

Trabajo Fin de Grado Grado en Ingeniería Aeroespacial

Estudio numérico de la deformación superficial de un chorro líquido impactando contra un objeto

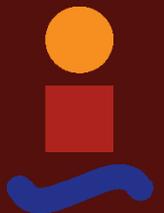
Autor: Francisco Javier Muñiz Gómez

Tutores: Pedro Ángel Vázquez González

Heliodoro González García

**Dpto. de Física Aplicada III
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2023



Trabajo Fin de Grado
Grado en Ingeniería Aeroespacial

Estudio numérico de la deformación superficial de un chorro líquido impactando contra un objeto

Autor:

Francisco Javier Muñoz Gómez

Tutores:

Pedro Ángel Vázquez González
Profesor Titular

Heliodoro González García
Profesor Titular

Dpto. de Física Aplicada III
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2023

Trabajo Fin de Grado: Estudio numérico de la deformación superficial de un chorro líquido impactando contra un objeto

Autor: Francisco Javier Muñoz Gómez

Tutores: Pedro Ángel Vázquez González
Heliodoro González García

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

A mi familia, que con su apoyo incondicional y esfuerzo incansable me han permitido superarme cada día, y a aquellos profesores, compañeros y amigos que me han hecho ser quien soy hoy.

Francisco Javier Muñiz Gómez
Sevilla, 2023

Resumen

Los estudios que se presentan en las siguientes páginas buscan introducir al lector en las técnicas y softwares más comunmente usados en la Mecánica de Fluidos Computacional y validar su capacidad para estudiar sistemas complejos de interés científico. De este modo, se presenta el software *openFOAM*, un software de uso libre de altas capacidades y numerosas utilidades que permitirá a cualquier usuario interesado introducirse en CFD (*Computational Fluid Mechanics*).

Con la ayuda de *openFOAM*, se obtendrán resultados numéricos de la deformación superficial de un chorro impactando contra un obstáculo, pudiéndose validar las predicciones lineales analíticas existentes basadas en los llamados modos espaciales y demostrando las utilidades de los softwares CFD. Además, se pretende mostrar las características principales de este sistema y recoger sus dependencias con variables de interés como el número de Weber o el número de Ohnesorge.

El presente trabajo también pretende servir como guía para que aquellos alumnos interesados en realizar simulaciones con *openFOAM* puedan dar sus primeros pasos con el software y entender sus principios y entorno, así como los fundamentos de CFD, principales técnicas y buenas prácticas. Es por ello que se detallará en las próximas páginas cómo realizar simulaciones con el software, cómo visualizar los resultados y cómo evaluar la fiabilidad de las simulaciones. De este modo, se establecerán las bases para que el usuario interesado pueda realizar simulaciones de forma autónoma con otros softwares de CFD más sofisticados y más comunmente usados en la industria tales como *Ansys Fluent* o *COMSOL Multiphysics*.

Abstract

The studies presented in the following pages aim to introduce the reader to those techniques and softwares most commonly used in Computational Fluid Mechanics and validate their capacity to study complex systems of scientific interest. Thus, the software *openFOAM* is presented, an open source software of high capabilities and many utilities that will allow any interested user to get introduced in CFD (Computational Fluid Mechanics).

Numerical results of the surface deformation of a jet impinging on an obstacle will be obtained with the aid of *openFOAM*, enabling the validation of the existing analytical linear predictions based on the so-called spatial modes, and demonstrating the utilities of CFD softwares. Also, it is intended to show the principal characteristics of the system and gather its dependencies with variables of interest such as Weber number or Ohnesorge number.

This report is also intended to serve as a guide for those students interested in performing simulations with *openFOAM* to be able to understand the software and its environment, as well as CFD fundamentals, most important techniques and good practices. For this reason, in the following pages it will be described how to perform simulations with the software, view results and how to evaluate the reliability of the simulations. Thus, the basis will be established for those interested users to perform simulations on more sophisticated and commonly used softwares such as *Ansys Fluent* or *COMSOL Multiphysics*.

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
1 Introducción	1
2 Descripción del Problema	3
2.1 Geometría e hipótesis de partida	3
2.2 Ecuaciones adimensionales	5
2.2.1 Condiciones de contorno	6
3 Modelo	9
3.1 Condiciones de contorno en <i>openFOAM</i>	9
3.2 <i>interFOAM</i> y algoritmo <i>PIMPLE</i>	10
3.3 Estudio de mallado	12
4 Entorno <i>Ubuntu</i> y <i>openFOAM</i>	19
4.1 Pasos previos: carpetas <i>0</i> , <i>system</i> y <i>constant</i>	19
4.1.1 Carpeta <i>0</i>	19
4.1.2 Carpeta <i>system</i>	24
4.1.3 Carpeta <i>constant</i>	31
4.2 Simulación en <i>openFOAM</i> : comandos básicos de <i>Ubuntu</i>	32
4.3 ParaView	34
4.3.1 Visualización de resultados	34
4.3.2 Obtención de resultados: puntos de la interfase	35
5 Análisis de resultados	37
5.1 Interfase agua-aire. Resultados para $Oh = 0.003$	37
5.2 Resultados para $Oh = 0.01$	41
6 Modelo teórico	47
7 Conclusiones y trabajos futuros	51
<i>Índice de Figuras</i>	53
<i>Bibliografía</i>	55

1 Introducción

En los últimos años, numerosas aplicaciones industriales y tecnológicas avanzadas tales como impresión por inyección, combustión interna en motores convencionales o poscombustión en cazas modernos han sufrido un desarrollo exponencial, requiriendo un elevado conocimiento en áreas de la Mecánica de Fluidos como son la atomización de chorros, formación de gotas o estabilidad de chorros, entre otras.

Basándose en los experimentos de Hancock & Bush, presentados en su publicación *Fluid Pipes* [9], este estudio presenta resultados numéricos sobre la deformación superficial que sufre un chorro líquido cuando impacta contra un objeto o un reservorio profundo de líquido. Estos resultados se obtendrán empleando el software de código libre *openFOAM*, desarrollado por la compañía CFD Direct, ampliamente usado en la industria por su gran número de bibliotecas y utilidades, que permiten simular gran cantidad de sistemas de forma eficiente y precisa. Con el fin de evaluar la validez del software y la exactitud de las simulaciones, los resultados obtenidos se contrastarán con aquellos obtenidos analíticamente por Guerrero et al., presentando el modelo teórico y el ajuste de los patrones de arrugas obtenidos con las simulaciones a los parámetros del modelo.

Numerosos son los estudios llevados a cabo en el ámbito de los fenómenos capilares, es decir, aquellos gobernados por la fuerzas de tensión superficial, remontándose las primeras publicaciones a principios del siglo XIX, con Young [15] y Laplace [5]. Años más tarde, Savart [13], en 1833, realizó una serie de experimentos cruciales sobre la formación de gotas, aunque no los asoció a la tensión superficial. Fue Plateau, en el año 1873, quien propuso el primer modelo que relacionaba la inestabilidad del chorro con la tensión superficial, abriendo el camino a los trabajos posteriores de Rayleigh (1892) [12], quien examinó analíticamente la inestabilidad de chorros líquidos en ausencia de la gravedad, demostrando de forma teórica que en aquellos casos donde la longitud de onda de las perturbaciones axisimétricas, λ , sean menores que la circunferencia del chorro, $2\pi R$, dichas perturbaciones serán estables, mientras que en aquellos casos donde $\lambda > 2\pi R$, las perturbaciones crecerán de forma exponencial y terminarán causando la rotura del chorro. Para el primer caso, si el chorro encuentra un obstáculo, se manifiestan perturbaciones que viajan aguas arriba, formando corrugaciones estables análogas a las que se observan cerca del hilo de una caña de pescar en un río que fluye laminarmente ("fishing line problem", Lord Rayleigh, *The Theory of Sound* [11]). Esta característica fue estudiada posteriormente por Anno (1977) [3], quien demostró analíticamente la existencia de un campo estable y estacionario de perturbaciones, completando el trabajo de Rayleigh y quedando validado su estudio por los experimentos de [9]. Con esto, en el presente trabajo se estudiará el caso cubierto por Anno, tratándose el sistema como un caso pseudo-estacionario, empleándose un estudio transitorio y eligiéndose un tiempo de simulación suficientemente largo como para que se alcancen las condiciones de equilibrio en las perturbaciones,

pudiendo así obtenerse los perfiles del chorro para diferentes números de Weber, apreciándose la evolución de las deformaciones con el cambio de las condiciones de partida.

El uso del software *openFOAM* se basa, principalmente, en su carácter de software libre, que permite que usuarios de todo el mundo puedan aportar novedosas soluciones e ideas que optimizan el funcionamiento de la herramienta y permiten utilizarla en un gran número de sistemas y casos. Por otro lado, es fácilmente adaptable gracias a su versatilidad y gran número de bibliotecas. Esto permitirá el estudio del sistema propuesto debido a la existencia de *solvers (interFOAM)* y condiciones de contorno (*freestreamPressure*) no existentes en otros programas de CFD, que logran reproducir con gran precisión los resultados observados experimentalmente. En las próximas páginas se explicará con mayor detalle las características de los elementos anteriores y los modelos empleados por el software para obtener los resultados.

A pesar de su flexibilidad y potencia, *openFOAM* requiere de un conocimiento básico en *Linux* y *C++*. Para poder lanzar las simulaciones y crear los mallados es necesario emplear el terminal de *Ubuntu* ya que *openFOAM* carece de interfaz, mientras que para modificar los parámetros de la simulación y del modelo se requiere modificar archivos de texto escritos en *C++*. Esto difiere de programas de CFD comerciales como *Ansys Fluent* o *Simcenter Star-CCM+*, que permiten al usuario llevar a cabo sus simulaciones a través de interfaces interactivas.

Por otra parte, el software empleado para la visualización de los resultados arrojados por las simulaciones será *ParaView*, de nuevo un software de código libre usado habitualmente para el post-procesado y representación de mallados y resultados de simulaciones llevadas a cabo con FEM.

2 Descripción del Problema

En las siguientes líneas se establecerán las ecuaciones que gobiernan el sistema de estudio, obtenidas a partir de la adaptación de las Ecuaciones de Navier-Stokes a las hipótesis de partida del modelo, así como las ecuaciones adimensionales que se implementarán en el estudio numérico. Para ello, será necesario detallar las condiciones en las que se dan las perturbaciones en la superficie del chorro, así como la geometría general del sistema, que será fundamental para desarrollar las ecuaciones adimensionales.

2.1 Geometría e hipótesis de partida

La geometría del problema se definirá a partir de los experimentos llevados a cabo en [9], usando como referencia la figura 2.1. Puede observarse la simetría del problema, presentando tanto el chorro como las corrugaciones que se dan en su base una geometría axisimétrica. Este aspecto será fundamental ya que permitirá llevar a cabo el estudio numérico mediante un modelo 2D axisimétrico, reduciendo así el costo computacional y el tiempo empleado en las simulaciones.

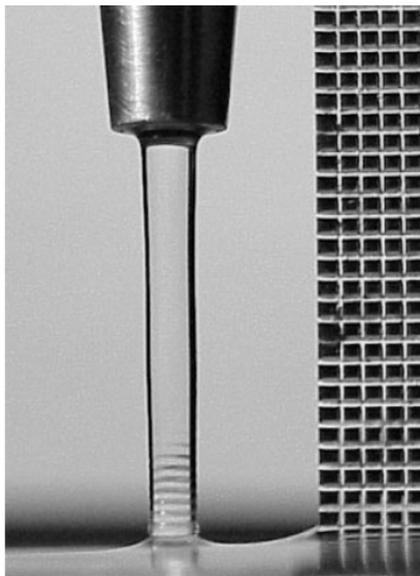


Figura 2.1 Experimento llevado a cabo en [9] en el que se basarán la geometría y el modelo a implementar en el estudio numérico.

Bajo estas premisas, se propone la geometría a implementar en el modelo del presente estudio en la siguiente figura:

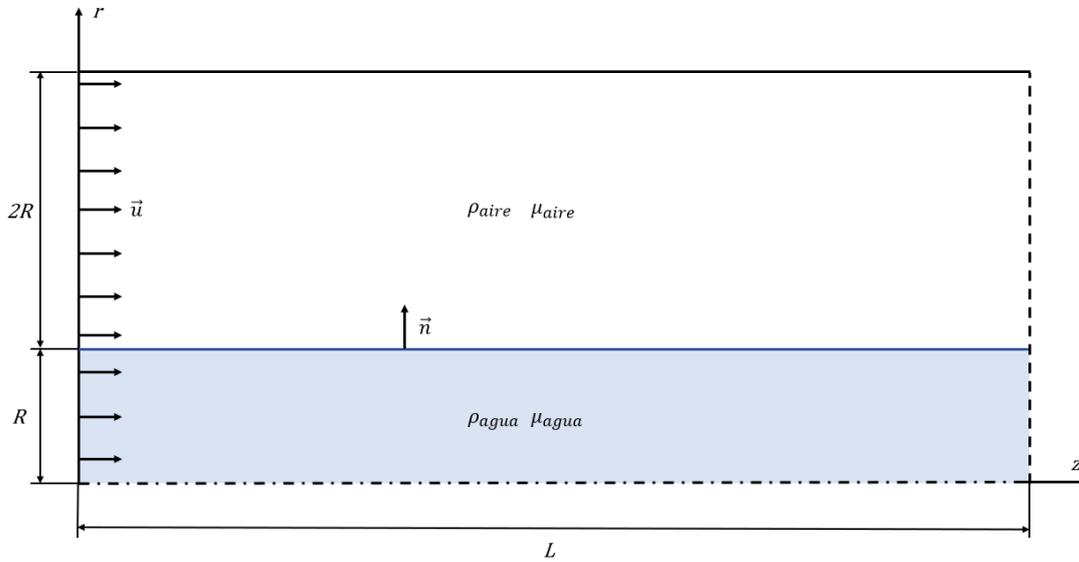


Figura 2.2 Esquemático de la geometría del modelo.

Se considerará un problema axisimétrico en el plano ORZ (semiplano de simetría), donde el eje z será el eje de simetría y el eje r la entrada tanto de aire como de agua. La dimensión R representa el radio inicial del chorro, el vector \vec{u} la velocidad axial en la entrada, que será uniforme y que tomará el valor de \sqrt{We} como veremos posteriormente (donde We es el número de Weber) y \vec{n} el vector normal a la superficie del chorro. Las cantidades ρ_{agua} y ρ_{aire} representan la densidad del agua y aire, respectivamente, mientras que μ_{agua} y μ_{aire} representan la viscosidad dinámica del agua y aire. Las condiciones de contorno y otras consideraciones sobre el modelo se detallarán en posteriores secciones.

Previa formulación de las ecuaciones que gobernarán el sistema, es necesario establecer las hipótesis de partida. En primer lugar, se considera que el chorro fluye libremente, considerando que el aire tiene una inercia despreciable que no tiene efecto apreciable sobre el desarrollo del chorro. Tal y como se expone en [6] y [8], esta hipótesis es válida cuando la velocidad relativa entre el chorro y el gas es pequeña, condición que se da en el sistema en estudio al considerarse que tanto el aire como el agua tienen la misma velocidad inicial.

Por otro lado, se desprecian aspectos energéticos al considerarse el sistema en condiciones ambiente constantes, de modo que la ecuación de la energía no tendrá que ser tenida en cuenta en el estudio. Además, tal y como se describe en [9], la interfaz agua-aire está claramente diferenciada, por lo que se puede considerar que ambos fluidos son inmiscibles. Estas dos consideraciones permiten afirmar que las viscosidades dinámicas del agua y aire serán constantes, teniendo en cuenta que la viscosidad del agua no sufrirá cambios apreciables debidos a la presión en el rango de presiones que se estudian en este trabajo ($p \sim 100$ kPa) [14], y que los cambios que puedan darse en la viscosidad del aire serán despreciables al ser $\mu_{aire} \sim (10^{-5} \text{ kg m}^{-1} \text{ s}^{-1})$.

Por último, en base a los experimentos llevados a cabo en [9] y debido al carácter de flujo a superficie libre del chorro, se puede afirmar que tanto la densidad del agua y del aire se mantendrán constantes.

Con esto, las ecuaciones de Navier-Stokes se reducen a las ecuaciones de continuidad y de conservación de la cantidad de movimiento:

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

$$\rho \frac{\partial \vec{u}}{\partial t} + \rho \vec{u} \cdot \nabla \vec{u} = -\nabla p + \mu \nabla^2 \vec{u} + \gamma \kappa \delta_s \vec{n} \quad (2.2)$$

Donde la expresión $\gamma \kappa \delta_s \vec{n}$ representa la fuerza capilar, siendo γ la tensión superficial entre agua-aire (toma el valor de 0.073 N m^{-1} en condiciones ambiente), δ_s la delta de Dirac, formulada de manera que la fuerza capilar se tenga en cuenta únicamente en la interfase agua-aire, donde los efectos de tensión superficial son aplicables, y κ la curvatura de la superficie del chorro, dada por la siguiente expresión:

$$\kappa = \nabla \cdot \vec{n} \quad (2.3)$$

2.2 Ecuaciones adimensionales

Con el fin de implementar de manera eficiente y general el modelo en el software de CFD, es conveniente adimensionalizar las ecuaciones con las variables representativas del sistema. Con esto, se propone introducir las siguientes variables adimensionales:

$$\begin{aligned} \bar{r} &= \frac{r}{R} & \bar{z} &= \frac{z}{R} \\ \bar{u} &= \frac{u}{u_c} & u_c &= \sqrt{\frac{\gamma}{\rho R}} \\ \bar{p} &= \frac{p}{p_c} & p_c &= \frac{\gamma}{R} \end{aligned}$$

$$T_c = \sqrt{\frac{\rho R^3}{\gamma}}$$

Donde \bar{r} e \bar{z} representan las coordenadas del plano *ORZ* adimensionales, \bar{u} y \bar{p} la velocidad y la presión adimensionales, respectivamente, u_c la velocidad capilar, p_c la presión capilar y T_c el tiempo capilar. La consideración de estas variables adimensionales en las ecuaciones darán lugar a la aparición de dos números adimensionales: el número de Weber y el número de Ohnesorge:

$$\begin{aligned} We &= \frac{\rho u^2 R}{\gamma} \\ Oh &= \frac{\mu}{\sqrt{\rho \gamma R}} = \frac{\sqrt{We}}{Re} \end{aligned}$$

Donde *Re* es el número de Reynolds, que tiene la siguiente expresión:

$$Re = \frac{\rho u R}{\mu}$$

Estos números adimensionales serán fundamentales en el estudio del sistema presentado y tienen significados físicos que merecen ser mencionados. El número de Weber, tal y como se ha definido, representa el ratio de la velocidad de salida del chorro con la velocidad capilar, es decir, representa

la importancia de la inercia del chorro con respecto a la capilaridad. Por otro lado, el número de Ohnesorge, construido a partir de la velocidad capilar en este caso, representa la importancia de la viscosidad en el sistema. Nótese que únicamente depende de la viscosidad del fluido y de la geometría del modelo, no interviniendo la velocidad del chorro.

Con esto, la ecuación de continuidad adimensional resulta:

$$\nabla \cdot \vec{u} = 0 \quad (2.4)$$

Mientras que la ecuación de conservación de la cantidad de movimiento adimensional se obtiene mediante las siguientes transformaciones, partiendo de la ecuación (2.2):

$$\rho_1 \bar{\rho} \frac{u_c}{T_c} \frac{\partial \vec{u}}{\partial \bar{t}} + \rho_1 \bar{\rho} \frac{u_c^2}{R} \vec{u} \cdot \nabla \vec{u} = -\frac{p_c}{R} \nabla \bar{p} + \frac{\mu}{R^2} u_c \nabla^2 \vec{u} + \frac{\gamma}{R^2} \bar{\kappa} \bar{\delta}_s \vec{n}$$

Donde $\rho_1 = \rho_{agua}$. Multiplicando la ecuación por $\frac{u_c}{T_c \rho_1}$, obtenemos:

$$\bar{\rho} \frac{\partial \vec{u}}{\partial \bar{t}} + \underbrace{\frac{\rho_1 u_c T_c}{R u_c \rho_1}}_{=1} \vec{u} \cdot \nabla \vec{u} = -\underbrace{\frac{p_c T_c}{R u_c \rho_1}}_{=1} \nabla \bar{p} + \underbrace{\frac{\mu u_c T_c}{R^2 \rho_w u_c}}_{=Oh} \nabla^2 \vec{u} + \underbrace{\frac{\gamma T_c}{R^2 \rho_1 u_c}}_{=1} \bar{\kappa} \bar{\delta}_s \vec{n}$$

Resultando en:

$$\bar{\rho} \frac{\partial \vec{u}}{\partial \bar{t}} + \vec{u} \cdot \nabla \vec{u} = -\nabla \bar{p} + Oh \nabla^2 \vec{u} + \bar{\kappa} \bar{\delta}_s \vec{n} \quad (2.5)$$

Por simplicidad, en las próximas líneas se escribirán las variables adimensionales como u , t , p , ρ , κ , \vec{n} y δ_s , indistintivamente. Con esto, las ecuaciones de continuidad y cantidad de movimiento adimensionales resultan:

$$\nabla \cdot \vec{u} = 0 \quad (2.6)$$

$$\rho \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\nabla p + Oh \nabla^2 u + \kappa \delta_s \vec{n} \quad (2.7)$$

Es importante tener en cuenta que los valores de ρ y Oh son diferentes en función de la fase en la que se encuentre el punto \vec{x} . De este modo, siendo f la fracción volumétrica de agua, se tiene que:

$$\rho = \begin{cases} 1 & \text{si } f = 1 \\ \frac{\rho_1}{\rho_2} & \text{si } f = 0 \end{cases} \quad Oh = \begin{cases} \frac{\mu_1}{\sqrt{\rho_1 \gamma R}} & \text{si } f = 1 \\ \frac{\mu_2}{\sqrt{\rho_1 \gamma R}} & \text{si } f = 0 \end{cases}$$

Donde $\rho_2 = \rho_{aire}$ y $\mu_2 = \mu_{aire}$. Por su parte, la fracción volumétrica del agua f tiene su propia ecuación, dada por el modelo VOF. Esta ecuación y el modelo se presentan de forma detallada en la siguiente sección.

2.2.1 Condiciones de contorno

Para escribir las condiciones de contorno se usará la geometría propuesta en la figura 2.2. En forma dimensional, las condiciones de contorno de entrada, pared, y salida, usando como referencia [6] y [7] son:

$$z = 0 \rightarrow u = cte \quad (2.8)$$

$$r = 3R \rightarrow \vec{u} \cdot \vec{n} = 0 \quad (2.9)$$

$$z = L \rightarrow freeStreamPressure \quad (2.10)$$

Donde la condición de contorno (2.8) representa la condición de velocidad uniforme tanto para el aire como para el agua a la entrada, en la dirección del eje z , la condición (2.9) representa la condición de impenetrabilidad en la pared, y la condición (2.10) es la condición *freeStreamPressure* de *openFOAM*, que se explicará detalladamente posteriormente. Estas dos últimas condiciones de contorno son condiciones artificiales que se han establecido para poder estudiar el problema numéricamente y que no tienen por qué describir los experimentos de [9]. Esto permitirá estudiar el obstáculo más simple que permite formular el programa y obtener resultados de un sistema análogo al real.

Las condiciones de contorno dimensionales del trazador o fracción volumétrica del agua f son:

$$z = 0 \rightarrow f = \begin{cases} 1, & \text{si } r \leq R \\ 0, & \text{si } r > R \end{cases} \quad (2.11)$$

$$r = 3R \rightarrow \frac{\partial f}{\partial r} = 0 \quad (2.12)$$

$$z = L \rightarrow \frac{\partial f}{\partial z} = 0 \quad (2.13)$$

donde la condición (2.11) establece que, inicialmente, para $r \leq R$ únicamente habrá agua, mientras que para $r > R$ habrá aire. Habiendo adimensionalizado las ecuaciones en la sección anterior, es conveniente adimensionalizar también las condiciones de contorno se que se emplearán para resolverlas. Estas condiciones de contorno son las que se implementarán en el estudio numérico y son las siguientes:

$$\bar{z} = 0 \rightarrow \bar{u} = \sqrt{We} = cte \quad (2.14)$$

$$\bar{r} = 3 \rightarrow \vec{\bar{u}} \cdot \vec{\bar{n}} = 0 \quad (2.15)$$

$$\bar{z} = 1 \rightarrow freeStreamPressure \quad (2.16)$$

Por último, mencionar las condiciones de contorno adimensionales del trazador, que también deberán ser implementadas en las simulaciones numéricas:

$$\bar{z} = 0 \rightarrow f = \begin{cases} 1, & \text{si } \bar{r} \leq 1 \\ 0, & \text{si } \bar{r} > 1 \end{cases} \quad (2.17)$$

$$\bar{r} = 3 \rightarrow \frac{\partial f}{\partial \bar{r}} = 0 \quad (2.18)$$

$$\bar{z} = 1 \rightarrow \frac{\partial f}{\partial \bar{z}} = 0 \quad (2.19)$$

3 Modelo

Con el fin de entender cómo obtiene *openFOAM* los resultados que se mostrarán en las siguientes secciones, es necesario conocer de antemano los modelos y los métodos que se emplearán para realizar las simulaciones, así como la adecuación de los mismos con el sistema a estudiar. También, es preciso tener en cuenta qué condiciones de contorno se eligen para el problema, pues son fundamentales en los resultados que se obtendrán con las simulaciones; un pequeño cambio en las condiciones de contorno de salida puede provocar que el sistema bajo estudio sea totalmente distinto al sistema objetivo, obteniéndose así resultados totalmente distintos a los obtenidos experimentalmente.

3.1 Condiciones de contorno en *openFOAM*

Una vez implementada la geometría y los parámetros del problema adimensionales en *openFOAM*, es necesario establecer en el software las condiciones de contorno adecuadas que cumplan con las condiciones (2.8) - (2.10).

La condición (2.8) implica que la velocidad de entrada tanto para el aire como para el agua es uniforme e igual a \sqrt{We} . Para cumplir con esta condición, se implementa en el software la condición de contorno *fixedValue* para la velocidad, que establece un valor fijo para la velocidad de entrada tanto para el aire como para el agua (*inletWater* e *inletAir*, respectivamente). El valor fijo establecido será de \sqrt{We} , donde el número de Weber variará según los parámetros del chorro a estudiar (velocidad del chorro y altura del chorro sobre la superficie).

Por otro lado, la condición de impenetrabilidad en la pared (2.9) se cumplirá aplicando la condición de contorno en velocidades *slip*, que elimina la componente normal de la velocidad y permite que la componente tangencial tenga un valor no nulo. Esto permitirá que el aire, fase que está en contacto con la pared superior, pueda tener un movimiento uniforme en la dirección axial.

Por último, la condición de salida (2.10) se cumplirá imponiendo la condición de contorno en presión *freestreamPressure*, que obliga a la presión del flujo a la salida a tener un gradiente nulo, mientras que si el flujo a la salida intenta entrar de nuevo en el dominio fluido, la presión tomará un valor determinado, en este caso establecido como nulo. Además, también se establece la condición de contorno en velocidad de *zeroGradient*, que obliga a la velocidad del flujo de salida a tener un gradiente nulo en sus tres componentes.

La implementación de estas condiciones de contorno en los archivos U y p_rgh de la carpeta 0 en los que se declaran las condiciones iniciales del sistema, permitirán obtener los patrones de arrugas formados cuando el chorro impacta contra un obstáculo cualquiera, pudiendo así identificar los modos espaciales que intervienen. Con el fin de poder analizar los resultados y poder extrapolarlos a los obtenidos experimentalmente en [9], se considera que los modos que forman el patrón de arrugas son los mismos para cualquier tipo de obstáculo, ya sea una piscina o una pared.

3.2 *interFOAM* y algoritmo *PIMPLE*

El modelo se basará en considerar al sistema en estudio como un sistema multifase conformado por dos fases inmiscibles (agua y aire) separados por una interfase diferenciada, donde los fluidos se tratarán como incompresibles y el sistema será isotermo. Estas condiciones se recogen en el *solver interFOAM*, que permitirá resolver las ecuaciones de Navier-Stokes empleando el método del volumen de fluido (VOF), obteniendo resultados precisos en la interfase del sistema, pudiendo así representar los fenómenos bajo estudio en este trabajo de forma precisa.

El método VOF fue introducido por Hirt & Nichols [10] en 1981 y supuso un cambio de paradigma en el estudio numérico de sistemas multifase. El método se basa en la concepción de una función rastreadora que identifica aquellos elementos del mallado que están ocupados por una fase u otra. Esto permite definir con precisión la interfase del sistema y resolver las ecuaciones de Navier-Stokes para cada fase. Las ecuaciones que se emplearán en este método son:

$$\nabla \cdot \vec{u} = 0 \quad (3.1)$$

$$\rho \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\nabla p + Oh \nabla^2 u + \kappa \delta_s \vec{n} \quad (3.2)$$

$$\frac{\partial \gamma}{\partial t} + \nabla \cdot (u\gamma) = 0 \quad (3.3)$$

Donde γ , que puede tomar valores entre 0 y 1 ($\gamma = 0$ se corresponde con la región donde hay aire y $\gamma = 1$ se corresponde con la región donde hay agua), representa la fracción de volumen de las fases, siendo la ecuación (3.3) la ecuación de transporte de la interfase, que debe resolverse junto a las ecuaciones de Navier-Stokes para conocer la posición y el estado de la interfase en cada momento.

Para resolver estas ecuaciones, *openFOAM* hace uso del algoritmo *PIMPLE*, una combinación de los algoritmos *SIMPLE* y *PISO*. En este algoritmo, la presión y la velocidad están acopladas y se resuelven en cada paso de tiempo (Δt) hasta que alcanzan la convergencia deseada, después de un número determinado de iteraciones de corrección de los términos. Una vez se obtienen los resultados deseados de presión y velocidad, el algoritmo toma estos valores y resuelve de nuevo las ecuaciones, obteniendo los resultados definitivos. Tras esto, se calcula el nuevo paso de tiempo a implementar en el algoritmo y se repite el proceso. Se muestra en la figura 3.1 un diagrama de flujo del algoritmo a modo de resumen.

Un aspecto importante a tener en cuenta a la hora de elegir el *solver* y ejecutar la simulación es la estabilidad, que puede medirse de forma cuantitativa mediante el número de Courant (Co), definido como:

$$Co = \frac{U\Delta t}{\Delta d} \quad (3.4)$$

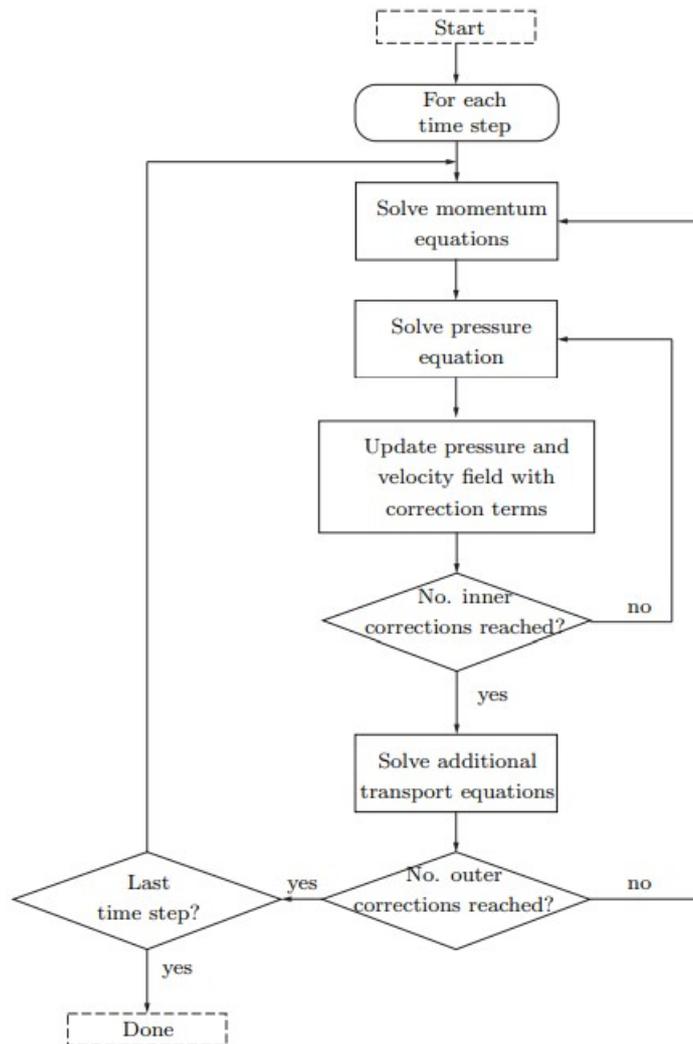


Figura 3.1 Diagrama de flujo del algoritmo *PIMPLE*.

Donde U es la velocidad local del elemento (en un punto o cara), Δt es el paso de tiempo y Δd es el tamaño característico (unidimensional) del elemento de mallado. Este número adimensional mide el grado de convergencia de ecuaciones diferenciales en derivadas parciales tales como las del método VOF, y marca un límite superior en el paso de tiempo de la simulación. Si el valor del número de Courant es superior a la unidad, la simulación puede volverse inestable y dar resultados incorrectos, por lo que es aconsejable asegurar que se obtiene un número de Courant máximo en las iteraciones menor que 1.

Cabe destacar que, a diferencia del algoritmo *PISO*, el algoritmo *PIMPLE* permite realizar simulaciones transitorias con números de Courant mucho mayores que 1, dando resultados estables sin necesidad de limitar los factores de relajación (en inglés, *under-relaxation factor*), es decir, sin cambiar manualmente cuánto puede cambiar una variable de una iteración a otra. No obstante, en este estudio no se hará uso de esa característica y se establecerá un número de Courant máximo de 0.5 para hacer que la simulación sea lo más estable posible para un tiempo de computación aceptable. Esto se conseguirá haciendo uso de una característica que incorpora *interFOAM*, que es la capacidad de adoptar un paso de tiempo adaptativo, calculado a partir del número de Courant

obtenido en la iteración anterior y de parámetros de entrada definidos por el usuario, tomando siempre 0.5 como número de Courant máximo permitido. De este modo, el paso de tiempo de cada iteración es el máximo posible para las imposiciones establecidas, permitiendo una simulación eficiente y fiable. Más información sobre esta característica de *interFOAM* puede encontrarse en la publicación de Márquez Damián [4].

Las simulaciones se realizarán con $t_{inicial} = 0$ y $t_{final} = 80$ (no se indican las unidades de t al haberse adimensionalizado el problema), momento en el que se alcanza el estado pseudo-estacionario buscado en la mayoría de casos, siendo el paso de tiempo dependiente del número de Weber elegido para cada simulación, pues la estabilidad (número de Courant), dependerá de las condiciones del sistema.

3.3 Estudio de mallado

Con el fin de elegir el mallado con el número mínimo de elementos que permite obtener resultados fiables para el menor costo computacional, así como asegurar que el mallado elegido no influye en los resultados obtenidos, se realiza en esta sección un estudio de malla en el que se comparan los resultados obtenidos en distintas simulaciones llevadas a cabo con mallados que, teniendo la misma estructura, tienen números de elementos apreciablemente diferentes. En las gráficas que se muestran a continuación se puede apreciar cómo las soluciones obtenidas dependen en gran medida del mallado escogido. Evitar este fenómeno es fundamental para garantizar la validez de los resultados.

Las simulaciones se llevarán a cabo para $We = 15$, valor del número de Weber que hace que la longitud de onda de las arrugas disminuya con respecto a los casos que se estudiarán en las siguientes secciones. Este fenómeno se aprecia a medida que aumenta el número de Weber, de forma que será necesario un mallado más fino para poder recoger el comportamiento del chorro, haciendo que el número de Weber tenga un efecto desfavorable en la simulación. Imponiendo el valor de $We = 15$ se establece uno de los casos más desfavorables de estudio y será útil para evitar problemas de convergencia en futuras simulaciones.

En la siguiente tabla se muestran las características de los diferentes mallados elegidos para llevar a cabo el estudio de malla, así como el tiempo empleado para cada simulación:

Tabla 3.1 Características principales de los mallados propuestos y tiempos de simulación.

ID Mallado	Nº de elementos	A_{min} elemento	AR_{max}	$T_{simulacion}$
1	16000	4.3624e-5	3.2253	1 h 19 min
2	28000	1.0906e-5	6.4266	3 h 2 min
3	44800	1.0906e-5	4.0162	6 h 23 min
4	63900	6.3309e-6	4.6812	12 h 36 min
5	83000	4.4670e-6	5.0132	16 h 49 min
6	134400	2.72648e-6	5.34454	36 h 25 min

Donde AR_{max} es la máxima relación de aspecto entre elementos, es decir, el cociente entre la superficie máxima y la superficie mínima de los elementos del mallado. A modo ilustrativo, se muestra en la figura 3.2 una comparativa entre el mallado más grueso y el mallado más fino.

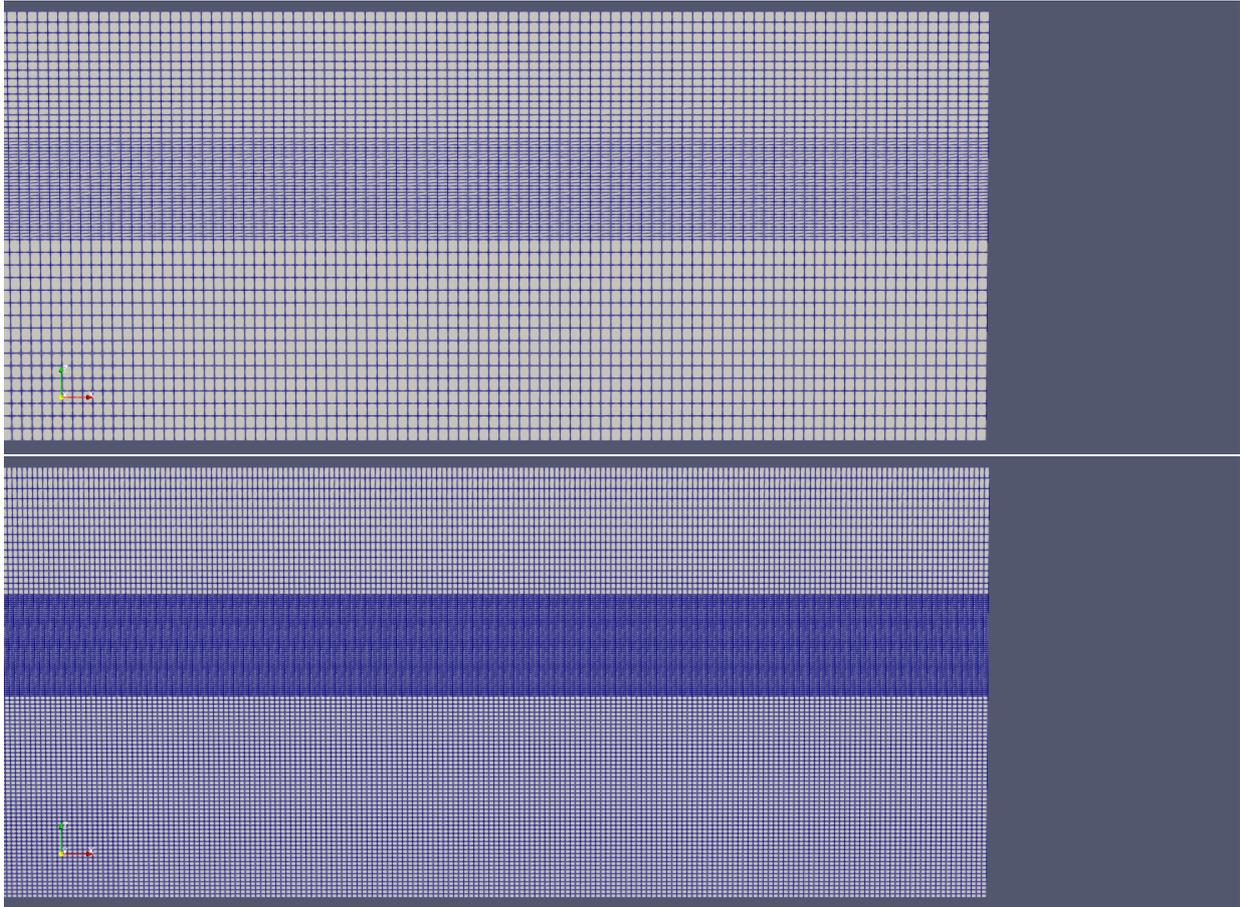


Figura 3.2 Comparativa entre las mallas 1 y 5.

Como se comentaba, es bien sabido que los resultados dependen de la malla elegida, aunque, además, con este estudio de mallado se ha hecho evidente una clara dependencia de los resultados con el número de Courant, volviéndose esta dependencia menos relevante para mallados más finos. De este modo, aún siendo las simulaciones estables ($Co < 1$), para los mallados más gruesos se han observado notables diferencias en las simulaciones llevadas a cabo con paso de tiempo adaptativo (fijado $Co = 0.5$) y paso de tiempo fijo de $0.005 s$, haciéndose relevante la necesidad de establecer el método para determinar el paso de tiempo que se empleará en cada iteración. En la figura 3.3 se hacen visibles estas diferencias para los mallados 3 y 5, siendo más notables en el mallado 3 al ser un mallado más grueso, especialmente cerca de la salida del chorro. Esto implica que los resultados dependen del paso de tiempo elegido y, por tanto, del número de Courant en cada iteración, por lo que se deberá buscar aquel mallado óptimo que proporcione los mejores resultados para el mayor paso de tiempo posible.

Para poder elegir el método a seguir para establecer el paso de tiempo en cada iteración, se compararán los resultados obtenidos con los mallados 3 y 5 con paso de tiempo adaptativo y paso de tiempo fijo con una simulación realizada con paso de tiempo adaptativo y un mallado muy fino (malla 6) de 134400 elementos y superficie mínima del elemento $2.72648e-6$, que asegura unos resultados fiables en los que basar la decisión. En la figura 3.4 se comparan ambos casos, pudiéndose observar cómo el paso de tiempo adaptativo es el método que mejores resultados proporciona, especialmente en la zona de salida del chorro. Por su parte, el paso de tiempo fijo hace que los resultados difieran notablemente, aunque también supone un menor tiempo de simulación, pues, de forma general, el paso de tiempo adaptativo suele ser dos veces más pequeño que el paso de tiempo

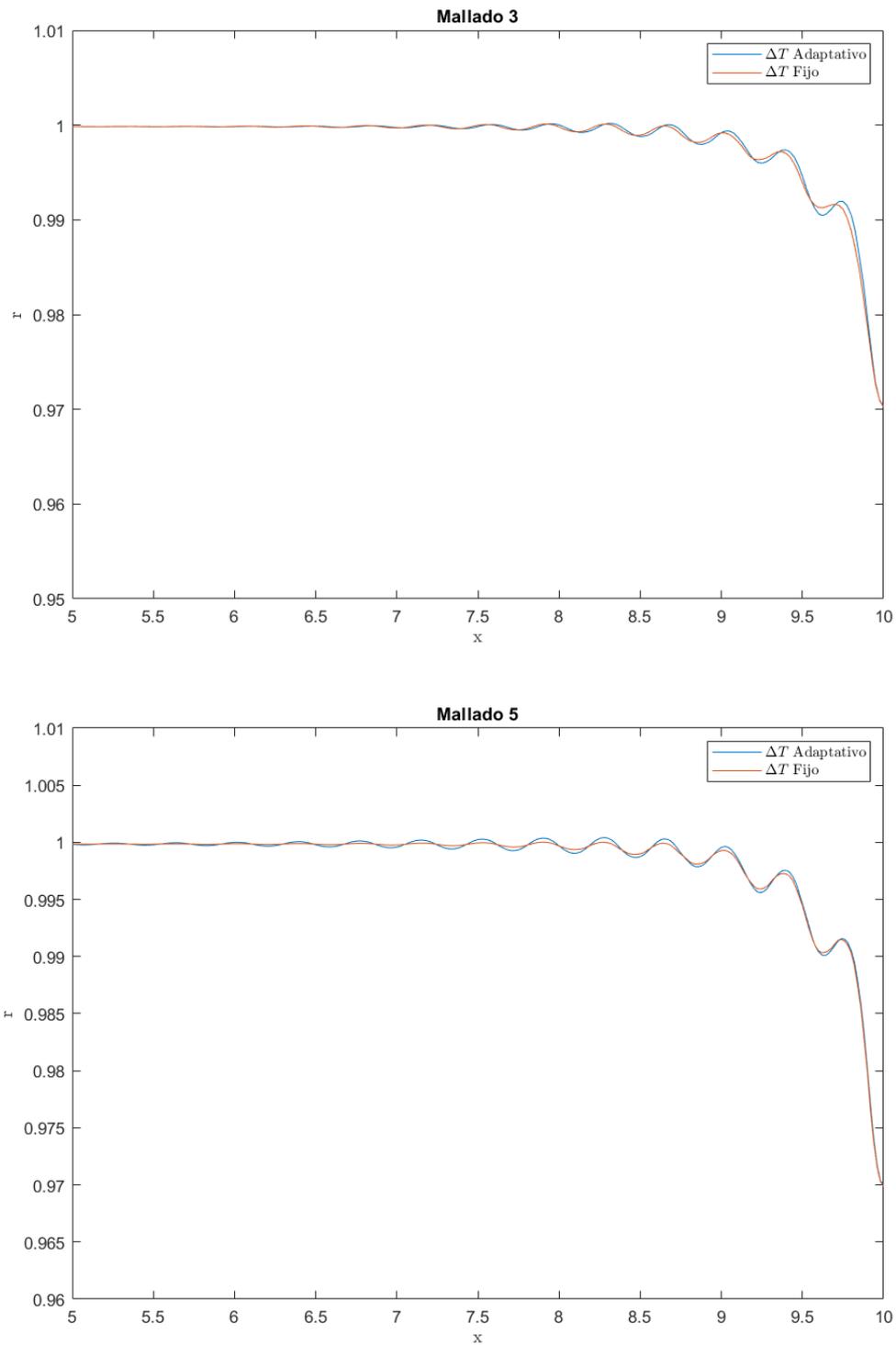


Figura 3.3 Interfases obtenidas para Δt adaptativo y Δt fijo, fijados $We = 15$ y $Oh = 0.003$.

fijo ($\Delta t_{fijo} = 0.005$).

Teniendo en cuenta estas conclusiones, se ha optado por realizar las simulaciones con paso de tiempo adaptativo y $Co_{max} = 0.5$ para asegurar la estabilidad de las mismas y la fiabilidad de los resultados.

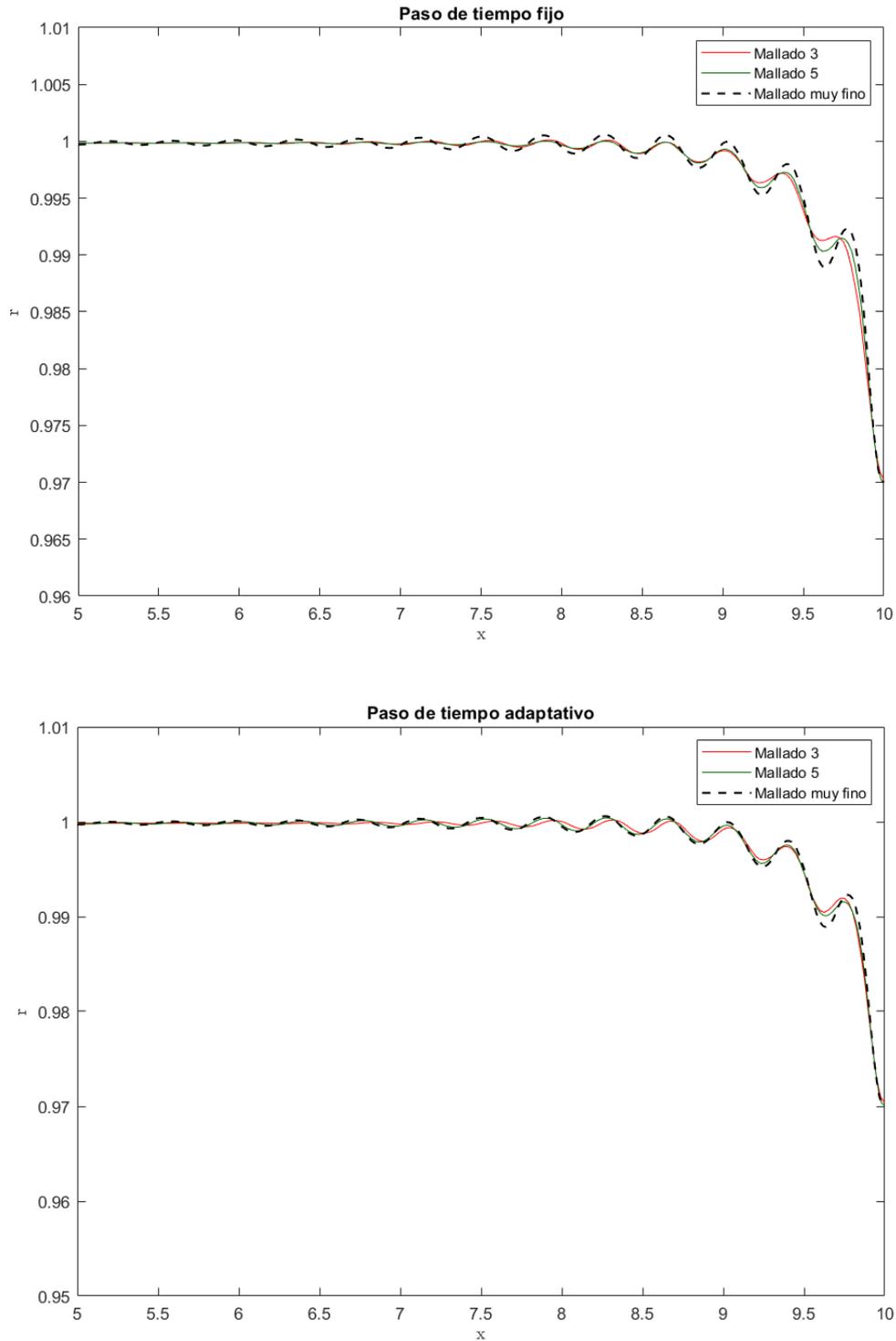


Figura 3.4 Resultados obtenidos con Δt fijo y Δt adaptativo frente a resultados obtenidos con un mallado muy fino para $We = 15$ y $Oh = 0.003$.

Por otro lado, volviendo a la finalidad principal del estudio de mallado, se debe elegir el mallado más grueso posible que proporcione resultados fiables, de forma que la simulación requiera el mínimo coste computacional permitido sin dejar de lado la precisión. Para ello, se comparan

los resultados obtenidos con los mallados de la tabla 3.1, aplicando paso de tiempo adaptativo y tomando como referencia los resultados obtenidos con el mallado muy fino que se ha presentado en las anteriores figuras. En la figura 3.5 puede observarse cómo los mallados más gruesos (1 y 2) ni siquiera llegan a reproducir el comportamiento esperado, por lo que es necesario aumentar el número de elementos para poder considerar la solución como válida y los resultados tengan un mínimo de rigor.

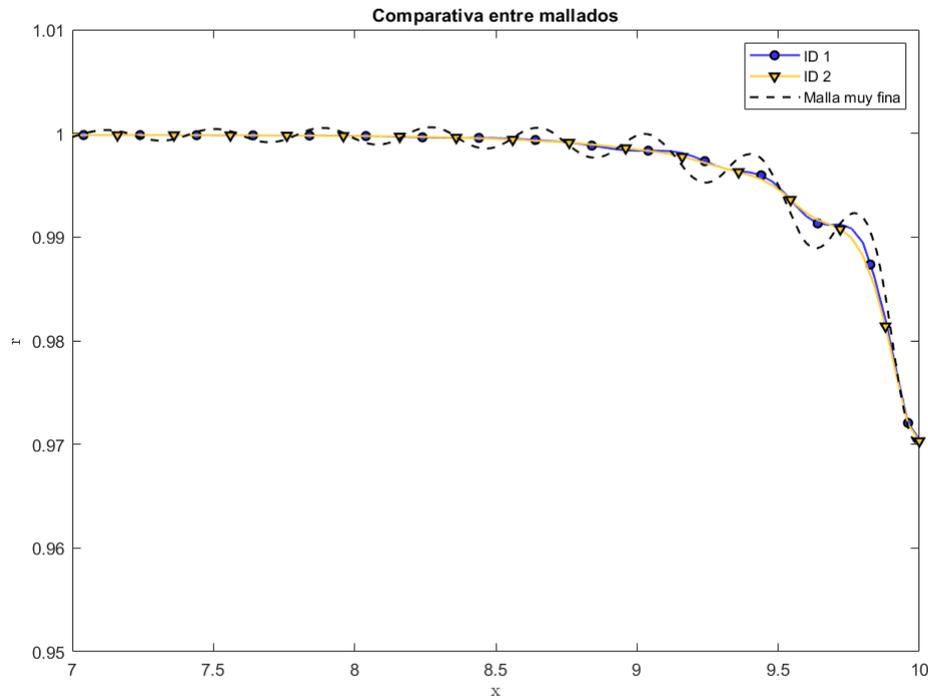


Figura 3.5 Resultados obtenidos con los mallados 1 y 2 frente a los obtenidos para un mallado muy fino de 134400 elementos para $We = 15$ y $Oh = 0.003$.

Aumentando el número de elementos puede observarse cómo los resultados mejoran notablemente, dejándose de apreciar la dependencia de los resultados con la malla. En la figura 3.6 se presentan los resultados obtenidos con los mallados medios y fino (3, 4 y 5) frente a los obtenidos para el mallado muy fino. En este punto se vuelve complejo notar las diferencias entre las mallas, especialmente entre las mallas 4 y 5 que, a pesar de tener tiempos de simulación notablemente distintos según la tabla 3.1 (la simulación realizada con el mallado 4 puede llegar a tardar cuatro horas menos que la realizada con el mallado 5), proporcionan unos resultados muy similares que se asemejan con gran exactitud a los obtenidos con el mallado muy fino.

La dependencia de los resultados con la malla puede verse también de forma clara mediante una comparativa entre las diferencias relativas medias entre los puntos de la interfase para los mallados propuestos. Se define dicha diferencia relativa media como:

$$\varepsilon = \sum_{i=1}^N \frac{|r_{i+1} - r_i|}{N} \quad (3.5)$$

donde r es la coordenada radial del chorro (que se corresponde con la coordenada z de *openFOAM*) y N es el número de nodos en la dirección x , es decir, es el número de nodos en los que se calcula la

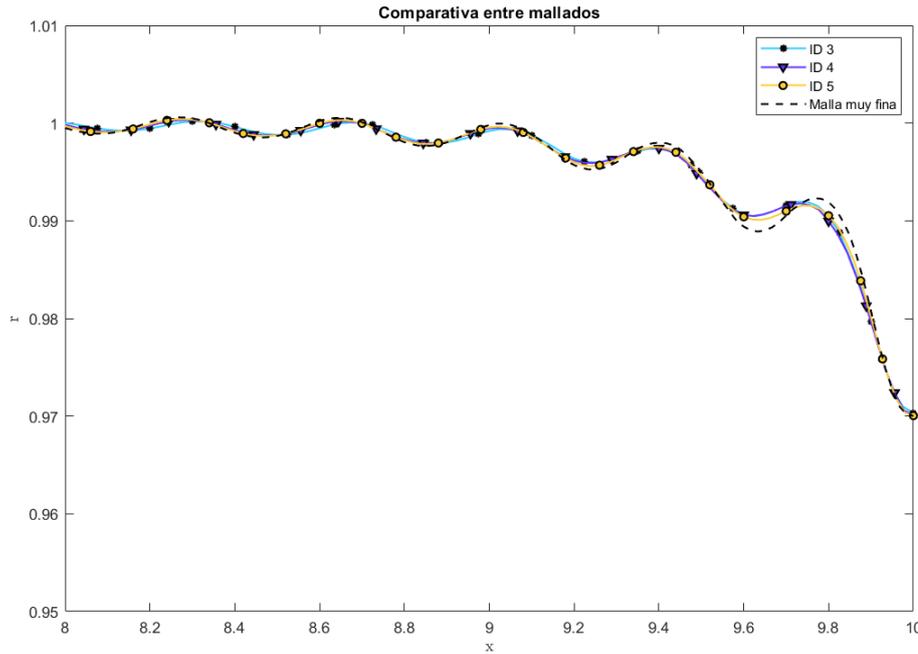


Figura 3.6 Resultados obtenidos con los mados 3, 4 y 5 frente a los obtenidos para un mado muy fino de 134400 elementos para $We = 15$ y $Oh = 0.003$.

interfase para cada mado. De este modo, representando $\log_{10}(\varepsilon)$ frente al número de puntos para los diferentes mados, se obtiene:

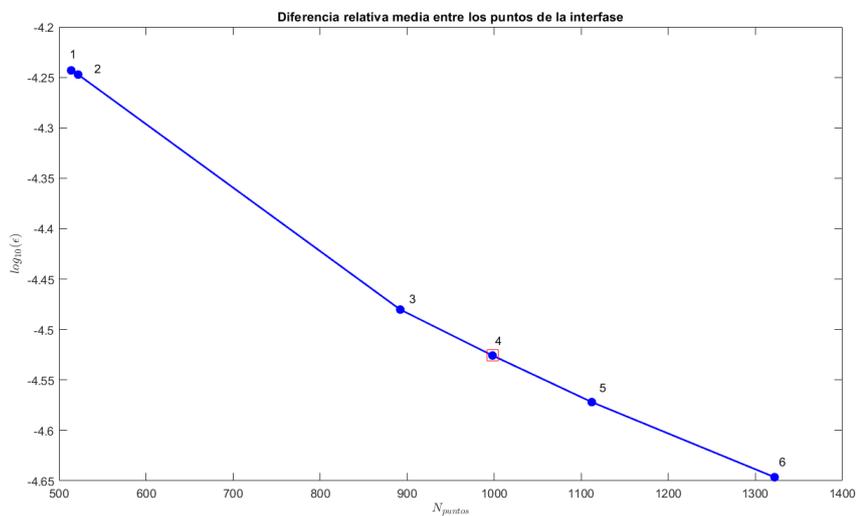


Figura 3.7 Diferencias relativas medias entre los puntos de la interfase obtenidas con los mados propuestos..

Nótese cómo las diferencias relativas medias para los mados gruesos (1 y 2) son notablemente diferentes a aquellas obtenidas para los otros mados, que sí modelan correctamente el comportamiento del chorro, indicando de nuevo que estos mados no son lo suficientemente buenos como para ser elegidos en este estudio. No obstante, para los otros mados se observa una convergencia

lineal hacia una diferencia relativa media constante e independiente del mallado, que estará en torno a $\log_{10}(\varepsilon) \approx -4.65$, lo que indica que los resultados obtenidos con el mallado 4 pueden ser considerados como aceptables ya que el error relativo entre los valores de ε obtenidos para este mallado y el mallado muy fino es menor al 2%, considerándose este un error aceptable teniendo en cuenta las diferencias en coste computacional que existen entre ambos mallados. Por su parte, el error que se obtiene para el mallado 5 con respecto al mallado 6 es del 1.6%, de forma que el mallado 5 puede resultar también una opción a tener en cuenta para realizar las simulaciones.

Considerando estos resultados y la necesidad de adoptar una solución de compromiso entre coste computacional y exactitud de los resultados, las simulaciones que se mostrarán en las siguientes secciones se realizarán con el mallado 4 y tiempo de paso variable, obteniendo así los mejores resultados posibles para unos tiempos de simulación razonables. Como comentario, estos tiempos de simulación se han alcanzado con un procesador INTEL CORE i5 de 8ª generación y cuatro núcleos con una velocidad máxima de 2.30 GHz, y podrían variar en función del hardware del equipo empleado.

4 Entorno *Ubuntu* y *openFOAM*

Como se comentó anteriormente, *openFOAM* es un software libre que debe ser ejecutado en *Linux* a través del terminal, es decir, se deben conocer los comandos fundamentales para operar en *Ubuntu* y en su entorno, además de aquellos comandos necesarios para correr las simulaciones en *openFOAM*. A continuación se describen los principales elementos a considerar para poder llevar a cabo simulaciones además de los pasos a seguir para ejecutarlas.

4.1 Pasos previos: carpetas *0*, *system* y *constant*

El primer paso para poder crear el modelo que representará el sistema de estudio y lanzar la simulación es crear los archivos y carpetas que usará el software. Para ello, no es necesario trabajar en *Ubuntu* aún y se podrán crear los archivos en entorno *Windows* (o *Linux* si se hace uso de este sistema operativo). El modelo deberá definirse en las carpetas *0*, *system* y *constant*. Estas carpetas contendrán los archivos necesarios para ejecutar la simulación y deberán adecuarse al sistema en estudio (modelo de interfase, flujo laminar o turbulento, consideraciones energéticas...), es decir, deberán contener los archivos necesarios para que el *solver* elegido pueda simular el sistema. En las próximas líneas se describen los archivos necesarios para el *solver* *interFOAM*.

4.1.1 Carpeta *0*

En primer lugar, la carpeta *0* contiene las condiciones iniciales del sistema. En el caso del presente estudio se consideran condiciones iniciales de velocidad, (archivo *U*), presión (archivo *p_rgh*) y fases (archivo *alpha*), aunque pueden añadirse otras condiciones de contorno que puedan ser útiles en el estudio de otros sistemas como, por ejemplo, condiciones iniciales en temperatura para aquellos casos en los que se considere la ecuación de la energía. Los archivos pueden abrirse y editarse haciendo uso del bloc de notas, aunque también pueden usarse otros editores de código como *Visual Studio Code* o *PyCharm*. A continuación se muestran los códigos de los archivos mencionados (nótese que el lenguaje que usa *openFOAM* es C++).

Código 4.1 Archivo *U*.

```
/*-----*- C++ -*-----*\n| ===== | |\n| \\ / F i e l d | O p e n F O A M : T h e O p e n S o u r c e C F D T o o l b o x | |\n| \\ / O p e r a t i o n | V e r s i o n : 2 2 1 2 | |\n| \\ / A n d | W e b s i t e : w w w . o p e n f o a m . c o m | |\n| \\ / M a n i p u l a t i o n | |
```


Nótese cómo se definen las condiciones de contorno en velocidad para el agua y el aire a la entrada de forma independiente ya que se busca que ambas fases tengan la misma velocidad al inicio del flujo para que éste sea estable y se obtengan los resultados esperados. También puede observarse cómo la condición de velocidad a la salida es *zeroGradient*, esto es, gradiente de velocidad nulo. Esta condición, junto con la condición en presión a la salida de *freestreamPressure* definida en el archivo *p_rgh*, será esencial para poder obtener el comportamiento del chorro cuando impacta sobre una superficie libre. La sustitución de esta condición por otras similares y que, en principio, pueden dar resultados parecidos, hace que el modelo cambie drásticamente y los resultados ya no se asemejen a la realidad. Por ejemplo, cabría considerar la posibilidad de imponer velocidad nula a la salida, es decir, tomar la condición *fixedValue* con $U = 0$, pero esto implicaría que el flujo neto a la salida sería nulo, lo cual es incorrecto en el caso de estudio ya que el flujo a la salida es libre. La única forma de garantizar esto, teniendo en cuenta que el chorro choca contra una superficie, es imponiendo que el gradiente de velocidad sea nulo.

Por otro lado, la pared superior se ha definido como *slip*, condición que permite que el flujo no se adhiera a la pared y no se den fenómenos de viscosidad que perturbarían el chorro. Además, se ha considerado que el eje de simetría (pared inferior del chorro) no tiene efecto sobre el flujo, por lo que se le ha definido como *empty*, mientras que las paredes anterior y posterior se han definido como *wedge*, condición que impone que las variables sean simétricas respecto a dichos planos.

Código 4.2 Archivo *p_rgh*.

```

/*-----*- C++ -----*/
/ ===== /
/ \ \ / F i e l d / O p e n F O A M : T h e O p e n S o u r c e C F D T o o l b o x /
/ \ \ / O p e r a t i o n / V e r s i o n : 2 2 1 2 /
/ \ \ / A n d / W e b s i t e : w w w . o p e n f o a m . c o m /
/ \ \ / M a n i p u l a t i o n /
/*-----*/
FoamFile
{
    version    2.0;
    format     ascii;
    class      volScalarField;
    object     p_rgh;
}
// * * * * * //
dimensions   [1 -1 -2 0 0 0 0];
internalField uniform 0;
boundaryField
{
    inletWater
    {
        type          zeroGradient;
    }
    inletAir
    {
        type          zeroGradient;
    }
    outlet

```

```

{
    type            freestreamPressure;
    freestreamValue    uniform 0;
}
top
{
    type            zeroGradient;
}
front
{
    type            wedge;
}
back
{
    type            wedge;
}
axis
{
    type            empty;
}
defaultFaces
{
    type            empty;
}
}
// ***** //

```

Al haber fijado la velocidad de entrada del agua y el aire, la presión debe dejarse como variable libre ya que fijar su valor puede resultar en incompatibilidades entre ambas condiciones. De hecho, una práctica habitual en CFD es fijar una de las dos variables y hacer que la otra variable cumpla una cierta condición, aunque nunca fijar también su valor. En este caso, se ha tomado que la presión a la entrada tenga gradiente nulo, es decir, tenga un valor constante.

Como se ha indicado en anteriores líneas, la presión a la salida tiene la condición de *freestream-Pressure* que, junto con la condición *freestreamValue* que establece una presión de flujo libre nula, permite modelar con precisión el sistema. Como en el caso anterior, las paredes anterior y posterior se han definido como *wedge*, mientras que en la pared superior se ha tomado la condición de *zeroGradient* para así evitar cualquier interferencia de las paredes sobre el chorro.

Código 4.3 Archivo alpha.

```

/*-----* C++ *-----*\
/ ===== /
/ \ \ / Field / OpenFOAM: The Open Source CFD Toolbox /
/ \ \ / Operation / Version: 2212 /
/ \ \ / And / Website: www.openfoam.com /
/ \ \ / Manipulation /
\*-----*/
FoamFile

```



```

import numpy as np
#Inicializar el tamaño del mallado
N = 63900
ny = np.arange(1,N+1)

# Dar el valor inicial de alpha
alpha_init = np.zeros(N)

# Tomar la mitad inferior con alpha = 1 y la mitad superior con alpha =
  0
def alpha_water(x):
    if x < (N/2):
        return 1
    else:
        return 0

alphapy = np.array([alpha_water(xi) for xi in ny])

# Guardar los valores en un archivo
with open('alphapy.water', 'w') as f:
    f.write('dimensions      [0 0 0 0 0 0 0];\n')
    f.write('internalField nonuniform List<scalar>\n')
    f.write(str(N) + "\n")
    for a in alphapy:
        f.write(str(a) + "\n")

```

4.1.2 Carpeta *system*

En la carpeta *system* se declaran la configuración y las características intrínsecas de la simulación, tales como el tiempo total de simulación del sistema, el paso de tiempo o el número de núcleos del procesador a emplear, entre otros. Los archivos más importantes de esta sección son *blockMeshDict*, donde se declara el mallado del modelo, *controlDict*, que establece la configuración de la simulación, y *setExprFields*, que define los valores iniciales de los campos representativos del sistema (en el caso de estudio, velocidad y γ). Los códigos contenidos en cada archivo se muestran a continuación:

Código 4.5 Archivo *blockMeshDict*.

```

/*-----* C++ -*-----*\
/ ===== / /
/ \ \ / F i e l d / O p e n F O A M : T h e O p e n S o u r c e C F D T o o l b o x /
/ \ \ / O p e r a t i o n / V e r s i o n : 2 2 1 2 /
/ \ \ / A n d / W e b s i t e : w w w . o p e n f o a m . c o m /
/ \ \ / M a n i p u l a t i o n / /
\*-----*/
FoamFile
{
    version    2.0;
    format     ascii;
    class      dictionary;

```



```
simpleGrading (1 1 1) // cell expansion ratios
hex (4 5 6 7 8 9 10 11) // vertex numbers
($nx 1 $nz1) // numbers of cells in each direction
simpleGrading (1 1 1) // cell expansion ratios
hex (8 9 10 11 12 13 14 15) // vertex numbers
($nx 1 $nz1) // numbers of cells in each direction
simpleGrading (1 1 1) // cell expansion ratios
hex (12 13 14 15 16 17 18 19) // vertex numbers
($nx 1 $nz2) // numbers of cells in each direction
simpleGrading (1 1 2) // cell expansion ratios
);
boundary
(
    inletWater
    {
        type patch;
        faces
        (
            (0 4 7 0)
            (4 8 11 7)
        );
    }
    inletAir
    {
        type patch;
        faces
        (
            (8 12 15 11)
            (12 16 19 15)
        );
    }

    outlet
    {
        type patch;
        faces
        (
            (1 1 6 5)
            (5 6 10 9)
            (9 10 14 13)
            (13 14 18 17)
        );
    }

    top
    {
        type patch;
        faces
        (
            (16 17 18 19)

```

```

    );
}
axis
{
    type empty;
    faces
    (
        (0 1 1 0)
    );
}
front
{
    type wedge;
    faces
    (
        (0 1 5 4)
        (4 5 9 8)
        (8 9 13 12)
        (12 13 17 16)
    );
}
back
{
    type wedge;
    faces
    (
        (1 0 7 6)
        (6 7 11 10)
        (10 11 15 14)
        (14 15 19 18)
    );
}
);
mergePatchPairs
(
);***** //

```

Como se comentaba, el archivo *blockMeshDict* es el que define el mallado del modelo, es decir, geometría y tamaño y posición de los elementos del mallado. En las primeras líneas de código se definen variables relativas a la geometría, detalladas en la figura 4.1, las variables que se usarán para definir las posiciones de los vértices del prisma del modelo, y el número de elementos de mallado en función del intervalo en z de la geometría (nótese que la coordenada z en *openFOAM* se corresponde con la coordenada r de la figura 4.1).

Posteriormente se definen las posiciones de los vértices haciendo uso de las variables definidas anteriormente, así como los bloques de mallado. También, por último, se definen las distintas caras del prisma (*inletWater*, *inletAir*, *outlet*, etc.).

En el caso descrito se ha definido el mallado a partir del número de elementos en función de la coordenada r , aunque podrían haberse seguido otras estrategias como, por ejemplo, definir el


```

writeInterval 1;
purgeWrite 0;
writeFormat ascii;
writePrecision 6;
writeCompression off;
timeFormat general;
timePrecision 6;
runTimeModifiable yes;
adjustTimeStep yes;
//adjustTimeStep no;
//maxCo 0.5;
maxCo 0.5;
//maxAlphaCo 0.5;
maxAlphaCo 0.5;
maxDeltaT 0.01;
//#sinclude "sampling"
// ***** //

```

En el archivo *controlDict* se definen las características principales de la simulación, tales como el instante inicial de la simulación, el tiempo final, el paso de tiempo o el máximo número de Courant admisible. Nótese como se ha establecido paso de tiempo adaptativo con la línea de código *adjustTimeStep yes*, lo que permitirá que el número de Courant nunca supere el valor de 0.5 definido en el código.

En este archivo también se puede definir cómo graba el programa los valores de las variables, estableciendo la frecuencia de guardado de las variables con el comando *writeInterval* y el número de cifras significativas que se tendrán en cuenta mediante la orden *writePrecision*. En este caso se ha tomado una frecuencia de 1 s con un grabado de 6 cifras significativas.

Código 4.7 Archivo setExprFields.

```

/*-----*- C++ -*-----*/
/ ===== /
/ \ \ / F i e l d / O p e n F O A M : T h e O p e n S o u r c e C F D T o o l b o x /
/ \ \ / O p e r a t i o n / V e r s i o n : 2 2 1 2 /
/ \ \ / A n d / W e b s i t e : w w w . o p e n f o a m . c o m /
/ \ \ / M a n i p u l a t i o n / /
/*-----*/
FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    object setExprFieldsDict;
}
// ***** //
defaultFieldValues
(
    volScalarFieldValue alpha.water 0
    volVectorFieldValue U (0. 0. 0.)

```

```
);
expressions
(
  alpha.water
  {
    field      alpha.water;

    constants
    {
    }

    variables
    (
      "value = 1.0"
      "R0    = 1.0"
    );

    fieldMask
    #{
      pos().z() < R0
    #};

    expression
    #{
      value
    #};
  }
  U
  {
    field      U;
    dimensions [0 1 -1 0 0 0 0];

    constants
    {
    }

    variables
    (
      "OH1 = 1.6"
      "OH2 = 6.8"
      "WE  = 10.5"
    );
    expression
    #{
      vector(3.2404,0,0)
    #};
  }
);
// ***** //
```

En este archivo se definen los valores iniciales de los campos fracción volumétrica de volumen γ y velocidad U .

4.1.3 Carpeta *constant*

La última carpeta necesaria para poder realizar la simulación es la carpeta *constant*, donde se definen las características del flujo. Los archivos que se deberán tener en cuenta son el archivo *g*, donde se define el valor de la gravedad, *transportProperties*, donde se establecen los valores de la viscosidad cinemática y densidad de cada fase, además de la tensión superficial (definida como *sigma* en el código), que será la unidad tal y como se vió en la sección 2.2, y *turbulenceProperties*, donde se declaran las características del flujo turbulento, en caso de ser aplicable. En el caso de estudio, al haber considerado que el sistema está siempre en régimen laminar, únicamente será necesario declarar que el régimen es laminar, sin necesidad de dar valores a ningún parámetro turbulento.

Se detallan los códigos comentados a continuación:

Código 4.8 Archivo *g*.

```

/*-----*- C++ -*-----*/
/ ===== /
/ \ \ / F i e l d / OpenFOAM: The Open Source CFD Toolbox /
/ \ \ / O p e r a t i o n / Version: 2212 /
/ \ \ / A n d / Website: www.openfoam.com /
/ \ \ / M a n i p u l a t i o n / /
/*-----*/

FoamFile
{
    version    2.0;
    format     ascii;
    class      uniformDimensionedVectorField;
    object     g;
}

// * * * * *
dimensions    [0 1 -2 0 0 0 0];
value         (0 -9.81 0);
// * * * * *

```

Código 4.9 Archivo *g*.

```

/*-----*- C++ -*-----*/
/ ===== /
/ \ \ / F i e l d / OpenFOAM: The Open Source CFD Toolbox /
/ \ \ / O p e r a t i o n / Version: 2212 /
/ \ \ / A n d / Website: www.openfoam.com /
/ \ \ / M a n i p u l a t i o n / /
/*-----*/

FoamFile
{
    version    2.0;
    format     ascii;

```


- Para cambiar de carpeta, se usa el comando `cd folder`, donde *folder* es la carpeta objetivo.
- Para poder volver a la carpeta original que contenga a la subcarpeta a la que se ha cambiado haciendo uso del comando anterior, se usará el comando `cd ..`.
- Para ver los contenidos de una carpeta se hace uso de `ls`.

En el siguiente ejemplo se presentan los pasos necesarios para llegar a la carpeta que contiene una de las simulaciones.

Código 4.11 Uso de los comandos `cd` y `ls`.

```

framungom1@DESKTOP-MDT6PQN:~$ cd ..
framungom1@DESKTOP-MDT6PQN:/home$ cd ..
framungom1@DESKTOP-MDT6PQN:/$ cd ..
framungom1@DESKTOP-MDT6PQN:/$ cd mnt
framungom1@DESKTOP-MDT6PQN:/mnt$ cd c
framungom1@DESKTOP-MDT6PQN:/mnt/c$ cd Users
framungom1@DESKTOP-MDT6PQN:/mnt/c/Users$ cd elnin
framungom1@DESKTOP-MDT6PQN:/mnt/c/Users/elnin$ cd OneDrive/
framungom1@DESKTOP-MDT6PQN:/mnt/c/Users/elnin/OneDrive$ cd Desktop/
framungom1@DESKTOP-MDT6PQN:/mnt/c/Users/elnin/OneDrive/Desktop$ cd TFG
framungom1@DESKTOP-MDT6PQN:/mnt/c/Users/elnin/OneDrive/Desktop/TFG$ cd
  Openfoam
framungom1@DESKTOP-MDT6PQN:/mnt/c/Users/elnin/OneDrive/Desktop/TFG/
  Openfoam$ cd ar01.1-We15/
framungom1@DESKTOP-MDT6PQN:/mnt/c/Users/elnin/OneDrive/Desktop/TFG/
  Openfoam/ar01.1-We15$ ls
0 10 20 30 40 50 60 Arrugas_Mesh_V2.tsv ar01-We15.foam constant
  system
framungom1@DESKTOP-MDT6PQN:/mnt/c/Users/elnin/OneDrive/Desktop/TFG/
  Openfoam/ar01.1-We15$

```

Una vez se llega a la carpeta deseada, los pasos a seguir para lanzar la simulación, en orden, son:

1. `blockMesh` → Genera el mallado definido en el archivo *blockMeshDict*.
2. `checkMesh` → Este comando permite ver más detalladamente las características del mallado, tales como tamaño mínimo del elemento, máxima relación de aspecto, máxima asimetría o *skewness*, etc. Estas características deben tener los valores adecuados para que la simulación sea lo suficientemente exacta.
3. `setExprFields` → Establece los valores iniciales de los campos declarados en el archivo *setExprFields*.
4. `interFoam` → Con este comando se da inicio a la simulación. Nótese que se corresponde con el *solver* usado, por lo que puede cambiar dependiendo del *solver* que se use.

Una vez se inicie la simulación, *openFOAM* mostrará en pantalla los resultados que se obtienen para cada instante de tiempo, así como el paso de tiempo establecido para cada iteración y el número de Courant. Es aconsejable controlar estos valores para asegurar el correcto desarrollo de la simulación.

4.3 ParaView

Con la simulación finalizada y los resultados guardados en la carpeta deseada, el software libre *ParaView* permite visualizar los resultados obtenidos. Este software puede descargarse fácilmente a través de la página [1].

El software tiene una gran cantidad de utilidades y permite visualizar muy diversos elementos a partir de los resultados. Por este motivo, en las siguientes líneas se describen aquellas utilidades empleadas para obtener los resultados que se muestran en las siguientes secciones. Para más información sobre todas las funcionalidades del software, el lector puede consultar el tutorial del software en la sección *Documentation* de [1].

4.3.1 Visualización de resultados

Una vez cargado el modelo en *ParaView* (archivo del tipo *.foam*), se pueden cambiar las variables que se muestran en pantalla y cómo se muestran a través de los desplegados (1) y (2), mostrados en la figura 4.2.

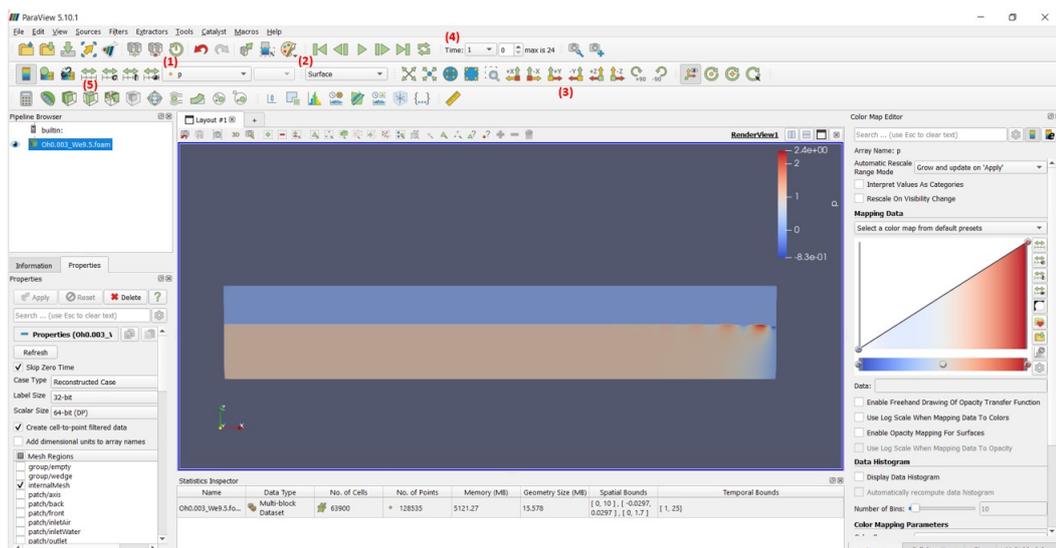


Figura 4.2 Principales utilidades de *ParaView*.

La opción (1) permite seleccionar la variable a mostrar en pantalla (presión, velocidad o fracción volumétrica), así como su componente XYZ, mientras que la opción (2) permite elegir cómo se muestra el modelo (como superficie, puntos, volumen, etc.). También puede seleccionarse la vista con la que se desean presentar los resultados haciendo uso del panel (3), que muestra la vista normal al eje elegido (en este caso, la vista normal al eje Y positivo).

Por otro lado, el desplegado (4) permite seleccionar el instante de tiempo de la simulación en el que se muestran los resultados, correspondiéndose dichos instantes con los establecidos en el archivo *controlDict* para grabar los resultados. De forma general, al cambiar de instante de tiempo los resultados no tengan la escala deseada para los datos mostrados por pantalla, por lo que es aconsejable cambiar la escala en el panel (5).

4.3.2 Obtención de resultados: puntos de la interfase

El objetivo de las simulaciones es obtener la interfase agua-aire de forma explícita, por lo que se deben grabar los puntos de la interfase en archivos del tipo .tsv o .csv, que permitirán representar gráficamente dichas interfases haciendo uso de *MATLAB* o *Gnuplot*. Para ello, se deben seguir los siguientes pasos:

1. Establecer un *Contour* haciendo uso de la herramienta (6), tomando *Contour by: alpha.water* con un valor de 0.5.
2. Haciendo click derecho sobre el *Contour*, tomar la opción *Add filter / Data Analysis / Integrate Variables*. La ventana que se muestra permitirá obtener los puntos de la interfase.
3. En la ventana de *Integrate Variables*, seleccionar el *Contour* creado en el primer paso e imponer que se muestren únicamente los puntos de dicho *Contour*. Es importante ordenar dichos puntos de menor a mayor (en este caso, en la dirección X) para que *MATLAB* o *Gnuplot* puedan beber de los datos correctamente.
4. Haciendo uso de la herramienta (7), guardar los puntos en un archivo .tsv o .csv, como se desee.

En la siguiente figura se muestra el resultado de los pasos seguidos en *ParaView*:

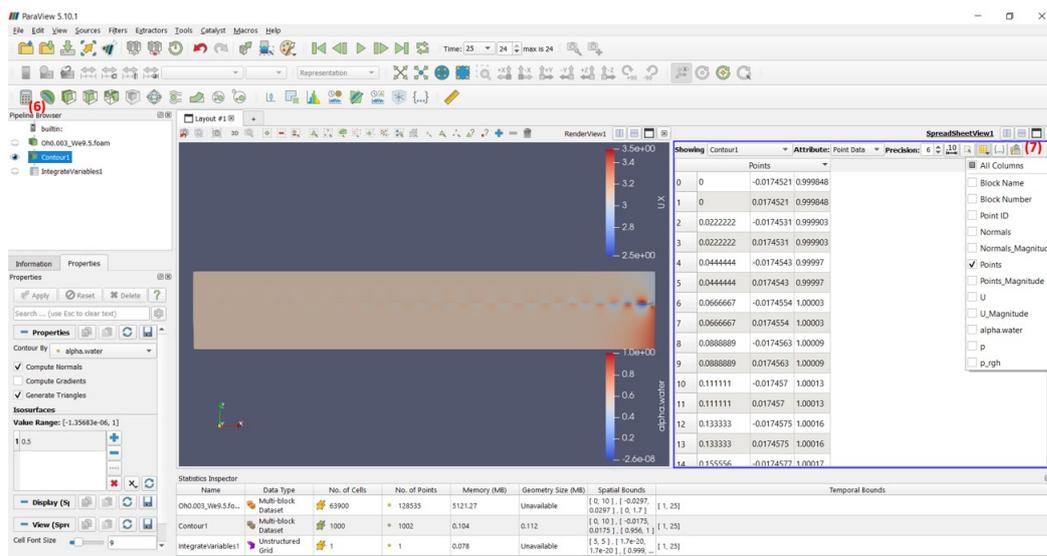


Figura 4.3 Cómo obtener los puntos de la interfase con *ParaView*.

Con el archivo guardado en la carpeta deseada, únicamente resta representar gráficamente la interfase. En *MATLAB* puede hacerse de forma sencilla empleando los siguientes comandos:

Código 4.12 MATLAB.

```
Data = table2array(readtable("Data.tsv", "FileType", "text", 'Delimiter',
    '\t'));
figure(1)
plot(Data(:,1),Data(:,3));
xlim([5,10])
ylim([0.95,1.01])
set(gcf, 'WindowState', 'maximized');
```

```
figData = 'figData.png';  
saveas(1,figData);
```

Obteniéndose la siguiente gráfica:

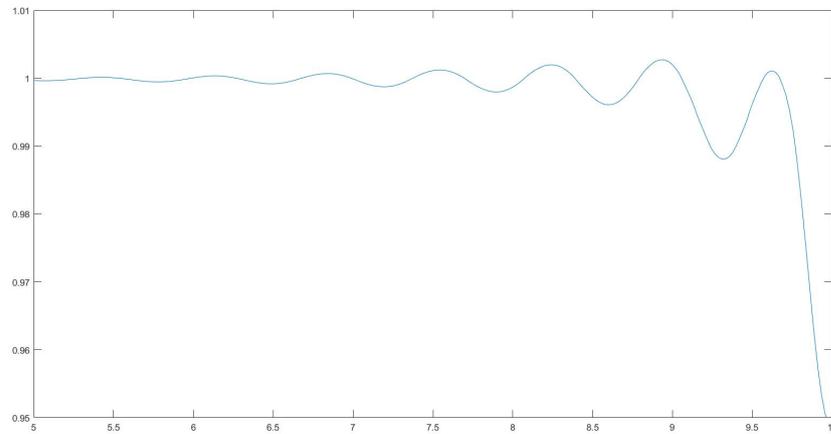


Figura 4.4 Representación de la interfase con *MATLAB*.

Mientras que con *Gnuplot* basta con ejecutar la siguiente línea:

```
plot "Data.tsv" using 1:3 with lines
```

Obteniéndose la siguiente gráfica:

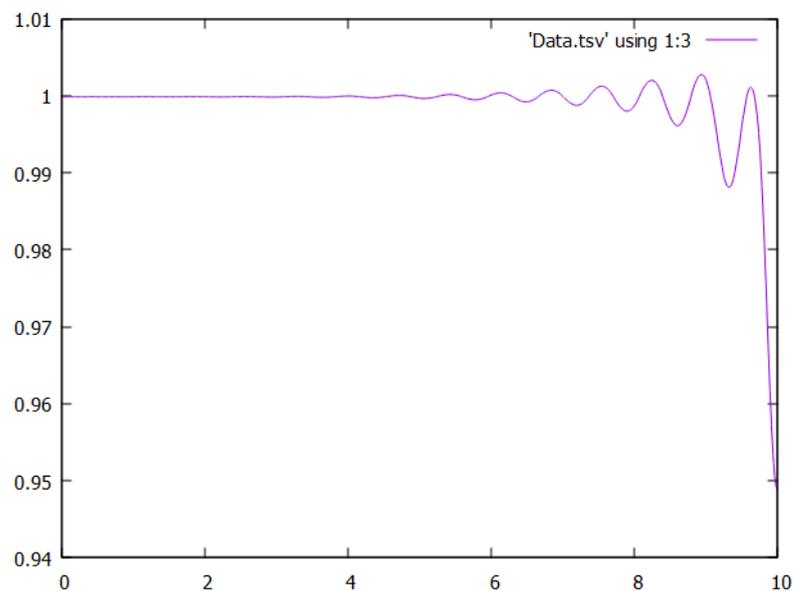


Figura 4.5 Representación de la interfase con *Gnuplot*.

5 Análisis de resultados

Establecidos el modelo, los parámetros de simulación y la configuración de *interFOAM*, se procede a simular el sistema para los valores $Oh = 0.003$ y $Oh = 0.01$ para aquellos números de Weber comprendidos entre 8 y 15 con saltos de 0.5. Estas simulaciones permitirán visualizar el comportamiento de la interfase agua-aire para el caso de $Oh = 0.003$ pudiéndose ver gráficamente la evolución del chorro en función del número de Weber. También se estudiará la dependencia de las arrugas del chorro con el número de Ohnesorge, comparándose los resultados obtenidos para $Oh = 0.003$ y $Oh = 0.01$.

De este modo, en las siguientes páginas se representarán las interfases obtenidas para los números de Weber mencionados, diferenciando los casos de $Oh = 0.003$ y $Oh = 0.01$. Cabe destacar que para el caso de $Oh = 0.003$ no ha sido posible obtener simulaciones estables para $We = 8$ y $We = 8.5$, de forma que no se tendrán en cuenta las interfases para estos números de Weber en este caso. Estos problemas de inestabilidad puntuales únicamente se observan en *openFOAM*, no apareciendo en simulaciones realizadas con *Basilisk*, por lo que podría tratarse de un problema relacionado con *interFOAM* o con el mallado elegido, pues estas inestabilidades no aparecen en las simulaciones con $Oh = 0.01$. Esta anomalía se tratará con mayor profundidad en el capítulo 7.

5.1 Interfase agua-aire. Resultados para $Oh = 0.003$

A continuación se presentan los resultados obtenidos para $Oh = 0.003$, representándose las interfases que se obtienen cuando el sistema alcanza un estado de pseudo-equilibrio, siendo el instante en el que se da esa condición dependiente del número de Weber. No obstante, ese instante se encuentra, aproximadamente, en torno a $t = 30$ (recordar que las variables del problema se han adimensionalizado) para la mayoría de los casos.

Con el fin de poder visualizar correctamente las arrugas del chorro en la zona cercana al obstáculo y su evolución a lo largo de la coordenada axial, se representan las interfases para $x \in [5,10]$ y $r \in [0.95,1.01]$, no perdiéndose información relevante sobre las arrugas y permitiendo comparar las ondas formadas para distintos números de Weber de forma sencilla.

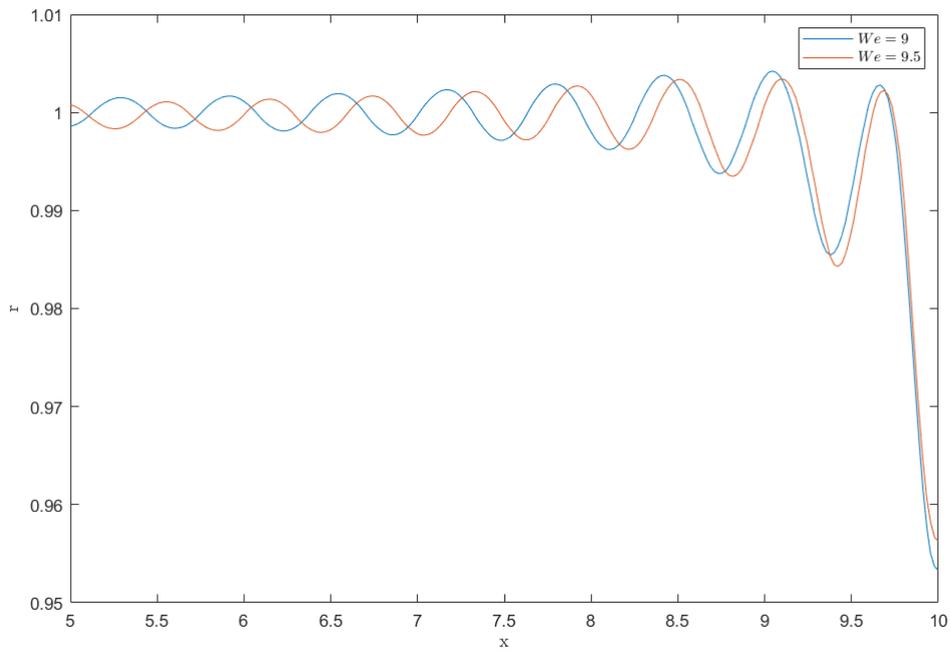


Figura 5.1 Interfases obtenidas para $We = 9$ y $We = 9.5$.

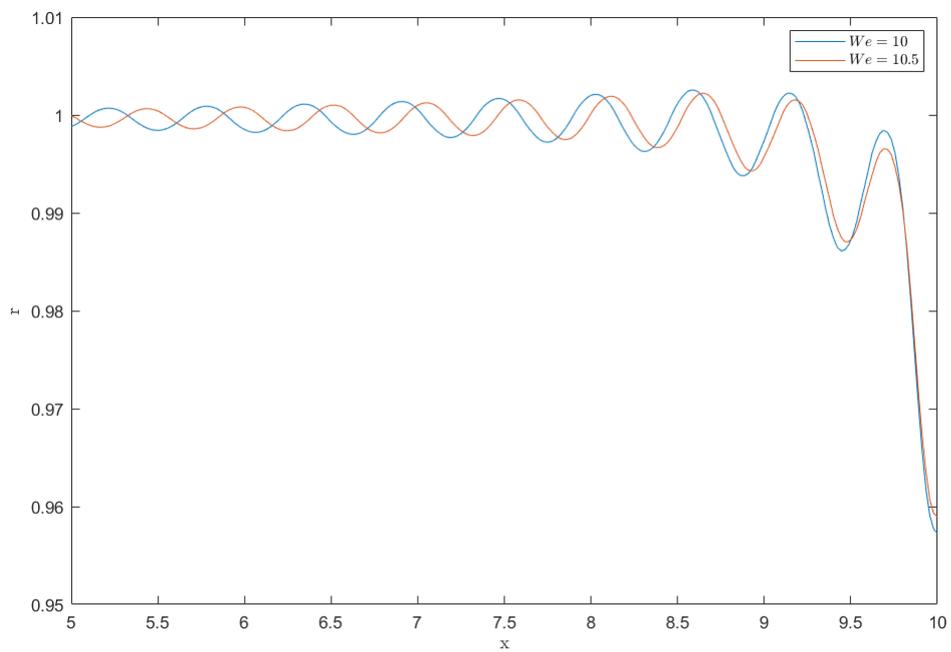


Figura 5.2 Interfases obtenidas para $We = 10$ y $We = 10.5$.

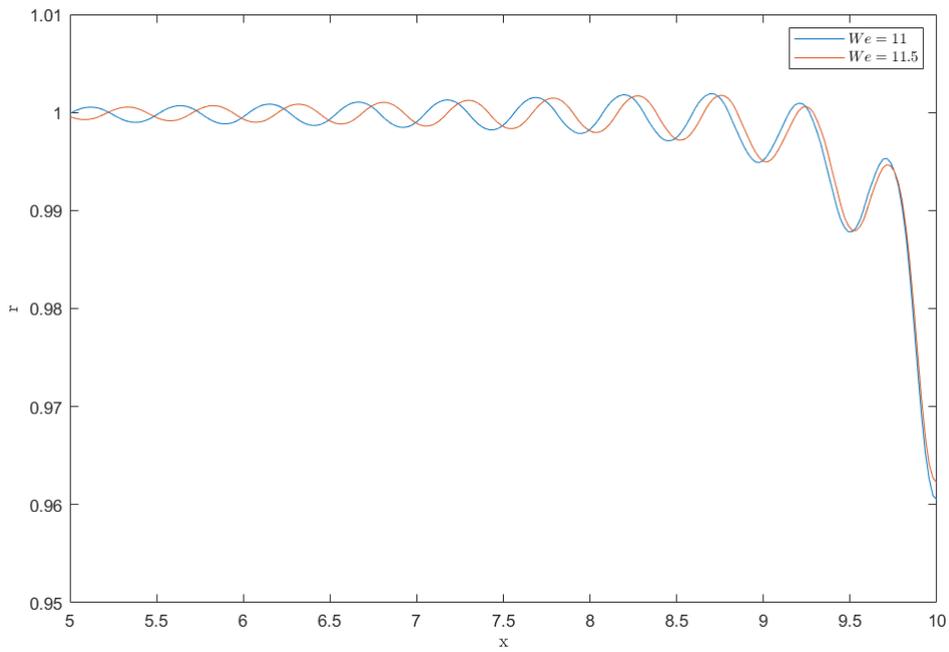


Figura 5.3 Interfases obtenidas para $We = 11$ y $We = 11.5$.

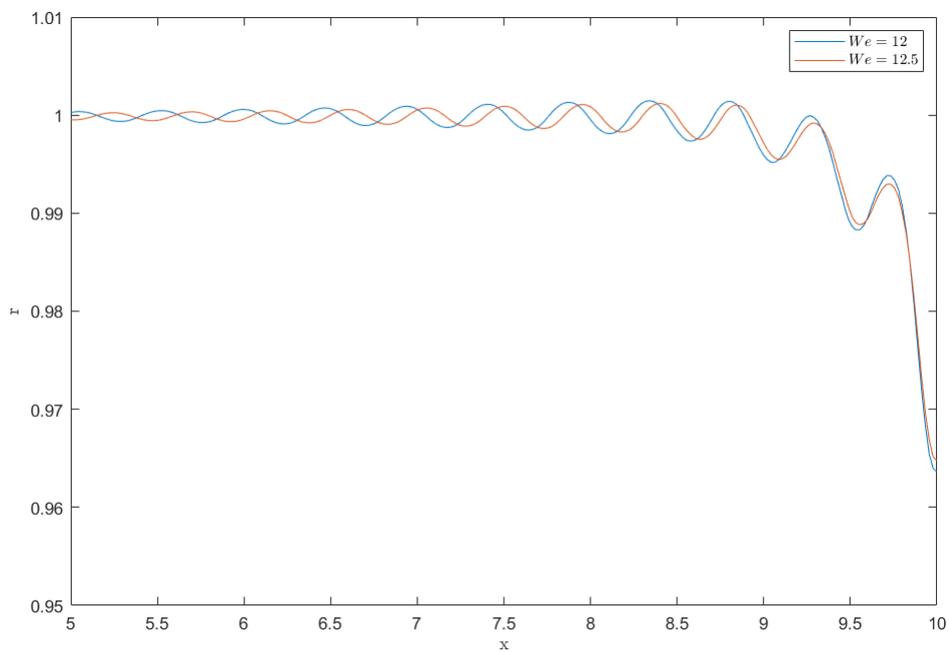


Figura 5.4 Interfases obtenidas para $We = 12$ y $We = 12.5$.

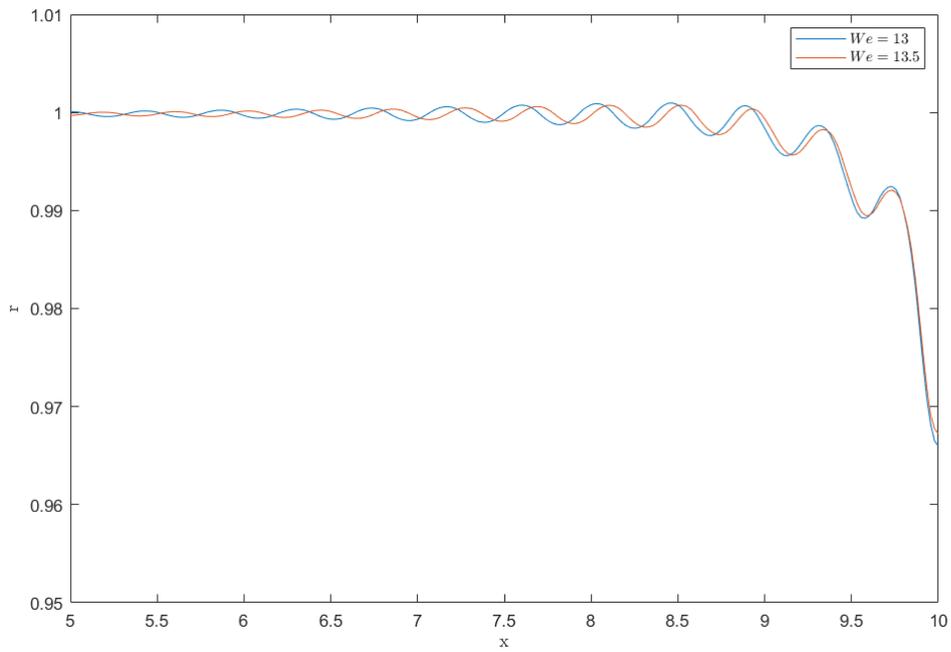


Figura 5.5 Interfases obtenidas para $We = 13$ y $We = 13.5$.

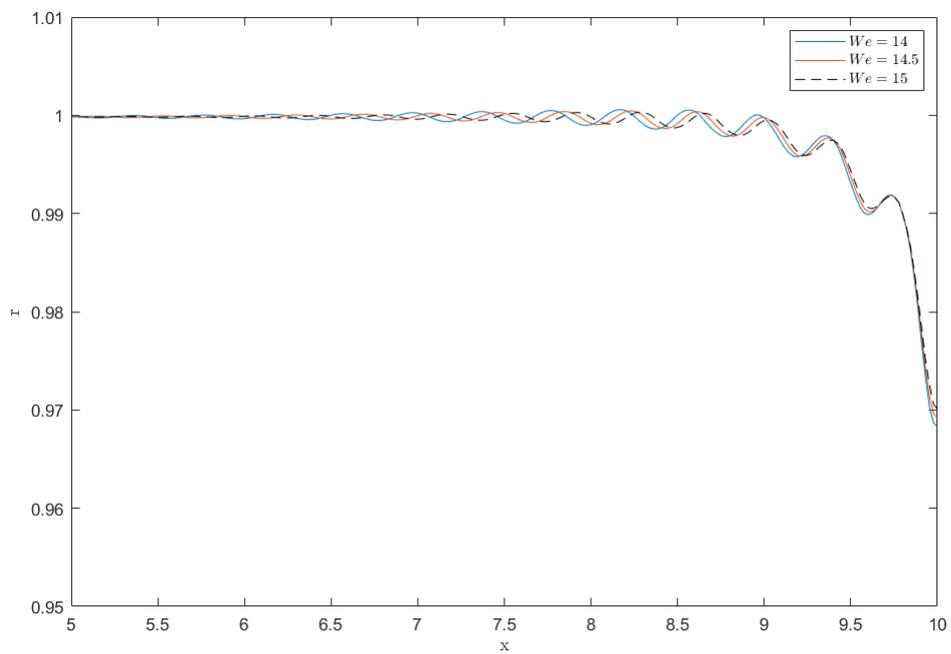


Figura 5.6 Interfases obtenidas para $We = 14$, $We = 14.5$ y $We = 15$.

Puede observarse cómo, a medida que aumenta el número de Weber, la longitud de onda de las arrugas disminuye, así como el decaimiento, esto es, pérdida de amplitud en la dirección axial (en el sentido decreciente de x), se acentúa. El efecto de estos parámetros sobre la interfase se verán más profundamente en el siguiente capítulo, en el que se comparan los resultados obtenidos de forma teórica y los obtenidos con *openFOAM*. Comparando las interfases obtenidas para $We = 9$ y $We = 15$ puede visualizarse fácilmente este fenómeno:

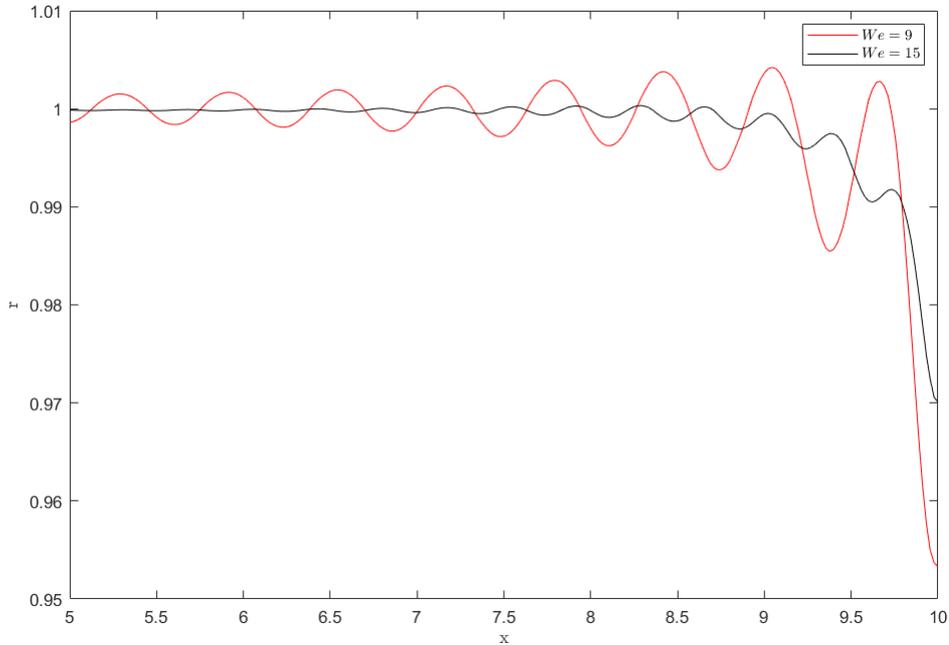


Figura 5.7 Comparativa entre las interfases obtenidas para $We = 9$ y $We = 15$.

5.2 Resultados para $Oh = 0.01$

Se presentan a continuación los resultados obtenidos para $Oh = 0.01$. Para este número de Ohnesorge, a diferencia del caso anterior, sí ha sido posible obtener simulaciones estables para $We = 8$ y $We = 8.5$, haciéndose notable el efecto del número de Ohnesorge sobre el sistema. Al igual que en el caso anterior, se representan las gráficas para $x \in [5, 10]$ y $r \in [0.95, 1.01]$.

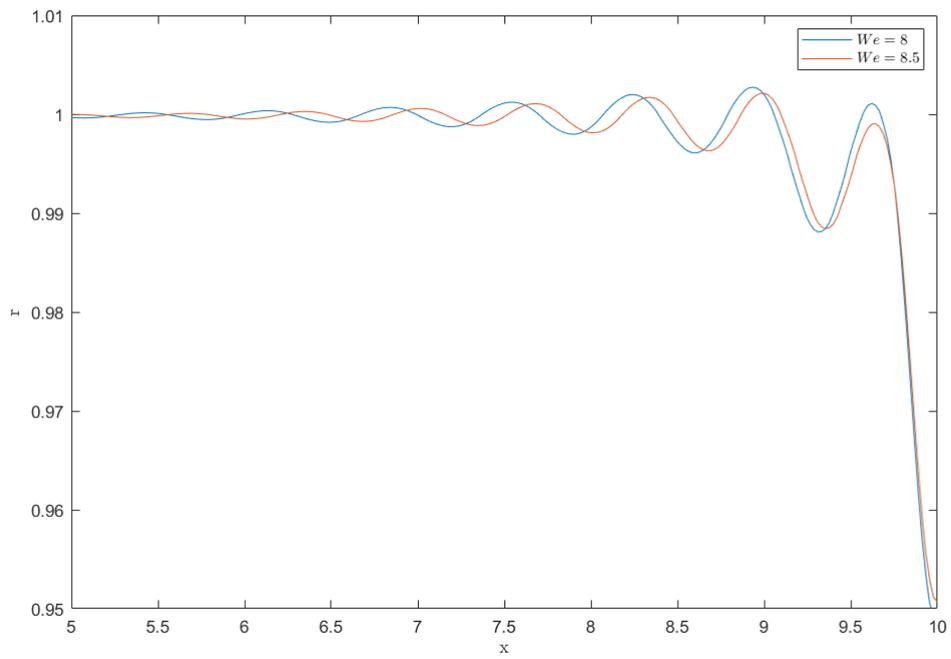


Figura 5.8 Interfases obtenidas para $We = 8$ y $We = 8.5$.

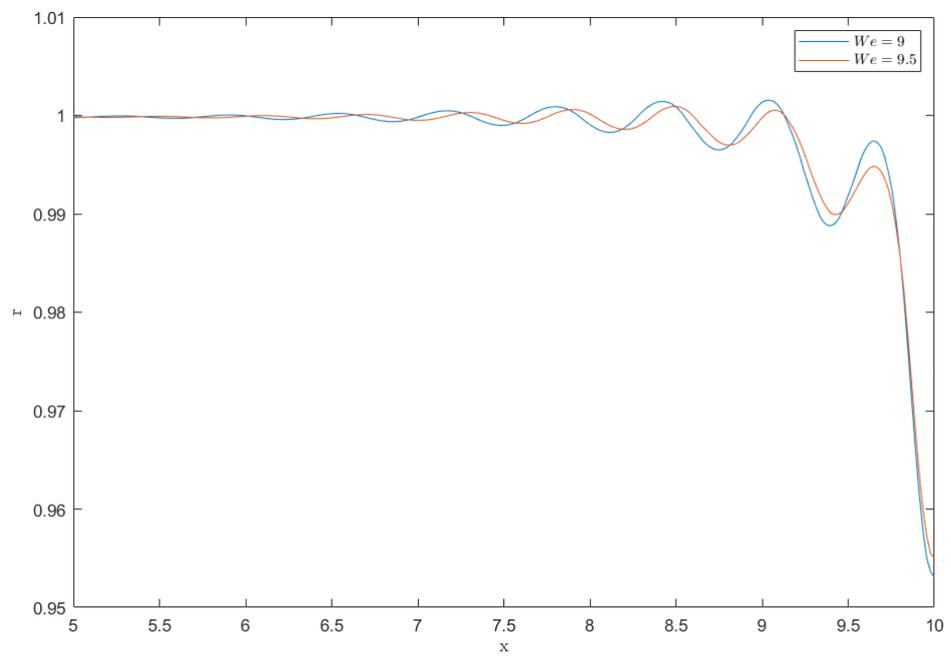


Figura 5.9 Interfases obtenidas para $We = 9$ y $We = 9.5$.

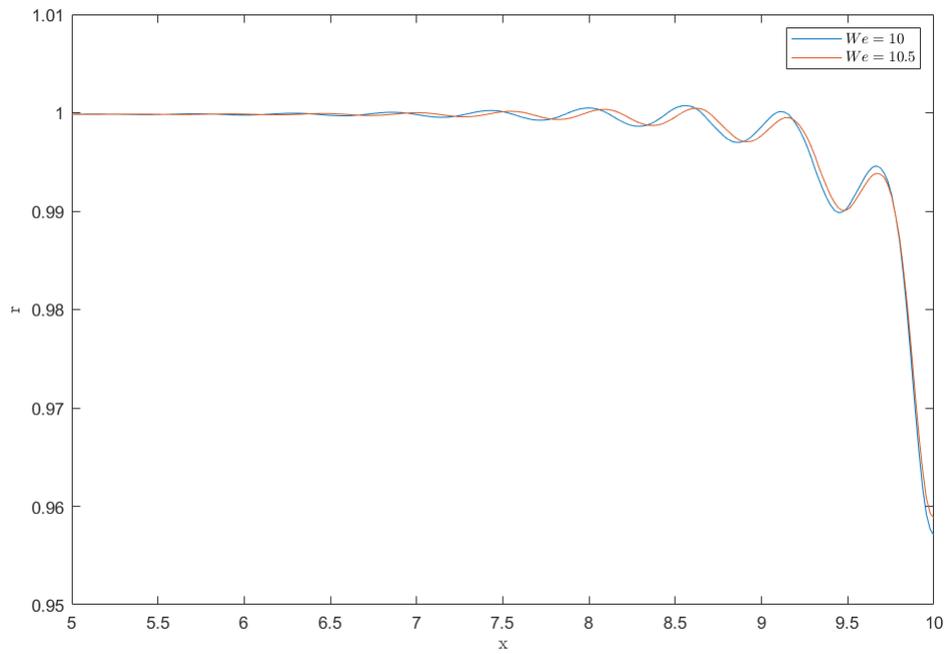


Figura 5.10 Interfases obtenidas para $We = 10$ y $We = 10.5$.

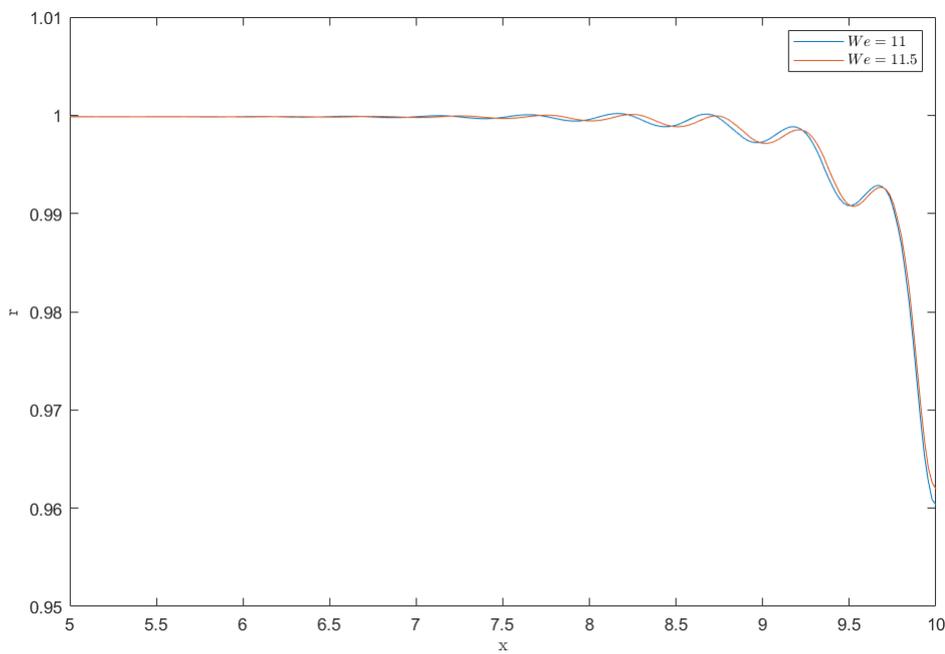


Figura 5.11 Interfases obtenidas para $We = 11$ y $We = 11.5$.

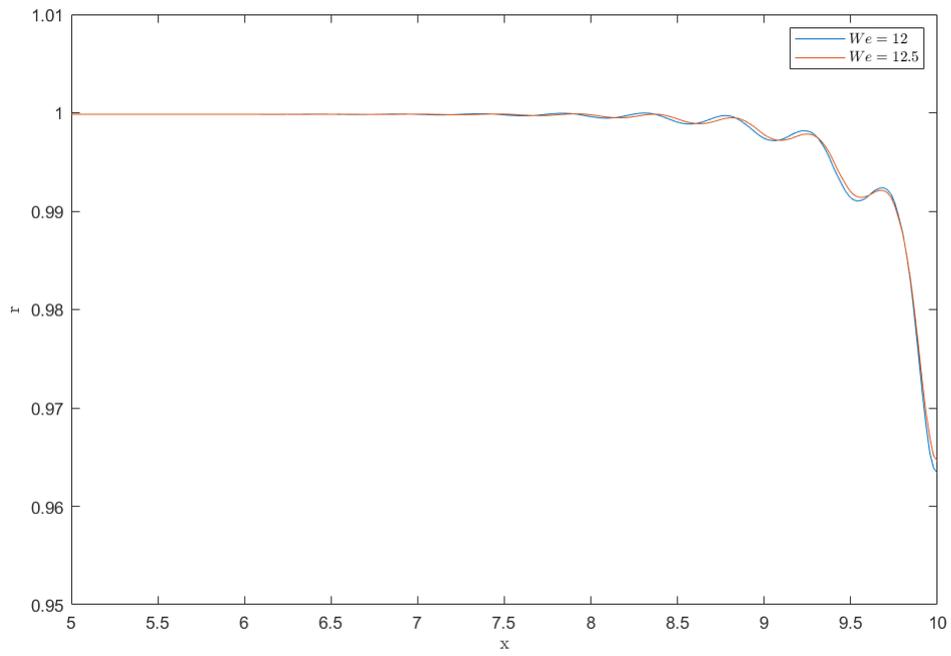


Figura 5.12 Interfases obtenidas para $We = 12$ y $We = 12.5$.

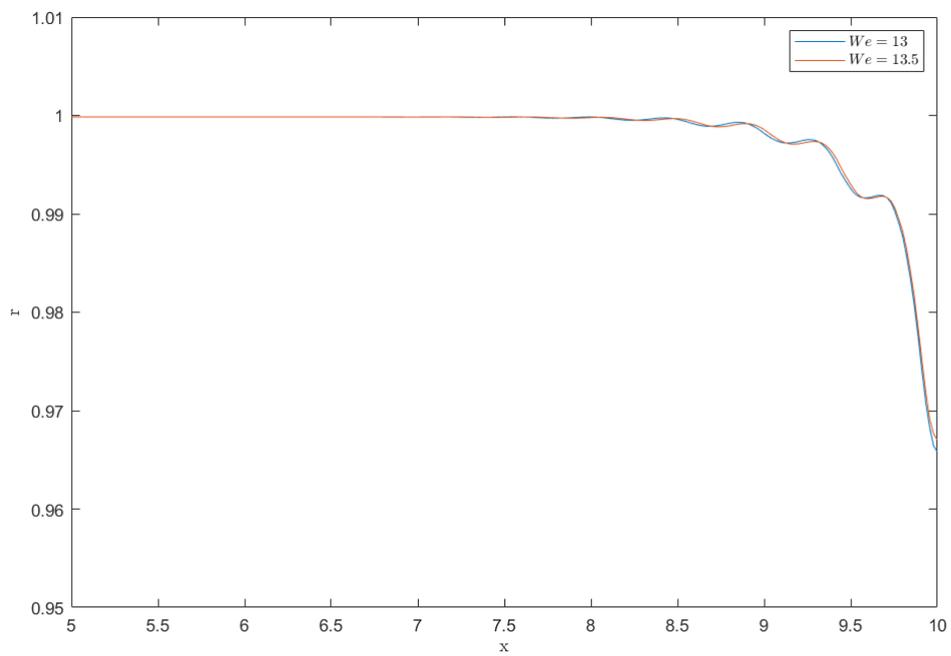


Figura 5.13 Interfases obtenidas para $We = 13$, $We = 13.5$.

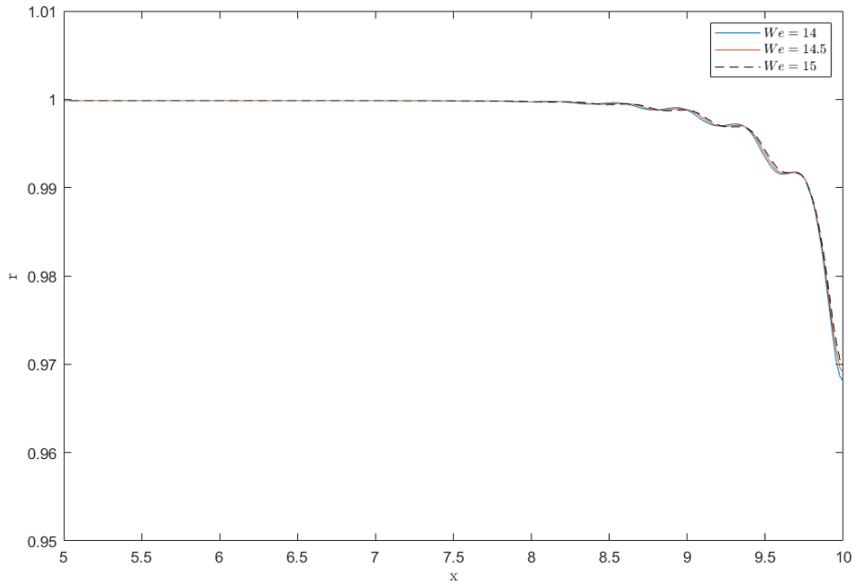


Figura 5.14 Interfases obtenidas para $We = 14$, $We = 14.5$ y $We = 15$.

Como sucedía en el caso anterior, puede apreciarse cómo la longitud de onda disminuye a medida que el número de Weber aumenta y cómo crece la tasa de decaimiento. Nótese que la tasa de decaimiento crece más rápidamente en este caso que para $Oh = 0.003$, pudiéndose ver claramente este fenómeno comparando las arrugas obtenidas con $We = 13$ para ambos números de Ohnesorge:

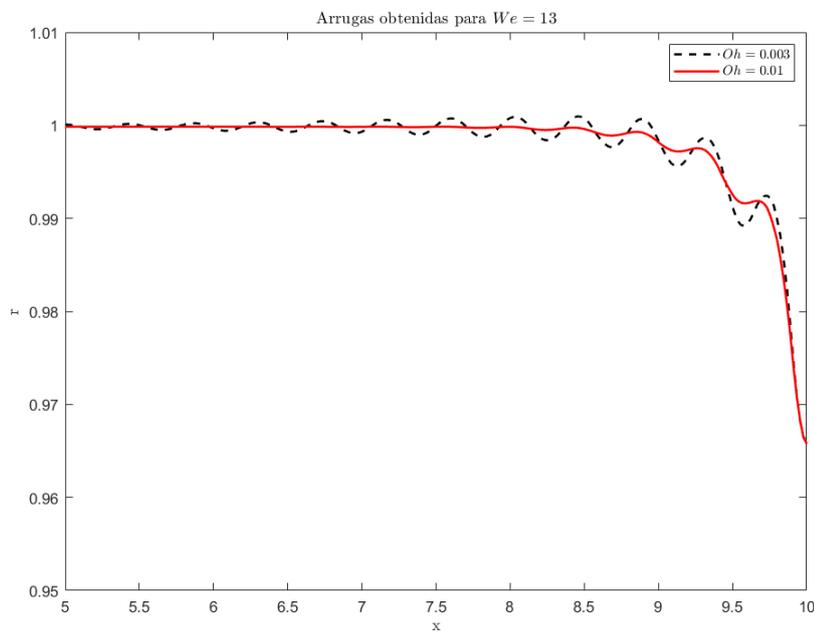


Figura 5.15 Arrugas obtenidas para $We = 13$ para los casos $Oh = 0.003$ y $Oh = 0.01$. Puede apreciarse claramente cómo la tasa de decaimiento en el caso de $Oh = 0.01$ es notablemente mayor, atenuándose en gran medida la onda originada.

El efecto del número de Weber también puede verse en la siguiente figura, donde se comparan las arrugas obtenidas para $We = 9$ y $We = 15$. Comparando estos resultados con los de la figura 5.7, de nuevo se puede apreciar cómo la longitud de onda disminuye más rápidamente y la tasa de decaimiento crece con más fuerza, evidenciando la dependencia de las arrugas con el número de Ohnesorge.

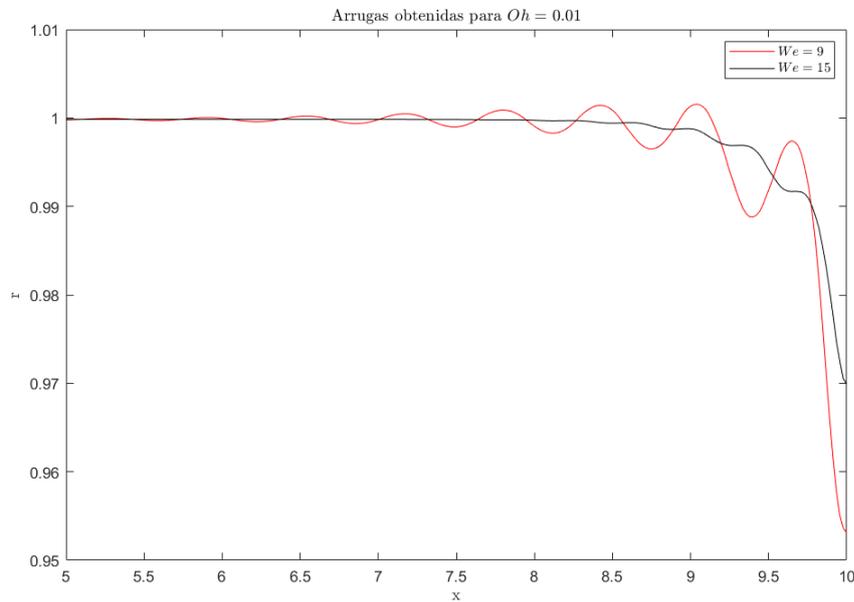


Figura 5.16 Arrugas obtenidas para $We = 9$ y $We = 15$. Puede apreciarse la mayor dependencia con respecto al caso de $Oh = 0.003$ de la longitud de onda y la tasa de decaimiento con el número de Weber..

Obtenidos estos resultados para varios números de Weber y números de Ohnesorge, se revela que las arrugas guardan una fuerte dependencia con el número de Weber, haciéndose esta dependencia más intensa cuanto mayor es el número de Ohnesorge. Como se comentaba, en todos los casos se observa que la longitud de onda disminuye con el incremento del número de Weber, así como la tasa de decaimiento aumenta, siendo estos efectos más notables para números de Ohnesorge mayores.

Estas dependencias se explican más detalladamente en el siguiente capítulo, en el que se describe el modelo teórico de las arrugas y su adecuación a los resultados obtenidos numéricamente.

6 Modelo teórico

En este capítulo se describe el modelo teórico que permite obtener las arrugas analíticamente y se estudia su adecuación a los resultados obtenidos de forma numérica. Para ello, se hará uso de la rutina *NonlinearModelFit* de *Mathematica*, que ajustará los datos de las simulaciones a los parámetros teóricos mediante el método más adecuado en cada caso, elegido en base a unos criterios internos del programa. Más información sobre esta rutina puede consultarse en [2].

El modelo teórico de la forma estática (pseudo-estacionaria) del chorro se construye a partir de una combinación de modos espaciales que son generados en el obstáculo y trepan aguas arriba, no participando los modos generados en el orificio que viajan aguas abajo al no existir para frecuencia nula. De este modo, se clasifican los modos en dos grandes familias:

- **Modos capilares.** Se consideran dos modos capilares, claramente asociados a la deformación superficial del chorro, y que combinados se describen por la función f_{cap} :

$$f_{cap} = A \exp[k_i(z - z_{max})] \sin[k_r(z - z_{max}) + \phi] \quad (6.1)$$

Donde k_i es la tasa de decaimiento asociada a la exponencial amortiguadora y k_r el número de onda de los modos capilares.

- **Modos inerciales.** Los modos inerciales son una familia infinita de modos puramente amortiguados, cuyos exponentes de decrecimiento son aproximadamente los ceros de la función de Bessel $J_0(x_n)$, con $n = 1, 2, 3, \dots$. La función de Bessel se puede expresar como:

$$J_0(x) = 1 - \frac{x^2}{2^2} + \frac{x^4}{2^2 4^2} - \frac{x^6}{2^2 4^2 6^2} \dots \quad (6.2)$$

Tienen una contribución significativa los modos $n = 1$ y $n = 3$, aunque este último tiene una amplitud mucho menor que el caso $n = 1$, pero un amortiguamiento mayor, apreciable especialmente en puntos cercanos al obstáculo. De esta forma, se descarta su contribución, considerándose únicamente la contribución del caso $n = 1$, resultando en la función de los modos inerciales:

$$f_{inc} = B \exp[k_1(z - z_{max})] \quad (6.3)$$

Más información sobre los modos espaciales y su contribución en la forma estática del chorro puede consultarse en [8]. Con esto, la función que se usará para describir las arrugas de forma teórica será la combinación de las funciones anteriores, esto es:

$$f = f_{cap} + f_{inc} = A \exp[k_i(z - z_{max})] \sin[k_r(z - z_{max}) + \phi] + B \exp[k_1(z - z_{max})] \quad (6.4)$$

Incluir más modos inerciales en la función anterior hace que se puedan describir con más precisión pequeñas zonas del dominio axial, aunque también hace que el número de parámetros a ajustar aumente en gran medida, dificultando un ajuste preciso en todo el dominio. Por este motivo únicamente se considera el modo inercial $n = 1$, que tiene un efecto considerable sobre todo el dominio.

Con el fin de comparar los resultados teóricos y los numéricos, se considera el caso de $Oh = 0.003$ y $We = 9$. Ajustando los parámetros de la función f a los resultados obtenidos numéricamente, se obtienen las arrugas representadas en la siguiente figura, que se comparan para aquellas obtenidas numéricamente:

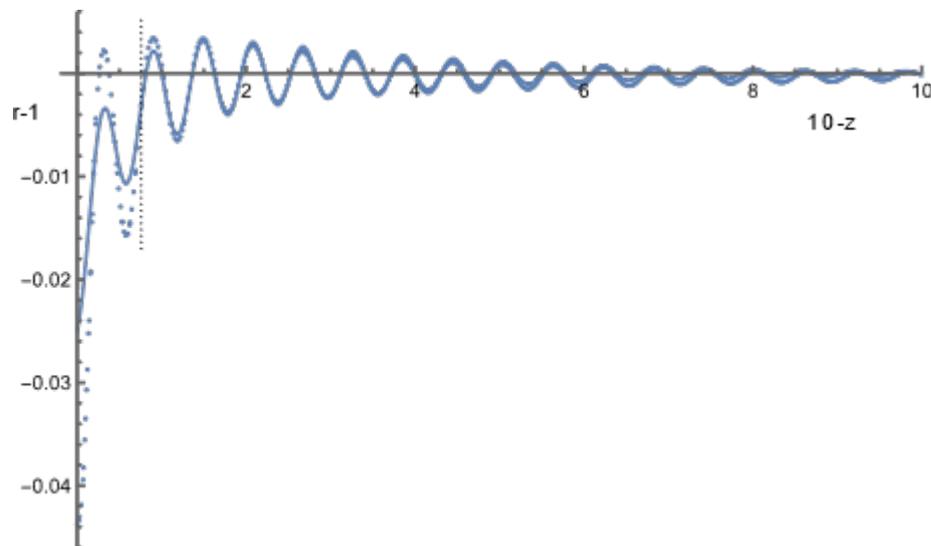


Figura 6.1 Comparativa de las arrugas obtenidas numéricamente (puntos) y las obtenidas por el modelo teórico (línea continua), donde se ha colocado el obstáculo a la izquierda ($x = 0$) y se ha restado la forma no perturbada del chorro ($r = 1$).

Nótese cómo el modelo teórico se ajusta con gran precisión a los resultados obtenidos con las simulaciones, demostrándose así la capacidad de las predicciones teóricas de describir correctamente las tendencias de variación de las arrugas con el número de Weber y de Ohnesorge.

Cabe destacar que, en la figura superior, la zona cercana al obstáculo (delimitada por la línea vertical) se ha descartado para el ajuste como consecuencia de no haber tenido en cuenta el modo inercial $n = 3$. No obstante, este ajuste es localizado y no afecta a la gran mayoría del dominio axial.

En la siguiente tabla se comparan los parámetros teóricos obtenidos mediante el ajuste a los resultados numéricos y sus valores teóricos:

Tabla 6.1 Valores obtenidos mediante el ajuste y valores predichos por la teoría.

	k_r	k_i	k_1
Ajuste	10.59	0.3431	2.114
Valor teórico	10.12	0.3295	2.696

Los valores teóricos de los modos capilares son los que mejor se ajustan a los obtenidos a partir de las simulaciones, deviándose k_r un 5% del valor numérico y la tasa de decaimiento, k_i , un 3%. No obstante, el valor k_1 del modo inercial se desvía un 22%, manifestándose la dificultad de ajustar los modos inerciales, aunque en este caso dicho ajuste no queda lejos de los resultados obtenidos mediante las simulaciones.

Por último, las predicciones teóricas describen correctamente la dependencia de las arrugas con los números de Weber y Ohnesorge que se vieron en el capítulo anterior. A mayor número de Weber, menor longitud de onda y mayor tasa de decaimiento, efecto que se acentúa cuanto mayor es el número de Ohnesorge. Además, cuanto mayor es el número de Weber, el efecto de los modos inerciales se hace más notable, detectándose cerca del obstáculo por la desviación de la línea base de las oscilaciones respecto del chorro sin perturbar ($r=1$).

7 Conclusiones y trabajos futuros

Las simulaciones llevadas a cabo con *openFOAM* demuestran la posibilidad de realizar simulaciones complejas de carácter científico con programas de CFD convencionales, dando resultados precisos a un costo computacional razonable. No obstante, la principal dificultad radica en aprender a usar el programa, conocer su entorno y saber diferenciar entre aquellos resultados acordes a la realidad y aquellos que se alejan considerablemente de lo esperado. No obstante, este trabajo supone el punto de partida a realizar otros estudios con este software, explorar su vasto repositorio de utilidades y exprimir al máximo sus capacidades estudiando otros sistemas complejos que puedan tener interés industrial o científico.

Con la ayuda de *openFOAM* se ha podido validar el modelo teórico de chorro desarrollado por Guerrero et al., descrito en su trabajo [8], quien también ha participado activamente en la redacción del capítulo 6. Se observa un ajuste preciso de las predicciones teóricas y los resultados numéricos, correspondiéndose las dependencias observadas en las simulaciones con aquellas descritas por la teoría. Dichas dependencias pueden resumirse en los siguientes puntos:

- **Dependencias con el número de Weber**

- LONGITUD DE ONDA. A medida que el número de Weber aumenta, la longitud de onda de la forma estática del chorro disminuye.
- TASA DE DECAIMIENTO. La atenuación de la forma estática del chorro se acentúa a medida que el número de Weber crece.
- MODOS INERCIALES. Los modos inerciales aumentan en amplitud a medida que el número de Weber aumenta.

- **Dependencia con el número de Ohnesorge**

El número de Ohnesorge del líquido hace que la tasa de decaimiento crezca más rápidamente.

Estos fenómenos se recogen en la función que describe teóricamente el chorro, definida en la ecuación (6.4), que describe con gran precisión las arrugas originadas en el sistema.

Si bien *openFOAM* es un software útil y versátil, capaz de simular sistemas complejos como el mostrado en este estudio, muestra ciertas desventajas que pueden sugerir usar otros softwares. Por una parte, el tiempo de simulación es excesivo con respecto a otros softwares como *Basilisk*, capaz de simular el mismo sistema bajo las mismas condiciones en un tiempo considerablemente menor (una simulación en *Basilisk* puede tomar treinta minutos mientras que una simulación en *openFOAM* toma aproximadamente doce horas). Este hecho puede deberse al *solver* elegido (*interFOAM*) o al mallado, aunque la diferencia de tiempos, previsiblemente, no se vería reducida en gran medida

modificando estos elementos.

Por otro lado, no ha sido posible simular varios casos por la presencia de inestabilidades ($We = 8$ y $We = 8.5$ para $Oh = 0.003$), no dándose este fenómeno en *Basilisk*. De nuevo, el *solver* puede ser el causante de esto, aunque no se le ha podido encontrar solución en el presente trabajo.

Con el fin de afianzar los resultados y asegurar la fiabilidad de *openFOAM*, se establece como trabajo futuro la realización de las simulaciones con *Basilisk*, software robusto y de gran fiabilidad que puede ayudar a establecer a *openFOAM* como alternativa para hacer estudios numéricos de sistemas complejos.

También se propone incluir la gravedad en las simulaciones y estudiar su efecto en el sistema, comparando los resultados obtenidos en este trabajo y los obtenidos teniendo en cuenta este factor. Además, incluir la gravedad permitirá también comprobar la exactitud de los resultados con aquellos obtenidos experimentalmente en la obra de Hancock & Bush, *Fluid Pipes* [9], validando así el software empleado.

Los resultados presentados en las anteriores secciones demuestran la utilidad del software y las posibilidades que brinda, pudiendo ser utilizado por cualquier usuario interesado al ser un software libre y existir una gran cantidad de contenido de formación (tutoriales, foros, investigaciones...). Además, puede asentar las bases del conocimiento sobre la Mecánica de Fluidos Computacional y allanar el aprendizaje de otros softwares más potentes enfocados en la industria como *Ansys Fluent* o *Simcenter Star-CCM+*.

Índice de Figuras

2.1	Experimento llevado a cabo en [9] en el que se basarán la geometría y el modelo a implementar en el estudio numérico	3
2.2	Esquemático de la geometría del modelo	4
3.1	Diagrama de flujo del algoritmo PIMPLE	11
3.2	Comparativa entre las mallas 1 y 5	13
3.3	Interfases obtenidas para Δt adaptativo y Δt fijo, fijados $We = 15$ y $Oh = 0.003$	14
3.4	Resultados obtenidos con Δt fijo y Δt adaptativo frente a resultados obtenidos con un mallado muy fino para $We = 15$ y $Oh = 0.003$	15
3.5	Resultados obtenidos con los mallados 1 y 2 frente a los obtenidos para un mallado muy fino de 134400 elementos para $We = 15$ y $Oh = 0.003$	16
3.6	Resultados obtenidos con los mallados 3, 4 y 5 frente a los obtenidos para un mallado muy fino de 134400 elementos para $We = 15$ y $Oh = 0.003$	17
3.7	Diferencias relativas medias entre los puntos de la interfase obtenidas con los mallados propuestos.	17
4.1	Esquema de la geometría definida en el archivo <i>blockMeshDict</i>	28
4.2	Principales utilidades de <i>ParaView</i>	34
4.3	Cómo obtener los puntos de la interfase con <i>ParaView</i>	35
4.4	Representación de la interfase con <i>MATLAB</i>	36
4.5	Representación de la interfase con <i>Gnuplot</i>	36
5.1	Interfases obtenidas para $We = 9$ y $We = 9.5$	38
5.2	Interfases obtenidas para $We = 10$ y $We = 10.5$	38
5.3	Interfases obtenidas para $We = 11$ y $We = 11.5$	39
5.4	Interfases obtenidas para $We = 12$ y $We = 12.5$	39
5.5	Interfases obtenidas para $We = 13$ y $We = 13.5$	40
5.6	Interfases obtenidas para $We = 14$, $We = 14.5$ y $We = 15$	40
5.7	Comparativa entre las interfases obtenidas para $We = 9$ y $We = 15$	41
5.8	Interfases obtenidas para $We = 8$ y $We = 8.5$	42
5.9	Interfases obtenidas para $We = 9$ y $We = 9.5$	42
5.10	Interfases obtenidas para $We = 10$ y $We = 10.5$	43
5.11	Interfases obtenidas para $We = 11$ y $We = 11.5$	43
5.12	Interfases obtenidas para $We = 12$ y $We = 12.5$	44
5.13	Interfases obtenidas para $We = 13$, $We = 13.5$	44
5.14	Interfases obtenidas para $We = 14$, $We = 14.5$ y $We = 15$	45

- 5.15 Arrugas obtenidas para $We = 13$ para los casos $Oh = 0.003$ y $Oh = 0.01$. Puede apreciarse claramente cómo la tasa de decaimiento en el caso de $Oh = 0.01$ es notablemente mayor, atenuándose en gran medida la onda originada 45
- 5.16 Arrugas obtenidas para $We = 9$ y $We = 15$. Puede apreciarse la mayor dependencia con respecto al caso de $Oh = 0.003$ de la longitud de onda y la tasa de decaimiento con el número de Weber. 46
- 6.1 Comparativa de las arrugas obtenidas numéricamente (puntos) y las obtenidas por el modelo teórico (línea continua), donde se ha colocado el obstáculo a la izquierda ($x = 0$) y se ha restado la forma no perturbada del chorro ($r = 1$). 48

Bibliografía

- [1] Paraview, <https://www.paraview.org/download/>.
- [2] Rutina `NonlinearModelFit` en Mathematica, <https://reference.wolfram.com/language/ref/NonlinearModelFit.html>.
- [3] J. N. Anno, *The mechanics of liquid jets*, Lexington, Mass., D. C. Heath and Co., 1977. 118 p., January 1977.
- [4] S Márquez Damian, *Description and utilization of interfoam multiphase solver*, International Center for Computational Methods in Engineering (2012), 1–64.
- [5] Pierre Simon de Laplace, *Sur l'action capillaire*, 1806.
- [6] JM Gordillo and M Pérez-Saborid, *Aerodynamic effects in the break-up of liquid jets: on the first wind-induced break-up regime*, *Journal of Fluid Mechanics* **541** (2005), 1–20.
- [7] JM Gordillo, M Pérez-Saborid, and AM Gañán-Calvo, *Linear stability of co-flowing liquid–gas jets*, *Journal of Fluid Mechanics* **448** (2001), 23–51.
- [8] J Guerrero, H González, and FJ García, *Spatial modes of capillary jets, with application to surface stimulation*, *Journal of fluid mechanics* **702** (2012), 354–377.
- [9] Matthew J Hancock and John WM Bush, *Fluid pipes*, *Journal of Fluid Mechanics* **466** (2002), 285–304.
- [10] Cyril W Hirt and Billy D Nichols, *Volume of fluid (vof) method for the dynamics of free boundaries*, *Journal of computational physics* **39** (1981), no. 1, 201–225.
- [11] John William Strutt Baron Rayleigh, *The theory of sound*, vol. 2, Macmillan, 1896.
- [12] Lord Rayleigh Sec. R.S., *Xvi. On the instability of a cylinder of viscous liquid under capillary force*, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **34** (1892), no. 207, 145–154.
- [13] Félix Savart, *Mémoire sur la constitution des veines liquides lancées par des orifices circulaires en mince paroi*, *Ann. de chim.* **53** (1833), 337–386.
- [14] Jörn WP Schmelzer, Edgar D Zanutto, and Vladimir M Fokin, *Pressure dependence of viscosity*, *The Journal of chemical physics* **122** (2005), no. 7, 074511.
- [15] Thomas Young, *An essay on the cohesion of fluids*, *Philosophical Transactions of the Royal Society of London* **95** (1805), 65–87.