



GRADO EN INGENIERÍA DE LAS TECNOLOGÍAS INDUSTRIALES
ESCUELA DE INGENIEROS INDUSTRIALES

Estudio numérico sobre transferencia de calor en tuberías con flujo pulsante

Realizado por
FEDERICO PÉREZ GONZÁLEZ
Tutor
JOSE MANUEL LÓPEZ ALONSO
Cotutor
PALOMA GUTIÉRREZ CASTILLO

ASIGNATURA
TRABAJO FIN DE GRADO

UNIVERSIDAD DE MÁLAGA
MÁLAGA, 2023, SEPTIEMBRE



UNIVERSIDAD
DE MÁLAGA



Agradecimientos:

No puedo comenzar sin expresar mi profundo agradecimiento a mi familia, cuyo apoyo constante ha sido como un faro en esta travesía académica. Su amor incondicional y su ánimo han iluminado cada paso de este camino, y estoy eternamente agradecido por tenerlos en mi vida.

A mi pareja Marta, quiero agradecerle su presencia constante y apoyo inquebrantable. Tu paciencia y tu aliento me han sostenido durante los peores momentos. Este logro también es tuyo.

A mis respetados tutores del TFG, Jose Manuel López Alonso y Paloma Gutiérrez Castillo, les debo una deuda de gratitud imposible de medir. Su orientación experta y su dedicación incansable hicieron posible que este proyecto se convirtiera en una realidad. Al grupo de investigación de mecánica de fluidos de la Universidad de Málaga, les agradezco por brindarme un entorno de aprendizaje colaborativo que amplió mis horizontes académicos. La pasión y el compromiso que comparten son verdaderamente inspiradores.

A todos los que, de una forma u otra, han contribuido a este logro, quiero decirles sinceramente: ¡Gracias! Este TFG no habría sido posible sin su apoyo constante y su confianza en mí.



Resumen:

Este trabajo de fin de grado (TFG) presenta un análisis numérico sobre transferencia de calor dentro de una tubería con flujo laminar, donde la velocidad del fluido sigue un patrón senoidal (conocido como flujo pulsante senoidal). El objetivo principal de este estudio ha sido examinar cómo las variaciones en la amplitud y frecuencia de esta onda senoidal impactan en la cantidad de calor que el fluido puede transportar en comparación con un flujo constante sin pulsaciones.

Las simulaciones realizadas han revelado que existe un pequeño rango de frecuencias y amplitudes en el cual la transferencia de calor experimenta un ligero aumento en comparación con un flujo constante (este aumento es siempre inferior al 3%). No obstante, en un amplio rango de frecuencias y amplitudes, se observa una reducción significativa en la transferencia de calor en comparación con el flujo no pulsante. Esta reducción puede ser considerable, llegando a superar el 70% para valores elevados de la amplitud y frecuencia.

Además, se ha demostrado que las combinaciones específicas de frecuencias y amplitudes que conducen a un aumento en la transferencia de calor están relacionadas con velocidades promedio cercanas a la pared de la tubería que son mayores que las del caso sin pulsaciones. Por otro lado, las combinaciones que resultan en una disminución de la transferencia de calor muestran velocidades promedio cercanas a la pared más bajas que en el caso de un flujo constante sin pulsaciones.

Abstract:

This undergraduate thesis presents a numerical analysis of heat transfer within a pipe with laminar flow, where the fluid velocity follows a sinusoidal pattern (referred to as sinusoidal pulsating flow). The main objective of this study has been to examine how variations in the amplitude and frequency of this sinusoidal wave impact the amount of heat that the fluid can transport compared to a constant flow without pulsations.

The conducted simulations have revealed that there exists a small range of frequencies and amplitudes in which heat transfer experiences a slight increase compared to a constant flow (this increase is always less than 3%). However, within a wide range of frequencies and amplitudes, a significant reduction in heat transfer is observed compared to the non-pulsating flow. This reduction can be substantial, exceeding 70% for high values of amplitude and frequency.

Furthermore, it has been demonstrated that specific combinations of frequencies and amplitudes that lead to an increase in heat transfer are associated with average velocities near the pipe wall that are higher than those in the case of non-pulsating flow. Conversely, combinations resulting in decreased heat transfer show average velocities near the wall that are lower than those in the case of a constant flow without pulsations.



Índice de contenidos

Nomenclatura	5
1. Introducción	7
1.1. Contexto y Antecedentes	7
1.1.1. Problemas climáticos	7
1.1.2. La evolución tecnológica de la tubería	8
1.2. Objeto de estudio	10
1.2.1. Transferencia de calor en tuberías	10
1.2.2. Estrategias para modificar la transferencia de calor en tuberías	10
1.3. Estado del Arte	11
1.4. Motivaciones y Competencias	11
2. Descripción del problema y metodología	13
2.1. Problema a resolver	13
2.1.1. Ecuaciones a resolver	15
2.1.2. Parámetros adimensionales	16
2.1.3. Condiciones iniciales, de contorno e imposición del flujo pulsante senoidal	17
2.2. Software de Simulación	18
2.2.1. NsPipe_flow	18
2.3. Lanzamiento a Picasso de las simulaciones	19
2.3.1. Scripts y ficheros de trabajo	19
2.4. Metodología: Simulaciones y postprocesado	21
2.4.1. Parámetros de control	21
2.4.2. Parámetros adicionales	22
2.4.3. Postprocesado de datos	23
3. Resultados de las simulaciones	25
3.1. Análisis del factor de impacto	25
3.2. Análisis de los perfiles de velocidad	29
3.2.1. Desfase de las ondas	33
4. Conclusiones y discusión de los resultados	35
Bibliografía	36
Apéndice A. Códigos	39
A.1. NsPipeFlow Picasso	39

A.1.1.	Ejemplo de input_nsPipeFlow	39
A.1.2.	Ejemplo de lanza.sh	40
A.2.	Postprocesado	41
A.2.1.	Postbueno.m	41
A.2.2.	generar_color_map.m	50
A.2.3.	trat_perfil_optimizado	52
A.2.4.	obtencioncurvas.m	56
A.2.5.	desfase.m	58

Nomenclatura

α	Coefficiente de difusividad térmica (m^2/s)
κ	Coefficiente de dilatación térmica ($^{\circ-1}$)
Λ	Factor de impacto
∇	Operador nabra
ν	Viscosidad cinemática (m^2/s)
Ω	Amplitud de la onda senoidal
Φ	Temperatura adimensional
ϕ	Desfase entre ondas ($^{\circ}$)
ρ	Densidad del fluido ($\frac{kg}{m^3}$)
\vec{v}	Velocidad (m/s)
C_p	Calor específico del fluido ($\frac{J}{kgK}$)
h	Coefficiente de transferencia de calor ($\frac{W}{Km^2}$)
k	Conductividad térmica ($\frac{W}{Km}$)
L	Longitud característica (m)
Nu	Número de Nusselt
P	Periodo de la onda de velocidad media (s)
Pe	Número de Peclet
Pr	Número de Prandtl
R	Radio de la tubería (m)
Re	Número de Reynolds
t	Tiempo de simulación (s)
T_w	Temperatura de la pared (K)
t_{min}	Tiempo donde se encuentra un mínimo de la onda (s)
v_{bulk}	Velocidad media del fluido (m/s)
w	Frecuencia de la onda ($\frac{rad}{s}$)

Wo Número de Womersley



CAPÍTULO 1

Introducción

1.1. Contexto y Antecedentes

1.1.1. Problemas climáticos

El cambio climático es uno de los mayores desafíos que ha enfrentado la humanidad. La ciencia está centrada en encontrar soluciones a este problema, que está dando lugar a incrementos irreparables en la temperatura media del planeta y a la aparición de catástrofes climáticas, como inundaciones y tormentas másivas.

Para mitigar dichos problemas climáticos, la Unión Europea está creando planes de contingencia para disminuir la polución, mejorar la eficiencia de los procesos contaminantes y fomentar el uso de energías renovables. Dichas medidas se deberán cumplir completamente en 2050.

Los reportes anuales exponen los niveles de contaminación actuales y las pautas para desarrollar una Europa Verde con contaminación cero. En el reporte de este año [1] se muestra como gran parte de los países pertenecientes a la **EU** no cumplen con los estándares exigidos por la **Agencia Europea del MedioAmbiente (EAA)**. Durante la pandemia del *COVID* – 19 se llegaron a los límites exigidos por la **EU** conllevando una mejora en el medioambiente y la biosfera. Esto fue, sin embargo, un espejismo debido a la congelación de la actividad industrial. [2]. Centrándonos en España. 1.1 observamos los niveles de polución obtenidos desde el 2004 hasta la actualidad, demostrando una reducción de la contaminación pero sin obtener los niveles exigidos por la **EU**.

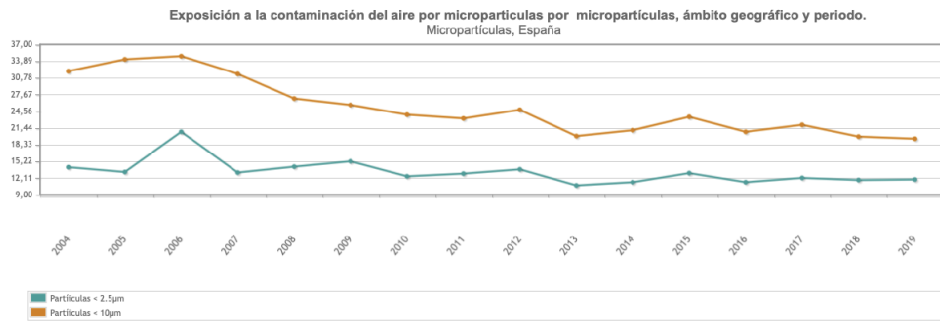


Figura 1.1: Diagrama de polución en España [1]

El principal obstáculo con el que se enfrenta esta transición hacia una Europa Verde es la influencia de intereses políticos. Estos intereses pueden ralentizar o acelerar el avance hacia un planeta libre de contaminación, según sus propias agendas. En particular, los países que cuentan con reservas naturales de combustibles fósiles tienden a presionar en contra de la creación de nuevas infraestructuras verdes, lo cual dificulta la optimización y el desarrollo de tecnologías limpias y sostenibles.

1.1.2. La evolución tecnológica de la tubería

El transporte de fluidos ha sido un desafío ancestral para la humanidad, presente desde los albores de las civilizaciones. Las tuberías se han erigido como una solución fundamental para satisfacer la necesidad de desplazar esos fluidos, ya sea para abastecer, irrigar o evacuar desde su origen hasta su destino requerido.

La historia registra evidencias arqueológicas e infraestructurales de la época romana, quienes destacaron como pioneros en el transporte forzado de fluidos. Sus ingeniosos acueductos permitieron llevar agua a los asentamientos humanos, conduciéndola desde su punto de origen hasta las zonas civilizadas. Posteriormente, los musulmanes desarrollaron acequias hechas de arcilla y piedras, mientras que en las dinastías chinas se utilizaba el bambú como canalizador de agua. [2]

Con el avance tecnológico, se descubrió la utilidad del cobre en Oriente Medio, extendiendo su aplicación a otros fluidos, además del agua. A partir del siglo XVIII, se inició el desarrollo de las primeras técnicas constructivas para sistemas de saneamiento y abastecimiento modernos. Sin embargo, el gran salto tecnológico ocurrió en el siglo XIX con la implementación de tuberías metálicas de hierro y plomo. En esta misma época, en Estados Unidos, se perfeccionó la tecnología de fabricación de tuberías de acero, permitiendo la construcción de ductos de alta calidad. [2]

En el siglo XX se presenció la aparición de nuevos materiales que, en gran medida, reemplazaron a los mencionados anteriormente, ya sea por sus cualidades mecánicas, costos de fabricación o por su menor impacto ambiental, siendo la sustitución del plomo un caso relativamente reciente [2].

En la actualidad, las tuberías se han convertido en un recurso de gran valor para nuestra sociedad, sin las cuales no podríamos disfrutar de los lujos y la calidad de vida que hoy consideramos fundamentales. Además de su uso en el ámbito doméstico, las tuberías

desempeñan un papel crucial en la industria, especialmente en la industria energética, donde las mejoras en los materiales han permitido nuevos usos y avances significativos.

- **Usos domésticos.**

A día de hoy, disfrutamos de ciertos lujos que consideramos esenciales, como la calefacción y los refrigeradores que nos permiten conservar nuestros alimentos de manera eficiente. Los considerables desarrollos en estos campos han mejorado significativamente la eficiencia de estos bienes.

- **La industria energética.**

El ser humano ha progresado notablemente gracias a su capacidad de manejar grandes cantidades de energía, lo que ha permitido aumentar nuestro tiempo disponible y reducir la duración de tareas que antes podrían llevar toda una tarde, como lavar la ropa. Estos avances en la industria energética han impulsado el progreso en otros campos, como la tecnología, creando un ciclo de retroalimentación positiva.

Durante muchos años, hemos dependido en gran medida de los combustibles fósiles para obtener grandes cantidades de energía a precios bajos. Sin embargo, estos combustibles fósiles generan gases nocivos para el medioambiente, como el NO_x y gases de efecto invernadero, provocando daños irreparables en la biosfera y, por ende, en nuestras vidas. Por ello, estamos enfocando nuestros avances tecnológicos en la generación de energías renovables, tales como molinos eólicos, plantas solares térmicas o fotovoltaicas y energía geotérmica, entre otras opciones.

En la actualidad, aproximadamente el 20 % de la energía eléctrica se produce mediante sistemas de bombeo, de los cuales el 25-50 % se utiliza en la propia industria [3]. Se ha comprobado que entre el 30-50 % de la energía consumida por un sistema de bombeo se puede reutilizar mediante sistemas de control adecuados, lo que conlleva una disminución en la contaminación y un mayor aprovechamiento de la energía, en consonancia con las demandas de la Unión Europea [1].



1.2. Objeto de estudio

Hemos comentado el desarrollo de los materiales utilizados en las tuberías; sin embargo, todavía existen procesos internos que no comprendemos completamente. Un mayor conocimiento y comprensión de estos procesos internos de los fluidos resultará en una notable mejora en los procesos de obtención y uso de energía. En particular, en este **Trabajo Final de Grado (TFG)** nos centraremos en el estudio de la transferencia de calor dentro de una tubería.

Generalmente, el flujo del fluido a través de una tubería se encuentra en régimen turbulento. No obstante, en este proyecto, nos hemos centrado en régimen laminar y estudiaremos el caso particular en el que la velocidad del fluido varía en el tiempo de forma senoidal. Este problema se conoce como flujo pulsante senoidal. El significado y la relevancia de este flujo se explicarán en los capítulos siguientes. Nuestro objetivo es estudiar como varía la transferencia de calor dentro de la tubería en función de la amplitud y frecuencia de la onda senoidal.

1.2.1. Transferencia de calor en tuberías

La transferencia de calor es un fenómeno físico fundamental que ocurre cuando existe una diferencia de temperatura entre dos cuerpos [4][5]. En este proceso, la energía fluye desde el cuerpo de mayor temperatura hacia el de menor temperatura, hasta que se alcanza el equilibrio térmico. En el contexto específico de las tuberías, la transferencia de calor se cuantifica mediante el número de Nusselt (Nu) 1.1. Este número adimensional relaciona la convección con la conducción en la superficie de las tuberías. Un aumento en el valor de Nusselt implica una mejora considerable en la transferencia de calor convectiva, lo que a su vez resulta en un aumento del coeficiente de fricción y la energía requerida para el bombeo.

$$Nu = \frac{hL}{k} = \frac{\text{trans. convectivo del fluido}}{\text{conduccion del material}} \quad (1.1)$$

1.2.2. Estrategias para modificar la transferencia de calor en tuberías

Existen dos estrategias claramente diferenciadas que permiten modificar la transferencia de calor en tuberías: métodos pasivos y métodos activos

- **Métodos Pasivos.**

Consiste en modificar la geometría de la tubería para incrementar las turbulencias cerca de la pared tuberías. No requiere energía adicional.

- **Métodos Activos.**

Consiste en la adición de nuevos mecanismos de impulsión o de inducción de turbulencias. Requieren energía adicional.

En este proyecto nos centraremos en los métodos activos. Las técnicas de dichos métodos incluyen la rotación de la tubería, la aplicación de campos magnéticos o eléctricos, usar flujos transitorios y flujos pulsantes como es el caso de dicho informe. Estos métodos son

complicados de aplicar pero permitirán una mayor variación del comportamiento del fluido y un mayor control sobre la transferencia de calor.

1.3. Estado del Arte

La transferencia de calor en el interior de una tubería con flujo senoidal ha sido estudiada durante mucho tiempo pero sin obtención de datos claros. La imposición de fluidos transitorios es una forma de aumentar la transferencia de calor bajo ciertas condiciones, por ejemplo, el estudio publicado en [6] mostró un incremento de transferencia de calor en aire bajo condiciones pulsantes en el rango de números de Reynolds de 4000 a 8000. En flujos laminares pulsantes, [7] realizó un estudio analítico de la transferencia de calor en un flujo laminar incompresible, concluyendo que la transferencia de calor incrementa proporcionalmente a la frecuencia. Al contrario, en el experimento publicado en [8], se encontró que la transferencia de calor o bien no varía con respecto al caso estacionario, o disminuye de forma sustancial para frecuencias grandes. [9] obtuvieron una expresión analítica de la velocidad, la distribución de temperatura y del número de Nusselt entre dos placas paralelas y observaron que dicho Nusselt aumentaba gradualmente con la disminución del número Prandtl. [10] encontraron numéricamente un aumento de la transferencia de calor para grandes frecuencias y pequeñas amplitudes. Esta conclusión fue más tarde corroborada por los experimentos presentados en [11]. Por otro lado, [12] obtuvieron que dicho aumento ocurre en ciertos rangos de amplitud y frecuencia, que no necesariamente corresponden a pequeñas amplitudes y grandes frecuencias. Finalmente [13], concluyó que la transferencia de calor en flujo pulsante laminar es idéntica al del caso no pulsante. Este es sin embargo un resultado dudoso debido a la utilización de una mala malla computacional y a la no convergencia mostrada en el artículo.

Del párrafo anterior se puede concluir que las investigaciones realizadas sobre este tema arrojan resultados inconclusos, evidenciando discrepancias entre los investigadores. Se han llevado a cabo investigaciones experimentales, donde se ha descubierto que existen mejoras y pérdidas en el coeficiente convectivo bajo ciertas condiciones de amplitud, frecuencia y número de Prandtl. Sin embargo, debido a las numerosas discrepancias encontradas, aún no es posible determinar con precisión los rangos específicos de mejora del número de Prandtl.

En cuanto a las investigaciones computacionales, son de muy baja calidad y no han podido verificar de forma fiable los resultados experimentales.

1.4. Motivaciones y Competencias

La utilización un potente software de simulación nos ha brindado la capacidad de obtener resultados concluyentes sobre el problema objeto de estudio, permitiéndonos comprender el comportamiento del fluidos y ampliar nuestro conocimiento sobre ello. Además, esta aproximación es más económica en comparación con las investigaciones experimentales, lo que la hace más accesible y eficiente.



En este proyecto, se creará una base de datos de simulaciones que posteriormente serán validadas mediante investigaciones experimentales. Esta base de datos estará disponible públicamente para el beneficio de la comunidad científica. La transferencia de calor en tuberías con flujo senoidal, como se ha explicado en apartados anteriores, ha sido objeto de estudio y controversia en el mundo científico, sin lograr resultados aceptables que permitan su total comprensión.

Sin embargo, tras revisar numerosos artículos científicos sobre esta temática, hemos llegado a la conclusión de que es posible modificar de manera significativa la transferencia de calor mediante la imposición de un flujo pulsante, lo que podría beneficiar en gran medida los procesos que requieren del transporte o transformación de energía mediante fluidos.

Competencias a obtener:

- Manipulación de un superordenador (Picasso).
- Introducción a nuevos lenguajes de programación (FORTRAN 90, MPI, OPENMP).
- Manipulación de una amplia base de datos.
- Interpretación de datos con su significado físico.
- Mejora de competencias en otros idiomas.

CAPÍTULO 2

Descripción del problema y metodología

2.1. Problema a resolver

Se considera el movimiento de un fluido de viscosidad μ , densidad ρ y difusividad térmica α a través de una tubería horizontal de radio R , rugosidad despreciable y paredes de conductividad térmica κ (ver esquema de la figura 2.1). La tubería se encuentra sometida a un flujo de calor constante \dot{q}_w , de forma que la temperatura del fluido incrementa linealmente desde la entrada a la salida de ésta.

En el presente estudio consideramos únicamente el caso de flujo laminar, donde el campo de velocidad sólo tiene componente en la dirección axial (eje z de la figura). Además, en condiciones de laminaridad, la velocidad y la presión dependen únicamente de la coordenada radial, simplificando de manera notable la simulación numérica del problema. El caso de la temperatura es algo más complejo, puesto que debido al flujo de calor impuesto, presenta variación tanto en dirección radial como axial. Sin embargo, como se mostrará luego en este informe, es posible reescalar la temperatura para eliminar dicha dependencia axial y resolver ecuaciones cuyas variables sólo tienen dependencia radial.

El objetivo principal del presente estudio es cuantificar la transferencia de calor asociada al fluido mediante el cálculo del número de Nusselt (ver sección 1.1 para ver su definición). Para el caso de flujo laminar en tuberías con velocidad media constante (caso correspondiente al perfil de velocidad mostrado en verde en la figura 2.1), el valor del número de Nusselt es constante e igual a $4,36$. Este valor puede ser obtenido fácilmente de forma analítica (ver, por ejemplo, [14]). La novedad del presente estudio radica en el hecho de que el flujo laminar considerado no avanza con velocidad media constante, sino que la velocidad varía periódicamente en el tiempo de forma senoidal, y por lo tanto las características de los perfiles de velocidad también varían con el tiempo. La figura 2.2 muestra un ejemplo de la variación de la velocidad media (normalizada por la velocidad media del ciclo) en el tiempo en una de nuestras simulaciones.

Tal como se ha explicado en la sección 1.3, la transferencia de calor asociada al fluido en estas condiciones no se conoce con certeza debido a que existen numerosas contradicciones en los estudios realizados. Motivados por esta observación, hemos realizado un estudio numérico sistemático sobre como la amplitud y la frecuencia de dicha onda senoidal afecta a la transferencia de calor en este sistema.

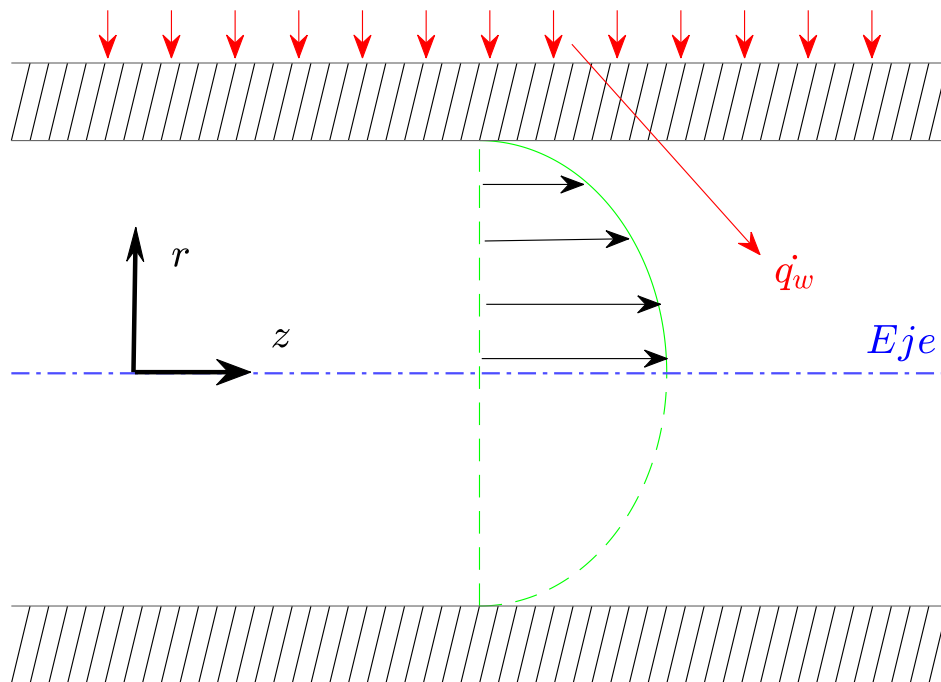


Figura 2.1: Esquema del problema

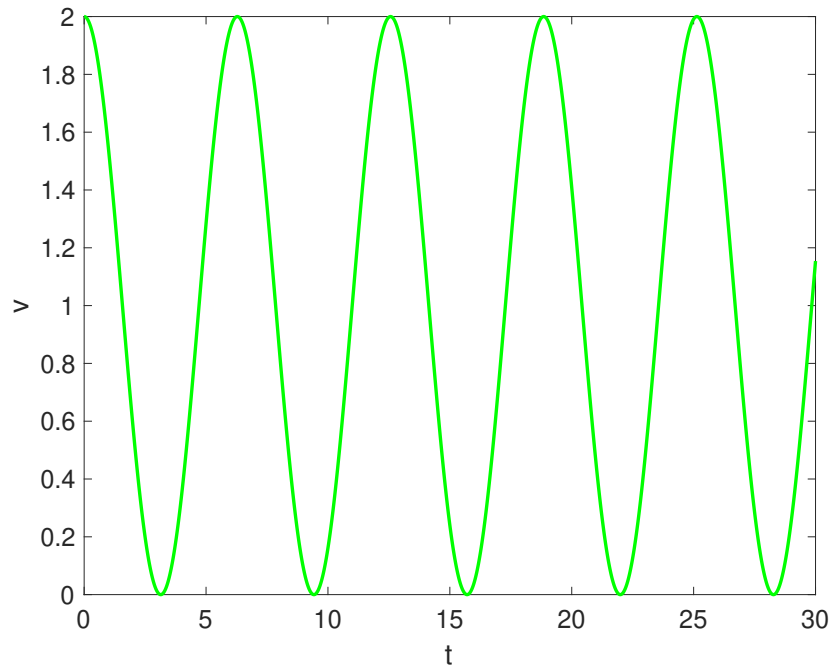


Figura 2.2: Ejemplo de variación senoidal de la velocidad de nuestras simulaciones

2.1.1. Ecuaciones a resolver

La dinámica del problema está gobernada por la ecuación de Navier-Stokes, junto con la ecuación de continuidad para un flujo incompresible y una ecuación de advección-difusión para la evolución temporal de la temperatura. Para simplificar la formulación del problema, descomponemos la temperatura de la siguiente forma, $T(r, z, t) = Tb(z) + T'(r, t)$, donde $Tb(z)$ es la temperatura promediada en la dirección radial y contiene el incremento lineal de temperatura debido al flujo de calor impuesto, mientras que T' es la fluctuación de la temperatura con respecto a dicho valor promedio. Al introducir este cambio de variable, puesto que en la ecuación de la temperatura no aparece la temperatura como tal, sino su gradiente, Tb desaparece de la formulación del problema (se convierte en un valor constante) y la ecuación queda únicamente en función de T' . Al eliminar el incremento de la temperatura en la dirección axial, todas las variables del problema dependen únicamente de la coordenada radial, simplificando notablemente la implementación de técnicas numéricas para resolver el problema. Para más detalles sobre esta metodología el lector puede consultar la referencia [15]. Como es habitual en estudios sobre transferencia calor en tuberías, no emplearemos directamente la temperatura T' en nuestras simulaciones, sino la temperatura reducida $\Phi^* = \langle T_w \rangle - T'(r)$, lo cual simplifica el cálculo del número de Nusselt. Usando la velocidad media v_{bulk} como velocidad característica, el radio de la tubería como longitud característica y $T^* = \frac{q_w}{\rho cp v_{bulk} R}$ [16], como temperatura característica, donde cp es el coeficiente de calor específico del fluido, las ecuaciones del problema en forma adimensional se escriben:

$$\nabla \cdot \vec{v} = 0 \quad (2.1)$$

$$\partial_t \vec{v} + \vec{v} \cdot \nabla \vec{v} = -\nabla P + \frac{1}{Re} \nabla^2 \vec{v} + F \vec{e}_z \quad (2.2)$$

$$\partial_t \Phi + \vec{v} \cdot \nabla \Phi + 4v_z = \frac{1}{Pe} \nabla^2 \Phi \quad (2.3)$$

Se recuerda que el vector velocidad sólo tiene componente axial en el caso de flujo laminar.

$$\vec{v} = v_z(r, t) \vec{e}_z \quad (2.4)$$

2.1.2. Parámetros adimensionales

A continuación, se mostrarán los parámetros adimensionales utilizados.

2.5 muestra la temperatura adimensional reducida, siendo T_w la temperatura de la pared, T la temperatura en el punto r y $T^* = \frac{q_w}{\rho_{cp} v_{bulk} R}$ [16].

$$\Phi = \frac{\langle T_w \rangle - T(r)}{T^*} \quad (2.5)$$

El número de *Peclet* (Pe) 2.6 proporciona información sobre la dominancia relativa del transporte convectivo sobre la difusión. En nuestro caso, este número se calcula considerando dos adimensionales adicionales: el número de *Prandtl* (Pr) 2.7 y el número de *Reynolds* (Re) 2.8.

$$Pe = Re \cdot Pr = \frac{\text{transporte molecular}}{\text{difusión viscosa}} \quad (2.6)$$

El número de *Prandtl* (Pr) 2.7 es crucial para comprender la transferencia de calor dentro del fluido, incluyendo la formación y el comportamiento de capas límite térmicas. Tal como muestra la expresión 2.7, dicho número adimensional cuantifica la relación entre la difusividad molecular y térmica del fluido. Fluidos típicos como el aire o el agua están caracterizados por números de Prandtl de orden unidad aproximadamente ($Pr \sim 0,7$ para el aire y $Pr \sim 3 - 7$ para el agua dependiendo de la temperatura). Sin embargo, para metales líquidos, como por ejemplo el mercurio, el número de Prandtl es bastante bajo ($Pr \sim 0,01$), y los aceites muy viscosos pueden tener números de Prandtl muy elevados ($Pr \sim 50 - 100$). Para realizar un estudio lo más completo posible, en nuestras simulaciones hemos considerado varios números de Prandtl, abarcando tres ordenes de magnitud de dicho parámetro (desde $Pr = 0,01$ a $Pr = 7$).

$$Pr = \frac{\nu}{\alpha} = \frac{\rho c_p \nu}{\kappa} = \frac{\text{Difusividad molecular}}{\text{Difusividad térmica}} \quad (2.7)$$

Por su parte, el número de *Reynolds* 2.8 desempeña un papel fundamental en la mecánica de fluidos, puesto que en muchos casos permite predecir si el fluido se encuentra en estado laminar o turbulento. En tuberías, la transición a la turbulencia ocurre para números de Reynolds de aproximadamente 2300. Como nuestro estudio se enfoca en flujo laminar, el número de Reynolds promedio de nuestras simulaciones es 1000 y la amplitud máxima es la unidad, de modo que el número de Reynolds máximo que se puede alcanzar en las simulaciones es 2000 (por debajo del umbral de la turbulencia). [4][5].

$$Re = \frac{v_{bulk} 2R}{\nu} = \frac{F.Inercia}{F.viscosa} \quad (2.8)$$

Wo es la frecuencia adimensional denominada número de Womersley (Wo). [13]

$$Wo = \frac{w \cdot r^2}{\nu} \quad (2.9)$$

Donde w es la frecuencia de la onda. El número de Womersley es de suma importancia para el proyecto, nos ha permitido variar la frecuencia de la velocidad media y por consiguiente las otras ondas producto de esta, por ello, hemos mapeado un rango muy amplio de Wo donde tenemos 14 frecuencias diferentes desde número de $Wo = 1$ hasta $Wo = 20$.

Por último, la amplitud de la onda ha sido adimensionalizada mediante el parámetro amplitud (Ω). La amplitud de la onda ha sido estudiada de 0, 1 hasta 1.

2.1.3. Condiciones iniciales, de contorno e imposición del flujo pulsante senoidal

Las ecuaciones presentadas en la sección 2.1.1 necesitan ser suplementadas con unas condiciones de contorno y una condición inicial. Como las ecuaciones del problema dependen únicamente de la coordenada radial, sólo necesitamos condiciones de contorno en esta dirección. En la pared, $r = R$, aplicamos la condición de no deslizamiento del fluido (velocidad nula) y una condición de tipo *Neumann* para la temperatura (el flujo de calor impuesto), mientras que en el eje, $r = 0$, aplicamos una condición de simetría par tanto para la velocidad como para la temperatura. Estas condiciones se expresan matemáticamente como:

$$\partial r|_{r=0} = 0 \quad \partial r|_{r=R} = -Pe \quad (2.10)$$

$$\frac{\partial v_z}{\partial r}|_{r=0} = 0 \quad v_z(r = R) = 0 \quad (2.11)$$

Volviendo a la ecuación 2.2, el término $F\vec{e}_z$ esta definido para forzar el movimiento senoidal en la tubería. Para llevar a cabo esta tarea, en cada instante de tiempo de la simulación, calculamos el valor de la velocidad media correspondiente a la onda senoidal empleada en la simulación 2.13 e imponemos dicha velocidad media mediante la ecuación 2.12. Esta ecuación se resuelve al mismo tiempo que las ecuaciones presentadas en la sección 2.1.1, de modo que el perfil de velocidad instantánea resultante satisface las condiciones de contorno, la condición de divergencia nula y tiene una velocidad media consistente con la onda senoidal impuesta.

$$2 \int_0^r v(r)rdr = v_{bulk}(t) \quad (2.12)$$

$$v_{bulk}(t) = 1 + \Omega \cos \frac{Wo^2}{Re \cdot t} \quad (2.13)$$

2.2. Software de Simulación

2.2.1. NsPipe_flow

Las ecuaciones descritas en la sección anterior se resuelven con la ayuda del código numérico altamente optimizado NSPipeFlow [17]. Dicho código implementa métodos numéricos avanzados, así como una estrategia de paralelización optimizada, permitiendo obtener soluciones precisas y detalladas del flujo de una manera muy eficiente. Es extensible y modular, lo que permite la introducción de nuevas funcionalidades y modelos. A continuación, se describen algunas de las características principales del software.

Arquitectura

La arquitectura de este sistema permite su utilización en diversas plataformas de alto rendimiento basadas en CPU, tales como **IBM Power**, **BlueGene x86₆₄**, **Xeon Phi**, **AMD EPYC** y **ARMv8.1**. Para generar el archivo ejecutable, se requiere un compilador moderno de Fortran, además de ciertas bibliotecas de código abierto tales como **MPI**, **BLAS/LAPACK**, **FFTW** y **HDF5**.

Dicho código es compatible tanto en computadoras portátiles como en sistemas de alto rendimiento de gran escala. Gracias a la paralelización a través de **MPI-OpenMP**, los cálculos se llevan a cabo de manera independiente en cada nodo. Adicionalmente, para los cálculos de términos no lineales, se requiere la biblioteca **MPI-Alltoall**.

Discretización espacial y temporal

Como en nuestro problema sólo tenemos dependencia radial, únicamente necesitamos discretizar esta coordenada. Para ello empleamos diferencias finitas centradas de alto orden, sobre un mallado de Gauss-Lobatto-Chebyshev, dónde la mayor concentración de puntos se encuentra cerca de la pared de la tubería. También es necesario que los puntos cercanos al origen de coordenadas ($r = 0$) estén suficientemente lejos de éste, debido a la singularidad de las coordenadas cilíndricas en el origen. Para la integración temporal, el software utiliza un método predictor-corrector, ajustando automáticamente el tamaño de paso temporal dentro de los parámetros establecidos por el usuario.

La condición CFL garantiza que el avance temporal sea lo suficientemente pequeño para capturar adecuadamente los cambios locales en el sistema y evitar la propagación de información más rápida de lo que la discretización espacial puede capturar 2.14.

$$\Delta t = C \min(\nabla/|\nu|) \quad (2.14)$$

donde ∇ en nuestro caso solo tendrá término en r siendo anulados en las demás direcciones. Para más información, consulte [18].

2.3. Lanzamiento a Picasso de las simulaciones

Las simulaciones realizadas para la resolución del problema son de gran coste computacional, es por esto que se han realizado en el superordenador de la Universidad de Málaga. El superordenador Picasso pertenece a la **RES** (Red Española de Supercomputación). Esta red está formada por 14 nodos de superordenadores repartidos a lo largo del territorio español con la finalidad de otorgar a los investigadores una cantidad considerable de recursos para cálculo matemáticos. En concreto, el superordenador Picasso fue fundado en 1997 funcionando hasta día de hoy con sus correspondientes actualizaciones de hardware y software. Actualmente cuenta 30616 núcleos de computación, 156TB de memoria RAM, 32xA100 GPUS con 13K cores tensores, 110K FP32 cores y 1280 GB de RAM, además, 960TB de memoria de almacenamiento entre otras características. [19],[20]

2.3.1. Scripts y ficheros de trabajo

En esta sección se describen los scripts y ficheros que es necesario modificar para llevar a cabo las simulaciones.

`input_nsPipeFlow`

[18] Este fichero contiene los valores de todos los parámetros de entrada con los que se ejecuta la simulación. Al comenzar la simulación, el primer paso consiste en leer la información de este fichero, por lo que es esencial introducir toda la información de forma adecuada, respetando la sintaxis explicada en los comentarios. En caso contrario, la simulación no se iniciará. Cabe indicar que este fichero ha sido modificado con respecto a versiones anteriores para incluir los parámetros asociados con un flujo pulsante senoidal A.1.1:

- Línea 1 – 7. Parámetros de mallado. Elección de los nodos para el mallado.
- Línea 9 – 16. Parámetros físicos. Los parámetros más utilizados han sido el número de Prandtl(Pr), la número de Womersley(Wo) y la Amplitud (Ω).
- Línea 18 – 25. Parámetros de tiempo. Donde hemos variado los datos tanto de pasos de tiempo (`numsteps`) como el tiempo inicial (`init_dt`) y el paso de tiempo máximo (`maxdt`). Además hay que tener en cuenta que tenemos un Δt variable, por lo tanto, el mismo código se ajusta entre los parámetros impuestos para obtener el mejor paso de tiempo posible.
- Línea 27 – 35. Parámetros de salida. Aquí podemos elegir la frecuencia de archivos de retorno.
- Línea 37 – 40. Parámetros de control. Tenemos la libertad de seleccionar un reinicio de simulación. Además, de elegir el tiempo que durará la simulación.
- Línea 42 – 48. Parámetros de perturbaciones. Es posible seleccionar los ejes donde comenzarán las primeras perturbaciones; en el caso de este trabajo no se ha realizado ninguna modificación sobre lo mostrado en el Anexo A.1.1.



lanza.sh

Se tratan de las instrucciones que necesita Picasso para realizar la simulación A.1.2. En este código podemos elegir desde el nombre de la simulación hasta como denominar los archivos de salida necesarios. Se comentarán los más importantes:

- Línea 2. Elección del nombre del trabajo.
- Línea 3. Tiempo de simulación.
- Línea 4. Número de tareas.
- Línea 5. Tareas por nodo. (Debe ser un número divisible del número de la Línea 4).
- Línea 8. Número de cpus por tarea.
- Línea 11. La cantidad de memoria RAM que necesitamos para la simulación.
- Línea 13-16. Llamamiento a los compiladores como a las librerías necesarias.
- Línea 17. Llamamiento a la paralelización. Donde el número que se muestra en el ejemplo siempre debe de coincidir con el número de tareas a realizar. En nuestro caso hemos tomado solo 2 cpus para la simulación.

Archivos de salida de las simulaciones

Tras la finalización correcta de las simulaciones, el código aporta una serie de datos. Cabe remarcar que la generación de dichos datos y su creación están impuestas según nuestros criterio. A continuación, se explicarán estos datos:

- **coeff_DNS1.*** Archivo binario que contiene los valores de las distintas variables del problema (velocidad, presión y temperatura reducida) en un instante de tiempo determinado. Dichos archivos nos permiten continuar la simulación en dichos puntos donde se han guardado.
- **Nusselt.** Matriz 2×2 , donde la primera columna es el tiempo y la segunda es el número de Nusselt.
- **friction.** Matriz 5×5 , primera columna es el tiempo, segunda columna la velocidad media, tercera columna es la velocidad en $r = 0$, la cuarta columna es la velocidad de fricción y la quinta columna es el coeficiente de fricción.
- **meanprof*** Perfil de velocidad.
- **Temp_meanprof*.** Perfil de temperatura.
- **parameters.** Parámetros de la simulación.
- **ke_***. Energía cinética. En nuestro caso los dos ficheros son iguales.

Además, el archivo **lanza.sh** A.1.2 genera un archivo denominado **nsPipeRe1000.err** donde se muestra el error en el caso que la simulación sea fallida y un archivo **nsPipe-Re3000.out** que contiene los outputs de las operaciones realizadas por el código.



2.4. Metodología: Simulaciones y postprocesado

2.4.1. Parámetros de control

En todas las simulaciones realizadas el número de Reynolds medio se ha fijado en

$$Re = 1000 \quad (2.15)$$

Los valores del número de Prandtl utilizados son los siguientes:

$$Pr = \begin{pmatrix} 0,01 \\ 0,1 \\ 0,7 \\ 7 \end{pmatrix} \quad (2.16)$$

Estos valores del número de Prandtl se han tomado en decisión de simular diferentes tipos de fluidos desde metales líquidos hasta agua, pasando por aire.

Los valores del número de Womersley son:

$$Wo = \begin{pmatrix} 1 \\ 1,5 \\ 2 \\ 2,5 \\ 3 \\ 4 \\ 5 \\ 6,5 \\ 8 \\ 10 \\ 12,5 \\ 15 \\ 17,5 \\ 20 \end{pmatrix} \quad (2.17)$$

Los valores a tomar de la amplitud van desde 0,1 hasta 1 con un espaciado de 0,1 entre amplitudes.

2.4.2. Parámetros adicionales

Además, de los parámetros adimensionales explicados anteriormente se han creado varios parámetros para cuantificar los resultados.

Factor de impacto

$$\Lambda = \frac{Nu_{simulacion}}{Nu_{steady}} \quad (2.18)$$

Su función es comparar el número de Nusselt obtenido en nuestras simulaciones con flujo pulsante frente al número de Nusselt de un flujo constante no pulsante, cuyo valor es $Nu = 4,36$. Este término facilita la comparación de ambos flujos, permitiendo cuantificar de forma sencilla el efecto del flujo pulsante sobre la transferencia de calor. Valores de (Λ) mayores que la unidad corresponderán a un incremento de la transferencia de calor con respecto al caso estacionario, mientras que valores menores que la unidad indicarán pérdidas de transferencia de calor con respecto a dicho caso.

Desfase entre la velocidad media y número de Nusselt

$$\varphi = \frac{t_{mínimo_vel} - t_{mínimo_Nusselt}}{P} \quad (2.19)$$

Dependiendo de la frecuencia y amplitud de la onda senoidal, la respuesta de la temperatura (y por tanto el número de Nusselt) a los cambios en la velocidad media no tiene por qué ser inmediata, dando lugar a que se genere un desfase entre estas dos variables. Para cuantificar dicho desfase medimos la distancia entre mínimos de las ondas de velocidad media y número de Nusselt ($t_{mínimo_vel}$ y $t_{mínimo_Nusselt}$, respectivamente), y normalizamos dicha diferencia con el periodo de la onda de velocidad media (A.2.5).

2.4.3. Postprocesado de datos

Los códigos han sido implementados en MATLAB y programados por el alumno Federico Pérez González. En concreto para el postprocesado nos hemos centrado en los outputs **Nusselt**, **friction**, **meanprof***, **Temp_meanprof***. Se puede dividir en dos vertientes conectadas.

- Obtención de base de datos.
- Simulaciones de mayor precisión.

Obtención de base de datos

Los códigos utilizados para el tratamiento:

- **Postbueno.m**. Mostrado en A.2.1, está compuesto de dos funciones principales. La primera función consiste en encontrar los mínimos locales de la simulación y junto a una tolerancia obtener los datos deseados. Tras el primer tratamiento, calcula Λ y φ , guardando los resultados en un archivo denominado 'datos.txt'. Además, existen una serie de automatismos comentados que fueron útiles al principio del tratamiento, puesto que permitieron identificar las simulaciones erróneas o con ciertas anomalías.
- **generar_colormap.m**. A.2.2 Este código ha sido aportado por el tutor Jose Manuel López Alonso. Utilizando los datos obtenidos en el anterior código A.2.1, crea un mapa de colores donde se representa en el eje X el número de Womersley (Wo), en el eje Y las Amplitudes y el color varía según el valor del factor de impacto Λ . Esta escala de colores ha sido modificada para obtener una mejor visualización de los resultados.

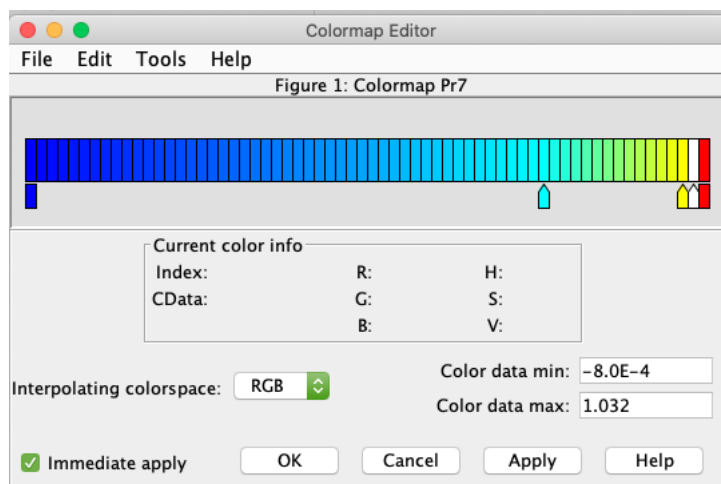


Figura 2.3: Escala de colores utilizada

Además existen varios códigos complementarios para ayudar al tratamiento de los datos, expuestos en A.2.

Simulaciones de mayor precisión

Para poder explicar algunos aspectos de los resultados obtenidos, hemos necesitado realizar simulaciones más precisas (reduciendo el paso de tiempo para poder capturar con mayor precisión la variación temporal de los perfiles de temperatura). Se han seleccionado valores que inducen pérdidas ($\Lambda < 1$), así como valores que inducen ganancias ($\Lambda > 1$), con el objetivo de investigar las estructuras que propician estos cambios en la transferencia de calor. Para ello, se han aumentado la frecuencia de archivos de retorno del Picasso. De esta forma, se ha conseguido un muestreo de datos mucho más precis. Los archivos con mayor frecuencia de retorno son: **Temp_meanprof***, **meanprof***, **Nusselt** y **friction**.

Hemos llegado a la conclusión que para $Wo = [1, 4]$ con un perfil cada 100 pasos de tiempo es suficiente para obtener perfiles precisos. Para $Wo = [5, 8]$ tenemos un perfil cada 10 pasos de tiempo y para $Wo = [10, 20]$ tenemos un perfil para cada paso de tiempo.

- **trat_perfil_optimizado.m**. Identifica las posiciones de dos mínimos locales contiguos en la velocidad media, con dichos mínimos obtiene los perfiles de velocidad y temperatura contenidos en un período de la onda P . Una vez identificados los perfiles de velocidad realiza una media de los perfiles y genera dos ficheros: "perfil_vel.txtz" "perfil_T.txtz" A.2.3.
- **obtencioncurvas.m** A.2.4 En este código toma los ficheros generados por el código anterior y lo dibujamos frente a los perfiles de una onda no pulsante para observar si existe alguna diferencia entre estos.
- **desfase.m**. A.2.5 Encuentra los mínimos locales de la velocidad media y el Nusselt, utilizando dos mínimos contiguos uno de la velocidad media y otro del Nusselt calcula φ .

CAPÍTULO 3

Resultados de las simulaciones

3.1. Análisis del factor de impacto

El factor de impacto (Λ), detallado en 2.18, nos proporciona una medida de las variaciones en el número de Nusselt (Nu) 1.1 en comparación con el caso no pulsante. Estas variaciones se clasifican en pérdidas ($\Lambda < 1$) o mejoras ($\Lambda > 1$) de la transferencia de calor. Para llevar a cabo este análisis, se realizaron 560 simulaciones, modificando en cada una de ellas los valores de los parámetros adimensionales de acuerdo a los valores expuestos en la sección 2.4.1.

Con el propósito de presentar los resultados obtenidos, muestran en la figura 3.1 los mapas de colores correspondientes al factor de impacto (Λ). Cada uno de estos mapas se genera individualmente para diferentes valores de Prandtl (Pr). En los cuatro paneles de esta figura, el eje X representa el número de Womersley (Wo), mientras que el eje Y refleja la amplitud (Ω). La escala de colores mostrada en 2.3 corresponde al factor de impacto (Λ).

Las zonas coloreadas en rojo indican mejoras en la transferencia de calor ($\Lambda > 1$), aunque son relativamente modestas, con un aumento máximo de tan solo el 3%. Las áreas de color blanco representan zonas en las que el número de Nusselt (Nu) no varía con respecto al caso no pulsante ($\Lambda = 1$). Las pérdidas, que pueden llegar a ser muy significativas (incluso del 100% en algunos casos), se representan en la escala de tonos amarillos y azules.

Tal y como se aprecia a simple vista en la figura, los mapas de color obtenidos para distintos valores del número de Prandtl son cualitativamente muy similares. En todos los casos, existe una pequeña región del espacio paramétrico $Wo - \Omega$ donde un flujo pulsante laminar incrementa ligeramente la transferencia de calor con respecto al caso no pulsante (mostrada en color rojo en la figura). Dicha región aparece para valores bajos de Wo (desde $Wo = 1$ hasta $Wo = 3$) y amplitudes elevadas (entre $\Omega = 0,6$ y $\Omega = 0,9$). Para valores del número de Prandtl menores que la unidad (3.1 (a), (b) y (c)), la región de incremento es prácticamente la misma en todos los casos, sin embargo, para números de Prandtl mayores que la unidad (3.1 (d)), el intervalo de valores del número Wo donde la transferencia de calor aumenta es algo menor (entre $Wo = 1$ y $Wo = 2$). Siguiendo con los casos correspondientes a frecuencias bajas ($Wo \leq 4$), se aprecia que para valores de $\Omega < 0,6$, el número de Nusselt no varía en comparación con el caso no pulsante.

A partir de $Wo = 5$ en adelante, el factor de impacto es menor que uno (zonas amarillas

y azules de los mapas de color) y, por lo tanto, el flujo pulsante da lugar a una reducción de la transferencia de calor con respecto a la que se obtiene en el caso estacionario. Dichas pérdidas de transferencia de calor se hacen más pronunciadas a medida que se incrementa la amplitud de la onda senoidal, alcanzando su máximo en $\Omega = 1$. De nuevo se aprecia una pequeña excepción a este comportamiento para el caso de $Pr = 7$ (3.1 (d)), donde no se encuentran variaciones en el número de Nusselt ($\Lambda = 1$) a bajas amplitudes ($\Omega = 0,1$ hasta $\Omega = 0,3$). En algunos casos de amplitud máxima ($\Omega = 1$), se obtienen pérdidas cercanas al 100 %, un fenómeno que aún carece de explicación.

Para mostrar en más detalle la dependencia cuantitativa del factor de impacto con el número de Prandtl, la figura 3.2 muestra como varía el factor de impacto (Λ) con la amplitud (Ω) cuando el número de Wo es constante ($Wo = 2, 5$ en el panel de la izquierda y $Wo = 8$ en el panel de la derecha). Estas figuras, representan en el eje X la amplitud (Ω), el eje Y muestra el factor de impacto (Λ) y cada línea en el gráfico corresponde a un número de Prandtl distinto. Se puede apreciar en estas figuras que el factor de impacto disminuye ligeramente a medida que el número de Prandtl incrementa. Esta dependencia, a pesar de ser muy pequeña, es una observación bastante interesante puesto que para un flujo estacionario el número de Nusselt en el caso laminar no depende del número de Prandtl. Nuestras simulaciones muestran que en el caso de flujo pulsante si que existe una ligera dependencia con este parámetro.

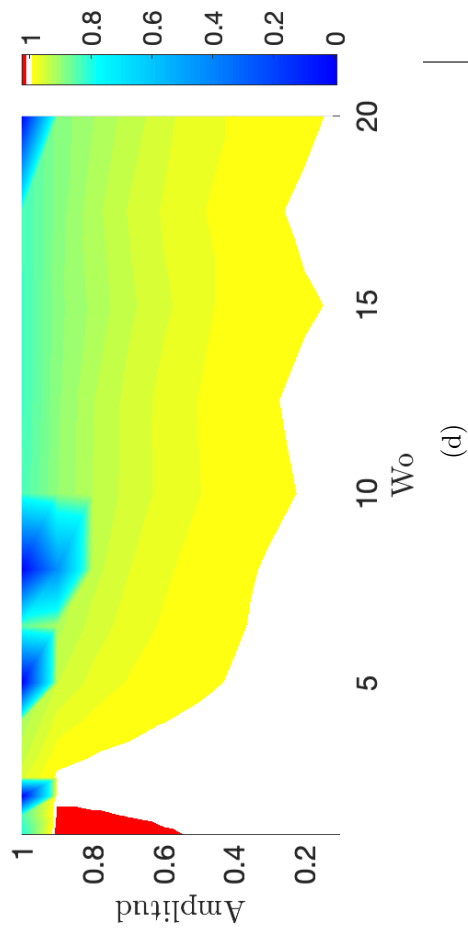
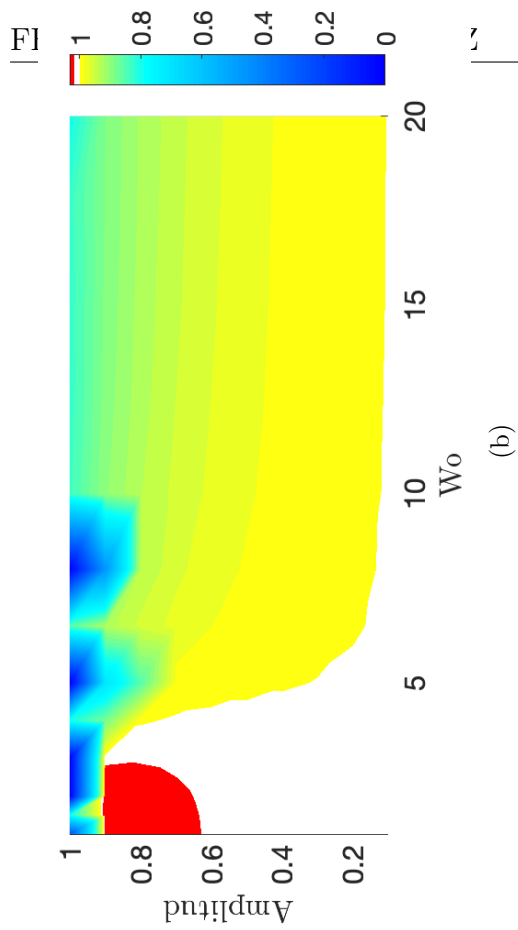
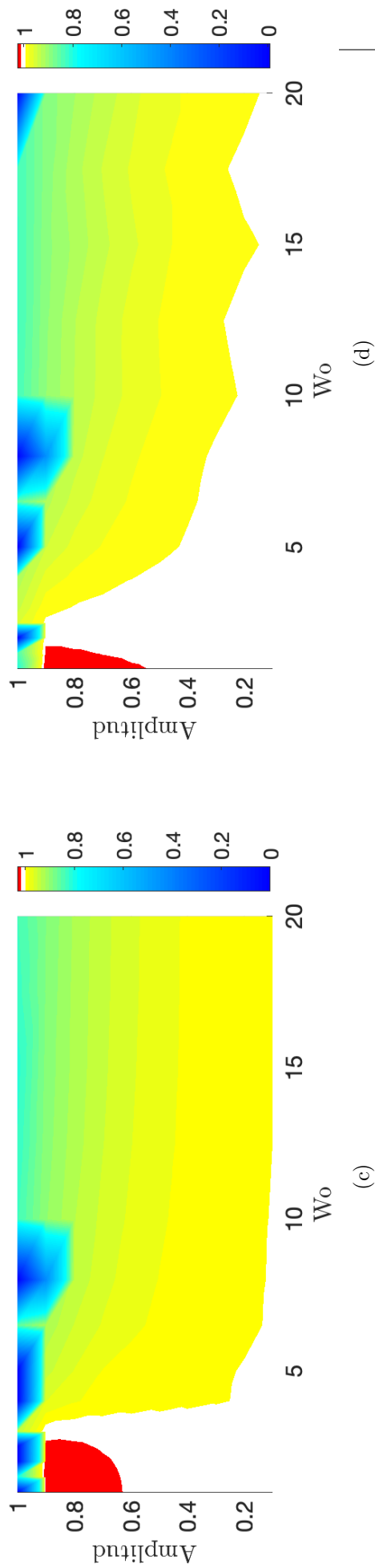
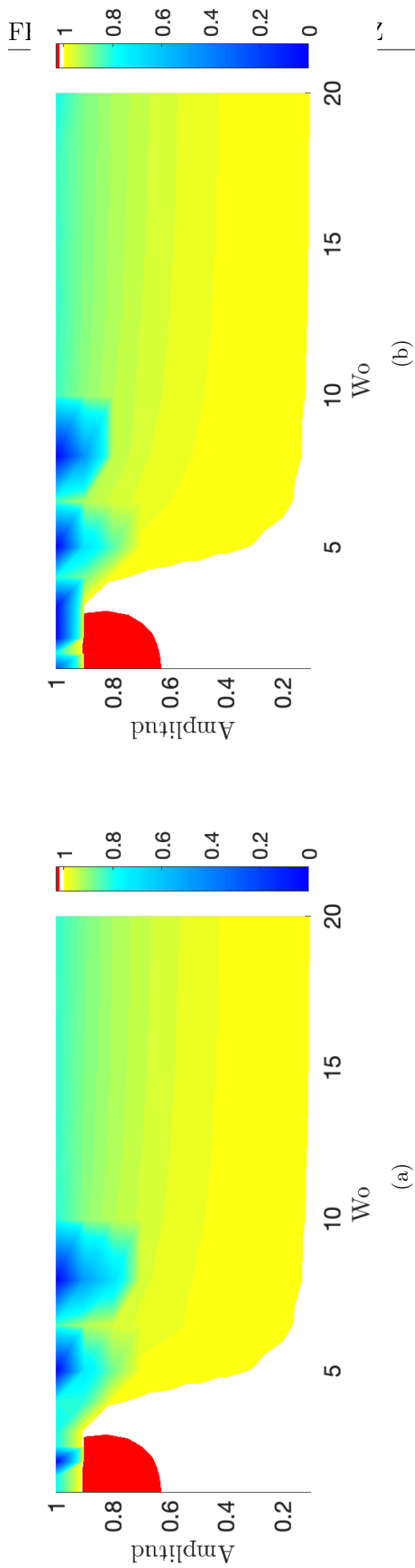
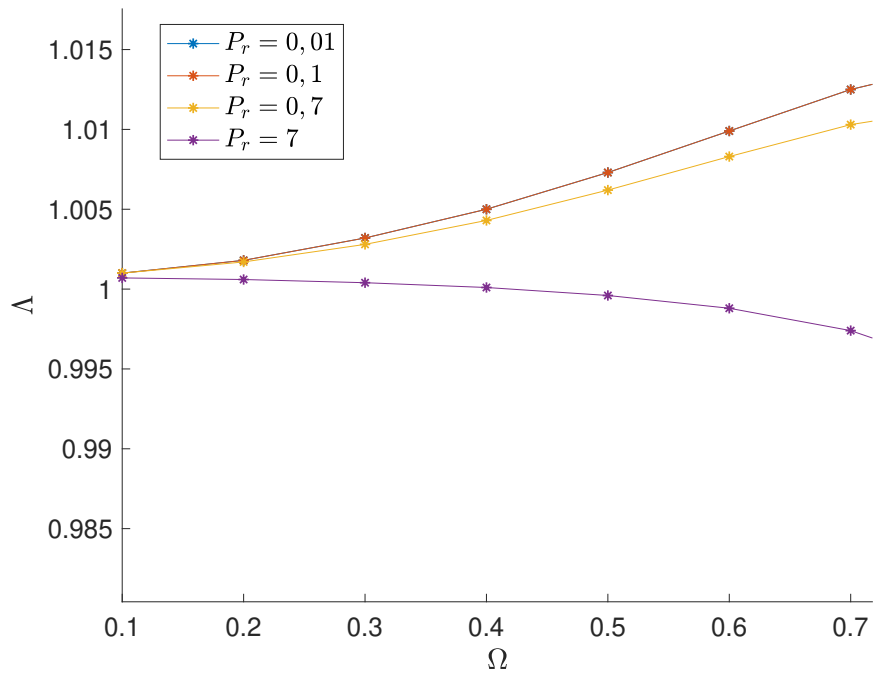
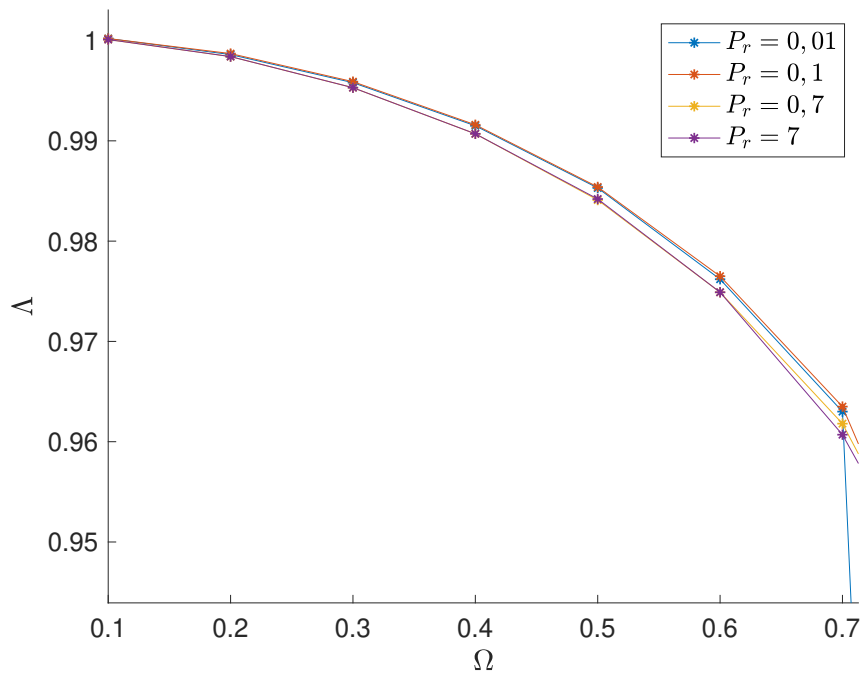


Figura 3.1: Mapas de color representando la variación del factor de impacto (Λ) con la amplitud (Ω) y frecuencia adimensional (Wo) de la onda senoidal para varios valores del número de Prandtl: (a) $Pr = 0, 01$, (b) $Pr = 0, 1$, $Pr = 0, 7$ y (d) $Pr = 7$



(a)



(b)

Figura 3.2: Factor de impacto (Λ) frente amplitud (Ω): (a) $Wo = 2,5$ y (b) $Wo = 8$

3.2. Análisis de los perfiles de velocidad

Con el fin de explicar las tendencias expuestas en la sección anterior 3.1, se ha realizado un análisis de los perfiles de velocidad en casos de mejora o pérdidas de la transferencia de calor. Los perfiles mostrados son medias de todos los perfiles obtenidos en un ciclo completo (desde mínimo a mínimo). Para más detalle sobre la obtención de los perfiles vaya a la sección 2.4.3.

Todas las gráficas mostradas en este capítulo, representan en el eje X el radio adimensional (r) y en el eje Y la velocidad adimensional (v). El perfil promedio obtenido en nuestras simulaciones está representado en color azul mientras que el perfil del caso no pulsante se muestra en color naranja.

La figura 3.3 presenta un perfil de velocidad obtenido para una combinación de amplitud y frecuencia ($\Omega = 5$ y $Wo = 0,2$) en el que no hay diferencias en la transferencia de calor ($\Lambda = 1$). Cuando esto ocurre, se observa que el perfil de velocidad promedio obtenido en el caso pulsante coincide con el perfil de velocidad del caso no pulsante. Esta gráfica exhibe uno de los tres comportamientos distintos de los perfiles de velocidad.

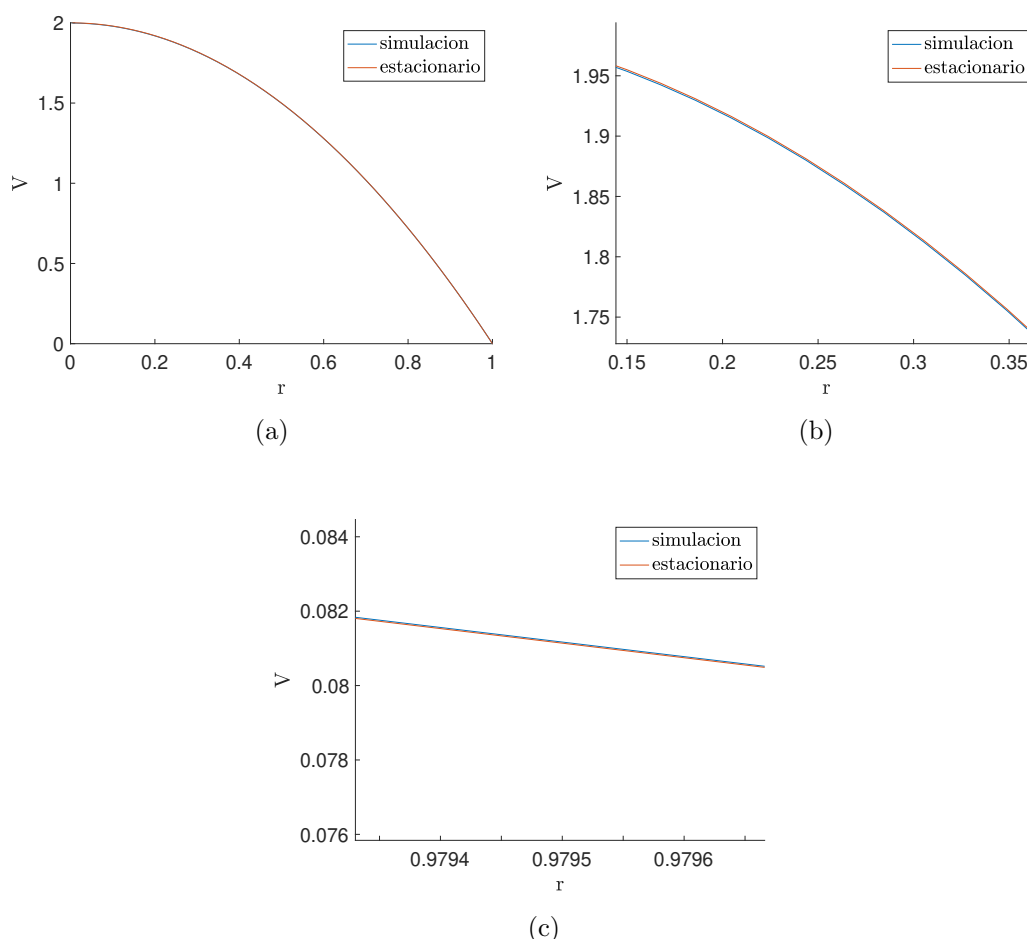


Figura 3.3: (a) Perfil de velocidad, (b) zoom cerca de la pared y (c) zoom cerca del centro

En la figura 3.4, se presentan dos perfiles de velocidad: en la primera columna, caso dónde se obtiene un incremento en la transferencia de calor ($Pr = 0,01$, $Wo = 2$ y $\Omega = 0,9$), y en la segunda, un caso dónde se obtienen pérdidas ($Pr = 0,01$, $Wo = 8$ y $\Omega = 0,8$). En

ambos casos los perfiles de velocidad obtenidos se comparan con el perfil de velocidad no pulsante. En la primera fila de la figura se muestran los perfiles de velocidad, la segunda fila muestra un zoom en la pared de la tubería ($r = 0$) y la tercera fila muestra un zoom en el centro de la tubería ($r = 1$), consideradas zonas de interés para el análisis.

En el análisis de la primera columna, centrandó nuestra atención cerca de la pared (subfigura (e)), se observa que el perfil obtenido se encuentra retrasado en comparación con el perfil no pulsante. Sin embargo, al examinar el comportamiento en el centro de la tubería (subfigura (c)), el perfil obtenido adelanta al perfil no pulsante.

En la segunda columna de la figura, se presentan patrones similares pero a la inversa. Cerca de la pared, (subfigura (f)), se observa que el perfil obtenido adelanta al perfil no pulsante. En contraste, en el centro de la tubería, (subfigura (d)), el perfil obtenido se retrasa en comparación con el perfil no pulsante.

El comportamiento ejemplificado por los perfiles de velocidad promedio mostrados en estas figuras se repite en todos los casos donde se produce un incremento o decremento de la transferencia de calor con respecto al caso estacionario. Este resultado es especialmente importante, puesto que permite predecir de una manera simple si una cierta combinación de frecuencia y amplitud dará lugar a pérdidas o ganancia de transferencia de calor con respecto al caso estacionario. Simplemente hay que calcular el perfil de velocidad promedio (existe una expresión analítica para el perfil de velocidad laminar de un flujo pulsante ([13]) y mirar si la velocidad cerca de la pared es mayor o menor que en el perfil de Hagen-Poiseuille. Si es mayor, el flujo pulsante dará lugar a una mayor transferencia de calor, mientras que al contrario, habrá pérdidas.

Después de haber identificado estos patrones en los perfiles según sus factores de impacto, el siguiente paso es profundizar en la explicación de estas diferencias. Para llevar a cabo esta tarea, se lleva a cabo un análisis detallado de los perfiles de velocidad sin la aplicación de medias, es decir, perfiles de velocidad instantáneos.

En primer lugar, se procederá a analizar los perfiles de velocidad asociados a mejoras en la transferencia de calor ($\Lambda > 1$). Para este propósito en la figura 3.5, se representan perfiles de velocidad instantáneos correspondientes a la primera columna de las figura en 3.4. En la región cercana a la pared, se observa que los perfiles instantáneos presentan en muchos instantes de tiempo velocidades mucho más altas que la velocidad del perfil de velocidad no pulsante. Cuando se realiza el promedio, esta tendencia ocasiona que el perfil de velocidad promediado adelante al perfil de velocidad no pulsante. Mientras tanto, en la región central el perfil de velocidad instantáneos se retrasa en comparación con el perfil de velocidad no pulsante, como se visualiza en la gráfica 3.4 (e).

A continuación, procederemos con un análisis de los perfiles de velocidad instantáneos en situaciones que resultan en pérdidas en la transferencia de calor. La Figura 3.6 muestra los perfiles de velocidad instantáneos que se emplearon para generar el perfil de la segunda columna en las gráficas 3.4 (b), (d) y (f).

En situaciones de pérdida, se produce un proceso contrario al descrito previamente. Las pendientes de los perfiles, al acercarse a $r = 1$, se hacen en muchos casos positivas, indicando la existencia de "backflow" en dicha zona. Esto resulta en el retraso del perfil de velocidad promediado en la pared de la tubería, tal como se observa en la gráfica 3.4 (d). Evidentemente, por conservación de la masa, es necesario que los perfiles cerca de $r = 0$, como se evidencia en la misma gráfica 3.4 (f).



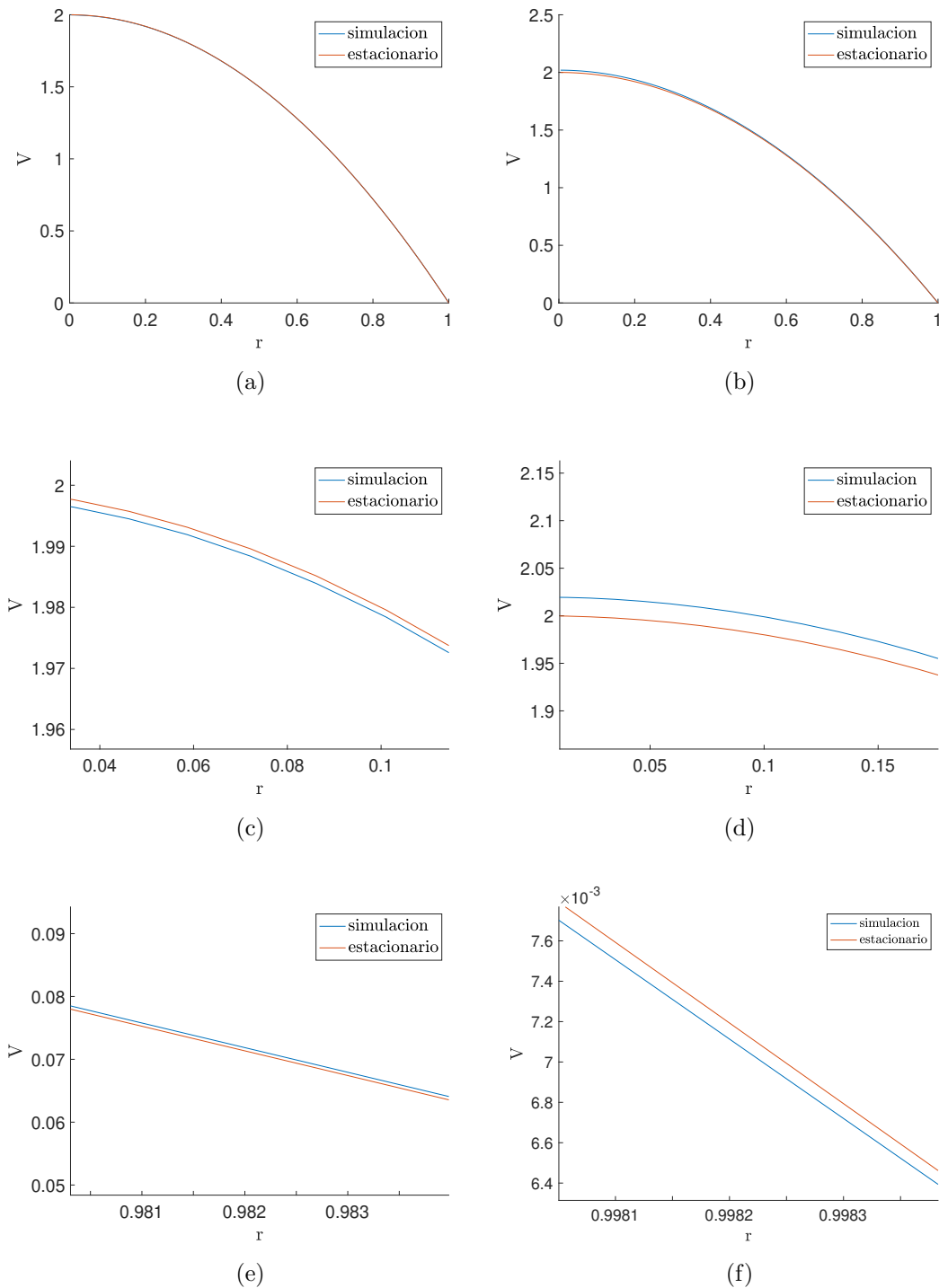


Figura 3.4: (a) perfil de velocidad $Pr = 0,01; Wo = 2; \Omega = 0,9; \Lambda = 1,028$, (b) perfil de velocidad $Pr = 0,01; Wo = 8; \Omega = 0,8; \Lambda = 0,6926$, (c) zoom de los perfiles mostrados en (a) en el centro de la tubería. Se aprecia como la velocidad en el caso estacionario es ligeramente mayor que la velocidad en el caso pulsante. (d) zoom de los perfiles mostrados en (b) en el centro de la tubería. Se observa que el caso estacionario es ligeramente menor que el caso promediado (e) zoom de los perfiles mostrados en (b) en la pared de la tubería. Se aprecia como el perfil promediado es mayor al perfil estacionario. (f) zoom del perfil con pérdidas en la pared de la tubería. Se observa como el perfil estacionario es mayor que el perfil promedio



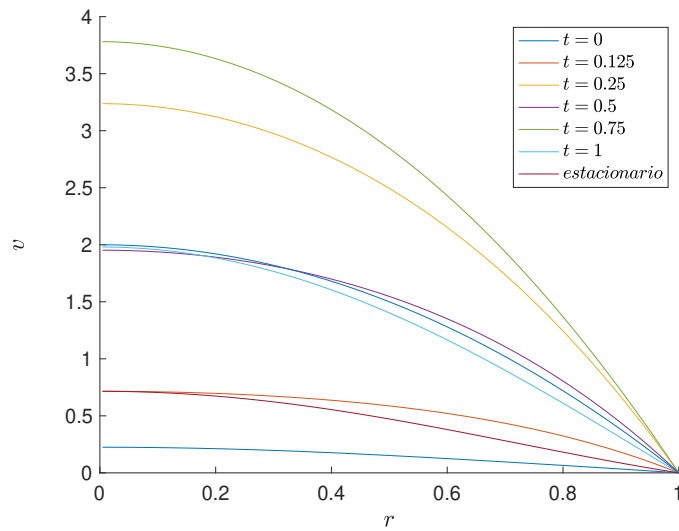


Figura 3.5: Perfiles no promediados para $Pr = 0,01; Wo = 2; \Omega = 0,9; \Lambda = 1,028$. El tiempo se encuentra normalizado mediante el periodo

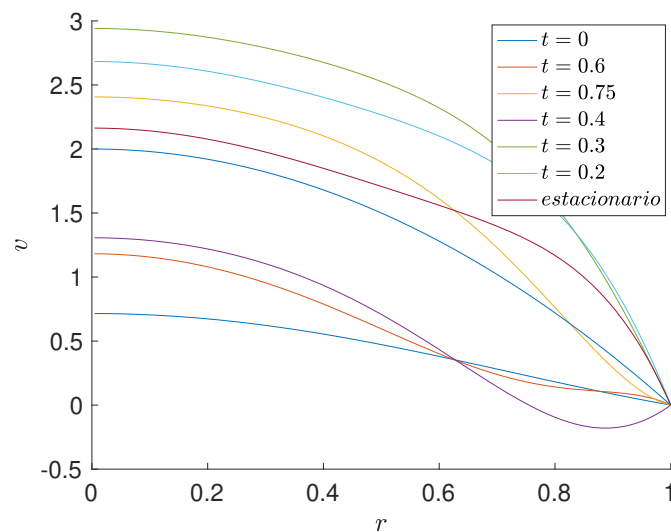


Figura 3.6: Perfiles no promediados para $Pr = 0,01; Wo = 8; \Omega = 0,8; \Lambda = 0,6926$. El tiempo se encuentra normalizado mediante el periodo

3.2.1. Desfase de las ondas

Durante la ejecución de las simulaciones, hemos observado la presencia de un desfase (φ) entre la velocidad media y el número de Nusselt (Nu). Nuestro objetivo ha sido investigar cuándo y cómo este desfase se manifiesta en función del factor de impacto. Utilizando un procedimiento similar al que se describió en la sección anterior, identificaremos los rangos en los cuales se producen mejoras o pérdidas en la transferencia de calor debido a este desfase.

La figura 3.7 muestra los parámetros utilizados para cuantificar este desfase. Entre las líneas verticales discontinuas, se presenta el periodo de la velocidad media utilizado para calcular el desfase, mientras que las líneas verticales continuas muestran el desfase entre los mínimos de la velocidad media y del número de Nusselt (Nu). Siguiendo la ecuación A.2.5, cuando los valores del desfase $\varphi < 0^\circ$ son negativos, el número de Nusselt (Nu) se encuentra adelantado en relación a la velocidad media. En el caso de valores positivos ($\varphi > 0^\circ$), la velocidad media se adelanta con respecto al número de Nusselt (Nu).

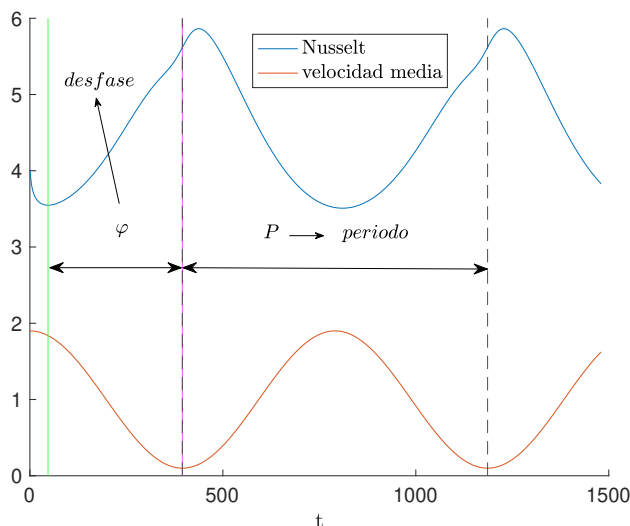


Figura 3.7: Desfase entre las onda de velocidad media y el número de Nusselt.

La figura 3.8 muestra una nube de puntos que abarca todos los valores obtenidos en las simulaciones (un total de 560 valores). Los puntos de color rojo muestran mejoras de la transferencia de calor ($\Lambda > 1$), mientras que los puntos de color azul muestran pérdidas ($\Lambda < 1$) en la transferencia de calor. Las mejoras de la transferencia de calor se encuentran entre los rangos $\varphi = \pm[70, 180]^\circ$, sin independencia de cual onda está adelantada. Las pérdidas de la transferencia de calor aparecen principalmente cuando el desfase entre ambas ondas es pequeño $\varphi = [4, 60]^\circ$,

Existen, sin embargo, algunos casos aislados dónde se han obtenido pérdidas a pesar de tener grandes desfases. Entre ellos, cabe destacar la obtención de unas pérdidas de aproximadamente 20% con respecto al caso estacionario, cuando el número de Nusselt adelanta a la velocidad media ($\varphi < 0$) y las ondas están aproximadamente en antifase ($|\varphi| \sim 180^\circ$). Por último, se observa que cuando se obtienen pérdidas del 100%, el desfase es nulo ($\varphi = 0^\circ$) entre los mínimos de la velocidad media y el número de Nusselt (Nu).

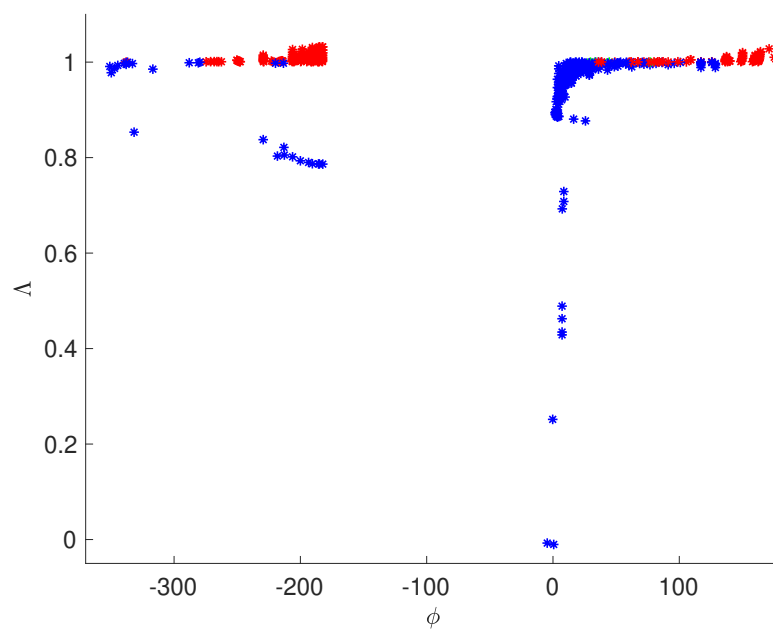


Figura 3.8: Desfase frente a factor de impacto

CAPÍTULO 4

Conclusiones y discusión de los resultados

Después de concluir esta investigación, se han establecido las bases para futuros estudios en esta área. A continuación, se resumen los resultados más relevantes obtenidos:

- Este estudio ha revelado que la transferencia de calor asociada a un flujo pulsante laminar en una tubería se ve afectada de manera significativa. Se ha alcanzado una mejora máxima del 3% en el rango de parámetros $Wo = [0,1, 0,3]$ y $\Omega = [0,6, 0,9]$. Sin embargo, al mantener constante el valor de Wo y variar la amplitud (Ω) en el intervalo $[0,1, 0,6)$, no se ha observado ninguna mejora en la transferencia de calor. Por otro lado, para valores $Wo \geq 5$, se ha identificado un aumento en las pérdidas a medida que la amplitud (Ω) crece. Este resultado concuerda con las observaciones experimentales de William y Barker [8]. Al igual que en nuestras simulaciones, estos experimentos encontraron pérdidas significativas a altas frecuencias, no se encontraron mejoras en la transferencia de calor. Esta discrepancia entre los resultados podría atribuirse a la limitada precisión de los instrumentos experimentales (presentan un error mínimo del 10%). Esta falta de precisión pudo haber imposibilitado la detección de las mejoras máximas del 3%, lo cual respalda y valida nuestros resultados obtenidos en simulaciones numéricas.
- Los perfiles de velocidad promediados muestran diferentes variaciones según el comportamiento del factor de impacto (Λ). Se ha identificado que para obtener mejoras ($\Lambda > 1$), cerca de la pared, el perfil de velocidad promediado se retrasa con respecto al perfil de velocidad no pulsante, mientras que en el centro de la tubería, ocurre lo contrario, es decir, el perfil de velocidad promediado adelanta al perfil de velocidad no pulsante. Por otro lado, cuando se obtienen pérdidas ($\Lambda < 1$), se observa el caso contrario. Cerca de la pared, el perfil de velocidad promediado adelanta al perfil de velocidad no pulsante, mientras que cerca del centro de la tubería se retrasa en comparación con el perfil de velocidad no pulsante. Por último, cuando no se obtienen ni ganancias ni pérdidas ($\Lambda = 1$), los perfiles de velocidad promediados y los perfiles de velocidad no pulsante no muestran diferencias significativas entre sí.

Estas variaciones en el perfil de velocidad proporcionarán la capacidad de realizar un pre-análisis antes de llevar a cabo las simulaciones. Esto permitirá evaluar la tendencia del factor de impacto (Λ) en términos de ganancias o pérdidas en la transferencia de calor. Sin embargo, es importante destacar que esta herramienta no posibilitará una cuantificación precisa del factor de impacto (Λ).

La presencia de tres tipos de perfiles de velocidad refuta la suposición realizada en [13], que asumía que el perfil de velocidad no variaría con respecto al obtenido por Hagen-Poiseuille.

- El desfase (φ) entre la velocidad media y el número de Nusselt (Nu) presenta una correlación con el factor de impacto (Λ). Si se obtienen ganancias, el desfase se encuentra entre los rangos $\varphi = \pm[70, 200]$. Mientras que si se obtienen pérdidas, el rango se encuentra entre $\varphi = [4, 60]$.
- En el caso particular de $Pr = 7$, los rangos de mejora y pérdida varían en comparación a los demás casos estudiados. Ilógico debido a que el número de Prandtl (Pr) se encuentra desacoplado de las ecuaciones resueltas y no debería de existir dependencia de este en la transferencia de calor. Este caso en particular debe de ser estudiado en detenimiento en futuros trabajos.

Como último propondré futuros trabajos tras la finalización de mi TFG:

- Estudio concreto de la transferencia de calor para $Pr = 7$.
- Estudio sobre otras formas de onda que magnifiquen los procesos de mejora en la transferencia de calor.
- Obtención de una base de datos para flujo turbulento.
- Demostración y aceptación con investigaciones experimentales.

Bibliografía

1. E. E. Agency, Healthy environment, healthy lives: how the environment influences health and well-being in Europe. *Publications Office of the European Union* (2020).
2. iagua, *Breve historia de las conducciones* (2019; <https://www.iagua.es/noticias/noksel-espana/breve-historia-conducciones>).
3. L. Frenning, Pump Life Cycle Cost. *Scientific Report of the Hydraulic Institute & Europum & U.S. Department of Energy Office of Industrial Technologies* (2001).
4. C. A. Fernandez, *Transferencia de calor*.
5. S. S. K. et al., *Introduction to Enhanced Heat Transfer*.
6. S. T. et al., Heat transfer behaviours in a solar air heater channel with multiple V-baffle vortex generators. *Sol. Energy* **110**, 720-735 (2014).
7. M. F. et al., Heat transfer with laminar pulsating flow in a pipe. *Lett. Heat Mass Transf.* **6**, 259-270 (1979).
8. W. et al, Transient measurements of the heat transfer coefficient in unsteady, turbulent pipe flow. *Int. J. Heat Mass Transf.* **43**, 3197-3207 (2000).
9. D. A. N. et al., Forced convection with laminar pulsating flow in a channel or tube. *Int. J. Therm. Sci.* **46**, 551-560 (2007).
10. Z. G. et al., Analysis of the Nusselt number in pulsating pipe flow. *Int. J. Heat Mass Transf.* **40**, 2486-2489 (1997).
11. S. Y. et al., Unsteady characteristics of turbulent heat transfer in a circular pipe upon sudden acceleration and deceleration of flow. *Int. J. H. Mass Transfer* **113**, 490-501 (2017).
12. H. W. C. et al., numerical solutions of pulsating flow and heat transfer characteristics in a pipe. *Int. J. Heat Mass Transf.* **11**, 321-330 (1990).
13. H. H. L. P. Cheng Y. Jin, Can pulsation unsteadiness increase the convective heat transfer in a pipe flow? *numerical Heat Transfer, Part B: Fundamentals* **78**, 160-174 (2020).
14. F. P. Incropera, *Fundamentos de la transferencia de calor*.
15. M. Piller, Direct numerical simulation of turbulent forced convection in a pipe. *Direct numerical simulation of turbulent forced convection in a pipe* **49**, 583-602 (2005).
16. M. G.-V. et al., Numerical simulation of heat transfer in a pipe with non-homogeneous thermal boundary conditions. *International Journal of Heat and Fluid Flow* **55**, 45-51 (2015).
17. J. manuel López et al., nsCouette-A high performance code for direct numerical simulations of turbulent Taylor-Couette flow. *SoftwareX* **11**, 1-5 (2020).
18. D. F. J. M. L. M. R. L. S. M. Avila, A user guide for NsCouette. *github* (2020).
19. RES, *Red Española de Supercomputación* (2023; <https://www.res.es/es/nodos-de-la-res/picasso>).

20. RES, *Picasso* (2019; <https://www.scbi.uma.es/site/Supercomputing%20and%20Bioinnovation%20Center>).

APÉNDICE A

Códigos

A.1. NsPipeFlow Picasso

A.1.1. Ejemplo de input_nsPipeFlow

```
1 &parameters_grid
2 m_r    = 64                ! radial points          => m_r
   grid points (radial)
3 m_th   = 1                ! azimuthal Fourier modes => 2*m_th+1
   grid points (azimuthal)
4 m_z0   = 1                ! axial Fourier modes    => 2*m_z0+1
   grid points (axial)
5 k_th0  = 1d0              ! azimuthal wavenumber   => L_th = 2*
   pi/k_th0 azimuthal length of grid
6 k_z0   = 1d0 ! axial wavenumber          => L_z = 2*pi/k_z0 axial
   length of grid
7 /
8
9 &parameters_physics
10 Re = 1000d0              ! Re
11 const_flux = T           ! T: constant flow rate; F: constant
   pressure gradient
12 Pr = 0.1                 !Prandtl number (it is set to zero in
   the isothermal case)
13 BC = 2                   !Type of boundary condition. 1.
   Dirichlet. 2: isoflux 3: Circumferentially varying
14 constant_T = 0d0        !Value to which the mean temperature is
   fixed
15 Wo = 1d0                 ! Womersley numeric
16 Omega = 0.1d0           ! Ro/Rm
17 /
18
19 &parameters_timestep
20 numsteps    = 4000000    ! number of steps
21 init_dt     = 1.0d-3     ! initial size of timestep
22 variable_dt = T         ! use a variable (=T) or fixed (=F)
```

```

    timestep
23 maxdt      = 0.01      ! maximum size of timestep
24 Courant    = 0.25      ! CFL safety factor
25 /
26
27 &parameters_output
28 fBase_ic   = 'DNS1'    ! identifier for coeff_ (checkpoint) and
    fields_   (hdf5) files
29 dn_coeff   = 10000     ! output interval [steps] for coeff (
    dn_coeff   = -1 disables output)
30 dn_ke      = 100      ! output interval [steps] for energy
31 dn_friction = 100      ! output interval [steps] for friction
    parameters
32 dn_nusselt = 100      ! output interval [steps] for nusselt
33 dn_hdf5    = 200000000 ! output interval [steps] for HDF5
    output
34 print_time_screen = 100 ! output interval [steps] for timestep
    info to stdout
35 /
36
37 &parameters_control
38 restart    = 0          ! initialization mode: 0=new run, 1=
    restart from checkpoint (keep time), 2=restart from checkpoint
    (set time=0 and create new output files)
39 runtime    = 345600    ! maximum runtime [s] for the job
40 /
41
42 &parameters_perturbation
43 first_mode_azi = 0 !First azimuthal mode to perturb
44 first_mode_axi = 1 !First axial mode to perturb
45 n_modes_azi   = 1 ! number of modes to perturb in azimuthal
    direction
46 n_modes_axi   = 1 ! number of modes to perturb in axial direction
47 amplitude_perturbation = 0d-0 !Amplitude of the perturbation
48 /

```

A.1.2. Ejemplo de lanza.sh

```

1 #!/bin/bash
2 #SBATCH -J nsPipeRe1000_PrOp7_Wo1_pulsatile_laminar
3 #SBATCH --time=4-00:15:00
4 #SBATCH -o ./nsPipeRe1000_laminar.out
5 #SBATCH --ntasks=2
6 #SBATCH --tasks-per-node=2
7 #SBATCH --constraint=sd
8 #SBATCH --cpus-per-task=1
9 #SBATCH --error=nsPipeRe1000_laminar.%J.err
10 #SBATCH --output=nsPipeRe3000_laminar.%J.out
11 #SBATCH --mem=1gb
12
13 module load hdf5/1.12.1_gcc7.5.0_mpi3

```



```

14 export OMP_NUM_THREADS=1
15 export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/mnt/home/soft/intel/
    programs/x86_64/mkl/lib/intel64/
16
17 mpirun -np 2 ./nsPipeFlow.x < input_nsPipeFlow

```

A.2. Postprocesado

A.2.1. Postbueno.m

```

1 %nuevo tratamiento de datos
2 %Cadena de string para las rutas automaticas
3 introducir aqui los nombres y
4 los datos a utilizar relacionados con ellos
5 tam_A = {'A0p1', 'A0p2', 'A0p3', 'A0p4', 'A0p5', 'A0p6', 'A0p7',
    'A0p8', 'A0p9', 'A1p0'};
6 tam_Wo = {'Wo1', 'Wo1p5', 'Wo2', 'Wo2p5', 'Wo3', 'Wo4', 'Wo5',
    'Wo6p5', 'Wo8', 'Wo10', 'Wo12p5', 'Wo15', 'Wo17p5', 'Wo20'};
7 tam_Pr = {'Pr_0p01', 'Pr_0p1', 'Pr_0p7', 'Pr_7'};
8 A=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1];
9 Wo=[1,1.5,2,2.5,3,4,5,6.5,8,10,12.5,15,17.5,20];
10 Pr=[0.01,0.1,0.7,7];
11 Nu_pro=4.36;
12 tolerance=0.5;
13 NU_MEDIA=[];
14 %% Abro los archivos donde se van a grabar los datos
15 filename1=fopen('Error.txt','w');
16 filename2=fopen('Datos.txt','w');
17 Ampl1=fopen('Ampl1.txt','w');
18 Ampl2=fopen('Ampl2.txt','w');
19 Ampl3=fopen('Ampl3.txt','w');
20 Ampl4=fopen('Ampl4.txt','w');
21 Ampl5=fopen('Ampl5.txt','w');
22 Ampl6=fopen('Ampl6.txt','w');
23 Ampl7=fopen('Ampl7.txt','w');
24 Ampl8=fopen('Ampl8.txt','w');
25 Ampl9=fopen('Ampl9.txt','w');
26 Ampl_max=fopen('Ampl_max.txt','w');
27 fmod_ampl='\n%6s %24s %48s\n';
28 fmod_ampl_num='\n%6.4f %24f %48f\n';
29 fmod1='\n%6s %12s %18s %24s %30s %36s\n';
30 fprintf(filename2,fmod1,'Pr','Wo','A','phi','Nu','Lambda');
31 fmod2='\n%6.4f %12.4f %18.4f %24.4f %30.4f %36.4f\n';
32 fmod_error='\n%6s %12s %18s %24s\n';
33 fmod_adv='\n%6s %18s %24s %30s\n';
34 fmod_salto='\n%6s\n';
35 % inicio el escrito de los archivos para
36 %introducir los parametros utilizados
37 fprintf(Ampl1,fmod_salto,cell2mat(tam_A(1)));

```



```

38 fprintf(Ampl1,fmod_ampl,'phi','Wo','Pr');
39 fprintf(Ampl2,fmod_salto,cell2mat(tam_A(2)));
40 fprintf(Ampl2,fmod_ampl,'phi','Wo','Pr');
41 fprintf(Ampl3,fmod_salto,cell2mat(tam_A(3)));
42 fprintf(Ampl3,fmod_ampl,'phi','Wo','Pr');
43 fprintf(Ampl4,fmod_salto,cell2mat(tam_A(4)));
44 fprintf(Ampl4,fmod_ampl,'phi','Wo','Pr');
45 fprintf(Ampl5,fmod_salto,cell2mat(tam_A(5)));
46 fprintf(Ampl5,fmod_ampl,'phi','Wo','Pr');
47 fprintf(Ampl6,fmod_salto,cell2mat(tam_A(6)));
48 fprintf(Ampl6,fmod_ampl,'phi','Wo','Pr');
49 fprintf(Ampl7,fmod_salto,cell2mat(tam_A(7)));
50 fprintf(Ampl7,fmod_ampl,'phi','Wo','Pr');
51 fprintf(Ampl8,fmod_salto,cell2mat(tam_A(8)));
52 fprintf(Ampl8,fmod_ampl,'phi','Wo','Pr');
53 fprintf(Ampl9,fmod_salto,cell2mat(tam_A(9)));
54 fprintf(Ampl9,fmod_ampl,'phi','Wo','Pr');
55 fprintf(Ampl_max,fmod_salto,cell2mat(tam_A(10)));
56 fprintf(Ampl_max,fmod_ampl,'phi','Wo','Pr');
57 %% comienza el bucle para el tratamiento de datos
58 for i=1:length(tam_Pr)
59     %si quiero que tambien me guarde el n de Pr de los datos
60     fprintf(filename2,fmod_salto,cell2mat(tam_Pr(i)));
61     for j=1:length(tam_Wo)
62         % si quiero que me guarde n Wo de los datos
63         fprintf(filename2,fmod_salto,cell2mat(tam_Wo(j)));
64         for n=1:length(tam_A)
65             % abro lo datos y llamo a la
66             %funcion de tratar Nusselt
67             Nusselt=load(['../',char(tam_Pr(i)),'/',char(tam_Wo(j)),'/',
68                 char(tam_A(n)),'/Nusselt']);
69             friction=load(['../',char(tam_Pr(i)),'/',char(tam_Wo(j)),'/',
70                 char(tam_A(n)),'/friction']);
71             [minimo,maximo,Nusselt_trat,t_trat,N_ciclos,flag_error,
72                 flag_adv] =trat_Nusselt(Nusselt);
73             % si la funcion Nusselt devuelve flag_error escribe error en
74             % el archivo
75             % error
76             % if flag_error==true
77             % fprintf(filename1,fmod_error,'Error',cell2mat(tam_Pr(i)),
78                 cell2mat(tam_Wo(j)),cell2mat(tam_A(n)));
79             %si la funcion Nusselt devuelve flag_adv nos escribe una
80             % advertencia en
81             %el archivo error puede que no este mal esas simulaciones pero
82             % existe
83             %algo anomalo
84             % elseif flag_adv==true
85             % [fric_trat,V_trat,tf_trat,T_fric]=trat_fric(friction,N_ciclos
86                 );
87             %trato los datos obtenidos
88             % NU_MEDIA(n)=mean(Nusselt_trat,'all');

```



```

81 % FRI_MEDIA=mean(fric_trat,'all');
82 % V_MEDIA(n)=mean(V_trat,'all');
83 % phi(n)=tf_trat(end)-t_trat(end);
84 % Lambda=NU_MEDIA(n)/Nu_pro;
85
86 % fprintf(filename1,fmod_adv,'Advertencia faltan ciclos',
      cell2mat(tam_Pr(i)),cell2mat(tam_Wo(j)),cell2mat(tam_A(n)));
87 % fprintf(filename2,fmod2,Pr(i),Wo(j),A(n),phi(n),NU_MEDIA(n),
      Lambda);
88 % elseif n>2
89 % NU_MEDIA(n)=mean(Nusselt_trat,'all');
90 % FRI_MEDIA=mean(fric_trat,'all');
91 % V_MEDIA(n)=mean(V_trat,'all');
92 % phi(n)=tf_trat(end)-t_trat(end);
93 %si la funcion Nusselt devuelve flag_adv nos escribe una
      advertencia en
94 %el archivo error puede que no este mal esas simulaciones pero
      existe
95 %algo anomalo, tambien nos plotea los dos datos para observar
      si esta la
96 %simulacion o si existe algo anomalo
97 % if abs((NU_MEDIA(n))-(NU_MEDIA(n-1)))>=tolerance
98 % fprintf(filename1,fmod_adv,'Anomalia en NU',cell2mat(tam_Pr(
      i)),cell2mat(tam_Wo(j)),cell2mat(tam_A(n)));
99 % Nusselt_2=load([char(tam_Pr(i)),'/',char(tam_Wo(j)),'/',char
      (tam_A(n-1)),'/Nusselt']);
100 % figure (1)
101 % hold on
102 % plot(Nusselt(:,2),'-b')
103 % plot(Nusselt_2(:,2),'-r')
104 % legend(char(tam_A(n)),char(tam_A(n-1)))
105 % title(sprintf('%s_%s_%s',char(tam_Pr(i)),'_',char(tam_Wo(j))
      ,'_ ',char(tam_A(n))))
106 % hold off
107 % cd Anomalia_NU
108 % saveas(gcf,sprintf('%s_%s_%s_%s_NU.jpeg',char(tam_Pr(i)),
      char(tam_Wo(j)),char(tam_A(n)),char(tam_A(n-1))));
109 % clf
110 % cd ../
111 % lo mismo anterior pero para el desfase
112 % elseif abs(phi(n))>(abs(phi(n-1))+10^4) && abs(phi(n))
      >=10^5 && abs(phi(n))<=10^4 || (abs(phi(n))-10^4)<abs(phi(n
      -1)) && abs(phi(n))>=10^5 && abs(phi(n))<=10^4
113 % fprintf(filename1,fmod_adv,'Anomalia en phi',cell2mat(tam_Pr
      (i)),cell2mat(tam_Wo(j)),cell2mat(tam_A(n)));
114 % Nusselt_2=load([char(tam_Pr(i)),'/',char(tam_Wo(j)),'/',char
      (tam_A(n-1)),'/Nusselt']);
115 % friction_2=load([char(tam_Pr(i)),'/',char(tam_Wo(j)),'/',
      char(tam_A(n-1)),'/friction']);
116 % media_v=mean(friction(:,2),'all');
117 % sum=NU_MEDIA(n)-V_MEDIA(n);

```



```

118 % figure (2)
119 % subplot(2,1,1);
120 % hold on
121 % plot(Nusselt(:,2),'-b');
122 % plot(friction(:,2)+sum,'-r');
123 % legend(sprintf('%s_%s',char(tam_A(n)),'_NU'),sprintf('%s_%s
    ',char(tam_A(n)),'_vel'))
124 % title(sprintf('%s_%s_%s_%s',char(tam_Pr(i)),char(tam_Wo(j)),
    char(tam_A(n)),char(tam_A(n-1))))
125 % hold off
126 % media_v2=mean(friction(:,2),'all');
127 % sum=NU_MEDIA(n-1)-V_MEDIA(n-1);
128 % subplot(2,1,2);
129 % hold on
130 % plot(Nusselt_2(:,2),'-b');
131 % plot(friction_2(:,2)+sum,'-r');
132 % legend(sprintf('%s_%s',char(tam_A(n-1)),'_NU'),sprintf('%s_%
    s',char(tam_A(n-1)),'_vel'))
133 % hold off
134 % cd Anomalia_phi
135 % saveas(gcf,sprintf('%s_%s_%s_%s_phi.jpeg',char(tam_Pr(i)),
    char(tam_Wo(j)),char(tam_A(n)),char(tam_A(n-1))));
136 % clf
137 % cd ../
138 % clear figure
139 % elseif abs(phi(n))>(abs(phi(n-1))+10^3) && abs(phi(n))
    <10^4 && abs(phi(n))>=10^3 || (abs(phi(n))-10^3)<abs(phi(n
    -1)) && abs(phi(n))<10^4 && abs(phi(n))>=10^3
140 % fprintf(filename1,fmod_adv,'Anomalia en phi',cell2mat(tam_Pr
    (i)),cell2mat(tam_Wo(j)),cell2mat(tam_A(n)));
141 % Nusselt_2=load([char(tam_Pr(i)),'/',char(tam_Wo(j)),'/',char
    (tam_A(n-1)),'/Nusselt']);
142 % friction_2=load([char(tam_Pr(i)),'/',char(tam_Wo(j)),'/',
    char(tam_A(n-1)),'/friction']);
143 % media_v=mean(friction(:,2),'all');
144 % sum=NU_MEDIA(n)-V_MEDIA(n);
145 % figure (2)
146 % subplot(2,1,1);
147 % hold on
148 % plot(Nusselt(:,2),'-b');
149 % plot(friction(:,2)+sum,'-r');
150 % legend(sprintf('%s_%s',char(tam_A(n)),'_NU'),sprintf('%s_%s
    ',char(tam_A(n)),'_vel'))
151 % title(sprintf('%s_%s_%s_%s',char(tam_Pr(i)),char(tam_Wo(j)),
    char(tam_A(n)),char(tam_A(n-1))))
152 % hold off
153 % media_v2=mean(friction(:,2),'all');
154 % sum=NU_MEDIA(n-1)-V_MEDIA(n-1);
155 % subplot(2,1,2);
156 %hold on
157 %plot(Nusselt_2(:,2),'-b');

```

```

158 %plot(friction_2(:,2)+sum, '-r');
159 %legend(sprintf('%s_%s', char(tam_A(n-1)), '_NU'), sprintf('%s_%s',
    char(tam_A(n-1)), '_vel'))
160 %hold off
161 %cd Anomalia_phi
162 %saveas(gcf, sprintf('%s_%s_%s_%s_phi.jpeg', char(tam_Pr(i)),
    char(tam_Wo(j)), char(tam_A(n)), char(tam_A(n-1))));
163 %clf
164 %cd ../
165 %clear figure
166 % elseif abs(phi(n))>(abs(phi(n-1))+10^2) && abs(phi(n))
    <10^3 && abs(phi(n))>=10^2 || (abs(phi(n))-10^2)<abs(phi(n-1))
    && abs(phi(n))<10^3 && abs(phi(n))>=10^3
167 % fprintf(filename1,fmod_adv,'Anomalia en phi',cell2mat(
    tam_Pr(i)), cell2mat(tam_Wo(j)), cell2mat(tam_A(n)));
168 % Nusselt_2=load([char(tam_Pr(i)), '/', char(tam_Wo(j)), '/',
    char(tam_A(n-1)), '/Nusselt']);
169 % friction_2=load([char(tam_Pr(i)), '/', char(tam_Wo(j)), '/',
    char(tam_A(n-1)), '/friction']);
170 % media_v=mean(friction(:,2), 'all');
171 % sum=NU_MEDIA(n)-V_MEDIA(n);
172 % figure (2)
173 % subplot(2,1,1);
174 % hold on
175 % plot(Nusselt(:,2), '-b');
176 % plot(friction(:,2)+sum, '-r');
177 % legend(sprintf('%s_%s', char(tam_A(n)), '_NU'), sprintf('%s_%s',
    char(tam_A(n)), '_vel'))
178 % title(sprintf('%s_%s_%s_%s', char(tam_Pr(i)), char(tam_Wo(j)),
    char(tam_A(n)), char(tam_A(n-1))))
179 % hold off
180 % media_v2=mean(friction(:,2), 'all');
181 % sum=NU_MEDIA(n-1)-V_MEDIA(n-1);
182 % subplot(2,1,2);
183 % hold on
184 % plot(Nusselt_2(:,2), '-b');
185 % plot(friction_2(:,2)+sum, '-r');
186 % legend(sprintf('%s_%s', char(tam_A(n-1)), '_NU'), sprintf('%s_
    %s', char(tam_A(n-1)), '_vel'))
187 % hold off
188 % cd Anomalia_phi
189 % saveas(gcf, sprintf('%s_%s_%s_%s_phi.jpeg', char(tam_Pr(i)),
    char(tam_Wo(j)), char(tam_A(n)), char(tam_A(n-1))));
190 % clf
191 % cd ../
192 % clear figure
193 % elseif abs(phi(n))>(abs(phi(n-1))+10) && abs(phi(n))<10^2
    && abs(phi(n))>=0 || abs(abs(phi(n))-10)<abs(phi(n-1)) && abs(
    phi(n))<10^2 && abs(phi(n))<=0
194 % fprintf(filename1,fmod_adv,'Anomalia en phi',cell2mat(
    tam_Pr(i)), cell2mat(tam_Wo(j)), cell2mat(tam_A(n)));

```



```

195 %     Nusselt_2=load([char(tam_Pr(i)), '/', char(tam_Wo(j)), '/',
char(tam_A(n-1)), '/Nusselt']);
196 %     friction_2=load([char(tam_Pr(i)), '/', char(tam_Wo(j)), '/',
char(tam_A(n-1)), '/friction']);
197 %     media_v=mean(friction(:,2), 'all');
198 %     sum=NU_MEDIA(n)-V_MEDIA(n);
199 %     figure (2)
200 %     subplot(2,1,1);
201 %     hold on
202 %     plot(Nusselt(:,2), '-b');
203 %     plot(friction(:,2)+sum, '-r');
204 %     legend(sprintf('%s_%s', char(tam_A(n)), '_NU'), sprintf('%s_%s
', char(tam_A(n)), '_vel'))
205 %     title(sprintf('%s_%s_%s_%s', char(tam_Pr(i)), char(tam_Wo(j))
, char(tam_A(n)), char(tam_A(n-1))))
206 %     hold off
207 %     media_v2=mean(friction(:,2), 'all');
208 %     sum=NU_MEDIA(n-1)-V_MEDIA(n-1);
209 %     subplot(2,1,2);
210 %     hold on
211 %     plot(Nusselt_2(:,2), '-b');
212 %     plot(friction_2(:,2)+sum, '-r');
213 %     legend(sprintf('%s_%s', char(tam_A(n-1)), '_NU'), sprintf('%s_
%s', char(tam_A(n-1)), '_vel'))
214 %     hold off
215 %     cd Anomalia_phi
216 %     saveas(gcf, sprintf('%s_%s_%s_%s_phi.jpeg', char(tam_Pr(i)),
char(tam_Wo(j)), char(tam_A(n)), char(tam_A(n-1))));
217 %     clf
218 %     cd ../
219 %     clear figure
220 %     end
221 %     Lambda=NU_MEDIA(n)/Nu_pro;
222 %     fprintf(filename2, fmod2, Pr(i), Wo(j), A(n), phi(n), NU_MEDIA(n)
, Lambda);
223 %     else
224 [fric_trat, V_trat, tf_trat, T_fric]=trat_fric(friction, N_ciclos
);
225 NU_MEDIA(n)=mean(Nusselt_trat, 'all');
226 FRI_MEDIA=mean(fric_trat, 'all');
227 phi(n)=tf_trat(end)-t_trat(end);
228 Lambda=NU_MEDIA(n)/Nu_pro;
229 %     %guardo los datos
230 fprintf(filename2, fmod2, Pr(i), Wo(j), A(n), phi(n), NU_MEDIA(n),
Lambda);
231
232 % end
233 %guardo datos
234 if tam_A(n)=="A0p1" && flag_error~=true
235     fprintf(Ampl1, fmod_ampl_num, T_fric, Wo(j), Pr(i));
236     elseif tam_A(n)=="A0p2"

```

```

237     fprintf(Ampl2,fmod_ampl_num,T_fric,Wo(j),Pr(i));
238     elseif tam_A(n)=="A0p3"
239         fprintf(Ampl3,fmod_ampl_num,T_fric,Wo(j),Pr(i));
240     elseif tam_A(n)=="A0p4"
241         fprintf(Ampl4,fmod_ampl_num,T_fric,Wo(j),Pr(i));
242     elseif tam_A(n)=="A0p5"
243         fprintf(Ampl5,fmod_ampl_num,T_fric,Wo(j),Pr(i));
244     elseif tam_A(n)=="A0p6"
245         fprintf(Ampl6,fmod_ampl_num,T_fric,Wo(j),Pr(i));
246     elseif tam_A(n)=="A0p7"
247         fprintf(Ampl7,fmod_ampl_num,T_fric,Wo(j),Pr(i));
248     elseif tam_A(n)=="A0p8"
249         fprintf(Ampl8,fmod_ampl_num,T_fric,Wo(j),Pr(i));
250     elseif tam_A(n)=="A0p9"
251         fprintf(Ampl9,fmod_ampl_num,T_fric,Wo(j),Pr(i));
252     elseif tam_A(n)=="A1"
253         fprintf(Ampl_max,fmod_ampl_num,T_fric,Wo(j),Pr(i));
254     end
255     clear NU_MEDIA
256     clear phi
257     end
258 end
259 end
260 fclose(filename1);
261 fclose(filename2);
262
263 %% parametros
264 function [min_t,max_t,Nusselt_trat,tN_trat,N_ciclos,flag_error,
        flag_adv]=trat_Nusselt(Nusselt)
265 %%trato de datos
266 flag=false;
267 flag_error=false;
268 flag_adv=false;
269 flag_pos=false;
270 max_val=[];
271 max_pos=[];
272 min_val=[];
273 min_pos=[];
274 tolerance=5e-3;
275 i=1;
276 %% inicializacion de bucles
277 dat=Nusselt(:,2);
278 pdif=diff(dat(:));
279 pre=diff(pdif(:));
280 %% busqueda de mins y max
281 while flag==false
282     if i < length(pdif) && i+1 < length(pdif)
283     if (pdif(i)<0) && (pdif(i+1)>0) && (pre(i)>0)
284         min_val=[min_val,Nusselt(i+1,2)];
285         min_pos=[min_pos,i+1];
286     elseif (pdif(i)<0) && (pdif(i+1)<0) && (pre(i)<0)

```



```

287     max_val=[max_val ,Nusselt(i+1,2)];
288     max_pos=[max_pos ,i+1];
289 end
290 elseif i==length(pdif-1)
291     flag=true;
292 end
293 i=1+i;
294 end
295
296 j=size(min_val ,2);
297 if min_val ~= isempty(min_val)
298 while flag_pos==false
299 if length(min_pos)<2
300     flag_error=true;
301     min_t=0;
302     max_t=0;
303     Nusselt_trat=0;
304     tN_trat=0;
305     N_ciclos=0;
306     return
307 elseif abs(abs(min_val(j))-(abs(min_val(j-1))))>=tolerance
308     cor=min_pos(j);
309     flag_pos=true;
310 elseif j==2
311     cor=min_pos(j);
312     flag_pos=true;
313 end
314 j=j-1;
315 end
316 j=j+1;
317 Nusselt_trat=Nusselt(cor:min_pos(end) ,2);
318 T=Nusselt(min_pos(end) ,1)-Nusselt(min_pos(end-1) ,1);
319 tN_trat=Nusselt(cor:min_pos(end) ,1);
320 max_t=max(Nusselt_trat);
321 min_t=min(Nusselt_trat);
322 N_ciclos=abs((min_pos(end)-cor)/T);
323 if N_ciclos<3 && N_ciclos>=2
324     flag_adv=true;
325 end
326 else
327     flag_error=true;
328     min_t=0;
329     max_t=0;
330     Nusselt_trat=0;
331     tN_trat=0;
332     N_ciclos=0;
333     return
334
335 end
336 end
337

```



```

338 %% segunda funcion
339 function [fric_trat ,V_trat ,tf_trat ,T_fric]=trat_fric(friction ,
    N_ciclos)
340 flag=false;
341 max_val=[];
342 max_pos=[];
343 min_val=[];
344 min_pos=[];
345 i=1;
346 %% inicializacion de bucles
347 dat=friction(:,2);
348 pdif=diff(dat(:));
349 pre=diff(pdif(:));
350 %% busqueda de mins y max
351
352 while flag==false
353 if i < length(pdif) && i+1 < length(pdif)
354 if (pdif(i)<0) && (pdif(i+1)>0) && (pre(i)>0)
355     min_val=[min_val,friction(i+1,2)];
356     min_pos=[min_pos,i+1];
357 elseif (pdif(i)<0) && (pdif(i+1)<0) && (pre(i)<0)
358     max_val=[max_val,friction(i+1,2)];
359     max_pos=[max_pos,i+1];
360 end
361 else
362     flag=true;
363 end
364 i=1+i;
365 end
366 T_fric=abs((friction(min_pos(end),2))-(friction(min_pos(end-1),2)
    ));
367 tf_trat=friction((min_pos(end)-T_fric*N_ciclos):min_pos(end),1);
368 V_trat=friction((min_pos(end)-T_fric*N_ciclos):(min_pos(end)),2);
369 fric_trat=friction((min_pos(end))-T_fric*N_ciclos):(min_pos(end)
    ),5);
370 end

```

Muestreo.m

```

1 %% muestreo
2 clc
3 clear all
4 close all
5 datos=load('../datos.txt');
6 filename=fopen('repetir.txt','w');
7 fmod='\n%6.5f %12.5f %18.5f %24.5f\n';
8 for i=1:length(datos)
9     impac=datos(i,6);
10    if impac<0
11        fprintf(filename,fmod,datos(i,1),datos(i,2),datos(i,3),datos(
            i,6));

```



```

12     end
13 end
14 fclose(filename);

```

A.2.2. generar_color_map.m

```

1 %Crear colormap para un Prandtl determinado
2
3 clear all
4 close all
5 clc
6
7 %Leer fichero de datos
8 a = load('../..//datos.txt');
9
10 disp('Pr para el cual quiere hacer el colormap ');
11 disp('Pr0p01----->1');
12 disp('Pr0p1----->2');
13 disp('Pr0p7----->3');
14 disp('Pr7----->4');
15 b=input('Elige:');
16 switch b
17     case 1
18         n=1;
19     case 2
20         n=2;
21     case 3
22         n=3;
23     case 4
24         n=4;
25     otherwise
26         return
27 end
28 %Pr0.01
29 %n=1;
30 %Pr0.1
31 %n=2;
32 %Pr0.7
33 %n=3;
34 %Pr7
35 %n=4;
36
37
38 i1=1+(n-1)*140;
39 i2=n*140;
40
41
42 A_total = a(i1:i2,3);
43 factor_impacto = a(i1:i2,6);
44 Wo_total = a(i1:i2,2);
45
46 nA=10;%estudiamos 10 amplitudes

```



```

47 A=A_total(1:nA);
48 nWo=length(Wo_total(1:nA:end));
49 Wo=Wo_total(1:nA:end);
50 values_fi = reshape_custom(nWo,nA,factor_impacto) ;
51
52 [A_grid,Wo_grid] = meshgrid(A,Wo);
53
54 %Dibujamos el mapa de color
55 f = figure();
56 set(gcf,'units','centimeters','position',[10,10,19,8])
57
58 colormap redblue
59 surf(Wo_grid,A_grid,values_fi)
60 view(2)
61 shading interp
62 colorbar
63 grid on
64 ylim([0.1 1]);
65 xlim([1 20]);
66 ylabel('Amplitud','Interpreter','Latex','FontSize',12);
67 xlabel('Wo','Interpreter','Latex','FontSize',12);
68 title('Colormap $Pr=7$','Interpreter','Latex','FontSize',14);
69 %Editar colormap para poner el blanco en el 1
70
71 load custom_colormap.mat
72
73 colormapeditor

```

Funciones complementarias

```

1 function c = redblue(m)
2 %REDBLUE      Shades of red and blue color map
3 % REDBLUE(M), is an M-by-3 matrix that defines a colormap.
4 % The colors begin with bright blue, range through shades of
5 % blue to white, and then through shades of red to bright red.
6 % REDBLUE, by itself, is the same length as the current figure'
   s
7 % colormap. If no figure exists, MATLAB creates one.
8 %
9 % For example, to reset the colormap of the current figure:
10 %
11 %             colormap(redblue)
12 %
13 % See also HSV, GRAY, HOT, BONE, COPPER, PINK, FLAG,
14 % COLORMAP, RGBPLOT.
15
16 % Adam Auton, 9th October 2009
17
18 if nargin < 1, m = size(get(gcf,'colormap'),1); end
19
20 if (mod(m,2) == 0)

```



```

21 % From [0 0 1] to [1 1 1], then [1 1 1] to [1 0 0];
22 m1 = m*0.5;
23 r = (0:m1-1)'/max(m1-1,1);
24 g = r;
25 r = [r; ones(m1,1)];
26 g = [g; flipud(g)];
27 b = flipud(r);
28 else
29 % From [0 0 1] to [1 1 1] to [1 0 0];
30 m1 = floor(m*0.5);
31 r = (0:m1-1)'/max(m1,1);
32 g = r;
33 r = [r; ones(m1+1,1)];
34 g = [g; 1; flipud(g)];
35 b = flipud(r);
36 end
37
38 c = [r g b];

```

```

1 function [A]=reshape_custom(nr,nz,vec)
2
3 A = zeros(nr,nz);
4
5 for ii=1:nr
6     for jj=1:nz
7         A(ii,jj) = vec((ii-1)*nz+jj);
8     end
9 end

```

A.2.3. trat_perfil_optimizado

```

1 %% tratamiento del perfil de velocidad
2 %directorio donde tengo que lanzar el codigo
3 %hay que utilizar restart 2
4 clear all
5 close all
6 clc
7
8 %% graficas
9 disp('Elige el perfil de velocidad que necesites: ')
10 disp('Ganacia')
11 disp(' factor de impacto=Wo1_A0p7_1.019 ----->1')
12 disp(' factor de impacto=Wo1p5_A0p9_1.031 ----->2')
13 disp(' factor de impacto=Wo2_A0p9_1.028 ----->3')
14 disp('Perdida')
15 disp(' factor de impacto=Wo8_A0p8_0.6926----->4')
16 disp(' factor de impacto=Wo8_A1p0_0.0046----->5')
17 disp(' factor de impacto=Wo15_A0p9_0.8867 ----->6')
18 disp('sin ganancia')
19 disp(' factor de impacto=Wo5_A0p2_1 ----->7') %guardar esta
    curva luego en un buffer parametrizada para la comparacion con
    los otros datos

```

```

20 disp(' factor de impacto=Wo5_A0p3_0.9946 ----->8')
21 d=input('Elige el caso: ');
22 switch d
23     case 1
24         cd '../perfiles/Pr0p01_perfil/Wo1_A0p7';
25     case 2
26         cd '../perfiles/Pr0p01_perfil/Wo1p5_A0p9';
27     case 3
28         cd '../perfiles/Pr0p01_perfil/Wo2_A0p9';
29     case 4
30         cd '../perfiles/Pr0p01_perfil/Wo8_A0p8';
31     case 5
32         cd '../perfiles/Pr0p01_perfil/Wo8_A1p0';
33     case 6
34         cd '../perfiles/Pr0p01_perfil/Wo15_A0p9';
35     case 7
36         cd '../perfiles/Pr0p01_perfil/Wo5_A0p2';
37     case 8
38         cd '../perfiles/Pr0p01_perfil/Wo5_A0p3';
39     otherwise
40         i=input('Quieres elegir otro caso no estipulado:
41                 ');
42         switch i
43             case 'si'
44                 d=input('Introduzca el directorio: ');
45                 cd (d);
46             case 'no'
47                 return
48             otherwise
49                 return
50         end
51     end
52 %%
53 % parametros
54 flag=false;
55 j=0;
56 n=1;
57 r=1;
58 t_trat=40000;
59 sum_rT=[];
60 sum_T=[];
61 sum_r=[];
62 sum_v=[];
63 num_bueno=[];
64 Nusselt=load('Nusselt');
65 friction=load('friction');
66 [min_pos_n,min_pos_nl1,trat_t,T_fric]=minimo(Nusselt,friction);
67 T
68 %hago un array con los directorios que tengo
69 archivos_vel=dir('meanprof*');
70 archivos_temp=dir('Temp_meanprof*');

```



```

70 %utilizando datos de la funcion borrar_Nusselt
71 for k=1:length(archivos_vel)
72     num_char = regexp(archivos_vel(k).name, '\d+', 'match');
73     num(k)= str2double(num_char);
74 end
75 %encuentrar el archivo justo despues del corte
76 while flag==false
77     if num(n)>=((min_pos_n1*100))
78         if isempty(num_bueno)
79             pos_ini=n;
80             end
81         num_bueno(r)=num(n);
82         r=r+1;
83     end
84     if num(n+1)>=(min_pos_n*100)
85         num_bueno(r)=num(n+1);
86         pos_fin=n;
87         flag=true;
88     end
89     n=n+1;
90 end
91 %guardo las variables cargadas y las sumo
92 % para el perfil de velocidad
93 % probablemente haya que cambiar los nombres segun el archivo
    primero ver
94 % el archivo que tratar
95 for i=pos_ini:1:pos_fin
96     vel=importdata(archivos_vel(i).name);
97     ra=str2double(vel.textdata(3:end,1));
98     v=str2double(vel.textdata(3:end,2));
99     for j= 1 : length(ra)
100     if isempty(sum_r)
101         sum_r=ra;
102         sum_v=v;
103     else
104         sum_r(j)=ra(j)+sum_r(j);
105         sum_v(j)=v(j)+sum_v(j);
106     end
107     end
108     end
109 %% para perfil de temperatura
110 for i=pos_ini:1:pos_fin
111     vel=importdata(archivos_temp(i).name);
112     ra=str2double(vel.textdata(3:end,1));
113     T=str2double(vel.textdata(3:end,2));
114     for j=1:length(ra)
115     if isempty(sum_rT)
116         sum_rT=ra;
117         sum_T=T;
118     else
119         sum_rT(j)=ra(j)+sum_r(j);

```

```

120     sum_T(j)=T(j)+sum_T(j);
121     end
122     end
123
124     end
125
126     perf_vel_radio=sum_r./length(num_bueno);
127     perf_vel_v=sum_v./length(num_bueno);
128     perf_T_radio=sum_rT./length(num_bueno);
129     perf_T_T=sum_T./length(num_bueno);
130     perfil_T=fopen('perfil_T.txt','w');
131     perfil_vel=fopen('perfil_vel.txt','w');
132     fmod_ampl_num='\n%6.4f %24.4f\n';
133
134     for i=1:length(perf_vel_radio)
135         fprintf(perfil_vel,fmod_ampl_num,perf_vel_radio(i),perf_vel_v(i)
136             );
137     end
138     for i=1:length(perf_T_radio)
139         fprintf(perfil_T,fmod_ampl_num,perf_T_radio(i),perf_T_T(i));
140     end
141     fclose(perfil_T);
142     fclose(perfil_vel);
143 %% funcion de Nusselt
144 function [min,min_1,t_trat,T_fric]=minimo(Nusselt,friction)
145 %hago las derivada
146 dat=Nusselt(:,2);
147 pdif=diff(dat(:));
148 pre=diff(pdif(:));
149 flag=false;
150 i=1;
151 min_val=[];
152 min_pos=[];
153 max_val=[];
154 max_pos=[];
155 min_val_f=[];
156 min_pos_f=[];
157 max_val_f=[];
158 max_pos_f=[];
159 %busco los minimos del Nusselt
160 while flag==false
161     if i < length(pdif) && i+1 < length(pdif)
162         if (pdif(i)<0) && (pdif(i+1)>0) && (pre(i)>0)
163             min_val=[min_val,Nusselt(i+1,2)];
164             min_pos=[min_pos,i+1];
165         elseif (pdif(i)<0) && (pdif(i+1)<0) && (pre(i)<0)
166             max_val=[max_val,Nusselt(i+1,2)];
167             max_pos=[max_pos,i+1];
168         end
169     else
170         flag=true;

```



```

170 end
171 i=1+i;
172 end
173 %busco el periodo de la velocidad impuesta
174 dat_f=friction(:,2);
175 pdif_f=diff(dat_f(:));
176 pre_f=diff(pdif_f(:));
177 flag_1=false;
178 i=1;
179 %mismo codigo que el anterior
180 while flag_1==false
181 if i < length(pdif_f) && i+1 < length(pdif_f)
182 if (pdif_f(i)<0) && (pdif_f(i+1)>0) && (pre_f(i)>0)
183     min_val_f=[min_val_f,friction(i+1,2)];
184     min_pos_f=[min_pos_f,i+1];
185 elseif (pdif_f(i)<0) && (pdif_f(i+1)<0) && (pre_f(i)<0)
186     max_val_f=[max_val_f,friction(i+1,2)];
187     max_pos_f=[max_pos_f,i+1];
188 end
189 else
190     flag_1=true;
191 end
192 i=1+i;
193 end
194 %segun el dato tengo que cambiar el minimo que saco de la funcion
195 T=(min_pos_f(end)-min_pos_f(length(min_pos_f)-1));
196 t_trat=friction(min_pos_f(length(min_pos_f)-1):min_pos_f(end),2);
197 min=min_pos_f(end);
198 min_1=min_pos_f(length(min_pos_f)-1);
199
200
201 end

```

A.2.4. obtencioncurvas.m

```

1 %% graficas
2 clear all
3 close all
4 clc
5
6 disp('Elige el perfil de velocidad que necesites: ')
7 disp('Ganacia')
8 disp(' factor de impacto=Wo1_A0p7_1.019 ----->1')
9 disp(' factor de impacto=Wo1p5_A0p9_1.031 ----->2')
10 disp(' factor de impacto=Wo2_A0p9_1.028 ----->3')
11 disp('Perdida')
12 disp(' factor de impacto=Wo8_A0p8_0.6926----->4')
13 disp(' factor de impacto=Wo8_A1p0_0.0046----->5')
14 disp(' factor de impacto=Wo15_A0p9_0.8867 ----->6')
15 disp('Sin ganancia')
16 disp('factor de impacto=Wo5_A0p2_1 ----->7') % guardar esta curva
    luego en un buffer parametrizada para la comparacion con los

```

```

    otros datos
17 disp(' factor de impacto=Wo5_A0p3_0.9946 ----->8')
18     d=input('Elige el caso: ');
19     switch d
20         case 1
21             cd '../perfiles/Pr0p01_perfil/Wo1_A0p7';
22         case 2
23             cd '../perfiles/Pr0p01_perfil/Wo1p5_A0p9';
24         case 3
25             cd '../perfiles/Pr0p01_perfil/Wo2_A0p9';
26         case 4
27             cd '../perfiles/Pr0p01_perfil/Wo8_A0p8';
28         case 5
29             cd '../perfiles/Pr0p01_perfil/Wo8_A1p0';
30         case 6
31             cd '../perfiles/Pr0p01_perfil/Wo15_A0p9';
32         case 7
33             cd '../perfiles/Pr0p01_perfil/Wo5_A0p2';
34         case 8
35             cd '../perfiles/Pr0p01_perfil/Wo5_A0p3';
36         case 'si'
37             %Hay que poner comillas para meter un directorio
38             i=input('Quieres elegir otro caso no estipulado:
39                 ');
40             switch i
41                 case 'si'
42                     d=input('Introduzca el directorio: ');
43                     cd (d);
44                 case 'no'
45                     return
46                 otherwise
47                     return
48             end
49         end
50
51 perfil_vel=load('perfil_vel.txt');
52 perfil_T=load('perfil_T.txt');
53 figure (1)
54 hold on
55 plot(perfil_vel(:,1),perfil_vel(:,2));
56 plot(perfil_vel(:,1),2.*(1-perfil_vel(:,1).^2))
57 legend('simulacion','estacionario','Interpreter','Latex','
58     Fontsize',8);
59 title('Perfil velocidad','Interpreter','Latex','Fontsize',12);
60 xlabel('r','Interpreter','Latex','Fontsize',10);
61 ylabel('V','Interpreter','Latex','Fontsize',10);
62 hold off
63 cd '../..//Pr0p01_perfil/Wo5_A0p2';
64 estacionario=load('perfil_T.txt');
65 figure (2)
66 hold on

```



```

65 plot(perfil_T(:,1),perfil_T(:,2));
66 plot(estacionario(:,1),estacionario(:,2));
67 legend('simulacion','estacionario','Interpreter','Latex','
        Fontsize',8);
68 title('Perfil de temperatura','Interpreter','Latex','Fontsize'
        ,12);
69 xlabel('r','Interpreter','Latex','Fontsize',10);
70 ylabel('T','Interpreter','Latex','Fontsize',10);
71 hold off

```

A.2.5. desfase.m

```

1 %% desfase de graficas
2 clear all
3 %close all
4 clc
5
6
7 %% carga de datos
8 disp('Elige el perfil de velocidad que necesites: ')
9 disp('Ganacia')
10 disp(' factor de impacto=Wo1_A0p7_1.019 ----->1')
11 disp(' factor de impacto=Wo1p5_A0p9_1.031 ----->2')
12 disp(' factor de impacto=Wo2_A0p9_1.028 ----->3')
13 disp('Perdida')
14 disp(' factor de impacto=Wo8_A0p8_0.6926----->4')
15 disp(' factor de impacto=Wo8_A1p0_0.0046----->5')
16 disp(' factor de impacto=Wo15_A0p9_0.8867 ----->6')
17 disp('Sin ganancia')
18 disp(' factor de impacto=Wo5_A0p2_1 ----->7')
19 % guardar esta curva luego en un buffer parametrizada para la
    comparacion con los otros datos
20 disp(' factor de impacto=Wo5_A0p3_0.9946 ----->8')
21 d=input('Elige el caso: ');
22 switch d
23     case 1
24         cd '../perfiles/Pr0p01_perfil/Wo1_A0p7';
25     case 2
26         cd '../perfiles/Pr0p01_perfil/Wo1p5_A0p9';
27     case 3
28         cd '../perfiles/Pr0p01_perfil/Wo2_A0p9';
29     case 4
30         cd '../perfiles/Pr0p01_perfil/Wo8_A0p8';
31     case 5
32         cd '../perfiles/Pr0p01_perfil/Wo8_A1p0';
33     case 6
34         cd '../perfiles/Pr0p01_perfil/Wo15_A0p9';
35     case 7
36         cd '../perfiles/Pr0p01_perfil/Wo5_A0p2';
37     case 8
38         cd '../perfiles/Pr0p01_perfil/Wo5_A0p3';
39     case 'si'

```



```

40         i=input('Quieres elegir otro caso no estipulado:
41                ');
42         switch i
43             case si
44                 d=input('Introduzca el directorio: ');
45                 cd (d);
46             case no
47                 return
48             otherwise
49                 return
50         end
51     end
52 Nusselt=load('Nusselt');
53 friction=load('friction');
54 [min_f_superior,min_f_inferior,min_Nuss_sup,min_Nuss_inf]=minimo(
55     Nusselt,friction);
56 figure (3)
57 hold on
58 plot(Nusselt(:,2))
59 plot(friction(:,2))
60 xline(min_f_superior,'m');
61 xline(min_f_inferior,'g');
62 xlabel('t','Interpreter','Latex','FontSize',10);
63 legend('Nusselt','friccion','Interpreter','Latex','FontSize',8);
64 hold off
65 %% funcion de Nusselt
66 function [min,min_1,min_Nuss_sup,min_Nuss_inf]=minimo(Nusselt,
67     friction)
68 %hago las derivada
69 dat=Nusselt(:,2);
70 pdif=diff(dat(:));
71 pre=diff(pdif(:));
72 flag=false;
73 i=1;
74 min_val=[];
75 min_pos=[];
76 max_val=[];
77 max_pos=[];
78 min_val_f=[];
79 min_pos_f=[];
80 max_val_f=[];
81 max_pos_f=[];
82 %busco los minimos del Nusselt
83 while flag==false
84     if i < length(pdif) && i+1 < length(pdif)
85         if (pdif(i)<0) && (pdif(i+1)>0) && (pre(i)>0)
86             min_val=[min_val,Nusselt(i+1,2)];
87             min_pos=[min_pos,i+1];
88         elseif (pdif(i)<0) && (pdif(i+1)<0) && (pre(i)<0)
89             max_val=[max_val,Nusselt(i+1,2)];
90             max_pos=[max_pos,i+1];

```



```

88 end
89 else
90     flag=true;
91 end
92 i=1+i;
93 end
94 %busco el periodo de la velocidad impuesta
95 dat_f=friction(:,2);
96 pdif_f=diff(dat_f(:));
97 pre_f=diff(pdif_f(:));
98 flag_1=false;
99 i=1;
100 %mismo codigo que el anterior
101 while flag_1==false
102     if i < length(pdif_f) && i+1 < length(pdif_f)
103         if (pdif_f(i)<0) && (pdif_f(i+1)>0) && (pre_f(i)>0)
104             min_val_f=[min_val_f,friction(i+1,2)];
105             min_pos_f=[min_pos_f,i+1];
106         elseif (pdif_f(i)<0) && (pdif_f(i+1)<0) && (pre_f(i)<0)
107             max_val_f=[max_val_f,friction(i+1,2)];
108             max_pos_f=[max_pos_f,i+1];
109         end
110     else
111         flag_1=true;
112     end
113     i=1+i;
114 end
115 %segun el dato tengo que cambiar el minimo que saco de la funcion
116 T=(min_pos_f(end)-min_pos_f(length(min_pos_f)-1));
117 t_trat=friction(min_pos_f(length(min_pos_f)-1):min_pos_f(end),2);
118 min=min_pos_f(end);
119 min_1=min_pos_f(length(min_pos_f)-1);
120 min_Nuss_sup=min_pos(end);
121 min_Nuss_inf=min_pos(length(min_pos)-1);
122
123
124 end

```