



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Aeroespacial
y Diseño Industrial

Simulación y análisis de flujos turbulentos en tuberías

Trabajo Fin de Máster

Máster Universitario en Ingeniería Aeronáutica

AUTOR/A: Piedrabuena Muñoz, Marcos

Tutor/a: Hoyas Calvo, Sergio

CURSO ACADÉMICO: 2022/2023

Abstract

The main objective of this work is to deepen the key concepts developed throughout the subject "Introduction to Supercomputing and Parallel Computing" following the learning process carried out in class. The need to learn parallel computing tools is exposed, and later a balance of the learning is made through examples taken from the daily work of everything learned in a Linux environment on which remote work will be done, as well as the process followed simultaneously with Fortran, HDF5 and OpenMP/MPI, developing codes and projects that are tested on the remote workstation. Finally, a balance is made on what has been learned.

All the source code available [here](#).

Resumen

A lo largo de este trabajo se aborda todo el proceso llevado a cabo para el desarrollo de una herramienta *software* destinada a la simulación de flujos turbulentos en tuberías a través de simulaciones DNS. En primer lugar se aborda el contexto y el estado del arte de las diferentes técnicas de estudio de flujos turbulentos describiendo el fenómeno de la turbulencia a nivel de sus características definitorias y tratando de manera específica la turbulencia de pared y los flujos canónicos. Posteriormente se desarrolla el modelo matemático sobre el que se fundamenta la simulación y todos los aspectos a la implementación de un código y el proceso de optimización de éste para lograr el desarrollo de una herramienta de simulación escalable así como su proceso de verificación llevado a cabo en equipos de supercomputación de diferente magnitud. Finalmente se exponen los resultados de la simulación tras el postprocesado e hitos alcanzados, las conclusiones del trabajo y el pliego de condiciones y presupuesto del proyecto.

Símbolos y abreviaturas

| | |
|---------------------|---|
| ω | Vorticidad. |
| ∇ | Operador nabla. |
| \times | Rotacional. |
| ρ | Densidad. |
| l_c | Longitud característica. |
| U | Velocidad. |
| μ | Viscosidad. |
| δ | Espesor de la capa límite. |
| Re_{cl} | Reynolds en la línea media. |
| U_{cl} | Velocidad en la línea media. |
| R | Radio. |
| Re_{τ} | Reynolds de fricción. |
| U_{τ} | Velocidad de fricción. |
| τ_w | Esfuerzos viscosos en la pared. |
| L_z | Longitud de la tubería. |
| α | Longitud de la tubería adimensional. |
| N_{θ} | Número de modos de Fourier azimutales. |
| N_z | Número de modos de Fourier axiales. |
| N_r | Número de puntos en dirección radial. |
| g | Constante gravitacional. |
| τ | Tensor de esfuerzos viscosos. |
| e | Energía específica. |
| $\langle U \rangle$ | Promedio de la velocidad. |
| u | Componente de perturbación de la velocidad. |
| T | Tiempo de muestreo. |
| M | Modos en dirección azimutal. |
| K | Modos en dirección axial. |
| C_{ij} | Término convectivo del tensor de esfuerzos de Reynolds. |
| Π_{ij} | Correlación de deformación de presión del tensor de Reynolds. |
| D_{ij}^P | Término de difusión depresión del tensor de Reynolds. |
| D_{ij}^T | Término de difusión turbulenta del tensor de Reynolds. |
| D_{ij}^V | Término de difusión viscosa del tensor de Reynolds. |
| ε_{ij} | Término de disipación turbulenta del tensor de Reynolds. |
| P_{ij} | Término de producción turbulenta del tensor de Reynolds. |

| | |
|------|--|
| ODS | Objetivos de desarrollo sostenible. |
| DNS | <i>Direct Numerical Simulation.</i> |
| DNS | <i>Direct Numerical Simulation.</i> |
| BC | <i>Boundary condition</i> , condición de contorno. |
| CFD | <i>Computational Fluid Dynamics.</i> |
| HPC | <i>High Performance COmputation..</i> |
| GB | <i>Gigabyte.</i> |
| IPCC | <i>Intergovernmental Panel on Climate Change.</i> |
| CERN | <i>European Organization for Nuclear Research.</i> |

Índice general

| | |
|---|-----------|
| Resumen | IV |
| Símbolos y abreviaturas | VI |
| 1. Introducción y objetivos | 1 |
| 1.0.1. 4 Educación de calidad: | 3 |
| 1.0.2. 9 Industria, innovación e infraestructuras: | 3 |
| 1.0.3. 11 Ciudades y comunidades sostenibles: | 3 |
| 1.0.4. 13 Acción por el clima: | 3 |
| 2. Turbulencia de pared. | 4 |
| 2.1. Naturaleza del estudio de flujos turbulentos | 4 |
| 2.2. Definición formal de turbulencia y descripción del fenómeno. | 5 |
| 2.2.1. Características. | 5 |
| 2.3. Turbulencia de pared. | 7 |
| 2.4. Flujos canónicos | 8 |
| 2.5. Caso de estudio de flujos turbulentos en tuberías. | 11 |
| 2.5.1. Conceptos clave. | 11 |
| 2.5.2. Detalles técnicos y configuración de la simulación. | 12 |
| 3. Modelo matemático. | 14 |
| 3.1. Modelos de estudio de la turbulencia. | 15 |
| 3.1.1. Modelo RANS (Reynolds-Averaged Navier-Stokes) | 15 |
| 3.1.2. Modelo LES (Large Eddy Simulation) | 16 |
| 3.1.3. Modelo DNS (Direct Numerical Simulation) | 16 |
| 3.2. Modelo matemático | 16 |
| 3.2.1. Resolución de la PPE. | 18 |
| 3.3. Mejoras realizadas al <i>solver</i> DNS. | 20 |
| 4. Esquema numérico | 21 |
| 4.1. Método numérico de resolución del modelo matemático. | 21 |
| 5. Desarrollo del software. | 23 |
| 5.1. Introducción a la supercomputación en turbulencia | 23 |
| 5.2. Metodología de trabajo y recursos disponibles | 24 |
| 5.2.1. Metodología de trabajo | 24 |
| 5.2.2. Recursos disponibles | 27 |
| 5.3. Cálculo en paralelo con Fortran. | 27 |

| | | |
|-----------|---|-----------|
| 5.4. | Punto de partida, problemas | 31 |
| 5.5. | Cambios del formato de gestión de datos | 32 |
| 5.5.1. | NetCDF y limitación en HPC | 32 |
| 5.5.2. | Conversión a HDF5, implementaciones realizadas y mejoras observadas. | 33 |
| 5.6. | Memoria | 33 |
| 5.6.1. | Ahorro de memoria | 33 |
| 5.6.2. | Memoria estática vs. memoria dinámica. | 35 |
| 5.6.3. | De memoria estática a dinámica | 35 |
| 5.7. | Makefile | 35 |
| 5.8. | Metodología de validación, parámetros en los que me fijo | 36 |
| 5.8.1. | Test sucesivos entre cambios. | 36 |
| 5.8.2. | Prueba de escalado intermedia. | 36 |
| 5.8.3. | Prueba de escalado con simulación a gran escala | 37 |
| 5.8.4. | Conclusión del proceso de verificación | 38 |
| 6. | Desarrollo del módulo de cálculo de las estadísticas | 39 |
| 6.1. | Objetivo | 39 |
| 6.2. | De la solución numérica a las velocidades medias y RMS | 39 |
| 6.2.1. | Resultados de la simulación. Velocidad en los dominios físico y espectral | 40 |
| 6.2.2. | Presión, método de cálculo | 41 |
| 6.3. | Metodología de exportación | 41 |
| 6.4. | Metodología para interpretación con matlab | 42 |
| 7. | Resultados | 43 |
| 7.1. | Datos de referencia | 43 |
| 7.2. | Perfiles de velocidad y presión | 43 |
| 7.2.1. | Perfiles de velocidad | 44 |
| 7.2.2. | Perfil de la presión | 49 |
| 7.3. | Turbulent budgets | 49 |
| 7.3.1. | Producción turbulenta. | 50 |
| 7.4. | Resultados de especial relevancia | 52 |
| 7.4.1. | Hitos en resolución espacial. | 53 |
| 7.4.2. | Modificación sobre la extensión de la subcapa logarítmica en la capa límite | 55 |
| 7.4.3. | Hallazgo de un criterio de convergencia objetivo conforme al <i>Eddie Turnover Time</i> (ETT) | 56 |
| 8. | Pliego de condiciones y presupuesto | 58 |
| 8.1. | Condiciones técnicas y materiales | 58 |
| 8.2. | Condiciones del puesto de trabajo | 59 |
| 8.3. | Presupuesto | 60 |
| 8.3.1. | Coste de personal | 60 |
| 8.3.2. | Coste de equipo informático | 60 |
| 8.3.3. | Presupuesto total del proyecto | 61 |

| | |
|----------------------------|-----------|
| 9. Trabajos futuros | 62 |
| 10. Conclusiones | 63 |
| Bibliografía | 66 |

Índice de figuras

| | | |
|------|--|----|
| 1.1. | Objetivos de Desarrollo Sostenible de la ONU. Fuente: ONU | 2 |
| 2.1. | Ejemplos de fenómenos naturales representativos de flujos turbulentos. Créditos: (a) Wandering bird. (b) Jeff Schmaltz, LANCE/EOSDIS Rapid Response/NASA | 4 |
| | (a). | 4 |
| | (b). | 4 |
| 2.2. | Ejemplo de estructuras vorticales (vortex structures). Fuente: Nortek Group | 5 |
| 2.3. | Ejemplo de autosemejanza en estructuras turbulentas. Fuente: Steven Mathey | 6 |
| 2.4. | Esquema del mecanismo de transmisión de energía turbulenta. Fuente: Altair | 7 |
| 2.5. | Transición de capa límite laminar a turbulenta. [Schlichting, 2000] | 8 |
| 2.6. | Simulación DNS del flujo canónico correspondiente a una placa plana. [Wenzel et al., 2019] | 9 |
| 2.7. | Ejemplo de flujo canónico en canales, se visualiza la velocidad instantánea, en la geometría aparecen las dimensiones y condiciones de contorno correspondientes. [Hamzehloo et al., 2021] | 10 |
| 2.8. | Contornos de velocidad representados para el caso de estudio del flujo canónico en tuberías. [Pirozzoli et al., 2022] | 11 |
| 3.1. | Esquema del dominio fluido con direcciones principales. Créditos: [Zhou et al., 2007] | 17 |
| 5.1. | Equipo de supercomputación SuperMUC, empleado en este trabajo para tareas de testeo y validación del <i>software</i> . Fuente: Leibniz Supercomputing Centre | 25 |
| 5.2. | Diferencia entre serie y paralelo. Fuente: Python Numerical Methods | 28 |
| 5.3. | Ejemplo de arquitectura MPI. Fuente: [?] | 29 |
| 5.4. | Estructura jerárquica de los datos en HDF5. Fuente: https://soloelectronicos.com | 30 |
| 5.5. | HDF5 aún a características destacables de otros formatos. Fuente: Apuntes de clase. | 31 |
| 6.1. | Descomposición de Reynolds. [Reggente, 2014] | 40 |
| 7.1. | Como se puede observar en esta representación la componente radial de la velocidad promediada, se corrobora que es nula como cabría esperar, siendo obtenido únicamente ruido numérico | 44 |

| | | |
|-------|--|----|
| 7.2. | Se observa que en la región más cercana de la línea media, la componente de perturbación se suaviza, dado que el flujo en esa región se encuentra poco influido por la pared y se estabiliza. En la región cercana a la pared ($1 - r = 0,2$), se observa que la perturbación de la velocidad radial es del orden de un 60 % la velocidad de fricción y siendo ésta máxima para $1 - r = 0,3$, donde se deduce al observar el resto de componentes perturbadas que se sitúa una región con alta intensidad turbulenta, en la pared ($1 - r = 0$) debido a la condición de no deslizamiento encontramos valores nulos. . . . | 45 |
| 7.3. | De manera similar a cómo ocurre con la componente radial, el promedio de la velocidad azimutal se considera ruido numérico. | 46 |
| 7.4. | Para el caso de la velocidad de perturbación de la componente azimutal de la velocidad se observa el fenómeno que ya se presentó para la el caso de la velocidad radial, en la región entre $r = 0,2$ y $r = 0,3$ se puede encontrar una región de alta intensidad turbulenta. | 46 |
| 7.5. | En esta componente se puede observar que el orden de magnitud es superior a las demás debido a que es la componente principal del flujo. En la pared es nula debido a la condición de no deslizamiento y es creciente a medida que nos acercamos a la región central, donde el flujo se encuentra menos perturbado. | 47 |
| 7.6. | En esta figura se puede observar de igual forma que el orden de magnitud es superior al de las componentes de perturbación de las otras direcciones principales del flujo. La tendencia es la opuesta a la de la media, pues en regiones cercanas a la pared donde se acentúan más la influencia de ésta esta componente de la velocidad al igual que las demás es donde más se desvía de la media debido a la alta intensidad turbulenta. A medida que observamos regiones cercanas al centro de la tubería, el flujo se encuentra menos perturbado. | 48 |
| 7.7. | Se presentan la presión media y su perturbación, a través de las cuales se puede identificar una región en torno al eje axial de la tubería en el que se observa que la presión fluctúa generando zonas de bajas presiones. | 49 |
| 7.8. | Se observa únicamente la componente axial del término de producción de energía turbulenta debido a que las otras dos son nulas. Como se puede inferir, en la región más cercana a la pared y en la que a partir de la observación de la perturbación de la velocidad hemos visto que es de alta intensidad turbulenta, es donde más energía turbulenta se produce, disminuyendo asintóticamente a 0 en las regiones próximas al eje axial. | 50 |
| 7.9. | En esta figura se pueden observar las tres componentes de la disipación viscosa, donde además se pueden observar las diferencias existentes entre sus órdenes de magnitud. Su distribución radial indica que tienen una mayor presencia en regiones muy próximas a la pared, nótese $1 - r$ en escala logarítmica. | 51 |
| 7.10. | Por último se presenta la magnitud que indica la intensidad con la que la que los efectos viscosos se difunden de la pared al centro de la tubería. De la misma manera que ocurre con la disipación, tiene una mayor presencia en regiones muy próximas a la pared. | 52 |

7.11. En esta figura se observa la distribución radial del espaciado entre puntos del dominio de cálculo para una simulación con $Re_\tau = 550$. En línea azul discontinua se representa el espaciado entre puntos, siendo éste aproximadamente un 50% más grande que la longitud de Kolmogorov. Por otro lado, se puede observar una línea continua de color azul que representa la distribución del espaciado entre puntos de la malla *LR*, *Low Resolution*, representativo de la resolución espacial del problema y por lo tanto, de la precisión de los resultados calculados. Por último, se muestra en color rojo una línea continua representativa del espaciado de malla para un caso de estudio de mayor resolución *HR*, *High Resolution* donde, como se puede observar que todos los puntos de la malla se encuentran entre sí a una distancia inferior a la mitad de la longitud de Kolmogorov, lo que se considera un hito y garantía de resultados precisos. 53

7.12. Se puede observar en la escala de la izquierda el espaciado de malla para un canal con $Re_\tau = 550$ y a la derecha el espaciado para el caso de una tubería. En este caso las distribuciones se comparan con respecto a la longitud de Corrsin, más grande que la longitud de Kolmogorov pero con la diferencia de que L_C tiene en cuenta la deformación turbulenta. Una vez más se observan que las distribuciones del espaciado entre puntos para el caso de las mallas generadas para el caso de baja resolución (*LR*) son del orden de entre el 5% y el 7% de la longitud de Corrsin y para los casos de alta resolución (*HR*), el espaciado entre puntos se reduce al entorno dle 3% de L_C , con lo que se anticipa que estas mallas de tan alta resolución espacial son capaces de permitir simular resultados con precisión. 54

7.13. En este caso se observa cómo se ha inferido acerca de la existencia de una ley de ajuste para la subcapa logarítmica como una combinación de un término logarítmico y otro lineal, ajustándose mejor que un ajuste puramente logarítmico y se ha hipotetizado sobre la no universalidad de la constante de von Karman, pues como se observa, según el tipo de ajuste que se haga (observando un mejor ajuste para solape logarítmico-linal), del código y de la relación de aspecto de los elementos del dominio, se obtiene un valor de los parámetros u otro. 55

7.14. En esta figura se observan los dos ajustes, en línea naranja discontinua (sin considerar términos lineales) se observa que la región de ajuste es menor que si se emplea un ajuste logarítmico-lineal, representado por la línea continua naranja. Esto aporta una perspectiva adicional que logra mejores resultados de los logrados con la metodología existente, hasta ahora 56

7.15. 57

Capítulo 1

Introducción y objetivos

Los flujos turbulentos que están limitados por una pared (conocidos como 'turbulencia de pared') son comunes en la naturaleza y la tecnología. Aproximadamente la mitad de la energía gastada en transportar fluidos a través de tuberías o vehículos a través del aire o el agua se disipa debido a la turbulencia cerca de las paredes (Jiménez, 2012). Por lo tanto, comprender mejor la física subyacente de estos flujos es esencial para la modelación y el control (Kim, 2011; Canton et al., 2016; Yao, Chen y Hussain, 2018). La capa límite en evolución espacial, el canal (plano) y la tubería son tres configuraciones geométricas canónicas de la turbulencia de pared. A diferencia de los flujos en capa límite y canal, la periodicidad azimutal es inherente a los flujos en tubería. Por lo tanto, el flujo en tubería es el caso más canónico, y queda completamente descrito por el número de Reynolds (Re) y la longitud axial; el efecto de este último es limitado si es suficientemente grande (El Khoury et al., 2013; Feldmann, Bauer y Wagner, 2018).

A lo largo de este trabajo se tratará en profundidad el proceso de desarrollo de una herramienta de simulación especializada en resolver mediante la técnica *Direct Numerical Simulation* flujos turbulentos en tuberías, donde cobra un papel importante el factor de la multidisciplinaridad pues son necesarios conocimientos en matemáticas, física estadística, técnicas de mecánica de fluidos computacional, métodos numéricos y técnicas de supercomputación.

Objetivos

El objetivo principal es el de lograr desarrollar una herramienta de simulación DNS para el estudio de la turbulencia en tuberías haciendo uso de técnicas de supercomputación.

Para ello es necesaria la consecución de estos objetivos intermedios:

- Aprender acerca de la física de la turbulencia.
- Aprender técnicas de estudio de la turbulencia.
- Desarrollar una herramienta que ayude en el desarrollo del estudio de la turbulencia.
- Aprender a programar en Fortran a nivel de supercomputación.
- Conocer métodos de paralelización y su uso a través de librerías.

- Conocer y desarrollar *software* en formatos específicos para el trabajo con *sets* de datos de gran volumen.
- Desarrollar herramientas de postproceso avanzadas con Matlab.
- Manejar equipos destinados a la computación avanzada.
- Conseguir un código escalable para poder ejecutar una simulación de alta demanda de recursos computacionales en un superordenador.
- Establecer una metodología de trabajo revisada semanalmente para cumplir con los objetivos.

Además, este trabajo está íntimamente relacionado con los Objetivos de Desarrollo Sostenible (ODS) en los siguientes puntos.

Objetivos de Desarrollo Sostenible

Este trabajo se desarrolla en el marco de los objetivos de desarrollo sostenible marcados por la Organización de las Naciones Unidas como parte de la Agenda 2030 para el Desarrollo Sostenible.

Esta agenda presenta un plan de acción que marca una serie de pautas a favor de las personas, el planeta y la prosperidad y que también tiene como fin fortalecer la paz universal y favorecer el acceso a la justicia.

Este trabajo al tener como finalidad el desarrollo de una herramienta de simulación para estudiar el fenómeno turbulento y que este fenómeno es meramente disipativos en términos energéticos, el desarrollo de este trabajo colabora en el proceso de generación de conocimiento nuevo para desarrollar métodos de control que actúen sobre la turbulencia para disminuir su impacto y mejorar la eficiencia energética en procesos de transporte de fluidos por el interior de conductos.



Figura 1.1: Objetivos de Desarrollo Sostenible de la ONU. Fuente: ONU

Este trabajo se alinea con los siguientes ODS:

1.0.1. 4 Educación de calidad:

Pues con el desarrollo de esta herramienta se ha generado material de estudio y publicaciones de carácter científico, expuestas en diversos congresos con la finalidad de tener la mayor difusión y que el conocimiento sea más accesible. Además se cede el código fuente a la Universidad con la finalidad de complementar la formación de aquellos estudiantes que asuman el rol de desarrolladores en un futuro.

1.0.2. 9 Industria, innovación e infraestructuras:

Debido a que aproximadamente la mitad de la energía destinada a forzar un fluido por una tubería se disipa debido a la turbulencia, una mayor comprensión del fenómeno y la consecuente mejora en la eficiencia repercutiría directamente en el desarrollo industrial, favoreciendo la innovación en los procesos de transporte y la adaptación de las infraestructuras.

1.0.3. 11 Ciudades y comunidades sostenibles:

De encontrarse un método de control de la disipación turbulenta, se podrían implementar mejoras en sistemas como el de distribución de agua y gas en los hogares para hacer el proceso más eficiente y sostenible.

1.0.4. 13 Acción por el clima:

Pues al reducir los costes energéticos del proceso de transporte de un fluido por el interior de una tubería se podría hacer con un menor coste en términos medioambientales.

Capítulo 2

Turbulencia de pared.

A lo largo de este primer capítulo introductorio se resumirá la historia del estudio de la turbulencia y se presentarán todos aquellos conceptos más relevantes para el caso concreto del estudio de la turbulencia de pared y dentro de éste, de la turbulencia en tuberías.

2.1. Naturaleza del estudio de flujos turbulentos

A lo largo de la historia, la observación de la naturaleza ha brindado al ser humano múltiples oportunidades para estudiar y comprender los flujos turbulentos en situaciones cotidianas. Estas observaciones han contribuido a la formación y el desarrollo de la teoría de la turbulencia, así como a la identificación de patrones y características típicas de los flujos turbulentos en diversos contextos, pero en primer lugar, en la naturaleza a través de la observación de fenómenos hidrológicos y atmosféricos, entre otros.



Figura 2.1: Ejemplos de fenómenos naturales representativos de flujos turbulentos.
Créditos: (a) Wandering bird. (b) Jeff Schmaltz, LANCE/EOSDIS Rapid Response/NASA

El estudio formal de los flujos turbulentos se remonta a varios siglos atrás, pero se intensificó a lo largo del siglo XX con el avance de la teoría de la mecánica de fluidos y el desarrollo de técnicas experimentales y numéricas más sofisticadas.

También a nivel industrial existe una preocupación por lograr entender y controlar este tipo de flujos. Pues un dato especialmente relevante es que aproximadamente la mitad de la energía cinética del flujo que discurre por el interior de una tubería —caso de

estudio actual—, se disipa debido a la turbulencia, lo que plantea un problema, que de solucionarse tendría implicación directa en un incremento notable de eficiencia de todos aquellos procesos, industriales o no, en los que se pueden encontrar flujos en régimen turbulento.

Este hecho tan reseñable ha motivado numerosos estudios centrados en el estudio y control de flujos turbulentos empleando las técnicas más novedosas de análisis matemático, técnicas de *big data* e inteligencia artificial.

2.2. Definición formal de turbulencia y descripción del fenómeno.

La turbulencia se puede definir formalmente como una propiedad de una corriente fluida en el cual las características de velocidad y presión varían de manera impredecible y caótica en el tiempo y el espacio que es debido a la interacción compleja entre las fuerzas inerciales y las fuerzas viscosas en el seno del fluido o entre el fluido y el medio externo. En los flujos turbulentos, se forman estructuras vorticales a diversas escalas espaciales 2.2, dando lugar a una mezcla intensa de las partículas de fluido y a una disipación significativa de la energía cinética del flujo en forma de calor.

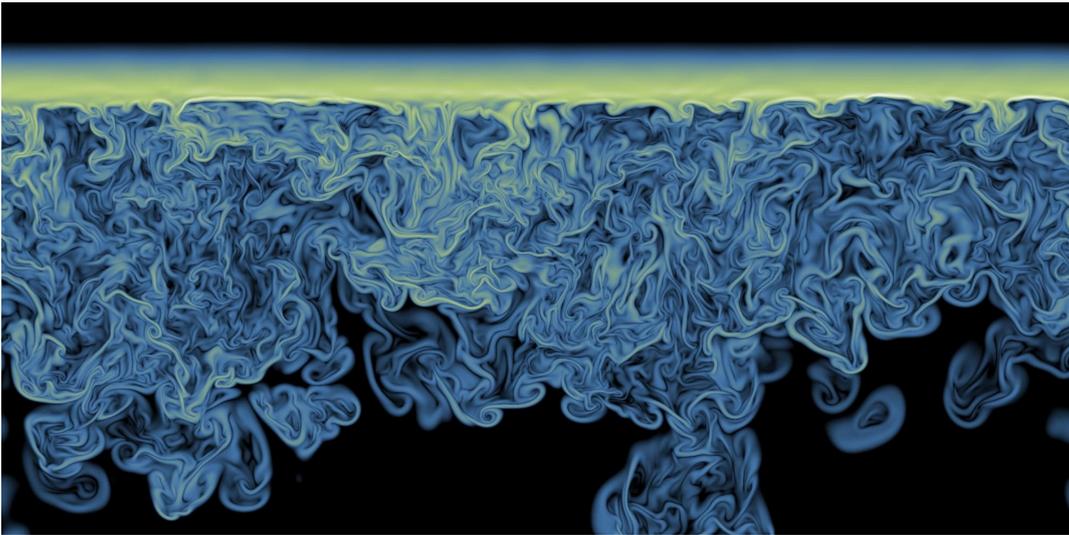


Figura 2.2: Ejemplo de estructuras vorticales (vortex structures).
Fuente: Nortek Group

Esta definición formal se ha llevado a cabo tras la revisión de literatura sobre fundamentales de la mecánica de fluidos y la teoría de la turbulencia [Pope, 2000] [Tennekes and Lumley, 1972] [Batchelor, 1967].

2.2.1. Características.

A continuación se expone una relación de las diferentes características definitorias de un flujo turbulento [Castro and Vanderwel, 2021].

Naturaleza caótica.

Esta propiedad intrínseca de los flujos turbulentos los hace exhibir comportamientos complejos, impredecibles y altamente sensibles a condiciones iniciales y pequeñas perturbaciones. Esta característica hace que sea extremadamente difícil predecir con precisión el comportamiento a largo plazo de un flujo turbulento, aunque se conozcan las ecuaciones que gobiernan el flujo.

Esto hace que la evolución espacio-temporal del flujo carezca de un patrón predecible y, aunque pueden observarse ciertas tendencias y características estadísticas, la naturaleza caótica del flujo impide la predicción precisa de su comportamiento a largo plazo.

Naturaleza multiescala.

Esta propiedad alude a la presencia de una variedad de estructuras turbulentas y patrones de flujo que ocurren en una amplia gama de escalas espaciales y temporales en un flujo turbulento. Es decir, los flujos turbulentos exhiben comportamientos y características que se desarrollan en múltiples niveles de frecuencias espacial y temporal, desde las grandes estructuras de flujo del orden del tamaño característico del problema hasta pequeños torbellinos a escala submilimétrica.

Además, existe autosemejanza en el flujo Figura 2.3 —las mismas características o patrones de flujo se repiten a diferentes niveles de magnificación, con presencia de estructuras vorticales que interactúan y se desarrollan a múltiples escalas espaciales— con lo que las estructuras turbulentas exhiben un comportamiento fractal.



Figura 2.3: Ejemplo de autosemejanza en estructuras turbulentas.
Fuente: Steven Mathey

Naturaleza vortical, tridimensionalidad.

Esta propiedad hace referencia a la vorticidad no nula del flujo (en otras palabras, no es irrotacional). La vorticidad en un fluido se define como el rotacional del campo vectorial de velocidades, i.e. $\omega = \nabla \times \vec{U}$. Esta magnitud, nula en flujos laminares, deja de serlo en el caso de que el flujo sea turbulento. Además, la vorticidad es relevante en el estudio de la turbulencia ya que está ligado al carácter tridimensional del mismo, pues en cualquier punto del dominio espacial en cualquier instante temporal, el vector velocidad (y vorticidad) puede estar orientado en cualquiera de las tres dimensiones espaciales. Además el vector vorticidad puede estirarse y comprimirse en cualquiera de estas tres direcciones,

fenómeno que deja de ocurrir si el dominio espacial considerado es bidimensional, de ahí que la turbulencia tenga naturaleza tridimensional y rotacional o vortical.

Naturaleza disipativa.

Es esencial comprender que la turbulencia disipa energía. A medida que los movimientos en escalas más pequeñas extraen energía de escalas más grandes a través del proceso de "cascada", esta se transmite a movimientos aún más pequeños, hasta que las escalas más pequeñas son de tal magnitud que son influenciadas por efectos viscosos y, en última instancia, pierden energía en forma de calor.

Se produce un flujo de energía en todas las escalas, desde las más grandes hasta las más pequeñas. Este proceso de cascada es fundamental en los flujos turbulentos.

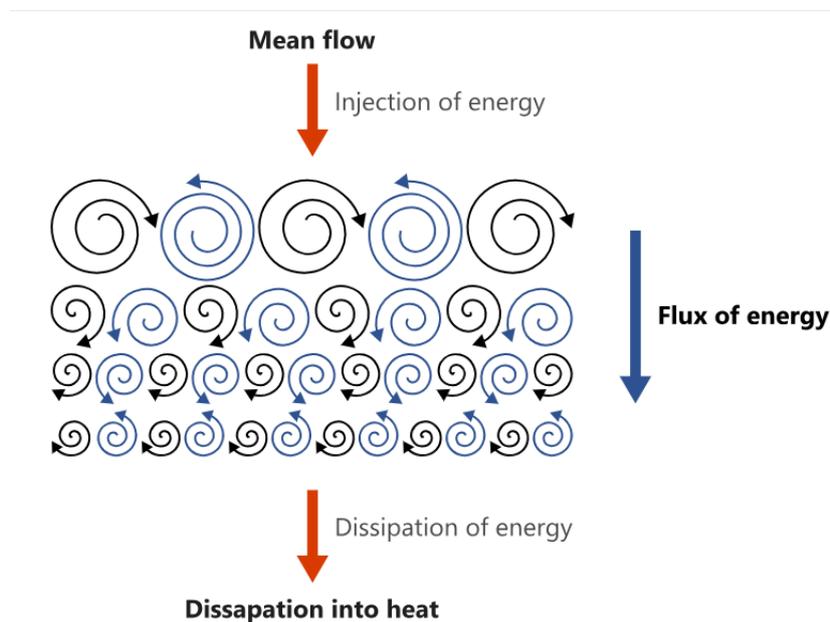


Figura 2.4: Esquema del mecanismo de transmisión de energía turbulenta. Fuente: Altair

En cuanto al origen y destino de esta energía en un flujo turbulento, la energía en las escalas más grandes de movimiento se extrae del flujo promedio y se disipa en forma de calor en las escalas más pequeñas, como se muestra en Figura 2.4.

2.3. Turbulencia de pared.

La turbulencia de pared se define como un tipo de flujo turbulento que ocurre cerca de una superficie sólida, donde la viscosidad del fluido tiene una influencia significativa en la dinámica del flujo [Pope, 2000]. En la turbulencia de pared, aparece un fenómeno característico: la capa límite.

La capa límite es una estructura fluida característica de flujos que transcurren cerca de una superficie, ya que tiene origen entre la interacción flujo-pared. En esta interacción, la fricción viscosa juega un papel relevante, pues es la encargada de disipar energía cinética del flujo. Este fenómeno se traduce en que perpendicularmente desde la pared, se genera

un perfil de velocidades en el flujo que puede tener diversa morfología, pudiéndose tratar de una capa límite laminar o turbulenta en función del número de Reynolds del flujo.

El número de Reynolds es un balance entre los efectos de origen inercial y los efectos viscosos, representando el orden de magnitud de uno sobre otro 2.2.

$$Re = \frac{l_c \rho U}{\mu} \quad (2.1)$$

Donde l_c es la longitud característica del problema, ρ la densidad del fluido, U la velocidad y μ la viscosidad del fluido.

En la región del flujo más cercana a la pared, donde predominan los esfuerzos viscosos y consecuentemente el número de Reynolds es del orden de la unidad, se comprueba que en la interfaz de ambos medios, la velocidad del flujo es nula, generando un gradiente de velocidades en sentido perpendicular a la superficie y que están asociados a unos esfuerzos tangenciales que generan una desviación en las líneas de corriente del flujo, por lo que se asocia a un flujo desordenado y caótico (aunque ese aparente desorden no es condición suficiente para el flujo turbulento). Por otro lado, el flujo laminar es aquel en el que la ausencia de esfuerzos tangenciales mantienen las líneas de corriente paralelas entre sí, no hay recirculación y se observa un flujo con aparente orden.

Para el caso del estudio de la turbulencia, la capa límite en origen laminar, atraviesa un proceso de transición hasta volverse turbulenta, como se puede observar en la Figura 2.5.

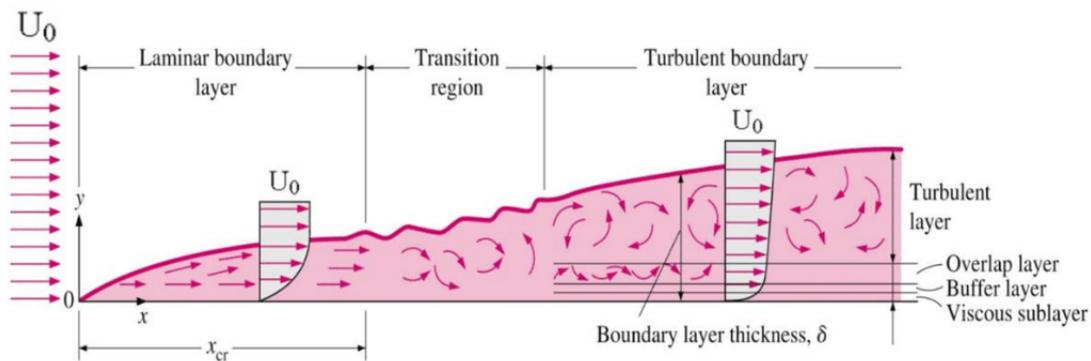


Figura 2.5: Transición de capa límite laminar a turbulenta. [Schlichting, 2000]

2.4. Flujos canónicos

A lo largo del proceso de estudio de la turbulencia como fenómeno se han optado por diversidad de experimentos llevados a cabo tanto en laboratorios físicos como a través de simulaciones CFD con la finalidad de comprender la física y las dinámicas del flujo turbulento o para elaborar modelos matemáticos precisos capaces de replicar efectos de comportamientos físicos complejos en este tipo de flujos. El punto de partida de estos estudios se encuentra en el estudio de los llamados flujos canónicos, que no son más que experimentos con flujos representativos de la turbulencia de pared que se desarrollan en

geometrías sencillas, de tal manera que estos flujos se desarrollan en condiciones controladas, conocidas, fácilmente replicables y simplificadas. A través de la experimentación con estos flujos canónicos, se puede examinar cómo se forman y evolucionan las estructuras turbulentas, así como cómo la energía se disipa y se transfiere a diferentes escalas.

- **Placa plana:** en este caso la geometría es lo más sencilla posible, pues el flujo se desarrolla cerca de una superficie sólida y plana. En este caso, se pueden estudiar diversos fenómenos como la formación de la capa límite, la transición de capa límite laminar a turbulenta o la interacción entre estructuras turbulentas, además de servir para la elaboración de modelos de turbulencia especialmente en cálculo de coeficientes de los mismos. De especial relevancia en el campo de la aerodinámica.

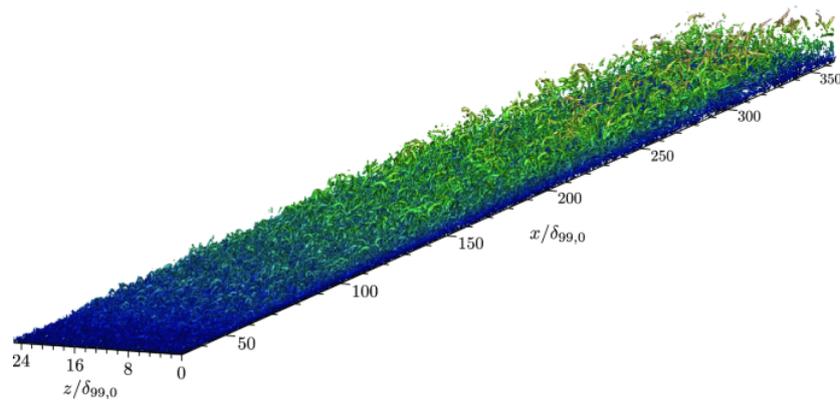


Figura 2.6: Simulación DNS del flujo canónico correspondiente a una placa plana. [Wenzel et al., 2019]

- **Canal:** el flujo discurre entre dos placas planas y paralelas. Este tipo de flujo es especialmente interesante porque se caracteriza por su naturaleza bidimensional en dos direcciones—las paralelas a las placas— y unidimensional en la tercera—la perpendicular— con ello, al ser bidimensional en dos direcciones, el flujo de canal presenta algunas propiedades especiales, como la organización de las estructuras turbulentas turbillonarias alargadas en el plano de las placas. Se ha estudiado en detalle cómo las estructuras turbulentas en las capas límite que aparecen cerca de las paredes interactúan y se mezclan en el plano medio del canal ya que la simplicidad de esta configuración geométrica permite un análisis más enfocado en la interacción de los torbellinos y las estructuras turbulentas que aparecen.

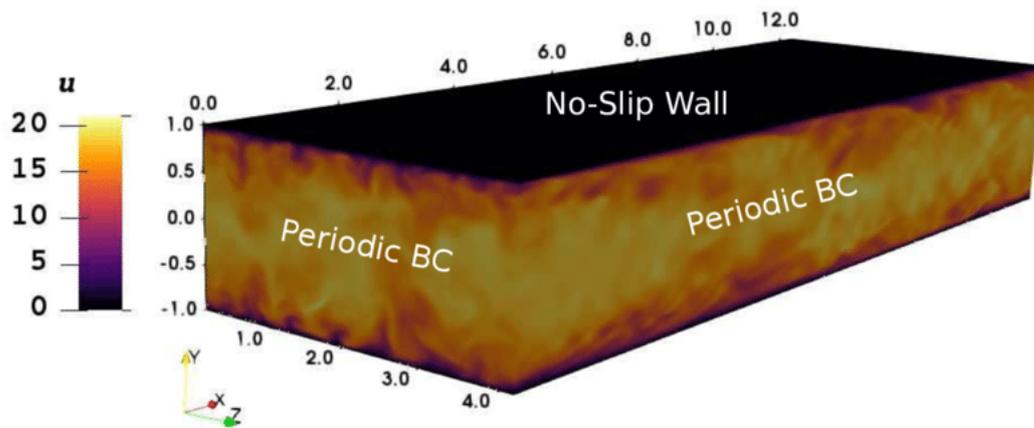


Figura 2.7: Ejemplo de flujo canónico en canales, se visualiza la velocidad instantánea, en la geometría aparecen las dimensiones y condiciones de contorno correspondientes. [Hamzehloo et al., 2021]

- **Tubería:** En el flujo de tubería, un fluido discurre a través de una tubería cilíndrica. Este flujo es interesante porque está confinado en todas las direcciones, lo que agrega mayor “tridimensionalidad” al flujo en comparación con el flujo de canal. Empleando esta geometría, se puede estudiar cómo las diferentes condiciones iniciales y geométricas influyen en la turbulencia proporcionando información sobre la distribución de velocidades, presiones y estructuras turbulentas presentes y que condicionan el comportamiento del flujo, todo ello con una geometría común a gran diversidad de soluciones ingenieriles e industriales, por lo que se estudio permitiría mejorar muchas de esas soluciones propuestas en pos de una mejora en la eficiencia del proceso de transporte de masa por su interior.

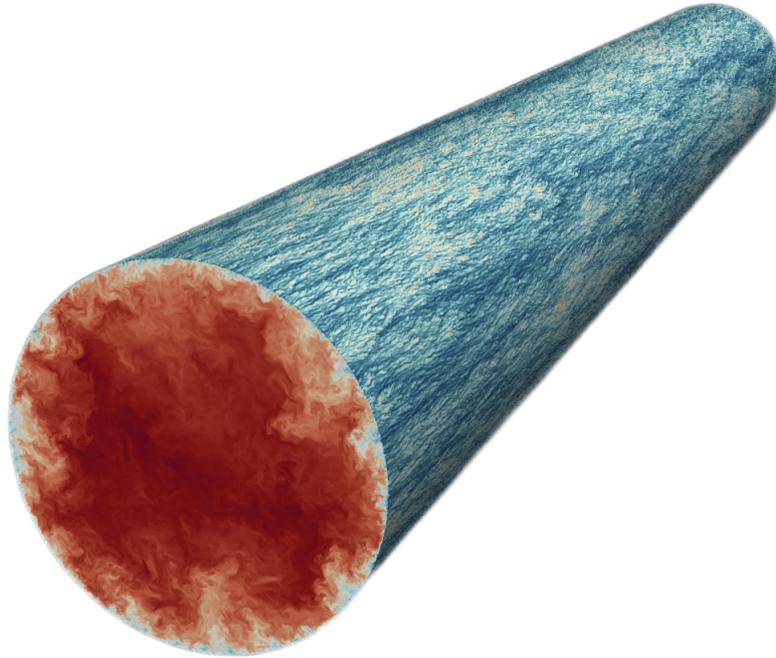


Figura 2.8: Contornos de velocidad representados para el caso de estudio del flujo canónico en tuberías. [Pirozzoli et al., 2022]

2.5. Caso de estudio de flujos turbulentos en tuberías.

En esta sección se abordarán los conceptos clave que hay tras la física del problema así como los detalles técnicos y configuración de la simulación.

2.5.1. Conceptos clave.

Como se ha mencionado anteriormente, el flujo por el interior de una tubería presenta simetría en la dirección azimutal, dependiendo únicamente del número de Reynolds característico y de la longitud de ésta, dependiendo únicamente del número de Reynolds para longitudes de tubería suficientemente grandes. En cuanto al fluido que discurre por el interior de la tubería se trata de un fluido genérico incompresible.

La longitud característica del problema es el radio R , cuyo valor será unitario.

Número de Reynolds en la línea media.

$$Re_{cl} = \frac{R\rho U_{cl}}{\mu} \quad (2.2)$$

Donde U_{cl} representa la velocidad en la línea media de la tubería.

Velocidad de fricción.

Es una velocidad característica en problemas de turbulencia de pared, se define a través de los esfuerzos tangenciales locales en la pared.

$$U_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad (2.3)$$

Número de Reynolds de fricción.

Es el número de Reynolds calculado con la velocidad de fricción. En ingeniería sirve para determinar si un flujo será predominantemente laminar o turbulento.

$$Re_\tau = \frac{\delta \rho U_\tau}{\mu} \quad (2.4)$$

Donde δ representa una longitud característica, que en el caso de tuberías es el diámetro hidráulico.

Longitud adimensional de la tubería.

Para el caso de estudio, con condiciones de contorno periódicas, se define un parámetro adimensional α .

$$\alpha = \frac{2\pi}{L_z} \quad (2.5)$$

Siendo L_z la longitud característica en dirección longitudinal de la tubería, que en el caso de estudio será de $L_z = 10\pi$.

Adimensionalización “en mases” (+)

En los casos de estudio de análisis de flujos turbulentos es de especial importancia referir los resultados a las magnitudes relevantes en el flujo turbulento. Estas referencias se hacen a través de la adimensionalización de los resultados con el número de Reynolds cuando se tratan de longitudes (téngase en cuenta que en regiones muy cercanas a la pared, de especial interés, la magnitud de las distancias es mínima). De igual manera ocurre con las velocidades, adimensionalizadas con la velocidad de fricción.

2.5.2. Detalles técnicos y configuración de la simulación.

En cuanto a los detalles técnicos, se debe mencionar que se hará uso de una simulación de referencia típica, a un número de Reynolds bajo. La razón que hay detrás es que a mayores números de Reynolds, mayor velocidad del flujo y menor tamaño de las estructuras turbulentas que aparecen [Pope, 2000], con lo que la resolución espacial de la malla ha de incrementarse, incrementando consigo el consumo de memoria y los tiempos de cálculo, por lo que interesa simular un problema lo menos demandante posible en términos computacionales durante la etapa de desarrollo, en la que se llevarán a cabo múltiples simulaciones.

Por aportar una referencia, para la simulación base se impone un $Re_{cl} = 5300$, con el que se obtiene en la pared un $Re_{\tau} = 180$, que es el caso de referencia en simulaciones de flujos en tuberías y canales más ampliamente reproducida y estudiada por ser el primer caso simulado históricamente.

Con estos valores, para obtener una malla con suficiente resolución espacial para representar de manera precisa el campo de velocidades $N_z = 1024$ modos de Fourier en dirección longitudinal de la tubería, $N_{\theta} = 256$ en dirección azimutal y $N_r = 192$ puntos en dirección radial, donde se aplican diferencias finitas. Además, por motivos de *dealiasing* es necesario colocar $\frac{3}{2}N_z$ puntos en dirección axial y $\frac{3}{2}N_{\theta}$, lo que hace que entre el dominio de cálculo y las matrices y vectores asociadas al proceso de cálculo de la simulación, se requieran ~ 40 GB de memoria RAM.

Además con el fin de establecer otra referencia, por cada vez que se duplique el Re_{τ} del problema, el coste computacional se multiplica por un factor de 7, lo que da un orden de magnitud de la complejidad y de los requerimientos técnicos tan elevados de simulaciones DNS así como se establece una motivación para optimizar los algoritmos al máximo, con el fin de minimizar estos costes.

Capítulo 3

Modelo matemático.

El estudio de la turbulencia en la actualidad se basa en el resultado de la aplicación de las ecuaciones fundamentales de la mecánica de fluidos (ecuaciones de Navier-Stokes) en cualquiera de los flujos canónicos expuestos anteriormente Sección 2.4.

Las ecuaciones de Navier-Stokes generalizadas son las siguientes:

- Ecuación de continuidad (conservación de la masa):

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

- Ecuación de conservación de la cantidad de movimiento:

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g}$$

- Conservación de la energía:

$$\frac{\partial(\rho e)}{\partial t} + \nabla \cdot (\rho e \mathbf{v}) = -\nabla \cdot \mathbf{q} + \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{v}) + \rho \mathbf{v} \cdot \mathbf{g}$$

A su vez, a la hora de aplicar dichas ecuaciones, es necesario destacar que se pueden aplicar algunas hipótesis que las simplifican, como se indicará a lo largo del desarrollo del modelo matemático. Además, debido al elevado coste computacional de su aplicación directa—recuérdese que se trata de un sistema de ecuaciones en derivadas parciales no lineales que han de aplicarse sobre una malla muy refinada para obtener la precisión y los resultados adecuados—, surgen diferentes maneras de hacerlo, según el grado de detalle con el que se quiera simular el flujo turbulento.

Para poder emplear las Capítulo 3 se descompone el campo de velocidades en dos términos. A esta descomposición se le denomina descomposición de Reynolds y es un concepto fundamental en el análisis de flujos turbulentos. Esta descomposición se basa en dividir una variable del problema —en este caso, cada una de las tres componentes de la velocidad instantánea en cada punto del espacio $\mathbf{v}(\mathbf{x}, t) = (u, v, w)$, análogamente se hace con la presión— en dos partes: una parte promediada o media y una parte fluctuante o turbulenta.

$$U_i = \bar{U}_i + u \quad (3.1)$$

Donde \bar{U}_i es el promedio temporal de la variable i.e.

$$\bar{U}_i = \langle U_i \rangle = \frac{1}{T} \lim_{T \rightarrow \infty} \int_{t_0}^{t_0+T} U_i dt \quad (3.2)$$

Donde U_i es la variable total y \bar{U}_i y u el valor medio y la perturbación instantánea de la variable respectivamente.

Esta descomposición es especialmente útil en el contexto de los flujos turbulentos, donde las velocidades y otras propiedades del fluido son variables aleatorias en el tiempo y el espacio y de esta manera, se facilita un análisis haciendo uso de elementos de análisis estadísticos, como se mostrará en el apartado de postprocesado de la simulación y resultados.

Por otro lado, el estudio de la turbulencia actualmente se centra mayoritariamente en el caso de flujos incompresibles debido a razones de complejidad y reducción de la inmensidad de recursos computacionales, necesarios en caso contrario. Estas hipótesis simplifican las ecuaciones, pues la densidad, ρ permanece constante y las fuerzas másicas se consideran nulas $\mathbf{g} = 0$, con lo que deja de ser necesaria la ecuación de la energía y las ecuaciones de conservación de la masa y de la cantidad de movimiento se reducen a:

$$\begin{cases} \nabla \cdot \mathbf{v} = 0 \\ \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \cdot \mathbf{v} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v} \end{cases} \quad (3.3)$$

3.1. Modelos de estudio de la turbulencia.

Una vez obtenidas las ecuaciones de la mecánica de fluidos para fluidos incompresibles, se exponen las distintas variantes según el grado de interés en recoger el carácter turbulento del flujo en los resultados.

3.1.1. Modelo RANS (Reynolds-Averaged Navier-Stokes)

El modelo RANS es de los más utilizados en CFD industrial para la simulación de flujos turbulentos. Estos modelos resuelven las ecuaciones de Navier-Stokes promediadas en el tiempo y utilizan modelos de turbulencia para representar los efectos turbulentos [Pope, 2000]. La utilización de estos modelos de turbulencia conlleva incluir una ecuación más al sistema.

Aplicando la descomposición de Reynolds y promediando en el tiempo ambas expresiones de la Ecuación 3.3, en cada una de las tres direcciones del flujo, se obtienen las ecuaciones correspondientes.

Los modelos $k-\epsilon$ y $k-\omega$ son ejemplos populares que describen la transferencia de energía y la disipación en turbulencia y se resuelven en conjunto con las de conservación de masa y cantidad de movimiento.

3.1.2. Modelo LES (Large Eddy Simulation)

es una técnica que resuelve las ecuaciones de Navier-Stokes para las escalas más grandes de la turbulencia y modela las escalas más pequeñas, conocidas como submalla [Pope, 2000]. La LES es más precisa en la captura de estructuras de flujo a gran escala, como vórtices, y se utiliza para estudiar flujos con alta resolución espacial y temporal, pero sin la gran necesidad de recursos computacionales de la que precisa la simulación DNS.

3.1.3. Modelo DNS (Direct Numerical Simulation)

En este caso, se resuelven completamente las ecuaciones de Navier-Stokes para todas las escalas de la turbulencia, lo que proporciona una descripción detallada y precisa del flujo. Sin embargo, debido a los altos requisitos computacionales, la DNS solo es posible para flujos con bajo número de Reynolds y en condiciones específicas. La técnica empleada en este trabajo para estudiar el flujo turbulento en tuberías será la de simulación DNS.

DNS, MODELOS numéricos (openpipeflow), cosas que se usarán e introducción a la parte computacional.

A lo largo de este capítulo se desarrollarán las ecuaciones de Navier-Stokes conforme a las hipótesis aplicadas y se expondrán todos aquellos detalles relativos a la preparación de la simulación DNS.

En primer lugar, el punto de partida es el programa que resuelve flujos turbulentos en tuberías mediante DNS *OpenPipeFlow*. Este código desarrollado en Fortran ofrece un punto de partida interesante, pues cuenta con una parte del código encargado de resolver el flujo que ofrece posibilidades de escalado. Se aprovechará para modificarlo y depurarlo con la finalidad de optimizar los recursos computacionales necesarios, hacer las adaptaciones necesarias para poder emplearlo para trabajos de *High Performance Computing* (HPC) con la finalidad de hacer simulaciones a altos números de Reynolds, así como también implementar una nueva parte en el código encargada de calcular las estadísticas del flujos con el fin último de poder estudiarlo y monitorizar las simulaciones.

3.2. Modelo matemático

EL modelo matemático de *OpenPipeFlow* que se empleará resuelve las ecuaciones de Navier-Stokes para el caso de fluido incompresible, ausencia de fuerzas másicas y flujo constante a través de una tubería de sección circular 3.1.

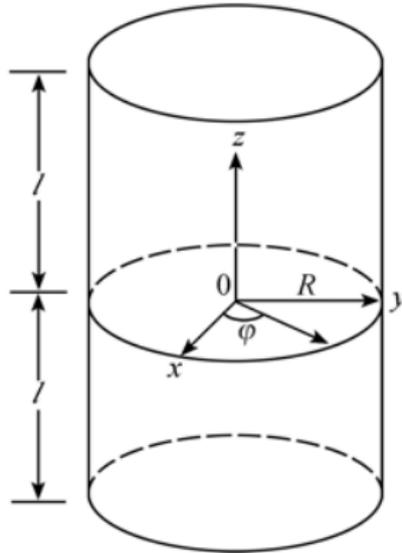


Figura 3.1: Esquema del dominio fluido con direcciones principales.
Créditos: [Zhou et al., 2007]

En primer lugar, se exponen los parámetros empleados en la adimensionalización del problema, los factores de escala empleados son:

- R , o radio de la tubería, para adimensionalizar longitudes.
- U_{cl} , o velocidad del flujo laminar en el centro de la tubería, para adimensionalizar velocidades.
- R/U_{cl} se toma como tiempo característico, para adimensionalizar tiempo.

Las ecuaciones empleadas [Willis, 2017] son:

$$\begin{cases} \nabla \cdot \mathbf{v} = 0 \\ (\partial_t + \mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla P + \frac{4}{Re_m} (1 + \beta) \hat{\mathbf{z}} + \frac{1}{Re_m} \nabla^2 \mathbf{v} \end{cases} \quad (3.4)$$

Y

$$(1 + \beta) = \frac{\langle \frac{dp}{dz} \rangle}{\frac{dP}{dz}} \quad (3.5)$$

Donde el término $\langle \frac{dp}{dz} \rangle$ hace referencia al gradiente medio de presión observado y $\frac{dP}{dz}$ al gradiente de presión laminar.

Para el caso de estudio en el que en la dirección axial se tiene un flujo constante, se impone un perfil de velocidades

$$W(r) = 1 - r^2 \quad (3.6)$$

Por lo que el vector velocidad resultante será $\mathbf{v} = (U, V, W + W(r))$, hecho que deberá ser tenido en cuenta en el posterior análisis.

Con todo ello, \mathbf{v} se puede escribir como $\mathbf{v} = W(r)\hat{\mathbf{z}} + \mathbf{v}'$ y al tomar la forma rotacional de la Ecuación 3.4 se tiene una ecuación de evolución para \mathbf{v}' y p' .

$$(\partial_t - \frac{1}{Re_m} \nabla^2) \mathbf{v}' = \mathbf{v}' \wedge (\nabla \wedge \mathbf{v}') - \frac{dW(r)}{dr} v'_r \hat{\mathbf{z}} - W \partial_z \mathbf{v}' + \frac{4\beta}{Re_m} \hat{\mathbf{z}} - \nabla p' \quad (3.7)$$

En la Ecuación 3.7 se demuestra que en el lado izquierdo de la expresión, el término laplaciano acopla el flujo en las direcciones $\hat{\mathbf{r}}$ y $\hat{\theta}$ i.e. las perturbaciones de la velocidad u' y v' , pues, por ejemplo, para la primera componente del laplaciano se tiene que $(\nabla^2 \mathbf{v}')_r = \nabla^2 u' - \frac{2}{r^2} \partial_\theta v' - \frac{1}{r^2} u'$.

Para desacoplar las velocidades U y V , se puede aplicar una descomposición en Fourier y con el cambio de variable

$$U_\pm = U \pm iV \quad (3.8)$$

Se obtiene que

$$U = \frac{1}{2}(U_+ + U_-)V = -\frac{i}{2}(U_+ - U_-) \quad (3.9)$$

Con lo que se obtiene una ecuación para las tres direcciones del flujo con las velocidades desacopladas

$$\begin{cases} (\partial_t + \nabla_\pm^2)U_\pm = N_\pm - (\nabla p)_\pm \\ (\partial_t + \nabla^2)W = N_z - (\nabla p)_z \end{cases} \quad (3.10)$$

donde $\nabla_\pm^2 = \nabla^2 - \frac{1}{r^2} \pm \frac{2i}{r^2} \partial_\theta$ y N hace referencia a los términos no-lineales agrupados.

Finalmente, agrupando los términos lineales y no-lineales y tomando la divergencia de ambos miembros de las ecuaciones, se llega a una ecuación similar a la de Poisson para la presión (Poisson's Pressure Equation, o PPE por sus siglas) empleada ampliamente para la resolución de problemas de flujo incompresible. Además, hay un detalle apreciable, ya que se reemplaza la expresión $\nabla \cdot \mathbf{v} = 0$ por la ecuación de Poisson que se obtiene. Nótese que la condición de flujo incompresible se sigue asegurando.

$$\begin{cases} \partial_t \mathbf{v} = \mathbf{L}\mathbf{v} + \mathbf{N} - \nabla P \\ \nabla^2 P = \nabla \cdot \mathbf{N} \end{cases} \quad (3.11)$$

En la Ecuación 3.11 \mathbf{L} es un operador lineal y \mathbf{N} representa al vector con todos los términos lineales, representados así para mayor simplicidad. Además, se impone la condición de contorno de no deslizamiento de que en la pared de la tubería $\mathbf{v} = 0$ y además $\nabla \cdot \mathbf{v} = 0$ ha de satisfacerse en todo el dominio de cálculo. No hay ninguna condición de contorno explícita para la presión.

3.2.1. Resolución de la PPE.

Se considera una discretización espacial tal que se consigue un grupo de problemas unidimensionales en dirección radial [Willis, 2017]. Puesto que se emplean un método numérico pseudoespectral—en dirección radial se emplean esquemas de diferencias finitas de alto orden mientras que para las direcciones azimutal y axial, periódicas espacialmente, se emplea una descomposición en modos de Fourier— se obtiene un grupo de problemas triviales distribuidos a lo largo de la coordenada radial, más detalles en [Willis, 2017]. Con todo ello el sistema resultante es el siguiente:

$$\begin{cases} \mathbf{X}\mathbf{v}^{q+1} = \mathbf{Y}\mathbf{v}^q + \mathbf{N}^{q+1/2} - \nabla P \\ \nabla^2 P = \nabla \cdot (\mathbf{Y}\mathbf{v}^q + \mathbf{N}^{q+1/2}) \end{cases} \quad (3.12)$$

El superíndice q denota el instante temporal, t_q y \mathbf{X} e \mathbf{Y} son las matrices asociadas a una discretización temporal implícita [Willis, 2017] para los términos viscosos.

En primer lugar, se resuelve el siguiente sistema, con la finalidad de encontrar lo que denomina el autor una “bulk solution” o solución aproximada tentativa $\{\bar{\mathbf{v}}, \bar{P}\}$.

$$\begin{cases} \mathbf{X}\bar{\mathbf{v}}^{q+1} = \mathbf{Y}\bar{\mathbf{v}}^q + \mathbf{N}^{q+1/2} - \nabla \bar{P} \\ \nabla^2 \bar{P} = \nabla \cdot (\mathbf{Y}\bar{\mathbf{v}}^q + \mathbf{N}^{q+1/2}) \end{cases} \quad (3.13)$$

Con condiciones de contorno $\bar{\mathbf{v}} = 0$ y $\partial_r \bar{P} = 0$.

A la solución aproximada, se le ha de añadir una serie de términos que corrijan la solución para obtener la exacta en el siguiente instante temporal, \mathbf{v}^{q+1} .

$$\mathbf{v}^{q+1} = \bar{\mathbf{v}} + \sum_{j=1}^4 a_j \mathbf{v}_j' \quad (3.14)$$

Para el cálculo de \mathbf{v}_j' se precisan los siguientes sistemas de ecuaciones [Willis, 2017]:

$$\begin{cases} \mathbf{X}\mathbf{v}' = -\nabla P' \\ \nabla^2 P' = 0 \end{cases} \quad (3.15)$$

Con condiciones de contorno $\mathbf{v}' = 0$ y $\partial_r P' = 0$ en $r = R$.

Y por otro lado,

$$\mathbf{X}\mathbf{v}' = 0 \quad (3.16)$$

Con condiciones de contorno $\mathbf{U}_\pm' = 0$, $\mathbf{U}_z' = i$ en $r = R$.

De la Ecuación 3.15 se obtiene la solución \mathbf{v}_1' que se puede añadir al valor de $\bar{\mathbf{v}}$ sin afectar al término de la derecha en la Ecuación 3.13, pero sí que altera el valor de \mathbf{v} para aplicar la corrección debida a la aplicación de las condiciones de contorno q través de la Ecuación 3.15, de manera análoga ocurre con la Ecuación 3.16, que aporta el valor de \mathbf{u}_j para $j = 2, 3, 4$, resultado de aplicar las condiciones de contorno para \mathbf{U}_\pm' y \mathbf{U}_z' respectivamente.

Con todo ello, a través de la Ecuación 3.14 se puede obtener otro sistema de ecuaciones para a_j , debido a que esos valores incógnita, han de satisfacer también la condición de contorno $\mathbf{v}^{q+1} = 0$ y la condición de fluido incompresible, $\nabla \cdot \mathbf{v}^{q+1} = 0$.

Sea $\mathbf{g}(\bar{\mathbf{v}})$ un vector de cuatro componentes que recoja estas últimas condiciones de contorno y de incompresibilidad, de tal manera que cuando estas se satisfacen, $\mathbf{g}(\bar{\mathbf{v}})$ se anula, sustituyendo en la Ecuación 3.14, se llega a la siguiente expresión:

$$\mathbf{A}\mathbf{a} = -\mathbf{g}(\bar{\mathbf{v}}) \quad (3.17)$$

Siendo A una matriz 4×4 — $A = A(\mathbf{g}(\mathbf{v}'))$ — y \mathbf{a} el vector que recoge los cuatro a_j . Como conclusión, tras resolver este sistema, se calculan los distintos a_j que permiten calcular a su vez \mathbf{v}^{q+1} a través de la Ecuación 3.14.

3.3. Mejoras realizadas al *solver* DNS.

A medida que se consiguió un mayor grado de compresión del funcionamiento del *solver*, se vio necesaria la implementación y mejora de una parte del código que calculase también el campo de presiones, no incluido en la versión disponible del *software*.

Para ello, una vez resueltos los sistemas expuestos anteriormente, al disponerse de la “bulk solution” de \bar{P} en Fourier, se reconstruye primero el campo de presiones en el dominio de Fourier (\tilde{P}) de la siguiente manera:

$$\tilde{P} = \bar{P} + a_1 P' \quad (3.18)$$

Para, tras aplicar una transformada inversa de Fourier para obtener la solución en el espacio físico del problema, sustraerle la presión dinámica debida a la velocidad del flujo así:

$$P = \tilde{P} - \frac{1}{2} \mathbf{V} \cdot \mathbf{V} \quad (3.19)$$

Los detalles del código generado aparecen en el repositorio indicado en el link del abstract.

Capítulo 4

Esquema numérico

En este capítulo se presentan aquellas herramientas matemáticas que hacen posible la computación de una solución numérica de las ecuaciones expuestas en el capítulo anterior.

En primer lugar hay que tener en cuenta la geometría base del caso de estudio, que se trata de una tubería modelizada como un cilindro recto. Por motivos de comodidad a la hora de implementar un algoritmo numérico de resolución de las ecuaciones, el dominio de cálculo se expresa en términos de un sistema de coordenadas cilíndrico centrado en el eje longitudinal del dominio.

Debido a este sistema de coordenadas, es necesario tener en cuenta los factores de forma presentes en los operadores matemáticos —recogidos en la referencia [Willis, 2017]—.

4.1. Método numérico de resolución del modelo matemático.

Como ya se ha expuesto en la Subsección 3.2.1, para la resolución numérica del sistema de ecuaciones se hace uso de métodos pseudoespectrales.

Los métodos espectrales son técnicas computacionales utilizada para aproximar soluciones numéricas de ecuaciones diferenciales, integrales o problemas matriciales como el que se trata en este trabajo, mediante la aplicación de técnicas basadas en el análisis espectral. Para este caso, al existir dos direcciones del dominio con condiciones de contorno periódicas, se va a emplear la descomposición modal de Fourier como técnica de análisis espectral, ya que como se expuso en la Subsección 3.2.1 permiten pasar de un problema complejo con ecuaciones acopladas entre sí a un conjunto amplio de problemas de resolución trivial.

Para este caso concreto, la principal diferencia entre un método espectral y los métodos numéricos tradicionales, como los métodos de diferencias finitas o los métodos de elementos finitos, los métodos espectrales se centran en el comportamiento del espectro de la matriz de soluciones (que contienen cada una de las componentes de la velocidad descompuesta en serie de Fourier) o el operador asociado al problema en cuestión.

Finalmente, como se puede observar, de las tres dimensiones principales del problema, únicamente la axial y la azimutal son periódicas, por lo que para el caso de la dirección radial, no es posible la aplicación de un método espectral, por lo que se emplean diferencias finitas de alto orden, es por ello que el método es pseudoespectral: no se aplican técnicas de análisis espectral en todas las direcciones del dominio.

Las direcciones del dominio θ y z se descomponen respectivamente en $M + 1$ y $2K + 1$ modos. Los modos “ M ” son los correspondientes a la dirección azimutal y van de $[0, M]$, mientras que los K son los modos axiales, $[-K, K]$.

$$A(r, \theta, z) = \sum_{|k| < K} \sum_{|m| < M} A_{km} e^{i(\alpha k z + m_p m \theta)} \quad n = 1, \dots, N. \quad (4.1)$$

Con todo ello, se consigue un conjunto de problemas, distribuidos en la dirección radial, en el que dicho conjunto a su vez, recoge el problema desacoplado de las direcciones axial y azimutal Ecuación 4.1. Sobre la colocación de los modos en las matrices computadas, en detalle en [Willis, 2017].

A modo de introducción al siguiente capítulo, es un hecho reseñable que la separación discreta del dominio en la dirección r en subdominios. Esto permite que el problema pueda ser resuelto por métodos computacionales de cálculo en paralelo, lo que diferencia significativamente el presente trabajo.

El algoritmo de resolución implementado se basa en los desarrollos recogidos en [Ku et al., 1987] y [Pulicani and Ouazzani, 1991], no expuestos explícitamente en el presente trabajo por no ser objeto directo de estudio y ser considerados meras herramientas.

Capítulo 5

Desarrollo del software.

Para la simulación y análisis de flujos turbulentos en tuberías, se ha optado por la técnica *Direct Numerical Simulation* (DNS). Para llevar a cabo la simulación hay que hacer una evaluación de los recursos necesarios y establecer una metodología de trabajo adecuada a las exigencias técnicas y computacionales que este tipo de simulación requiere.

5.1. Introducción a la supercomputación en turbulencia

El estudio de la turbulencia está fuertemente ligado a la ciencia de la supercomputación debido a la complejidad intrínseca y al amplio rango de escalas que están involucrados en los flujos turbulentos, pues la turbulencia es un fenómeno complejo y su estudio requiere la capacidad de procesamiento y la potencia de cálculo proporcionadas por las supercomputadoras.

Algunas de las razones por las que el estudio de la turbulencia está íntimamente relacionado con la supercomputación son:

- **Simulación directa de la realidad** Pues al querer estudiar la turbulencia es necesario conocer el comportamiento del flujo cuando interactúa con la pared de la misma manera que lo haría en el mundo real, con lo que no se emplean modelos que recreen estas interacciones. Estos otros modelos a su vez, se desarrollan y mejoran a partir de simulaciones directas de la realidad y gracias a este hecho, se calculan parámetros fundamentales de dichos modelos, lo que permite *a posteriori* modelar flujos turbulentos en vez de simularlos directamente, conllevando una gran reducción en los recursos computacionales necesarios para ello.
- **Gran cantidad de datos y resolución espacial y temporal** Los flujos turbulentos presentan una amplia gama de escalas, desde las escalas del tamaño característico del problema hasta las más pequeñas, de longitud característica de Kolmogorov, lo que resulta en un gran volumen de datos que deben ser almacenados, empleados a la vez que el código se ejecuta y posteriormente, ser analizados. Para obtener una comprensión precisa de los detalles de la turbulencia, se requiere una alta resolución espacial y temporal, por ello se emplean equipos de supercomputación, pues estos tienen la capacidad de manejar estos enormes volúmenes de datos y realizar cálculos a alta resolución.

- **Simulaciones numéricas detalladas** Las simulaciones numéricas son una herramienta fundamental para estudiar la turbulencia. Estas simulaciones resuelven las ecuaciones de Navier-Stokes en un dominio computacional, lo que permite modelar el comportamiento turbulento con un alto grado de detalle, simulando todas las escalas. Sin embargo, la complejidad de las simulaciones turbulentas requiere una gran potencia de cálculo y de gestión de datos, que solo las supercomputadoras pueden proporcionar.
- **Investigación en múltiples escalas** Como se ha mencionado, la turbulencia es un fenómeno multiescala, es decir, que se desarrolla en una amplia variedad de tamaños de torbellinos. Para comprender completamente la naturaleza de la turbulencia, es necesario investigar cómo interactúan estas escalas a lo largo del tiempo. Las supercomputadoras pueden simular y analizar estas interacciones en detalle.

5.2. Metodología de trabajo y recursos disponibles

Este trabajo se enmarca dentro del ámbito de la Supercomputación o *High Performance Computing* (HPC), dado que las técnicas de cálculo numérico y las características del problema son altamente demandantes en cuanto a recursos computacionales se requiere.

5.2.1. Metodología de trabajo

Con todo ello, es necesaria una metodología de trabajo que se oriente hacia la optimización del uso de los recursos computacionales existentes. Es por ello que el desarrollo del *software* se llevará a cabo en el lenguaje de programación Fortran y en un entorno Linux, debido a las ventajas que esta forma de trabajo ofrece.

Linux

A lo largo del desarrollo de este trabajo, se ha usado como sistema operativo Ubuntu. Ubuntu es un sistema operativo basado en Linux y ha sido escogido, no solo por la flexibilidad que ofrece, sino porque desde un punto de vista formativo en el ámbito de la Supercomputación, los sistemas operativos basados en Linux son los empleados por antonomasia.

A pesar de ser un sistema operativo con interfaz de usuario gráfica, se optará por trabajar con el terminal de comandos, ya que de tal manera se puede explotar todo el potencial de este sistema operativo además de permitir al alumno ser capaz de tener pleno control sobre cualquier equipo de supercomputación basado en Linux a través de un terminal básico y en remoto, con una cantidad mínima de recursos y con una mayor velocidad que si se llevara a cabo dicha tarea a través de una interfaz gráfica en remoto.

Sistemas remotos En el ámbito de la Supercomputación, la forma de empleo de estos equipos se basa en hacer uso de un superordenador (o de cualquier equipo con capacidades de computación relativamente altas, Figura 5.1) a través de un terminal remoto, ya que generalmente se trata de equipos voluminosos o que requieren de instalaciones especiales para su correcto uso, como pudiera ser una sala fría. Es por ello que se sigue la misma

operativa para el presente trabajo: por un lado se dispone de un equipo destinado a la ejecución de los diversos programas elaborados y por otro, nuestro terminal de usuario encargado de mandar las instrucciones al primer equipo de manera remota, de tal forma que el único recurso necesario por parte del usuario es de un ordenador con conexión a internet sin necesidad de que dicho equipo disponga de grandes recursos, ya que únicamente se empleará para mandar comandos al equipo de computación.



Figura 5.1: Equipo de supercomputación SuperMUC, empleado en este trabajo para tareas de testeo y validación del *software*. Fuente: Leibniz Supercomputing Centre

Fortran

En este capítulo se recogerán todos aquellos aspectos importantes con los que se ha trabajado a lo largo del desarrollo del presente proyecto con el lenguaje de programación Fortran y en el estándar de manejo de datos de alto rendimiento HDF5.

Fortran es un lenguaje de programación de propósito general desarrollado originalmente en 1957 por un equipo de científicos e ingenieros liderados por John Backus en IBM [Backus et al., 1957]. Es uno de los lenguajes de programación más antiguos que aún se utilizan en la actualidad y destaca por su popularidad en el campo de la computación científica.

Fortran se ha utilizado para desarrollar muchos programas científicos y de ingeniería importantes, incluyendo el código del proyecto Apollo, el código del modelo climático del IPCC y el código del acelerador de partículas CERN. Sigue siendo un lenguaje popular para el desarrollo de software científico y se utiliza en una amplia gama de aplicaciones, incluyendo la investigación en física, química, biología, ingeniería, meteorología y finanzas.

Aquí hay algunas de las características clave de Fortran [Haveraaen et al., 2013]:

- Es un lenguaje de programación compilado, es decir, el código Fortran se traduce a código de máquina antes de ser ejecutado. Esto hace que Fortran sea muy rápido y eficiente y dada esta eficiencia es que se escoge como opción en el ámbito de la supercomputación.
- Es un lenguaje de programación de propósito general, lo que significa que se puede utilizar para desarrollar una amplia gama de aplicaciones.
- Es un lenguaje de programación muy fácil de leer y escribir, lo que lo convierte en una buena opción en el ámbito científico y de la ingeniería.

Estructura de un programa escrito en Fortran Desde un comienzo ya se observan ciertas diferencias a la hora de emplear Fortran con respecto a programas que hasta esos primeros momentos eran los más habituales como por ejemplo Matlab, pues al ser un lenguaje de tipado estático y fuerte, facilita la detección de errores, aunque la implementación del código debe seguir una estructura determinada.

Un programa Fortran está compuesto por declaraciones, que son instrucciones que sigue el compilador. Las declaraciones se pueden agrupar en bloques, que son secciones de código que están relacionadas entre sí. Los bloques se pueden anidar, lo que significa que un bloque puede estar contenido dentro de otro bloque.

La estructura de un programa Fortran es la siguiente:

- **Bloque de cabecera:** Parte del programa donde se indican la inclusión de módulos de subrutinas empleados en el programa y funciones y la sentencia “program” con la que se delimita el programa en sí.
- **Bloque de declaraciones:** El bloque de declaraciones contiene las declaraciones que definen las variables y otras estructuras de datos utilizadas por el programa.
- **Bloque de código:** El bloque de código contiene las instrucciones que realizan el procesamiento real del programa.
- **Delimitador final:** Contiene el delimitador “end program” a modo de clausura del código.

Módulos de subrutinas Un módulo en Fortran es una unidad de código que puede ser importada y usada por otros módulos o programas. Los módulos se utilizan para organizar el código y para hacer que el código sea más reutilizable. Los módulos se declaran usando la palabra clave “module”. El cuerpo del módulo contiene las declaraciones de las variables, las funciones y los subprogramas que están contenidos en el módulo.

GitHub, GIT

GitHub y Git son dos herramientas esenciales que se han utilizado a lo largo del proceso de desarrollo del software de este trabajo. Estas herramientas desempeñan un papel fundamental en la gestión del desarrollo del proyecto y en el seguimiento del progreso del trabajo.

- **GIT**
 - **Control de versiones** Cada iteración del código y los recursos utilizados en el trabajo, desde el código fuente hasta los datos de entrada y salida, se han registrado y almacenado en repositorios Git. Esto ha permitido rastrear cómo ha evolucionado el trabajo a lo largo del tiempo y facilitar la reversión a versiones anteriores si es necesario.
 - **Experimentación y ramificación** Git permite crear ramas separadas para probar nuevas ideas o enfoques sin afectar directamente el trabajo principal. Esto es especialmente útil en la investigación de flujos turbulentos, donde diferentes métodos y algoritmos pueden requerir pruebas y ajustes continuos.

- **Gestión de conflictos** Cuando varias personas realizan cambios en el mismo archivo simultáneamente, pueden surgir conflictos. Git proporciona herramientas para manejar estos conflictos y fusionar los cambios de manera ordenada.

- **GitHub**

- **Almacenamiento remoto** GitHub proporciona un lugar centralizado para alojar los repositorios Git, lo que garantiza que los archivos estén seguros y accesibles desde cualquier ubicación.
- **Revisión del código** GitHub facilita la revisión de código mediante la función de solicitudes de extracción (pull requests). Los miembros del equipo pueden revisar y comentar los cambios propuestos antes de que se fusionen en la rama principal.

5.2.2. Recursos disponibles

Para el presente trabajo se hará uso de dos tipos de recursos computacionales. Esta división se hará en base a la dimensión del equipo.

- Equipo de pequeña capacidad de HPC: de 80 unidades de proceso y 120GB de RAM. Será el empleado en tareas de desarrollo y validación. Para probar los códigos se prefiere simular a pequeña escala, pues las grandes simulaciones llevan un enorme coste asociado por hora de cálculo, aunque en el caso de este trabajo, las horas de cálculo han sido cedidas por distintas entidades, no obstante es un recurso escaso y se prefiere hacer pruebas preliminares en equipos de capacidad más modesta, reservando las simulaciones completas para el siguiente tipo de máquina.
- Supercomputadoras: para el caso del presente trabajo, se ha trabajado con 3 supercomputadoras, en orden, de menor a mayor capacidad: MareNostrum4, SuperMUC y ShaheenII. Estas máquinas serán las encargadas de, una vez validado el código en el equipo de pequeña capacidad, llevar a cabo las simulaciones más demandantes de recursos computacionales.

5.3. Cálculo en paralelo con Fortran.

En este capítulo se tratarán los fundamentos del cálculo en paralelo, que nace de las limitaciones computacionales de procesos que, de ser ejecutados en serie en un único procesador, el tiempo necesario para llevar esta tarea a cabo sería inasumible, por no hablar de otras limitaciones como las prestaciones de la unidad de procesamiento y de la memoria de la que dispone.

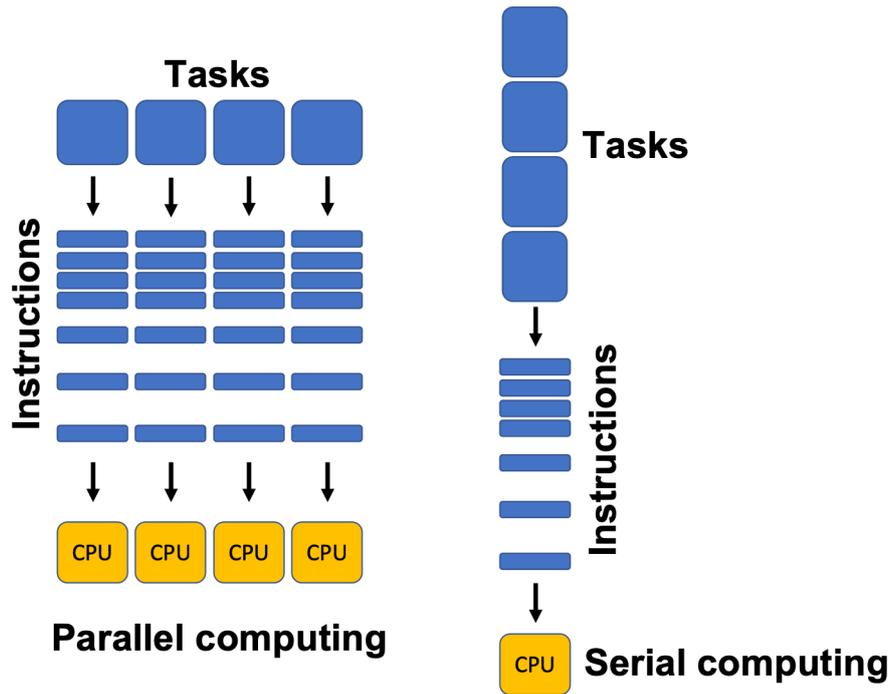


Figura 5.2: Diferencia entre serie y paralelo. Fuente: Python Numerical Methods

El procesamiento en paralelo se muestra como la única opción válida a la hora de ejecutar ciertos programas altamente demandantes de recursos computacionales, aunque no sin cumplir ciertos requisitos.

Un requisito fundamental es que dicho proceso ha de poder dividirse conforme a, al menos uno de estos criterios:

- **División del dominio:** es el caso que se puede encontrar cuando se necesita procesar una serie amplia de elementos de manera similar o igual, por lo que es posible distribuir grupos de elementos entre las distintas unidades de procesamiento. Por ejemplo como ocurre con las simulaciones CFD, donde el dominio se subdivide en otros donde se aplican las ecuaciones de la dinámica de fluidos.
- **División por bloques funcionales:** ocurre cuando unidades de procesamiento distintas se encargan de tareas distintas, sin conllevar necesariamente una división previa del dominio. Un ejemplo interesante puede ser el procesamiento de imágenes donde se aplican filtros distintos a la misma imagen para identificar características distintas.

También se ha observado que el método de paralelización óptimo depende de otros elementos, como la arquitectura física del sistema, en particular la forma de gestión de la memoria, ya sea memoria compartida o memoria distribuida, como se tratará en los siguientes capítulos.

Por ejemplo, un sistema con memoria compartida será más adecuado para aplicaciones que requieren el acceso concurrente a la memoria, mientras que un sistema con memoria distribuida será más adecuado para aplicaciones que requieren el procesamiento de grandes cantidades de datos.

Otro fenómeno es el de la granularidad, es decir, la cantidad de subproblemas más pequeños en los que se divide el principal. Esta división se puede realizar de diferentes

maneras, y la elección de la granularidad óptima afecta al rendimiento del programa paralelo. Por un lado, si se utilizan demasiados subproblemas, el tiempo de comunicación entre los procesadores puede aumentar significativamente, lo que puede degradar el rendimiento. Por otro lado, si se utilizan muy pocos subproblemas, el paralelismo no se puede aprovechar y el rendimiento también se degrada, por no hablar de que podría haber una repartición desigual de tareas y existiese un *idle time* o tiempo destinado a esperar al resto de procesadores para continuar con la ejecución de un programa en el se haya aplicado descomposición funcional, por poner un ejemplo.

En definitiva, si por ejemplo, se trata de un problema que se puede dividir en 1000 subproblemas, no es adecuado usar 10000 procesadores para resolverlo. Esto se debe a que la mayor parte del tiempo se emplearía en las comunicaciones entre los procesadores, y no en el cálculo real. Un número óptimo de procesadores para este problema sería de alrededor de 100, por ofrecer una aproximación.

En general, la elección del método de paralelización óptimo depende de una serie de factores, como la arquitectura física del sistema [Culler et al., 1999], la granularidad del problema [Kwiatkowski, 2002], la capacidad de cálculo del equipo, la cantidad de memoria disponible y la infraestructura de red. Estos factores se tratarán con más detalle en los siguientes capítulos, aunque se han visto otros factores que afectan al rendimiento como la escritura o lectura de datos, puestos a prueba a través de programas de *benchmarking* como el mostrado en el Código ?? [??], donde se comparan los tiempos de cálculo según se lean o escriban los elementos de una matriz transpuesta correlativos en memoria o no, o experimentando con el papel que juega el uso de otros métodos para en este caso, calcular matrices transpuestas.

Message Passing Interface (MPI)

MPI es una interfaz estándar para comunicación entre procesos ejecutados en paralelo de memoria distribuida. Se trata de una biblioteca de funciones que permite a los programadores enviar y recibir mensajes entre procesos que se ejecutan en diferentes nodos de una red de unidades de procesamiento y se muestra como una herramienta poderosa que se puede utilizar para construir aplicaciones paralelas eficientes y escalables.

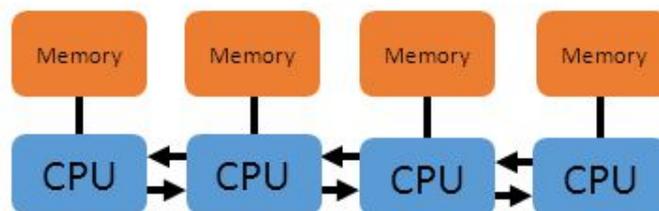


Figura 5.3: Ejemplo de arquitectura MPI. Fuente: [?]

Se trata por tanto, de una interfaz orientada a mensajes, lo que significa que los procesos se comunican enviando y recibiendo mensajes entre sí, mensajes pueden ser de cualquier tamaño y pueden contener cualquier tipo de datos. Además, MPI no solo proporciona una serie de funciones para enviar y recibir mensajes sino también para gestionar el flujo de mensajes entre los procesos, permitiendo al usuario establecer diferentes formas

de comunicación entre procesos, ya sea síncrona o asíncrona, dependiendo de la naturaleza del proceso a ejecutar o de otras cuestiones como podría ser el rendimiento.

HDF5 (*Hierarchical Data Format*)

Una cuestión importante en supercomputación es la de la lectura, escritura y el manejo de datos, ya que como se puede suponer, a la hora de resolver problemas que precisan de equipos con varios cientos -si no miles- de unidades de procesamiento, se generan cantidades de datos cuya gestión no es trivial. Además a la hora de leer y escribir datos se consume un tiempo que deseablemente, será el menor posible, ya que disponer de estos equipos generalmente es siempre caro el coste por hora de uso.

Pero sobre todas las posibilidades que ofrece HDF5, quizá la que más nos interesa es que es posible leer y escribir en este formato de manera paralela, dejando atrás la lectura/escritura en serie que en el caso de trabajar con grandes volúmenes de datos, constituiría un cuello de botella que pondría en compromiso la eficiencia del código.

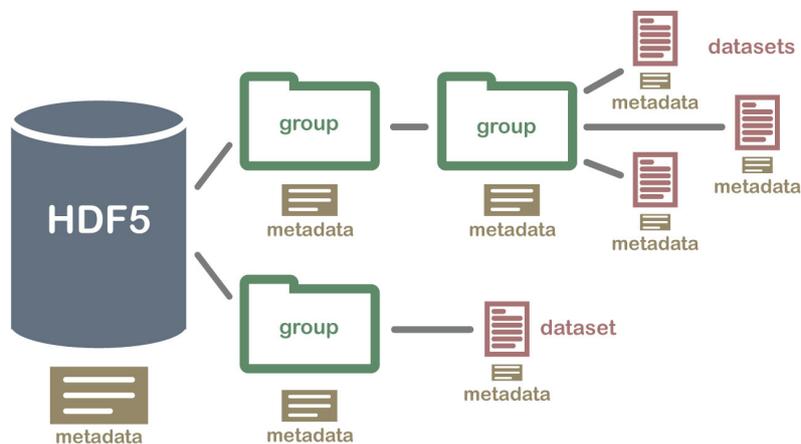


Figura 5.4: Estructura jerárquica de los datos en HDF5. Fuente: <https://soloelectronicos.com>

Es entonces cuando se presentó el sistema HDF5, ya que su uso es altamente recomendable en Supercomputación debido a que es un formato de archivo estándar abierto y de alto rendimiento que se utiliza para almacenar grandes cantidades de datos estructurados. HDF5 es un formato de archivo binario que ofrece una serie de ventajas sobre otros formatos de archivo empleados tradicionalmente.

Por otro lado, es un formato de archivo popular que se utiliza en una variedad de campos, incluyendo la ciencia, la ingeniería, la medicina y la finanzas.

Algunas de las ventajas de usar HDF5 son:

- Escalabilidad: HDF5 es un formato de archivo escalable, ya que puede almacenar grandes cantidades de datos sin ocupar demasiado espacio en el disco.
- Velocidad: al ser un formato de archivo estructurado jerárquicamente, lo que se traduce en que se puede acceder a los datos de forma rápida y eficiente.
- Versatilidad: HDF5 es un formato versátil, ya que se puede utilizar para almacenar una gran variedad de datos.

- Portabilidad: por lo tanto, se puede utilizar en una variedad de plataformas.
- Estándar abierto: es gratuito de usar y está disponible para cualquiera.

Como se muestra en la Figura 7.15 HDF5 nace como respuesta a la diversidad de formatos existentes, aglutinando sus características más atractivas.

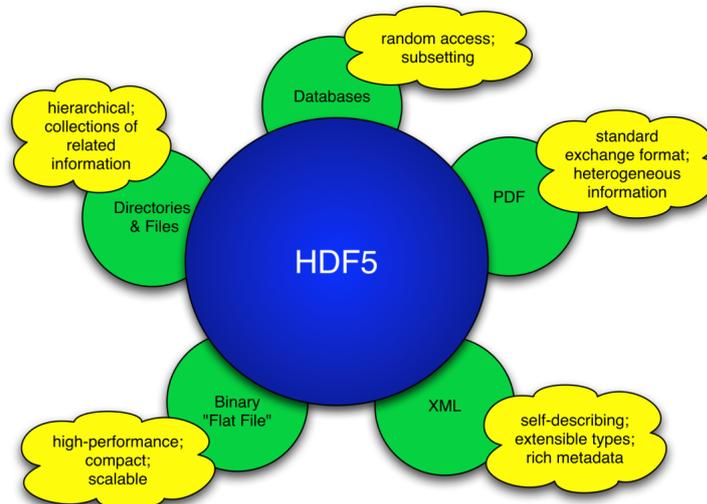


Figura 5.5: HDF5 aúna características destacables de otros formatos. Fuente: Apuntes de clase.

5.4. Punto de partida, problemas

El punto de partida es el software para la resolución de problemas CFD en tuberías *OpenPipeFlow* [Willis, 2017].

Este software cuenta con diversas herramientas para la resolución de flujos turbulentos con diversas técnicas en función de la alta o baja resolución que se desee obtener al calcular flujos turbulentos en tuberías. Para el caso del presente trabajo se emplea un *solver* que emplea la técnica DNS.

Originalmente este software fue concebido únicamente para simulaciones a números de Reynolds bajos o moderados, con geometrías pequeñas, por lo que presentaba en principio problemas graves para ofrecer una solución computacional escalable.

La escalabilidad de una herramienta de software alude a la cualidad del mismo para ser usado como una herramienta genérica, capaz de resolver eficazmente problemas numéricos como es una simulación DNS en una variedad amplia de condiciones. Para ello, la herramienta ha de ser eficaz y resultar óptima en cuanto a uso de recursos computacionales tanto en simulaciones a bajo número de Reynolds como a altos, siendo la única diferencia los recursos computacionales necesarios.

Tras una primera etapa de test de la herramienta *OpenPipeFlow* se comprobó que se ofrecía como una solución *ad-hoc* para resolver flujos turbulentos a números de Reynolds relativamente bajos, con lo que su escalabilidad era cuestionable. Esta poca escalabilidad no se podía garantizar debido a razones de consumo de memoria principalmente, pues

se estimó que para una simulación con número de Reynolds del orden de cualquiera del estado del arte actual, se requeriría disponer de una memoria RAM de 1.6 TB.

Esto hecho representa un coste computacional prohibitivo y una mala utilización de los recursos disponibles.

Para ello, ha sido necesaria una labor de estudio y comprensión de las diversas partes del código para entender su funcionamiento y detectar con la mayor concreción posible, los puntos débiles a mejorar. Estos puntos débiles son fundamentalmente dos:

- La gestión de la lectura y escritura de los datos.
- La gestión de la memoria necesaria por la máquina en el tiempo de ejecución.

Estos cambios se abordarán en detalle en estas últimas secciones de este capítulo.

5.5. Cambios del formato de gestión de datos

Desde un primer momento se propuso como objetivo cambiar la manera en la que los datos se leen y escriben, pues con el formato de almacenamiento de datos empleado originalmente, *Network Common Data Form*, *NetCDF* se comprometía la escalabilidad del *software* desarrollado.

5.5.1. NetCDF y limitación en HPC

es un formato de archivo diseñado para almacenar y compartir datos científicos y de ingeniería, particularmente aquellos que son multidimensionales y que pueden ser de gran tamaño. NetCDF es ampliamente utilizado en el ámbito de la supercomputación y la modelización numérica debido a su eficiencia en el almacenamiento y manipulación de datos complejos.

Su uso, extendido en el ámbito de la HPC se debe a las siguientes características y limitaciones:

- **Multidimensionalidad:** esencial para datos científicos que involucran variables que varían en más de una dimensión.
- **Multiplataforma:** pues los archivos NetCDF pueden ser leídos y escritos empleando equipos con diferentes sistemas operativos o lenguajes de programación, siendo una herramienta multiplataforma, no obstante, cabe remarcar que no es un estándar de almacenamiento de datos.
- **Acceso eficiente a los datos:** o a los subconjuntos de datos, lo que es esencial cuando se trata con conjuntos de datos grandes caso particularmente importante en supercomputadoras, donde el acceso rápido a los datos es crucial para el rendimiento. Sin embargo, no permite la lectura o escritura en paralelo, lo que compromete una vez más la escalabilidad del software y constituye la mayor motivación para cambiar de formato de gestión de datos.
- **Librería precompilada y de fácil instalación:** siendo esta su mayor ventaja frente a la alternativa escogida.

Con todo ello, y contando con la experiencia del tutor, se migró todas aquellas partes del código en las que se leen o escriben datos, recogidas en el módulo *io.f90*.

Concretamente, se cambió y se implementó la siguiente relación de soluciones, incluidas en el nuevo módulo *io.F90*, adjunto en el anexo.

5.5.2. Conversión a HDF5, implementaciones realizadas y mejoras observadas.

- Archivo de configuración *hre.dat*, donde ya solo se incluyen las rutas de los archivos de entrada y salida, en vez del otro método rudimentario.
- En la inicialización de la simulación, se permite la lectura de datos en paralelo, lo que conlleva una disminución del tiempo de lectura para la simulación de referencia ($Re_\tau = 180, N_z \times N_t \times N_r = 512 \times 128 \times 192$) de los 10s segundos a los 0.13s.
- Paralelización del proceso de escritura, reduciéndolo en la misma medida que el de lectura.

Con todo ello, ya se observa un incremento en la velocidad de simulación considerable, pues para el caso de la simulación de referencia se consiguió llegar al punto de flujo estadísticamente estacionario (10.000 iteraciones) con un tiempo de computación de unas 15 horas, siendo en el caso del empleo de NetCDF este tiempo superior al día.

5.6. Memoria

Otro gran punto clave de las mejoras desarrolladas incluyen la reducción del consumo de memoria durante la ejecución del código. En primer lugar, se observó el hecho de que una simulación relativamente pequeña en términos de la máquina disponible, ($Re_\tau = 180, N_z \times N_t \times N_r = 512 \times 128 \times 192$) requería más memoria de la disponible, por lo que comenzaron las primeras investigaciones con el fin de solucionar esta limitación a la escalabilidad del *software*.

5.6.1. Ahorro de memoria

Por otro lado, se llevaron a cabo simulaciones de prueba donde se observó el consumo de memoria real observando una reducción notable en éste, pues a través de los sucesivos cambios reflejados en las versiones disponibles en el repositorio adjunto, se consiguió reducir a 40 GB un consumo de memoria que originalmente suponía 53 GB, es decir, una reducción relativa del 24,5 %.

Es un hecho a reseñar que, para el caso de este tipo de simulaciones, a la hora de resolver el sistema de ecuaciones en el que se fundamenta la simulación, hay dos vertientes a la hora de la elaboración del código:

- Código intensivo en recursos de memoria: en este caso, se almacenan copias del sistema en cada unidad de procesamiento, por lo que el consumo de memoria puede imposibilitar el uso del código por falta de memoria en el procesador, como ocurre en el caso del código base de este trabajo, lo que impidió su escalabilidad.

- Código intensivo en unidades de procesamiento: en este caso no se requieren copias del sistema a resolver en cada procesador, pero en cambio, se necesita de más procesadores trabajando en paralelo, pues la operación a realizar con la matriz del sistema (generalmente suele ser invertirla) es más demandante de recursos al tratarse de una matriz de mayor tamaño.

Se concluyó que por la manera de trabajo del código, el paradigma empleado es el del consumo de memoria pues como se puede observar en el código, las estructuras *lu-mesh* [consultar en el repositorio de GitHub adjunto], contienen una copia del sistema de ecuaciones al completo.

Fue a partir de ese momento que comenzó un proceso de estudio y comprensión del código para proponer mejoras y optimizar el código, pues se requería conseguir los mismos resultados con una cantidad de memoria sustancialmente bajo, objetivo poco trivial de detectar en este caso.

A continuación se recoge una relación de todos aquellos elementos que se detectaron y que contribuyeron al desempeño pobre del código original:

- **Memoria redundante:** Se determinó que en algunas partes del código estaban declaradas variables o estructuras de memoria destinadas a la misma función pero con nombre distinto, lo que llevaba a la creación de memoria redundante que únicamente empeora el consumo de memoria del código.
- **Memoria que no se usa, para uso en serie:** el código original, por razones de versatilidad, se podía usar con un procesador, ejecutando la simulación en serie, o bien, en paralelo. Para el caso de estudio debido a la magnitud de las simulaciones estudiadas, éstas, van a requerir siempre ser ejecutadas en paralelo, por lo que aquellas estructuras de memoria y variables empleadas en el cálculo en serie se retiraron del código.
- **Memoria no aprovechada:** Se observó que existía una gran cantidad de memoria definida y que no se leía/escribía a la vez, lo que condujo a detectar aquellos procesos en los que se declaraba memoria para ejecutar una tarea y tras la ejecución no se volvía a usar. La medida correctiva que se llevó a cabo fue reutilizar esa memoria en otros procesos una vez que ésta dejase de ser utilizada para su primer cometido.
- **Memoria auxiliar reaprovechable, módulo *wksp.f90*:** Tras todas estas tareas de diagnóstico y corrección sobre el código, se desarrolló un nuevo módulo, *wksp.f90* cuya función es la de contener todas aquellas estructuras de memoria reutilizable para tenerlas en un sólo módulo de subrutinas y poder controlar más fácilmente el uso de memoria de todo el código. Además estas estructuras al estar definidas una sola vez en el módulo, se garantiza la no-redundancia de memoria en el código.

Con todo ello, se preparó de nuevo una simulación de referencia que, tras cada cambio efectuado, mostró la siguiente reducción de memoria, con una reducción total del **20 %**..

Aunque al realizar de nuevo pruebas de escalabilidad, se observó un nuevo problema: con el modelo de memoria estática empleado, había un error de desbordamiento, es decir, el código seguía requiriendo más memoria de la disponible en cada procesador. Para entender este problema en profundidad, se desarrollará la idea a continuación.

5.6.2. Memoria estática vs. memoria dinámica.

Una tarea de las más importantes a la hora del desarrollo del código objeto de estudio en este trabajo es la de desarrollar los recursos de *software* necesarios para que el programa haga el máximo uso de memoria dinámica posible.

Esto se debe a que en el caso de la memoria dinámica, puede haber reutilización de memoria, pues ésta, tras ser declarada y reservada por el sistema, puede destinarse a almacenar matrices de dimensión variable a demanda, además, se puede almacenar tantos datos como permita la memoria RAM, mientras que para el caso de la memoria estática, no solo no se puede cambiar de dimensión, sino que además, se almacena directamente en cada procesador (recuérdese que esta memoria suele ser del orden de *Megabytes*), por lo que simulaciones relativamente pequeñas ocuparán más memoria de la disponible. Es por ello que se cambió el tipo de memoria de aquellas estructuras más grandes (recogidas en *wksp.f90*) de estática a dinámica.

5.6.3. De memoria estática a dinámica

Para ello, se emplea la función *allocate*, que permite definir *allocatables*, que es el nombre que reciben aquellas estructuras de memoria, usualmente vectores o matrices cuando se definen como variables dinámicas.

Además, con este cambio, las *flags* de compilación para optimizar el código resultaron de mayor utilidad, pues se redujo más aún el consumo de memoria.

Por último, al ser la memoria dinámica, dejó de haber desbordamientos de memoria del *stack* en los procesadores, pues ahora se hace uso del *heap*, como ocurría en el caso de emplear memoria estática.

5.7. Makefile

Una de las tareas llevadas a cabo con *bash* ha sido la de elaborar *Makefiles* para automatizar la compilación de los programas escritos en *Fortran* de cierta complejidad, incluyendo los módulos de subrutinas empleados en el programa principal, *main.f90*.

En general, los *makefiles* son un tipo de *script* que se utiliza para automatizar la construcción de proyectos de *software*. Se trata de un archivo de texto que contiene una lista de objetivos, dependencias y reglas.

- Los objetivos son los resultados que se desean crear, en el caso de la simulación, compilar un programa escrito en *Fortran* y que emplea el formato de datos *HDF5*, hecho relevante, pues es necesario de un compilador especial que trabaje con esta librería y permita la lectoescritura de los datos en paralelo.
- Las dependencias son los archivos que se necesitan para construir los objetivos, es este caso un código compilado *main.o* con el que se ejecute el programa principal y se tenga en cuenta las múltiples dependencias.
- Las reglas son los comandos que se utilizan para construir los objetivos, en nuestro caso, el nombre compilador que se desea usar y las *flags* u opciones de compilación que ofrece el mismo.

Los cambios e implementaciones en el *Makefile* destacadas son:

- Organización de las rutas de los archivos: organizando los ficheros en directorios según su funcionalidad. Al definir las distintas rutas en el *Makefile*, incluídas las de las rutas de las librerías como *lapack* (librería con las funciones más utilizadas en operaciones algebraicas de matrices, optimizadas numéricamente) o *fftw3* (para las transformadas rápidas de Fourier), es más fácil la detección de errores en el código debidos a cuestiones de organización.
- Compilador *hpf5c*: necesario para usar el formato HDF5 y usar MPI simultáneamente en Fortran.
- Inclusión de los nuevos módulos teniendo en cuenta sus dependencias: tareas llevada a cabo, por ejemplo, durante el desarrollo del módulo *wksp.f90*, para poder ser empleado por el resto del código.
- Seguimiento y corrección de errores, mención a la memoria: pues con las distintas *flags* de optimización y de *debugging* es posible determinar qué variables no se usan o en qué línea de programa hay un error (recuérdese que Fortran es un lenguaje compilado y salvo el uso de *flags* específicas, no alerta de dónde se encuentra el error).

5.8. Metodología de validación, parámetros en los que me fijo

Por último, se llevaron a cabo labores de verificación previas al lanzamiento de las simulaciones que constituyen los resultados del presente trabajo. Esta etapa de verificación es indispensable, pues debido a la gran complejidad del código, al tratar de optimizarlo, es frecuente realizar cambios sobre partes del mismo que son clave y por tanto no contar con un código funcional, aunque lo sea en apariencia.

5.8.1. Test sucesivos entre cambios.

Para asegurar un seguimiento adecuado de los sucesivos cambios que se han realizado, se ha seguido la siguiente metodología; tras cada cambio, se ha hecho una simulación y se han vigilado los parámetros clave de la simulación durante los primeros 500 pasos temporales, con la finalidad de observar anomalías.

Al tratarse de un código tan complejo, con multitud de módulos y subrutinas interrelacionados, el proceso de modificación no es trivial, pues cabe la posibilidad de que un cambio pueda tener consecuencias no esperadas, de aquí que se opte por testear el código tras cada cambio realizado.

5.8.2. Prueba de escalado intermedia.

Durante el proceso de liberación de memoria redundante, se determinó que hay una subrutina en la que no se puede sustituir el empleo de la memoria por defecto por memoria definida en el módulo auxiliar *wksp.f90*. Se hace la distinción entre prueba de escalado

intermedia y a gran escala porque las pruebas de liberación de memoria antes mencionadas y la metodología para las pruebas de escalado intermedias se han llevado a cabo en un equipo de capacidad computacional moderada/baja, de 80 núcleos y 120GB de RAM como se ha mencionado.

Además, en el módulo *timestep.f90* se ha encontrado dificultad para sustituir algunas matrices por las auxiliares, pues se observó que esto conlleva dos tipos específicos de problema:

- El CFL disminuye a la mitad: este problema se observó que aparece cuando se modifica unas variables de pequeño tamaño que se emplean en el cálculo del CFL. Trae como consecuencia que el CFL es más pequeño que el correspondiente para la simulación de referencia, con lo que se simula a un ritmo notablemente más lento y en condiciones lejos de las óptimas. No obstante se ha dejado como está, pues el impacto en el uso de memoria es despreciable, frente al tiempo que se ha empleado en encontrar una solución que surtiese efecto, con lo que se opta por dejarlo como trabajo propuesto.
- La simulación se para: también se observó un comportamiento anómalo al tratar de eliminar memoria de la subrutina *timestep.f90* con lo que se optimizó con la salvedad de aquellas variables que hacen que el diferencial de tiempo se reduzca progresivamente hasta hacerse nulo y parar la simulación con un código de error que indica que se ha alcanzado un flujo estacionario, condición que es imposible de obtener en este tipo de simulaciones y por tanto, se considera errónea la simulación. También se propone como trabajo de optimización en el módulo *timestep.f90*.

5.8.3. Prueba de escalado con simulación a gran escala

En este apartado se recoge aquella información relativa a las pruebas de escalado del *software* desarrollado que se han llevado a cabo en equipo de supercomputación, como Marenostrum o SuperMUC.

En este caso, y dado que aquellas simulaciones más grandes consideradas en este trabajo no se pueden llevar a cabo en el equipo personal del que se ha dispuesto debido a que no se dispone de memoria suficiente, se han tratado de simular en estos equipos de altas prestaciones anteriormente mencionados.

Tras haber liberado toda la memoria estática posible en el código, se trataron de recrear las simulaciones más grandes del trabajo de [Yao et al., 2023] con lo que se observó que el número de procesadores necesarios era desorbitado para una simulación de esas características. Se llegó a la conclusión de que esas simulaciones se podrían recrear, pero con un coste computacional lejos del óptimo. Este sobrecoste computacional se debe a que la simulación no cabe en la memoria *stack* de los procesadores, hecho que se solucionó cambiando la memoria estática a dinámica. De la misma manera que en la sección anterior, tras cada cambio se ejecutaba una simulación a modo de verificación del cambio implementado.

Tras el proceso de migración de memoria estática a dinámica, se volvió a recrear simulaciones de alto coste computacional sin el problema anteriormente mencionado.

5.8.4. Conclusión del proceso de verificación

A modo de conclusión del proceso de verificación se expone una publicación [Piedrabuena et al., 2023], resultado del empleo de el código desarrollado en este presente trabajo y que muestra la herramienta desarrollada como un medio útil a la hora de llevar a cabo experimentos empleando simulaciones DNS en tuberías para estudiar el flujo turbulento.

Capítulo 6

Desarrollo del módulo de cálculo de las estadísticas

En esta sección se recoge todo el procedimiento llevado a cabo para el cálculo de lo que constituirá los resultados finales de la simulación.

En el caso de la turbulencia, se emplea el análisis estadístico debido a la naturaleza aleatoria y caótica de este tipo de flujo, que sin embargo, exhibe una serie de características, representadas a través de la media y los distintos momentos estadísticos de la velocidad en el dominio de cálculo propias del flujo estudiado y que es reproducible mediante experimentación.

6.1. Objetivo

El objetivo principal es el de, una vez ha sido simulado el flujo, realizar los cálculos correspondientes a su tratamiento estadístico con la finalidad de obtener unos resultados que se compararán con otras simulaciones de referencia. Así mismo, también cobra especial importancia que a lo largo del proceso de desarrollo del módulo en el que se calculan las estadísticas se ha tenido en cuenta el impacto de estas operaciones adicionales, que implican un coste computacional extra en la simulación con la finalidad de obtener una solución sin elementos redundantes y lo más optimizada posible.

6.2. De la solución numérica a las velocidades medias y RMS

Tras la aplicación del algoritmo de cálculo de las velocidades para cada punto del dominio en cada instante temporal, se propone una metodología de análisis de los resultados con la finalidad de obtener una serie de conclusiones acerca del flujo estudiado.

En primer lugar, a modo de verificación de los resultados obtenidos mediante el *solver*, se analizan las velocidades y presiones del flujo, a las que se les aplica un tratamiento estadístico.

6.2.1. Resultados de la simulación. Velocidad en los dominios físico y espectral

Como resultado de la simulación se obtienen las tres componentes de la velocidad $\mathbf{v} = (U, V, W)$, que, como ya se ha expuesto anteriormente, son variables aleatorias caracterizadas por su promedio y sus momentos de distinto orden. Además, como método de estudio —además de para la resolución del sistema de ecuaciones principal— se emplean herramientas de análisis espectral por comodidad. Tras una descomposición en modos de Fourier de las velocidades en cada instante temporal, se almacenan los coeficientes resultantes y en primer lugar se promedian para todos los instantes temporales (para calcular el valor medio) y, por otro lado, se promedia su cuadrado para posteriormente encontrar el valor RMS de la componente en cuestión de la velocidad, este método se detalla más adelante en el capítulo.

Aplicando este proceso a cada una de estas variables del flujo se pueden estudiar en detalle tras hacer una descomposición de Reynolds.

Descomposición de Reynolds.

Esta técnica utilizada en la dinámica de fluidos para analizar flujos turbulentos, permite separar una variable física —en este caso, la velocidad— en una componente media—para el caso de la U — $\langle U \rangle$ y una componente fluctuante, tal que $U = \langle U \rangle + u'$, lo que facilita el estudio de las fluctuaciones turbulentas alrededor de un valor promedio. Esta técnica es de gran utilidad a la hora de comprender las complejas interacciones en el seno del flujo.

Por tanto, para el caso de estudio, se comienza por establecer un método de cálculo para la obtención de la media y de la fluctuación turbulenta de cada componente de la velocidad, a partir del valor de la velocidad calculado por el *solver*, $\mathbf{v} = (U, V, W)$. Cada componente de la velocidad representa dicha componente para un instante de tiempo y para un punto del dominio en concreto, por lo que será a través de la obtención de múltiples campos de velocidades, que se podrá dar un tratamiento estadístico a la velocidad.

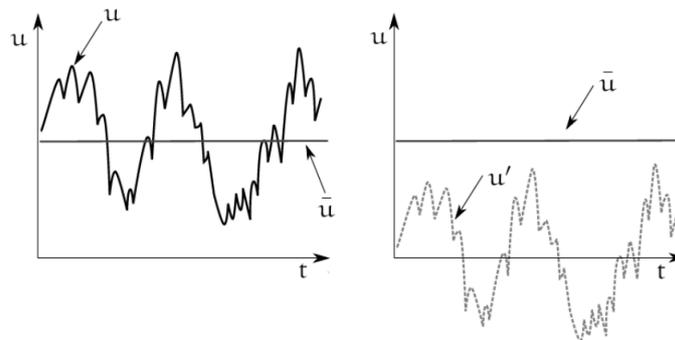


Figura 6.1: Descomposición de Reynolds. [Reggente, 2014]

De esta descomposición se puede concluir que:

Velocidad en los dominios físico y espectral

Como se puede observar en [Willis, 2017], tras aplicar el algoritmo de cálculo de la velocidad en el dominio de la tubería, se obtiene un conjunto de coeficientes de Fourier —representados por su parte real e imaginaria— para cada valor de la discretización radial. A través de una transformada inversa de Fourier de este campo, se puede obtener el campo de velocidades en variables físicas: r , para la coordenada radial, θ , para la coordenada azimutal y z , para la coordenada axial. Con ello, se puede hacer una distinción entre variables *colocadas* (hace referencia a la colocación por modos de Fourier a lo largo de la coordenada radial cuando se trata de variables en dominio de la frecuencia espacial) o variables *físicas* para referirnos al dominio (r, θ, z) .

Un hecho destacable es que tras la aplicación del algoritmo de resolución del campo de velocidades, estas velocidades se encuentran en variables *colocadas*, es decir, en el dominio de Fourier, con lo que a la hora de calcular las estadísticas se buscará emplear la velocidad en este dominio y no en el físico en la medida de lo posible, debido al coste computacional asociado al cambio de dominio. Por otro lado, para el cálculo de la presión, se emplean las variables en el dominio físico por comodidad, pues el hecho de que cada componente de la velocidad esté descompuesto a su vez en una serie de coeficientes reales y complejos representa una complejidad adicional, por lo que a efectos prácticos, tras el proceso de cálculo, se dispone de la velocidad en ambos dominios.

6.2.2. Presión, método de cálculo

Para el cálculo de la presión se ha decidido implementar un algoritmo que reconstruya el campo de presiones a partir de los resultados del *solver*, sin conllevar un coste computacional o un uso de memoria adicional.

Este algoritmo se ha desarrollado a partir de lo expuesto en [Willis, 2017] y de los cálculos representados por la 3.19.

Gracias a ello, ahora se dispone de un campo de presiones tanto en el dominio de Fourier como en el físico, análogamente a como ocurre con la velocidad. Este hecho es significativo no solamente para validar el campo de presiones, sino para comparar los *turbulent budgets*, magnitudes que se presentarán adecuadamente en el siguiente capítulo de resultados.

6.3. Metodología de exportación

El proceso de escritura y exportación de datos se lleva a cabo en la segunda mitad del programa *sta.f90*, en el que se pueden observar todas las funciones de la librería *MPI* y *HDF5* realizar las siguientes operaciones —detalladas en los comentarios del código—.

En primer lugar, todos los procesadores envían el dato en cuestión —por ejemplo, la distribución a lo largo del radio de la velocidad U — sumando todos los coeficientes de Fourier que lo representan, el valor de esta suma para cada valor del radio se manda a un procesador en concreto denominado comúnmente *master*, para finalmente, escribirlo en un archivo HDF5. Este proceso se realiza en paralelo, pues esta distribución radial de

sumas y la operación de escritura, se hace de manera distribuida.

Préstese atención al hecho de que es menos intensivo en memoria almacenar el valor de una suma para cada r que almacenar todos los coeficientes de Fourier en una matriz, distribuidas estas matrices a su vez en el dominio radial. Además, simplemente dividiendo el valor de esta suma entre el número total de muestras, se obtiene la media de esa magnitud.

6.4. Metodología para interpretación con matlab

Finalmente, los archivos generados con todas las magnitudes exportadas —véase *sta.f90* donde se recogen todas las variables calculadas— son leídos del archivo HDF5, interpretados, normalizados por el factor correspondiente y presentados gráficamente gracias a una herramienta de postproceso programada en Matlab, lo que permite automatizar el proceso de interpretación de los datos.

Capítulo 7

Resultados

Este capítulo recoge todo el trabajo dedicado al desarrollo del código incluido en el módulo *sta.f90* cuya finalidad es calcular los *turbulent budgets* y los distintos perfiles de velocidad característicos de la simulación, con la finalidad de llevar a cabo un proceso de validación por comparación con otras simulaciones llevadas a cabo por otros grupos de investigación de turbulencia en pared.

Se destaca que el número de muestras recogidas es de 100,000 con un tiempo de simulación aproximado de una semana en el equipo destinado a testeo, de capacidades de cómputo moderadas.

7.1. Datos de referencia

En primer lugar, se llevaron a cabo labores de investigación de trabajos previos sobre simulación de flujos turbulentos en tuberías, tomando consciencia de la dificultad técnica que entraña llevar a cabo esta labor debido a los altos costes computacionales y al hecho de que es necesario desarrollar herramientas *software* propias y adecuadas a cada estudio que se plantee llevar a cabo.

Seguidamente, tras tomar la decisión de adaptar un código DNS existente para cumplir con los requisitos necesarios de escalabilidad, tras la obtención de los primeros resultados, se procedió a buscar bases de datos para llevar a cabo las tareas de validación de los resultados [Yao et al., 2023].

[Enlace a la base de datos \[Yao et al., 2023\]](#)

7.2. Perfiles de velocidad y presión

Las primeras magnitudes representadas fueron las distribuciones radiales de las medias y de la perturbación de la velocidad y la presión. Se recuerda a que la variable aleatoria que representa cada componente de la velocidad y a la presión se le aplica una descomposición de Reynolds para su estudio.

7.2.1. Perfiles de velocidad

A continuación se representan las magnitudes obtenidas tras la simulación. En primer lugar se presenta la velocidad, y cabe destacar que se ha adimensionalizado con la velocidad de fricción U_τ para expresar $\mathbf{v}^+ = [U_r^+, U_\theta^+, U_z^+]$, posteriormente se actuará de igual manera con la presión, adimensionalizada con la velocidad de fricción al cuadrado U_τ^2 y por último, se representarán tres ejemplos de *turbulent budgets* cada uno adimensionalizado en unidades coherentes.

También es necesario aclarar que cabe esperar, la componente de la velocidad promedio más significativa será la axial, por ser esa la dirección principal del flujo. El promedio en las direcciones radial y azimutal es nula, y representa únicamente ruido numérico.

Componente radial de la velocidad U_r^+

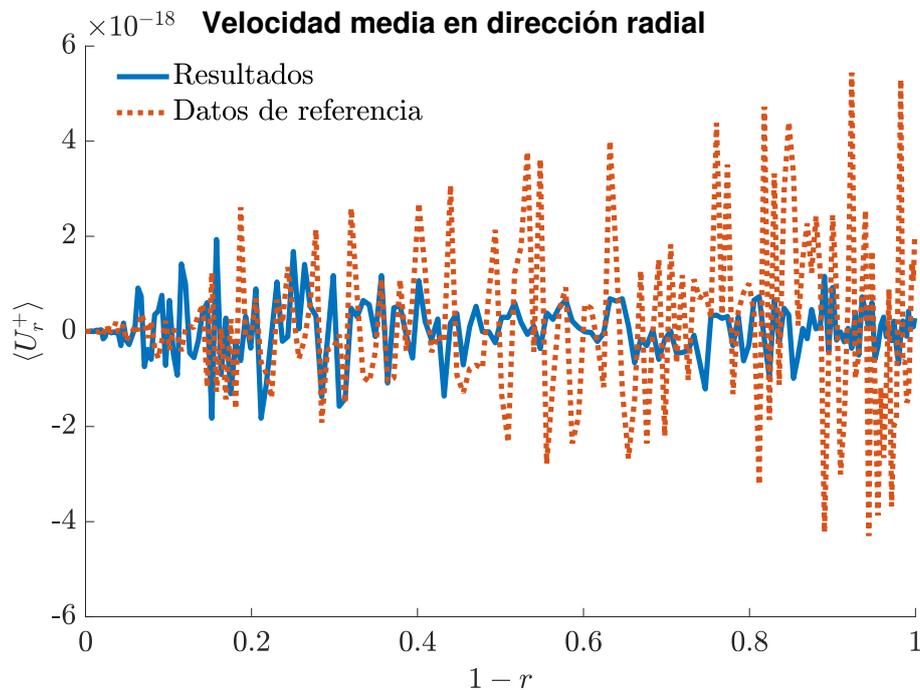


Figura 7.1: Como se puede observar en esta representación la componente radial de la velocidad promediada, se corrobora que es nula como cabría esperar, siendo obtenido únicamente ruido numérico

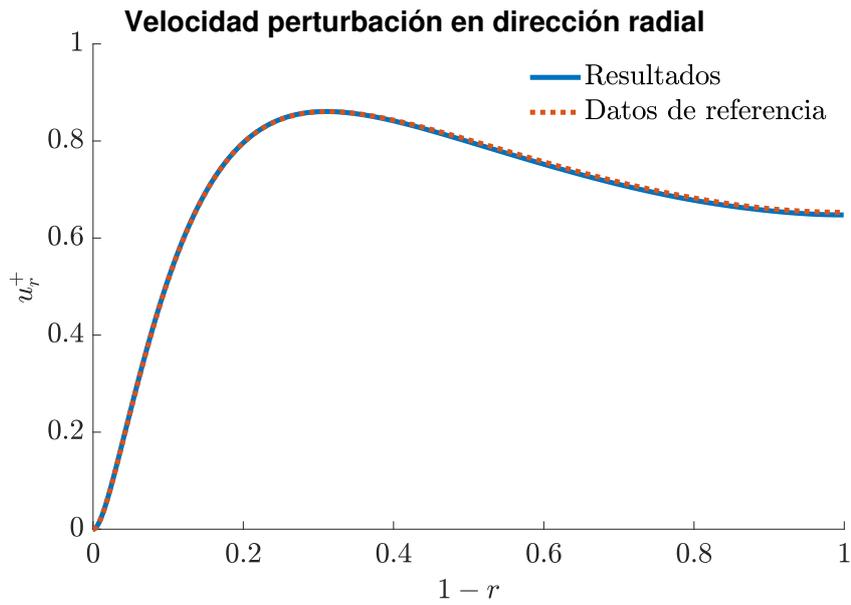


Figura 7.2: Se observa que en la región más cercana de la línea media, la componente de perturbación se suaviza, dado que el flujo en esa región se encuentra poco influido por la pared y se estabiliza. En la región cercana a la pared ($1-r=0,2$), se observa que la perturbación de la velocidad radial es del orden de un 60% la velocidad de fricción y siendo ésta máxima para $1-r=0,3$, donde se deduce al observar el resto de componentes perturbadas que se sitúa una región con alta intensidad turbulenta, en la pared ($1-r=0$) debido a la condición de no deslizamiento encontramos valores nulos.

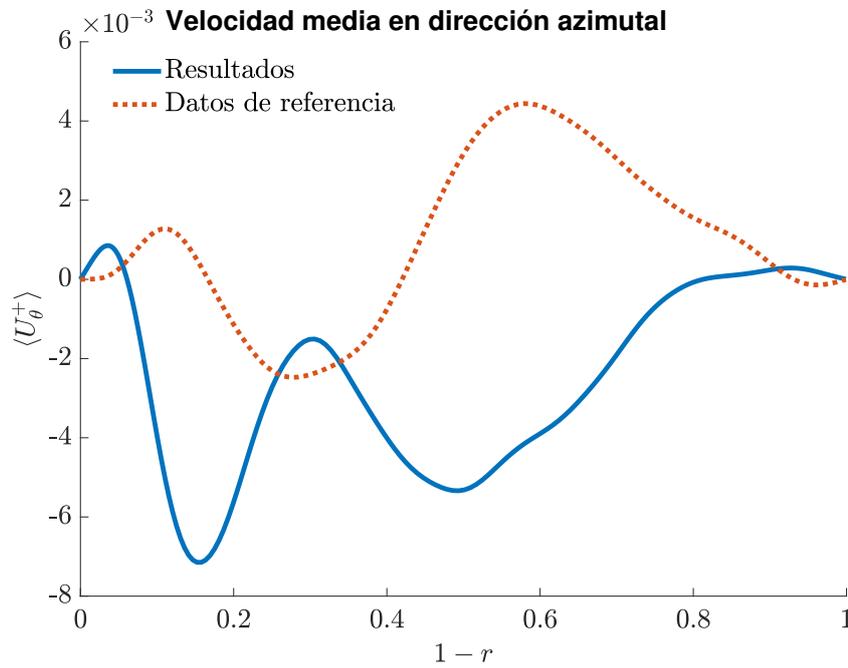
Componente azimutal de la velocidad U_θ^+ 

Figura 7.3: De manera similar a cómo ocurre con la componente radial, el promedio de la velocidad azimutal se considera ruido numérico.

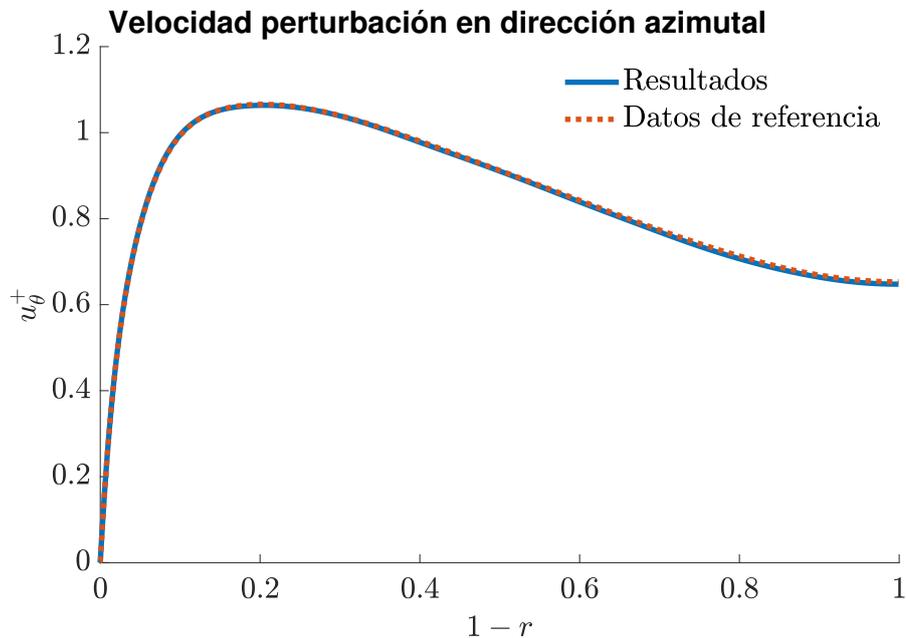


Figura 7.4: Para el caso de la velocidad de perturbación de la componente azimutal de la velocidad se observa el fenómeno que ya se presentó para la el caso de la velocidad radial, en la región entre $r = 0,2$ y $r = 0,3$ se puede encontrar una región de alta intensidad turbulenta.

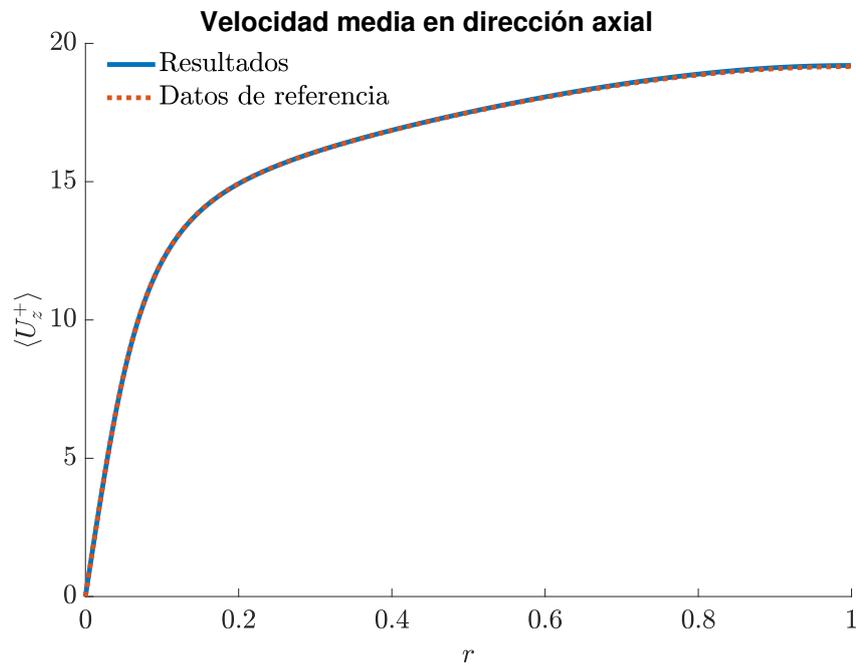
Componente axial de la velocidad U_z^+ 

Figura 7.5: En esta componente se puede observar que el orden de magnitud es superior a las demás debido a que es la componente principal del flujo. En la pared es nula debido a la condición de no deslizamiento y es creciente a medida que nos acercamos a la región central, donde el flujo se encuentra menos perturbado.

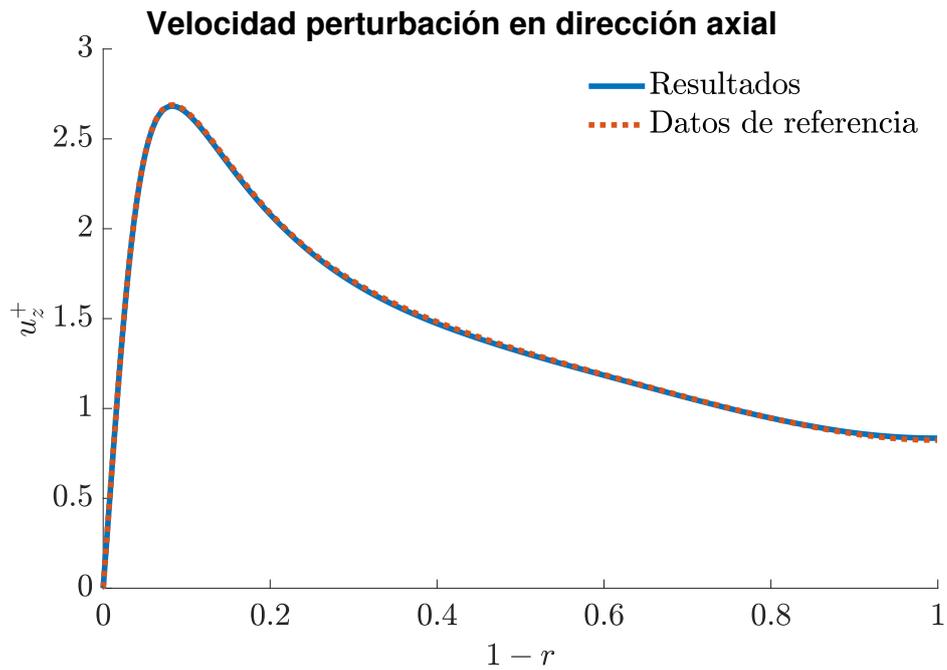


Figura 7.6: En esta figura se puede observar de igual forma que el orden de magnitud es superior al de las componentes de perturbación de las otras direcciones principales del flujo. La tendencia es la opuesta a la de la media, pues en regiones cercanas a la pared donde se acentúan más la influencia de ésta esta componente de la velocidad al igual que las demás es donde más se desvía de la media debido a la alta intensidad turbulenta. A medida que observamos regiones cercanas al centro de la tubería, el flujo se encuentra menos perturbado.

7.2.2. Perfil de la presión

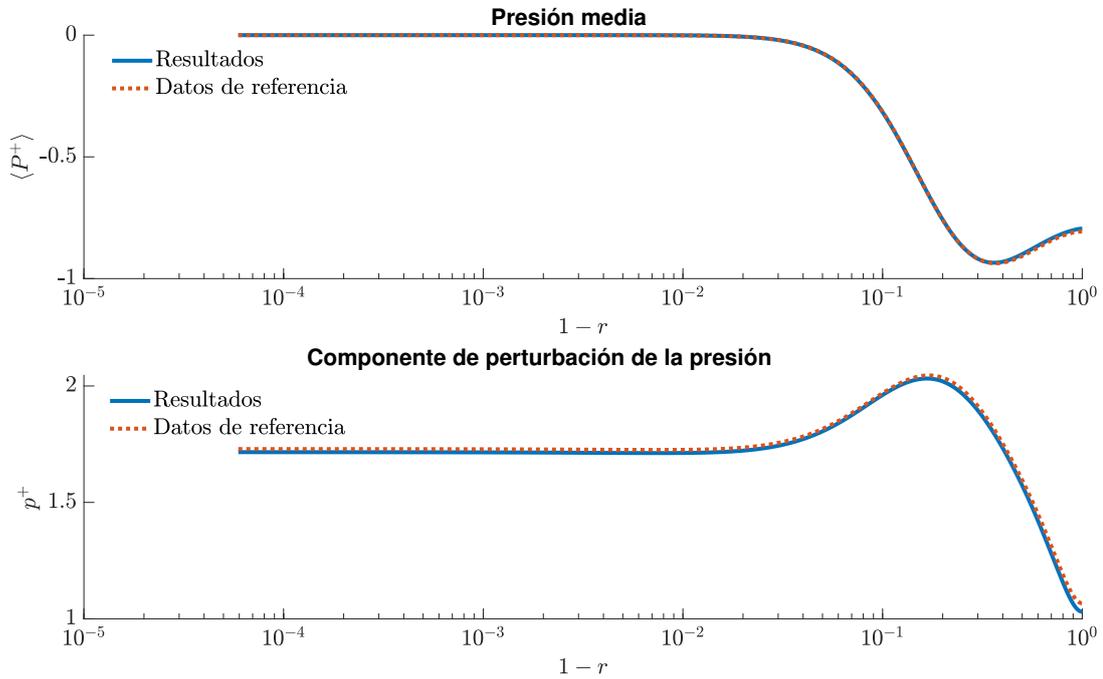


Figura 7.7: Se presentan la presión media y su perturbación, a través de las cuales se puede identificar una región en torno al eje axial de la tubería en el que se observa que la presión fluctúa generando zonas de bajas presiones.

7.3. Turbulent budgets

El término *turbulent budget* deriva del concepto de balance de energía cinética en la turbulencia, representada en la ecuación de conservación de la cantidad de movimiento por el tensor de esfuerzos de Reynolds, donde se consideran las fuentes, las transferencias y las disipaciones de energía turbulenta [Pope, 2000]. Esta ecuación está constituida por varios términos, relacionados cada uno con tránsitos de energía en el seno del flujo que hacen que magnitudes como la presión o la velocidad se transporten, o términos que representan la producción y disipación de energía cinética turbulenta, entre otros. En general, el análisis de los *turbulent budgets* implica descomponer las ecuaciones de Navier-Stokes, en términos de términos que representan diferentes fenómenos turbulentos.

El tensor de esfuerzos de Reynolds se descompone en los siguientes *turbulent budgets* [Kadu et al., 2020]:

$$0 = C_{ij} + \Pi_{ij} + D_{ij}^P + D_{ij}^T + D_{ij}^v + P_{ij} + \varepsilon_{ij} \quad (7.1)$$

En la 7.1 aparecen los siguientes términos:

- C_{ij} : término convectivo, nulo [Kadu et al., 2020].

- Π_{ij} : correlación de la deformación de la presión.
- D_{ij}^P : término de la difusión de la presión.
- D_{ij}^T : difusión turbulenta.
- D_{ij}^V : difusión viscosa.
- ε_{ij} : término de la disipación.
- P_{ij} : término de la producción.

A continuación se representan algunos de los términos más relevantes, obtenidos a través de expresiones que implican computar derivadas de la velocidad y la presión promediadas [Kadu et al., 2020].

7.3.1. Producción turbulenta.

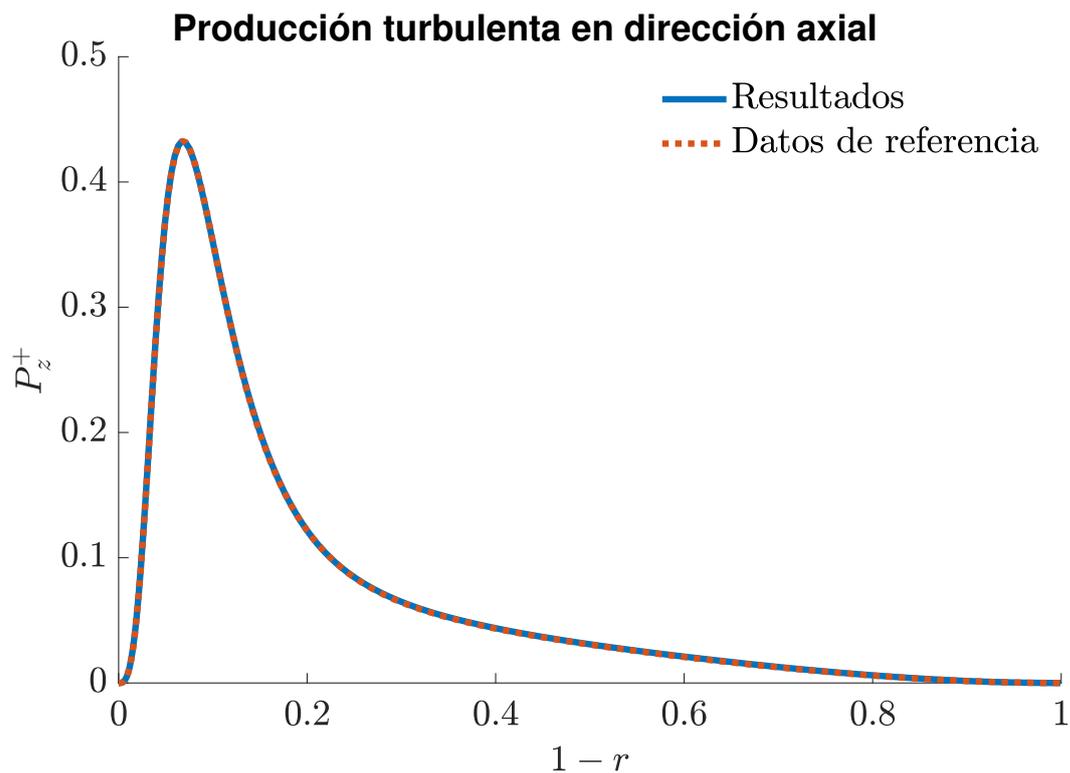


Figura 7.8: Se observa únicamente la componente axial del término de producción de energía turbulenta debido a que las otras dos son nulas. Como se puede inferir, en la región más cercana a la pared y en la que a partir de la observación de la perturbación de la velocidad hemos visto que es de alta intensidad turbulenta, es donde más energía turbulenta se produce, disminuyendo asintóticamente a 0 en las regiones próximas al eje axial.

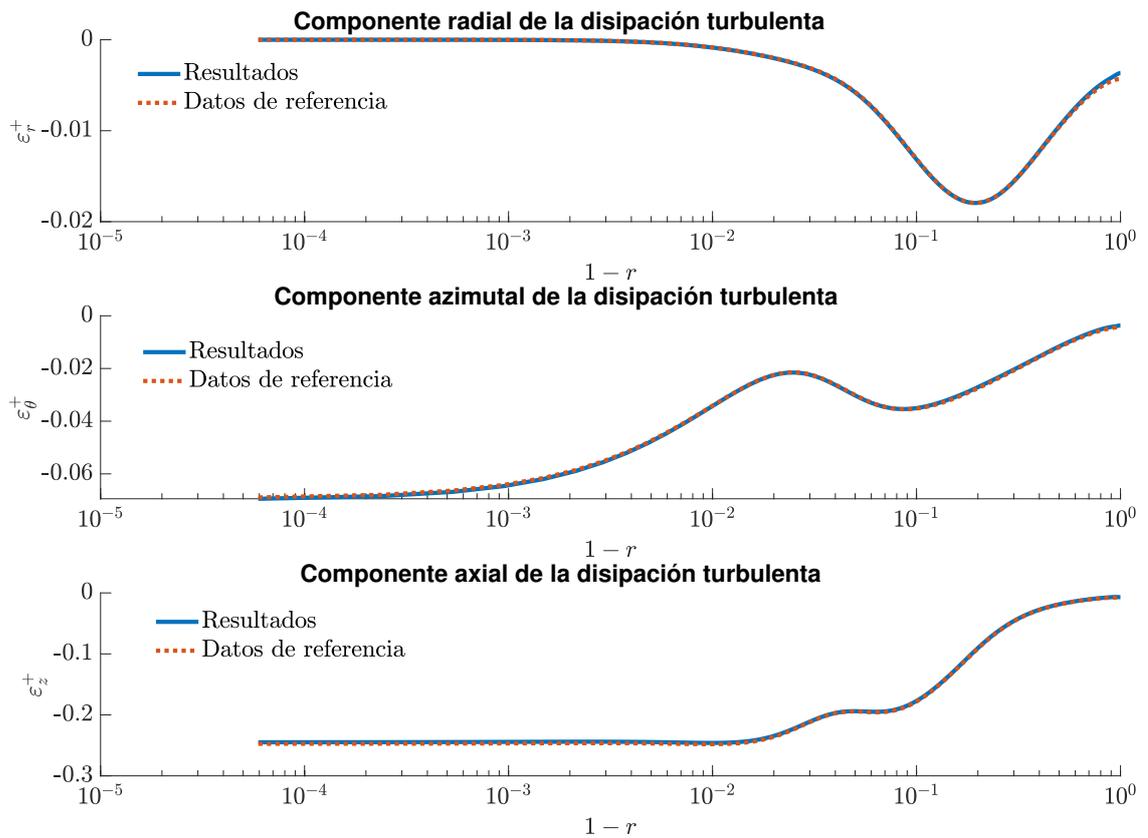


Figura 7.9: En esta figura se pueden observar las tres componentes de la disipación viscosa, donde además se pueden observar las diferencias existentes entre sus órdenes de magnitud. Su distribución radial indica que tienen una mayor presencia en regiones muy próximas a la pared, nótese $1-r$ en escala logarítmica.

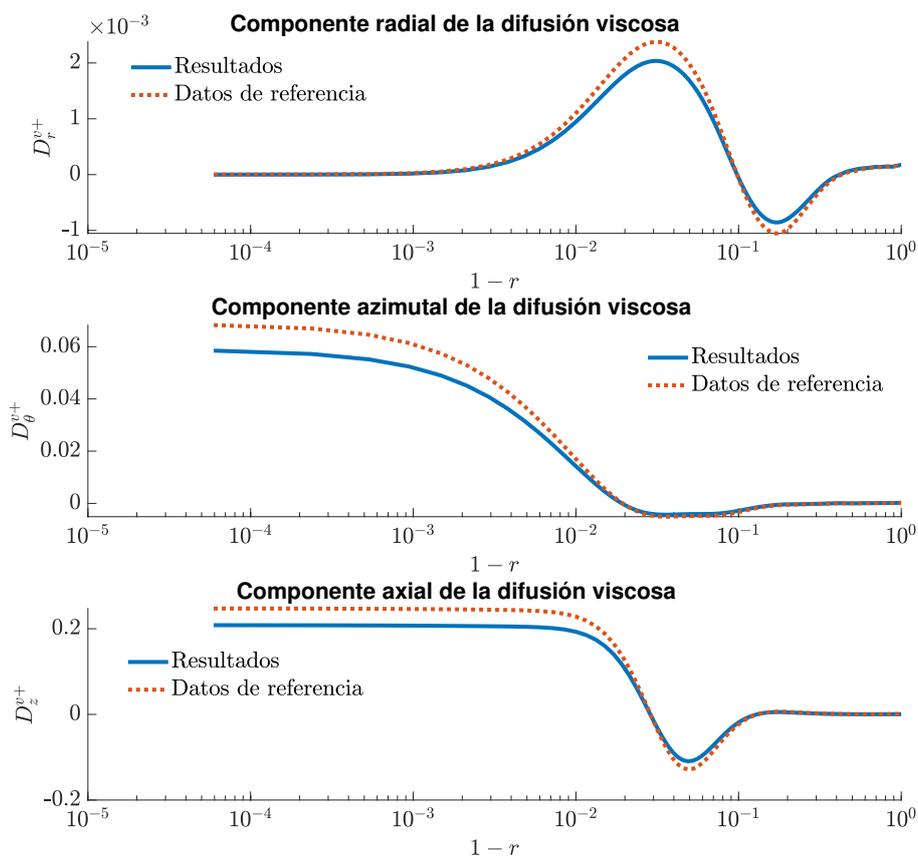


Figura 7.10: Por último se presenta la magnitud que indica la intensidad con la que los efectos viscosos se difunden de la pared al centro de la tubería. De la misma manera que ocurre con la disipación, tiene una mayor presencia en regiones muy próximas a la pared.

7.4. Resultados de especial relevancia

Además de los resultados anteriores, que más allá de representar los perfiles de velocidad y presión característicos del flujo y de permitir la obtención de los *turbulent budgets* como parte del proceso de validación del código, se han llevado a cabo diversos estudios y se han llevado a congresos de ámbito internacional resultados y conclusiones de gran relevancia en el ámbito del estudio de flujos turbulentos y de las simulaciones DNS, obtenidos gracias a la aportación realizada por el presente trabajo a través del código desarrollado para simulación DNS.

7.4.1. Hitos en resolución espacial.

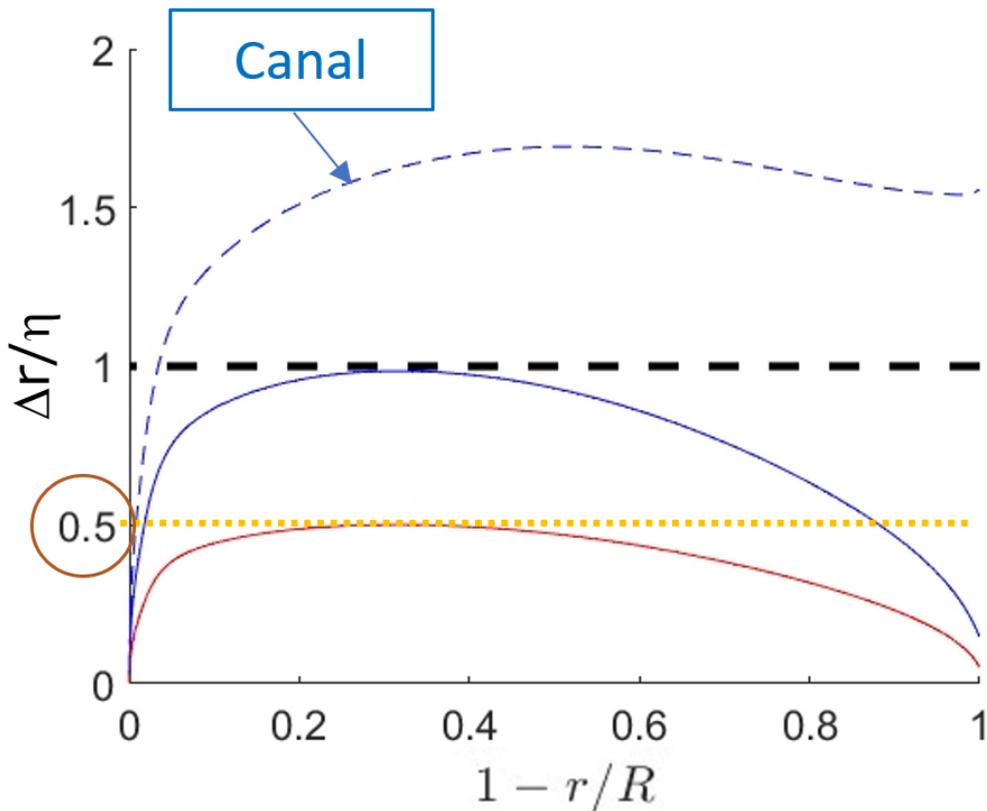
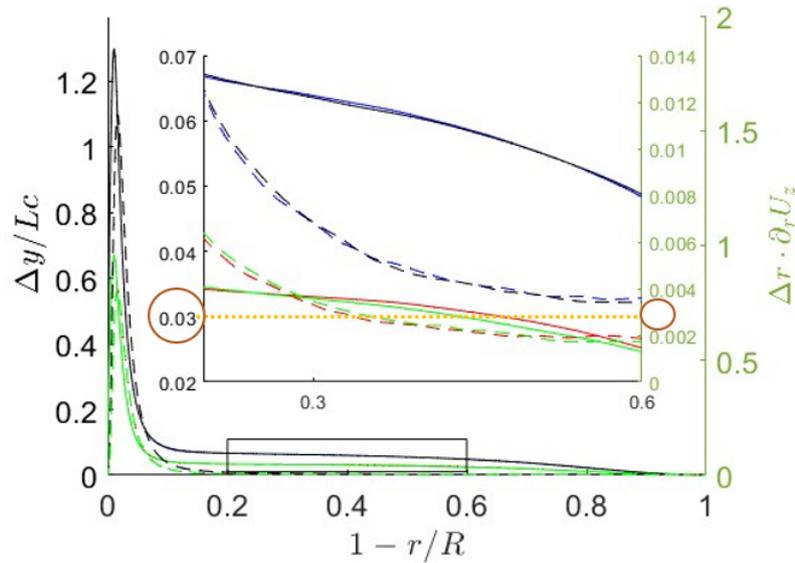


Figura 7.11: En esta figura se observa la distribución radial del espaciado entre puntos del dominio de cálculo para una simulación con $Re_\tau = 550$. En línea azul discontinua se representa el espaciado entre puntos, siendo éste aproximadamente un 50% más grande que la longitud de Kolmogorov. Por otro lado, se puede observar una línea continua de color azul que representa la distribución del espaciado entre puntos de la malla *LR*, *Low Resolution*, representativo de la resolución espacial del problema y por lo tanto, de la precisión de los resultados calculados. Por último, se muestra en color rojo una línea continua representativa del espaciado de malla para un caso de estudio de mayor resolución *HR*, *High Resolution* donde, como se puede observar que todos los puntos de la malla se encuentran entre sí a una distancia inferior a la mitad de la longitud de Kolmogorov, lo que se considera un hito y garantía de resultados precisos.



| Case | Line Color | Δy | es el espaciado para Canal |
|------|------------|------------|------------------------------|
| LRF | Blue | Δr | es el espaciado para Tubería |
| HRF | Red | η | es la longitud de Kolmogórov |
| LRC | Black | L_c | es la longitud de Corrsin |
| HRC | Green | | |

Figura 7.12: Se puede observar en la escala de la izquierda el espaciado de malla para un canal con $Re_\tau = 550$ y a la derecha el espaciado para el caso de una tubería. En este caso las distribuciones se comparan con respecto a la longitud de Corrsin, más grande que la longitud de Kolmogorov pero con la diferencia de que L_C tiene en cuenta la deformación turbulenta. Una vez más se observan que las distribuciones del espaciado entre puntos para el caso de las mallas generadas para el caso de baja resolución (LR) son del orden de entre el 5% y el 7% de la longitud de Corrsin y para los casos de alta resolución (HR), el espaciado entre puntos se reduce al entorno del 3% de L_C , con lo que se anticipa que estas mallas de tan alta resolución espacial son capaces de permitir simular resultados con precisión.

7.4.2. Modificación sobre la extensión de la subcapa logarítmica en la capa límite

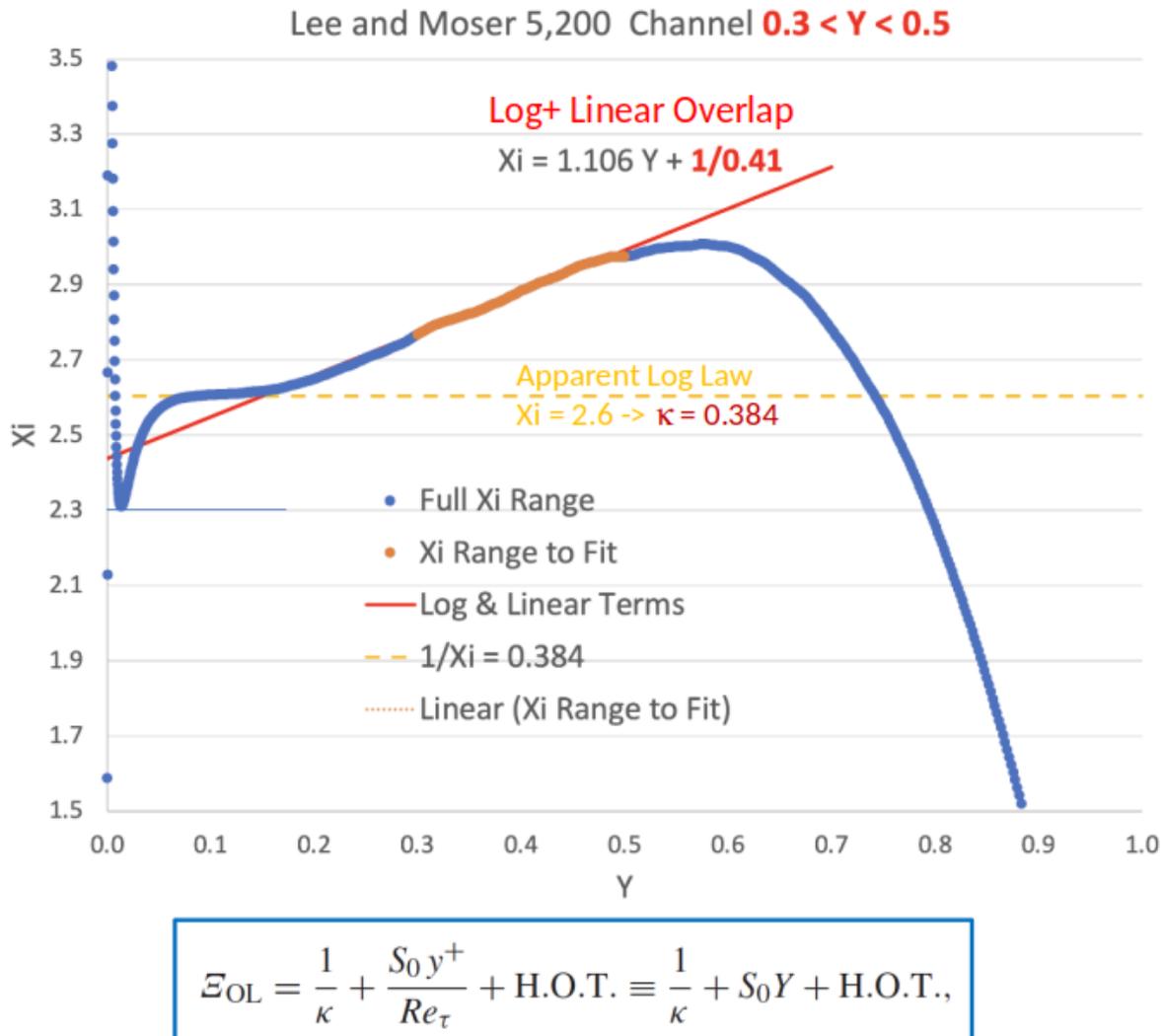


Figura 7.13: En este caso se observa cómo se ha inferido acerca de la existencia de una ley de ajuste para la subcapa logarítmica como una combinación de un término logarítmico y otro lineal, ajustándose mejor que un ajuste puramente logarítmico y se ha hipotetizado sobre la no universalidad de la constante de von Karman, pues como se observa, según el tipo de ajuste que se haga (observando un mejor ajuste para solape logarítmico-lineal), del código y de la relación de aspecto de los elementos del dominio, se obtiene un valor de los parámetros u otro.

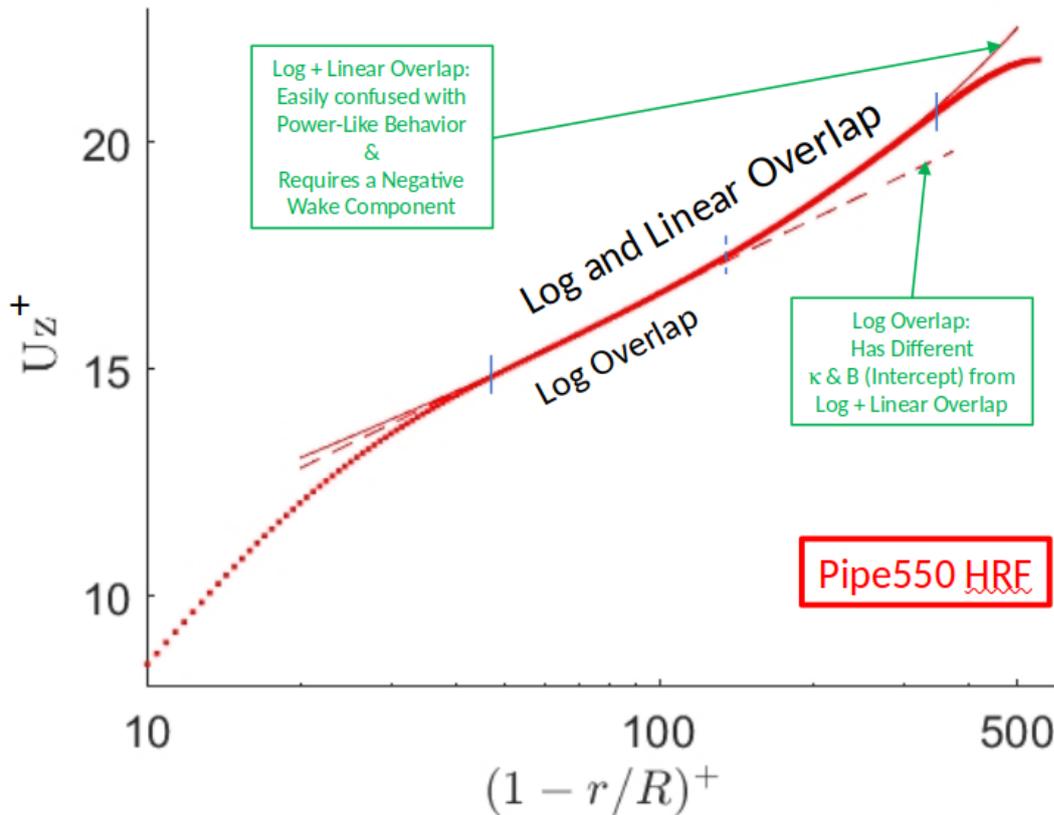


Figura 7.14: En esta figura se observan los dos ajustes, en línea naranja discontinua (sin considerar términos lineales) se observa que la región de ajuste es menor que si se emplea un ajuste logarítmico-lineal, representado por la línea continua naranja. Esto aporta una perspectiva adicional que logra mejores resultados de los logrados con la metodología existente, hasta ahora

7.4.3. Hallazgo de un criterio de convergencia objetivo conforme al *Eddie Turnover Time* (ETT)

En la siguiente imagen se puede observar la evolución con respecto al ETT de **poner con detalle**. De la figura se extrae que hasta que no transcurren del orden de 50 ETT's, la simulación no se puede dar por convergida adecuadamente, pues la magnitud representada se encuentra íntimamente relacionada con la precisión de la solución alcanzada.

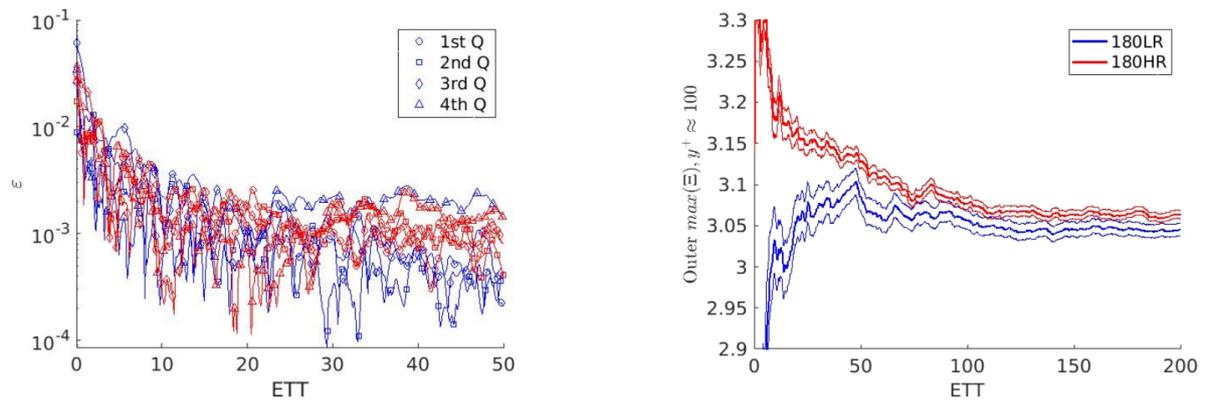


Figura 7.15: .

El hecho relevante es que tradicionalmente las simulaciones se ejecutan durante un periodo de tiempo comprendido entre los 10 y los 20 ETT's con lo que se concluye que es manera de proceder (que obedece a criterios de limitación del uso de recursos computacionales durante un tiempo demasiado largo) es errónea.

Capítulo 8

Pliego de condiciones y presupuesto

8.1. Condiciones técnicas y materiales

En cuanto a las condiciones técnicas necesarias del desarrollo del presente trabajo, se consideran tres elementos: la oficina de trabajo —sito en el Instituto Universitario de Matemática Pura y Aplicada (IUMPA), ubicado en la Ciudad Politécnica de la innovación— hardware —ordenador personal y equipos de cálculo, suministrado por el tutor— y el software —en su mayoría *open source* y alguna licencia concreta, especificada en el presupuesto—.

Las condiciones del puesto de trabajo se detallarán en la sección correspondiente a las condiciones laborales.

El hardware empleado como equipo personal es un ordenador portátil con las siguientes características:

- **Marca:** MSI.
- **Modelo:** Stealth 15M.
- **Sistema Operativo:** Windows 11.
- **Procesador:** Intel® Core™ i7-1185G7 (4 núcleos / 8 *threads*, frecuencia desde 1.2GHz hasta 4.8GHz).
- **Memoria RAM:** 16GB (2 x 8GB) SO-DIMM DDR4 3200 MHz.
- **Almacenamiento:** 512GB SSD M.2 2280 PCIe 3.0x4 NVMe.
- **Tarjeta gráfica:** NVIDIA® GeForce RTX™ 2060 Max-Q 6GB GDDR6.
- **Batería:** Integrada, 56Wh.
- **Pantalla:** Resolución FHD de 15.6" con 60 Hx tipo IPS.
- **Ratón:** Estándar.

El resto del hardware está compuesto por un equipo para testeo del código generado, empleado para supercomputación a escala reducida y tres supercomputadores.

El equipo de testeo es un ordenador con 80 unidades de proceso y 160GB de RAM.

Los equipos de supercomputación son los supercomputadores **MareNostrum**, **SuperMUC** y **Shaheen II** cuyas fichas técnicas exactas actualizadas son (clic sobre el nombre): MareNostrum4, SuperMUC y ShaheenII.

En cuanto al *software*, se han empleado herramientas de desarrollo usuales en este tipo de trabajos. Se ha empleado *Visual Studio Code* para el desarrollo de código fuente, las librerías *HDF5* de *Ubuntu*, el correspondiente compilador *h5pfc* y *Matlab R2021b* para el postproceso de los resultados, todo ello en el entorno del sistema operativo *Ubuntu 20.04*, con la licencia de estudiante proporcionada por la UPV, el resto de *software* es *open source*.

Para la redacción de la memoria, se ha utilizado la plataforma de edición de texto *online Overleaf* para la realización de la memoria en formato \LaTeX .

8.2. Condiciones del puesto de trabajo

Es fundamental tener en consideración los posibles riesgos a los que el autor puede verse expuesto al llevar a cabo un proyecto de esta naturaleza, dado que estos riesgos pueden repercutir tanto en el avance del proyecto como en la salud del individuo. Estos riesgos incluyen:

- **Seguridad:** Contactos eléctricos.
- **Higiene industrial:** Iluminación, ruido, temperatura, etc.
- **Ergonomía:** Fatiga visual y fatiga física por la postura de trabajo.

El Real Decreto 488/1997, emitido el 14 de abril, reúne los requisitos mínimos imprescindibles para asegurar la salud y seguridad del autor que desempeña tareas que involucran el uso de equipos con pantallas de visualización. A continuación, se presentan de manera resumida las condiciones mínimas que deben cumplirse en el lugar de trabajo, en estricta conformidad con este decreto, teniendo en cuenta los riesgos potenciales asociados:

- **Equipo:** debe considerarse ciertas variables como la luminosidad de la pantalla y su posición.
- **Periféricos:** teclado y ratón independiente para aumentar la comodidad del usuario conforme a criterios de ergonomía. Un monitor externo facilita la visión y reduce la fatiga visual.
- **Mesa:** debe tener dimensiones suficientes para la colocación del equipo y del material. Debe ser estable.
- **Asiento:** debe ser apropiado para largos periodos de tiempo así como libertad de movimientos. Se debe poder regular la altura y ajustar el respaldo.
- **Entorno de trabajo:** debe cumplir con los requisitos de espacio, iluminación, límite de ruido, rango de humedad, temperatura y emisiones.

8.3. Presupuesto

En esta sección se proporciona un presupuesto destinado a la ejecución del proyecto. Para su elaboración, se considerarán tanto los costes de la mano de obra como los requerimientos de equipamiento necesarios para llevar a cabo las tareas. A continuación, se exponen los gastos generales, los márgenes de beneficio de la empresa y la declaración del Impuesto de Valor Añadido (IVA).

8.3.1. Coste de personal

El coste de personal es el correspondiente al autor de este trabajo y al tutor académico. El coste es proporcional a las horas dedicadas al proyecto por parte del autor, donde se recogen las horas de realización de los diferentes estudios mostrados como la redacción de la memoria del Trabajo de Fin de Master. Teniendo en cuenta que el TFM abarca 13,5 créditos, y que según el Plan Bolonia cada crédito son 25 horas de trabajo, por lo que en total se corresponden a 337,5 horas invertidas. Estas horas se desglosan de la siguiente manera:

- **Desarrollo *software*:** 240 horas
- **Postproceso:** 27.5 horas
- **Redacción de memoria:** 70 horas

En la Tabla 8.1 se muestra el coste de cada trabajador y el total de la mano de obra. Se considera un sueldo de 21€/h para el autor y una asignación de 29.5€/h para el tutor.

| Concepto | Tiempo [h] | Sueldo [€/h] | Valor [€] |
|--------------|------------|--------------|-----------------|
| Autor | 337.5 | 21 | 7 087.50 |
| Tutor | 30 | 29.5 | 885.00 |
| Total | | | 7 972.50 |

Tabla 8.1: Coste de mano de obra

De esta forma, se obtiene que el presupuesto para la mano de obra suma un total de **SIETE MIL NOVECIENTOS SETENTA Y DOS EUROS CON CINCUENTA CÉNTIMOS (7 972.50 €)**

8.3.2. Coste de equipo informático

Otro de los costes del proyecto son los equipos informáticos y los *softwares* utilizados.

El equipo que se ha utilizado para la realización del proyecto es un ordenador portátil MSI Stealth 15M, cuyo valor de compra fue de 1 700.00 €. Teniendo en cuenta que se compró el equipo hace 3 años y asumiendo una amortización del 20%, el coste de amortización es $CA_{portatil} = 1020,00$ €.

En cuanto al *software* utilizado, el coste es el correspondiente a una licencia anual de Matlab, el resto representa un coste nulo por ser todo *open source*.

| Concepto | Valor [€] |
|---------------------------------|--------------|
| Ubuntu 20.04 | 0 |
| MATLAB | 400 |
| Microsoft Visual Studio Code | 0 |
| L ^A T _E X | 0 |
| Portátil | 1 020.00 |
| Total | 1 420 |

Tabla 8.2: Coste de soporte computacional

El coste del soporte computacional es de **MIL CUATROCIENTOS VEINTE EUROS (1 420€)**.

8.3.3. Presupuesto total del proyecto

El presupuesto total del proyecto queda representado en la siguiente tabla:

| Concepto | Valor [€] |
|-------------------------|------------------|
| Coste de personal | 7 972.50 |
| Coste computacional | 1 420.00 |
| Gastos generales (14 %) | 1 362.01 |
| IVA (21 %) | 2 258.45 |
| Total | 13 012.96 |

Tabla 8.3: Coste total

El coste total del proyecto es de **TRECE MIL DOCE EUROS CON NOVENTA Y SEIS CÉNTIMOS (13 012.96 €)**.

Capítulo 9

Trabajos futuros

Tras el proceso de desarrollo de la herramienta de simulación se han detectado puntos de mejora a implementar, no implementados durante la realización del presente trabajo debido a limitaciones temporales.

- Hacer *reshape* a las matrices más grandes del sistema aprovechando que han sido declaradas en memoria dinámica para almacenar elementos de menor tamaño, para evitar declararlos y reducir su coste asociado en memoria.
- Analizar con mayor profundidad la algoritmia aplicada en el proceso de resolución del campo fluido con la finalidad de acelerar el código y reducir su consumo de memoria.
- Hacer una guía de uso con las mejoras implementadas para que en trabajos sucesivos el desarrollador pueda continuar con el trabajo más rápidamente.

Capítulo 10

Conclusiones

Finalmente, se concluye que:

- Se han desarrollado herramientas eficientes de simulación y de postproceso para una simulación DNS con resultado satisfactorio de acuerdo con la finalidad académica y didáctica de un Trabajo Fin de Máster.
- Se han cumplido los objetivos de aprendizaje de la física de la turbulencia y de su estudio a través de la interpretación de los resultados y de la interpretación de los diferentes algoritmos de resolución implementados.
- Se ha ganado una visión más global de los métodos estadísticos para el estudio de la turbulencia a través del promediado de las magnitudes calculadas para el cálculo de los *turbulent budgets*.
- Se ha aprendido a programar en Fortran haciendo uso de la librería para cálculo paralelo MPI, sirviendo de apoyo al proceso de aprendizaje sobre el uso del cálculo en paralelo en simulaciones destinadas a estudiar la turbulencia.
- Se ha adquirido la habilidad de trabajar con equipos remotos de capacidades de computación avanzadas.
- Se ha garantizado la escalabilidad de un código gracias al empleo de memoria dinámica.
- Se han cumplido con los objetivos de desarrollo en plazo tras establecer la metodología de de trabajo semanal.

Además, es de especial interés que estos trabajos han sido expuestos en distintos congresos como *European Turbulence Conference 18*, celebrada en Valencia, la *2nd Spanish Fluid Mechanics Conference*, en Barcelona, la *iTi Conference of Turbulence X* en Italia y en noviembre de 2023 se expondrá en el *Annual Meetin of the American Physical Society Division of Fluid Dynamics* que tendrá lugar en en Washington y recibirá a más de 3500 asistentes de todo el mundo.

Por último es necesario destacar que gracias al desarrollo de la herramienta presentada en este trabajo se ha formado parte de un equipo internacional conformado por expertos

de gran relevancia en el campo del estudio de la turbulencia, participando de manera colaborativa con entidades como el *KTH Royal Institute of Technology* en Suecia, el *Illinois Institute of Technology* en Estados Unidos o la *King Abdullah University of Science and Technology* en Arabia Saudí.

Aunando todos estos factores, se han conseguido lograr todos los objetivos propuestos de manera satisfactoria, logrando además la obtención de unos resultados que han tenido una repercusión directa en el estado del arte del estudio de la turbulencia durante el periodo de 2022 y 2023, etapa en la que se ha llevado a cabo este Trabajo Final de Máster.

Bibliografía

- [Backus et al., 1957] Backus, J. W., Beeber, R. J., Best, S., Goldberg, R., Haibt, L. M., Herrick, H. L., Nelson, R. A., Sayre, D., Sheridan, P. B., Stern, H., et al. (1957). The fortran automatic coding system. In *Papers presented at the February 26-28, 1957, western joint computer conference: Techniques for reliability*, pages 188–198.
- [Batchelor, 1967] Batchelor, G. K. (1967). *An introduction to fluid dynamics*. Cambridge university press.
- [Castro and Vanderwel, 2021] Castro, I. P. and Vanderwel, C. (2021). *Turbulent Flows: an Introduction*. 2053-2563. IOP Publishing.
- [Culler et al., 1999] Culler, D., Singh, J. P., and Gupta, A. (1999). *Parallel computer architecture: a hardware/software approach*. Gulf Professional Publishing.
- [Hamzehloo et al., 2021] Hamzehloo, A., Lusher, D., Laizet, S., and Sandham, N. (2021). On the performance of weno / teno schemes to resolve turbulence in dns / les of high-speed compressible flows. *International Journal for Numerical Methods in Fluids*, 93.
- [Haveraaen et al., 2013] Haveraaen, M., Morris, K., and Rouson, D. (2013). High-performance design patterns for modern fortran. In *Proceedings of the 1st International Workshop on Software Engineering for High Performance Computing in Computational Science and Engineering*, SE-HPCCSE '13, page 1–8, New York, NY, USA. Association for Computing Machinery.
- [Kadu et al., 2020] Kadu, P., Sakai, Y., Ito, Y., Iwano, K., Sugino, M., Katagiri, T., Hayase, T., and Nagata, K. (2020). Budgets of turbulent kinetic energy and reynolds normal stresses in coaxial jets with and without swirl: A numerical study. *Journal of Energy and Environmental Sustainability*, 9.
- [Ku et al., 1987] Ku, H. C., Hirsh, R. S., and Taylor, T. D. (1987). A pseudospectral method for solution of the three-dimensional incompressible navier-stokes equations. *Journal of Computational Physics*, 70(2):439–462.
- [Kwiatkowski, 2002] Kwiatkowski, J. (2002). Evaluation of parallel programs by measurement of its granularity. In Wyrzykowski, R., Dongarra, J., Paprzycki, M., and Waśniewski, J., editors, *Parallel Processing and Applied Mathematics*, pages 145–153, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Piedrabuena et al., 2023] Piedrabuena, M., Hoyas, S., Sabaresh, E., Vinuesa, R., and Nagib, H. (2023). Importance of achieving convergence for computation of turbulence through pipes and other ducts.

-
- [Pirozzoli et al., 2022] Pirozzoli, S., Romero, J., Fatica, M., Verzicco, R., and Orlandi, P. (2022). Dns of passive scalars in turbulent pipe flow. *Journal of Fluid Mechanics*, 940:A45.
- [Pope, 2000] Pope, S. B. (2000). *Turbulent flows*. Cambridge University Press.
- [Pulicani and Ouazzani, 1991] Pulicani, J. and Ouazzani, J. (1991). A fourier-chebyshev pseudospectral method for solving steady 3-d navier-stokes and heat equations in cylindrical cavities. *Computers Fluids*, 20(2):93–109.
- [Reggente, 2014] Reggente, M. (2014). *Statistical Gas Distribution Modelling for Mobile Robot Applications*. PhD thesis.
- [Schlichting, 2000] Schlichting, H., G. K. (2000). *Boundary layer theory: Eighth edition*. Springer.
- [Tennekes and Lumley, 1972] Tennekes, H. and Lumley, J. L. (1972). *A first course on turbulence*. MIT Press.
- [Wenzel et al., 2019] Wenzel, C., Peter, J. M. F., Selent, B., Weinschenk, M. B., Rist, U., and Kloker, M. J. (2019). Dns of compressible turbulent boundary layers with adverse pressure gradients. In Nagel, W. E., Kröner, D. H., and Resch, M. M., editors, *High Performance Computing in Science and Engineering ' 18*, pages 229–242, Cham. Springer International Publishing.
- [Willis, 2017] Willis, A. P. (2017). The Openpipeflow Navier–Stokes solver. *SoftwareX*, 6:124–127.
- [Yao et al., 2023] Yao, J., Rezaeiravesh, S., Schlatter, P., and Hussain, F. (2023). Direct numerical simulations of turbulent pipe flow up to $re_\tau \approx 5200$. *Journal of Fluid Mechanics*, 956:A18.
- [Zhou et al., 2007] Zhou, G., Zhaofeng, L., Li, Z., and Zhang, Y. (2007). A tubular focused sonochemistry reactor. *Chinese Science Bulletin - CHIN SCI BULL*, 52:1902–1905.

