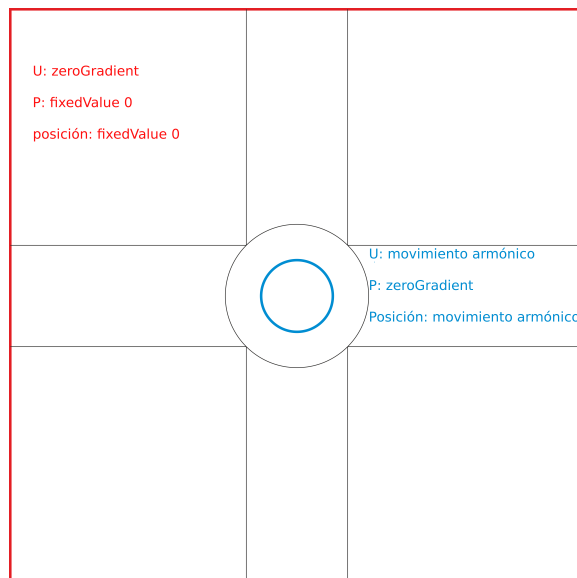


Figura 4.2: Geometría del sistema.

Para el cilindro las condiciones se marcan en azul y son:

- **Velocidad:** armónica según movimiento del cilindro.
- **Presión:** $\frac{\partial p}{\partial n} = 0$.
- **Posición:** *movimiento harmónico*

Para los bordes las condiciones de contorno se marcan en rojo y son:

- **Velocidad:** $\frac{\partial p}{\partial n} = 0$.
- **Presión:** $p=0$.
- **Posición:** Posición fija.

4.2.3. Mallados

Siguiendo con lo comentado en apartados anteriores, para poder realizar el análisis del sistema con la geometría comentada en el apartado 4.1 es necesario definir un mallado. En este trabajo se han realizado numerosas pruebas con diferentes mallados, pero a modo de síntesis, solo se comentará las tres más importantes.

- **Malla gruesa** (Figura 4.3): Esta es una malla “gruesa”, es decir, tiene un mallado compuesto por un bajo número de celdas lo que lleva a celdas grandes. Esta malla, ha servido principalmente para hacer estimaciones de cálculo debido a que su menor número de celdas implica un menor tiempo de cálculo computacional y no por ende implica resultados extremadamente distintos a los obtenidos con otras mallas. Para hacer una estimación de las celdas que posee esta malla se dice que posee 7488 hexaedros.
- **Malla normal** (Figura 4.4): Con esta malla se realizan aproximaciones más completas y de menor grado de incertidumbre. Esta, no obstante, tiene un mallado con mayor número de hexaedros que la **Malla Gruesa** y presenta resultados bastante aceptables con un tiempo de cálculo moderado. Además, se comienza a apreciar que los resultados de esta malla en comparación con la **Malla Fina** (ver Figura 4.5), son muy cercanos presentando un error de 1,75 %. Como síntesis, esta tiene 29952 celdas.
- **Malla fina** (Figura 4.5): Esta es la malla con la que se obtendrán todos los resultados finales debido a que los resultados son obtenidos con mayor precisión. Sin embargo, presenta el inconveniente de que el tiempo de cálculo computacional es mayor principalmente porque esta malla está compuesta por 117888 celdas.

Finalmente queda destacar que para la elección de las mallas finales, primero se ha hecho un pequeño análisis de convergencia. De este se puede destacar que se simuló las diferentes mallas haciendo mover el cilindro con un movimiento armónico para conseguir unos resultados de fuerza y velocidad armónicos. Esto es, si al analizar los resultados se veía que no seguían movimientos armónicos (se ven señales con ruido) se sabría que esas mallas eran objeto de descarte ya que no estaban “trabajando correctamente”. Un claro ejemplo lo vemos en la figura 4.6 donde se muestra 6 tipos diferentes de malla en donde *Malla 3*, *Malla*

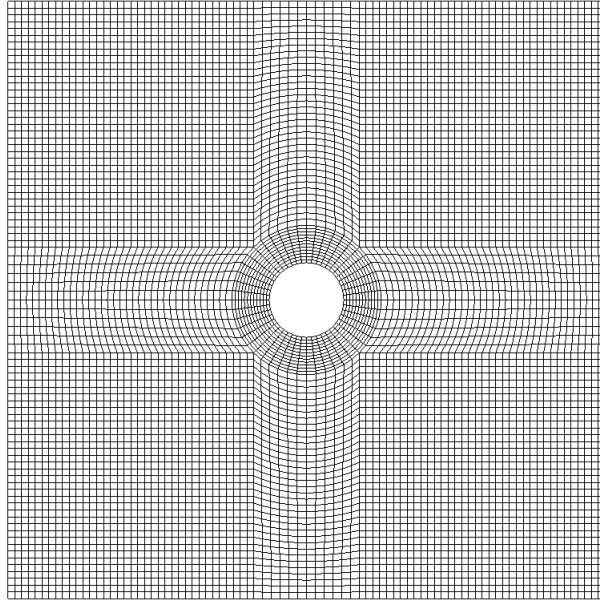


Figura 4.3: Malla Gruesa.

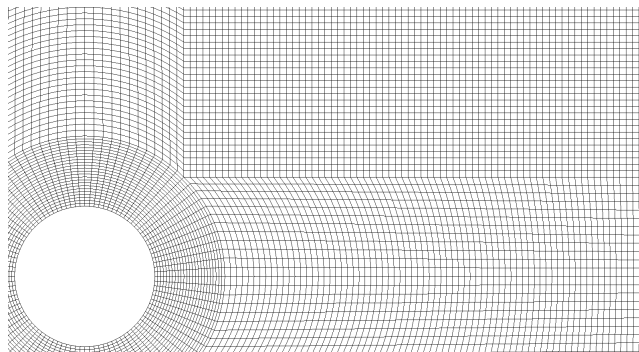


Figura 4.4: Sección de Malla Normal.

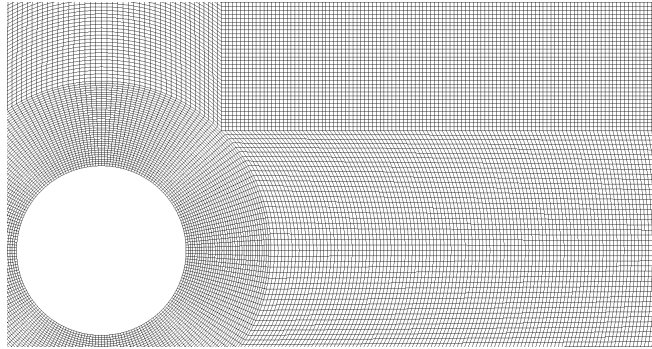


Figura 4.5: Sección de Malla Fina.

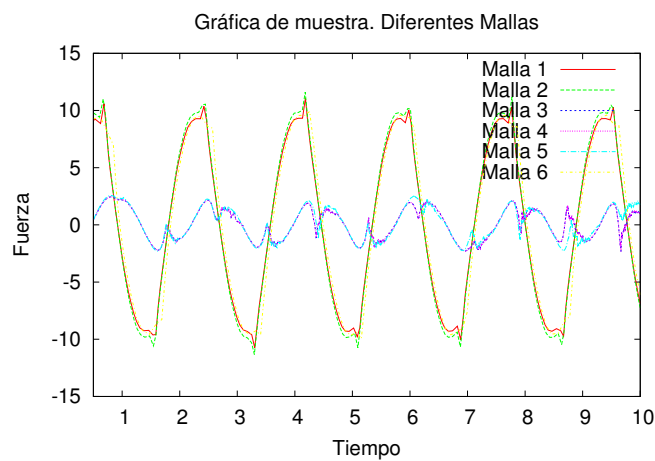


Figura 4.6: Análisis de 6 mallas diferentes (desechables).

4 y *Malla 5* presentan unos resultados con mucho ruido y bastante lejos de la realidad, y el resto de las mallas (*Malla 1*, *Malla 2*, y *Malla 6*) presentaron unos resultados próximos a ser armónicos pero aún con algo de ruido (por tanto se desechan todas las mallas).

Por otro lado, se presentan los resultados de las tres mallas en donde se puede apreciar que son senoides casi perfectas y con muy poca variación entre ellas (figura 4.7).

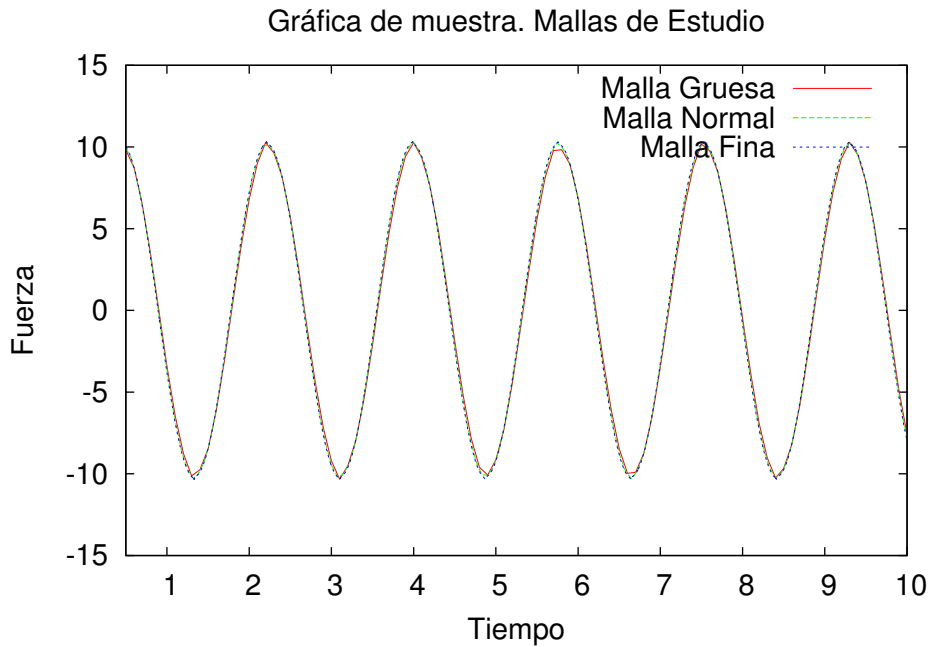


Figura 4.7: Análisis de las 3 mallas finales.

4.3. Números adimensionales

En una primera toma de contacto con el problema, y sabiendo que las componentes de fuerza y el sistema de respuesta dependen de diferentes parámetros, se propone discutir los coeficientes adimensionales más significativos y que serán necesarios en el problema.

- **Número de Reynolds (Re):** Es un número adimensional que es significativo en el diseño o modelo de cualquier sistema donde el efecto de viscosidad es importante para el control de las velocidades o en el flujo de un fluido.

$$\frac{UD}{\nu} = \frac{\text{fuerza de inercia}}{\text{fuerzas viscosas}} \quad (4.1)$$

Donde es ν la viscosidad cinemática del fluido, y es igual a la viscosidad dinámica (μ) dividido por la densidad del fluido. El número de *Reynolds* da medida de la transición de laminar a flujo turbulento.

- **Número de Reynolds cinético (R_k):** En un fluido en reposo o con flujo oscilatorio, la fuerza del fluido es función de este número, que viene dado por:

$$R_k = \frac{\omega D^2}{\nu} \quad (4.2)$$

Donde ω es la frecuencia angular de las oscilaciones.

- **Parámetro de Keulegan – Carpenter (K_c):** En fluidos armónicos este número está definido por:

$$K_c = \frac{U_o T_m}{L} \quad (4.3)$$

Donde U es la amplitud de velocidad del flujo y T es el periodo.

Para un cilindro oscilando en un flujo en reposo, $U = a \omega$ y $T = \frac{2\pi}{\omega}$, donde ω es la frecuencia angular de la oscilación del cilindro y a es el desplazamiento del cilindro (amplitud).

$$K_c = \frac{2\pi A_0}{L} \quad (4.4)$$

En el artículo de Triesch y Kim [16], se introduce la ecuación 4.3 de forma errónea.

- **Parámetro de frecuencia.** Este parámetro viene dado por la relación entre el número de Reynolds y el Parámetro de Keulegan–Carpenter.

$$\beta = \frac{Re}{K_c} = \frac{f L^2}{\nu} \quad (4.5)$$

4.4. Solución analítica para un cilindro en el seno de un fluido ideal

4.4.1. Consideraciones iniciales

Es interesante comenzar introduciendo que una componente estructural moviéndose a velocidad constante en un fluido infinito ideal, no encuentra resistencia. Este fenómeno es comúnmente conocido como la paradoja de D'Alembert. Por otro lado un cuerpo moviéndose a velocidad variable, incluso bajo la condición de flujo potencial, encuentra resistencia. El cuerpo se comporta como si una porción de masa del fluido quedara fijada y por lo tanto moviéndose conjuntamente. Además, cuando el cuerpo es sometido a excitación, no solo la

masa del cuerpo se acelera sino que la masa del fluido lo hace también. Podremos decir que la fuerza adicional para acelerar el cuerpo viene dada por:

$$g = -m_a \frac{\partial^2 u}{\partial t^2} \quad (4.6)$$

donde m_a está referido a la masa añadida (véase 3.5).

Ahora, conocida la velocidad de las partículas de agua en un sistemas de coordenadas polar bidimensional en que cada P punto del plano se determina por una distancia r y un ángulo θ , donde r es la distancia de un punto P al origen O del plano y θ es el ángulo formado entre el eje polar y la recta dirigida que va del origen O al punto P , unido a la definición de potencial de velocidad ϕ se puede enunciar la ecuación de velocidad de flujo 4.7. Esta, será una ecuación a tener en cuenta como punto de partida en este análisis en el que se marca como objetivo encontrar una ecuación para el flujo potencial genérica que se pueda aplicar a los casos:

1. Cilindro rígido en reposo en el seno de un fluido oscilante.
2. Cilindro oscilante en el seno de un fluido inicialmente en reposo.

$$\vec{U} = \nabla \phi \quad (4.7)$$

La ecuación de Laplace para flujo potencial viene dada por 4.8 (definición de Laplaciano en coordenadas polares).

$$\nabla^2 \phi(r, \theta, t) = \frac{\partial^2 \phi}{\partial r^2} + \frac{1}{r} \frac{\partial \phi}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \phi}{\partial \theta^2} = 0 \quad (4.8)$$

donde ϕ es la velocidad potencial del flujo y ∇^2 es el operador Laplaciano.

Seguidamente, la ecuación 4.9 en la cual se enuncia la presión del fluido.

$$p = -\rho \frac{\partial \phi}{\partial t} \quad (4.9)$$

Seguidamente se puede enunciar la ecuación 4.10 donde $F_\theta(\theta)$ y $F_r(r)$ son funciones de ϕ aún no conocidas.

$$\phi(r, \theta, t) = F_r(r) F_\theta(\theta) e^{i\omega t} \quad (4.10)$$

Aplicando la ecuación 4.10 a la ecuación 4.8:

$$F_\theta(\theta) \frac{\partial^2 F_r(r)}{\partial r^2} + \frac{1}{r} F_\theta(\theta) \frac{\partial F_r(r)}{\partial r} + \frac{1}{r^2} F_r(r) \frac{\partial^2 F_\theta(\theta)}{\partial \theta^2} = 0 \quad (4.11)$$

Dividiendo entre $F_r(r)$ y $F_\theta(\theta)$

$$\frac{1}{F_r(r)} \frac{\partial^2 F_r(r)}{\partial r^2} + \frac{1}{r F_r(r)} \frac{\partial F_r(r)}{\partial r} + \frac{1}{r^2} \frac{1}{F_\theta} \frac{\partial^2 F_\theta(\theta)}{\partial \theta^2} = 0 \quad (4.12)$$

Multiplicando por r^2 y reordenando:

$$\frac{r^2}{F_r(r)} \frac{\partial^2 F_r(r)}{\partial r^2} + \frac{r}{F_r(r)} \frac{\partial F_r(r)}{\partial r} = -\frac{1}{F_\theta} \frac{\partial^2 F_\theta(\theta)}{\partial \theta^2} = \lambda \quad (4.13)$$

Donde λ es una constante de valor propio. Seguidamente, por separación de variables, se obtienen dos ecuaciones.

$$-\frac{1}{F_\theta} \frac{\partial^2 F_\theta(\theta)}{\partial \theta^2} = \lambda \quad \Rightarrow \quad \frac{\partial^2 F_\theta(\theta)}{\partial \theta^2} + F_\theta \lambda = 0 \quad (4.14a)$$

$$r^2 \frac{\partial^2 F_r(r)}{\partial r^2} + r \frac{\partial F_r(r)}{\partial r} - F_r(r) \lambda = 0 \quad (4.14b)$$

Donde:

- La solución de 4.14a es del tipo:

$$F_\theta = \alpha_1 \sin \sqrt{\lambda} \theta + \alpha_2 \cos \sqrt{\lambda} \theta \quad (4.15)$$

- La solución de 4.14b es del tipo:

$$F_r = A_1 e^{-\sqrt{\lambda} r} + A_2 e^{\sqrt{\lambda} r} \quad (4.16)$$

Ahora que se ha planteado las ecuaciones 4.15 y 4.16 se debe comprobar si la ecuación 4.16 es solución. Para ello, tomando en cuenta 4.17 aplicándolo sobre 4.16 y derivando parcialmente obtenemos 4.18.

$$\cosh(x) = \frac{e^x + e^{-x}}{2} \quad \sinh(x) = \frac{e^x - e^{-x}}{2} \quad (4.17)$$

$$\frac{\partial F_r}{\partial r} = \frac{A_1 \sinh(\sqrt{\lambda} \ln(r)) \sqrt{\lambda}}{r} + \frac{i A_2 \cosh(\sqrt{\lambda} \ln(r)) \sqrt{\lambda}}{r} \quad (4.18)$$

$$\begin{aligned} \frac{\partial^2 F_r}{\partial r^2} = & A_1 \sqrt{\lambda} \frac{\cosh(\sqrt{\lambda} \ln(r)) \sqrt{\lambda} - \sinh(\sqrt{\lambda} \ln(r)) \sqrt{\lambda}}{r^2} \\ & + i A_2 \sqrt{\lambda} \frac{-\cosh(\sqrt{\lambda} \ln(r)) \sqrt{\lambda} + \sinh(\sqrt{\lambda} \ln(r)) \sqrt{\lambda}}{r^2} \end{aligned} \quad (4.19)$$

Sustituyendo en 4.14b y resolviendo queda:

$$\begin{aligned} & A_1 \sqrt{\lambda} \{ \sqrt{\lambda} \cosh[\sqrt{\lambda} \ln(r)] - \sinh[\sqrt{\lambda} \ln(r)] \} + \\ & i A_2 \sqrt{\lambda} \{ \sqrt{\lambda} \sinh[\sqrt{\lambda} \ln(r)] - \cosh[\sqrt{\lambda} \ln(r)] \} + \\ & A_1 \sinh[\sqrt{\lambda} \ln(r)] \sqrt{\lambda} + i A_2 \cosh[\sqrt{\lambda} \ln(r)] \sqrt{\lambda} - \\ & \{ \lambda A_1 \cosh[\sqrt{\lambda} \ln(r)] + i A_2 \lambda \sinh[\sqrt{\lambda} \ln(r)] \} = 0 \end{aligned} \quad (4.20)$$

Con lo que se puede expresar $F_r(r)$ como:

$$F_r = A_1 \cosh(\sqrt{\lambda} \ln(r)) + i A_2 \sinh(\sqrt{\lambda} \ln(r)) \quad (4.21)$$

Sabiendo que en este caso la parte imaginaria es nula y volviendo a tener en cuenta 4.17 se propone:

$$F_r = A_1 e^{-\sqrt{\lambda} r^{-1}} + A_2 e^{\sqrt{\lambda} r} \quad (4.22)$$

De aquí:

$$\frac{\partial^2 F_r}{\partial r} = -A_1 e^{-\sqrt{\lambda} r^{-2}} + A_2 e^{\sqrt{\lambda} r} \quad (4.23)$$

$$\frac{\partial^2 F_r}{\partial r^2} = 2A_1 e^{-\sqrt{\lambda} r^{-3}} \quad (4.24)$$

Sustituyendo en 4.14b obtenemos 4.25.

$$2 A_1 e^{-\sqrt{\lambda} r^{-1}} - A_1 e^{-\sqrt{\lambda} r^{-1}} + A_2 e^{\sqrt{\lambda} r} - A_1 e^{-\sqrt{\lambda} r^{-1}} \lambda - A_2 e^{\sqrt{\lambda} r} \lambda = 0 \quad (4.25)$$

$$A_1 e^{-\sqrt{\lambda} r^{-1}} (1 - \lambda) + A_2 e^{\sqrt{\lambda} r} (1 - \lambda) = 0 \quad (4.26)$$

$$(A_1 e^{-\sqrt{\lambda} r^{-1}} + A_2 e^{\sqrt{\lambda} r}) (1 - \lambda) = 0 \quad (4.27)$$

Ecuación que se verifica y demuestra la ecuación 4.16 ya que de 4.27 despejando $\lambda = 1$ podemos obtener 4.28.

$$F_r(r) = A_1' r^{-1} + A_2' r \quad (4.28)$$

Seguidamente introduciendo los valores de 4.14a y 4.30 en 4.10 se obtiene 4.29

$$\phi = (A_1' r^{-1} + A_2' r) (\alpha_1 \sin \theta + \alpha_2 \cos \theta) e^{i\omega t} \quad (4.29)$$

Nótese que la ecuación 4.29 aparece en el documento de referencia [2] donde sin embargo existen varias erratas significativas tal y como se ha demostrado en los desarrollos anteriores.

4.4.2. Cilindro oscilante en el seno de un fluido inicialmente en reposo

Para el caso que se está tratando en este apartado, cilindro oscilante en el seno de un fluido inicialmente en reposo, se enunciarán sus condiciones de contorno. De estas se conoce que en el infinito, el fluido no está perturbado y en donde existen interferencias entre el fluido y la estructura, la velocidad normal es igual que la del cilindro.

$$\begin{aligned} u_r &= 0 & \text{para} & \quad r = \infty \\ u_r &= u \cos\theta & \text{para} & \quad r = R \end{aligned} \quad (4.30)$$

Tomando en cuenta la ecuación 4.29 donde se sabe que α_1 debe ser cero para que se cumpla la condición de contorno 4.30 para el caso que $r = R$ y también que $A_2' = 0$ para que cumpla la condición de contorno 4.30 para el caso que $r = \infty$.

$$(A_1' r^{-1} + A_2' r)(\alpha_2 \cos\theta) e^{i\omega t} = 0 \quad (4.31)$$

Sustituyendo y sacando factor común A_1' y α_2 queda:

$$\phi = a \left(\frac{R^2}{r} + r \right) \cos\theta e^{i\omega t} \quad (4.32)$$

Aplicando 4.9 teniendo en cuenta 4.30 y 4.32 obtenemos 4.33.

$$P = -\rho \frac{\partial\phi}{\partial t} = -a \left(\frac{R^2}{r} \right) \cos\theta e^{i\omega t} i\omega \quad (4.33)$$

Por tanto, teniendo en cuenta la ecuación 4.33 la fuerza del fluido actuando sobre el cilindro será:

$$\begin{aligned} g &= - \int_0^{2\pi} p(r, \theta, t) |_{r=R} R \cos\theta \, d\theta = - \int_0^{2\pi} \rho R a \cos\theta i\omega e^{(i\omega t)} R \cos\theta \, d\theta \\ &= - \rho R^2 \underbrace{ai\omega e^{(i\omega t)}}_{\frac{d^2u}{dt^2}} \underbrace{\int_0^{2\pi} \cos^2\theta \, d\theta}_{\pi} \end{aligned} \quad (4.34)$$

Tomando en cuenta 4.6 y 4.34 se llega a que:

$$g = -m_a \frac{d^2u}{dt^2}; \quad m_a = C_m \pi \rho R^2 \quad (4.35)$$

este caso que es uno de los más simples vemos que C_m es igual a 1.

$$C_m = \frac{m_a}{\pi \rho R^2} = 1 \quad (4.36)$$

4.4.3. Cilindro rígido en reposo en el seno de un fluido oscilante

En este caso, el cilindro no es oscilante. De hecho está quieto siendo el fluido el que se mueve bajo un movimiento armónico. Aquí se puede decir que las condiciones de contorno nos dicen que el fluido permanece quieto cerca del cilindro pero en el infinito siguen dicho movimiento armónico.

$$\begin{aligned} u_r &= u \cos \theta & \text{para } r &= \infty \\ u_r &= 0 & \text{para } r &= R \end{aligned} \quad (4.37)$$

Tomando en cuenta la ecuación 4.29 se sabe que α_1 debe ser cero para que se cumpla la condición de contorno 4.37 para el caso que $r = \infty$.

$$(A_1' r^{-1} + A_2' r)(\alpha_2 \cos \theta) e^{i\omega t} = 0 \quad (4.38)$$

Despejando A_1 :

$$\begin{aligned} \frac{-A_1'}{R^2} + A_2' &= 0 \\ A_1' &= A_2' R^2 \end{aligned} \quad (4.39)$$

Sustituyendo 4.39 en 4.38 y sacando factor común queda 4.40.

$$\phi = a \left(\frac{R^2}{r} + r \right) \cos \theta e^{i\omega t} \quad (4.40)$$

Aplicando 4.9 teniendo en cuenta 4.37 y 4.40 obtenemos 4.41.

$$P = -\rho \frac{\partial \phi}{\partial t} = -a \left(\frac{R^2}{r} + r \right) \cos \theta e^{i\omega t} i\omega \quad (4.41)$$

Por tanto, teniendo en cuenta la ecuación 4.41 la fuerza del fluido actuando sobre el cilindro será:

$$\begin{aligned} g &= - \int_0^{2\pi} p(r, \theta, t) |_{r=R} R \cos \theta d\theta = - \int_0^{2\pi} 2\rho R a \cos \theta i\omega e^{(i\omega t)} R \cos \theta d\theta \\ &= - 2\rho R^2 \underbrace{ai\omega e^{(i\omega t)}}_{\frac{d^2 u}{dt^2}} \underbrace{\int_0^{2\pi} \cos^2 \theta d\theta}_{\pi} \end{aligned} \quad (4.42)$$

Tomando en cuenta 4.6 y 4.42 se llega:

$$g = -m_a \frac{d^2u}{dt^2}; \quad m_a = 2 C_m \pi \rho R^2 \quad (4.43)$$

este caso se ve que C_m es igual a 2.

$$C_m = \frac{g}{\pi \rho R^2} = 2 \quad (4.44)$$

4.4.4. Comentarios

En lo que se ha venido desarrollando en el capítulo 4.4 no se ha encontrado bibliografía alguna que demuestre la relación entre las situaciones previamente comentadas y demostradas. De hecho, en algunos documentos de referencia estos conceptos no quedan claros. Por ejemplo Troesch and Kim [16] indica en su introducción que los casos 4.4.3 y 4.4.2 son iguales, algo que se ha demostrado que no es verdad ya que los coeficientes de C_m según el caso son completamente diferentes.

4.5. Solución numérica del cilindro oscilante

4.5.1. Definición del problema

Es el momento de pasar a realizar las simulaciones. Dichas simulaciones se realizarán para un fluido no ideal, inicialmente en reposo y que se contrastarán los resultados con Troesch and Kim [16]. Téngase en cuenta que en Troesch and Kim [16] no se especifican las propiedades específicas del fluido.

Las simulaciones que se realizarán con OpenFoam tal y como se ha comentado (Véase el apéndiceA para ver la estructura de un caso de OpenFoam), se supondrá que el fluido es agua a 20 °C y con $\nu = 1.003 \cdot 10^{-6} \frac{m^2}{s}$. Además, se calcularán los valores de 4.3 ya que son necesarios para realizar las simulaciones para los valores de ?? y ??. Dichos valores se introducirán en OpenFoam y se resolverá las ecuaciones de Navier–Stokes para los regímenes laminar y turbulento. Para el régimen turbulento se usará el modelo k–epsilon (apartado 2.4.2) con los resultados de las ecuaciones obtenidas de la misma.

Las simulaciones que se van a llevar a cabo para el caso laminar y turbulento están clasificadas según el valor de β . Dicha clasificación es, por un lado para β igual a 23200 y con K_c 0.2 y 0.8, y por otro lado para β igual a 48600 y con K_c 0.2 y 0.5. Estos valores de β y de K_c han sido seleccionados con el mismo valor con el que se presentan los resultados

en Troesch and Kim [16] ya que así se podrá comparar los resultados de dicha publicación con los que se calculan en este trabajo.

Por otro lado, a partir de las dimensiones ya dadas del cilindro (diámetro igual a 0.203 metros) se procede al cálculo de los valores de los números adimensionales llegando a los resultados que se muestran:

K_c / Números	Re	A_0 metros	ω $\frac{\text{rad}}{\text{seg}}$	T (seg)
0.2	4640.0	0.00646169068953	3.54794362455	1.77093718843
0.8	18560.0	0.0258467627581	3.54794362455	1.77093718843

Tabla: 4.1: Parámetros para $\beta = 23200$

K_c / Números	Re	A_0 metros	ω $\frac{\text{rad}}{\text{seg}}$	T (seg)
0.2	9720.0	0.00646169068953	7.43233017901	0.845385653738
0.5	24300.0	0.0161542267238	7.43233017901	0.845385653738

Tabla: 4.2: Parámetros para $\beta = 48600$

4.5.2. Resultados

En este apartado se presentan los resultados correspondientes a los valores de C_m y C_d obtenidos.

4.5.2.1. Resultados obtenidos desde Openfoam

Estos resultados han sido obtenidos a partir de las simulaciones que se han realizado en Openfoam. Con los resultados ahí obtenidos se aplican las ecuaciones 3.8 y 3.9 dando los siguientes resultados.

Simulación	K_c	C_D	C_M
Simulación 1	0.2	0.0243809268677	1.00091920741
Simulación 2	0.8	0.099934169034	0.998987211059

Tabla: 4.3: Coeficientes para régimen laminar y $\beta = 23200$

Simulación	K_c	C_D	C_M
Simulación 3	0.2	0.234427185678	1.00097060632
Simulación 4	0.5	0.107185302706	0.99363533139

Tabla: 4.4: Coeficientes para régimen laminar y $\beta = 48600$

Simulación	K_c	C_D	C_M
Simulación 5	0.2	1.93903948485	1.02630147321
Simulación 6	0.8	0.450366362956	0.969726732117

Tabla: 4.5: Coeficientes para régimen turbulento y $\beta = 23200$

Simulación	K_c	C_D	C_M
Simulación 7	0.2	1.93903948485	1.01647393317
Simulación 8	0.5	0.796731149697	1.01180918256

Tabla: 4.6: Coeficientes para régimen turbulento y $\beta = 48600$

4.5.2.2. Resultados Método Morison

Seguidamente se presentan los resultados obtenidos mediante la aplicación del Método de Morison.

Simulación	K_c	C_D	C_M
Simulación 1	0.2	0.037490238398782565	0.990587258852439
Simulación 2	0.8	0.10522420358501083	0.9848628359714976

Tabla: 4.7: Coeficientes obtenidos por el Método Morison para régimen laminar y $\beta = 23200$

Simulación	K_c	C_D	C_M
Simulación 3	0.2	0.24792303026210608	0.9882409763528609
Simulación 4	0.5	0.11627596958177051	0.9815624547584448

Tabla: 4.8: Coeficientes obtenidos por el Método Morison para régimen laminar y $\beta = 48600$

Simulación	K_c	C_D	C_M
Simulación 5	0.2	1.8985206686097085	1.0161158298005322
Simulación 6	0.8	0.6345645785755704	1.0150167178166984

Tabla: 4.9: Coeficientes obtenidos por el Método Morison para régimen turbulento y $\beta = 23200$

Simulación	K_c	C_D	C_M
Simulación 7	0.2	1.5610783975026201	1.0063850118108275
Simulación 8	0.5	0.7741525004277429	1.0044990223817338

Tabla: 4.10: Coeficientes obtenidos por el Método Morison para régimen turbulento y $\beta = 48600$

4.5.3. Análisis de resultados

Una vez vistos los resultados se precisa de comparación con resultados obtenidos por otros autores. Es por ello que se realizará la comparativa de resultados con el artículo Troesch and Kim [16].

- Análisis de C_m

Se observa claramente valores prácticamente iguales entre las simulaciones que se han realizado para los regímenes laminar y turbulento. Como se ha demostrado, dichos valores tienen que valer 1. En el caso de nuestras simulaciones, dicho valor es prácticamente 1 con pequeñas variaciones de decimales relacionadas con los errores propios de las simulaciones y las FFT's. Sin embargo vemos que en Troesch and Kim [16] le salen valores próximos a 3 para el caso de $\beta = 23200$, lo cual es evidentemente

una errata a la luz incluso de las expresiones que ellos presentan en el mismo trabajo para su cálculo.

Además, si se tienen en cuenta los valores de C_m en dados por Troesch and Kim [16] para el caso de $\beta = 48600$ se ve que los valores son cercanos a dos, algo que se ha demostrado tanto analíticamente como mediante simulaciones que no corresponde con el caso del cilindro oscilando en el seno de un fluido inicialmente en reposo ya que el valor de C_m debe ser 1. EL valor ofrecido, 2, corresponde al caso de un cilindro quieto en el seno de un fluido vibrante armónicamente.

Finalmente comparando el Método Morison con los resultados obtenidos por las simulaciones vemos que se encuentran muy cercanos.

■ Análisis de C_d

Para los valores de C_d lo primero destacable son los valores mostrados en las figuras 4.8 y 4.9 para el régimen laminar. Éstos son del orden de los valores ofrecidos por Troesch and Kim [16].

Por otro lado, si se compara la simulación realizada para régimen turbulento con los valores empíricos de C_d presentados en Troesch and Kim [16], se aprecia que los resultados numéricos reproducen la tendencia y los valores empíricos, teniendo en cuenta su dispersión.

Además viendo los resultados obtenidos a partir del Método Morison y comparándolos con las simulaciones realizadas se ve que prácticamente se obtienen los mismos resultados, llegando a la conclusión que dicho método ha proporcionado resultados aptos.

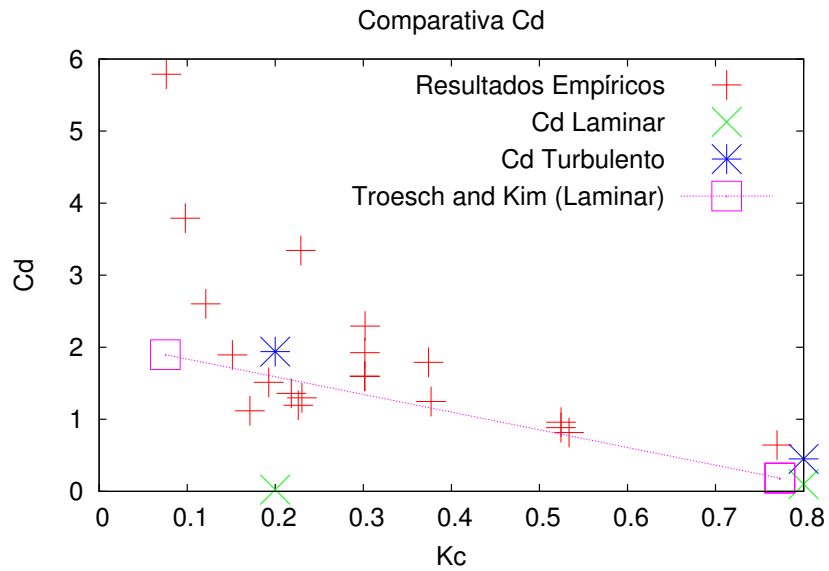


Figura 4.8: Comparativa de resultados de Cd para $\beta = 23200$

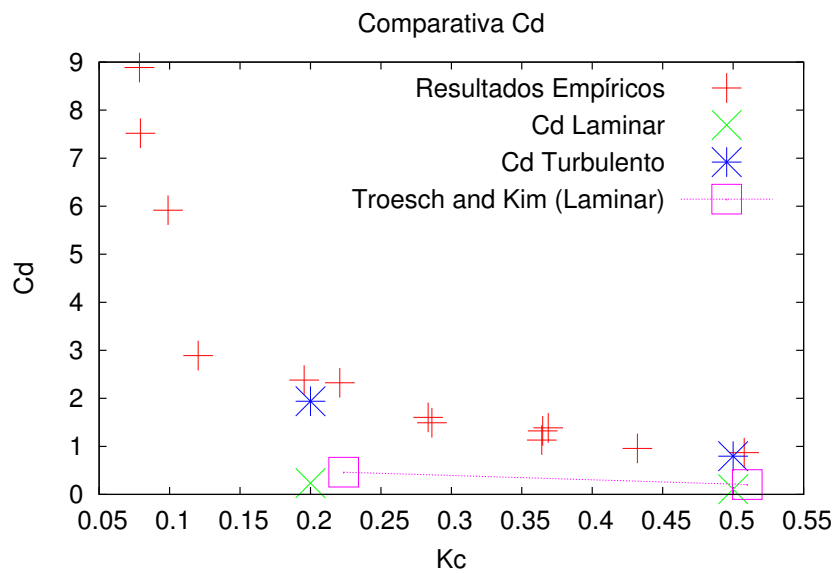


Figura 4.9: Comparativa de resultados de Cd para $\beta = 48600$

Capítulo 5

Conclusiones y trabajo futuro

5.1. Resumen

En este trabajo se ha valorado la utilización del método de los volúmenes finitos para el cálculo de los parámetros de masa y amortiguamiento añadidos en cilindros sumergidos oscilando en un fluido inicialmente en reposo modelado utilizando las ecuaciones de Navier-Stokes particularizadas para flujo incompresible turbulento. Como herramienta, se ha utilizado el software OpenFoam. Se ha visto que el parámetro de masa añadida puede obtenerse de manera relativamente directa en los casos estudiados, en los que los valores se aproximan mucho a los ofrecidos por la teoría de potencial. La estimación de los parámetros de amortiguamiento añadido ha dado peores resultados, posiblemente por que el modelo de turbulencia k - ϵ utilizado, así como el modelo de capa límite, no son los más adecuados

5.2. Trabajo futuro

En este trabajo se ha aprendido numerosas cosas que van desde el rigor científico hasta la iniciación al uso de los métodos numéricos. Esto debe ayudar a seguir estudiando con mayor detenimiento este problema. Finalmente, se considerará una serie de propuestas de cara a posibles trabajos futuros:

- Estudio de cilindro oscilando y fluido oscilando libremente.
- Estudio de geometrías más complejas para los casos comentados en 2D.
- Estudio de geometrías 3D para los casos comentados.

Bibliografía

- [1] J. R. Chaplin. Hydrodynamic damping of a cylinder at $\beta \approx 10^6$. *Journal of Fluids and Structures*, 14(8):1101–1117, 2000.
- [2] S. S. Chen. *Flow-Induced Vibration Of Circular Cylindrical Structures*. Argonne National Laboratory. Components Technology Division, 1985.
- [3] J. H. Ferziger and M. Perić. *Computational Method for Fluid Dynamics*, volume Trird, rev. edition. Springer, 2002.
- [4] J. F. Wilson. *Dynamics Of Offshore Structures*. Jhon Wiley and Sons, Inc., second edition edition, 2003.
- [5] F. Chirino Godoy. *Volúmenes Finitos*. ULPGC, 2014 – 2015. Notas de la asignatura Computación Naval.
- [6] S. Becker H. Dütsch, F. Durst and H. Lienhart. Low-reynolds-number flow around an oscillating circular cylinder at low Keulegan–Carpenter numbers. *Journal of Fluids Mechanics*, 360:249 – 271, 1998.
- [7] A. Lavrov J.F.M. Gadelho, J.M. Rodrigues and C. Guedes Soares. Determining hydrodynamic coefficients of a cylinder with Navier–Stokes equations. *Maritime Technology and Engineering*, pages 1001– 1007, 2014.
- [8] J.M.J Journée and W.W. Massie. *Offshore Hydromechanics*. Delf University of Technology, 2001.
- [9] G. H. Keulegan and Ll. H. Carpenter. Forces on cylinders and plates in an oscillating fluid. *Journal of Research of the National Bureau of Standards*, 60(5):423 – 440, May 1958.
- [10] S. Brizzolara L. Bonfiglio and C. Chrysosostomidis. Added mass and damping of oscillating bodies: a fully viscous numerical approach. *Recent Advances in Fluid Mechanics, Heat and Mass Transfer and Biology*, 2012.

- [11] H. K. Versteeg & W. Malalasekera. *An introduction to Computational Fluid Dynamics. The finite Volume Method*. Longman Scientific & Technical, 1995.
- [12] J. I. Cardesa Due nas. *Notas sobre Dinámica de Fluidos Computacional*, 2014.
- [13] J. M. Fernandez Oro. *Técnicas Numéricas en Ingeniería de Fluidos. Introducción a la dinámica de fluidos computacional (CFD) por el método de volúmenes finitos*. Reverté, 2012.
- [14] T. Sarpkaya. Force on a circular cylinder in viscous oscillatory flow at low keulegan–carpenter numbers. *Journal of Fluids Mechanics*, 165:61–71, 1985.
- [15] T. Sarpkaya. *Wave Forces on Offshore Structures*. Cambridge University Press, 2010.
- [16] A. W. Troesch and S. K. Kim. Hydrodynamic forces acting on cylinders oscillating at small amplitudes. *Journal of Fluids and Structures*, 5(1):113 – 126, 1991.
- [17] F. M. White. *Mecánica de Fluidos*, volume Quinta edición. Mc Graw Hill, 2003.
- [18] www.cfdonline.com.
- [19] www.openfoam.org.

Apéndice A

Utilidades en OpenFoam

A.1. Posición

```
/*-----*- C++ -*-----*\
| ===== | |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 2.1.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\/ M a n i p u l a t i o n | |
\*-----*/
FoamFile
{
    version      2.0;
    format      ascii;
    class      pointVectorField;
    location    "0";
    object      pointMotionU;
}
// *****

dimensions      [0 1 0 0 0 0 0];

internalField    uniform (0 0 0);

boundaryField
```



```
{
  out
  {
    type          uniformFixedValue;
    uniformValue  (0 0 0);
  }
  Top
  {
    type          uniformFixedValue;
    uniformValue  (0 0 0);
  }
  Down
  {
    type          uniformFixedValue;
    uniformValue  (0 0 0);
  }
  in
  {
    type          uniformFixedValue;
    uniformValue  (0 0 0);
  }
  cylinder
  {
    type          oscillatingDisplacement;
    amplitude     (0.0258 0 0);
    omega         3.5479;
    value         uniform (0 0 0);
  }
  back
  {
    type          empty;
  }
  front
  {
    type          empty;
  }
}
```

```
// ***** //
```

A.2. Velocidad

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 2.1.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n |
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    object       U;
}
// ***** //

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    out
    {
        type      zeroGradient;
    }
    Top
    {
        type      zeroGradient;
    }
    Down
}
```

```

    {
        type            zeroGradient;
    }
    in
    {
        type            zeroGradient;
    }
    cylinder
    {
        type            movingWallVelocity;
        value           uniform (0 0 0);
    }
    back
    {
        type            empty;
    }
    front
    {
        type            empty;
    }
}
// ***** //

```

A.3. Presión

```

/*-----*- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 2.1.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n | |
/*-----*- C++ -*-----*\
FoamFile
{
    version 2.0;
}

```



```

    back
    {
        type            empty;

    }
    front
    {
        type            empty;

    }
}
// ***** //

```

A.4. controlDict

```

/*-----*- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 2.1.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n | |
\*-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       controlDict;
}
// ***** //

application     pimpleDyMFoam;

startFrom       startTime;

startTime       0;

```

```
stopAt      endTime;

endTime     50;

deltaT      0.000400211793997;

writeControl  timeStep;

writeInterval 150;

purgeWrite   0;

writeFormat  binary;

writePrecision 6;

writeCompression off;

timeFormat   general;

timePrecision 6;

runTimeModifiable true;

adjustTimeStep no;

maxCo        0.0163108540042;

functions
{
    probes
    {
        type          probes;
        functionObjectLibs ("libsampling.so");
        outputControl  timeStep;
        outputInterval 1;
        probeLocations
```

```

    (
      ( 0.15 0 0 )
      ( 0.3 0 0 )
      ( 0.6 0 0 )

    );
    fixedLocations false;
    fields
    (
      p U
    );
}
forces
{
  type forces;
  functionObjectLibs ( "libforces.so" );
  outputControl timeStep;
  outputInterval 1;
  patches (cylinder);
  pName p;
  UName U;
  rhoName rhoInf;
  log true;
  rhoInf 1000;//DENSIDAD AGUA
  CofR (0 0 0);// NO ES UN VALOR SOLICITADO ASÍ QUE 0
}

// ***** //

```

A.5. fvSchemes

```

/*-----* C++ *-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 2.1.1 |

```

```

|  \ \ /   A nd           | Web:      www.OpenFOAM.org           |
|  \ \ /   M anipulation |                                     |
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       fvSchemes;
}
// * * * * *

ddtSchemes
{
    default      Euler;
}

gradSchemes
{
    default      Gauss linear;
    grad(p)      Gauss linear;
}

divSchemes
{
    default      none;
    div(phi,U)   Gauss linear;
    div((nuEff*dev(T(grad(U)))) Gauss linear;
}

laplacianSchemes
{
    default      none;
    laplacian(nu,U) Gauss linear corrected;
    laplacian(rAUf,pcorr) Gauss linear corrected;
    laplacian(rAU,pcorr) Gauss linear corrected;
    laplacian(rAU,p) Gauss linear corrected;
    laplacian(rAUf,p) Gauss linear corrected;
}

```



```

    laplacian(diffusivity,cellMotionU) Gauss linear uncorrected;
    laplacian(nuEff,U) Gauss linear uncorrected;
    laplacian(diffusivity,cellDisplacement) Gauss linear uncorrected;
}

interpolationSchemes
{
    default          linear;
    interpolate(HbyA) linear;
}

snGradSchemes
{
    default          corrected;
}

fluxRequired
{
    default          no;
    pcorr            ;
    p                ;
}

// ***** //

```

A.6. fvSolution

```

/*-----*- C++ -*-----*\
| ===== |
| \\      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
| \\      / O p e r a t i o n | Version: 2.1.1 |
|  \\    / A n d           | Web:      www.OpenFOAM.org |
|   \\  / M a n i p u l a t i o n | |
\*-----*/
FoamFile
{

```

```
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       fvSolution;
}
// * * * * *

solvers
{
    pcorr
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       0.02;
        relTol          0;
    }

    p
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-06;
        relTol          0.01;
    }

    pFinal
    {
        $p;
        tolerance       1e-06;
        relTol          0;
    }

    U
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0.1;
    }
}
```

```
    }

    UFinal
    {
        $U;
        tolerance      1e-05;
        relTol         0;
    }

    "cellDisplacement.*"
    {
        solver          GAMG;
        tolerance       1e-5;
        relTol          0;
        smoother        GaussSeidel;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }

}

PIMPLE
{
    correctPhi          yes;
    nOuterCorrectors   2;
    nCorrectors        1;
    nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
    fields
    {
    }
    equations
    {
```

```

        "U.*"          1;
    }
}

// ***** //

```

A.7. dynamicMeshDict

```

/*-----*- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 2.1.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n |
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       dynamicMeshDict;
}
// ***** //

dynamicFvMesh dynamicMotionSolverFvMesh;

motionSolverLibs ( "libfvMotionSolvers.so" );

solver displacementLaplacian;

displacementLaplacianCoeffs
{
    // diffusivity uniform;
    // diffusivity directional (1 200 0);
    // diffusivity motionDirectional (1 1000 0);
    // diffusivity file motionDiffusivity;
}

```

```

    diffusivity inverseDistance 1(cylinder);
}

```

```
// ***** //
```

A.8. transportProperties

```

/*-----*- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 2.1.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n |
\*-----*-*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       transportProperties;
}
// ***** //

transportModel  Newtonian;

nu              nu [ 0 2 -1 0 0 0 0 ] 1.003e-06;

```

```
// ***** //
```

A.9. turbulenceProperties

```
/*-----*- C++ -*-----*\
```

```

| ===== | |
| \\      / F ield      | OpenFOAM: The Open Source CFD Toolbox |
| \\      / O peration  | Version: 2.1.1 |
|  \\    /  A nd        | Web:      www.OpenFOAM.org |
|   \\  /   M anipulation | |
|*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       turbulenceProperties;
}
// ***** //

simulationType laminar;

// ***** //

```