

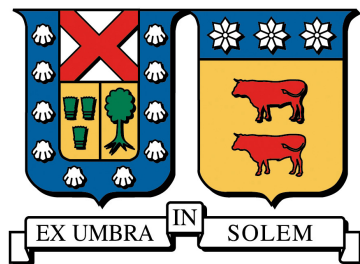
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INGENIERIA MECÁNICA
VALPARAISO- CHILE

**”SIMULACIÓN NUMÉRICA DE UN FLUJO
AGUA-PETROLEO EN UN INYECTOR DE
TURBINA A GAS”**

Nicolás Ignacio Thiers Moggia

Magíster en Ciencias de la Ingeniería Mecánica

Julio - 2016



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INGENIERÍA MECÁNICA
VALPARAISO- CHILE

**”SIMULACIÓN NUMÉRICA DE UN FLUJO AGUA-PETROLEO EN UN
INYECTOR DE TURBINA A GAS”**

NICOLÁS IGNACIO THIERS MOGGIA

Tesis de grado para optar al grado de:
Magíster en Ciencias de la Ingeniería Mecánica
y al título de:
Ingeniero Civil Mecánico

Profesor Guía: Dr. Ing. Romain Gers

Profesor Correferente: Dr. Ing. Olivier Skurtys

Profesor Correferente externo: Dr. Ing. Juan Carlos Elicer

Julio - 2016

TITULO DE LA TESIS:

**”SIMULACIÓN NUMÉRICA DE UN FLUJO AGUA-PETROLEO EN UN INYECTOR
DE TURBINA A GAS”**

AUTOR:

Nicolás Ignacio Thiers Moggia

TRABAJO DE TESIS, presentado en cumplimiento parcial de los requisitos para el Grado de Magíster en Ciencias de la Ingeniería Mecánica de la Universidad Técnica Federico Santa María.

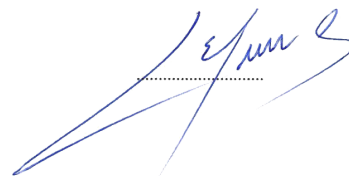
Dr. Ing. Romain Gers



Dr. Ing. Olivier Skurtys



Dr. Ing. Juan Carlos Elicer



VALPARAISO, CHILE, Julio - 2016

*Recuerde el alma dormida,
avive el seso y despierte
contemplando
cómo se pasa la vida,
cómo se viene la muerte
tan callando,
cuán presto se va el placer;
cómo, después de acordado,
da dolor;
cómo, a nuestro parecer,
cualquiera tiempo pasado
fue mejor.*

Agradecimientos

A mis padres por el cariño, la paciencia, el amor y el apoyo incondicional para seguir mis sueños y brindarme una educación. A mi familia por el amor y cariño en todos estos años. A mis amigos por los miles de momentos de alegrías y penas que compartimos en esta etapa y finalmente a mi amada esposa por todos estos años de amor y apoyo incondicional en los momentos mas difíciles, nada de esto habría sido posible sin ti.

Al CCT-Val y al profesor Yuri Ivanov por facilitar acceso al centro de computación de alto rendimiento para poder realizar las simulaciones.

A CONICYT Chile por apoyar mis estudios de postgrado.

CONICYT-PCHA/MagisterNacional/2014-66201.

Abstract

In order to decrease the total emissions of NO_x, gas turbines has the possibility to use a mixture of fuel with water, this way can reach lower combustion temperature and colder exhaust gases. Has been noticed the presence of termoacoustic instabilities in the combustion chamber when this mixture is used, resulting in serious structural damage in the unit. At the beginning, methodology for multiphase simulation and disperse turbulents multiphase models are reviewed. Then the mathematical basis is introduced: Navier-Stokes equations for modeling incompressible flows, Turbulence modelling using Large-Eddy simulation and sub-grid model for kinetic turbulent energy, energy transport equation for model the internal temperature distribution and finally volume fraction equations used in “Volume of fluids” multiphase model. Next the Finite Volume Method (FVM) is introduced in order to discretize this set of partial differential equations and algorithms to solve the corresponding matricial equations. Next chapter presents the problem of interest treated in this work, physical dimensions and the standard operating conditions of the injector. The main goals of the work are to study and fully understand this phenomenon through biphasic numerical simulation, immiscible two-liquid flow. To make this, a code was developed by modifying preexisting multiphase solver included in OpenFOAM and validated later using two cases: first, the heat exchange across circular pipes and second, Rayleigh-Bénard instability. The obtained results of the simulations allow to determine stagnation zones where water phase tend to acumulate, this zones are dependent on the inlet flows conditions, moving upstream when the oil flux is increased and growing when the water flux raise. Temperature fields also seems to be influenced by the inlets flows, the lowest temperatures occur near the axial axis and the maximum temperatures are developed on the plane of 45° degress, increasing the inlet temperature at 18 %. Finally, spectral analysis was performed in order to determine the main frequency

of oscillation in the pressure field inside the mix chamber. It concludes that the instability can't be explained only taking accounts the hidroynamics conditions,being necessary to consider the process of combustion of the mixture.

Resumen

Con el objetivo de disminuir las emisiones de NOx térmico y cumplir así con las nuevas normas de emisiones de contaminantes para las centrales térmicas, una de las alternativas que se presentan corresponde a realizar un pre-mezclado entre el combustible y agua con el fin que esta absorba parte de la energía liberada durante la combustión, disminuyendo la temperatura de la combustión y los gases de escape e inhibiendo el mecanismo de formación de NOx térmico. Pero se ha observado que este proceso genera una inestabilidad termoacústica en la cámara de combustión, la cual genera daños sobre el inyector provocando su destrucción. En el presente trabajo se presenta una simulación numérica del proceso de mezclado de ambos fluidos al interior de la cámara de pre-mezclado del inyector para distintos caudales mediante la implementación de un código basado en el software libre OpenFOAM para flujos turbulentos multifásicos acoplado a la ecuación de energía. se analizó las estructuras coherentes del campo turbulento desarrollado así como también se realizó una descripción de la morfología de la fase dispersa.

Índice general

Agradecimientos	9
Abstract	11
Resumen	13
1. Introducción	31
2. Estado del Arte	35
2.1. Simulación de flujos multifásicos	35
2.1.1. Direct Numerical Simulation (DNS)	35
2.1.2. Modelo Elemento de Fase Dispersa Discreto (DPE)	36
2.1.3. Two-Fluid Model	37
2.2. Turbulencia en flujos multifásicos	41
2.2.1. Enfoque Dusty Gas y equilibrio Euleriano	41
2.2.2. Enfoque Euleriano	43
2.2.3. Lagrangian Point-Particle	43
2.2.4. Enfoque Completamente Resuelto (Fully Resolved)	43
3. Marco Teórico	47
3.1. Modelación Matemática	47
3.1.1. Ley de Conservación	47
3.1.2. Conservación de Masa	48
3.1.3. Conservación de Momentum	50
3.1.4. Conservación de Energía	52

3.1.5.	Método de Volúmenes de Fluido VOF	53
3.1.6.	Modelación de la turbulencia	54
4.	Dinámica de Fluidos Computacional	67
4.1.	Método de Volúmenes Finitos	67
4.1.1.	Aproximación Integrales de Superficie	69
4.1.2.	Aproximación Integrales de Volúmenes	73
4.1.3.	Aproximación Derivada Temporal	73
4.1.4.	Estabilidad Numérica	74
4.1.5.	Condiciones de contorno	75
4.2.	Métodos de Resolución de Sistemas Lineales	78
4.2.1.	Método del Gradiente Conjugado	78
4.2.2.	Método Gauss-Seidel	80
4.2.3.	Método Multi-Malla (MGM)	81
5.	Simulación Numérica	85
5.1.	OpenFOAM	85
5.2.	Código	87
5.3.	Validación del código	92
5.3.1.	Primer caso de validación del código multifásico: "Zalesak Slotted disk problem", Tony W.H. Sheu *, C.H. Yu, P.H. Chiu JCP (2009)	92
5.3.2.	Segundo caso de validación del código multifásico: Deformación de una esfera tridimensional, Douglas Enright et. al, 2002	93
5.3.3.	Tercer caso de validación del código multifásico: Inestabilidad de Plateau-Rayleigh en un jet laminar	94
5.3.4.	Primer caso de validación para el código modificado: Transferencia de calor en tubería	95
5.3.5.	Segundo caso de validación para el código modificado: Convección Rayleigh-Bénard	102
5.4.	Problema de estudio	106
5.5.	Dominio Físico y Discretización Espacial	109
5.6.	Condiciones de contorno e inicial	114
5.6.1.	Entrada de Petróleo	115

5.6.2. Entrada de Agua	117
5.6.3. Pared	118
5.6.4. Salida	120
5.6.5. Simetría	121
5.6.6. Condiciones iniciales	121
5.7. Casos de estudio	122
5.7.1. Estructura de los archivos	124
5.7.2. Hardware utilizado para la simulación y tiempos de cálculo	124
6. Resultados	127
6.1. Simulación petróleo con agua	128
6.2. Simulación solo petróleo	142
6.3. Simulaciones con variación de los caudales	147
6.3.1. Variación del flujo de petróleo	147
6.3.2. Variación flujo agua	168
6.4. Análisis espectral	190
6.5. Presión en la alimentación de agua	195
7. Conclusiones y trabajos futuros	197
A. Código desarrollado	201
B. Malla quemador tridimensional	255
C. Casos simulados	277
C.1. Caso Control	277
C.1.1. 0/alpha.oil	277
C.1.2. 0/alpha.water	280
C.1.3. 0/alphas	282
C.1.4. 0/k	284
C.1.5. 0/nuSgs	286
C.1.6. 0/p_rgh	288
C.1.7. 0/T	291
C.1.8. 0/U	293

C.1.9. constant/LESProperties	295
C.1.10. constant/transportProperties	298
C.1.11. system/controlDict	300
C.1.12. system/fvSchemes	304
C.1.13. system/fvSolution	306
C.1.14. system/setFieldsDict	310
C.2. Sin flujo de agua	312
C.2.1. 0/U	312
C.3. Variación flujo petroleo: Re 10.000	314
C.3.1. 0/U	314
C.4. Variación flujo petroleo: Re 20.000	316
C.4.1. 0/U	316
C.5. Variación flujo petroleo: Re 25.000	318
C.5.1. 0/U	318
C.6. Variación flujo agua: Re 2.000	320
C.6.1. 0/U	320
C.7. Variación flujo agua: Re 4.000	322
C.7.1. 0/U	322
C.8. Variación flujo agua: Re 5.000	324
C.8.1. 0/U	324

Bibliografía**326**

Índice de figuras

1.1. Aportaciones estimadas de los tres mecanismos del NO en la formación de NO_x totales en la combustión de carbon mineral (cortesía de Air and Waste Management Association)	33
2.1. Esquema Modelo Euler-Lagrange: La fase dispersa se mueve producto del arrastre que genera la fase continua sobre esta, la posición de las partículas o gotas es rastreada sobre un marco Lagrangiano mientras que la fase dispersa se resuelve sobre un marco Euleriano.	37
2.2. Esquema Modelo Euler-Euler	38
2.3. Esquema Surface Tracking Method, <i>a</i>) Rastreo de la interfaz mediante partículas sin masas y <i>b</i>) rastreo de la interfaz mediante una función de altura	39
2.4. Esquema Moving Mesh Methods	39
2.5. Tipos de esquemas “Volume Tracking Method” en los cuales las fases son marcadas por <i>a</i>) partículas trazadoras sin masa, <i>b</i>) una función de nivel o <i>c</i>) una función de fracción volumétrica	42
2.6. Set-up experimental utilizado por Djamel Lakhel et. al	44
3.1. Diagrama esquemático de la cascada de energía para un número de Reynolds muy grande, (Pope, 2000)	59
3.2. Esquema del espectro de energía de Kolmogorov en escala logarítmica y las diferentes escalas en las que se desarrolla la turbulencia	59

3.3. Campo de velocidades filtrado utilizando un filtro Gaussiano con longitud de filtro $\Delta = 0,35$. Arriba: Campo de velocidad $U(x)$ y su correspondiente campo de velocidad filtrado $\bar{U}(x)$. Abajo: campo de velocidades residual $u'(x)$ y su correspondiente campo de velocidad residual filtrada $\bar{u}'(x, t)$	61
4.1. Esquema para una malla bidimensional y nomenclatura utilizada en el Método de Volúmenes Finitos	68
4.2. Esquema de aproximación UDS de la función ϕ sobre una malla unidimensional para un flujo que se mueve en dirección x	70
4.3. Esquema de aproximación CDS de la función ϕ sobre una malla unidimensional para un flujo que se mueve en dirección x	71
4.4. Comparación entre la solución obtenida para la ecuación de advección pura unidimensional entre un esquema TVD (arriba) y uno no-TVD (abajo).	72
4.5. Representación de un dominio Ω de contorno Γ donde se busca la solución de una EDP.	76
4.6. Esquema de las mallas utilizadas en el método multi-mallas para problemas unidimensionales	82
5.1. Diagrama de flujos global código MTBoussinesqIFoam	89
5.2. Diagrama de flujos del preámbulo previo al bucle de resolución temporal del código	90
5.3. Diagrama de flujos del algoritmo PIMPLE para resolución de las ecuaciones de Navier-Stokes	91
5.4. Representación de la posición de la interfaz y la influencia en la resolución de la malla para el problema “slotted disk”, a) Solución teórica del problema b) malla de 50×50 , c) malla de 100×100 y d) malla de 150×150	93
5.5. Comparación de la posición de la interfaz obtenida por el método “Level Set”(a) con la posición de la interfaz obtenida por el método “Volume of Fluid” implementado en OpenFOAM (b)	93
5.6. Comparación de la evolución temporal de la interfaz obtenida por el método “Volume of Fluid” implementado en OpenFOAM (a) con la solución predicha por el método “Level Set” (b)	94

5.7. Inestabilidad de Plateau-Rayleigh para una perturbación con razón longitud de onda sobre diámetro $k = 0,75$. <i>a)</i> perturbación inicial y <i>b)</i> interfaz luego de la desintegración resultando en una gota principal y una gota satélite	94
5.8. Comparación entre los radios de la gotas principal y satélite predicha por la teoría y lo reportado por la simulación para distintas razones de perturbación	95
5.9. Esquema solución teórica caso validación	96
5.10. Esquema simulación caso validación	98
5.11. Perfiles de temperaturas a la salida de la tubería para distintos números de Reynolds	100
5.12. Comparación entre la simulación y la solución teórica para la temperatura media de masa adimensionalizada obtenida a la salida de la tubería a distintos numeros de Reynolds simulados.	101
5.13. Error relativo porcentual cometido en la simulación en comparación con la solución empírica.	101
5.14. Esquema de la celda de convección Rayleigh-Bénard	104
5.15. Perfil de velocidades en dirección x sobre $x = 0,5$, en negro los resultados reportados por Ouertatani et al, 2008 y en color los resultados obtenidos con el código modificado.	105
5.16. Perfil de velocidades en dirección y sobre $y = 0,5$, en negro los resultados reportados por Ouertatani et al, 2008 y a color los resultados obtenidos con el código modificado.	106
5.17. Distribución de temperaturas al interior de la cavidad. A la izquierda los resultados reportados por Ouertatani et al, 2008 y a la derecha los obtenidos en la simulación para <i>a)</i> $Ra = 10^4$, <i>b)</i> $Ra = 10^5$, <i>c)</i> $Ra = 10^6$.	107
5.18. Numero de Nusselt local sobre la pared inferior $y = 0$ reportado por <i>a)</i> Ouertatani et al, 2008 y <i>b)</i> resultados obtenidos con el código.	108
5.19. Vista en corte del inyector de la turbina a gas. En verde la línea de admisión de petróleo, en azul la línea de admisión de agua, en rojo la cámara de pre-mezclado y en gris la línea auxiliar de gas	110
5.20. Vista en explosión del inyector de la turbina a gas	110

5.21. Plano del inyector de la turbina a gas estudiado (dimensiones en milímetros)	111
5.22. Comparación de la solución obtenida para una malla de <i>a)</i> 200.000 volúmenes y <i>b)</i> 600.000 volúmenes	111
5.23. Malla tridimensional con cuasi-simetría axial	112
5.24. Malla tridimensional con cuasi-simetría axial	113
5.25. Representación de la frontera correspondiente a la entrada para el caudal de petróleo.	115
5.26. Ilustración del método de simulación precursora (Stevens et al. 2014b), a la izquierda el dominio de la simulación precursora y a la derecha el dominio de la simulación real, la región demarcada sobre el dominio precursor es inyectada sobre el dominio de la simulación real	116
5.27. Turbulencia sintética como condición de entrada para una simulación Large-Eddy con coherencia espacial y su evolución en el tiempo para <i>a)</i> $t = 0,001$, <i>b)</i> $t = 0,002$ y <i>c)</i> $t = 0,003$ segundos (Jørgensen,2012)	116
5.28. Representación de la frontera correspondiente a la entrada para el caudal de agua	117
5.29. Representación de la frontera correspondiente a las paredes del inyector .	118
5.30. Representación de la frontera correspondiente a la salida del inyector . . .	120
5.31. Representación de la frontera correspondiente a los planos de simetría . .	121
5.32. Representación esquemática de la estructura de los archivos necesarios para realizar cada simulación	125
6.1. Líneas de flujo al interior de la cámara de mezclado para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$	130
6.2. Campo de velocidades promedio para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en <i>a)</i> 0 grados, <i>b)</i> 22,5 grados, <i>c)</i> 45 grados, <i>d)</i> 67,5 grados y <i>e)</i> 90 grados	131
6.3. Campo de velocidades promedio para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en los radios <i>a)</i> 0.2, <i>b)</i> 0.4, <i>c)</i> 0.6 y <i>d)</i> 0.8	132
6.4. Campo de Temperaturas promedio adimensionalizado para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en <i>a)</i> 0 grados, <i>b)</i> 22,5 grados, <i>c)</i> 45 grados, <i>d)</i> 67,5 grados y <i>e)</i> 90 grados	133

6.5. Distribución del campo de temperatura promedio para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en los radios a) 0.2, b) 0.4, c) 0.6 y d) 0.8	134
6.6. Distribución espacial promedio de la fracción volumétrica de la fase petróleo para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en a) 0 grados, b) 22,5 grados, c) 45 grados, d) 67,5 grados y e) 90 grados	135
6.7. Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$	136
6.8. Distribución del tamaño máximo de gotas de la fase dispersa según el criterio del numero de Weber critico para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$	137
6.9. Visualización de los vórtices del campo de velocidades medio utilizando el criterio Q de detección de vórtices para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$	138
6.10. Visualización de los vórtices del campo de velocidades medio utilizando el criterio λ_2 de detección de vórtices para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$	139
6.11. Visualización de los vórtices mediante el criterio Q para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en los radios a) 0.2, b) 0.4, c) 0.6 y d) 0.8	140
6.12. Visualización de los vórtices mediante el criterio Q para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en a) 0 grados, b) 22,5 grados, c) 45 grados, d) 67,5 grados y e) 90 grados	141
6.13. Campo de velocidades promedio para la simulación con $Re_{agua} = 0$ y $Re_{petroleo} = 15000$ en cortes longitudinales en a) 0 grados, b) 22,5 grados, c) 45 grados, d) 66,5 grados y e) 90 grados	143
6.14. Campo de velocidades promedio para la simulación con $Re_{agua} = 0$ y $Re_{petroleo} = 15000$ en cortes longitudinales en los radios a) 0.2, b) 0.4, c) 0.6 y d) 0.8	144

6.15. Distribución temperatura promedio adimensionalizada para la simulación con $Re_{agua} = 0$ y $Re_{petroleo} = 15000$ en cortes longitudinales en a) 0 grados, b) 22,5 grados, c) 45 grados, d) 66,5 grados y e) 90 grados	145
6.16. Distribución del campo de temperatura promedio adimensionalizada para la simulación con $Re_{agua} = 0$ y $Re_{petroleo} = 15000$ en cortes longitudinales en los radios a) 0.2, b) 0.4, c) 0.6 y d) 0.8	146
6.17. Líneas de flujo al interior de la cámara para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$	147
6.18. Líneas de flujo al interior de la cámara para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 20000$	148
6.19. Líneas de flujo al interior de la cámara para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 25000$	149
6.20. Campo de velocidad promedio en corte longitudinal a 0 grados para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	151
6.21. Campo de velocidad promedio en corte longitudinal a 22,5 grados para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	152
6.22. Campo de velocidad promedio en corte longitudinal a 45 grados para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	153
6.23. Campo de velocidad promedio en corte longitudinal sobre el radio 0.2 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	153
6.24. Campo de velocidad promedio en corte longitudinal sobre el radio 0.4 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	154
6.25. Campo de velocidad promedio en corte longitudinal sobre el radio 0.6 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	154

6.26. Campo de velocidad promedio en corte longitudinal sobre el radio 0.8 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	155
6.27. Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de petroleo sobre el plano $r = 0,2$	155
6.28. Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de petroleo sobre el plano $r = 0,8$	156
6.29. Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de petroleo sobre el plano $\theta = 0$	156
6.30. Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de petroleo sobre el plano $\theta = 45$	157
6.31. Campo de Temperaturas promedio adimensionalizada en corte longitudinal a 0 grados para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$, c) $Re_{petroleo} = 25000$	157
6.32. Campo de Temperaturas promedio adimensionalizada en corte longitudinal a 22,5 grados para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$, c) $Re_{petroleo} = 25000$	158
6.33. Campo de Temperaturas promedio adimensionalizada en corte longitudinal a 45 grados para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$, c) $Re_{petroleo} = 25000$	159
6.34. Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.2 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	159
6.35. Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.4 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	160
6.36. Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.6 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	160
6.37. Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.8 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	161

6.38. Campo de fracción volumétrica promedio de petróleo en corte longitudinal a 0 grados para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$, c) $Re_{petroleo} = 25000$	161
6.39. Campo de fracción volumétrica promedio de petróleo en corte longitudinal a 22,5 grados para la simulación con $Re_{agua} = 3,000$ y a) $Re_{petroleo} = 10,000$, b) $Re_{petroleo} = 20,000$, c) $Re_{petroleo} = 25,000$	162
6.40. Campo de fracción volumétrica promedio de petróleo en corte longitudinal a 45 grados para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$, c) $Re_{petroleo} = 25000$	163
6.41. Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 10000$	164
6.42. Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 20000$	165
6.43. Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 25000$	166
6.44. Distribución de la fracción volumétrica de petróleo al interior de la cámara de mezcla para $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 15000$, c) $Re_{petroleo} = 20000$, d) $Re_{petroleo} = 25000$	167
6.45. Distribución del tamaño máximo de gotas de la fase dispersa según el criterio del número de Weber crítico para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 15000$, c) $Re_{petroleo} = 20000$, d) $Re_{petroleo} = 25000$	167
6.46. Líneas de flujo al interior de la cámara para las simulación con $Re_{petroleo} = 15000$ y $Re_{agua} = 2000$	168
6.47. Líneas de flujo al interior de la cámara para las simulación con $Re_{petroleo} = 15000$ y $Re_{agua} = 4000$	169
6.48. Líneas de flujo al interior de la cámara para las simulación con $Re_{petroleo} = 15000$ y $Re_{agua} = 5000$	170
6.49. Campo de velocidad promedio en corte longitudinal a 0 grados para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	172

6.50. Campo de velocidad promedio en corte longitudinal a 22,5 grados para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	173
6.51. Campo de velocidad promedio en corte longitudinal a 45 grados para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	173
6.52. Campo de velocidad promedio en corte longitudinal sobre el radio 0.2 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	174
6.53. Campo de velocidad promedio en corte longitudinal sobre el radio 0.4 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	174
6.54. Campo de velocidad promedio en corte longitudinal sobre el radio 0.6 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	175
6.55. Campo de velocidad promedio en corte longitudinal sobre el radio 0.8 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	175
6.56. Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de agua sobre el plano $r = 0,2$	176
6.57. Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de agua sobre el plano $r = 0,8$	176
6.58. Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de agua sobre el plano $\theta = 0$	177
6.59. Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de agua sobre el plano $\theta = 45$	177
6.60. Campo de Temperatura adimensionalizada promedio en corte longitudinal a 0 grados para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	178
6.61. Campo de Temperatura adimensionalizada promedio en corte longitudinal a 22,5 grados para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	179

6.62. Campo de Temperatura adimensionalizada promedio en corte longitudinal a 45 grados para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	180
6.63. Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.2 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	180
6.64. Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.4 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	181
6.65. Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.6 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	181
6.66. Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.8 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	182
6.67. Campo de fracción volumétrica promedio de petroleo en corte longitudinal a 0 grados para la simulación con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$, c) $Re_{agua} = 5000$	182
6.68. Campo de fracción volumétrica promedio de petroleo en corte longitudinal a 22,5 grados para la simulación con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$, c) $Re_{agua} = 5000$	183
6.69. Campo de fracción volumétrica promedio de petroleo en corte longitudinal a 45 grados para la simulación con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$, c) $Re_{agua} = 5000$	184
6.70. Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 2000$ y $Re_{petroleo} = 15000$	185
6.71. Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 4000$ y $Re_{petroleo} = 15000$	186
6.72. Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 5000$ y $Re_{petroleo} = 15000$	187

6.73. Distribución de la fracción volumétrica de petróleo al interior de la cámara de mezcla para $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 3000$, c) $Re_{agua} = 4000$, d) $Re_{agua} = 5000$	188
6.74. Distribución del tamaño máximo de gotas de la fase dispersa según el criterio del número de Weber crítico para la simulación con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 3000$, c) $Re_{agua} = 4000$, d) $Re_{agua} = 5000$	188
6.75. Variación del tamaño de gotas de la fase dispersa a distinta relación entre las velocidades del caudal de agua y petróleo	189
6.76. Espectro de frecuencias para la oscilación de la presión a la salida del inyector en la posición axial para las distintas simulaciones: a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	191
6.77. Espectro de frecuencias para la oscilación de la presión a la salida del inyector en la posición axial para las distintas simulaciones: a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	192
6.78. Espectro de energía del campo turbulento al interior de la cámara de mezclado para los distintos casos simulados: a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$	193
6.79. Espectro de energía del campo turbulento al interior de la cámara de mezclado para los distintos casos simulados: a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$	194
6.80. Variación de la presión en la línea de alimentación del agua con respecto al caso de control para distintas relaciones de velocidad entre las fases. . .	195

Índice de cuadros

1.1. Límites de emisión para centrales termoeléctricas existentes en $[mg/Nm^3]$ establecidos en el decreto N^o 13 de la norma de emisión para centrales termoeléctricas	34
1.2. Límites de emisión para centrales termoeléctricas nuevas en $[mg/Nm^3]$ establecidos en el decreto N^o 13 de la norma de emisión para centrales termoeléctricas	34
5.1. Parámetros físicos utilizados en la simulación de validación	99
5.2. Dimensiones físicas del dominio utilizado en la validación	99
5.3. Cuadro resumen de los casos estudiados y de los parámetros que se variaron.	123
5.4. Parámetros físicos utilizados en la simulación.	123
5.5. Valor de la Tensión interfacial para la interfaz agua-petróleo.	123

Capítulo 1

Introducción

La combustión es un proceso ampliamente utilizado en el mundo moderno para la generación de energía eléctrica utilizando la energía química almacenada en los combustibles. Sin embargo este proceso conlleva la emisión de diversos productos contaminantes a la atmósfera, desde material particulado como las cenizas y el hollín, pasando por gases de efecto invernadero como el dióxido de carbono CO_2 , hasta gases nocivos para la salud de las personas como el monóxido de carbono CO , óxidos de azufre SO_x y los denominados óxidos de nitrógeno. En términos generales el concepto de óxidos de nitrógeno considera los siguientes compuestos: NO , NO_2 , N_2O_2 , N_2O_4 , N_2O , N_2O_3 y NO_3 . Los principales óxidos de nitrógeno emitidos durante la combustión de combustibles fósiles son el óxido nítrico (NO), dióxido de nitrógeno (NO_2) y óxido nitroso (N_2O), siendo el término NO_x utilizado para referirse exclusivamente a los dos primeros. Las emisiones de NO_x de la combustión son predominantes (90%) [óxido nítrico]. La formación de los NO_x responde a tres mecanismos principales: NO_x instantáneo o flash el cual se forma por la conversión del nitrógeno molecular en presencia de hidrocarburos en el frente de llama, NO_x del combustible producto de la oxidación del nitrógeno contenido en el combustible y el denominado NO_x térmico producido por la oxidación del nitrógeno contenido en el aire de la combustión. La figura 1.1 representa el comportamiento de la formación de los NO_x en función de la temperatura de los gases, siendo el mecanismo térmico el principal responsable a temperaturas elevadas.

A nivel nacional, la emisión de contaminantes para centrales térmicas esta regulada en la “Norma de Emisión para Centrales Termoeléctricas” (del Medio Ambiente, 2013)

promulgada el 18 de enero del 2011 y publicada el 23 de junio del 2011 la cual entro en vigencia el 25 de febrero del 2015. La promulgación de esta nueva norma aumento la exigencia impuesta a las centrales termoeléctricas (tabla 1.1) ya existentes e impuso una condición mayor para las centrales que ingresarán en funcionamiento como se observa en la tabla 1.2. Con el fin de cumplir con estas exigencias se hace necesario la implementación de nuevas tecnologías que permitan un funcionamiento optimo y eficiente. Estas tecnologías se pueden clasificar en medidas primarias, las cuales se enfocan en el control de la formación directamente en la cámara de combustión y secundarias como el tratamiento de los humos de combustión. Dentro de las medidas primarias se encuentran el empleo de quemadores de bajo NO_x , combustión con bajo exceso de aire, reducción del pre-calentamiento del aire e inyección en etapas del aire en la cámara de combustión mediante *Blased burner firing (BBF)*, *Burners out of service (BOOS)* y *Overfire air (OFA)*. Estas tecnologías controlan la producción limitando la presencia de nitrógeno y de oxígeno disponible en la zona de reacción. Otra alternativa para controlar la producción de NO_x consiste en inyectar agua junto con el combustible con el fin de reducir la temperatura de la combustión. Se ha observado que la combustión de esta mezcla es responsable de la generación de termo pulsaciones e inestabilidades de la combustión, producto de ondas de presión que ocasionan fallas de nivel estructural en los inyectores. Para evitar estas fallas y daño en los equipos se disminuye la potencia de operación de las turbinas a gas, lo que se traduce en perdidas monetarias importantes al no poder generar el máximo de energía posible para las instalaciones.

En una primera parte se procede a realizar una revisión al estado del arte sobre los métodos utilizados en la simulación de flujos multifásicos así como también los modelos utilizados para incluir el fenómeno de turbulencia cuando se trata de flujos multifásicos dispersos. A continuación se presenta la base matemática que sustenta dichos modelos, las ecuaciones de Navier-Stokes que permiten modelar flujos incompresibles, las ecuaciones utilizadas en la modelación Large-Eddy Simulation para flujos turbulentos, la ecuación de transporte de energía para simular la distribución de temperaturas al interior de la mezcla y finalmente la ecuación de la fracción volumétrica que permite modelar flujos multifásicos mediante el método "Volume of Fluids". Posteriormente se presenta el método de volúmenes finitos utilizado para resolver numéricamente las ecuaciones en derivadas parciales descritas en las secciones previas y los algoritmos utilizados para

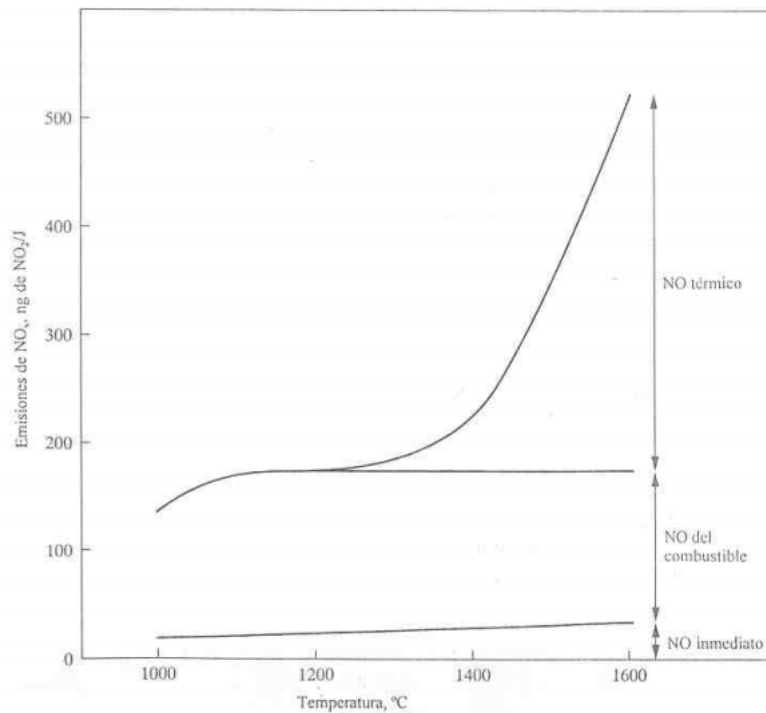


Figura 1.1: Aportaciones estimadas de los tres mecanismos del NO en la formación de NO_x totales en la combustión de carbon mineral (cortesía de Air and Waste Management Association)

la resolución de los sistemas matriciales obtenidos, mediante la discretización de las ecuaciones. En el siguiente capítulo se presenta el problema de estudio que da origen al presente trabajo, las geometrías del inyector y las condiciones de operación del mismo. El objetivo del trabajo es estudiar y entender este fenómeno mediante la simulación numérica de un flujo bifásico líquido-líquido, una emulsión acu-oleosa de agua dispersa en petróleo al interior de un inyector de turbina a gas. Se prosigue presentando el código utilizado para la simulación, el cual corresponde a una adaptación realizada al código libre OpenFOAM y su respectiva validación la que se realiza mediante una comparación con la transferencia de calor hacia el interior de una tubería y estudiando la inestabilidad de Rayleigh-Bénard.

Combustible	Material Particulado (MP)	Dióxido de azufre (SO ₂)	Óxidos de Nitrógeno (NO _x)
Sólido	50	400	500
Líquido	30	30	200
Gas	n.a.	n.a.	50

Cuadro 1.1: Límites de emisión para centrales termoeléctricas existentes en $[mg/Nm^3]$ establecidos en el decreto N° 13 de la norma de emisión para centrales termoeléctricas

Combustible	Material Particulado (MP)	Dióxido de azufre (SO ₂)	Óxidos de Nitrógeno (NO _x)
Sólido	30	200	200
Líquido	30	10	120
Gas	n.a.	n.a.	50

Cuadro 1.2: Límites de emisión para centrales termoeléctricas nuevas en $[mg/Nm^3]$ establecidos en el decreto N° 13 de la norma de emisión para centrales termoeléctricas

Capítulo 2

Estado del Arte

2.1. Simulación de flujos multifásicos

La gran mayoría de los flujos, que son de interés en el ámbito de la ingeniería, involucran la interacción entre distintas fases i.e.: transporte de sedimentos, partículas en un fluido como agua o aire como sería el transporte de material de relave producido en el proceso minero por ejemplo, bombeo de petróleo desde los pozos mediante la inyección de agua, generando una mezcla de fluidos inmiscibles, la propagación de olas en la superficie de un estanque, la atomización de combustible en la cámara de combustión, el fenómeno de cavitación producido en el rotor de las bombas.

En el último tiempo se han desarrollado múltiples métodos numéricos y distintos enfoques con el fin de poder estudiar el comportamiento de este tipo de flujos mediante la dinámica de fluidos computacional (Rusche, 2002). En el presente capítulo se hace una revisión del estado del arte y de los distintos métodos utilizados para la simulación de flujos multifásicos, en particular del tratamiento para flujos multifásicos dispersos en campos turbulentos.

2.1.1. Direct Numerical Simulation (DNS)

En la Simulación Numérica Directa (DNS por sus siglas en inglés), las ecuaciones de Navier-Stokes son resueltas directamente para todos los fluidos que interactúan en el problema. Además la posición y la topología de la interfaz son obtenidas como parte de la

solución.

Este tipo de simulaciones requiere una discretización del espacio de alto orden, con el fin de poder capturar y resolver todas las escalas del movimiento espacial y temporal que se desarrollan. Estas escalas varían desde el tamaño de los vórtices hasta las pequeñas gotas (o burbujas) que pueden provenir de una de las fases al interior de las otras. En este último caso, la discretización del dominio debe ser suficientemente pequeña para poder resolver el movimiento del fluido al interior de esta fase dispersa (gota o burbuja). Por otro lado, se debe mantener un rastreo sobre todas las porciones de la fase dispersa, lo que implica un incremento en el consumo de recursos computacionales, siendo este proporcional al número de partículas en la fase dispersa.

Debido a estas limitaciones, la simulación numérica directa está limitada a flujos con número de Reynolds relativamente bajo y un limitado número de partículas en comparación con los problemas clásicos de la ingeniería.

2.1.2. Modelo Elemento de Fase Dispersa Discreto (DPE)

En el modelo “Elemento de Fase Dispersa discreto” (Discrete DPE Model) se asume una tipología del flujo multifásico de naturaleza dispersa: Se establece una fase continua distribuida sobre todo el espacio físico y una fase dispersa la cual se mueve producto del campo de velocidad y el arrastre que genera la fase continua sobre esta fase dispersa como se ve representado en la figura 2.1. Una descripción macroscópica de la fase dispersa es obtenida utilizando una formulación discreta de las ecuaciones de conservación a nivel microscópico. El movimiento de la fase dispersa es modelado mediante una ecuación de conservación de momentum planteada sobre un marco Lagrangiano, en el cual las variables dependientes son propiedades de partículas materiales que son seguidas en su movimiento. Por otra parte, las ecuaciones de conservación para la fase continua son expresadas sobre un marco Euleriano, donde las propiedades del fluido son consideradas como funciones del espacio y del tiempo, en un marco de referencia absoluto inercial. Debido a esto es conocido también como modelo Euler-Lagrange. Cuando se trata de flujos disueltos, i.e baja fracción volumétrica y dependiendo de la precisión deseada, se puede ignorar la reacción que generan las partículas sobre la fase continua, ya sea la fuerza de reacción producto del arrastre, sustentación o momentum angular de la fase dispersa.

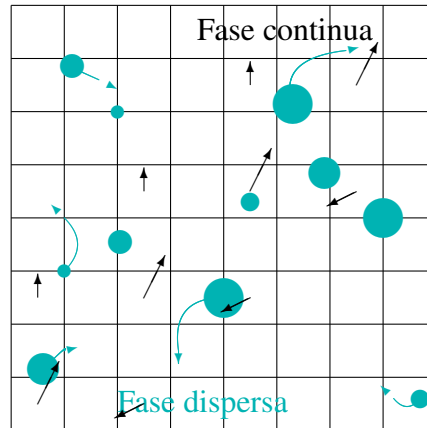


Figura 2.1: Esquema Modelo Euler-Lagrange: La fase dispersa se mueve producto del arrastre que genera la fase continua sobre esta, la posición de las partículas o gotas es rastreada sobre un marco Lagrangiano mientras que la fase dispersa se resuelve sobre un marco Euleriano.

2.1.3. Two-Fluid Model

En el modelo Two-Fluid, conocido también como modelo Euler-Euler, ambas fases son descritas, utilizando las ecuaciones de conservación expresadas sobre un marco euleriano. Ambas fases son tratadas como un continuo y son representadas por las ecuaciones de conservación promediadas. Este proceso de promediar ambas fases introduce una variable extra denominada fracción volumétrica, la cual define la probabilidad de que cierta fase esté presente en un determinado punto del espacio y del tiempo (Rusche, 2002). La figura 2.2 muestra un esquema del modelo. Las velocidades de ambas fases son representadas por los vectores y la fracción volumétrica se representa como el número en la esquina del volumen de control. Debido al proceso de promediar las ecuaciones de conservación de cada fase, se produce cierta pérdida de información. Esto se traduce en la aparición de términos adicionales en la ecuación de conservación de momentum denominado “termino medio de transferencia de momentum de la interfaz”, el cual requiere de la implementación de un modelo externo para poder cerrar el problema.

A diferencia del modelo DPE en el cual se puede optar por despreciar los efectos de la fase dispersa sobre la fase continua para los valores bajos de la fracción volumétrica, el modelo Two-Fluids incorpora por definición un acoplamiento bidireccional de las fases. Además permite trabajar con fracciones volumétricas más elevadas a un menor coste

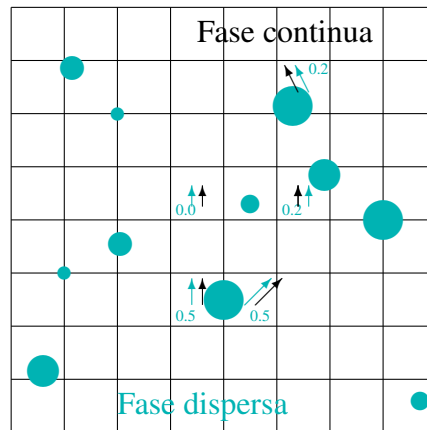


Figura 2.2: Esquema Modelo Euler-Euler

computacional.

Surface Tracking Methods

Los métodos denominados “Surface Tracking Methods” rastrean la ubicación de la interfaz a lo largo del tiempo lo cual permite mantener una interfaz aguda y precisa. Esto puede ser realizado mediante el rastreo de partículas sin masa ubicadas en la interfaz de ambas fases o mediante la evolución de una función de altura como se esquematizan en las figuras 2.3 (Ferziger y Peric, 2002), (Farmer et al., 1993), (Babu, 1994).

Moving Mesh Methods

Los métodos con malla dinámica asocian la interfaz entre los fluidos con ciertos nodos de la malla computacional. De este modo se obtiene como parte de la solución una interfaz bien definida, la cual se ubica en las caras que separan los volúmenes de control pertenecientes a los distintos fluidos, como se aprecia en la figura 2.4. Estos métodos poseen la ventaja de predecir y mantener con alta precisión la forma y la ubicación de la interfaz. Sin embargo el costo computacional resulta elevado en comparación con los demás métodos de malla estática, precisamente por el proceso de remallado que debe ser realizado cada vez que se desee actualizar la posición de la interfaz.

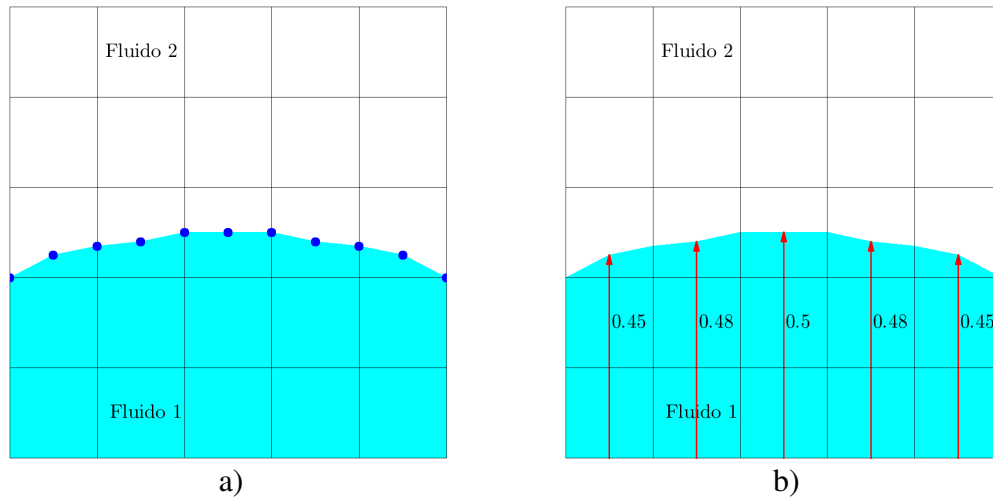


Figura 2.3: Esquema Surface Tracking Method, *a)* Rastreo de la interfaz mediante partículas sin masas y *b)* rastreo de la interfaz mediante una función de altura

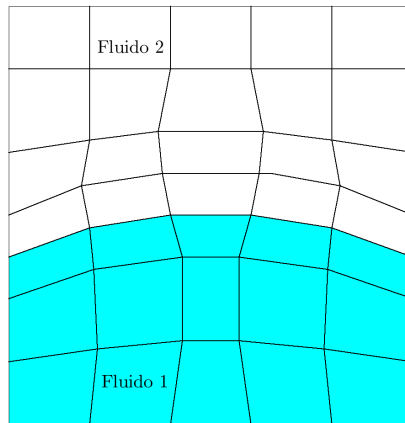


Figura 2.4: Esquema Moving Mesh Methods

Volume Tracking Method

En este tipo de métodos, se marcan y rastrean las fases sobre todo el dominio y no solo sobre la interfaz a diferencia de los métodos de superficie. Este enfoque permite mejorar los costos y tiempos de calculo en comparación con los anteriores a un costo de la disminución en la precisión de la detección de la interfaz. Existen tres tipos principales de este tipo de método los cuales se representan esquemáticamente en la figura 2.5 y se detallan a continuación:

- **Particle Marker Method** La fase es marcada mediante partículas carentes de masa las cuales son transportadas por el dominio producto del campo de velocidad del fluido, la discriminación entre fases se realiza posteriormente en función de la posición de las partículas.

- **Level Set Method** El método fue propuesto por Osher y Sethian. Se define una función Φ sobre todo el dominio. Φ mide la distancia entre la coordenada y la interfaz. Esta función evoluciona mediante su propia ecuación de conservación. La posición de la interfaz queda determinada por la isosuperficie $\Phi = 0$. Los valores positivos y negativos de la función Φ corresponden a una u otra fase respectivamente. A la fecha, distintos trabajos y modificaciones al método se han realizado. Crandall y Lions (1983,1984) trabajaron sobre la teoría y la estrategia para seleccionar la solución débil (weak solution) correcta cuando se pierde la suavidad de la función de nivel en la solución viscosa de las ecuaciones de Hamilton-Jacobi. Trabajos sobre la propagación y su vinculo con leyes de conservación de tipo hiperbólicas fueron desarrollados por Sethian (1982,1985,1987). Técnicas para transportar las funciones de nivel adyacentes a la correspondiente a la interfaz debido a la discontinuidad presente en esta fueron introducidas por Malladi et al. (1995), Zhao et al. (1996), Chen et al. (1997) y Adalsteinsson & Sethian (1999). Una revisión mas completa y detallada de este se puede encontrar en el trabajo realizado por Sethian y Smereka (Sethian y Smereka, 2003).

- **Volume of Fluid** Metodo propuesto por Hirt y Nichols en 1981 (Hirt y Nichols, 1981). La función marcadora α , denominada fracción volumétrica, determina la relación de la celda que se encuentra ocupada por una fase especifica. La evolución de la fracción

volumétrica se determina mediante la respectiva ley de conservación correspondiente a la ecuación de advección. En el capítulo siguiente se realiza una revisión más detallada de este método el cual fue utilizado en el presente trabajo.

2.2. Turbulencia en flujos multifásicos

En esta sección se realizará un repaso general de los métodos más utilizados para la simulación de flujos multifásicos dispersos en una fase continua turbulenta. Posteriormente se citarán algunos trabajos realizados en el último tiempo sobre la materia.

2.2.1. Enfoque Dusty Gas y equilibrio Euleriano

El enfoque denominado “Dusty Gas” fue propuesto en un principio por Carrier (1958) y retomado posteriormente por Marble (1970). En este enfoque se asume que las partículas son de un tamaño tan pequeño que pueden seguir perfectamente la fase continua. Se define el número de Stokes de la partícula $St = \frac{\tau_p}{\tau_k}$, el cual relaciona el tiempo de respuesta de la partícula τ_p en comparación con el tiempo característico del flujo τ_k . La hipótesis requiere un número de Stokes muy bajo ($St \ll 1$). De este modo se puede tratar el fluido junto con las partículas como un solo fluido, el cual varía su densidad en función de la fracción másica local de las partículas en suspensión. El modelo del equilibrio Euleriano conserva la simplicidad del enfoque “Dusty Gas”, pero permite que la velocidad de las partículas sea diferente a la de la fase continua que la rodea. Se asume que el movimiento de las partículas es dictado exclusivamente por el fluido que las rodea debido a su pequeño tamaño (Ferry & Balachandar 2001; Ferry et al. 2003). La velocidad de la partícula se calcula como función de la velocidad del fluido y el número de Stokes de la partícula:

$$v = u + \begin{cases} -St(1 - \beta) \frac{Du}{Dt} & |w| \ll St \\ w - St(1 - \beta) \frac{Du}{Dt} & |w| \sim O(St) \\ w - St \left[(1 - \beta) \frac{Du}{Dt} + w \nabla u \right] & |w| \sim O(1) \end{cases} \quad (2.1)$$

donde $\tau_p = \frac{d^2 \rho}{18 \nu \Phi Re}$ corresponde a la escala de tiempo propia de la partícula, τ_k es la escala de tiempo de Kolmogorov, $\beta = \frac{3}{2\rho+1}$ redefine la razón entre las densidades del fluido y

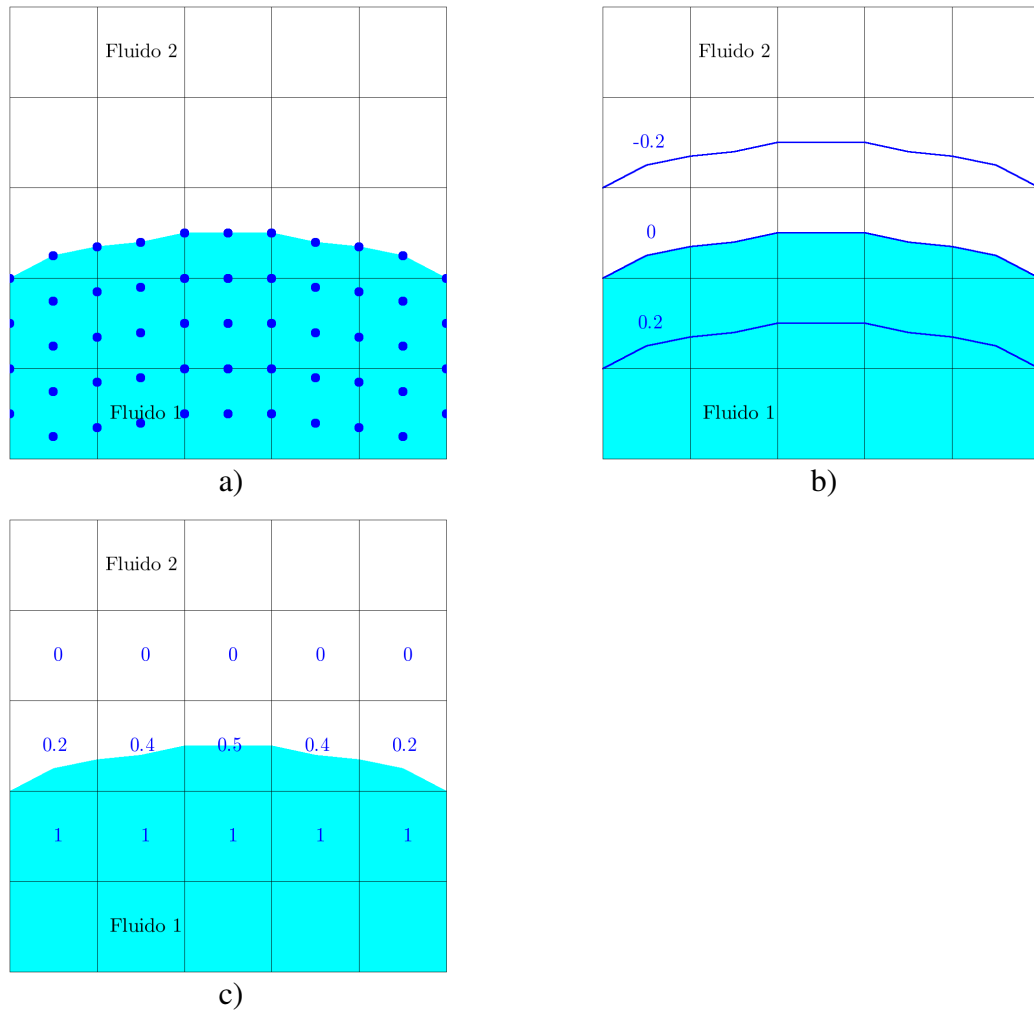


Figura 2.5: Tipos de esquemas “Volume Tracking Method” en los cuales las fases son marcadas por a) partículas trazadoras sin masa, b) una función de nivel o c) una función de fracción volumétrica

las partículas, ν es la viscosidad cinemática y d el diámetro de la partícula (Balachandar y Eaton, 2010).

2.2.2. Enfoque Euleriano

Este enfoque trata los fluidos de ambas fases como fluidos interpenetrantes. Las propiedades de la fase de las partículas son tratadas mediante una representación de campos (Crowe et al. 1998; Druzhinin & Elghobashi 1998; Fevrier et al. 2005). Esta formulación requiere ecuaciones adicionales para la fase de partículas que permite el intercambio de energía y momento con la fase continua.

2.2.3. Lagrangian Point-Particle

Corresponde al método clásico para el cálculo de flujos turbulentos multifásicos dispersos. Los primeros esfuerzos se enfocaron en resolver mediante simulación numérica directa DNS un flujo disperso con acoplamiento en un sentido (Elghobashi 1991; Maxey 1987; Riley & Patterson 1974; Squires & Eaton 1991). Sin embargo trabajos más recientes se han desarrollado para acoplamiento en dos direcciones (Elghobashi & Truesdell 1993; Ferrante & Elghobashi 2003; Squires & Eaton 1990) o cuatro direcciones si incluye la interacción entre partículas (Vance et al. 2006; Yamamoto et al. 2001) y modelación "Large Eddy Simulation" para la fase continua (Portela & Oliemans 2002; Wang & Squires 1997). En este enfoque se conserva la descripción lagrangiana de las partículas resolviendo las ecuaciones para rastrear la posición, masa, momentum y energía de estas.

2.2.4. Enfoque Completamente Resuelto (Fully Resolved)

Para partículas de tamaño del orden de la escala más pequeña de la fase continua, no es válida la aproximación de partícula a punto espacial utilizada en los métodos mencionados previamente. En este caso la única opción es realizar una simulación numérica directa completamente desarrollada (DNS Fully Resolved) en la cual todas las escalas de la turbulencia y las escalas de movimientos introducidas por las partículas son calculadas. Debido al elevado costo computacional de estas simulaciones, estas están limitadas a un número relativamente bajo de partículas para la fase dispersa (Bagchi & Balachandar

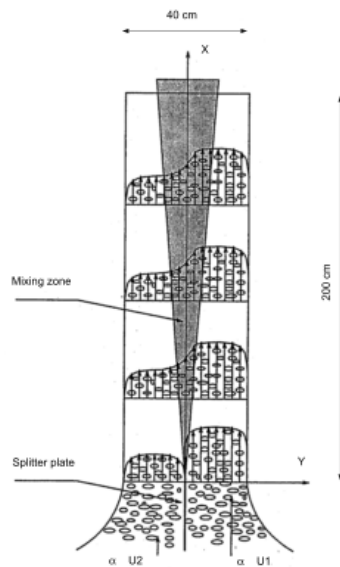


Figura 2.6: Set-up experimental utilizado por Djamel Lakhel et. al

2003, 2004; Burton & Eaton 2005; Merle et al. 2005; Zeng et al. 2008; Kajishima et al. 2001; Lu & Tryggvason 2006; Pan & Banerjee 1997; Ten Cate et al. 2004; Uhlmann 2008)

Djamel Lakehal, Brian Smith y Massimo Milelli En la presente sección se revisara el artículo “L.E.S. of a vertical Mixing Layer Laden with bubbles” (Lakehal, 2002) en el cual se presenta el avance de la aplicación de las simulaciones Large Eddy sobre capas de mezcla verticales que contienen burbujas a una baja fracción volumétrica, y el empleo de un modelo para fase dispersa de sub malla. El método utilizado se basa en las ecuaciones de flujos multifásicos del modelo “Mixture” correspondiente al promedio ponderado en función de la fracción volumétrica de las fases. El modelo de sub-malla es basado en el modelo de Smagorinsky y el modelo dinámico de Germano. Se estudió la capa de mezcla producida en un canal vertical el cual es alimentado por la parte inferior por dos caudales independientes de agua con burbujas como se muestra en la figura 2.6. La simulación se realizó utilizando una aproximación en 2 dimensiones, estableciendo una condición de deslizamiento sobre las fronteras laterales. Se estableció una condición de presión fija en la cara superior y en la cara interior. Las fases fueron inyectadas en base al perfil de velocidad medido experimentalmente con una fracción volumétrica de 1,9 % con burbujas

de 3 milímetros y velocidades promedio de $0,22\text{m/s}$ y $0,54\text{m/s}$ en los canales lento y rápido respectivamente. El modelo de sub-malla propuesto permitió representar de buena forma el flujo al interior de la capa de mezcla además de entregar una predicción bastante exacta sobre el comportamiento de la fracción volumétrica dentro de la capa de mezcla.

Capítulo 3

Marco Teórico

En esta sección se presentan las bases matemáticas sobre las cuales se sustentan los distintos modelos utilizados. Se explican en primera instancia las leyes de conservación de la masa y de la cantidad de movimiento para un elemento fluido, correspondientes a las ecuaciones de Navier-Stokes. Posteriormente se introducen las ecuaciones de conservación de la energía para un flujo newtoniano incompresible y la aproximación de Boussinesq con la cuál se busca incluir el fenómeno de flotación producido por el cambio de densidad en función de la temperatura. A continuación se revisa el modelo para flujos multifásicos “Volume of Fluid”, para finalmente presentar los fundamentos matemáticos básicos de la modelación de la turbulencia mediante el método “Large-Eddy Simulation”.

3.1. Modelación Matemática

3.1.1. Ley de Conservación

Las leyes de conservación hacen referencia a leyes físicas que estipulan que dado un sistema aislado de materia o masa de control (MC), la evolución temporal de ciertas propiedades extensivas Φ tales como la masa, momentum linear, momentum angular y energía entre otros, dependen de las fuentes de dicha propiedad dentro de la materia de control MC .

$$\frac{d(\Phi)}{dt} = \sum Source \quad (3.1)$$

Por definición, la propiedad extensiva Φ se obtiene al integrar la correspondiente propiedad intensiva ϕ sobre la masa de control MC .

$$\Phi = \int_{\Omega_{MC}} \rho \phi \, d\Omega \quad (3.2)$$

Reemplazando la ecuación 3.1 en la ecuación 3.2 se obtiene la ley de conservación para una propiedad intensiva ϕ sobre una masa de control MC .

$$\frac{d}{dt} \int_{\Omega_{MC}} \rho \phi \, d\Omega = \sum Source \quad (3.3)$$

Debido a la dificultad que representa seguir un sistema aislado de materia (MC) cuando se trata del estudio de un fluido, es preferible enfocarse en cambio en una región espacial denominada volumen de control (VC). En los flujos de la propiedad intensiva que atraviesan la frontera Ω de dicho volumen de control VC . Aplicando el teorema de transporte de Reynolds (TTR) a la ecuación 3.3 y acomodando algunos términos, es posible escribir la ecuación de conservación de una propiedad intensiva ϕ sobre un volumen de control VC :

$$\int_{\Omega_{VC}} \frac{d}{dt} \rho \phi \, d\Omega = - \int_{\Gamma_{VC}} \rho \phi (\vec{v} - \vec{v}_b) \cdot \vec{n} \, d\Gamma + \sum Source \quad (3.4)$$

Donde Ω_{VC} representa el VC , Γ_{VC} la frontera del VC , v la velocidad del fluido, v_b la velocidad de la frontera Γ y \vec{n} el vector normal de la frontera Γ .

3.1.2. Conservación de Masa

El caso particular de la Conservación de la propiedad extensiva Masa se obtiene al evaluar la ley de conservación para $\phi = 1$. Si consideramos además que no se genera masa al interior de nuestro volumen de control y que, por simplificación, este se encuentra en reposo ($\vec{v}_b = 0$), la expresión 3.4 queda:

$$\int_{\Omega_{VC}} \frac{d}{dt} \rho \, d\Omega = - \int_{\Gamma_{VC}} \rho \vec{v} \cdot \vec{n} \, d\Gamma \quad (3.5)$$

La ecuación 3.5 corresponde a la forma integral de la ecuación de conservación de la masa. El lado derecho de la ecuación 3.5 correspondiente a una integral de superficie puede ser reescrito como una integral sobre el volumen mediante el teorema de la divergencia de Gauss como sigue:

$$\int_{\Gamma_{VC}} \rho \vec{v} \cdot \vec{n} \, d\Gamma = \int_{\Omega_{VC}} \nabla \cdot (\rho v) \, d\Omega \quad (3.6)$$

Reemplazando la ecuación (3.6) en la ecuación (3.5) y agrupando convenientemente los terminos obtenemos:

$$\int_{\Omega_{VC}} \frac{d\rho}{dt} \, d\Omega + \int_{\Omega_{VC}} \nabla \cdot (\rho v) \, d\Omega = 0 \quad (3.7)$$

$$\int_{\Omega_{VC}} \left(\frac{d\rho}{dt} + \nabla \cdot (\rho v) \right) \, d\Omega = 0 \quad (3.8)$$

Dado que no se impuso alguna condición particular al VC , la ecuación 3.8 sera cierta siempre que el integrando sea idénticamente 0, es decir:

$$\frac{d\rho}{dt} + \nabla \cdot (\rho v) = 0 \quad (3.9)$$

La cual corresponde a la forma diferencial de la ley de conservación de masa, también conocida como ecuación de continuidad. En un fluido incompresible se tiene que la densidad es constante $\rho = cte$ e independiente de las demas propiedades termodinamicas del fluido:

$$\nabla \cdot v = 0 \quad (3.10)$$

Esta ecuación, desde el punto de vista de un volumen de control, establece que dado la ausencia de fuentes, el flujo neto de masa que atraviesa la frontera del volumen de control es idénticamente 0:

$$\int_{\Gamma_{VC}} \vec{v} \cdot \vec{n} \, d\Gamma = 0 \quad (3.11)$$

3.1.3. Conservación de Momentum

La ley de conservación para la cantidad de movimiento corresponde la segunda ley de Newton. La variación temporal del momentum de una MC $\int \int \int \rho \vec{v} d\Omega$ es equivalente a la sumatoria de las fuerzas externas $\sum \vec{F}$ ejercidas sobre esta:

$$\frac{d}{dt} \int_{\Omega_{MC}} \rho \vec{v} d\Omega = \sum \vec{F} \quad (3.12)$$

Esta se obtiene al evaluar la ecuación 3.4 para la propiedad intensiva del momentum $\phi = \vec{v}$:

$$\frac{d}{dt} \int_{\Omega_{VC}} \rho \vec{v} d\Omega = - \int_{\Gamma_{VC}} \rho \vec{v} (\vec{v} - \vec{v}_b) \cdot \vec{n} d\Gamma + \sum Sources \quad (3.13)$$

Los términos fuente de momentum según lo establecido por la segunda ley de Newton corresponden a las fuerzas externas que actúan sobre el volumen de control. En el estudio de la dinámica de fluidos se hace distinción entre dos grupos principales, las fuerzas de cuerpo que actúan sobre toda la materia al interior del volumen de control, las fuerzas de superficie, las que tienen influencias sobre el contorno del volumen de control. Dentro de las fuerzas de cuerpo se encuentran la atracción gravitacional, fuerzas electromagnéticas, fuerzas inerciales como la aceleración de Coriolis entre otras. Dado el problema de estudio solo se tendrá en consideración el efecto de la gravedad, por lo tanto la componente de las fuentes asociadas a las fuerzas de cuerpo $\sum \vec{F}_{body}$ queda como sigue:

$$\sum \vec{F}_{body} = \int_{\Omega_{VC}} \rho \vec{g} d\Omega \quad (3.14)$$

En el caso de las fuerzas de superficie destacan:

- la fuerza provocada por la presión del fluido
- los esfuerzos cortantes producto de la viscosidad
- los efectos de la tensión superficial e interfásial para el caso e fluidos multifásicos

$$\sum \vec{F}_{surface} = \underbrace{\int_{\Gamma_{VC}} \underline{T} \cdot \vec{n} \, d\Gamma}_{\text{Termino asociado al tensor de esfuerzos}} + \vec{F}_{\sigma} \quad (3.15)$$

donde \underline{T} representa el tensor de esfuerzos y \vec{n} el vector normal unitario de la superficie del VC , \vec{F}_{σ} la fuerza producto de la tensión superficial o interfacial. Reemplazando los términos de las ecuaciones 3.14 y 3.15 en la ecuación 3.13 se obtiene la formulación integral de la ecuación de conservación de momentum para un fluido:

$$\int_{\Omega_{VC}} \frac{d}{dt} \rho \vec{v} \, d\Omega = - \int_{\Gamma_{VC}} \rho \vec{v} (\vec{v} - \vec{v}_b) \cdot \vec{n} \, d\Gamma + \int_{\Omega_{VC}} \rho \vec{g} \, d\Omega + \int_{\Gamma_{VC}} \underline{T} \cdot \vec{n} \, d\Gamma + \vec{F}_{\sigma} \quad (3.16)$$

Para el caso de un fluido tridimensional, la ecuación vectorial (3.16) esta compuesta de tres ecuaciones independientes, las incógnitas del sistema corresponden a las tres coordenadas del vector de velocidad mas las nueve componentes del tensor de esfuerzos \underline{T} pues este corresponde a un tensor de orden dos. Para poder cerrar el sistema de ecuaciones (3.16), es necesario agregar un modelo constitutivo que permita relacionar el tensor de esfuerzos \underline{T} con el campo de velocidades del fluido. Para los fluidos Newtonianos incompresibles, el tensor de esfuerzos \underline{T} se modela como función del tensor tasa de deformación \underline{S} como sigue:

$$\underline{T} = -p\underline{I} + 2\mu\underline{S} \quad (3.17)$$

con p el campo de presión estática, \underline{I} el tensor unidad, μ la viscosidad dinámica y \underline{S} el tensor tasa de deformación tal que:

$$\underline{S} = \frac{1}{2} \left(\nabla \vec{v} + (\nabla \vec{v})^t \right) \quad (3.18)$$

Para el caso de fluidos no newtonianos el modelo constitutivo que permite relacionar el tensor de esfuerzos con el campo de velocidades adquiere distintas formas. Dependiendo de la naturaleza del fluido el comportamiento de estos puede ser plastico, pseudoplastico, dilatante, reopéctico, tixotrópico entre otros.

3.1.4. Conservación de Energía

Al realizar la aproximación de fluidos incompresibles, no se tienen en consideración los cambios de las propiedades termodinámicas producto del campo de temperaturas, así como tampoco se considerará la transformación de energía cinética en energía interna producto de la disipación de los esfuerzos viscosos. La energía interna del fluido evolucionará entonces según una ecuación de transporte simple para un escalar pasivo (Ferziger y Peric, 2002) ;(Versteeg y Malalasekera, 1995).

Para la propiedad intensiva $\phi = T$, la ecuación de conservación (3.4) se escribe como:

$$\frac{\partial}{\partial t} \int_{\Omega_{VC}} \rho T \, d\Omega + \int_{\Gamma_{VC}} \rho T \vec{v} \cdot \vec{n} \, d\Gamma = \sum Sources \quad (3.19)$$

Para el caso particular donde no se considera la transformación de cualquier tipo en energía interna, el termino fuente solo considera el flujo de energía interna por medio de la difusión térmica:

$$f_T^d = \int_{\Gamma_{VC}} D \vec{\nabla} T \cdot \vec{n} \, d\Gamma \quad (3.20)$$

donde ρ corresponde a la densidad, T la temperatura y D la difusividad térmica del fluido:

$$D = \frac{\nu}{Pr} \quad (3.21)$$

con ν viscosidad cinemática y Pr el numero de Prandtl.

Finalmente La ecuación de conservación de la energía en su forma integral se plantea como sigue.

$$\int_{\Omega_{VC}} \frac{\partial}{\partial t} \rho T \, d\Omega + \int_{\Gamma_{VC}} \rho T \vec{v} \cdot \vec{n} \, d\Gamma = \int_{\Gamma_{VC}} \frac{\mu}{Pr} \vec{\nabla} T \cdot \vec{n} \, d\Gamma \quad (3.22)$$

Aproximación de Boussinesq

La aproximación de Boussinesq permite agregar a un fluido incompresible el efecto de las fuerzas de flotación producto del efecto de la temperatura sobre la densidad. Dicha aproximación es valida siempre que el cambio de densidad sea pequeño ($\Delta\rho \ll 1$).

El efecto de la expansión volumétrica del fluido producto de la variación de la temperatura

puede ser aproximado como una función lineal del coeficiente de expansión volumétrica β mediante la siguiente ecuación (Wirth, 2014):

$$\frac{\rho}{\rho_0} = 1 - \beta (T - T_0) \quad (3.23)$$

ρ_0 corresponde a la densidad del fluido a temperatura T_0 .

Esta densidad $\rho = \rho(T)$ se reemplaza en la ecuación de conservación de cantidad de movimiento (3.16) en el termino correspondiente a la fuerza de cuerpo $\rho \vec{g}$.

3.1.5. Método de Volúmenes de Fluido VOF

La modelación del flujo multifásico se realizó mediante el enfoque de volúmenes de fluido (VOF por sus siglas en ingles) propuesto originalmente por C. W. Hirt y B. D. Nichols (1979) en el cual se estudia la evolución de un escalar pasivo α sobre el dominio de estudio.

La fracción volumétrica α determina que porcentaje de volumen esta siendo ocupado por cada fluido en cada uno de los volúmenes de control. Este campo escalar es transportado a través del dominio como un escalar pasivo de manera análoga a las leyes de conservación expresadas en la sección previa. La ley de conservación para dicho campo considera unicamente el flujo convectivo a través de las paredes de los volúmenes de control, no existiendo flujo por mecanismos difusivos.

A las ecuaciones original planteadas por Hirt y Nichols se le agrega un termino fuente adicional, el cual agrega un efecto compresivo sobre la interfaz, disminuyendo la falsa difusión numérica de la interfaz permitiendo obtener una mejor precisión sobre la ubicación de esta.

$$\frac{\partial \alpha^{(k)}}{\partial t} - \frac{\partial u_i \alpha^{(k)}}{\partial x_i} + \sum_{m=1}^k \sum_{\substack{n=1 \\ m \neq n}}^k \frac{\partial}{\partial x_i} \left(U_{rel}^{(m,n)} \alpha^{(m)} \alpha^{(n)} \right) = 0 \quad (3.24)$$

El método VOF considera que todo el dominio de estudio esta ocupado por un único fluido equivalente, el cual varia sus propiedades termodinámicas θ_{eq} como el promedio de las N propiedades termodinámicas $\theta^{(k)}$ de cada fase “ k ”, ponderado por su respectiva fracción

volumétrica $\alpha^{(k)}$:

$$\theta_{eq} = \sum_{k=1}^N \theta^k \alpha^{(k)} \quad (3.25)$$

La ecuación de conservación de momentum (3.16) es modificada para incluir los efectos de la tensión superficial. La tensión superficial en la interfaz genera un gradiente de presión adicional que se traduce en una fuerza que se evalúa por unidad de volumen utilizando el modelo “Continuum Surface Force” (CSF):

$$f_i^{(m,n)} = \sigma^{(m,n)} \kappa^{(k)} \left(\alpha^{(m)} \frac{\partial \alpha^{(n)}}{\partial x_i} + \alpha^{(n)} \frac{\partial \alpha^{(m)}}{\partial x_i} \right) \quad (3.26)$$

siendo $f_i^{(m,n)}$ la fuerza de tensión superficial entre las fases m y n , $\sigma^{(m,n)}$ el valor de la tensión superficial entre las fases m y n medido en $[N \cdot m^{-1}]$ y $\kappa^{(k)}$ la curvatura media de la interfaz, determinada según la expresión

$$\kappa^{(k)} = -\frac{\partial}{\partial x_i} \left(\frac{\frac{\partial \alpha^{(k)}}{\partial x_i}}{\sqrt{\left(\frac{\partial \alpha^{(k)}}{\partial x_j} \frac{\partial \alpha^{(k)}}{\partial x_j} \right)}} \right) \quad (3.27)$$

Finalmente es posible reescribir el término correspondiente a la fuerza de tensión superficial F_σ de la ecuación (3.16) haciendo uso de las ecuaciones (3.26) y (3.27)

$$F_\sigma = f_i^{(m,n)} = -\sigma^{(m,n)} \frac{\partial}{\partial x_i} \left(\frac{\frac{\partial \alpha^{(k)}}{\partial x_i}}{\sqrt{\left(\frac{\partial \alpha^{(k)}}{\partial x_j} \frac{\partial \alpha^{(k)}}{\partial x_j} \right)}} \right) \left(\alpha^{(m)} \frac{\partial \alpha^{(n)}}{\partial x_i} + \alpha^{(n)} \frac{\partial \alpha^{(m)}}{\partial x_i} \right) \quad (3.28)$$

3.1.6. Modelación de la turbulencia

a) Fenomenología

Todos los flujos encontrados en problemas de ingeniería, tanto los más simples como chorros bidimensionales, flujos al interior de una tubería o sobre una lamina plana y los más complejos como aquellos con comportamiento tridimensional, presentan

un comportamiento inestable cuando se sobrepasa cierto valor crítico del número de Reynolds. El número de Reynolds, concepto introducido en 1851 por George Gabriel Stokes, y popularizado en 1883 por Osborne Reynolds, corresponde a la relación entre las fuerzas inerciales y las fuerzas viscosas. Matemáticamente se calcula como:

$$Re = \frac{\text{Fuerzas inerciales}}{\text{Fuerzas viscosas}} = \frac{\rho v L}{\mu} \quad (3.29)$$

donde ρ y μ corresponden a la densidad y viscosidad dinámica del fluido respectivamente, v la velocidad del flujo y L una dimensión característica del problema de estudio. Los esfuerzos viscosos que predominan en los flujos laminares dejan de tener una fuerte influencia sobre el comportamiento del flujo cuando se sobrepasa dicho valor crítico. Cuando aumenta más el número de Reynolds predominan los esfuerzos inerciales que generan una transición hacia un flujo turbulento, el cual se caracteriza por un estado de movimiento caótico y azaroso en el cual la velocidad y la presión cambian continuamente con el tiempo dentro de las regiones sustanciales de flujo (Versteeg y Malalasekera, 2005). Para poder entender y caracterizar mejor el fenómeno de la turbulencia, Richardson introdujo en 1922 el concepto de “cascada de energía”, el cual establece que la energía cinética ingresa al flujo turbulento a través de ciertos mecanismos de producción en las más grandes escalas del movimiento. Los vórtices cuya dimensión característica l_0 es comparable a la escala propia del fluido \mathcal{L} tienen una velocidad característica U_0 comparable a la velocidad característica del flujo \mathcal{U} . El número de Reynolds asociados a estos vórtices, $u_0 l_0 / \nu$, es por consiguiente comparable con el del flujo, lo que implica que las fuerzas viscosas sobre estos son despreciables. Estos vórtices son inestables y se fraccionan en vórtices más pequeños, transfiriendo su energía. Esta es posteriormente transferida por procesos viscosos hacia escalas cada vez más pequeñas hasta que finalmente esta energía sea disipada por acción de la viscosidad sobre los vórtices más pequeños en los cuales el número de Reynolds es suficientemente pequeño como se aprecia esquemáticamente en la figura 3.1.

En 1941, Kolmogorov postuló tres hipótesis que permitieron cuantificar e identificar las diferentes escalas en las que se desarrollan la turbulencia (Pope, 2000):

1) Hipótesis de isotropía local :

La hipótesis de isotropía local establece que, para un número de Reynolds

suficientemente grande, la información respecto a las condiciones de contorno del fluido que afectan las escalas más grandes causando su anisotropía, se va perdiendo a medida que la energía es traspasada mediante la escala de energía a las escalas menores, como consecuencia de ello, la estadística del movimiento de estas escalas ($l \ll l_0$) es en cierto modo universal y se pueden considerar estadísticamente isotrópica.

2) Primera hipótesis de Similaridad :

Para flujos con numero de Reynolds lo suficientemente grande, la tasa de transferencia de energía desde las escalas mayores a las menores y la tasa a la que esta energía es transformada en energía interna en las escalas disipativas corresponden a los fenómenos predominantes. Por lo tanto, la estadística en el movimiento de las pequeñas escalas ($l \ll l_0$) tiene una forma universal y esta determinada exclusivamente por la viscosidad del fluido ν y la tasa de disipación ε .

3) Segunda hipótesis de Similaridad :

Para flujos con numero de Reynolds lo suficientemente grande, los vórtices pertenecientes al subrango inercial ($\eta \ll l \ll l_0$) son todavía demasiado grandes para ser susceptibles a los efectos de la viscosidad y demasiado pequeños para sufrir los efectos de los contornos por lo que la estadística de los movimientos de estas escalas están determinadas exclusivamente por la tasa de disipación de la energía cinética ε independiente de la viscosidad ν .

Espectro de energía de Kolmogorov

Se define el tensor de correlación de velocidad para la velocidad turbulenta R_{ij} como sigue:

$$R_{ij}(\vec{r}') = \langle u'_i(\vec{x}) u'_j(\vec{x} + \vec{r}') \rangle \quad (3.30)$$

donde $\langle u'_i \rangle$ corresponde al valor promedio de la velocidad, este promedio puede ser calculado sobre un volumen V :

$$\langle u'_i \rangle = \frac{1}{V} \int_V u'_i dv \quad (3.31)$$

o a lo largo de un intervalo de tiempo τ :

$$\langle u'_i \rangle = \frac{1}{\tau} \int_t^{t+\tau} u'_i dt \quad (3.32)$$

Para el caso de un flujo turbulento completamente desarrollado, la hipótesis de Taylor establece que ambos promedios son idénticamente iguales. Debido a que se considera homogeneidad e isotropía para el campo de velocidades turbulentas, el tensor de correlación de velocidad (3.30) es función exclusivamente de la distancia \vec{r} entre dos puntos y no de la posición espacial \vec{x} de estos.

Debido al amplio rango de escalas espaciales involucradas en un flujo turbulento, resulta conveniente analizar el campo de velocidad en el espacio de números de onda κ mediante el uso de la transformada de Fourier del campo de velocidades. Así se puede definir el tensor del espectro de energía Φ_{ij} simplemente como la representación del tensor de correlación de velocidad R_{ij} en el espacio de Fourier:

$$\Phi_{ij}(\vec{\kappa}) = \frac{1}{(2\pi)^3} \int R_{ij}(\vec{r}) \exp^{-i\vec{\kappa} \cdot \vec{r}} d^3r \quad (3.33)$$

La función del espectro de energía $E(\kappa)$ se define en función de la traza del tensor espectro de energía Φ_{ij} :

$$E(\kappa) = 2\pi\kappa^2 \Phi_{ii}(\vec{\kappa}) \quad (3.34)$$

Para fluidos Newtonianos, la tasa ε_d a la cual la energía cinética es disipada por la viscosidad ν es

$$\varepsilon_d = 2\nu \int_0^\infty \kappa^2 E(\kappa) d\kappa \quad (3.35)$$

Debido a la hipótesis de Kolmogorov de auto-similaridad de la turbulencia, el espectro de energía debe ser de la forma:

$$E(\kappa) = \nu^{\frac{5}{4}} \varepsilon_d^{\frac{1}{4}} E_*(\eta\kappa) \quad (3.36)$$

con E_* la función adimensionalizada del espectro de energía en base al número de onda adimensionalizado $\eta\kappa$ utilizando la escala de Kolmogorov η , esta escala corresponde al

tamaño mas pequeño de los vórtices en el fluido.

$$\eta = \left(\frac{\nu^3}{\varepsilon_d} \right)^{\frac{1}{4}} \quad (3.37)$$

para aquellos números de onda κ pertenecientes al subrango inercial, de acuerdo con la segunda hipótesis de similaridad de Kolmogorov, el espectro de energía tiene una forma universal determinada exclusivamente por la tasa de disipación de la energía cinética ε , siendo independiente de la viscosidad ν , esta hipótesis predice que, para el subrango inercial, la función espectro debe ser de la forma de una ley de potencia:

$$E_*(\eta\kappa) = \alpha (\eta\kappa)^n \quad (3.38)$$

con α y n factores de forma para la ley de potencias. evaluando las expresiones de las ecuaciones (3.37) y (3.38) en la ecuación (3.36) se obtiene

$$E(\kappa) = \nu^{\frac{5}{4}} \varepsilon_d^{\frac{1}{4}} \alpha \left(\left(\frac{\nu^3}{\varepsilon_d} \right)^{\frac{1}{4}} \kappa \right)^n \quad (3.39)$$

con el fin de obtener independencia entre el espectro de energía y la viscosidad cinemática ν el exponente n de la ley de potencia debe tener un valor de $n = -5/3$, se obtiene así la forma que adopta el espectro de energía para un campo de velocidad turbulento en el subrango inercial, denominado “Espectro del Kolmogorov” representado cualitativamente en la imagen 3.2.

$$E(\kappa) = \alpha \varepsilon^{\frac{2}{3}} \kappa^{\frac{-5}{3}} \quad (3.40)$$

b) Modelo de turbulencia: Large-Eddy Simulation

Para la modelación de la turbulencia se utilizó el método Large Eddy Simulation (LES). Este método consiste en resolver directamente los movimientos turbulentos tridimensionales no estacionarios que se ven influenciados por la geometría del flujo, y que no poseen un comportamiento o estadística de movimientos universal mientras que los efectos de las pequeñas escalas son modelados pues, según las hipótesis de Kolmogorov poseen una estadística de movimiento universal.

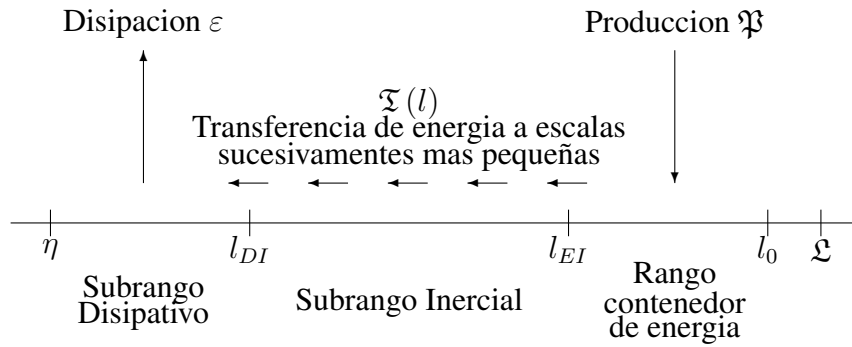


Figura 3.1: Diagrama esquemático de la cascada de energía para un número de Reynolds muy grande, (Pope, 2000)

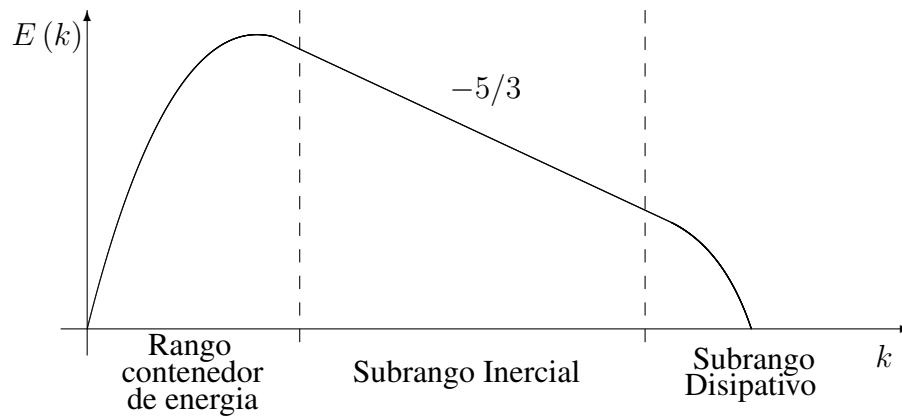


Figura 3.2: Esquema del espectro de energía de Kolmogorov en escala logarítmica y las diferentes escalas en las que se desarrolla la turbulencia

El primer paso consiste en aislar las escalas grandes que se van a resolver directamente de las mas pequeñas. Esto se realiza mediante una operación de filtrado sobre el campo de velocidad como muestra la ecuación (3.41):

$$\bar{U}(x, t) = \int G(r, x) U(x - r, t) dr \quad (3.41)$$

Donde $U(x, t)$ es el campo de velocidades, $\bar{U}(x, t)$ corresponde al campo de velocidades con las pequeñas escalas filtradas y $G(r, x)$ es la función de filtrado “Box”.

$$G(r) = \frac{1}{\Delta} H\left(\frac{1}{2}\Delta - |r|\right) \quad (3.42)$$

Para el caso particular del filtro “Box” (3.42), la velocidad filtrada $\bar{U}(x, t)$ corresponde simplemente al promedio del campo real $U(x', t)$ en el intervalo $x - \frac{1}{2}\Delta < x' < x + \frac{1}{2}\Delta$, con Δ el ancho del volumen de control.

Esta operación de filtrado permite descomponer el campo de velocidades como la suma entre la velocidad filtrada $\bar{U}(x, t)$ y la velocidad residual $u'(x, t)$ como se ve representado gráficamente en la imagen 3.3:

$$U(x, t) = \bar{U}(x, t) + u'(x, t) \quad (3.43)$$

Al realizar la operación de filtrado (3.41) sobre la ecuación de conservación de momentum se obtiene:

$$\frac{\partial \bar{U}_i}{\partial t} + \frac{\partial \bar{U}_j \bar{U}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\mu}{\rho} \frac{\partial^2 \bar{U}_i}{\partial x_j \partial x_j} - (1 - \beta(T - T_0)) g_i \quad (3.44)$$

donde $\bar{p}(x, t)$ es el campo de presión filtrado. La ecuación (3.44) difiere de la ecuación de Navier-Stokes pues el producto filtrado $\bar{U}_i \bar{U}_j$ es diferente del producto de las velocidades filtrada $\bar{U}_i \bar{U}_j$. Se define el tensor de esfuerzo residual como dicha diferencia:

$$\tau_{ij}^R = \bar{U}_i \bar{U}_j - \bar{U}_i \bar{U}_j \quad (3.45)$$

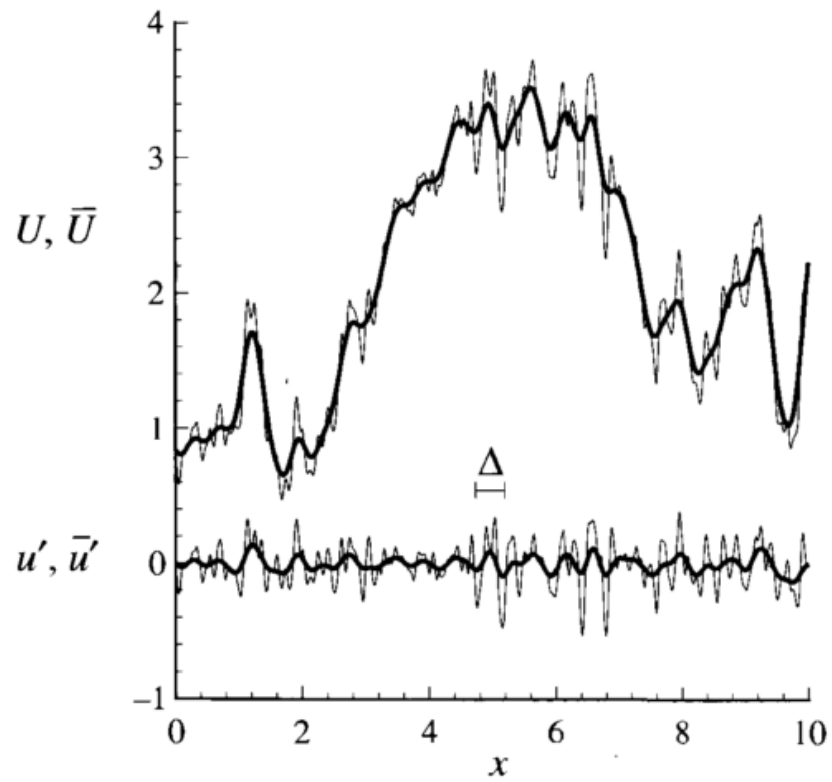


Figura 3.3: Campo de velocidades filtrado utilizando un filtro Gaussiano con longitud de filtro $\Delta = 0,35$. Arriba: Campo de velocidad $U(x)$ y su correspondiente campo de velocidad filtrado $\bar{U}(x)$. Abajo: campo de velocidades residual $u'(x)$ y su correspondiente campo de velocidad residual filtrada $\bar{u}'(x, t)$

La energía cinética residual k_r se define como la traza de dicho tensor:

$$k_r \equiv \frac{1}{2} \tau_{ii}^R \quad (3.46)$$

y el tensor anisotrópico de esfuerzo residual τ_{ij}^r esta definido por:

$$\tau_{ij}^r \equiv \tau_{ij}^R - \frac{2}{3} k_r \delta_{ij} \quad (3.47)$$

La componente isotrópica del tensor de esfuerzo residual se incluye en el campo modificado de presión filtrado:

$$\bar{p} \equiv \bar{p} + \frac{2}{3} k_r \delta_{ij} \quad (3.48)$$

Reemplazando (3.45), (3.47) y (3.48) en la ecuación (3.44), esta puede ser reescrita de la siguiente forma:

$$\frac{\partial \bar{U}_i}{\partial t} + \frac{\partial \bar{U}_j \bar{U}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\mu}{\rho} \frac{\partial^2 \bar{U}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ji}^r}{\partial x_j} - (1 - \beta (T - T_0)) g_i \quad (3.49)$$

Para poder cerrar las ecuaciones (3.49) es necesario aplicar un modelo para el tensor de esfuerzo residual anisotrópico τ_{ij}^r que permita cuantificar los efectos de las escalas de submalla que fueron filtradas. En el presente trabajo se utilizo el modelo “One Equation Eddy” como modelo de clausura.

Modelo “One Equation Eddy” para el tensor de esfuerzo residual anisotropico

En el presente trabajo se utilizó el modelo “One Equation Eddy”, el cual modela el tensor de esfuerzo residual anisotrópico como una variación lineal del tensor tasa de deformación filtrado \bar{S}_{ij} de manera análoga al modelo clásico de Smagorinsky:

$$\tau_{ij}^r = -2\nu_r \bar{S}_{ij} \quad (3.50)$$

$$\bar{S}_{ij} = \frac{1}{2} \left(\frac{\partial \bar{U}_i}{\partial x_j} + \frac{\partial \bar{U}_j}{\partial x_i} \right) \quad (3.51)$$

La constante de linealidad ν_r , denominada viscosidad turbulenta de sub-malla, está definida como función de la energía cinética residual (o de sub-malla) k_r y del tamaño del filtro Δ en la ecuación (3.52):

$$\nu_r = C_k \sqrt{k} \Delta \quad (3.52)$$

Para el cálculo de la energía cinética turbulenta de sub-malla k_r se resuelve una ecuación de transporte adicional:

$$\frac{\partial k_r}{\partial t} + \frac{\partial (\bar{U}_i k_r)}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\nu_{eff} \frac{\partial k_r}{\partial x_i} \right) = 2\nu_r |\bar{S}|^2 - \frac{C_e k_r^{\frac{3}{2}}}{\Delta} \quad (3.53)$$

$$\nu_{eff} = (\nu + \nu_r) \quad (3.54)$$

en la cual aparecen dos términos fuente en el lado derecho. El primero representa el decaimiento de la energía cinética turbulenta desde las escalas resueltas a las escalas de sub-malla mediante la “Cascada de Energía”. El segundo término fuente corresponde a la disipación turbulenta. La viscosidad efectiva ν_{eff} corresponde a la viscosidad del fluido más la viscosidad de sub-malla (3.54). Utilizando el modelo de tensor de esfuerzos residual τ_{ij}^r (3.50) y la definición del tensor tasa de deformación filtrado \bar{S}_{ij} , se puede cerrar el conjunto de ecuaciones (3.49):

$$\frac{\partial \bar{U}_i}{\partial t} + \frac{\partial \bar{U}_j \bar{U}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \nu_{eff} \frac{\partial^2 \bar{U}_i}{\partial x_j \partial x_j} - (1 - \beta (T - T_0)) g_i \quad (3.55)$$

Para incluir el efecto anisotrópico de la turbulencia en las regiones cercanas a las paredes se utilizó la corrección de Van Driest para el cálculo de la viscosidad turbulenta de sub-malla:

$$\nu_r = C_k \sqrt{k} \Delta \left(1 - e^{(-y^+/A^+)} \right) \quad (3.56)$$

donde y^+ es la distancia adimensional a la pared medida en largos de escala viscosa, A^+ es una constante con un valor igual a 26, las constantes C_e y C_k tienen un valor de 1,048 y 0,094 respectivamente.

c) Métodos para la identificación de vórtices

Existen diversos métodos ampliamente utilizados para la identificación y detección de vórtices en un campo de velocidades turbulento, los cuales pueden ser agrupados en distintas categorías:

- Métodos basados en el tensor gradiente de velocidad
- Vorticidad
- Métodos Lagrangianos
- Otros

En el presente trabajo se utilizaron los criterios “Q” y “ λ_2 ” pertenecientes a la categoría de métodos que utilizan en tensor gradiente de velocidad, los cuales se detallan a continuación.

El tensor gradiente de velocidad $D_{ij} = \frac{\partial u_i}{\partial x_j}$ corresponde a un tensor de segundo orden que puede ser descompuesto como la suma de un tensor simétrico S_{ij} y uno anti-simétrico Ω_{ij} de la siguiente manera:

$$\frac{\partial u_i}{\partial x_j} = \underbrace{\frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)}_{S_{ij}} + \underbrace{\frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right)}_{\Omega_{ij}} \quad (3.57)$$

La ecuación característica para el gradiente de la velocidad viene dada por:

$$\lambda^3 + P\lambda^2 + Q\lambda + R = 0 \quad (3.58)$$

con P , Q y R los tres invariantes del tensor gradiente de velocidad, los cuales pueden ser reescritos utilizando la descomposición de la ecuación 3.57:

$$P = -\frac{\partial u_i}{\partial x_i} = -tr(D) \quad (3.59)$$

$$Q = \frac{1}{2} \left(\left(\frac{\partial u_i}{\partial x_i} \right)^2 - \frac{\partial u_i}{\partial x_j} \frac{\partial u_j}{\partial x_i} \right) = \frac{1}{2} [(tr(D))^2 - tr(D^2)] \quad (3.60)$$

$$R = -\varepsilon_{ijk} \frac{\partial u_1}{\partial x_i} \frac{\partial u_2}{\partial x_j} \frac{\partial u_3}{\partial x_k} = -Det(D) \quad (3.61)$$

$$(3.62)$$

Criterio Q El criterio Q define un vórtice como una región del fluido en la cual el segundo invariante del tensor gradiente de velocidad tiene un valor positivo mientras que el valor de la presión local en dicha zona es inferior a la presión del ambiente.

$$Q > 0 \quad (3.63)$$

$$P < P_a \quad (3.64)$$

Criterio λ_2 El criterio λ_2 define un vórtice como una región en la cual el tensor gradiente de velocidad $S_{ij} + \Omega_{ij}$ tiene dos eigen-valores negativos, El tensor gradiente de velocidad $S_{ij} + \Omega_{ij}$ es simétrico y definido positivo, esto implica que todos sus eigen-valores son reales. Al ordenar estos de forma creciente $\lambda_1 < \lambda_2 < \lambda_3$, el criterio se cumple si el segundo de sus eigen-valores es negativo $\lambda_2 < 0$.

Capítulo 4

Dinámica de Fluidos Computacional

La Modelación matemática del problema desarrollada en el capítulo anterior conduce a la resolución de un sistema de ecuaciones diferenciales parciales no lineales que no tienen todavía resolución analítica. Por eso se utilizará un método numérico que permita aproximar de manera discreta dicha solución analítica en puntos discretos del dominio físico del problema. En el presente capítulo se presenta el método de volúmenes finitos utilizado en el presente trabajo para resolver las ecuaciones de Navier-Stokes. En primer lugar se introduce los esquemas de aproximaciones utilizados para las integrales de superficie y volúmenes, como también el esquema de integración temporal para una simulación de tipo transiente. A continuación se realiza un breve repaso de los tipos de condiciones de contorno que se utilizaron en el presente estudio para finalmente presentar los algoritmos de resolución de ecuaciones matriciales con los cuales se resolvieron las ecuaciones lineales obtenidas del proceso de discretización mediante el método de volúmenes finitos.

4.1. Método de Volúmenes Finitos

El método de volúmenes finitos utiliza como partida las ecuaciones de conservación en su forma integral (Ferziger y Peric, 2002):

$$\frac{d}{dt} \int_{\Omega} \rho \phi \, d\Omega + \int_S \rho \phi \vec{v} \cdot \vec{n} \, d\Omega = \int_S \Gamma \vec{\nabla}(\phi) \cdot \vec{n} \, d\Omega + \int_{\Omega} q_{\phi} \, d\Omega \quad (4.1)$$

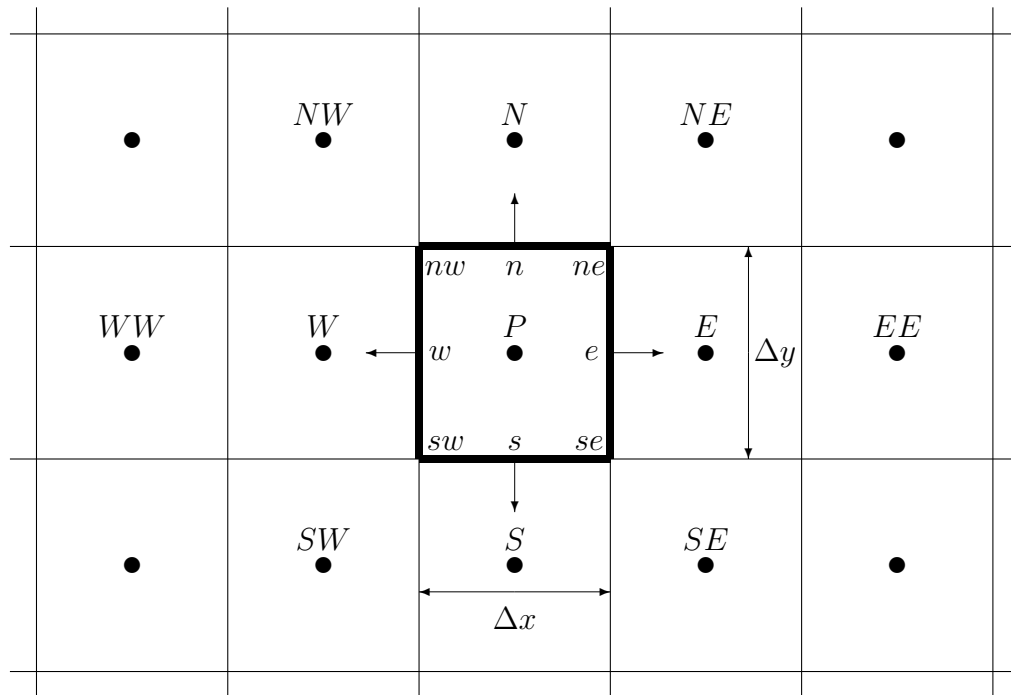


Figura 4.1: Esquema para una malla bidimensional y nomenclatura utilizada en el Método de Volúmenes Finitos

El dominio en el cual buscamos la solución es dividido en un número finito de volúmenes de control (VC) más pequeños. En cada uno se plantea la ecuación de conservación 4.1. Para esto, es necesario aproximar los valores de las integrales de superficies y de volúmenes. La notación estándar utilizada en el método de volúmenes finitos se ve representada en la figura 4.1 en la cual, para un volumen centrado en el nodo P , los volúmenes adyacentes se denominan con letras mayúsculas en base a la posición relativa que ocupan en el espacio, N , S , W , E para los volúmenes ubicados encima, debajo, izquierda y derecha respectivamente. La nomenclatura utilizada para las caras de los volúmenes de control sigue la misma lógica, utilizando letras minúsculas para diferenciarlas.

4.1.1. Aproximación Integrales de Superficie

Las integrales de superficie que aparecen en la ecuación 4.1 se pueden descomponer como la suma de las integrales sobre cada “cara” del volumen:

$$F(\phi) = \int_S f(\phi) \, dS = \sum_k \int_{S_k} f_k(\phi) \, dS \quad (4.2)$$

siendo $f(\phi)$ el flujo convectivo o difusivo de la variable (ϕ) que se desea integrar sobre las k superficies que componen la superficie total S del volumen de control. La integral sobre cada cara se calcula utilizando el valor promedio del flujo f sobre la cara integrada

$$\overline{f(\phi)}_{S_k} = \int_{S_k} f(\phi)_k \, dS \quad (4.3)$$

El valor promedio $\overline{f(\phi)}_k$ puede ser aproximado de diversas maneras: integración Gaussiana, regla del trapecio, regla de Simpson, etc. En el presente trabajo se utilizó el valor del punto medio de la cara como aproximación. En particular, el flujo promedio sobre la cara “east” (e) se obtiene

$$\overline{f(\phi)}_{east} \approx f(\phi)_e \quad (4.4)$$

Para poder evaluar los flujos convectivos y difusivos sobre las caras según la ecuación (4.4), es necesario interpolar los valores de ϕ desde los nodos de la malla (W, E, S, N) hasta los puntos medios de dichas caras (w, e, s, n). A continuación se detallan los esquemas de interpolación utilizados.

Upwind Differencing Scheme

El esquema de interpolación Upwind Differencing Scheme (UDS) aproxima el valor de la variable ϕ sobre la superficie deseada en función de la dirección del flujo que la atraviesa como se ve representado en la imagen 4.2. Si el flujo ingresa al volumen de control, se aproxima al valor de la variable dentro del volumen de control vecino, por el contrario si el flujo es saliente, se aproxima al valor nodal del propio volumen de control (Ferziger y Peric, 2002)(Versteeg y Malalasekera, 1995).

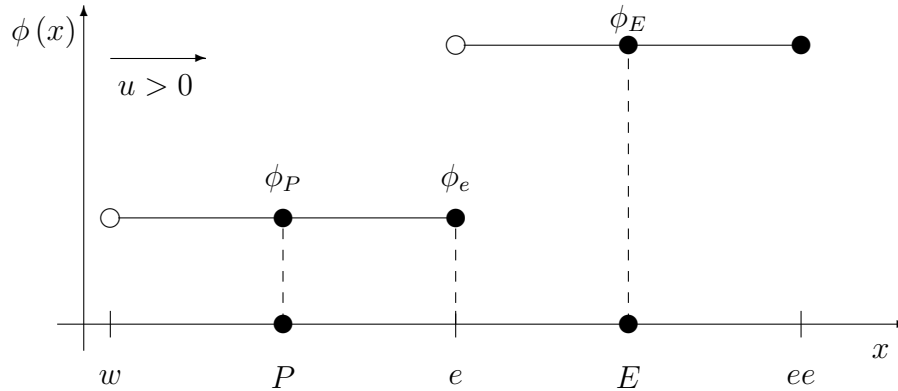


Figura 4.2: Esquema de aproximación UDS de la función ϕ sobre una malla unidimensional para un flujo que se mueve en dirección x

$$\phi_e = \begin{cases} \phi_P & \text{si } (u \cdot n)_e > 0 \\ \phi_E & \text{si } (u \cdot n)_e < 0 \end{cases} \quad (4.5)$$

El esquema de interpolación “UDS” corresponde a un esquema de primer orden $O(h)$, es decir, la magnitud del error cometido en la aproximación tiene igual orden de magnitud que la dimensión del volumen de control. Esto puede ser verificado realizando la expansión en series de Taylor alrededor del punto P . Como esquema de primer orden, no presenta oscilaciones espurias en zonas donde la función ϕ presente cambios abruptos, sin embargo posee características disipativas.

Central Differencing Scheme

El esquema de interpolación lineal “Central Differencing Scheme” aproxima el valor de la variable ϕ sobre las fronteras del volumen de control mediante una interpolación lineal de los valores nodales entre los volúmenes que comparten dicha frontera (Versteeg y Malalasekera, 1995) como se ve representado en la figura 4.3.

$$\phi_e = \phi_E \frac{x_e - x_P}{x_E - x_P} + \phi_P \left(1 - \frac{x_e - x_P}{x_E - x_P} \right) \quad (4.6)$$

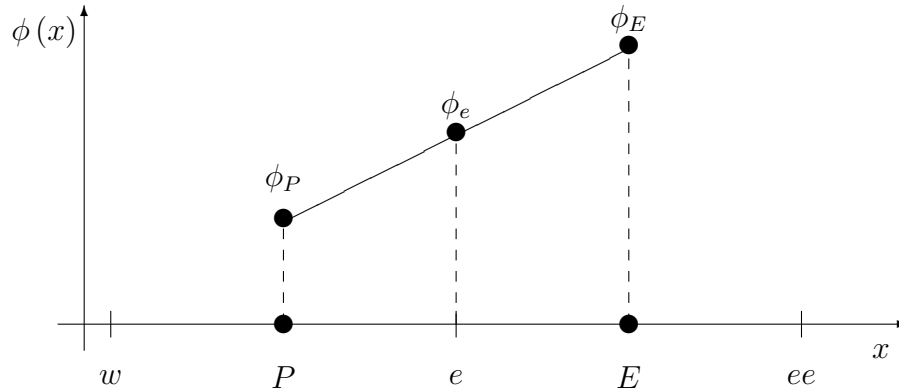


Figura 4.3: Esquema de aproximación CDS de la función ϕ sobre una malla unidimensional para un flujo que se mueve en dirección x

El esquema “CDS” a diferencia del “UDS”, corresponde a un esquema de segundo orden $O(h^2)$ el cual puede ser verificado de manera análoga al esquema anterior. Este esquema no presenta gran dispersión numérica a diferencia del “UDS” por lo que permite capturar de mejor manera la posición y el comportamiento de la función ϕ en zonas de gran variación. Por el contrario, presenta un comportamiento dispersivo, generando oscilaciones espurias las cuales pueden causar que los valores obtenidos localmente sean físicamente incorrectos, i.e. valores para la fracción volumétrica α superiores a 1 o inferiores a 0

Total Variation Diminishing

La familia de esquemas de discretización Total Variation Diminishing (TVD) utilizado para la resolución de ecuaciones diferenciales parciales de tipo hiperbólicas (Versteeg y Malalasekera, 1995) tienen la particularidad de limitar la aparición de oscilaciones espurias en el campo que se está resolviendo en zonas de discontinuidades y de cambios agudos, como por ejemplo el campo de densidad ρ sobre la interfaz entre fluidos inmiscibles. La figura 4.4 muestra comparativamente la aparición de oscilaciones espurias entre un esquema TVD y uno no-TVD en una ecuación de advección pura unidimensional. Consideremos la ecuación de advección pura unidimensional semi-discretizada mediante

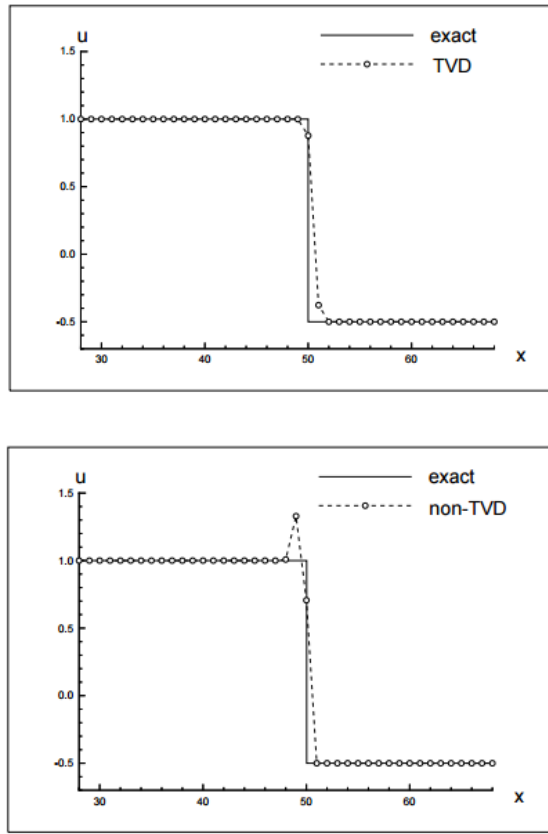


Figura 4.4: Comparación entre la solución obtenida para la ecuación de advección pura unidimensional entre un esquema TVD (arriba) y uno no-TVD (abajo).

el método de volúmenes finitos de la forma

$$\frac{\partial \phi_p}{\partial t} + \frac{1}{\Delta X_p} [F(\phi_e) - F(\phi_w)] = 0 \quad (4.7)$$

donde $F(\phi_e)$ y $F(\phi_w)$ representan los flujos advectivos sobre las caras w y e respectivamente. Estos flujos pueden ser representados como un promedio entre un esquema de resolución de alto y bajo orden ponderados mediante una función limitadora de flujo $\psi(r)$

$$F(\phi_e) = f_e^{low} - \psi(r_P) (f_e^{low} - f_e^{high}) \quad (4.8)$$

$$F(\phi_w) = f_w^{low} - \psi(r_W) (f_w^{low} - f_w^{high}) \quad (4.9)$$

donde r representa la tasa de gradientes sucesivos

$$r_P = \frac{\phi_P - \phi_W}{\phi_E - \phi_P} \quad (4.10)$$

La elección de la función limitadora de flujo determina el método TVD en particular. El esquema de Van Leer utilizado en el presente trabajo utiliza la función limitadora de flujo:

$$\phi(r) = \frac{r + |r|}{1 + |r|} \quad (4.11)$$

Se utilizó además los esquemas de interpolación “Upwind Differencing Scheme” y “Central Differencing Scheme” como los esquemas de bajo y alto orden respectivamente.

En la literatura es posible encontrar diversos métodos de aproximación de distinto orden de error como por ejemplo: QUICK, Power Law, Lax–Wendroff, etc. En el presente trabajo se utilizaron exclusivamente los explicados en esta sección.

4.1.2. Aproximación Integrales de Volúmenes

De manera análoga al tratamiento realizado con las integrales de superficie, las integrales sobre los volúmenes de control pueden ser calculadas mediante el valor promedio de la fuente al interior de la celda (volumen de control):

$$\int_{\Omega} q_{\phi} d\Omega = \overline{q(\phi)} V \quad (4.12)$$

A modo de simplificar los cálculos, se consideró una distribución uniforme de la variable fuente $q(\phi)$ dentro del volumen de control, correspondiente al valor del nodo central P :

$$\overline{q(\phi)} \approx q(\phi_P) \quad (4.13)$$

4.1.3. Aproximación Derivada Temporal

El primer término de la ecuación (4.1) correspondiente a la variación temporal de la variable ϕ dentro del volumen de control P es aproximada utilizando las expresiones

(4.12) y (4.13):

$$\frac{\partial}{\partial t} \int_{\Omega} \rho \phi \, d\Omega = \frac{\partial}{\partial t} (\rho \phi_P V_P) \quad (4.14)$$

Como se estableció previamente, en el presente trabajo, se usará la simplificación de fluidos incompresibles, lo que permite reescribir la ecuación (4.14) como sigue:

$$\frac{\partial}{\partial t} (\rho \phi_P V_P) = \rho V_P \frac{\partial \phi_P}{\partial t} \quad (4.15)$$

Utilizando estas expresiones, se puede reescribir la ecuación (4.1) de forma genérica:

$$\frac{\partial \phi_P}{\partial t} = \frac{1}{\rho V_P} F(\phi, x, t) \quad (4.16)$$

donde $F(\phi, x, t)$ agrupa la suma de los flujos convectivos y difusivos, además de los términos fuentes.

Finalmente la derivada temporal se calcula utilizando el esquema de Crank Nicolson

$$\frac{\partial \phi_P}{\partial t} \approx \frac{\phi_P(t + \Delta t) - \phi_P(t)}{\Delta t} = \frac{1}{2\rho V_P} (F(\phi, x, t + \Delta t) + F(\phi, x, t)) \quad (4.17)$$

El esquema de integración temporal Crank-Nicolson corresponde a un esquema de segundo orden $O(k^2)$ incondicionalmente estable, lo que permite obtener una mejor precisión de la solución con un costo adicional en tiempo de cálculo y de memoria, pues requiere almacenar tanto el intervalo de tiempo que se está calculando como el intervalo previo.

4.1.4. Estabilidad Numérica

Se dice que el esquema de discretización numérico es estable si no magnifica la magnitud de los errores que van surgiendo en el proceso de la solución numérica. En el caso de problemas transiente la estabilidad garantiza que el método produzca una solución acotada cuando la solución exacta también lo es (Ferziger y Peric, 2002). Para garantizar la estabilidad de un esquema se utiliza la condición de Courant-Friedrichs-Lewy (CFL) el

cual establece un valor máximo sobre el numero de Courant Co :

$$Co = \frac{u\Delta t}{\Delta x} \leq Co_{max} \quad (4.18)$$

donde u es la magnitud de la velocidad, Δx el tamaño de la discretización espacial en una dimension y Δt el paso de tiempo. El esquema de integración temporal Crank-Nicolson se dice incondicionalmente estable pues la condición de estabilidad CFL se cumple para todo valor del numero de Courant. Si bien en el código empleado todas las integraciones temporales son realizadas utilizando un esquema implícito que garantiza estabilidad incondicional para todas las ecuaciones por separado, el acoplamiento entre estas se realiza de forma semi-implícita. El esquema de discretización de las ecuaciones acopladas es por lo tanto condicionalmente estable. Sin embargo la expresión analítica para la condición de estabilidad CFL no es simple de obtener. De la literatura y realizando distintas pruebas se obtuvo una estabilidad en la simulación para números de Courant menores o igual a 0,5 en cada dirección.

$$Co = \frac{u\Delta t}{\Delta x} \leq 0,5 \quad (4.19)$$

4.1.5. Condiciones de contorno

En la resolución de problemas de ecuaciones diferenciales, ya sean ordinarias o parciales, se necesita información adicional sobre el comportamiento de la solución sobre los contornos del dominio. Esta información permite discernir entre la familia de funciones que satisfacen la ecuación. Considere la ecuación diferencial en derivadas parciales del tipo

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} = g(x, y) \quad (4.20)$$

correspondiente a la ecuación de Poisson en dos dimensiones sobre el dominio Ω cuya frontera esta delimitada por las curvas Γ_1 , Γ_2 y Γ_3 como se observa en la figura 4.5.

Las condiciones de frontera pueden ser de tres tipos:

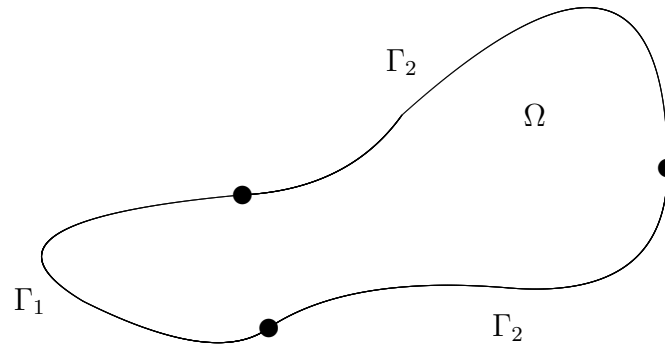


Figura 4.5: Representación de un dominio Ω de contorno Γ donde se busca la solución de una EDP.

1) Dirichlet

Condición de contorno de primer tipo en la cual se fija directamente el valor que adopta la variable sobre el contorno del dominio:

$$\Phi(\Gamma_i) = f(\Gamma_i) \quad i = 1, 2, 3 \quad (4.21)$$

2) Neumann

Condición de contorno de segundo tipo en la cual se fija directamente el valor que adopta el gradiente de la variable en la dirección normal al contorno del dominio:

$$\frac{\partial \Phi(\Gamma_i)}{\partial n} = f(\Gamma_i) \quad i = 1, 2, 3 \quad (4.22)$$

3) Robin

Condición de contorno mixta que corresponde a una combinación lineal de una condición de tipo Dirichlet con una de tipo Neumann:

$$\frac{\partial \Phi(\Gamma_i)}{\partial n} + \Phi(\Gamma_i) = f(\Gamma_i) \quad i = 1, 2, 3 \quad (4.23)$$

Las condiciones de contorno para las ecuaciones de Navier Stokes y conservación de energía utilizadas en el presente estudio son casos específicos de alguno de estos tres tipos. Sobre las paredes del dominio, se establece la condición física de no deslizamiento. Esto quiere decir que la velocidad relativa entre el fluido y la pared es 0. Siendo esta última definida por el problema, para casos donde la pared se encuentra estática, la velocidad del fluido sobre la pared es 0:

$$\vec{U}(\Gamma) - \vec{U}_{wall} = 0 \quad (4.24)$$

(Condición de Dirichlet)

Para la frontera que corresponde a la entrada de fluido, se establece el campo de velocidad en función al perfil de entrada deseado:

$$\vec{U}(\Gamma) = \vec{U}_{entrada} \quad (4.25)$$

(Condición de Dirichlet)

Sobre los planos de simetría, se establece una condición de Dirichlet sobre la velocidad en dirección normal, mientras que se utiliza una condición de Neumann sobre la componente de la velocidad tangencial al plano, de esta forma se fuerza a que no exista flujo a través de la frontera y además a que la componente deviatórica del tensor de esfuerzos sea cero (Ferziger y Peric, 2002):

$$U_{\perp} = 0 \quad (4.26)$$

$$\frac{\partial U_{\parallel}}{\partial n} = 0 \quad (4.27)$$

donde U_{\perp} y U_{\parallel} corresponden a la componente de la velocidad perpendicular y tangencial a la frontera Γ respectivamente, mientras que n representa la dirección normal de esta. Para las fronteras por donde sale el flujo, es común utilizar una condición de contorno de gradiente cero, vale decir, la velocidad no varía en la dirección normal a la frontera

$$\frac{\partial U_{\perp}}{\partial n} = 0 \quad (4.28)$$

(Condición de Neumann)

4.2. Métodos de Resolución de Sistemas Lineales

La discretización espacial y temporal de las ecuaciones de Navier Stokes que gobiernan el comportamiento de los fluidos Newtonianos permite transformar el sistema de ecuaciones diferenciales parciales, con sus respectivas condiciones de contorno, en un sistema lineal de ecuaciones, el cual debe ser resuelto para cada paso de tiempo a lo largo de la simulación. A continuación se presentan los métodos para resolver sistemas de ecuaciones lineales utilizados en el presente estudio. Por simplicidad se presentarán los métodos con la ecuación de Poisson unidimensional, el cual puede ser extendido a un problema tridimensional por analogía.

Sea la ecuación diferencial que se desea resolver de la forma:

$$\frac{d^2\phi}{dx^2} - f(x) = 0 \quad (4.29)$$

Discretizandola mediante el método de volúmenes finitos sobre una malla regular obteniendo la ecuación matricial a resolver:

$$A \cdot \phi - b = 0 \quad (4.30)$$

donde la matriz A contiene los términos obtenidos producto del proceso de discretización de las integrales que se encuentran multiplicando a la función incógnita ϕ , mientras que el vector b agrupa los términos que son independientes de la función incógnita.

4.2.1. Método del Gradiente Conjugado

El método del gradiente conjugado permite resolver la ecuación matricial (4.30) para A matriz simétrica definida positiva. Resolver este sistema de ecuaciones resulta similar a minimizar su forma cuadrática

$$q(x) = \langle x, Ax \rangle - 2\langle x, b \rangle \quad (4.31)$$

utilizando la notación de producto interno entre dos vectores:

$$\langle x, y \rangle = x^t y \quad (4.32)$$

Para esto se busca una sucesión de vectores x_k tal que $q(x_{k+1}) \leq q(x_k)$. Esto se logra tomando el vector de la forma $x_{k+1} = x_k + tv$, con t escalar y v vector unitario perteneciente a una base ortogonal de A $\langle v^{(i)}, Av^{(j)} \rangle = 0, i \neq j$:

$$\begin{aligned}
 q(x + tv) &= \langle x + tv, A(x + tv) \rangle - 2\langle x - tv, b \rangle \\
 &= \langle x, Ax \rangle + 2t\langle x, Av \rangle + t^2\langle v, Av \rangle - 2\langle x, b \rangle - 2t\langle v, b \rangle \\
 &= q(x) + 2t\langle v, Ax \rangle - \langle v, b \rangle + t^2\langle v, Av \rangle \\
 &= q(x) + 2t\langle v, Ax - b \rangle + t^2\langle v, Av \rangle
 \end{aligned} \tag{4.33}$$

La ecuación (4.33) es cuadrática para t y alcanza su valor mínimo en

$$\hat{t} = \frac{\langle v, b - Ax \rangle}{\langle v, Av \rangle} \tag{4.34}$$

En particular, para este valor de \hat{t} se cumple que $q(x_{k+1}) \leq q(x_k)$

$$q(x_k) - q(x_{k+1}) = \frac{\langle v, b - Ax_k \rangle^2}{\langle v, Av \rangle} \tag{4.35}$$

Dado que A es una matriz definida positiva $\langle v, Av \rangle > 0$.

Para la elección del vector v en el cual se buscará el vector solución x se construye una base ortogonal utilizando la propiedad adicional que la sucesión de residuos forman un sistema ortogonal $\langle r_i, r_j \rangle = 0, i \neq j$.

El algoritmo de resolución se puede expresar como sigue:

$$\begin{aligned}
 r_0 &= b - Ax_0 \\
 v_0 &= r_0 \\
 t_k &= \frac{\langle r_k, r_k \rangle}{\langle r_k, Av_k \rangle} \\
 x_{k+1} &= x_k + t_k v_k \\
 r_{k+1} &= r_k - t_k Av_k \\
 s_k &= \frac{\langle r_{k+1}, r_{k+1} \rangle}{\langle r_k, r_k \rangle} \\
 v_{k+1} &= r_{k+1} + s_k v_k
 \end{aligned} \tag{4.36}$$

Para una matriz A de dimensión n , la base ortogonal estará compuesta por n vectores linealmente independientes, vale decir el numero de direcciones necesarias para encontrar la solución exacta del problema es finito e igual a n , siendo este un método de resolución de sistemas matriciales directo. En la practica es posible disminuir el costo computacional requerido para obtener la solución exacta realizando un numero menor de iteraciones hasta obtener un residuo que este bajo una tolerancia establecida.

4.2.2. Método Gauss-Seidel

El método iterativo Gauss-Seidel permite resolver sistemas lineales de la forma (4.30) bajo las siguientes condiciones:

- La matriz A es diagonal dominante

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \forall i \in [1, n]$$

- La matriz A es Simétrica y definida positiva

$$A^T = A$$

$$u^T A u > 0, \forall u \neq 0$$

Consideremos la descomposición de la matriz A como la suma de una matriz triangular inferior L , una matriz diagonal D y una triangular superior U :

$$\underbrace{\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}}_A = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ a_{2,1} & 0 & 0 \\ a_{3,1} & a_{3,2} & 0 \end{bmatrix}}_L + \underbrace{\begin{bmatrix} a_{1,1} & 0 & 0 \\ 0 & a_{2,2} & 0 \\ 0 & 0 & a_{3,3} \end{bmatrix}}_D + \underbrace{\begin{bmatrix} 0 & a_{1,2} & a_{1,3} \\ 0 & 0 & a_{2,3} \\ 0 & 0 & 0 \end{bmatrix}}_U \quad (4.37)$$

la cual nos permite reescribir la ecuación (4.30) de la siguiente forma:

$$(L + D + U) \phi = b$$

$$D\phi + (L + U) \phi = b$$

$$D\phi = b - (L + U) \phi$$

$$\phi = D^{-1} (b - L\phi - U\phi) \quad (4.38)$$

De esta forma, para el volumen i , el valor de ϕ es aproximado en la iteración $k + 1$ como:

$$\phi_i^{k+1} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} \phi_j^{k+1} - \sum_{j=i+1}^n a_{i,j} \phi_j^k \right) \quad (4.39)$$

para el calculo del valor en la iteración $k + 1$. Este método requiere tanto los valores de la iteración k como los valores ya calculados $k + 1$. El error se vuelve una función suave en el espacio al cabo de una pocas iteraciones lo que permite utilizarlo en el método multi-mallas (Burden y Faires, 2011).

4.2.3. Método Multi-Malla (MGM)

El método Multi Malla se basa sobre una observación de los métodos iterativos. En estos, la velocidad de convergencia de la solución entre iteraciones sucesivas va disminuyendo a medida que se acerca a la solución real. Luego de unas pocas iteraciones el error de iteración se vuelve una función suave en el espacio coordinado, por lo que es posible aproximar dicho error sobre una malla el doble de gruesa que la malla original, disminuyendo el coste computacional (Ferziger y Peric, 2002).

Sea ρ^j el vector residual correspondiente a la aproximación de $\phi^j \approx \phi$ luego de j iteraciones, la ecuación (4.30) puede ser reescrita como sigue:

$$\rho^j = A \cdot \phi^j - b \quad (4.40)$$

Combinando las ecuaciones (4.30) y (4.40), obtenemos la ecuacion matricial correspondiente al error de iteracion ε^j .

$$\rho^j = A \cdot \varepsilon^j \quad (4.41)$$

Para obtener el valor del error de iteración, se procede a aproximar mediante un método iterativo la ecuación (4.41) pero sobre una malla mas gruesa que la original, para esto se debe definir la relación entre las mallas, el operador diferencial finito en la malla gruesa, el cual debe ser consistente con el utilizado en la malla fina, un método de transformar el residuo desde la malla fina a la gruesa (restricting) y un metodo para interpolar el error de iteración desde la malla gruesa a la malla fina (prolonging) (Ferziger y Peric, 2002).

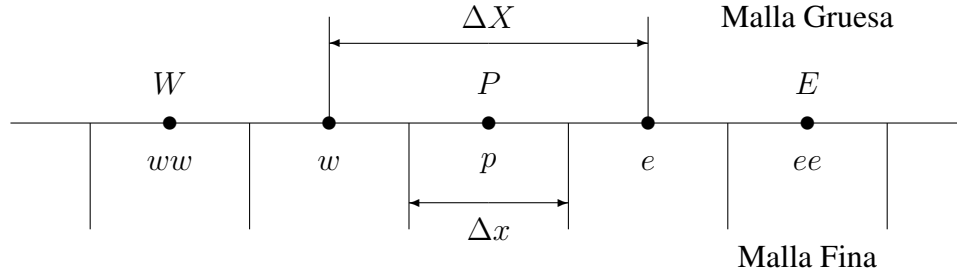


Figura 4.6: Esquema de las mallas utilizadas en el método multi-mallas para problemas unidimensionales

Consideremos la ecuación (4.41) discretizada utilizando un esquema de diferencias centradas, el residuo para el volumen p y sus volúmenes adyacentes w y e son respectivamente:

$$\frac{1}{\Delta x^2} (\varepsilon_w - 2\varepsilon_p + \varepsilon_e) = \rho_p \quad (4.42)$$

$$\frac{1}{\Delta x^2} (\varepsilon_{ww} - 2\varepsilon_w + \varepsilon_p) = \rho_w \quad (4.43)$$

$$\frac{1}{\Delta x^2} (\varepsilon_p - 2\varepsilon_e + \varepsilon_{ee}) = \rho_e \quad (4.44)$$

estos residuos (4.42) (4.43) (4.44) pueden combinarse convenientemente, por ejemplo:

$$\frac{1}{(2\Delta x)^2} (\varepsilon_{ww} - 2\varepsilon_p + \varepsilon_{ee}) = \frac{\rho_w + 2\rho_p + \rho_e}{4} \quad (4.45)$$

La ecuación (4.45) será iterada sobre la malla gruesa representada en la figura 4.6, utilizando la relación entre el tamaño de ambas mallas $\Delta X = 2\Delta x$ y la transformación (restricting) del residuo desde la malla fina a la gruesa $\rho_P = \frac{\rho_w + 2\rho_p + \rho_e}{4}$

$$\frac{1}{\Delta X^2} (\varepsilon_W - 2\varepsilon_P + \varepsilon_E) = \rho_P \quad (4.46)$$

El algoritmo de resolución general para un esquema de dos mallas se puede resumir de la siguiente forma:

- Realizar iteraciones en la malla fina mediante un método que entregue un error de iteración suave
- Calcular el residuo de la aproximación obtenida en el paso previo
- traspasar la información a la malla gruesa
- iterar el error sobre la malla gruesa
- interpolar el error calculado sobre la malla gruesa a la malla fina
- calcular el residuo en la malla fina
- verificar si el valor del residuo esta bajo la tolerancia permitida, si es mayor volver al primer paso

Dado que la aproximación mediante un método iterativo es mas rápida mientras mas gruesa sea la malla, se puede continuar trabajando con mallas cada vez mas gruesas hasta abarcar el dominio completo, a esta variación del algoritmo de dos mallas se le conoce como método multi mallas.

Capítulo 5

Simulación Numérica

En este capítulo se realiza una breve descripción del software libre OpenFOAM utilizado en el presente trabajo para proceder a explicar en detalle el funcionamiento del código modificado. A continuación se presentan dos casos utilizados para validar el desempeño del código: estos son la transferencia de calor hacia el interior de una tubería de sección circular con un flujo de calor constante a lo largo de toda la tubería, y la inestabilidad de Rayleigh-Bénard en una celda de aire para distintos números de Rayleigh. Una vez presentado y validado el código, se procede a plantear el problema que se desea analizar. Luego se presenta el dominio físico y la discretización espacial utilizada así como las condiciones iniciales y de borde para las distintas ecuaciones a resolver. Finalmente se presentan un resumen con los distintos casos a estudiar, acompañados los parámetros físicos de los fluidos empleados.

5.1. OpenFOAM

Para resolver numéricamente el problema del flujo multifásico de este trabajo, se utilizó el código libre open source para dinámica de fluidos computacional OpenFOAM. OpenFOAM es distribuido bajo la licencia de publico general GNU. Es desarrollado por contribuidores guiados por Henry Weller, creador de OpenFOAM, director de CFD Direct y de la fundación OpenFOAM.

OpenFOAM (Open source Field Operation And Manipulation) es un conjunto de códigos escritos en c++ sobre los que se puede desarrollar “solvers” numéricos y códigos para

pre/post procesar, solucionar problemas de la mecánica de medios continuos como dinámica de fluidos computacional y mecánica de sólidos. Tiene un enfoque modular gracias a la programación orientada a objetos, lo que permite al usuario programar y compilar por ejemplo condiciones de contorno propias, modelos de turbulencias o modelos constitutivos, y posteriormente ser utilizados por los solvers existentes.

Existen dos categorías principales de aplicaciones:

Solvers Realizan el calculo para resolver un problema especifico de la mecánica de medios continuos.

Utilidades Utilizadas antes y después de realizar el calculo, permiten preparar la malla, preparar el caso a simular, estableciendo por ejemplo las condiciones iniciales o postprocesar los resultados.

Este conjunto de aplicaciones permite resolver problemas tales como:

- Flujos compresibles e incompresibles laminares y turbulentos mediante DNS o con modelos de turbulencia RANS y LES
- Flujos con fuerzas de flotacion mediante la aproximación de Boussinesq
- Flujos multifásicos
- Rastreo de partículas
- Transferencia de calor y combustión
- Dinámica Molecular
- Mecánica de sólidos
- Electromagnetismo

Las principales ventajas que presenta OpenFOAM son la posibilidad y facilidad de crear aplicaciones nuevas que se ajusten a las necesidades del usuario modificando alguna aplicación preexistente. Por ejemplo se puede agregar transferencia de energía y fenómeno de flotación al solver para flujos multifásicos. La capacidad de ejecutar las aplicaciones en paralelo, reduce significativamente los tiempos de cálculo de las simulaciones.

5.2. Código

Como punto de partida se utilizó el “solver” preexistente “multiphaseInterFoam”, el cual permite resolver la interacción entre n fluidos incompresibles con captura de la interfaz. Este “solver” incluye una modelación para la tensión superficial, el ángulo de contacto entre la interfaz y las paredes, además de una modelación para la turbulencia utilizando modelo RANS y LES. Este se modificó para que resuelva además la transferencia de calor entre los fluidos e incluir el efecto de la flotación producto de la variación en la densidad a distintas temperaturas utilizando la aproximación de Boussinesq, la cual es válida para pequeñas variaciones en la densidad producto del cambio de temperatura $\Delta\rho(T) \ll 1$ descrita en las secciones previas. La figura 5.1 representa el diagrama de flujos a modo general del código. El primer paso correspondiente al preámbulo consiste en cargar una serie de librerías de OpenFOAM como se muestra en el diagrama de la figura 5.2. Estas librerías definen operaciones matemáticas como la divergencia, gradiente y laplaciano además de definir la base para trabajar con el método de volúmenes finitos, en conjunto con los modelos de turbulencias flujos multifásicos. Luego de cargar las librerías necesarias para el código se realiza una verificación de la integridad de los archivos necesarios para correr la simulación: malla, condiciones de contorno e iniciales, ficheros de control para posteriormente almacenar en la memoria esta información. Realizado esto, inicializa las variables a resolver como también variables auxiliares de control i.e. errores de iteración y acumulados. Con toda esta información se calcula el primer paso de tiempo de tal manera que se cumplan los criterios para el número de Courant tanto del fluido como de la interfaz como del paso de tiempo máximo definido por el usuario. Finalizado las tareas del preámbulo comienza el bucle de evolución temporal el cual se repetirá hasta que el tiempo simulado alcance el tiempo final definido por el usuario. Lo primero que se verifica es si se ingresó algún cambio en los parámetros de control del caso simulado y se actualiza el paso de tiempo. A continuación se resuelve la ecuación de la fracción volumétrica α para determinar la distribución de los distintos flujos en el dominio, con esta distribución se procede a calcular las propiedades de la mezcla según el modelo “mixture” (ecuación (3.25)) para posteriormente ingresar al algoritmo de resolución para Navier Stokes PIMPLE modificado para incluir la temperatura, esquematizado en el diagrama de la figura 5.3. Este cuenta con dos parámetros de control los cuales definen la cantidad de veces que

se realizan las correcciones al campo de velocidad y de presión. Este algoritmo se puede resumir como sigue:

1. Calculo de un campo de velocidad intermedio a partir de un campo de presión dado de manera arbitraria, el cual cumple con la conservación de momento pero no con la conservación de la masa.
2. Calculo del campo de temperaturas utilizando el campo preliminar de velocidades.
3. Resuelve la ecuación de corrección de presión.
4. Calculo de los flujos máxicos sobre las caras de los volúmenes de control.
5. Corrige el campo de presión.
6. Actualiza los valores de la presión en las condiciones de contorno.
7. Corrige los flujos máxicos sobre las caras de los volúmenes de control.
8. Resuelve la ecuación para la energía turbulenta de sub-malla k .
9. Calcula la viscosidad de sub-malla ν_{sg} .
10. Si no se ha alcanzado el número de iteraciones máxicas, regresa al punto 1, donde vuelve a resolver la ecuación de conservación de momento pero esta vez utilizando el campo de presión y flujos máxicos corregidos en los puntos anteriores.

Al salir del bucle PIMPLE, verifica si al tiempo de simulación actual la información calculada debe ser guardada, avanza al siguiente punto temporal y verifica si se ha alcanzado el tiempo final de la simulación, de no ser así vuelve a comenzar el bucle temporal. Al comienzo de cada paso temporal, este se ajusta para cumplir con los criterios de estabilidad señalados previamente, de tal manera que en todo el dominio simulado, el número de Courant no sobrepase un valor máxico.

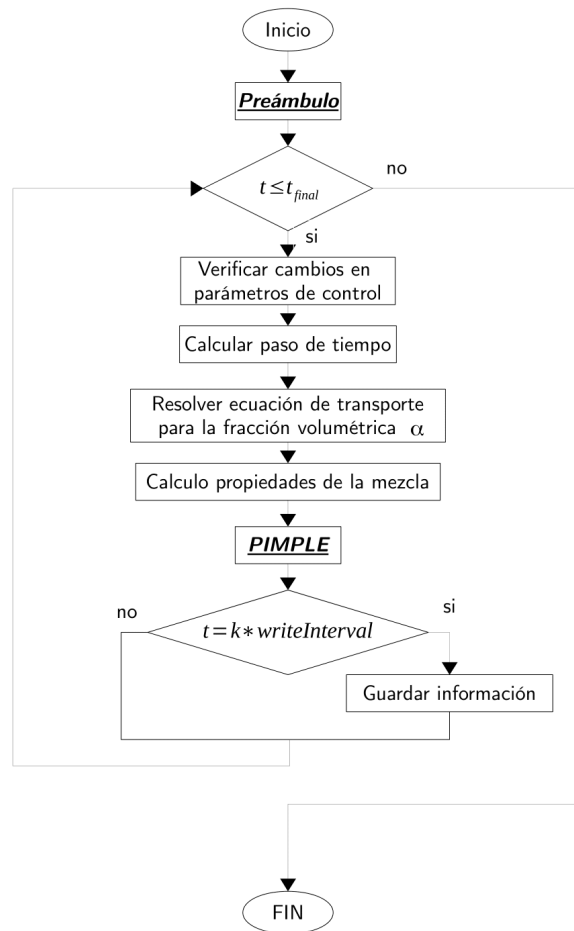


Figura 5.1: Diagrama de flujos global código MTBoussinesqIFoam

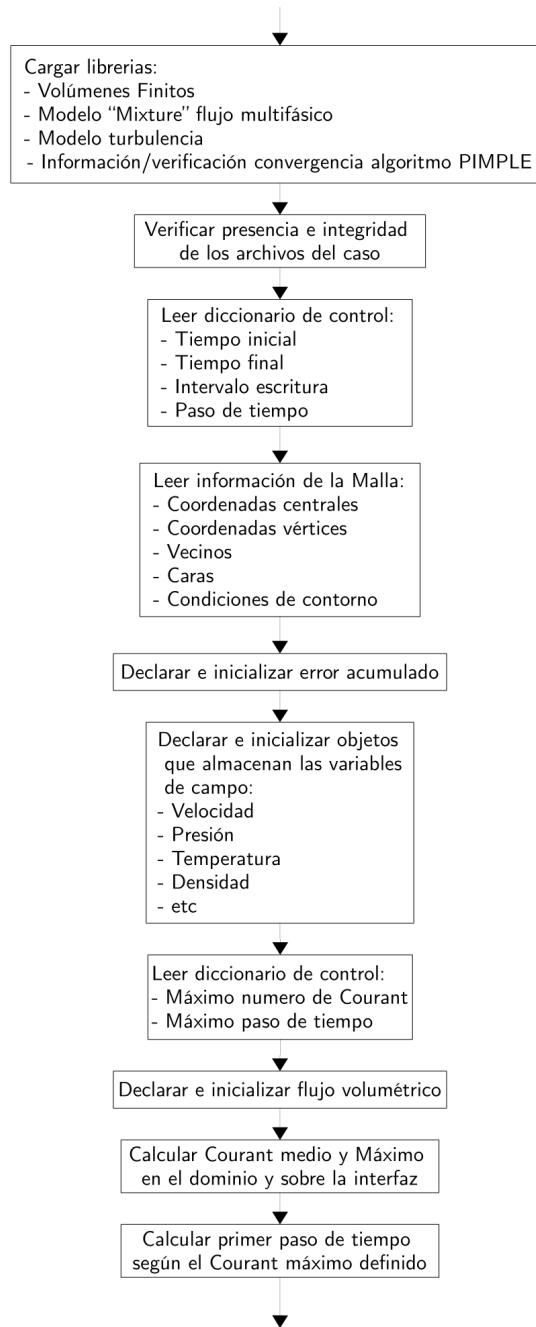


Figura 5.2: Diagrama de flujos del preámbulo previo al bucle de resolución temporal del código

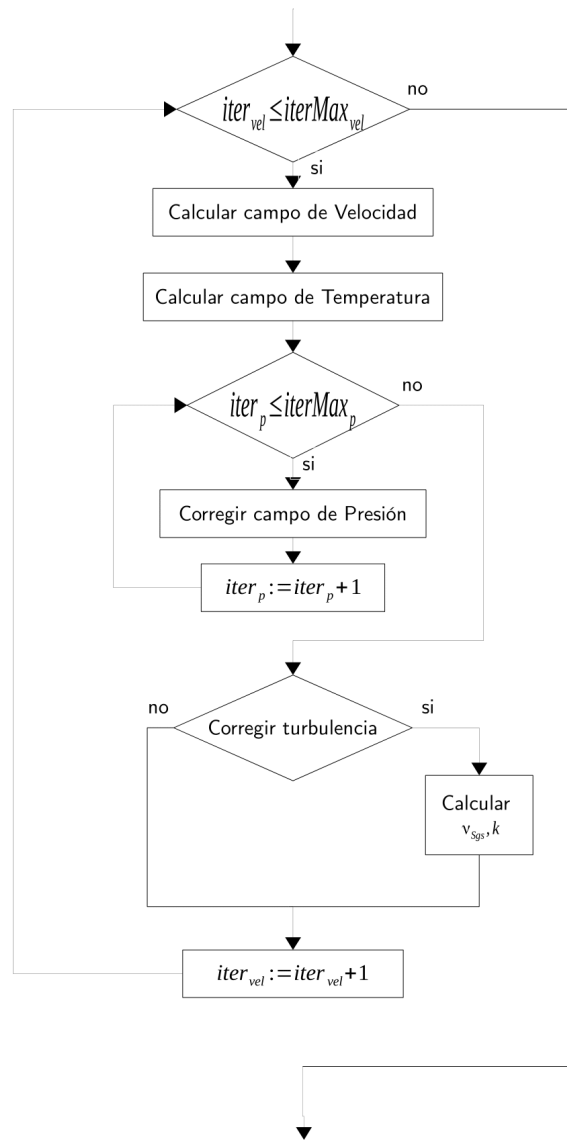


Figura 5.3: Diagrama de flujos del algoritmo PIMPLE para resolución de las ecuaciones de Navier-Stokes

5.3. Validación del código

El código utilizado en este trabajo, como se explicó previamente, corresponde a una modificación del código base “multiphaseInterFoam” desarrollado por la fundación OpenFOAM. Se le agregó, en primera instancia, la ecuación de energía para poder estudiar la distribución de temperatura al interior de fluidos multifásicos y posteriormente, la aproximación de Boussinesq para tratar la flotación producto de la variación en la densidad del fluido a distintas temperaturas. Debido a esto, es necesario verificar que la solución entregada por el nuevo código sea coherente con la física. La validación del código se realizó por partes. En primera instancia se presentan tres estudios de validación para la representación e implementación del método “Volume of Fluids” en OpenFOAM y finalmente, se simulan dos casos para validar la resolución de la ecuación de energía. El primero corresponde a la transferencia de calor hacia un fluido que circula al interior de un tubo de sección circular (Yunus A, 2007), y el segundo caso corresponde a la transferencia de calor por convección en una celda de Rayleigh-Bénard (Nasreddine Ouertatani et al., 2008)

5.3.1. Primer caso de validación del código multifásico: ”Zalesak Slotted disk problem”, Tony W.H. Sheu *, C.H. Yu, P.H. Chiu JCP (2009)

Se presenta el estudio realizado por Tony W.H. Sheu, C.H. Yu, P.H. Chiu JCP en 2009 en el cual estudiaron la topología de la interfaz en el problema de ”Zalesak Slotted disk” la cual es sometida a un campo de velocidad constante utilizando el código multifásicos de OpenFOAM, fue contrastado con la solución teórica y lo predicho por el método “Level-Set”. Las figuras 5.4 y 5.5 presentan los resultados que obtuvieron y la importancia del grado de refinamiento de la malla en la reconstrucción de la interfaz en el método VOF.

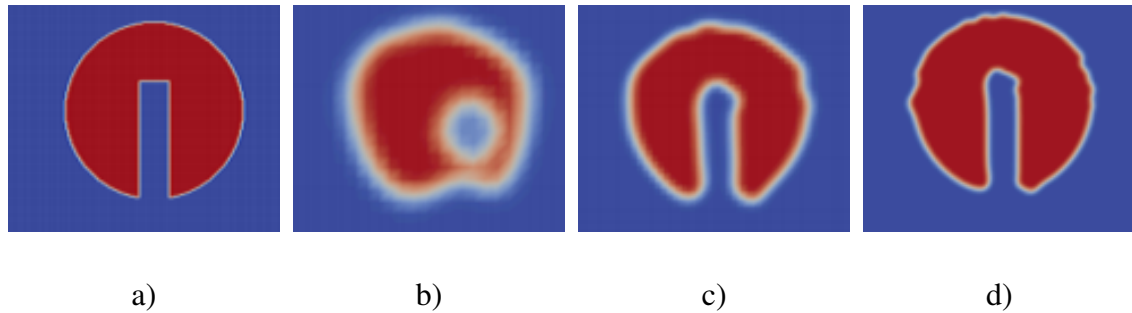


Figura 5.4: Representación de la posición de la interfaz y la influencia en la resolución de la malla para el problema “slotted disk”, *a)* Solución teórica del problema *b)* malla de 50×50 , *c)* malla de 100×100 y *d)* malla de 150×150

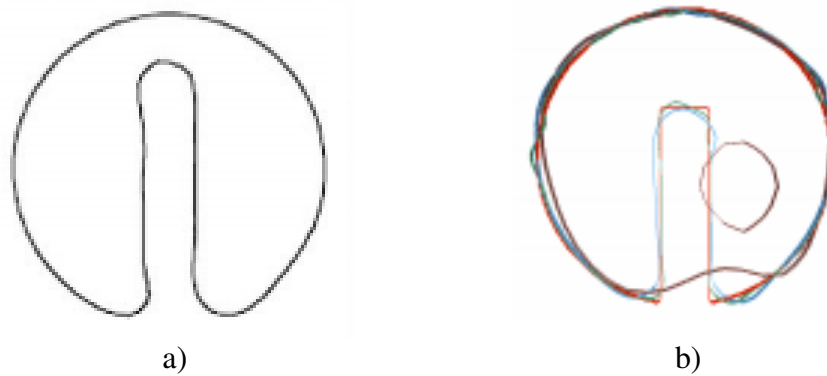


Figura 5.5: Comparación de la posición de la interfaz obtenida por el método “Level Set”(a) con la posición de la interfaz obtenida por el método “Volume of Fluid” implementado en OpenFOAM (b)

5.3.2. Segundo caso de validación del código multifásico: Deformación de una esfera tridimensional, Douglas Enright et. al, 2002

Se presenta el estudio realizado por Enright et. al en el cual estudiaron el proceso de deformación de la interfaz de una esfera tridimensional líquida sometida a un campo de velocidades oscilante en el tiempo y se contrastó los resultados obtenidos por OpenFOAM versus el método “Level-set” (figura 5.6).

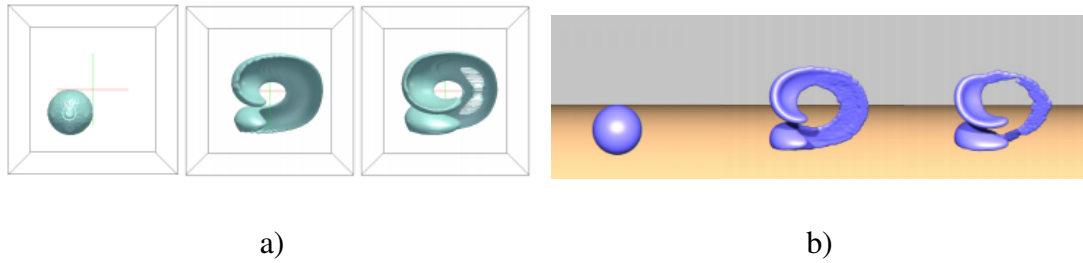


Figura 5.6: Comparación de la evolución temporal de la interfaz obtenida por el método “Volume of Fluid” implementado en OpenFOAM (a) con la solución predicha por el método “Level Set” (b)

5.3.3. Tercer caso de validación del código multifásico: Inestabilidad de Plateau-Rayleigh en un jet laminar

Se estudió el desempeño del código multifásico implementado en OpenFOAM para estudiar la inestabilidad que se produce cuando un jet liquido laminar esbelto se descompone en pequeñas gotas. Este chorro es desestabilizado por perturbaciones interfaciales cuya longitud de onda es superior a cierto valor crítico, dependiente del diámetro del jet. Las figuras 5.7 y 5.8 presentan la evolución de la interfaz luego de ocurrido la inestabilidad y la relación entre los diámetros de las gotas en función de la perturbación inicial.

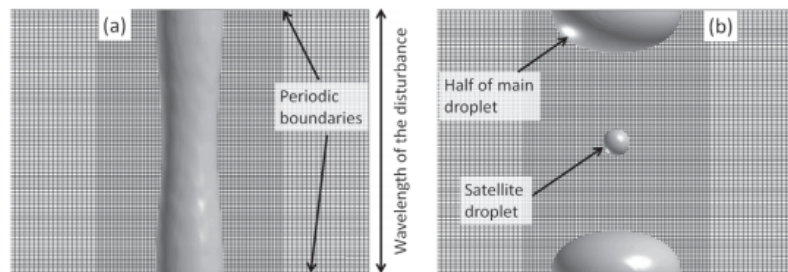


Figura 5.7: Inestabilidad de Plateau-Rayleigh para una perturbación con razón longitud de onda sobre diámetro $k = 0,75$. a) perturbación inicial y b) interfaz luego de la desintegración resultando en una gota principal y una gota satélite

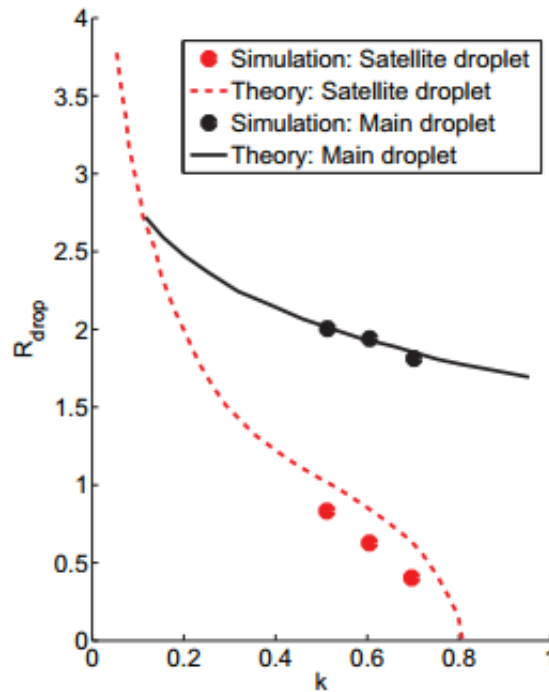


Figura 5.8: Comparación entre los radios de la gotas principal y satélite predicha por la teoría y lo reportado por la simulación para distintas razones de perturbación

Con base en los estudios presentados, es posible validar la correcta implementación, modelación de flujos multifásicos y rastreo de la interfaz desarrollada en los códigos multifásicos incluidos en OpenFOAM para problemas multifásicos que involucran interacciones de tipo líquido-líquido como gotas de fluido dispersas en otro.

5.3.4. Primer caso de validación para el código modificado: Transferencia de calor en tubería

La transferencia de energía se realiza mediante la imposición de una temperatura de pared de la tubería, superior a la temperatura de entrada del fluido como se aprecia en el esquema de la figura 5.9.

La temperatura presenta una distribución tanto en el eje radial como axial de la tubería. A modo de simplificación y para poder tratar el problema de manera unidimensional, se utiliza como medida característica para cada sección transversal, la temperatura promedio

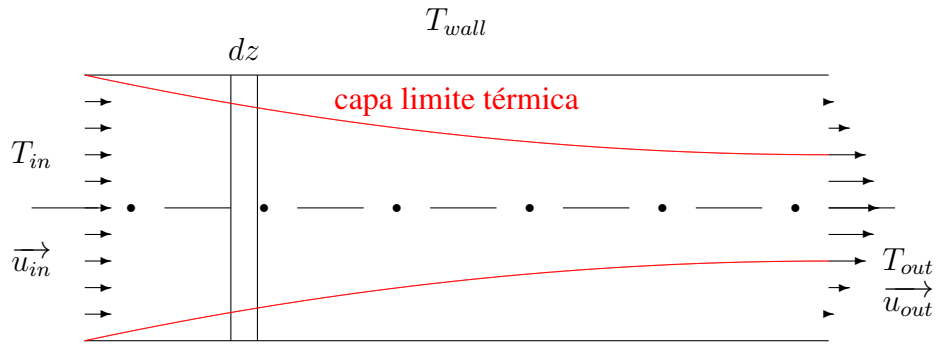


Figura 5.9: Esquema solución teórica caso validación

de esta en base al flujo másico que atraviesa la sección en función del radio:

$$T_b(z) = \frac{\int_0^R T(z, r) U(r) r dr}{\int_0^R U(r) r dr} \quad (5.1)$$

Para el cálculo de la transferencia de calor por convección desde las paredes de la tubería, se utiliza esta temperatura media de mezcla. El balance de energía en estado estacionario sobre el elemento diferencial dz

$$\rho C_p \bar{v}_z A T_b(z) - \rho C_p \bar{v}_z A (T_b(z) + dT_b) + h 2\pi R (T_{wall} - T_b(z)) dz = 0 \quad (5.2)$$

$$- \rho C_p \bar{v}_z A dT_b + h 2\pi R (T_{wall} - T_b(z)) dz = 0 \quad (5.3)$$

$$\frac{dT_b}{dz} = \frac{h 2\pi R}{\rho C_p \bar{v}_z A} (T_{wall} - T_b(z)) \quad (5.4)$$

donde ρ corresponde a la densidad del fluido, C_p el calor específico, \bar{v}_z la velocidad en dirección axial promedio sobre la sección, A representa el Área de la sección transversal de la tubería, R el radio de la tubería y h el coeficiente de convección. Realizando el cambio de variables:

$$\Theta(z) = \frac{T_{wall} - T_b(z)}{T_{in} - T_{wall}} \quad (5.5)$$

la ecuación (5.4) se reescribe en términos de la temperatura adimensional:

$$\frac{d\Theta}{dz} = \frac{h2\pi R}{\rho C_p \bar{v}_z A} \Theta(z) \quad (5.6)$$

La solución de la ecuación diferencial (5.6) viene dada por:

$$\Theta(z) = \exp\left(-\frac{h2\pi R}{\rho C_p \bar{v}_z A} z\right) \quad (5.7)$$

Para calcular el coeficiente de convección h , se utiliza el numero de Nusselt referido a tuberías circulares de radio r :

$$Nu = \frac{h2r}{k_f} \quad (5.8)$$

donde k_f corresponde a la conductividad térmica del fluido, y r al radio de la sección circular. El calculo del valor del numero de Nusselt para flujos laminares térmicos no desarrollados, con temperatura de pared constante, se encuentra ampliamente estudiado. Existe en la literatura diferentes correlaciones empíricas [Edwards et al., 1979]. Sin embargo para diferencias de temperatura entre la entrada y la temperatura de pared considerables, se debe tener en consideración la variación de la viscosidad [Sieder and Tate (1936)]:

$$Nu = 1,86 \left(\frac{RePr2r}{L}\right)^{\frac{1}{3}} \left(\frac{\mu_b}{\mu_{wall}}\right)^{0,14} \quad (5.9)$$

donde Re corresponde al numero de Reynolds, Pr al numero de Prandtl, r y L representan el radio y largo de la tubería respectivamente, μ_b la viscosidad del fluido evaluada a la temperatura media entre la entrada y la salida, μ_{wall} la viscosidad dinámica evaluada a la temperatura de la pared.

La simulación realizada para la validación del código está representada en el esquema 5.10, correspondiente a una tubería de largo $L2$ a la cual se le impone una temperatura de pared T_{wall} superior a la temperatura inicial T_{in} . Debido a que el código resuelve flujos multifásicos y presenta inestabilidades en la solución al trabajar con solo una fase, i.e tubería llena de agua, se agregó una sección de largo $L1$ en la cual se establece la zona de la interfaz entre el agua y el fluido 2. De esta forma se garantiza la estabilidad del código al tener presente 2 fases y se logra evitar que esta interfiera en el proceso de transferencia de calor que se quiere analizar.

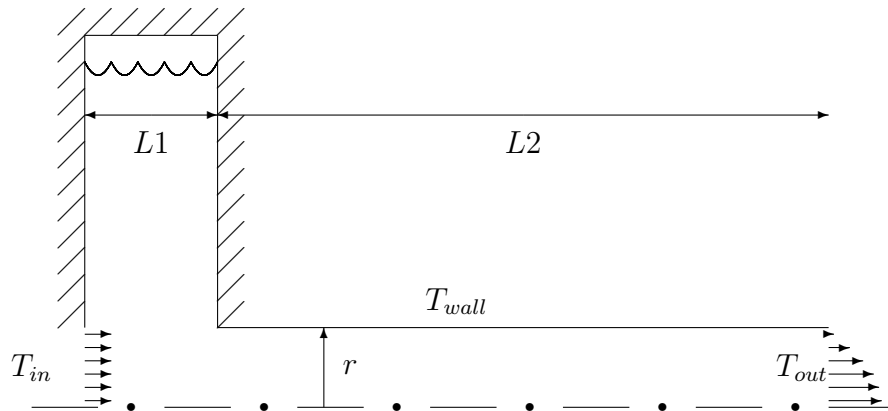


Figura 5.10: Esquema simulación caso validación

En el cuadro 5.1 se presentan las constantes termodinámicas y físicas utilizadas en el caso de validación.

Con el fin de minimizar las perturbaciones y oscilaciones que aparecen en la zona cercana a la interfaz producto de la rápida variación de las propiedades como la densidad o la viscosidad, se utilizaron dos fluidos de iguales propiedades, pues no nos interesa la interacción entre ambos fluidos, solo resolver la transferencia de calor al interior de la tubería.

Las condiciones de contorno de la simulación se resumen a continuación:

- Pared adiabática para la primera zona en donde se encuentra el fluido 2.
- Temperatura de pared impuesta T_{wall} igual a 30 grados
- Temperatura de entrada del fluido 1 T_{in} de 0 grados
- Perfil de velocidad de entrada del fluido 1 constante a lo largo de la sección transversal, variando para obtener números de Reynolds al interior de la tubería de 1,000, 1,250, 1,500, 1,750 y 2,000

En la figura 5.11 se puede apreciar el perfil de temperatura que se alcanza en el fluido al salir de la tubería, se puede observar que en todos los casos simulados, la influencia térmica de la pared no logra generar efectos apreciables sobre el eje de la tubería, lo cual

Propiedad	Fluido 1	Fluido 2
$\rho \left[\frac{kg}{m^3} \right]$	1,000	1,000
$Cp \left[\frac{J}{kgK} \right]$	4,185	4,185
$Pr[-]$	5	5
$\nu \left[\frac{m^2}{s} \right]$	$1,002 \times 10^{-6}$	$1,002 \times 10^{-6}$

Cuadro 5.1: Parámetros físicos utilizados en la simulación de validación

$L1 [mm]$	1
$L2 [mm]$	1,000
$r [mm]$	4

Cuadro 5.2: Dimensiones físicas del dominio utilizado en la validación

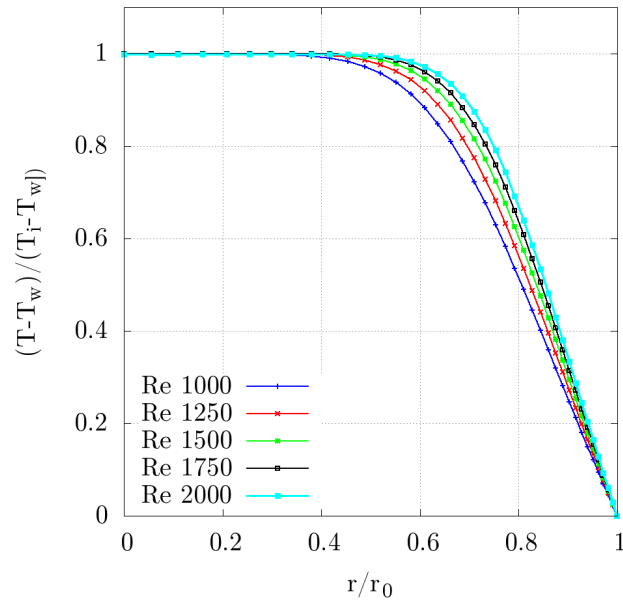


Figura 5.11: Perfiles de temperaturas a la salida de la tubería para distintos números de Reynolds

implica que el flujo aun no se encuentra térmicamente desarrollado, requisito necesario para poder comparar dichas simulaciones con la solución teórica (5.9).

Al analizar los resultados obtenidos en las simulaciones para distintos valores del numero de Reynolds y su comparación con lo predicho por la teoría, como se aprecia en la imagen 5.12, se observa un comportamiento coherente de física resuelta por el código en lo que se refiere a la transferencia de energía al interior de la tubería. La diferencia existente entre la simulación y la solución empírica se puede explicar por la naturaleza de esta ultima, en la cual se aproxima el cambio en las propiedades del fluido utilizando la temperatura media entre otros. Como se observa en la figura 5.13, el error porcentual cometido por la simulación es decreciente a medida que se incrementa el numero de Reynolds, oscilando entre un 1 % y 2 %, el cual esta dentro del rango de precisión para la aproximación de la solución empírica por lo que se considera aceptable.

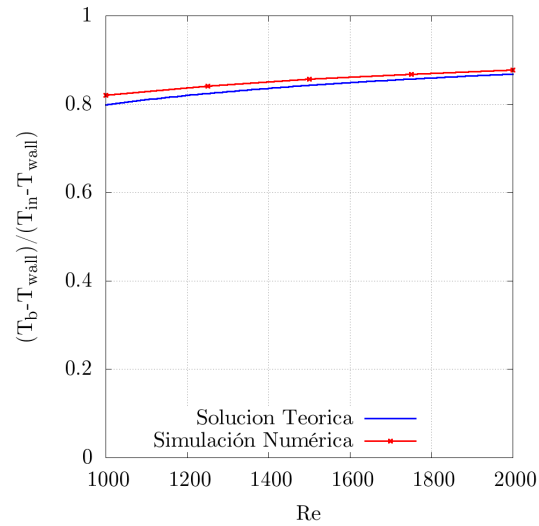


Figura 5.12: Comparación entre la simulación y la solución teórica para la temperatura media de masa adimensionalizada obtenida a la salida de la tubería a distintos numeros de Reynolds simulados.

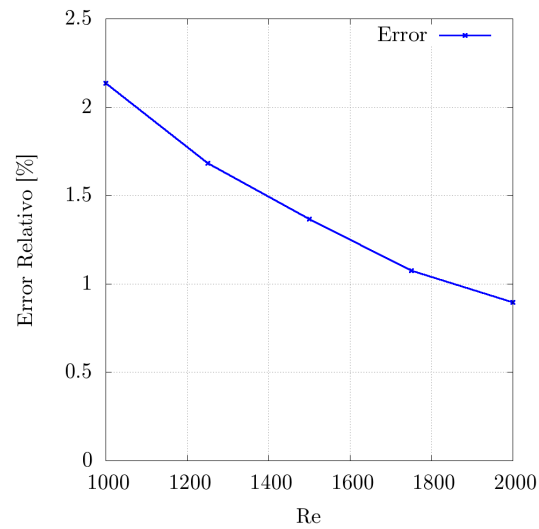


Figura 5.13: Error relativo porcentual cometido en la simulación en comparación con la solución empírica.

5.3.5. Segundo caso de validación para el código modificado: Convección Rayleigh-Bénard

El segundo caso de validación estudiado corresponde a la convección al interior de una celda de Rayleigh-Bénard en dos dimensiones desarrollado por Ouertatani et al (Nasreddine Ouertatani et al., 2008), en el cual se estudia la transferencia de calor al interior de la celda llena de aire para distintos valores del número de Rayleigh $\frac{g\beta\Delta TL^3}{\nu\alpha}$, variando de 10^4 hasta 10^6 , donde g es la aceleración de gravedad, β el coeficiente de expansión térmica del fluido, ΔT la diferencia de temperaturas, L un largo característico de la geometría, α la conductividad térmica del fluido y ν la viscosidad cinemática respectivamente. Para esto, se utilizó la forma adimensionalizada de las ecuaciones de Navier-Stokes y conservación de energía descritas en los capítulos previos:

$$\frac{\partial u_j^*}{\partial x_j^*} = 0 \quad (5.10)$$

$$\frac{\partial u_i^*}{\partial t^*} + \frac{\partial (u_i^* u_j^*)}{\partial x_j^*} = -\frac{p^*}{\partial x_i^*} + \left(\frac{Pr}{Ra}\right)^{\frac{1}{2}} \frac{\partial^2 u_i^*}{\partial x_j^* \partial x_j^*} + \delta_{i2} T^* \quad (5.11)$$

$$\frac{\partial T^*}{\partial t^*} + \frac{\partial (u_i^* T^*)}{\partial x_i^*} = (PrRa)^{-\frac{1}{2}} \frac{\partial^2 T^*}{\partial x_i^* \partial x_i^*} \quad (5.12)$$

donde $u_i^* = (u^*, v^*)$, p^* y T^* corresponden respectivamente al campo de velocidades adimensionalizada, la componente deviatórica del tensor de presión adimensionalizada y el campo de temperaturas adimensionalizadas; δ_{ij} representa el delta de Kronecker.

Estas ecuaciones se obtienen adimensionalizando las variables según sus respectivas

escalas características:

$$x^* = \frac{x}{H} \quad (5.13)$$

$$t^* = t \frac{(g\beta H (T_{hot} - T_{cold}))^{\frac{1}{2}}}{H} \quad (5.14)$$

$$u_i^* = \frac{u_i}{(g\beta H (T_{hot} - T_{cold}))^{\frac{1}{2}}} \quad (5.15)$$

$$p^* = \frac{p}{\rho (g\beta H (T_{hot} - T_{cold}))} \quad (5.16)$$

$$T^* = \frac{T - \frac{1}{2}(T_{hot} + T_{cold})}{(T_{hot} - T_{cold})} \quad (5.17)$$

donde H corresponde al alto de la celda de Rayleigh-Bénard, g a la aceleración de gravedad, β el coeficiente de dilatación volumétrica del fluido, T_{hot} y T_{cold} las temperaturas de las paredes inferior y superior respectivamente, y ρ la densidad del fluido.

Los parámetros adimensionales que rigen el fenómeno corresponden a los números de Rayleigh y Prandtl, los cuales se calculan como sigue:

$$Ra = \frac{g\beta (T_{hot} - T_{cold}) H^3}{\alpha\nu} \quad (5.18)$$

$$Pr = \frac{\nu}{\alpha} \quad (5.19)$$

donde α y ν corresponden a la difusividad térmica y la viscosidad cinemática respectivamente.

La figura 5.14 representa esquemáticamente el caso y sus condiciones de contorno.

Los resultados obtenidos mediante las simulaciones muestran un comportamiento coherente en comparación con lo reportado por Ouertatani et al, 2008. Al comparar los perfiles de velocidades en ambas direcciones se observa congruencia tanto en el perfil desarrollado como en las magnitudes de las velocidades alcanzadas como se puede apreciar en las figuras 5.15 y 5.16. El error porcentual máximo es del orden del 8% para el caso con un numero de Rayleigh $Ra = 10^4$, decreciendo a medida que este parámetro aumenta. Al analizar los campos de Temperaturas representados en la figura 5.17, se observa congruencia entre las estructuras internas desarrolladas en ambos casos, sin apreciarse diferencias significativas entre ambos resultados. Con

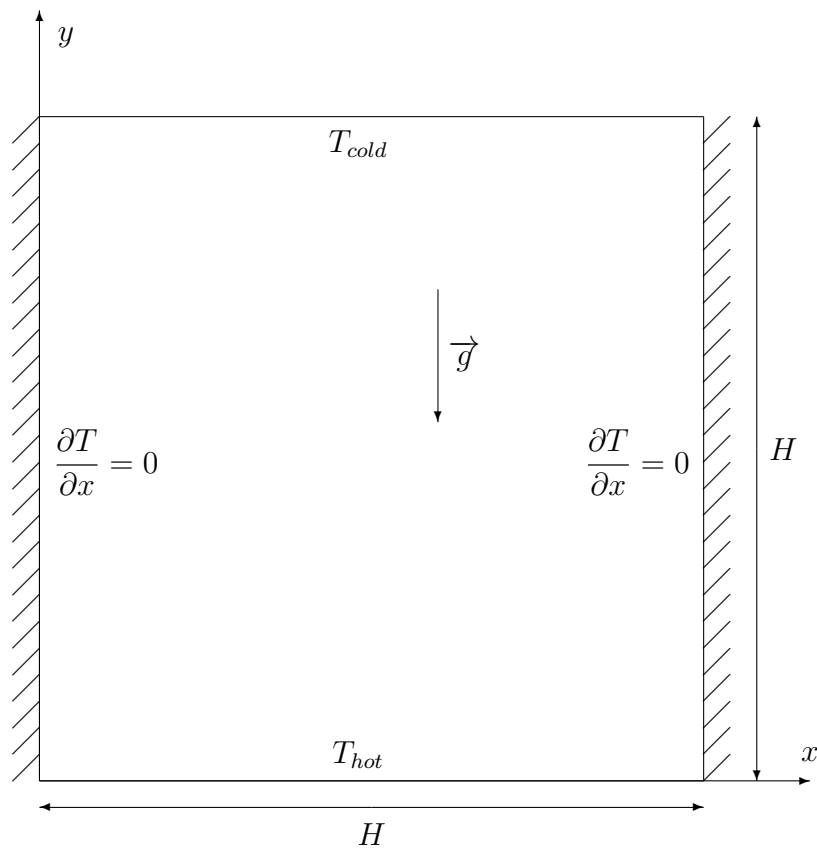


Figura 5.14: Esquema de la celda de convección Rayleigh-Bénard

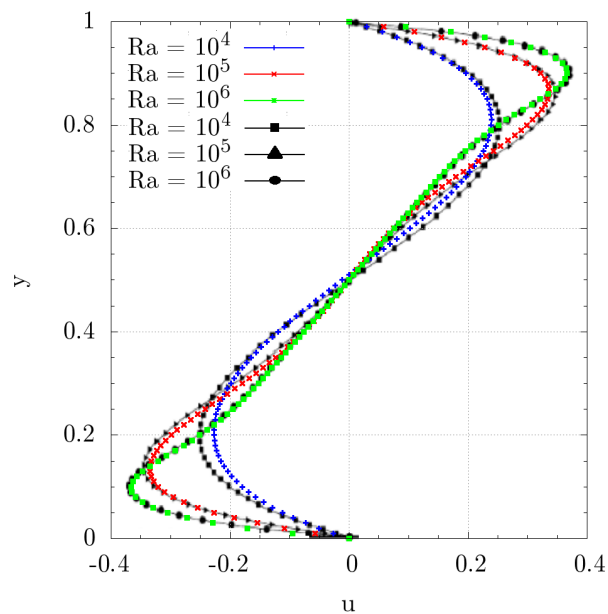


Figura 5.15: Perfil de velocidades en dirección x sobre $x = 0,5$, en negro los resultados reportados por Ouertatani et al, 2008 y en color los resultados obtenidos con el código modificado.

respecto a la transferencia de calor, el perfil para el número de Nusselt local presenta un comportamiento similar a los resultados previos. El error porcentual máximo en este caso es del orden del 8 % y ocurre para la simulación con menor número de Rayleigh, con un comportamiento decreciente a medida que incrementamos dicho parámetro.

Dado los resultados obtenidos por el código en ambos escenarios y su respectiva comparación con los resultados teóricos, es posible validar el código y su representación física del fenómeno de transferencia de calor y del fenómeno de flotación.

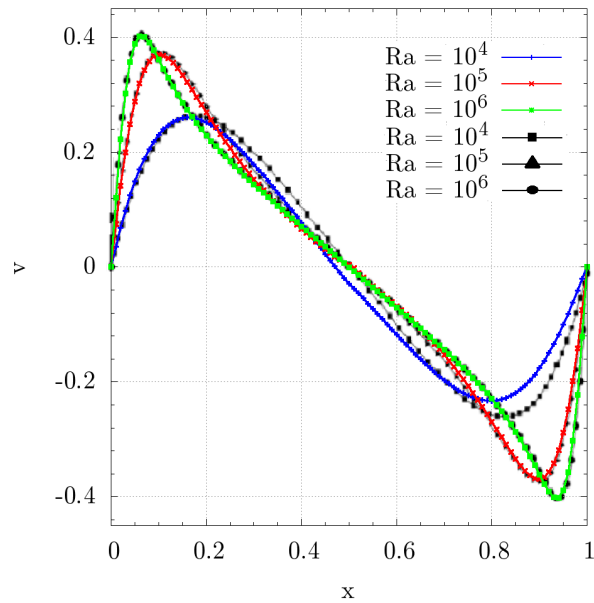


Figura 5.16: Perfil de velocidades en dirección y sobre $y = 0,5$, en negro los resultados reportados por Ouertatani et al, 2008 y a color los resultados obtenidos con el código modificado.

5.4. Problema de estudio

El presente estudio busca comprender el proceso de mezclado que ocurre al interior de un diseño particular de inyector de turbina a gas utilizado en distintas centrales térmicas. Los objetivos principales del proyecto se detallan a continuación:

- Estudiar el proceso de mezclado de los distintos fluidos al interior del inyector, determinando zonas de acumulación de los fluidos y predominancia de las fases.
- Determinar el comportamiento de los fluidos bajo distintos caudales de alimentación.
- Estudiar la influencia de un campo de temperatura externo sobre la mezcla de los fluidos
- Estudiar la presión al interior de la línea de alimentación del agua, con el fin de determinar variaciones en las condiciones de operación de las bombas alimentadoras.

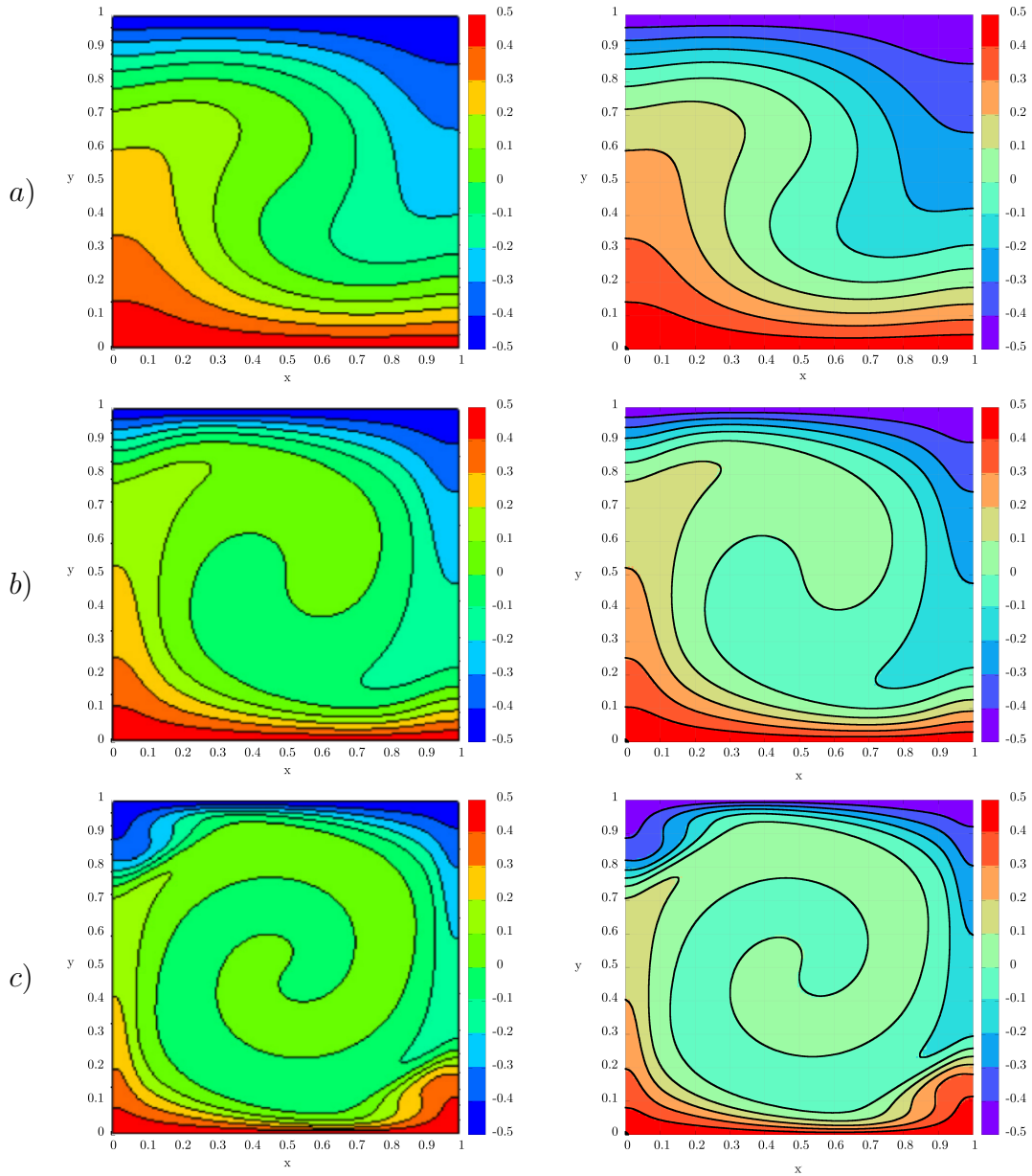


Figura 5.17: Distribución de temperaturas al interior de la cavidad. A la izquierda los resultados reportados por Ouertatani et al, 2008 y a la derecha los obtenidos en la simulación para a) $Ra = 10^4$, b) $Ra = 10^5$, c) $Ra = 10^6$

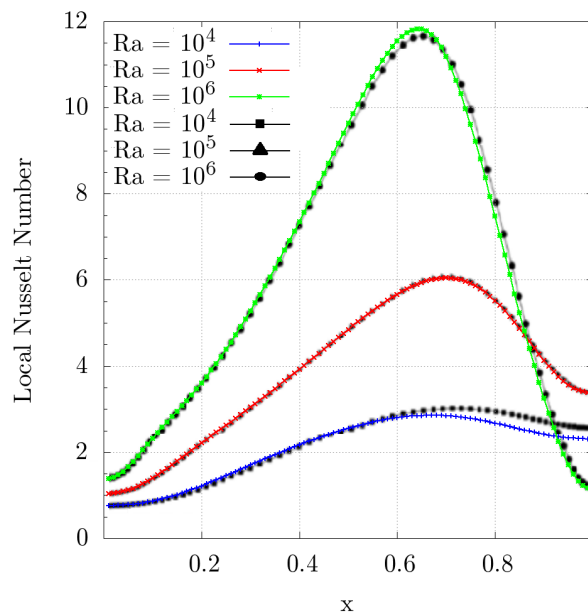


Figura 5.18: Numero de Nusselt local sobre la pared inferior $y = 0$ reportado por a) Ouertatani et al, 2008 y b) resultados obtenidos con el código.

- Estudiar la frecuencia de oscilación en la presión a la salida del inyector para determinar si existen variaciones de esta al variar el caudal de los fluidos y determinar las frecuencias de oscilación predominantes para las distintas configuraciones

5.5. Dominio Físico y Discretización Espacial

El dominio del problema corresponde al interior del inyector de una turbina a gas como se observa en las figuras 5.19 y 5.20. El inyector está compuesto por:

- Una línea de alimentación para el petróleo de color verde en la imagen 5.19, la cual tiene en su extremo una placa con cuatro perforaciones distribuidas de forma simétrica de diámetro 3 milímetros, y un cono truncado a continuación de la placa. Esta configuración permite una distribución de la velocidad máxima lejos del eje de la tubería facilitando el proceso de mezclado de ambas fases.
- Una línea de alimentación para el agua que se ubica de manera concéntrica con la tubería de petróleo de color azul en la imagen 5.19.
- Una cámara de pre-mezclado de color rojo en la imagen 5.19 en la que inyectan directamente las líneas de petróleo y agua. En esta cámara se produce el proceso de mezclado de ambas fases previo al proceso de atomización y de ignición en la cámara de combustión.
- Una línea de alimentación de gas natural o gas licuado de petróleo de color gris en la imagen 5.19, esta línea inyecta el combustible a la cámara de combustión de manera paralela a la línea de petróleo.

Debido a que la línea de alimentación de gas es paralela a las líneas de petróleo y agua, y que ésta no tiene interferencia en la cámara de pre-mezcla, no se incluirá esta en el dominio de las simulaciones.

Las dimensiones y detalles de las piezas que componen el inyector se detallan en la figura 5.21.

La Discretización espacial del dominio se realizó mediante el uso del software de código libre GMSH (Geuzaine y Remacle, 2009), generando una malla semi-estructurada por bloques compuesta de volúmenes finitos hexahedricos. Se utilizó en primera instancia una malla gruesa compuesta por aproximadamente 200.000 de estos volúmenes, para refinarla progresivamente y verificar la convergencia de esta. Se observó que la solución obtenida dentro de la cámara de mezcla se estabiliza con una malla compuesta por aproximadamente 1.100.000 volúmenes como se representa en la figura 5.22 pero el

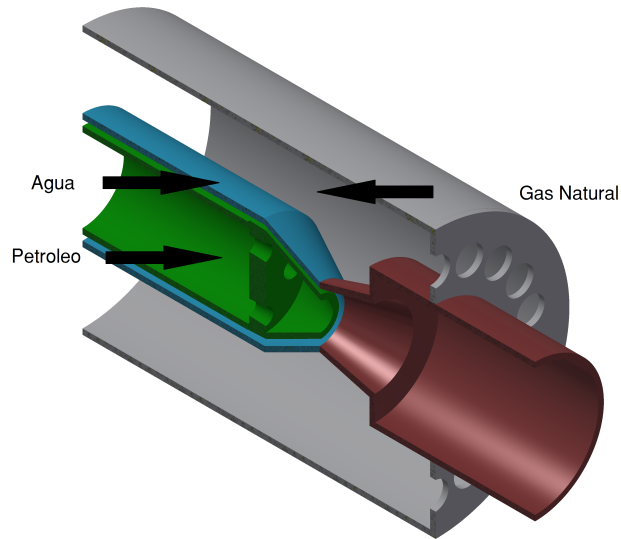


Figura 5.19: Vista en corte del inyector de la turbina a gas. En verde la línea de admisión de petróleo, en azul la línea de admisión de agua, en rojo la cámara de pre-mezclado y en gris la línea auxiliar de gas

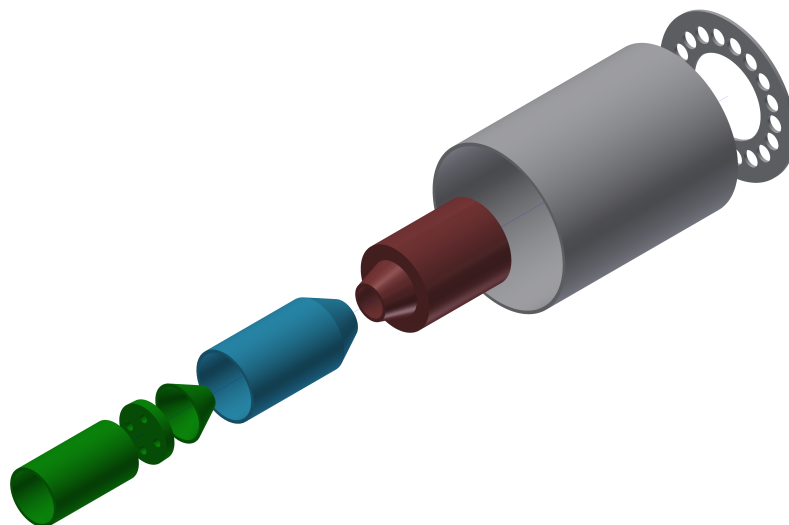


Figura 5.20: Vista en explosión del inyector de la turbina a gas

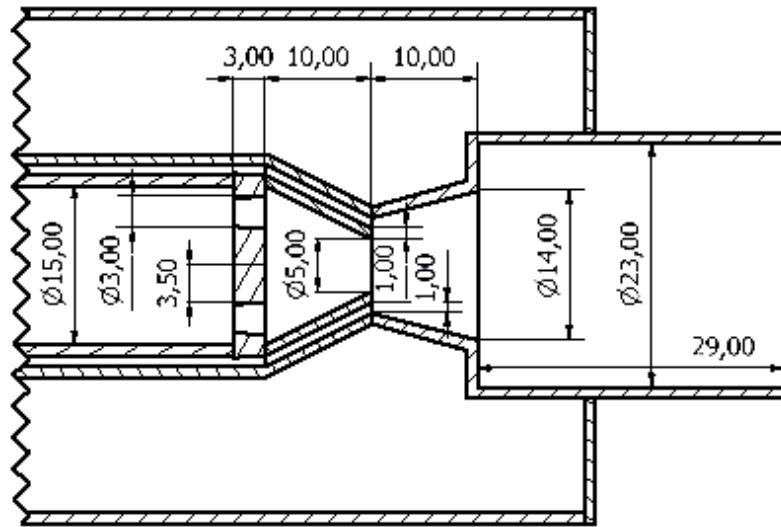


Figura 5.21: Plano del inyector de la turbina a gas estudiado (dimensiones en milímetros)

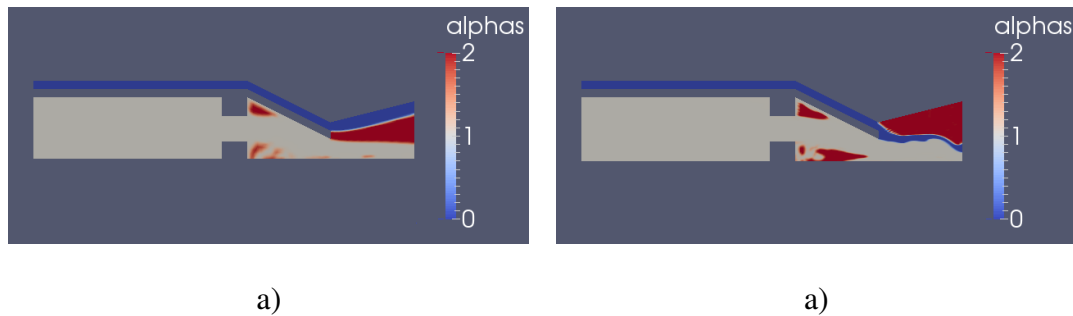


Figura 5.22: Comparación de la solución obtenida para una malla de a) 200.000 volúmenes y b) 600.000 volúmenes

coste computacional y tiempo de calculo necesario no hacia factible la realización de la simulación por lo que se optó por utilizar una malla con el mismo nivel de refinamiento al interior de la cámara de mezcla de los fluidos y un refinamiento menor al interior de las tuberías independientes.

Se definió finalmente utilizar una malla compuesta por aproximadamente 730.000 volúmenes. Se optó por utilizar esta configuración de la malla pues permite una mayor precisión en la solución obtenida, disminuye la necesidad de recursos computacional y acelera los tiempos de calculo en comparación con el uso de una malla no estructurada. En las figuras 5.23 y 5.24 se representa la malla utilizada en las simulaciones

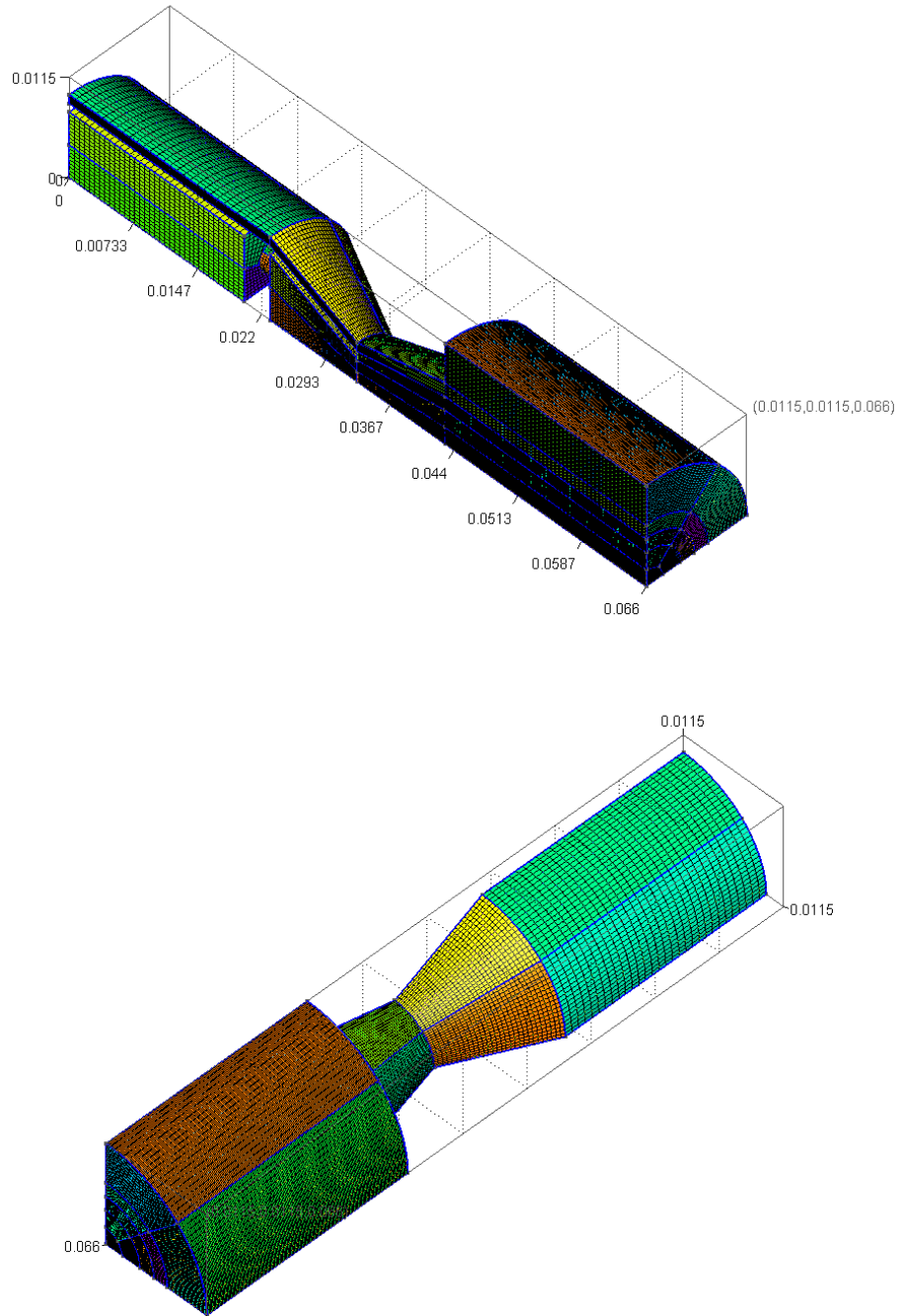


Figura 5.23: Malla tridimensional con cuasi-simetría axial

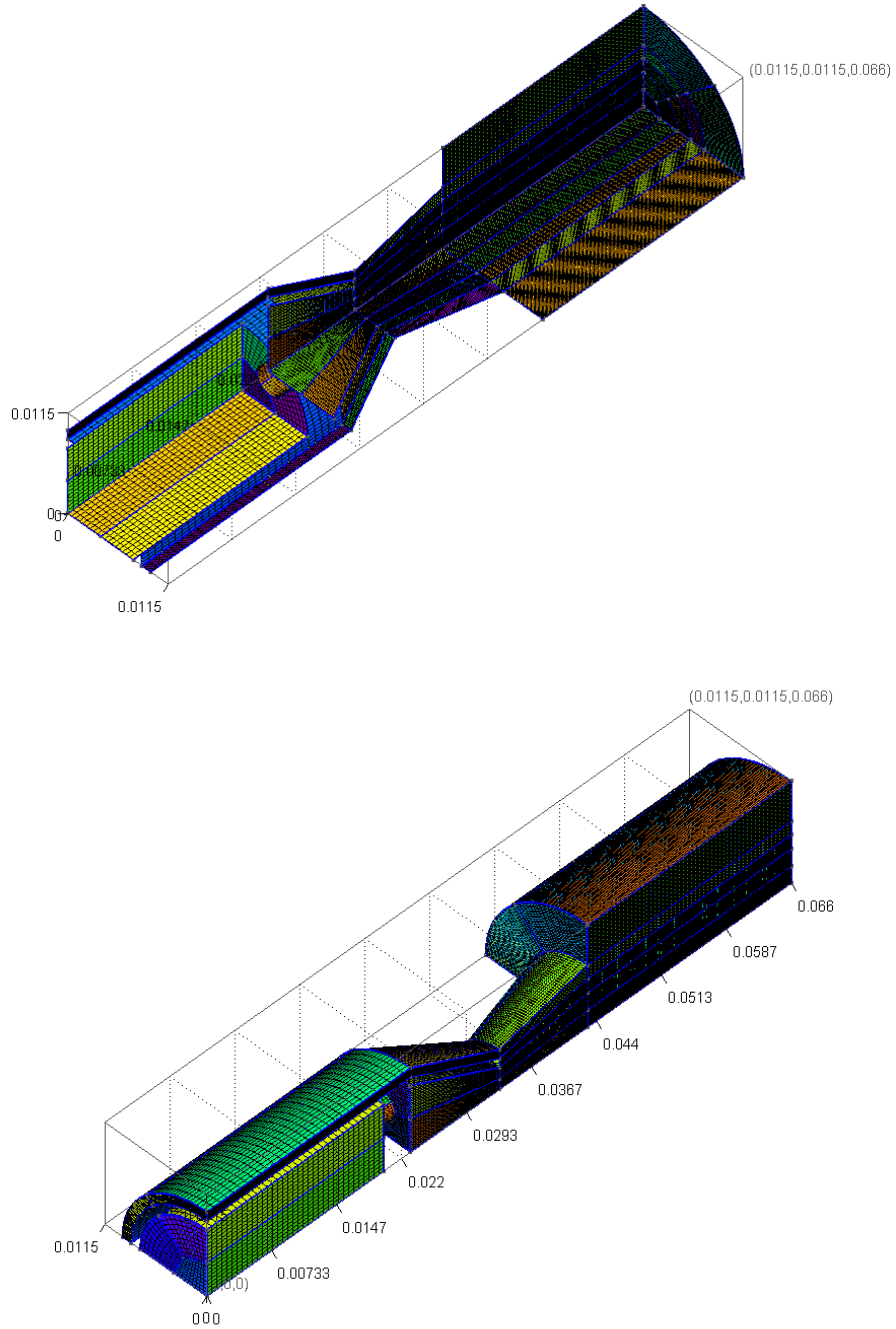


Figura 5.24: Malla tridimensional con cuasi-simetría axial

5.6. Condiciones de contorno e inicial

En esta sección se explican las condiciones iniciales y de contorno utilizadas en las simulaciones. El dominio físico utilizado esta conformado por cinco fronteras físicas sobre las cuales se deben especificar las condiciones. Estas son:

- Entrada de petróleo
- Entrada de agua
- Plano de simetría
- Pared
- Salida

Las ecuaciones a resolver implican que se deben establecer condiciones de contorno para los siguientes campos:

- Velocidad
- Presión
- Temperatura
- Fracción Volumétrica de Petróleo
- Fracción Volumétrica de Agua
- Energía Cinética Turbulenta de sub-malla
- Viscosidad de sub-malla

Esta ultima es necesaria debido a la forma en que esta construido el código OpenFOAM, aún cuando no se resuelva ecuación diferencial alguna para dicha variable por lo que la condición que se establezca no tiene efecto alguno sobre la simulación.

5.6.1. Entrada de Petróleo

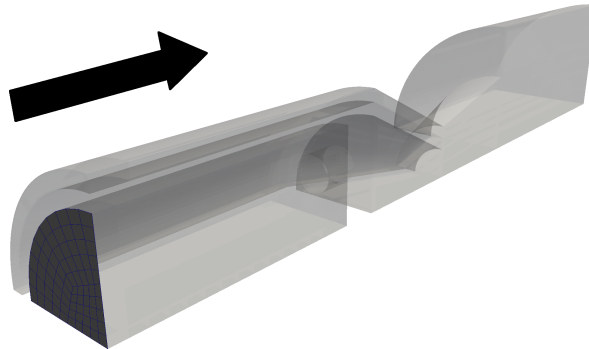


Figura 5.25: Representación de la frontera correspondiente a la entrada para el caudal de petróleo.

Velocidad Para las simulaciones de tipo RANS (Reynolds Averaging Navier-Stokes), basta con especificar un perfil de velocidades promedio del campo turbulento, mientras que en las simulaciones de tipo Large Eddy la condición de entrada del campo de velocidades cumple un rol fundamental en el desarrollo de la turbulencia aguas abajo de la inyección. Debido a esto, se utilizan tres métodos principales para especificar una condición de contorno turbulenta:

- Realizar una simulación precursora en paralelo sobre la cual se desarrolle un campo de velocidades turbulento real, que cumpla tanto con el espectro de energías como con la coherencia espacial de las estructuras propias de la turbulencia. El perfil de velocidades que se utiliza para la condición de contorno del dominio real corresponde al muestreo sobre un plano paralelo en el dominio precursor como se observa en la figura 5.26.
- Generar un perfil de velocidad turbulento de manera artificial que cumpla con un flujo másico promedio igual al deseado en la simulación. Existen diversas formas para lograr esto. La más básica consiste en agregar una pequeña oscilación aleatoria sobre el perfil de velocidad promedio. También se puede realizar un perfil de velocidades en el espacio de Fourier, que es posteriormente modulado de acuerdo al espectro de energía de Kolmogorov con el fin de generar un perfil de velocidades que

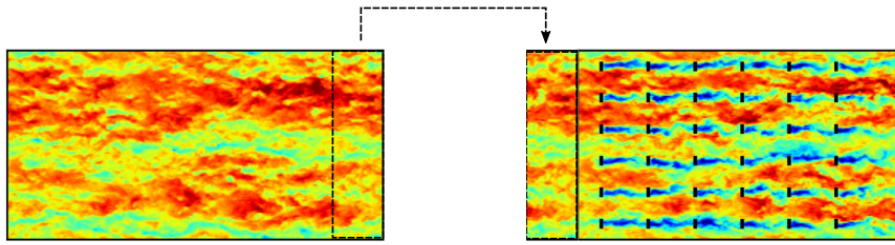


Figura 5.26: Ilustración del método de simulación precursora (Stevens et al. 2014b), a la izquierda el dominio de la simulación precursora y a la derecha el dominio de la simulación real, la región demarcada sobre el dominio precursor es inyectada sobre el dominio de la simulación real

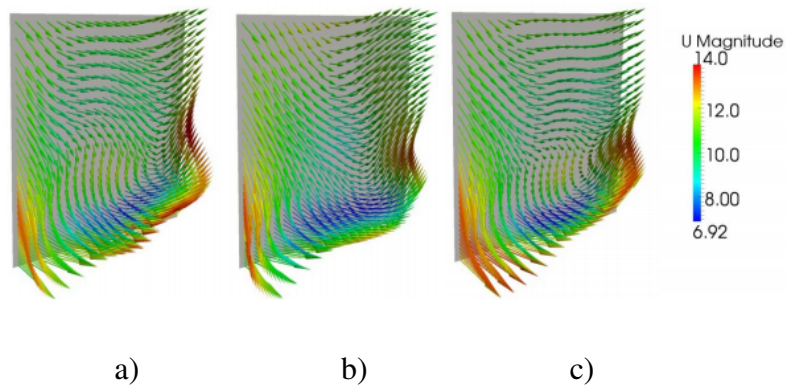


Figura 5.27: Turbulencia sintética como condición de entrada para una simulación Large-Eddy con coherencia espacial y su evolución en el tiempo para a) $t = 0,001$, b) $t = 0,002$ y c) $t = 0,003$ segundos (Jørgensen,2012)

tiene un comportamiento más acorde con la naturaleza de la turbulencia. Finalmente existen diversos métodos con que permiten agregar al perfil de velocidades una coherencia espacio-temporal acorde con las estructuras características de un flujo turbulento como lo son los vórtices. La figura 5.27 representa un perfil generado mediante turbulencia sintética con coherencia espacial y temporal (Jørgensen, 2012).

- Generar una zona entre la entrada del fluido y el área de interés que se desea estudiar con el fin que la turbulencia se desarrolle en dicha zona para que el fluido alcance la zona de interés con una turbulencia real completamente desarrollada.

Debido a la simplicidad de implementación y la complejidad de la geometría propia del

dominio de estudio, se optó por utilizar el último método, definiendo un perfil de entrada homogéneo y dejando que la turbulencia se desarrolle en la zona previa a la cámara de mezcla. Se utilizaron perfiles de velocidad homogéneo, variando desde un numero de Reynolds 10000 hasta 25000 con incrementos graduales de 5000.

Presión Se ajustó el gradiente de presión de tal manera que el flujo másico a través de la superficie corresponda exactamente con el especificado por la condición de contorno de la velocidad. Condición de contorno denominada “FixedFluxPressure” en OpenFOAM.

Temperatura Valor fijo y constante de 20 grados Celsius

Fracción Volumétrica Petroleo Valor fijo y constante igual a uno

Fracción Volumétrica Agua Valor fijo y constante igual a cero

Energía Cinética Turbulenta de sub-malla Valor fijo y constante igual a $10^{-5} [m^2s^{-2}]$

Viscosidad turbulenta de sub-malla Condición de gradiente cero

5.6.2. Entrada de Agua

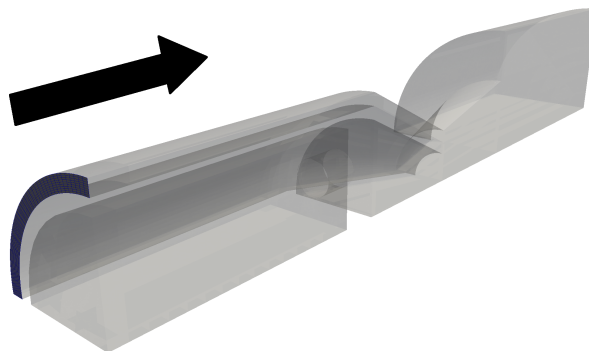


Figura 5.28: Representación de la frontera correspondiente a la entrada para el caudal de agua

Velocidad Perfil de velocidades homogéneo el cual se varia desde un Reynolds 2000 hasta 5000 con incrementos graduales de 1000.

Presión Se ajustó el gradiente de presión de tal manera que el flujo másico a través de la superficie corresponda exactamente con el especificado por la condición de contorno de la velocidad. Condición de contorno denominada “FixedFluxPressure”.

Temperatura Valor fijo y constante de 20 grados Celsius

Fracción Volumétrica Petroleo Valor fijo y constante igual a cero

Fracción Volumétrica Agua Valor fijo y constante igual a uno

Energía Cinética Turbulenta de sub-malla Valor fijo y constante igual a $10^{-5} [m^2s^{-2}]$

Viscosidad turbulenta de sub-malla Condición de gradiente cero

5.6.3. Pared

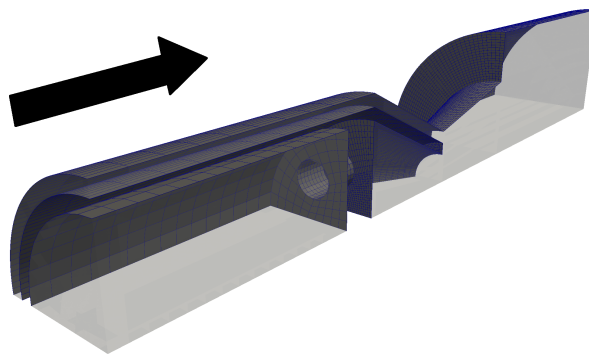


Figura 5.29: Representación de la frontera correspondiente a las paredes del inyector

Velocidad Sobre las paredes se impuso la condición física de no deslizamiento explicada previamente $U = 0$.

Presión Se ajustó el gradiente de presión de tal manera que el flujo másico a través de la superficie corresponda exactamente con el especificado por la condición de contorno de la velocidad. Condición de contorno denominada “FixedFluxPressure”.

Temperatura Sobre la pared se estableció un gradiente de temperatura de $\frac{\partial T}{\partial n} = 3,000 \left[\frac{K}{m} \right]$ resultante de la transferencia de calor por radiación de la llama sobre la cara externa del inyector

$$\frac{\partial T}{\partial r} = - \frac{\sigma \varepsilon (T_{flame}^4 - T_{inyector}^4)}{k_{inyector}} \quad (5.20)$$

donde σ , ε y $k_{inyector}$ corresponden a la constante de Stefan-Boltzmann, la emisividad de la cara exterior del inyector y la conductividad térmica del material que este construido el inyector respectivamente, T_{flame} y $T_{inyector}$ la temperatura de la llama al interior de la cámara de combustión y la temperatura externa del inyector respectivamente. La transferencia de calor que recibe el inyector desde el exterior es bastante mas compleja que la presentada en el estudio. Esta simplificación se realizó debido al elevado costo computacional que presentaba el modelo completo de transferencia de calor por radiación y posterior conducción a través del metal del inyector, el cual dependía a su vez de la temperatura interna del fluido.

Fracción Volumétrica Petroleo Gradiente cero, es decir, el agua se adhiere a las paredes, generando un angulo de contacto de la interfaz de 90° .

Fracción Volumétrica Agua Gradiente cero, es decir, el agua se adhiere a las paredes, generando un angulo de contacto de la interfaz de 90° .

Energía Cinética Turbulenta de sub-malla Condición de gradiente cero.

Viscosidad turbulenta de sub-malla Condición de gradiente cero.

5.6.4. Salida

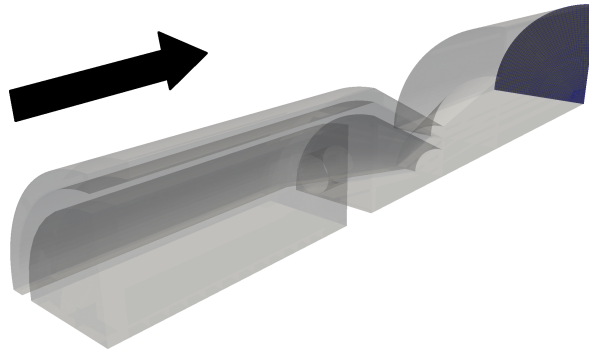


Figura 5.30: Representación de la frontera correspondiente a la salida del inyector

Velocidad Condición de gradiente cero.

Presión Presión total correspondiente a la presión estática más la dinámica se fija en un valor igual a 0

$$p_0 = p + \frac{1}{2}\rho|U|^2 \quad (5.21)$$

Temperatura Para la condición de salida se impuso un gradiente de temperatura negativo (ingresa energía al inyector) debido al frente de combustión que se encuentra a la derecha del inyector. Este gradiente de temperatura se aproximó como el flujo de calor promedio que viaja aguas arriba en una simulación de la combustión de un spray de petróleo diésel con igual flujo másico y dimensiones que las tratadas en el presente estudio.

Fracción Volumétrica Petróleo Condición de gradiente cero.

Fracción Volumétrica Agua Condición de gradiente cero.

Energía Cinética Turbulenta de sub-malla Condición de gradiente cero.

Viscosidad turbulenta de sub-malla Condición de gradiente cero.

5.6.5. Simetría

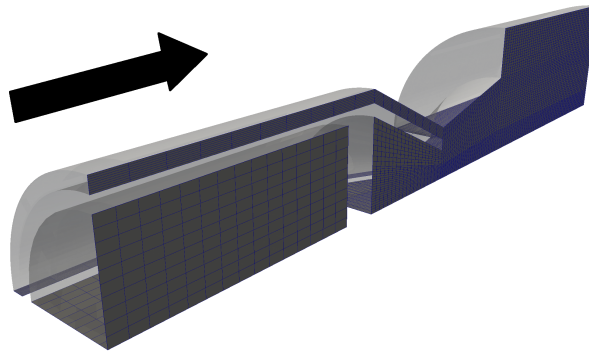


Figura 5.31: Representación de la frontera correspondiente a los planos de simetría

Velocidad Condición de simetría explicada previamente.

Presión Condición de simetría explicada previamente.

Temperatura Condición de simetría explicada previamente.

Fracción Volumétrica Petroleo Condición de simetría explicada previamente.

Fracción Volumétrica Agua Condición de simetría explicada previamente.

Energía Cinética Turbulenta de sub-malla Condición de simetría explicada previamente.

Viscosidad turbulenta de sub-malla Condición de simetría explicada previamente.

5.6.6. Condiciones iniciales

En cuanto a la inicialización de los distintos campos para la simulación correspondiente al caso de control se realizó como sigue:

- Campo de velocidad homogéneo en todo el dominio igual a cero $\vec{U} = 0$, es decir, condición de reposo.

- Campo de presión homogéneo en todo el dominio igual a cero:
 $p = 0$
- Campo de temperatura homogéneo igual a 20 grados Celsius en todo el dominio:
 $T = 20$.
- Fracción volumétrica de agua con un valor $\alpha_{agua} = 1$ al interior de la tubería de agua y cero en el resto del dominio.
- Fracción volumétrica de petróleo con un valor $\alpha_{oil} = 1$ al interior de la tubería de petróleo y cero en el resto del dominio.
- campo de energía turbulenta de sub-malla homogéneo en todo el dominio igual a cero:
 $k_{sg} = 0$.
- campo de viscosidad de sub-malla homogéneo en todo el dominio igual a cero:
 $\nu_{sg} = 0$.

El estado estacionario alcanzado con la simulación del caso de control fue utilizado posteriormente como condición inicial para los demás casos simulados con el fin de disminuir los tiempos de calculo para lograr el estado estacionario.

5.7. Casos de estudio

Se realizaron distintas simulaciones variando los caudales de petróleo y agua que se inyectan en la cámara de mezcla. Las tablas 5.3, 5.4 y 5.5 resumen los casos estudiados, parámetros físicos utilizados y el valor de la tensión superficial entre los fluidos respectivamente.

El horizonte temporal simulado en todos los casos se calculó en función del tiempo de advección característico en cada caso, este tiempo se calcula como sigue:

$$t_{adv} = \frac{\text{Velocidad característica}}{\text{Largo característico}} \quad (5.22)$$

Seleccionando convenientemente la velocidad característica como la velocidad promedio que ingresa al dominio y el largo característico como el largo del dominio, el tiempo de

Caso	Reynolds del agua	Reynolds del petróleo
control	3.000	15.000
Oil.Re.10000	3.000	10.000
Oil.Re.20000	3.000	20.000
Oil.Re.25000	3.000	25.000
water.Re.2000	2.000	15.000
water.Re.4000	4.000	15.000
water.Re.5000	5.000	15.000
only.oil	0	15.000

Cuadro 5.3: Cuadro resumen de los casos estudiados y de los parametros que se variaron.

	$\nu \left[\frac{m^2}{s} \right]$	$\rho \left[\frac{kg}{m^3} \right]$	$Cp \left[\frac{j}{kgK} \right]$	$Pr [-]$	$\beta \left[\frac{1}{K} \right]$
Agua	$1,002 \times 10^{-6}$	1,000	4,185	5	0,05
Petroleo	3×10^{-6}	850	2,130	5	0,05

Cuadro 5.4: Parámetros físicos utilizados en la simulación.

advección cuantifica cuanto se demora el fluido en recorrer completamente el dominio. La escala temporal de la simulación puede ser adimensionalizada utilizando este parámetro de la siguiente forma:

$$t^* = \frac{t}{t_{adv}} \tag{5.23}$$

Se estableció así el horizonte de simulación temporal como $t^* = 30$, vale decir, el tiempo necesario para que el fluido recorra completamente el dominio unas treinta veces. Dicho horizonte fue fijado en base a la literatura y las restricciones en el tiempo de computo de la simulación, siendo este en promedio de unos 40 días de calculo por cada caso.

Respecto a la tensión interfasial entre las fases fluidas, no se considero su variación en función de la temperatura de la mezcla, fijandose esta en un valor constante como se observa en la tabla 5.5.

Interfaz	$\sigma [Nm]$
Agua-Petróleo	0,032

Cuadro 5.5: Valor de la Tensión interfasial para la interfaz agua-petróleo.

5.7.1. Estructura de los archivos

La estructura general de los archivos necesarios para realizar cada simulación se representa esquemáticamente en la imagen 5.32. Cada caso simulado consta de una carpeta raíz la cual contiene:

- archivo ejecutable “Allrun” el cual contiene las instrucciones para transformar el archivo de malla al formato utilizado por OpenFOAM.
- Subcarpeta “0” la cual contiene la información respecto a las condiciones de contorno e iniciales.
- Subcarpeta “Constant” la cual contiene la información de la malla, las propiedades termodinámicas de los fluidos, la información sobre los modelos de turbulencia utilizado y la información de la fuerza de gravedad.
- subcarpeta “System” la cual contiene la información sobre los parámetros de control de la simulación, los esquemas de discretización utilizados para las distintas ecuaciones tratadas y la selección de los algoritmos de resolución para los sistemas matriciales, además un conjunto de diccionarios con la información para realizar la descomposición para el trabajo en paralelo y el seteo de las condiciones iniciales de ser requerido.

Un mayor detalle de estos archivos puede ser encontrado en el anexo C

5.7.2. Hardware utilizado para la simulación y tiempos de cálculo

Las simulaciones fueron realizadas con la ayuda del Centro Científico Tecnológico de Valparaíso CCT-VAL quienes facilitaron el acceso al Cluster de cálculo el cual cuenta con las siguientes especificaciones técnicas:

- 16 máquinas con procesador 2 x Xeon X5560, 2.80GHz y 16GB de RAM, 16 x 16 núcleos
- 8 máquinas con procesador 2 x Xeon X5675, 3.07GHz y 32GB RAM, 8 x 24 núcleos
- Capacidad de almacenamiento de 40 TB

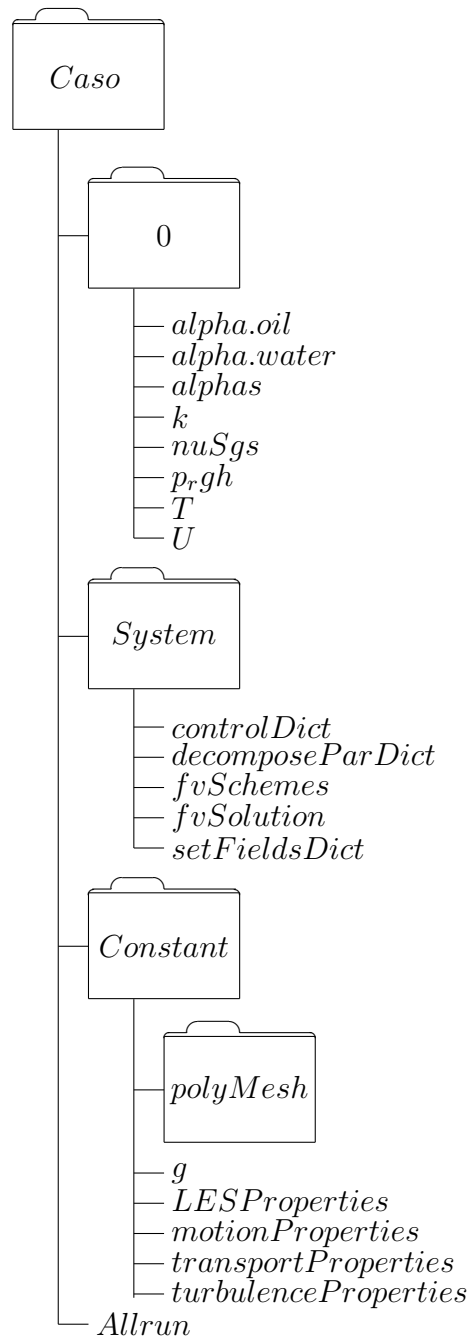


Figura 5.32: Representación esquemática de la estructura de los archivos necesarios para realizar cada simulación

- Sistema operativo Scientific Linux 6.5

El tiempo de computo necesario para la realización de las simulación con las características y el hardware explicado previamente fue de aproximadamente 40 días por cada una de las simulaciones que se realizaron, esto fue utilizando código paralelizado trabajando con 24 núcleos por simulación.

Capítulo 6

Resultados

En esta sección se presentan los resultados obtenidos de las distintas simulaciones para los casos de interés analizados y explicados en las secciones previas. Se presentan en primera instancia las simulaciones del caso de control y de la simulación sin flujo de agua correspondientes a las condiciones nominales de operación del sistema. Posteriormente se presentan y comparan los resultados cuando se varían los flujos de petróleo y agua que se inyectan en el sistema. Finalmente se presentan los resultados del análisis espectral tanto del campo de velocidad y su correspondiente espectro de energía como del campo de presiones, y el estudio de las frecuencias de oscilación principal a la salida del inyector. Los campos medios presentados corresponden al promedio temporal de los campos instantáneos utilizando una ventana temporal adimensionalizada de $t^* = 3$:

$$\langle \Phi \rangle = \frac{1}{3} \int_{t^*}^{t^*+3} \Phi dt^* \quad (6.1)$$

Los campos presentados han sido adimensionalizados utilizando las siguientes expresiones:

$$u^* = \frac{u}{u_{max}} T^* = \frac{T - T_{in}}{T_{in}} \quad (6.2)$$

donde u_{max} y T_{in} corresponden a la velocidad mayor entre las velocidades del petróleo y el agua impuestas como condición de contorno, y la temperatura a la que ingresan los fluidos respectivamente.

Del análisis de los gráficos se puede observar una tendencia hacia la simetría con respecto al plano $\theta = 45^\circ$. Sin embargo esta no se encuentra completamente desarrollada lo que implica que al horizonte de simulación realizado aun no se obtiene un estado estacionario. Sin embargo es importante destacar que el tiempo necesario para la realización de cada caso limita el horizonte temporal de la simulación.

6.1. Simulación petroleo con agua

El primer caso analizado corresponde a la simulación de control con los caudales nominales de operación. Del análisis de las líneas de flujo en base al campo de velocidades promedio al interior de la cámara de mezclado representado en la figura 6.1, se puede observar cierta simetría axial en el plano correspondiente a los 45 grados. Las líneas de flujo sobre los planos $\theta = 0^\circ$ y $\theta = 90^\circ$ tienden a abrirse hacia la pared exterior de la cámara y generar un flujo aguas arriba sobre el plano $\theta = 45^\circ$. Se puede apreciar también que las magnitudes de las velocidades desarrolladas decrecen, siendo el valor máximo de 40 metros por segundo.

Al analizar los vórtices generados mediante los criterios Q y λ_2 , se pueden distinguir ciertas estructuras coherentes importantes. Primero se distinguen dos vórtices principales alineados en la dirección axial de la cámara y de vorticidad opuesta. Dichos vórtices afectan principalmente a las líneas de corriente provenientes de la tubería de agua, generando una zona de re-circulación en la posición axial adimensional de $z^* = 0,6$ aproximadamente, los cuales coalescen aguas abajo en la zona cercana a la salida del inyector. Se puede apreciar también un vórtice de características anular ubicado en el cono de la cámara de mezcla el cual se origina producto del desprendimiento de la capa limite en esa zona y del ángulo con el que es inyectado el agua en la cámara de mezcla. En la región cercana al eje axial del inyector se pueden distinguir estructuras coherentes delgadas y alargadas con un comportamiento similar a vórtices tipo “swirl” los cuales están compuestos principalmente de petroleo. Finalmente se puede apreciar una zona de estancamiento en la pared posterior del inyector en la cual se forma un pequeño vórtice de recirculación en la esquina que forma esta pared con la pared radial del inyector. Es importante destacar además que si bien se evidencian pequeñas diferencias entre las estructuras identificadas según ambos criterios de detección de vórtices, tanto las formas

de estas como las proporciones y ubicaciones son coherentes por lo que ambos criterios entregan la misma información y debido a esto solo se presentarán en las siguientes simulaciones los gráficos del criterio Q.

Respecto al campo de temperaturas desarrollado al interior de la cámara de mezcla (imágenes 6.4 y 6.5), los fluidos tienden a incrementar su temperatura a medida que se mueven por la cámara. Se puede observar además la influencia de los vórtices descritos previamente: al generar zonas de velocidades más bajas permiten un incremento mayor en la temperatura. El incremento de la temperatura alcanza un máximo cercano al 14 % sobre el plano $\theta = 45^\circ$ y la posición radial $r^* = 0,8$.

Respecto a la composición de la mezcla y la distribución de la fracción volumétrica de petróleo representada en la figura 6.6, se puede apreciar que la composición de la mezcla corresponde principalmente a petróleo en la gran mayoría del dominio. La zona de predominancia del agua corresponde al área inmediata de inyección de la tubería de agua. Esta zona se extiende en la dirección axial, comenzando en la posición $z^* = 0,2$ hasta $z^* = 0,4$, siendo máxima en los planos interiores producto de la presencia de los vórtices axiales. Se puede apreciar además una tendencia de la fase acuosa a ubicarse en las zonas más alejadas del eje axial. Los vórtices descritos previamente generan un efecto ciclón sobre la mezcla, tendiendo a separar las fases producto de la diferencia en las densidades. La figura 6.7 muestra la distribución de la fracción volumétrica instantánea para el tiempo $t^* = 30$ de simulación. Se puede observar una distribución bastante homogénea de las fases. La fase acuosa corresponde a la fase dispersa y el petróleo a la fase continua.

Respecto a la morfología de la fase dispersa, se analizó el tamaño máximo del diámetro de las gotas que se puede estabilizar bajo los regímenes del flujo según los números adimensionales de Weber y Ohnsorge. La figura 6.8 muestra la distribución del tamaño de gotas de la fase dispersa para la simulación del caso de control. El quinto decil de dicha distribución corresponde a un tamaño de gota de 0,13673 milímetros.

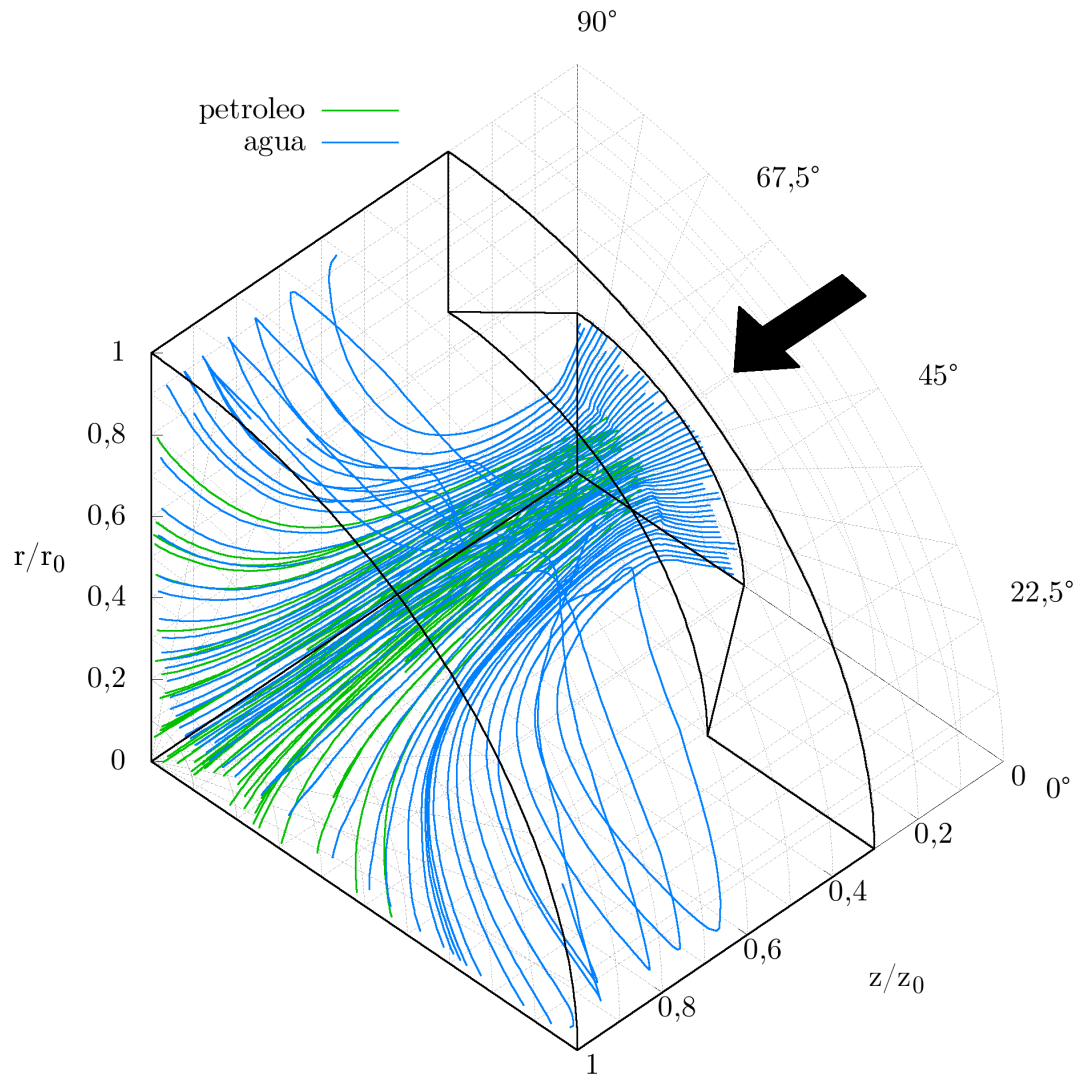


Figura 6.1: Líneas de flujo al interior de la cámara de mezclado para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$

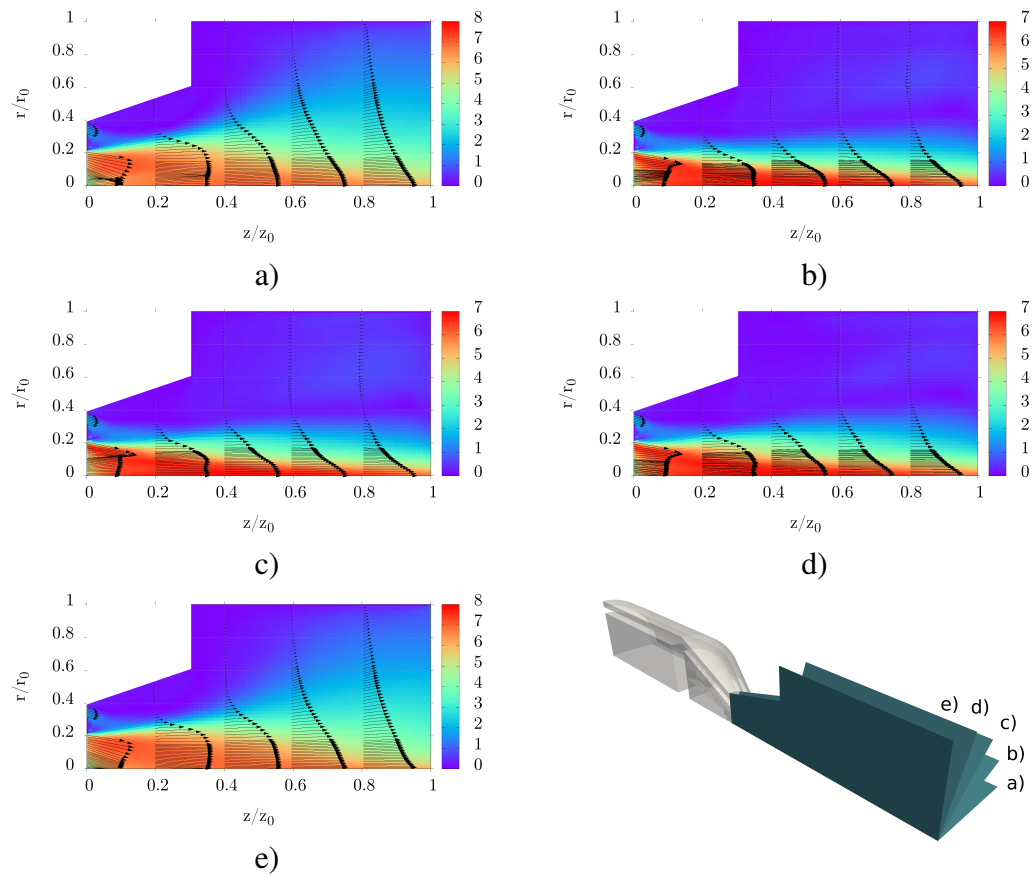


Figura 6.2: Campo de velocidades promedio para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en a) 0 grados, b) 22,5 grados, c) 45 grados, d) 67,5 grados y e) 90 grados

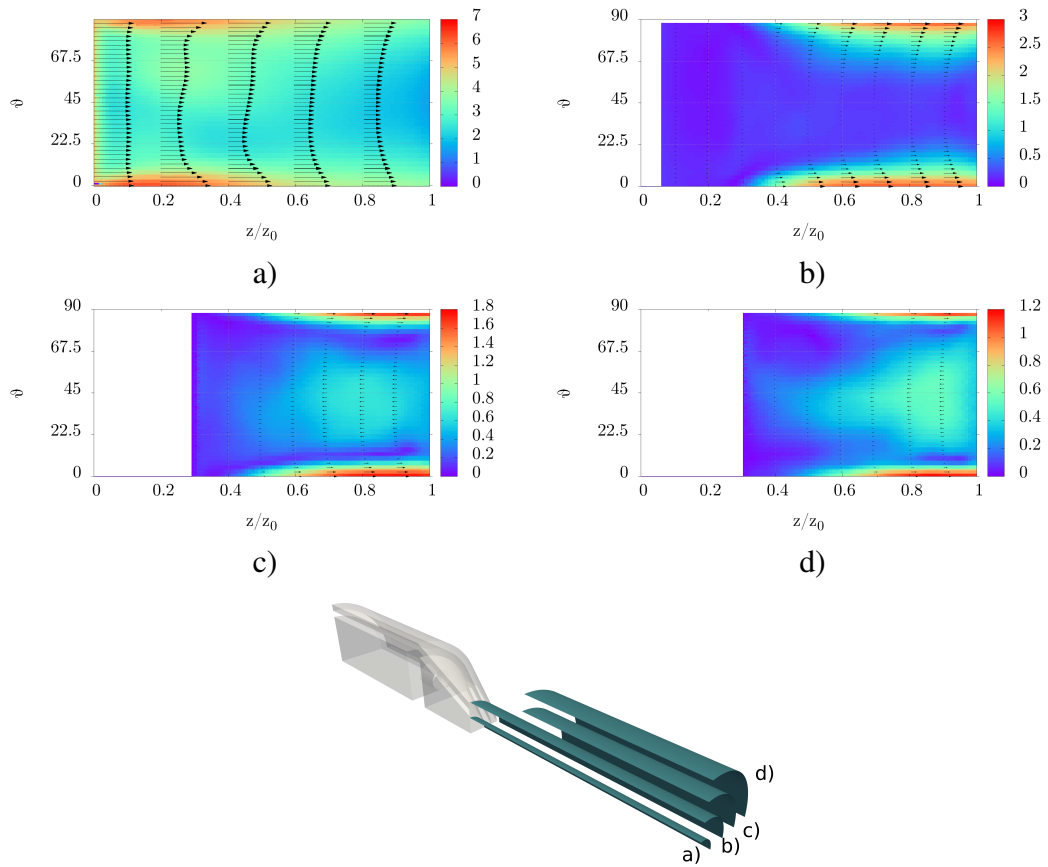


Figura 6.3: Campo de velocidades promedio para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en los radios a) 0.2, b) 0.4, c) 0.6 y d) 0.8

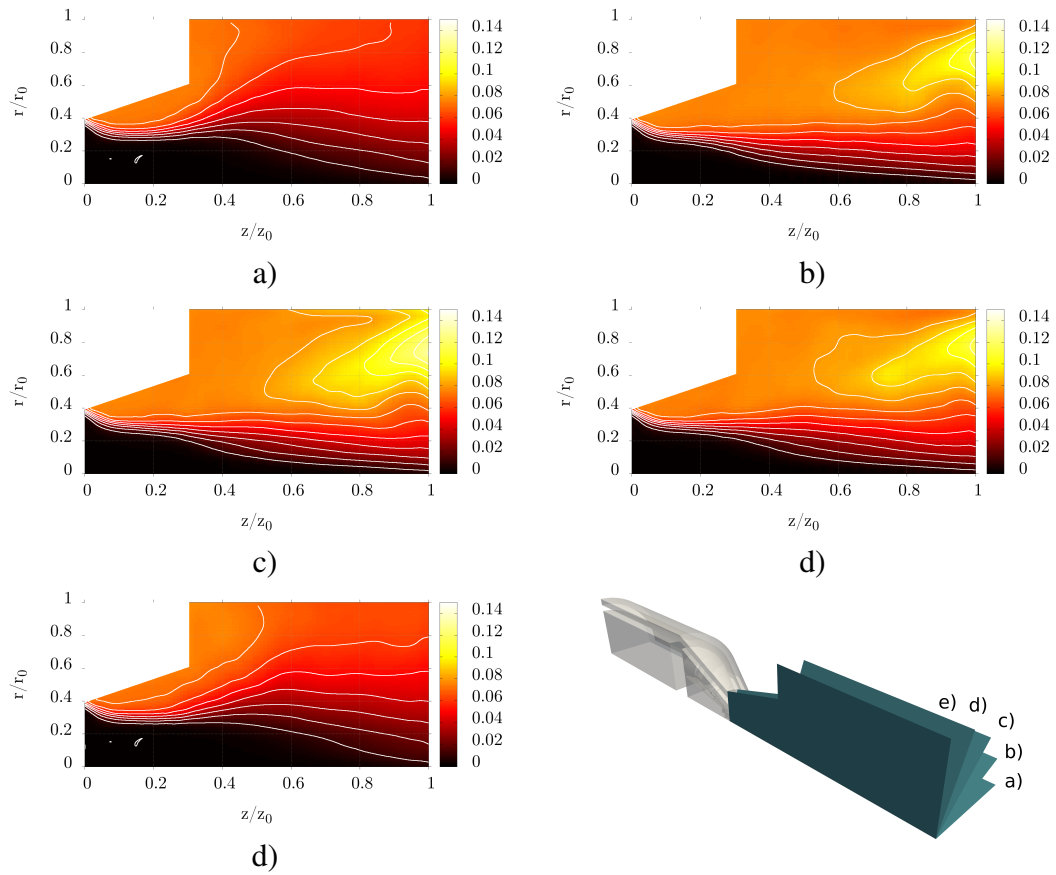


Figura 6.4: Campo de Temperaturas promedio adimensionalizado para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en a) 0 grados, b) 22,5 grados, c) 45 grados, d) 67,5 grados y e) 90 grados

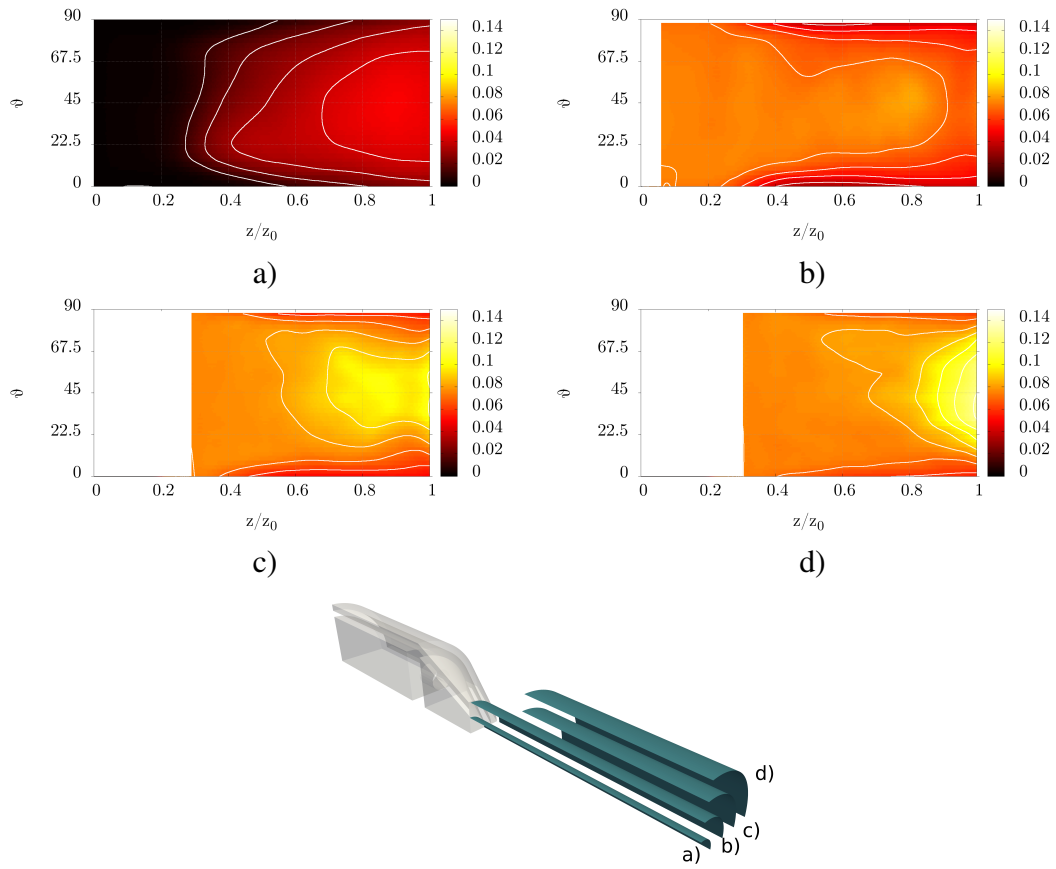


Figura 6.5: Distribución del campo de temperatura promedio para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en los radios a) 0.2, b) 0.4, c) 0.6 y d) 0.8

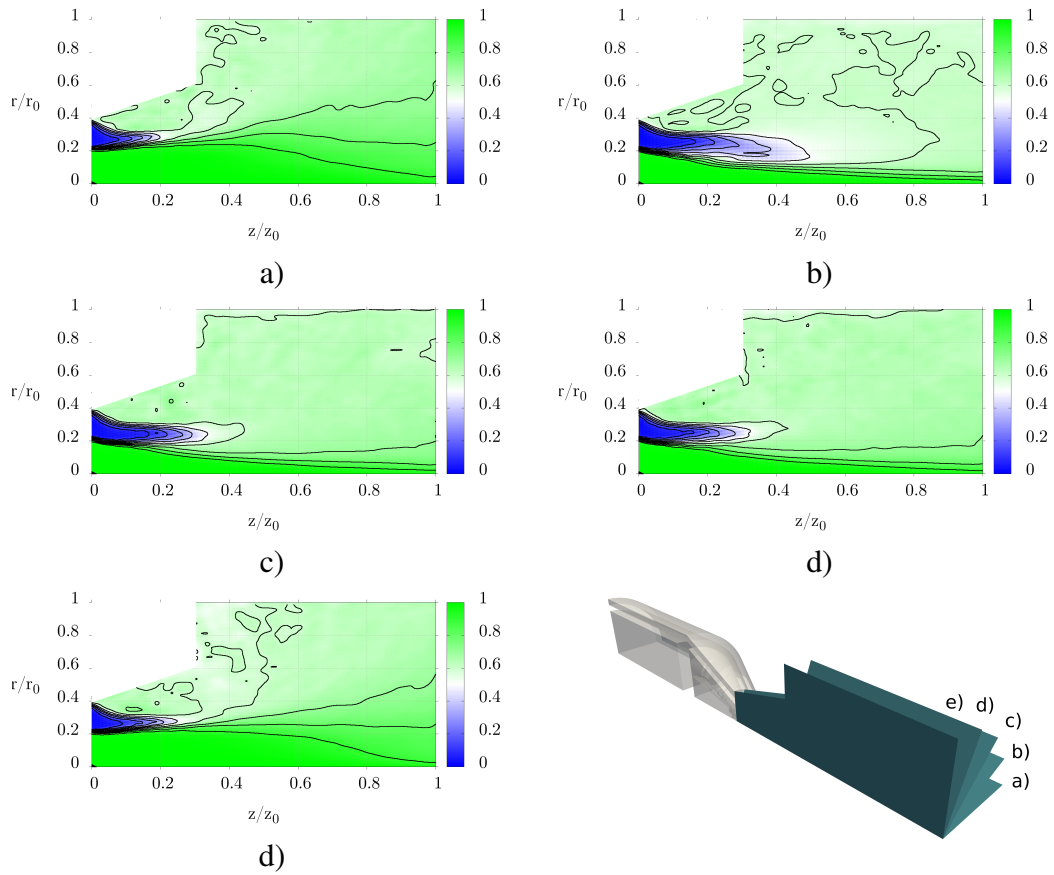


Figura 6.6: Distribución espacial promedio de la fracción volumétrica de la fase petróleo para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en a) 0 grados, b) 22,5 grados, c) 45 grados, d) 67,5 grados y e) 90 grados

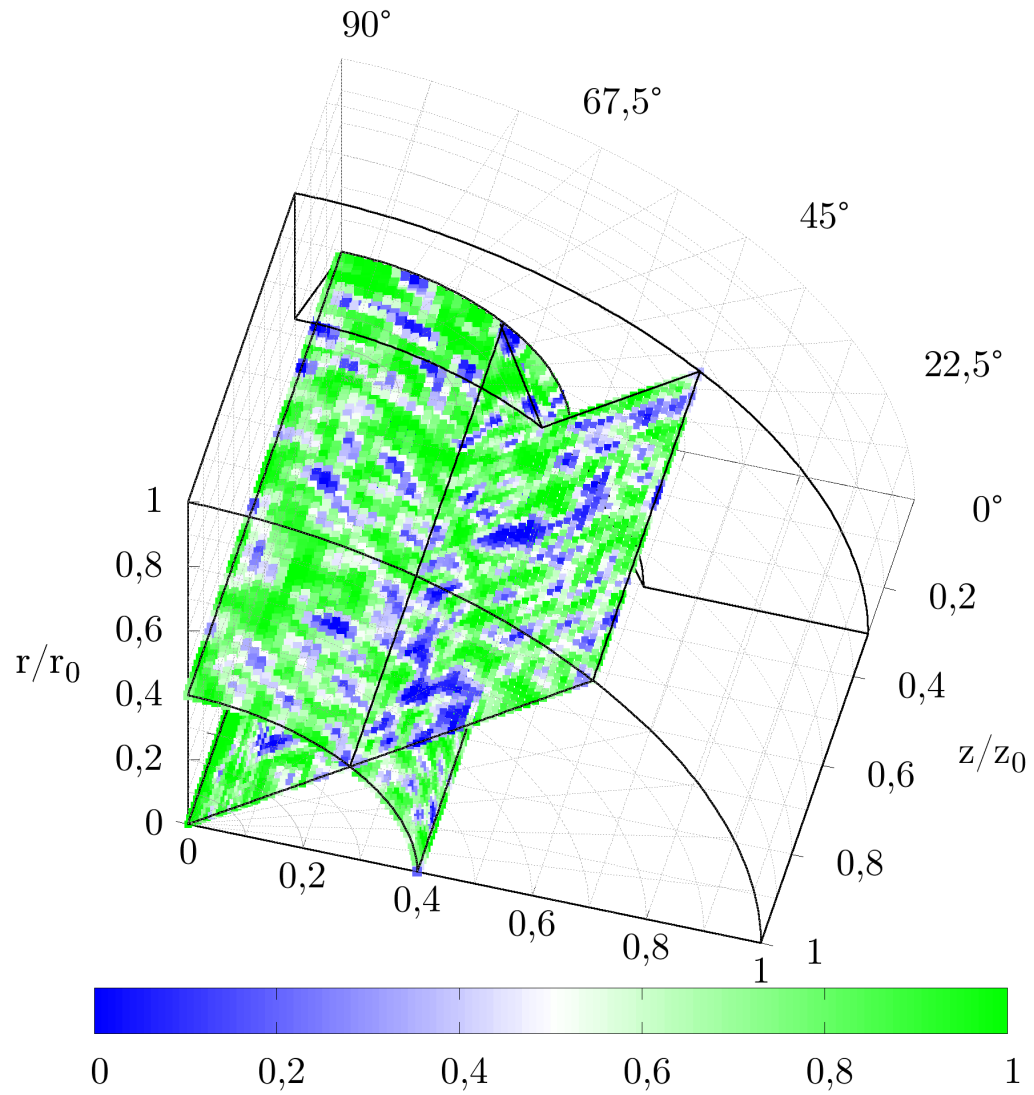


Figura 6.7: Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$

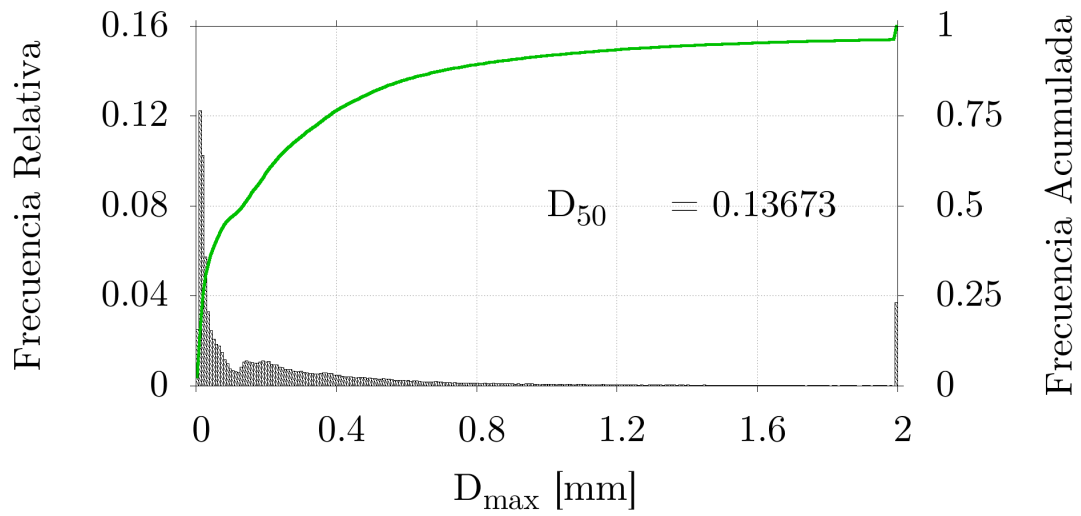


Figura 6.8: Distribución del tamaño máximo de gotas de la fase dispersa según el criterio del número de Weber crítico para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$

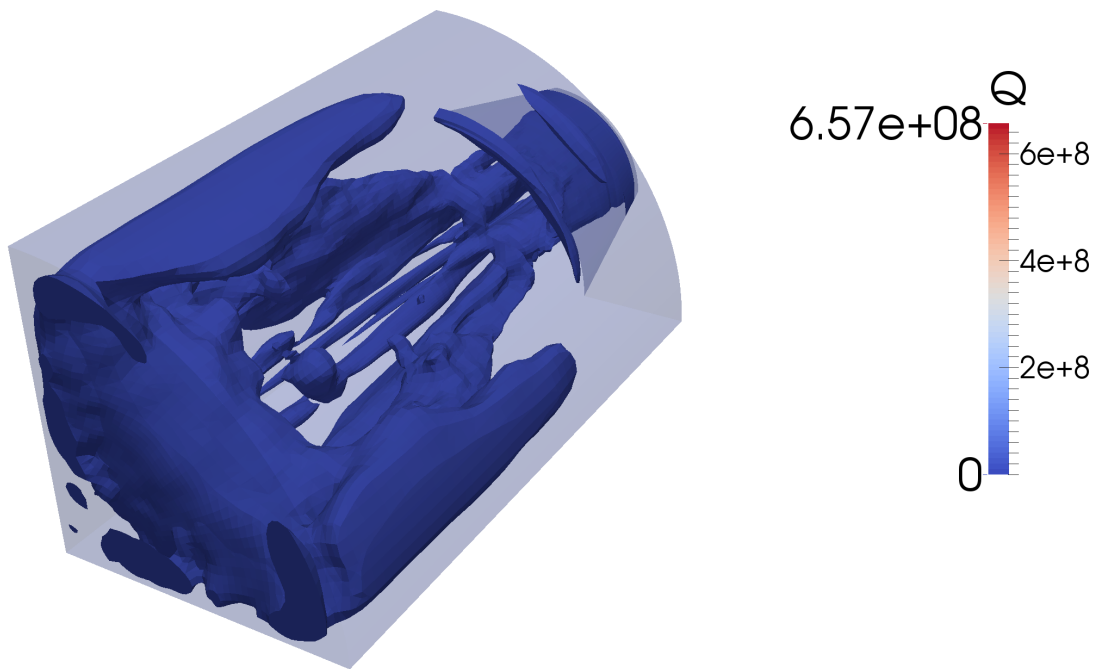


Figura 6.9: Visualización de los vórtices del campo de velocidades medio utilizando el criterio Q de detección de vórtices para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$

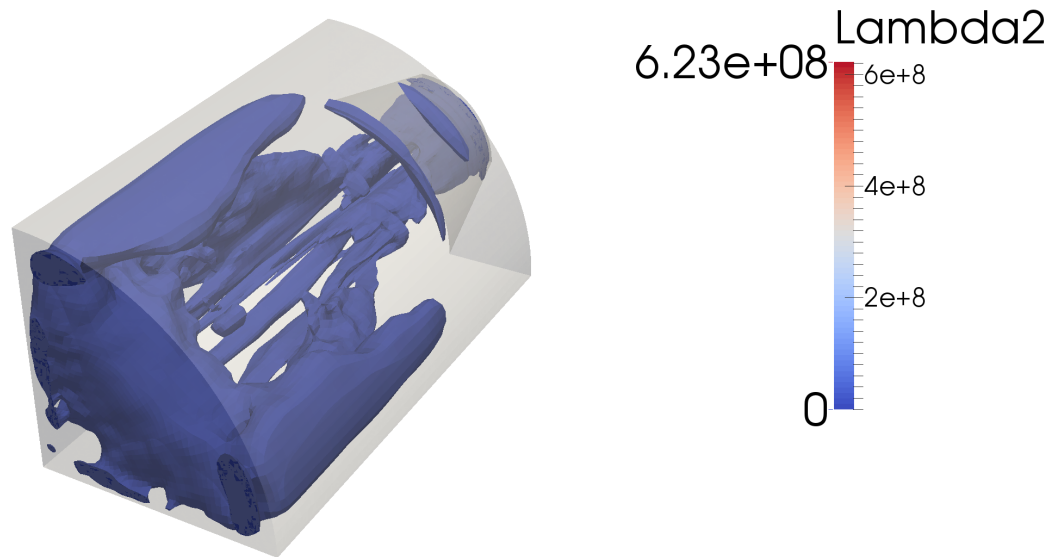


Figura 6.10: Visualización de los vórtices del campo de velocidades medio utilizando el criterio λ_2 de detección de vórtices para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$

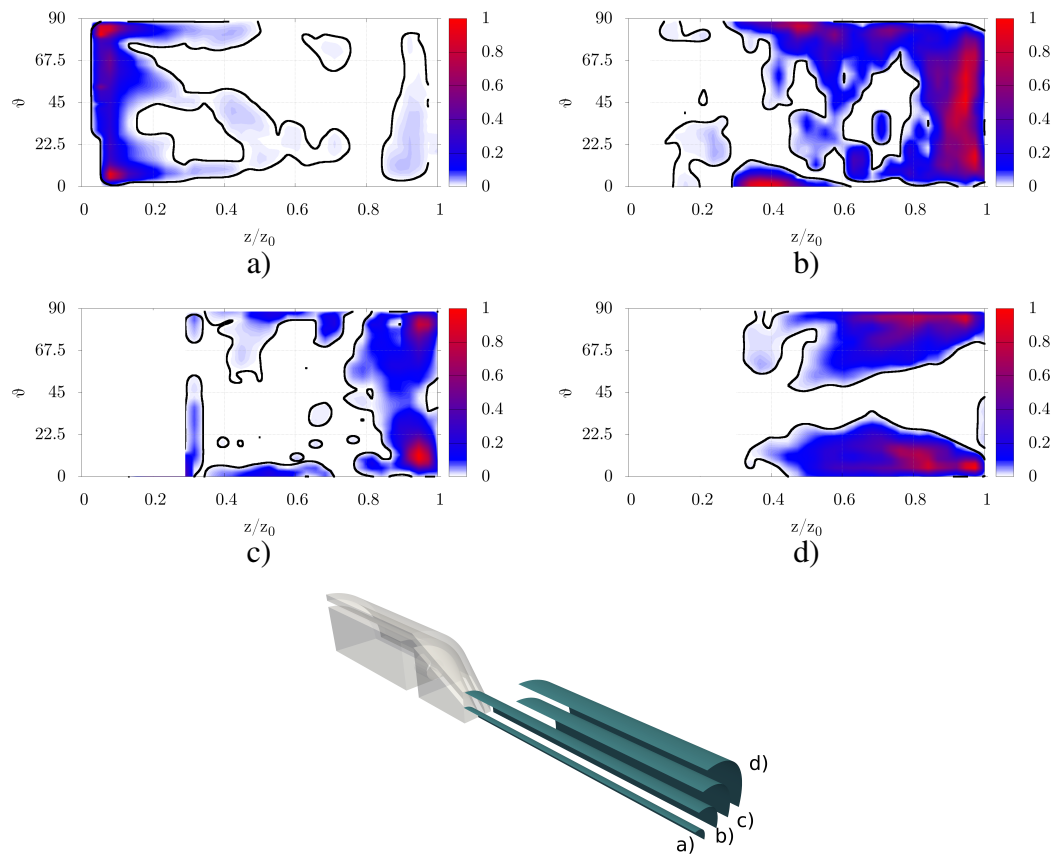


Figura 6.11: Visualización de los vórtices mediante el criterio Q para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en los radios a) 0.2, b) 0.4, c) 0.6 y d) 0.8

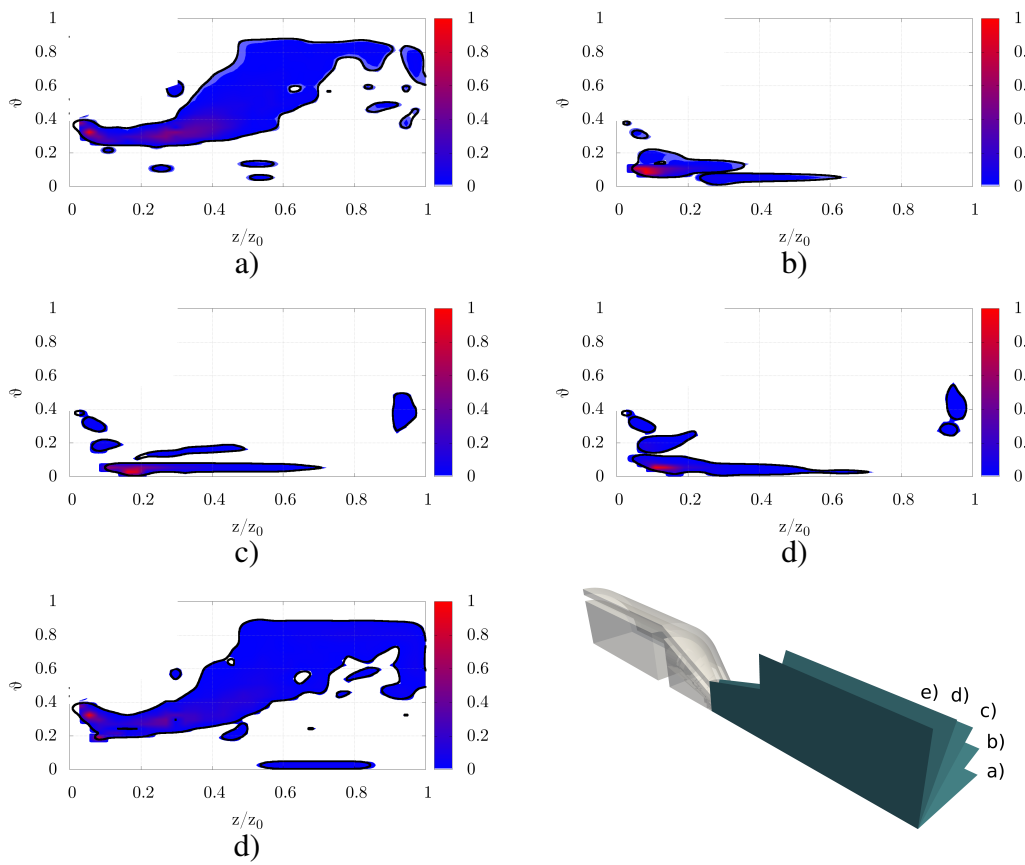


Figura 6.12: Visualización de los vórtices mediante el criterio Q para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 15000$ en cortes longitudinales en a) 0 grados, b) 22,5 grados, c) 45 grados, d) 67,5 grados y e) 90 grados

6.2. Simulación solo petróleo

El segundo caso analizado corresponde a la simulación sin flujo de agua, solo con inyección de petróleo, incrementando el caudal de este para comenzar la disminución en el caudal de agua en comparación con el caso de control. La morfología y las estructuras desarrolladas en este caso son similares a las analizadas en la simulación del caso anterior como se observa en las imágenes 6.13 y 6.14. Sin embargo no se observa el efecto de compresión radial que genera la inyección de agua sobre el flujo de petróleo. Esto comprime los vórtices de recirculación descritos en el caso previo contra la pared externa de la cámara de mezclado, disminuyendo su zona de influencia en comparación al flujo anterior. Se puede apreciar también que las magnitudes de las velocidades desarrolladas decrecen, siendo el valor máximo de 40 metros por segundo.

Respecto a la distribución de temperaturas representadas en las imágenes 6.15 y 6.16, la zona afectada térmicamente también se ve incrementada en la ausencia de agua, esto se debe a dos factores principalmente: al desarrollarse velocidades menores al interior de la cámara, el tiempo medio de residencia del fluido se incrementa, permitiendo una transferencia total de calor mayor sobre las partículas de fluido antes de abandonar el dominio y, por otro lado, las propiedades físicas del petróleo que favorecen la transferencia de calor y el incremento global de la temperatura. En este caso se logró variar la temperatura de entrada de la mezcla en un 18 % ubicándose esta sobre el plano $\theta = 45^\circ$ y la posición radial $r^* = 0,8$.

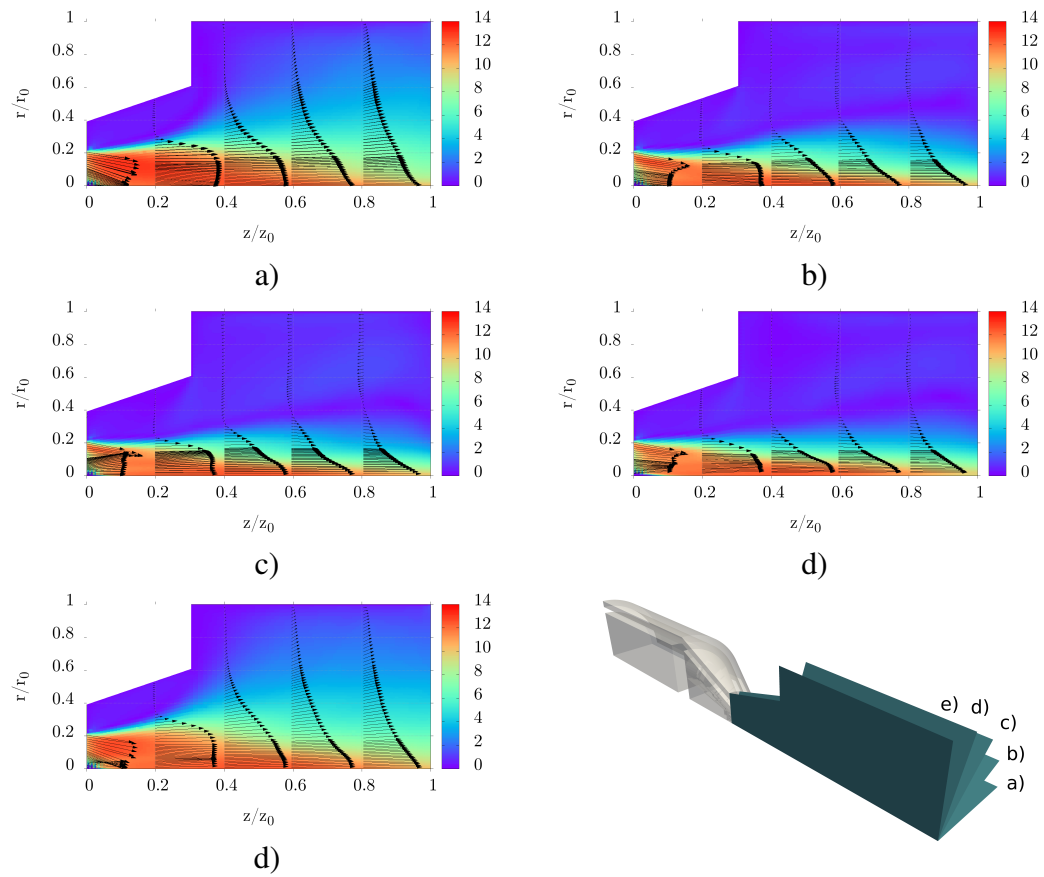


Figura 6.13: Campo de velocidades promedio para la simulación con $Re_{agua} = 0$ y $Re_{petroleo} = 15000$ en cortes longitudinales en a) 0 grados, b) 22,5 grados, c) 45 grados, d) 66,5 grados y e) 90 grados

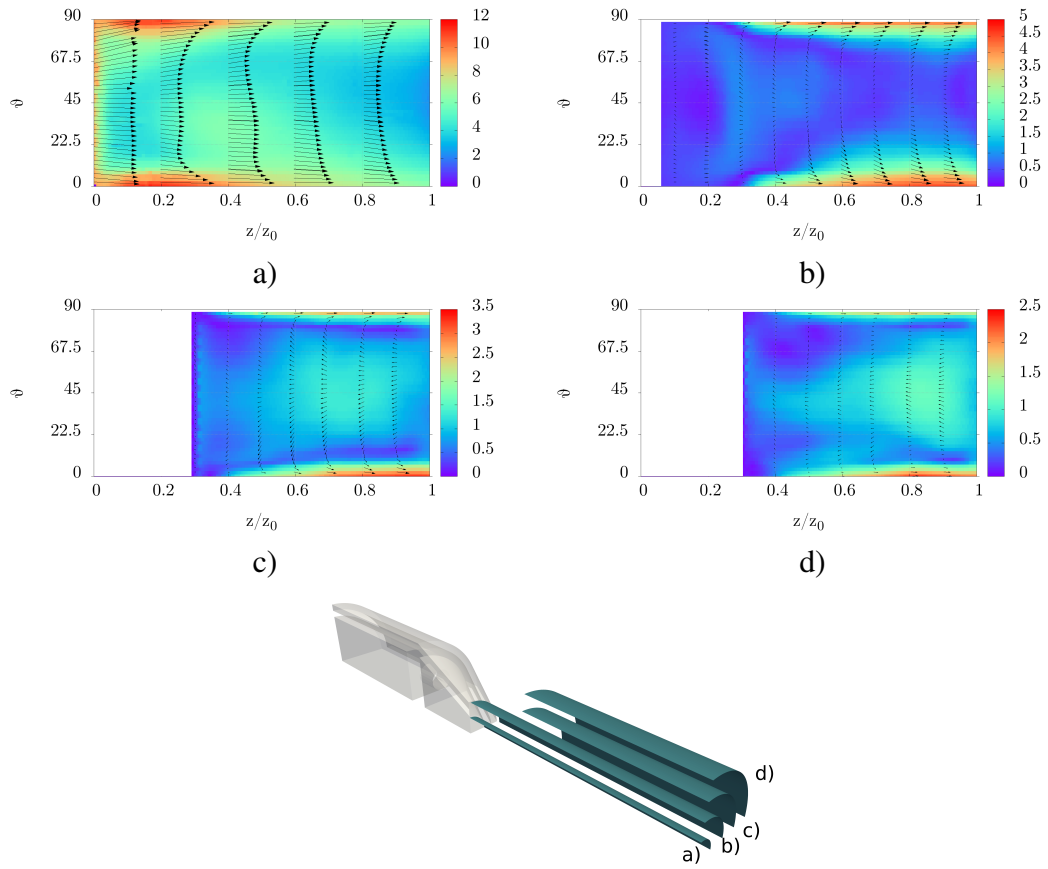


Figura 6.14: Campo de velocidades promedio para la simulación con $Re_{agua} = 0$ y $Re_{petroleo} = 15000$ en cortes longitudinales en los radios a) 0.2, b) 0.4, c) 0.6 y d) 0.8

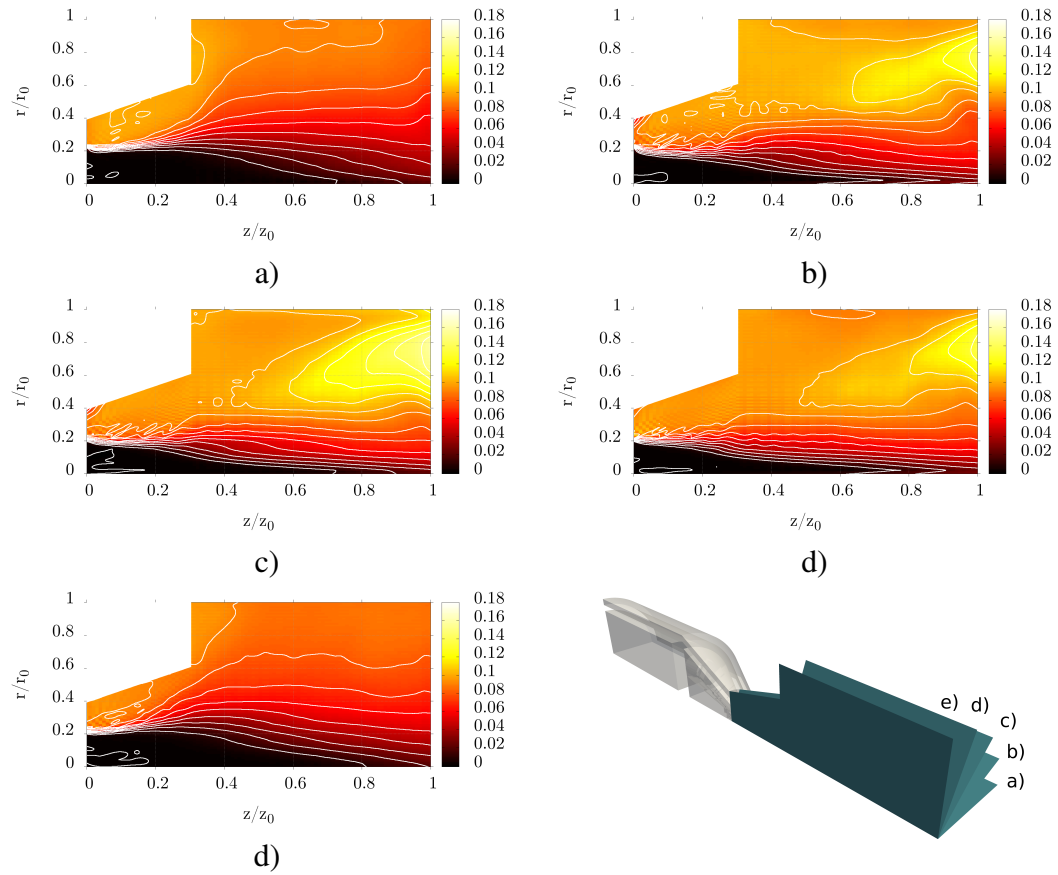


Figura 6.15: Distribución temperatura promedio adimensionalizada para la simulación con $Re_{agua} = 0$ y $Re_{petroleo} = 15000$ en cortes longitudinales en a) 0 grados, b) 22,5 grados, c) 45 grados, d) 66,5 grados y e) 90 grados

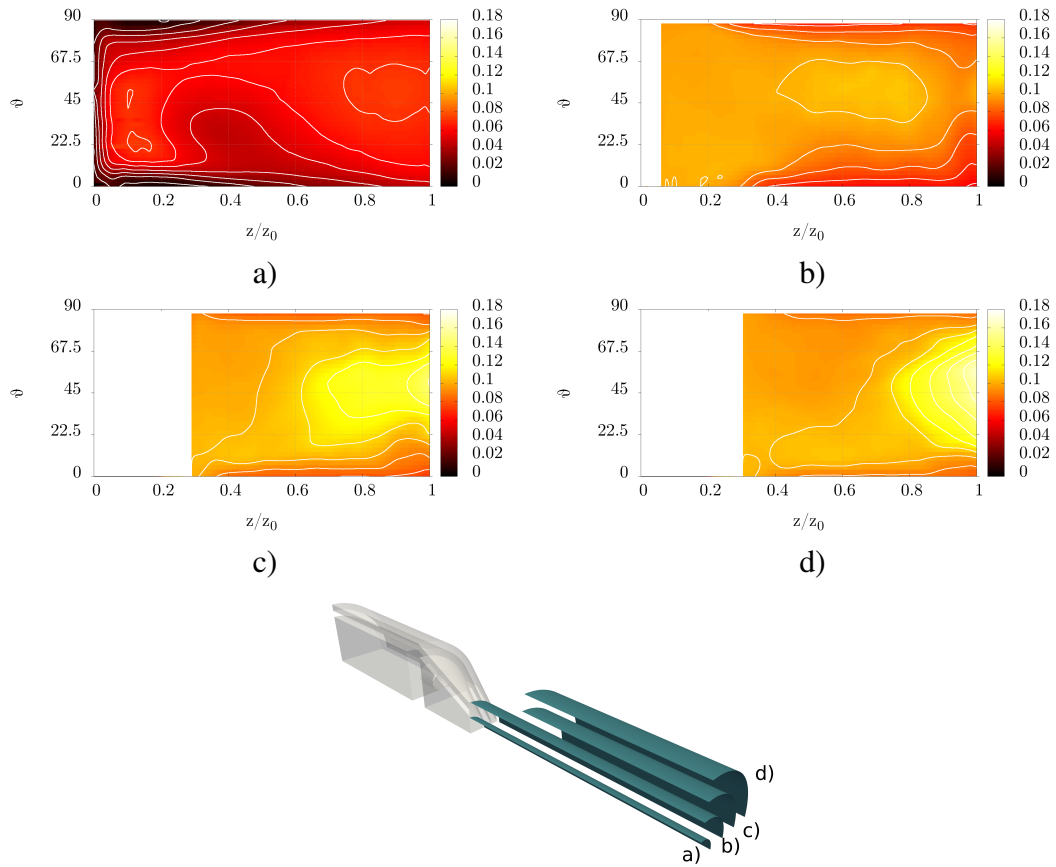


Figura 6.16: Distribución del campo de temperatura promedio adimensionalizada para la simulación con $Re_{agua} = 0$ y $Re_{petroleo} = 15000$ en cortes longitudinales en los radios a) 0.2, b) 0.4, c) 0.6 y d) 0.8

6.3. Simulaciones con variación de los caudales

6.3.1. Variación del flujo de petroleo

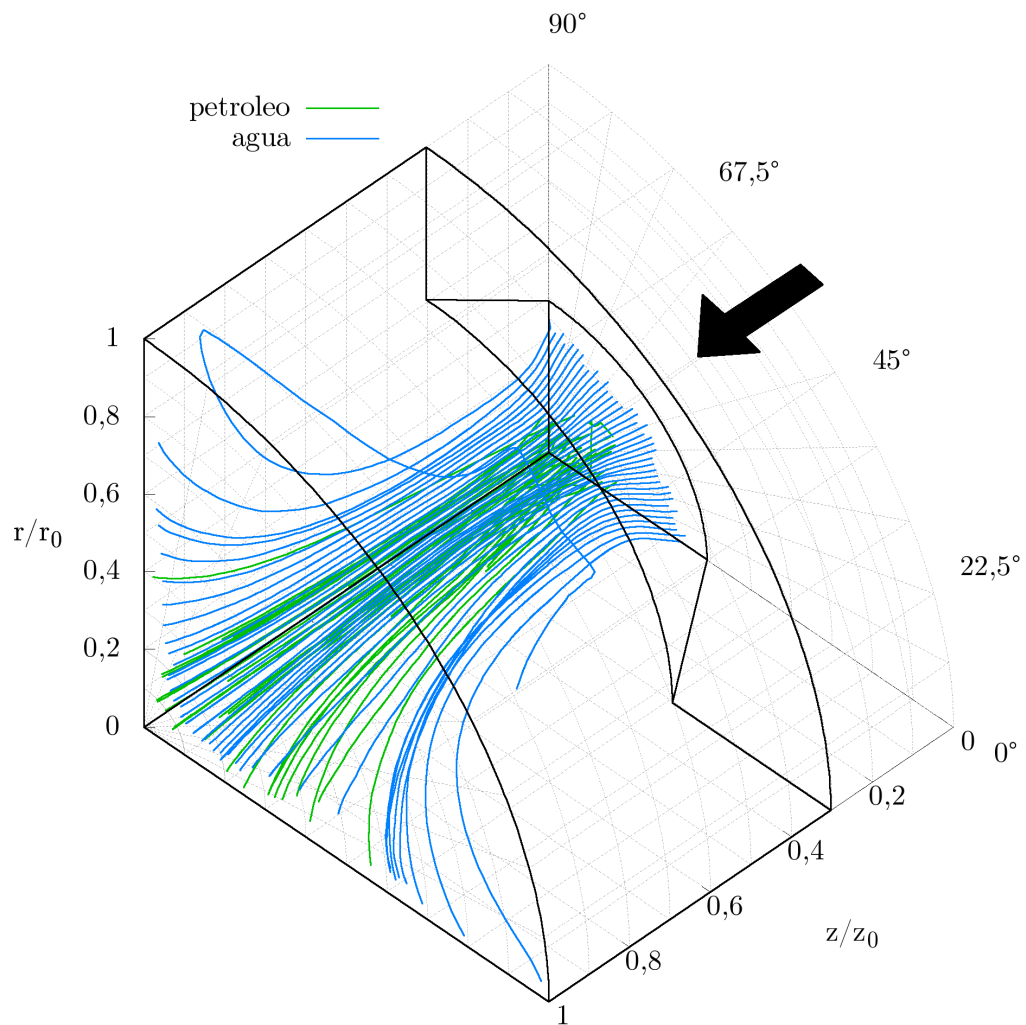


Figura 6.17: Líneas de flujo al interior de la cámara para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$

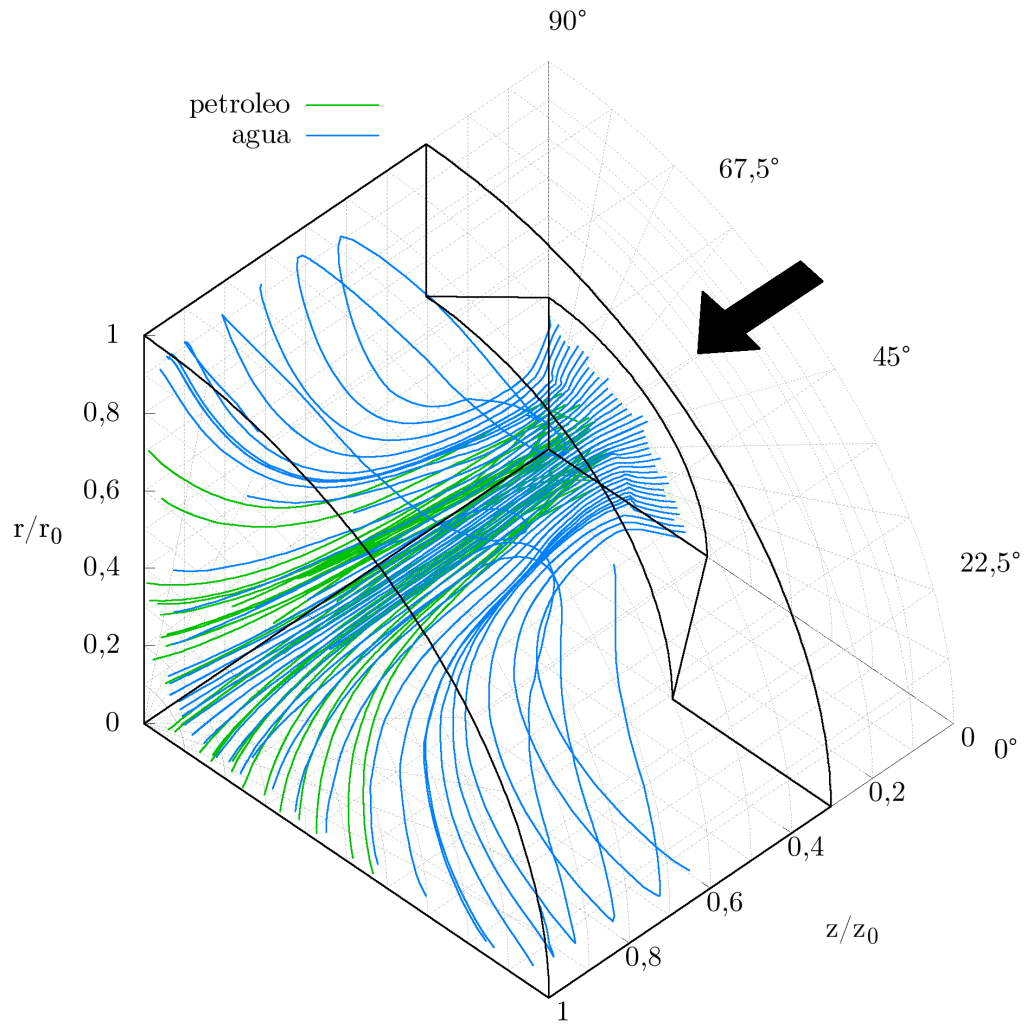


Figura 6.18: Líneas de flujo al interior de la cámara para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 20000$

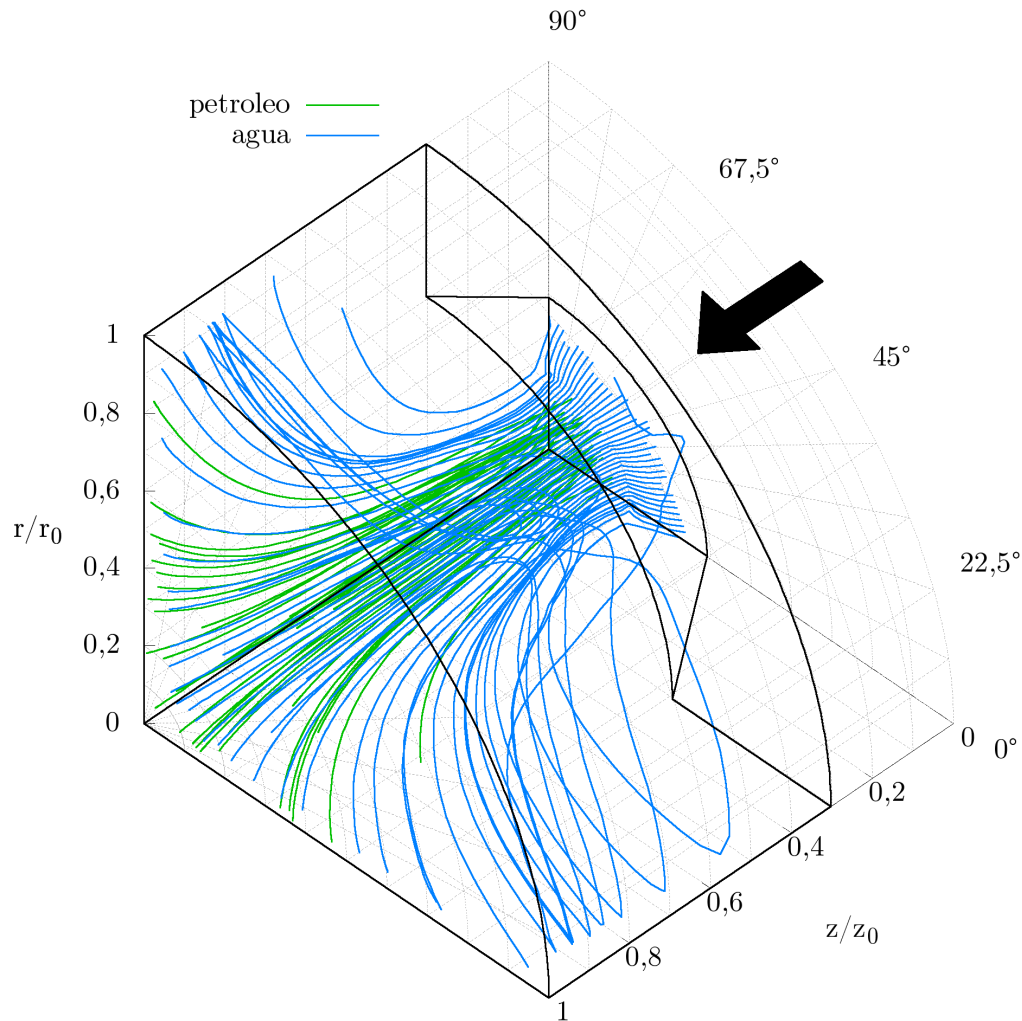


Figura 6.19: Líneas de flujo al interior de la cámara para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 25000$

Las figuras 6.17, 6.18 y 6.19 corresponden a las líneas medias de corriente para los casos en los que se varió el caudal de petróleo, dejando fijo el caudal de agua. Todas las simulaciones presentan coherencia en las estructuras desarrolladas al interior del fluido, presentando variaciones únicamente en las magnitudes involucradas, respecto a las líneas de flujo provenientes de la tubería de petróleo. Se observa un incremento gradual en la posición radial sobre la salida del inyector. Al incrementar los caudales de

petroleo, la intensidad de los vórtices y de las corrientes a contra flujo se ve aumentada, manifestándose como líneas de corrientes más compactas entre si.

Las velocidades máximas obtenidas en las simulaciones incrementan a medida que se aumenta el caudal de petroleo como era de esperarse. La posición radial de los vórtices se ve desplazada hacia la pared del inyector. Sin embargo el ángulo que abarcan estos vórtices se ve levemente aumentado como se puede apreciar de los gráficos 6.23, 6.24, 6.25 y 6.26. Los vórtices tipo “swirl” que se generan en la zona cercana al eje axial del inyector tienden a retraerse a medida que se ve incrementado el caudal de petroleo mientras que los vórtices axiales de recirculación se desplazan aguas arriba en la cámara y se comprimen aún más en la zona cercana a los planos de simetría $\theta = 0$ y $\theta = 90$, como se puede apreciar en las figuras 6.27 y 6.28. Al analizar la estructura de los vórtices mediante el criterio Q y Lambda 2 sobre los planos axiales se observa en primera instancia una retracción de los vórtices en los planos externos $\theta = 0$ y $\theta = 90$, además de un incremento en la contribución del flujo de petroleo en estos, pasando de ser vórtices compuestos principalmente de agua a tener una composición mixta como se puede apreciar en la imagen 6.29.

La distribución de temperaturas también se ve afectada cuando se varían los caudales del petróleo. Las temperaturas promedio del fluido decrece a medida que se incrementa la inyección debido a la disminución del tiempo medio de residencia. La zona de influencia térmica se desplaza en dirección radial acercándose al eje axial de la cámara lo que puede apreciarse en el perfil de temperaturas en la posición radial $r^* = 0,2$ representada en la figura 6.34.

La composición de la mezcla se mantiene homogénea para los caudales estudiados. En todos los casos se nota una tendencia de la fase acuosa a posicionarse sobre la pared de la cámara producto del efecto ciclón generado por los vórtices aunque la penetración de la fase acuosa se ve disminuida al incrementar el caudal de petróleo. El fenómeno de mezcla de las fases se ve potenciado al incrementar el flujo de petroleo. Para todos los caudales estudiados, se observa la predominancia del petroleo sobre el agua es decir una mezcla de agua dispersa en petroleo como se puede apreciar en las figuras 6.41, 6.42 y 6.43. La distribución de la fracción volumétrica sobre el dominio se presenta en los histogramas de la figura 6.44. Para todos los casos simulados, la fracción volumétrica promedio es mayor que 0,5, lo cual indicaría una distribución igualitaria entre ambas fases.

En cuanto al tamaño máximo de gotas de la fase dispersa representado en la figura 6.45 se aprecia una coherencia en el tipo de distribución que se presenta en todas las simulaciones y un descenso en el valor del quinto decil de dichas distribuciones a medida que se incrementa el caudal de petróleo, variando desde un diámetro de gota de 0,183628 a 0,0898477 milímetros.

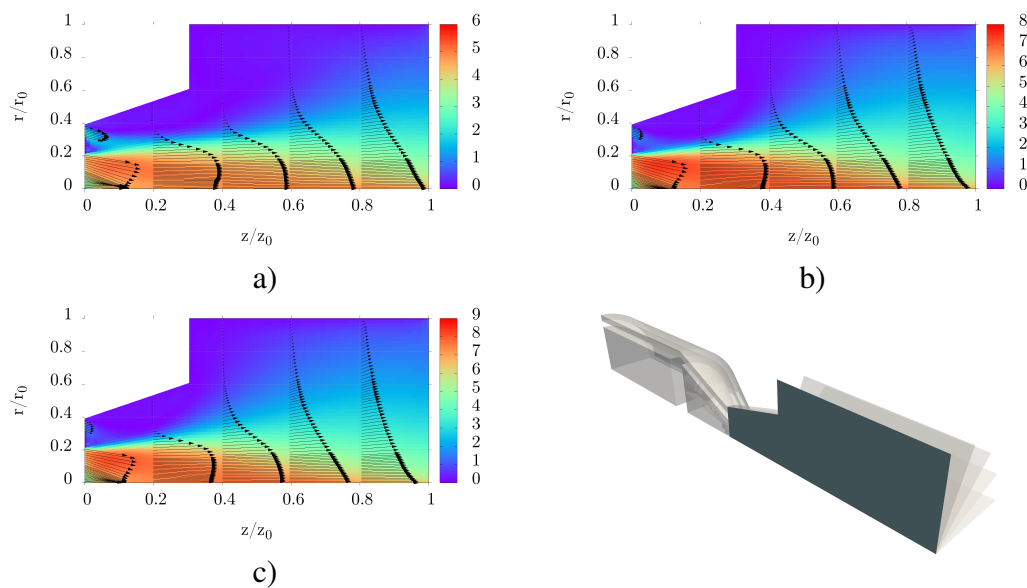


Figura 6.20: Campo de velocidad promedio en corte longitudinal a 0 grados para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$.

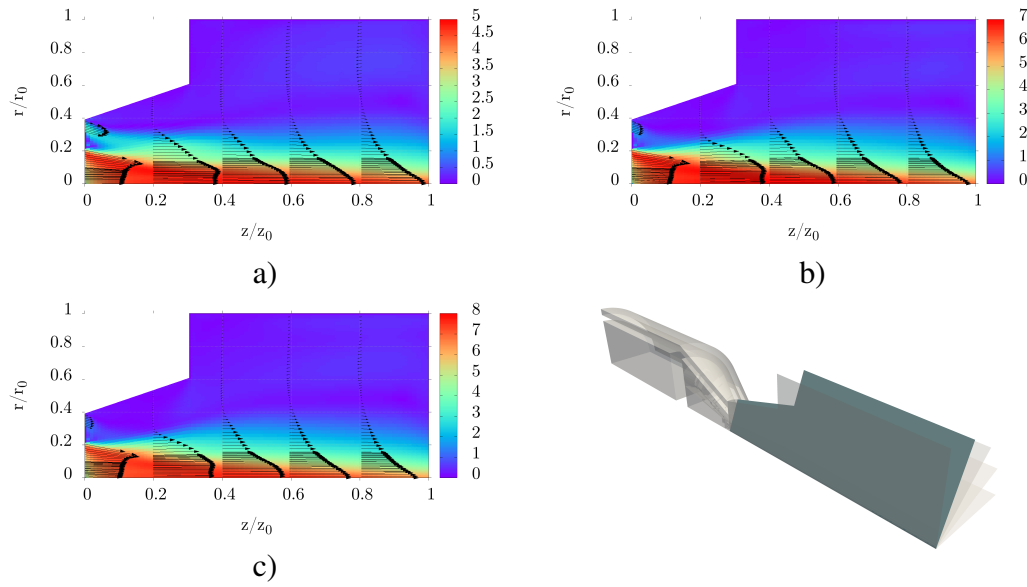


Figura 6.21: Campo de velocidad promedio en corte longitudinal a 22,5 grados para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$.

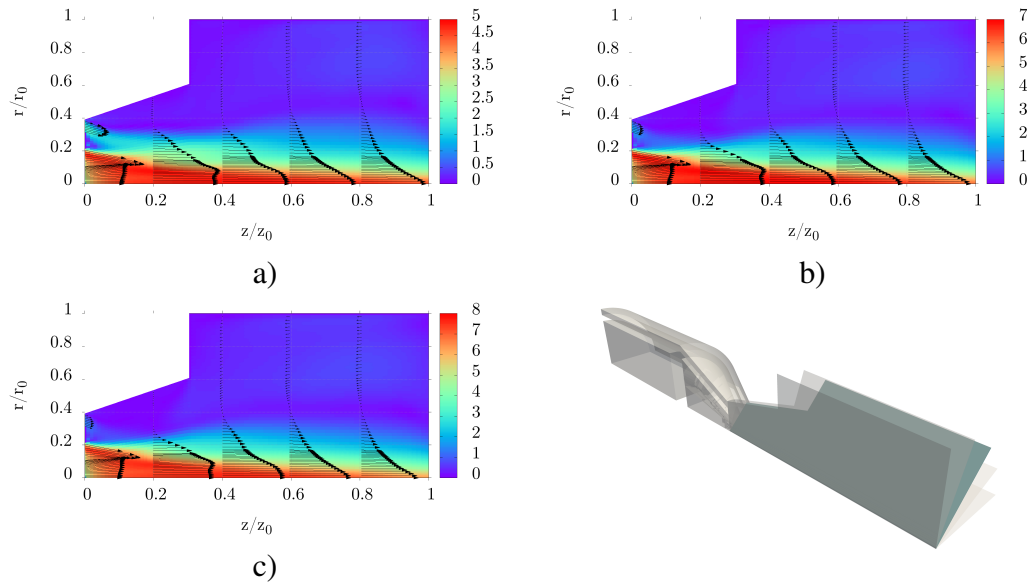


Figura 6.22: Campo de velocidad promedio en corte longitudinal a 45 grados para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$.

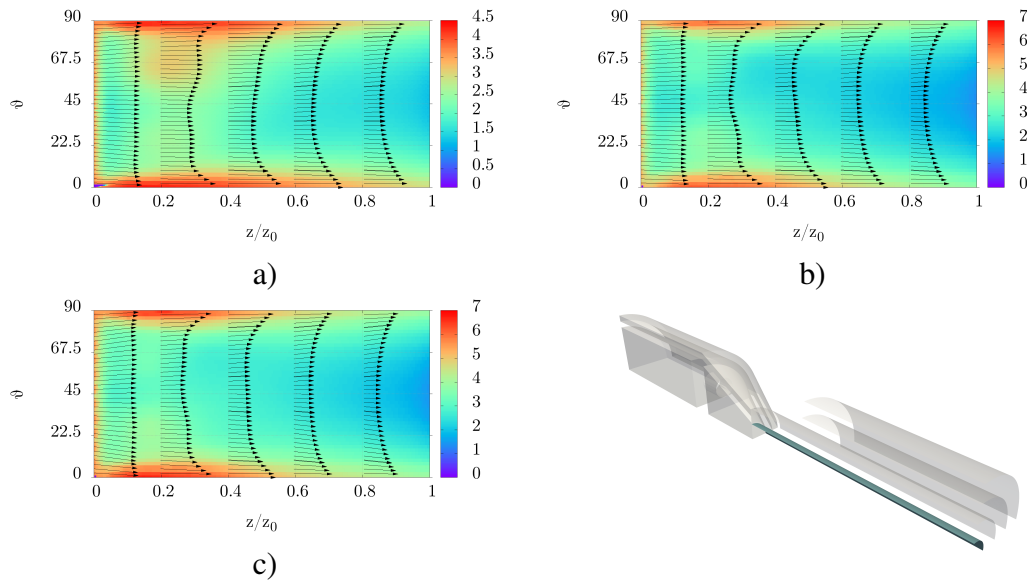


Figura 6.23: Campo de velocidad promedio en corte longitudinal sobre el radio 0.2 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$.

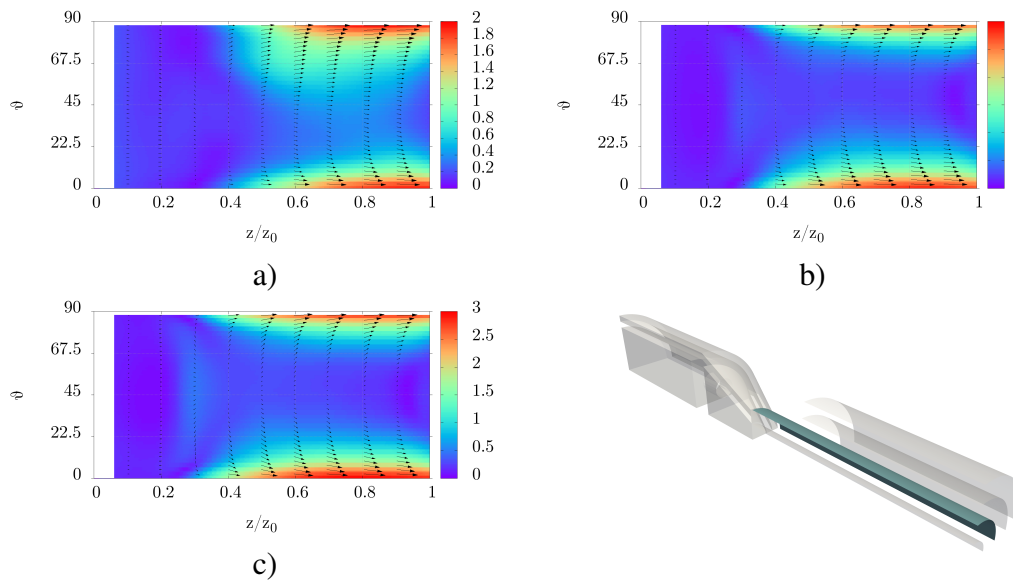


Figura 6.24: Campo de velocidad promedio en corte longitudinal sobre el radio 0.4 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$.

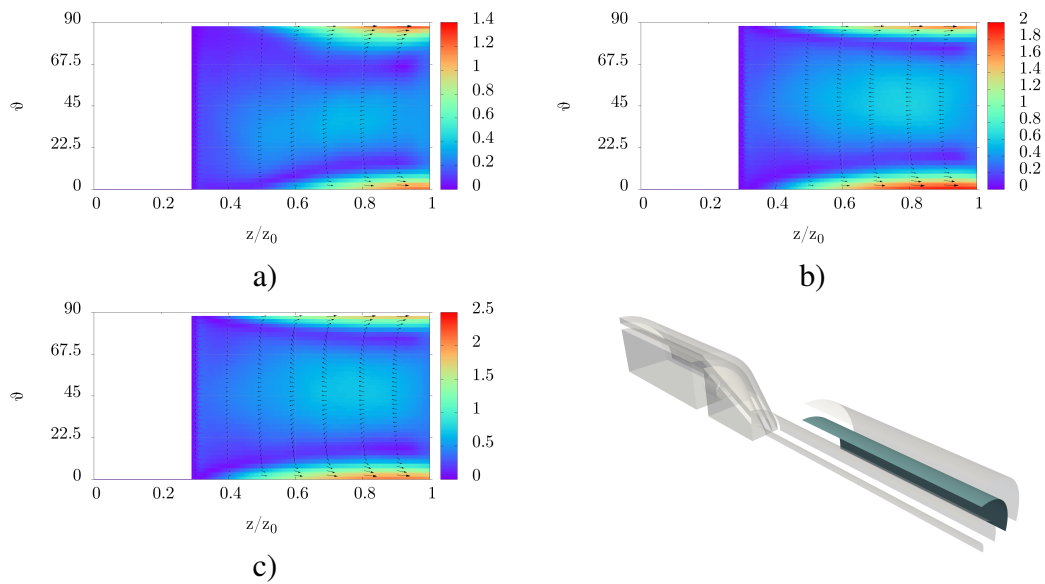


Figura 6.25: Campo de velocidad promedio en corte longitudinal sobre el radio 0.6 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$.

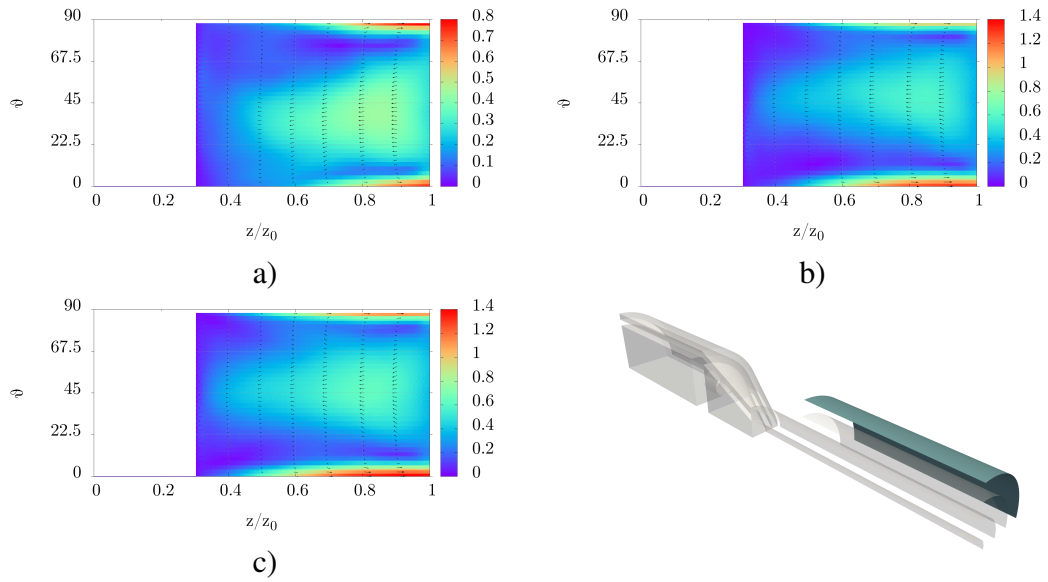


Figura 6.26: Campo de velocidad promedio en corte longitudinal sobre el radio 0.8 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$.

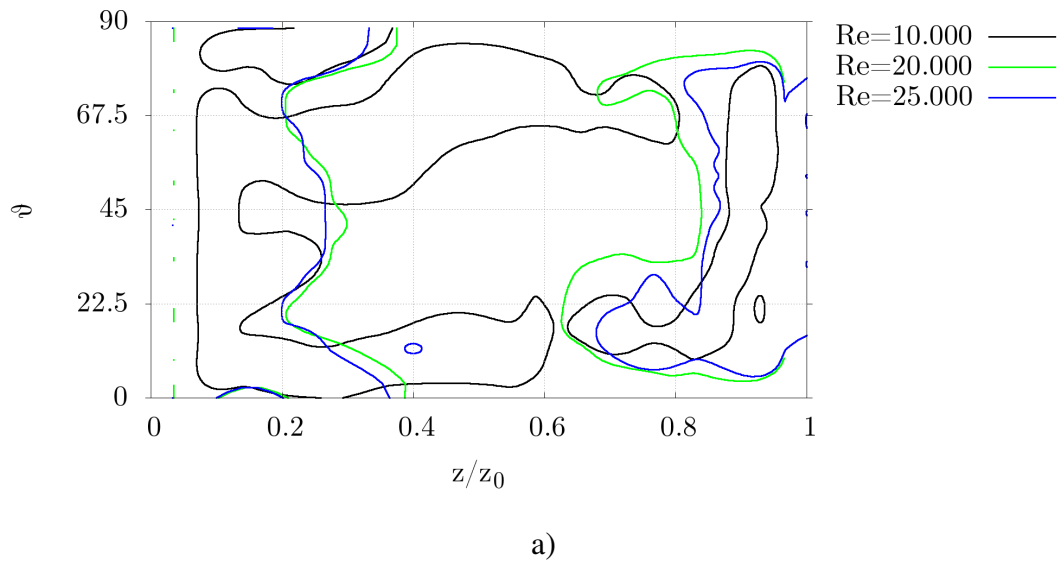
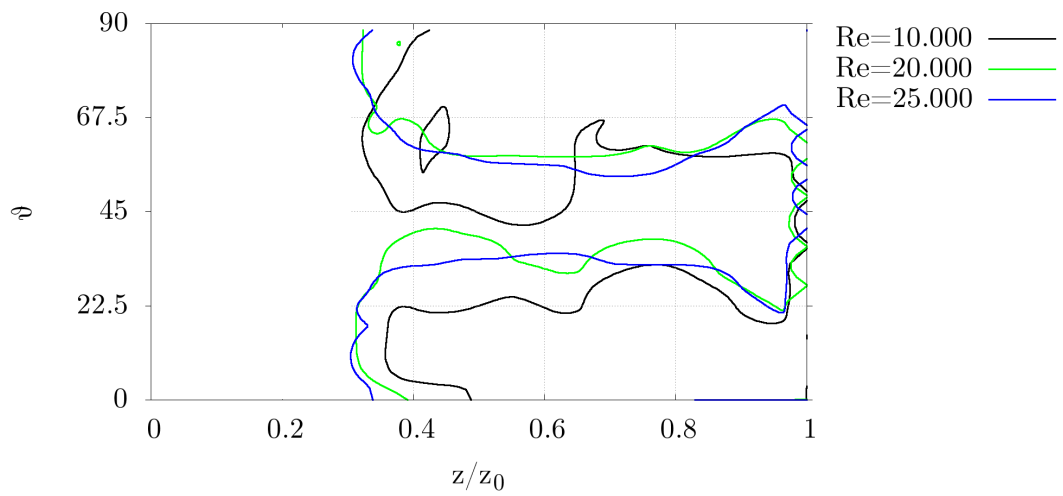
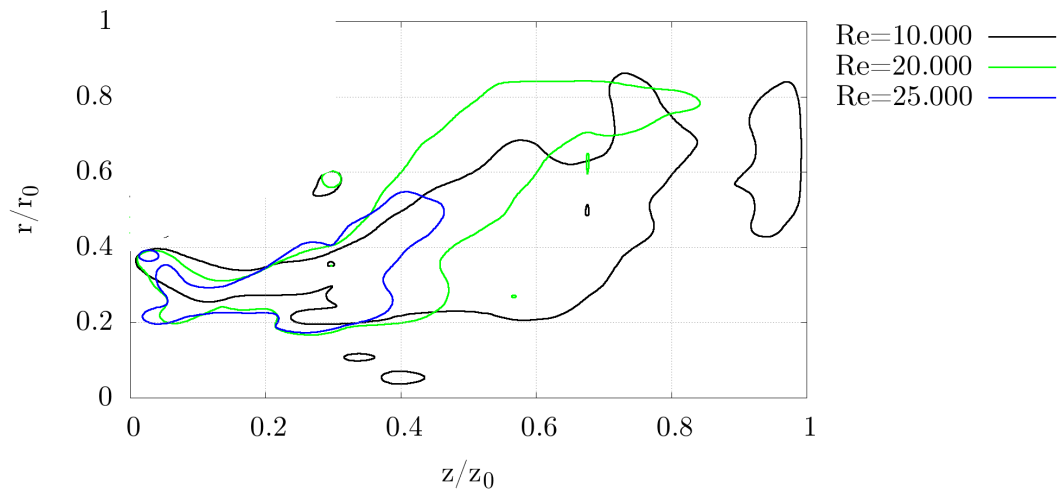


Figura 6.27: Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de petroleo sobre el plano $r = 0,2$



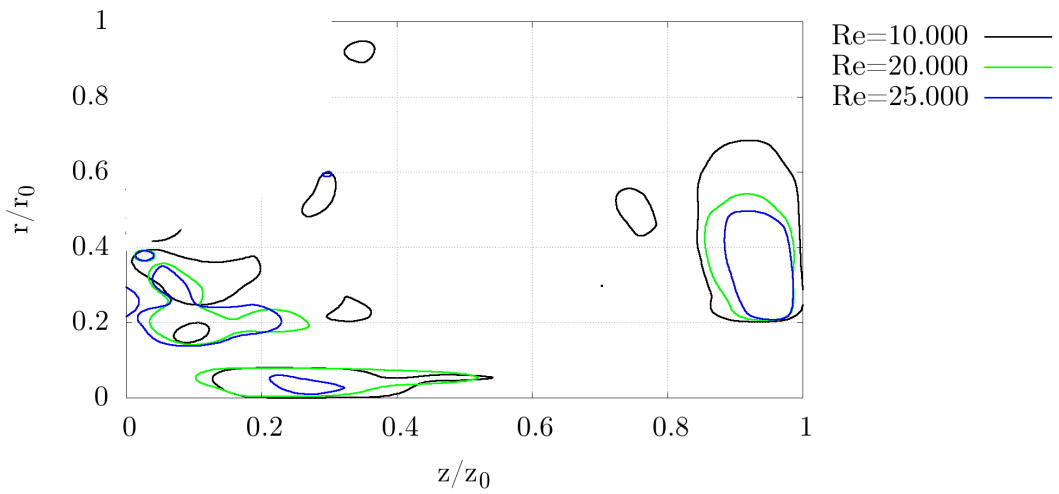
a)

Figura 6.28: Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de petróleo sobre el plano $r = 0,8$



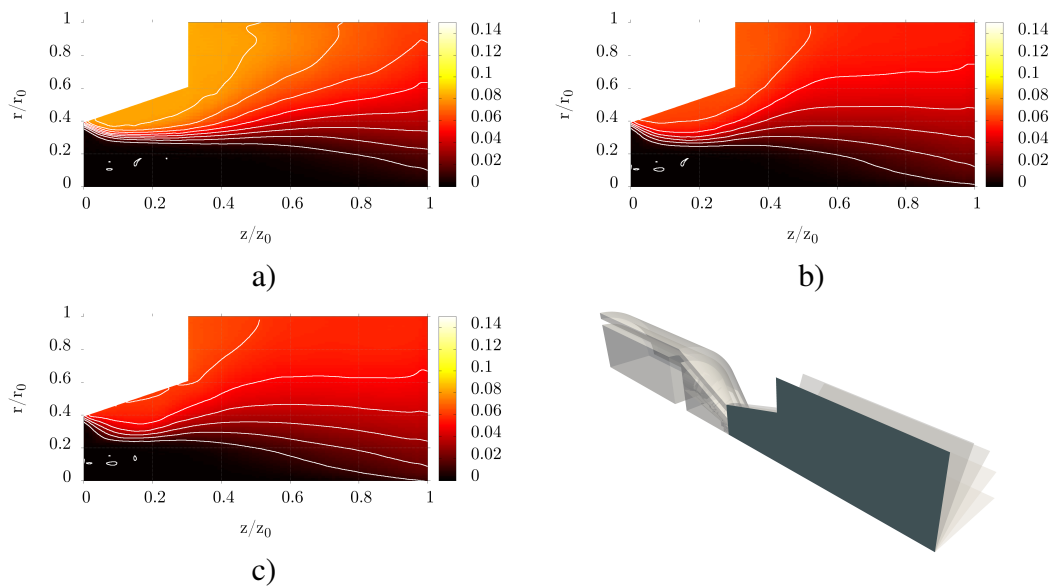
a)

Figura 6.29: Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de petróleo sobre el plano $\theta = 0$



a)

Figura 6.30: Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de petróleo sobre el plano $\theta = 45$



a)

b)

c)

Figura 6.31: Campo de Temperaturas promedio adimensionalizada en corte longitudinal a 0 grados para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$, c) $Re_{petroleo} = 25000$

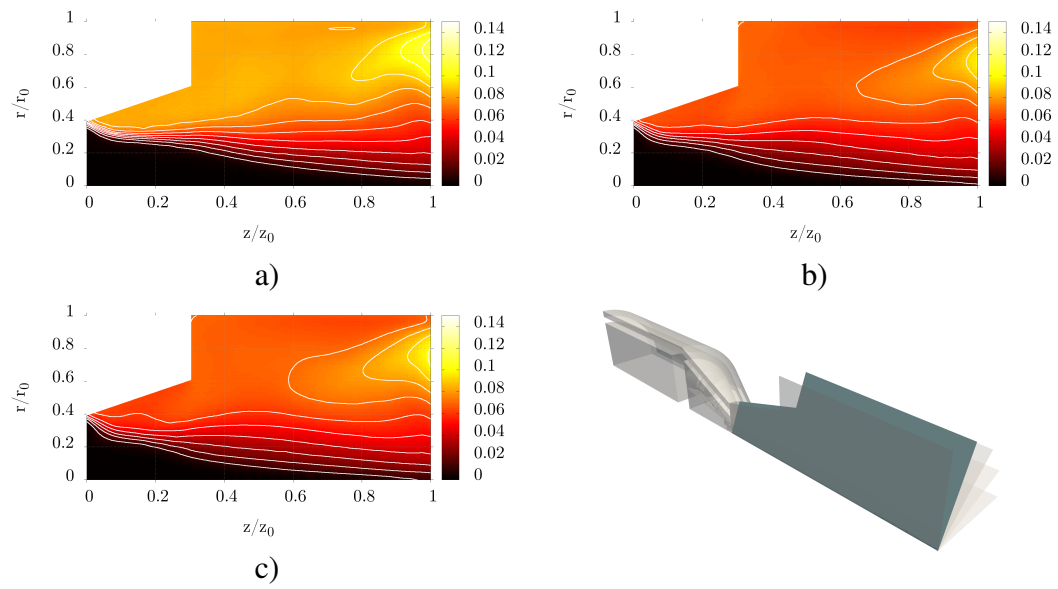


Figura 6.32: Campo de Temperaturas promedio adimensionalizada en corte longitudinal a 22,5 grados para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$, c) $Re_{petroleo} = 25000$

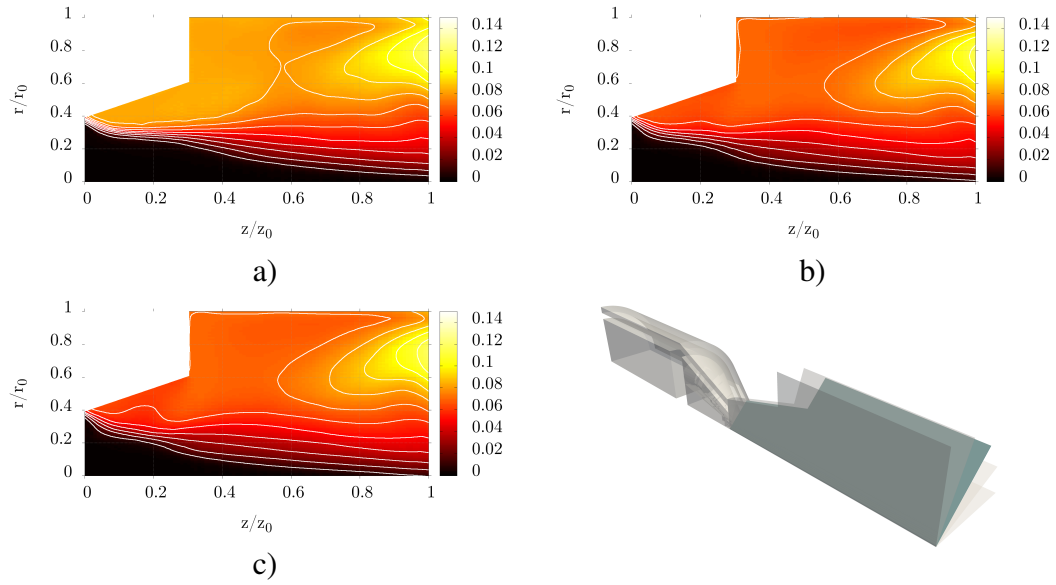


Figura 6.33: Campo de Temperaturas promedio adimensionalizada en corte longitudinal a 45 grados para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$, c) $Re_{petroleo} = 25000$

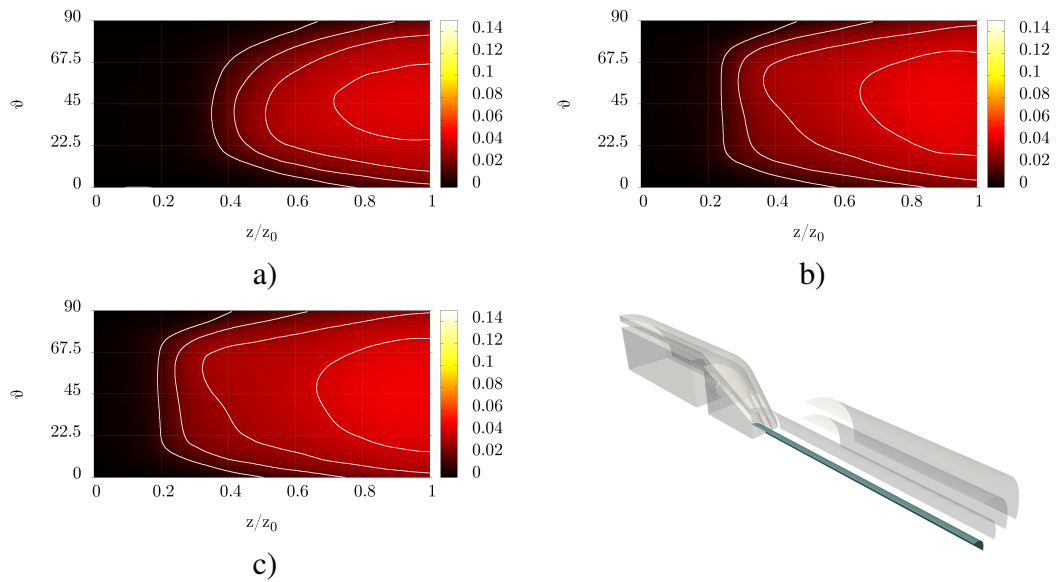


Figura 6.34: Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.2 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$

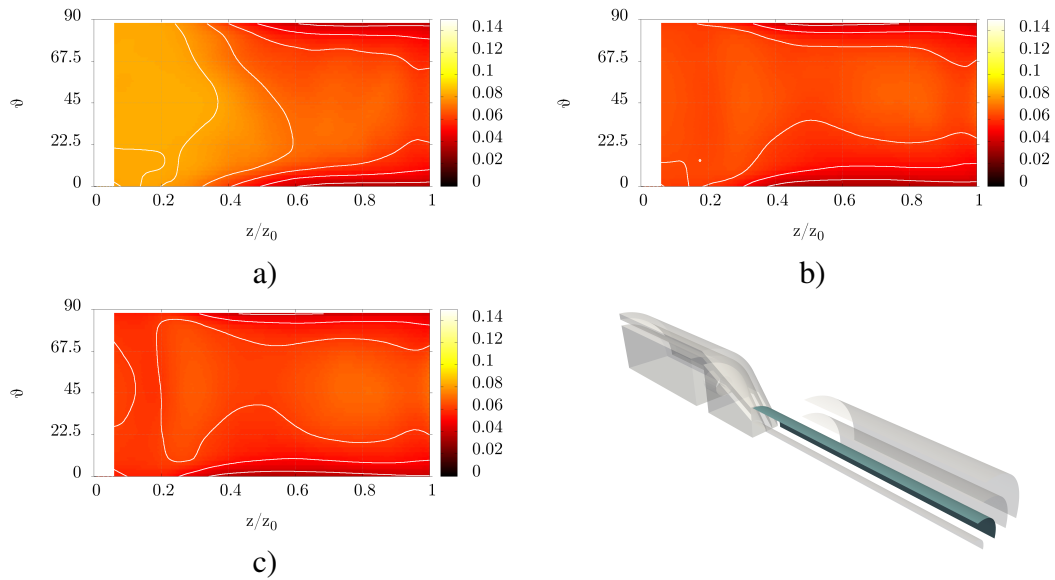


Figura 6.35: Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.4 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$

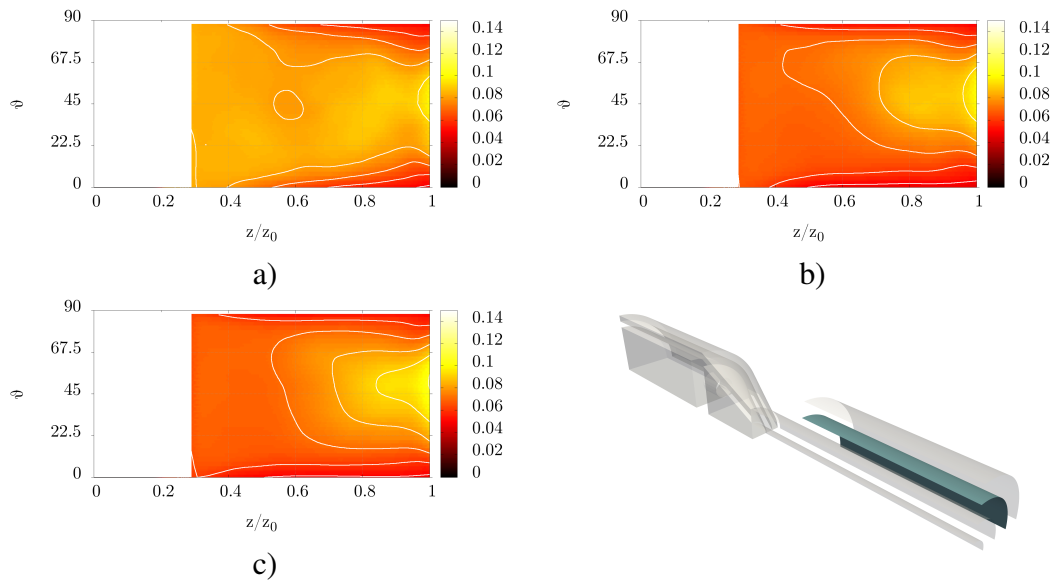


Figura 6.36: Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.6 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$

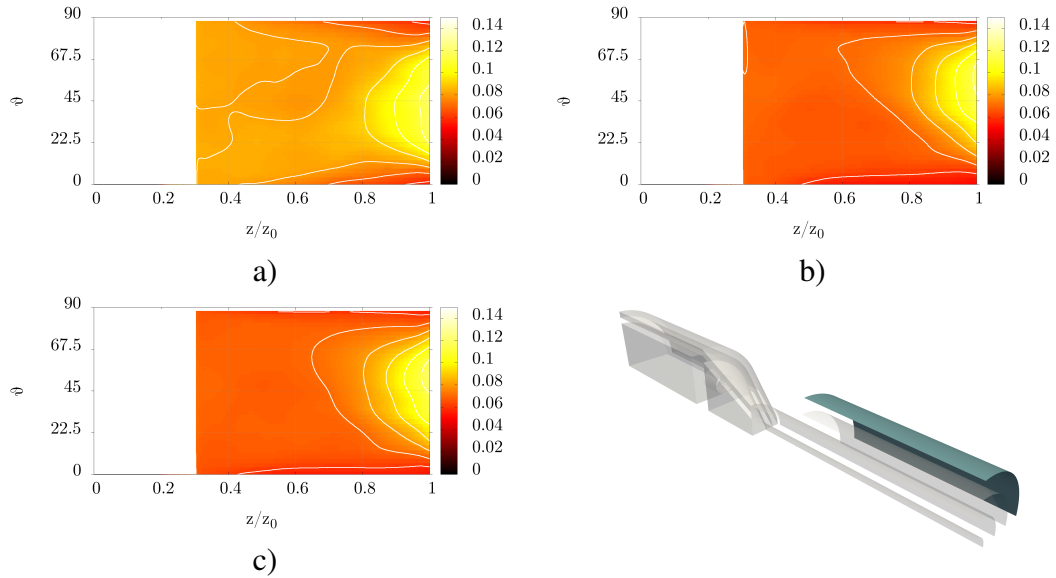


Figura 6.37: Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.8 para las simulaciones con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$

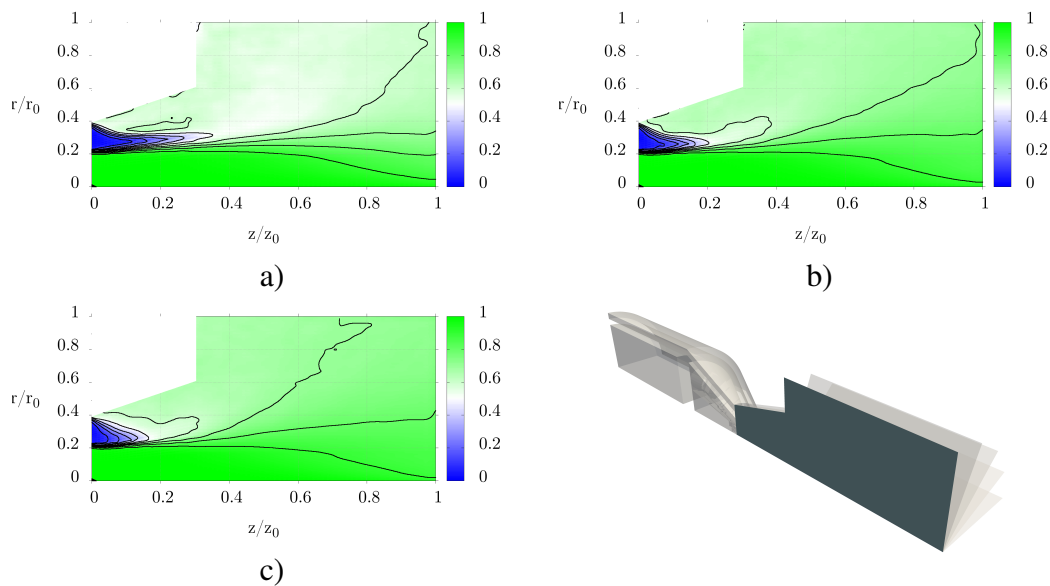


Figura 6.38: Campo de fracción volumétrica promedio de petróleo en corte longitudinal a 0 grados para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$, c) $Re_{petroleo} = 25000$

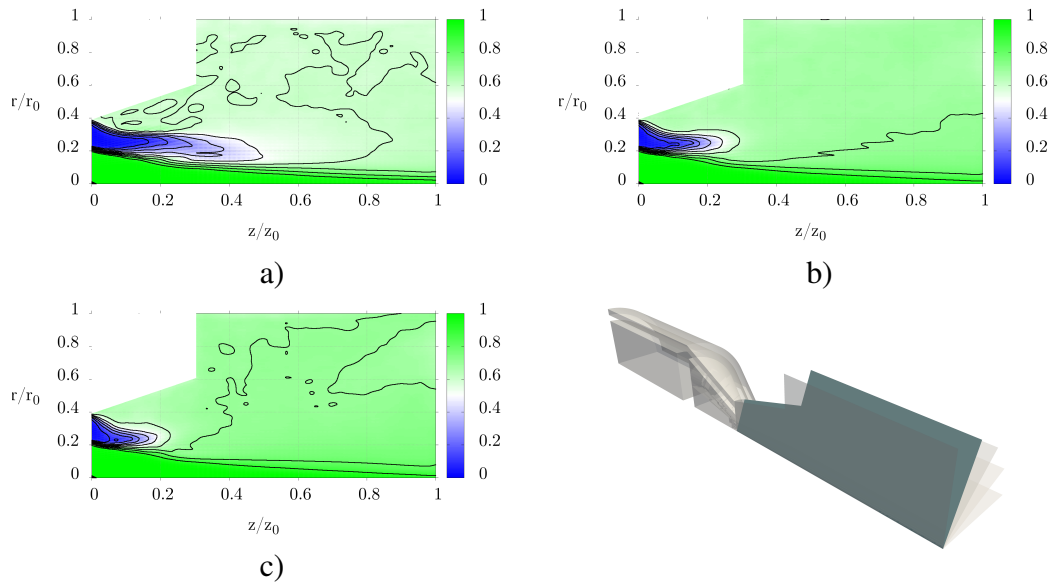


Figura 6.39: Campo de fracción volumétrica promedio de petróleo en corte longitudinal a 22,5 grados para la simulación con $Re_{agua} = 3,000$ y a) $Re_{petroleo} = 10,000$, b) $Re_{petroleo} = 20,000$, c) $Re_{petroleo} = 25,000$

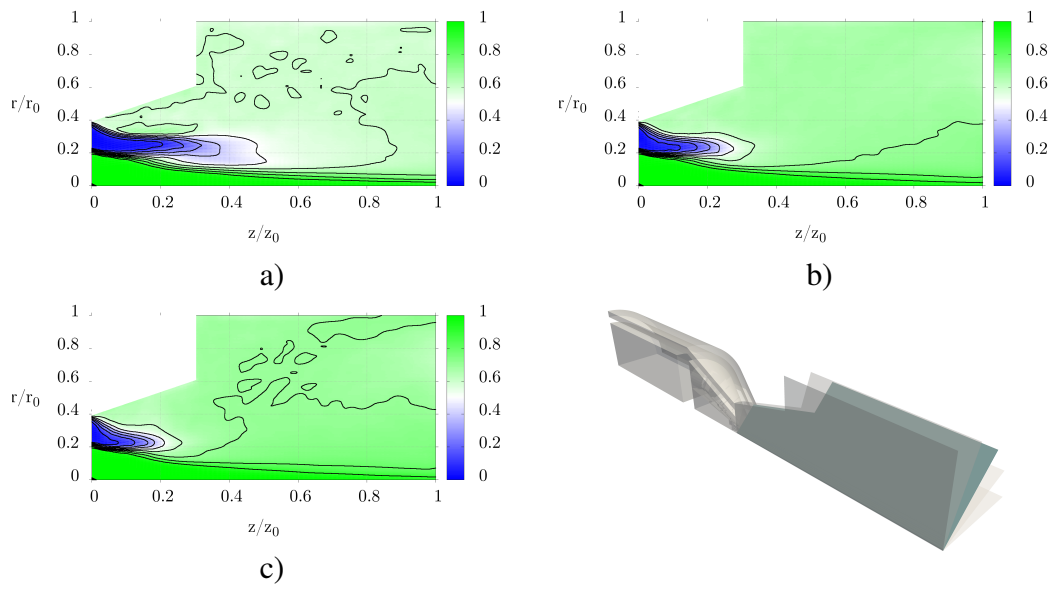


Figura 6.40: Campo de fracción volumétrica promedio de petróleo en corte longitudinal a 45 grados para la simulación con $Re_{agua} = 3000$ y a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$, c) $Re_{petroleo} = 25000$

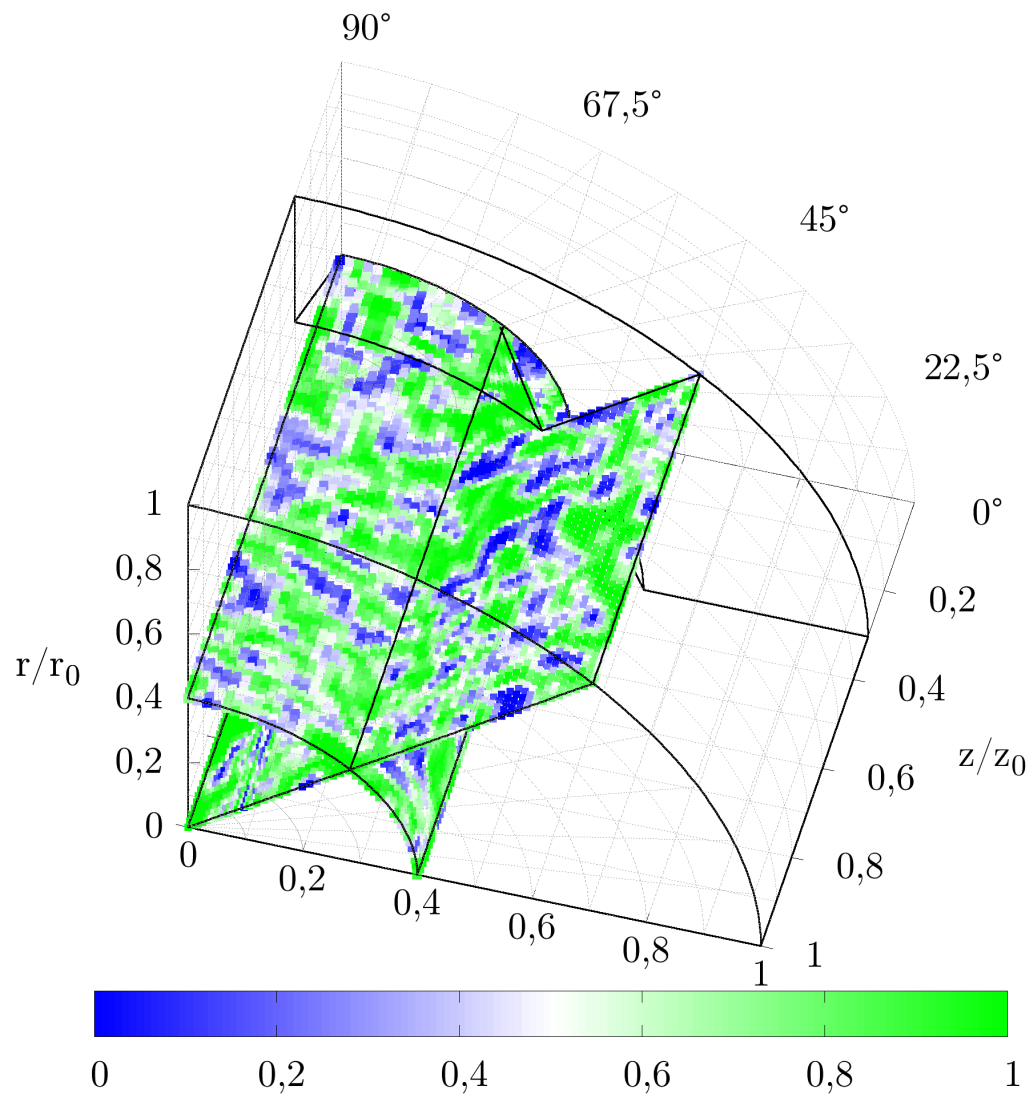


Figura 6.41: Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 10000$

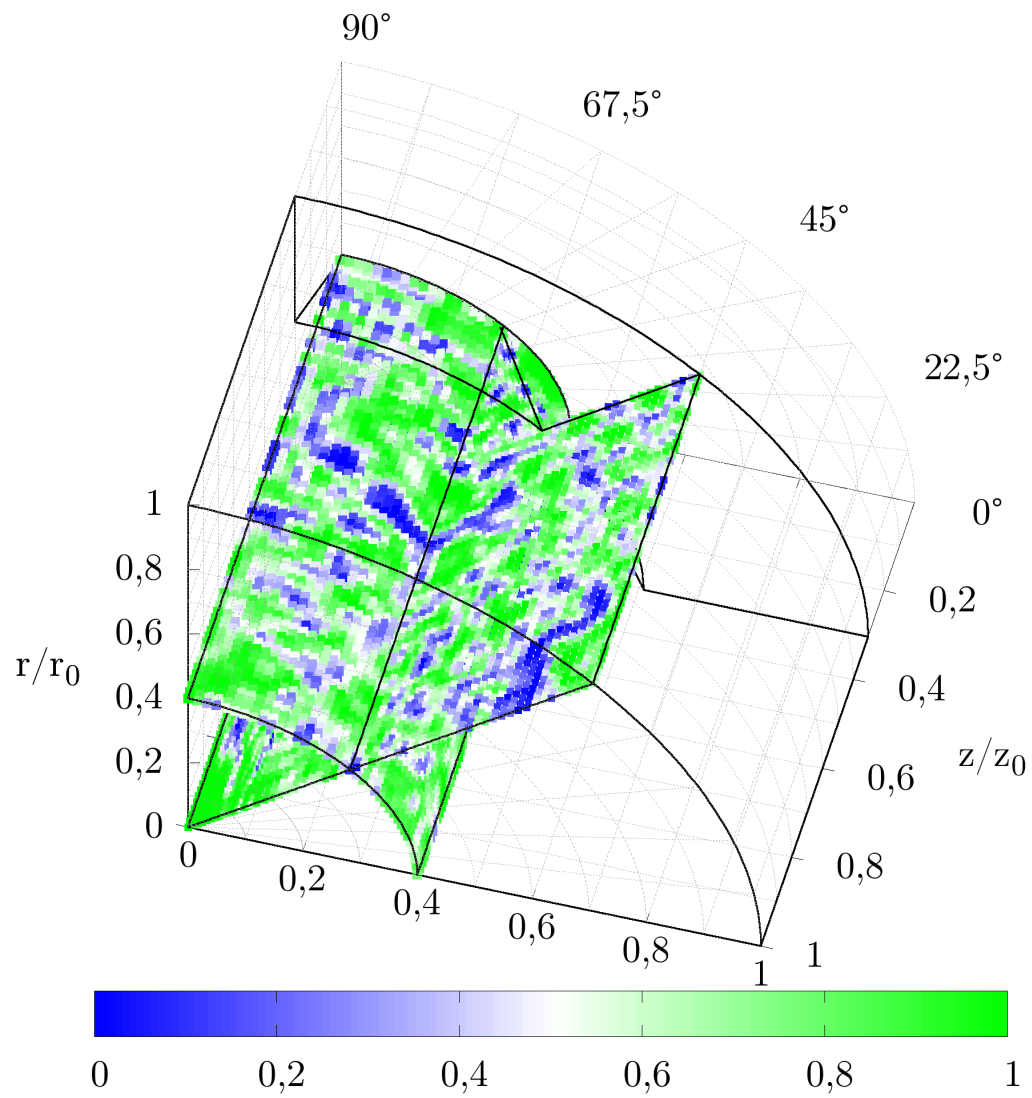


Figura 6.42: Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 20000$

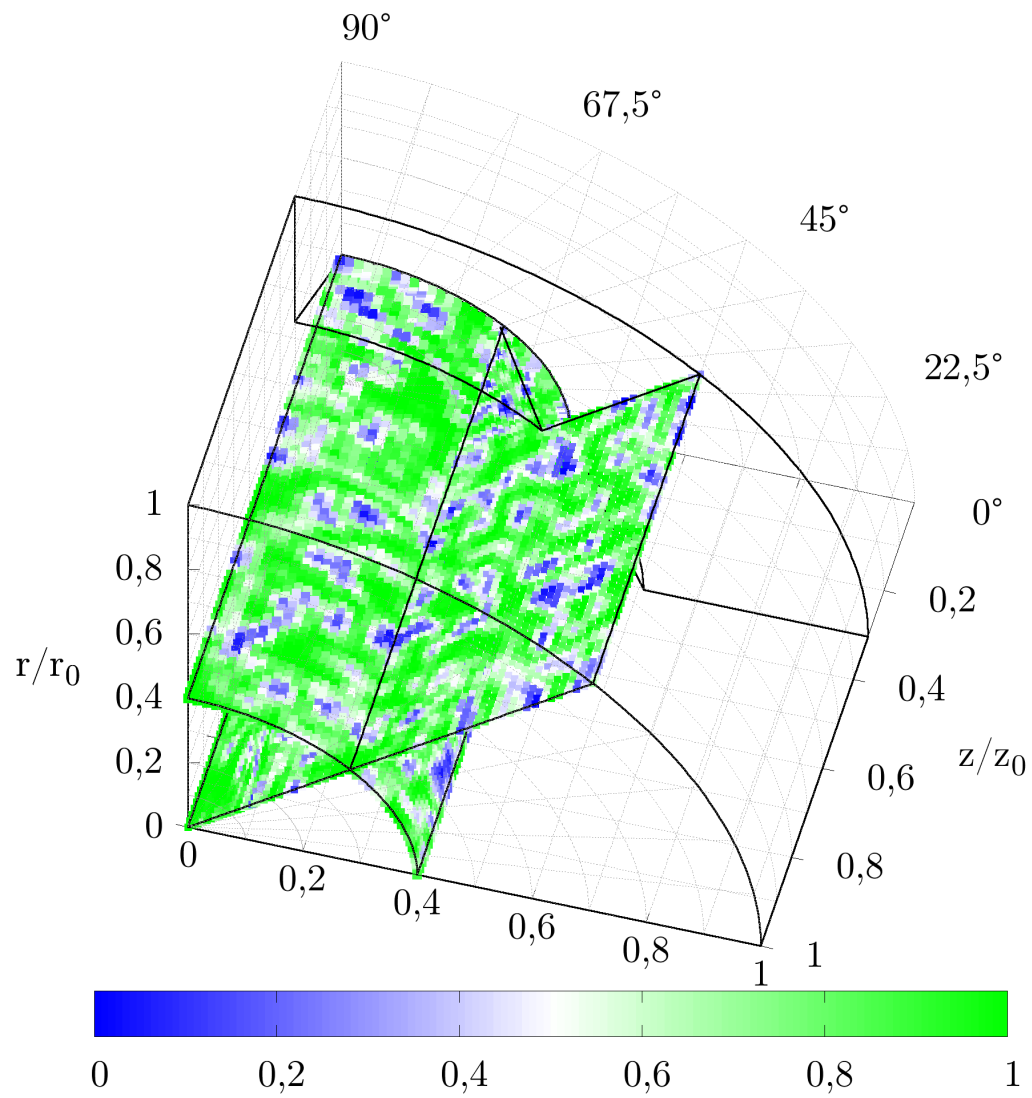


Figura 6.43: Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 3000$ y $Re_{petroleo} = 25000$

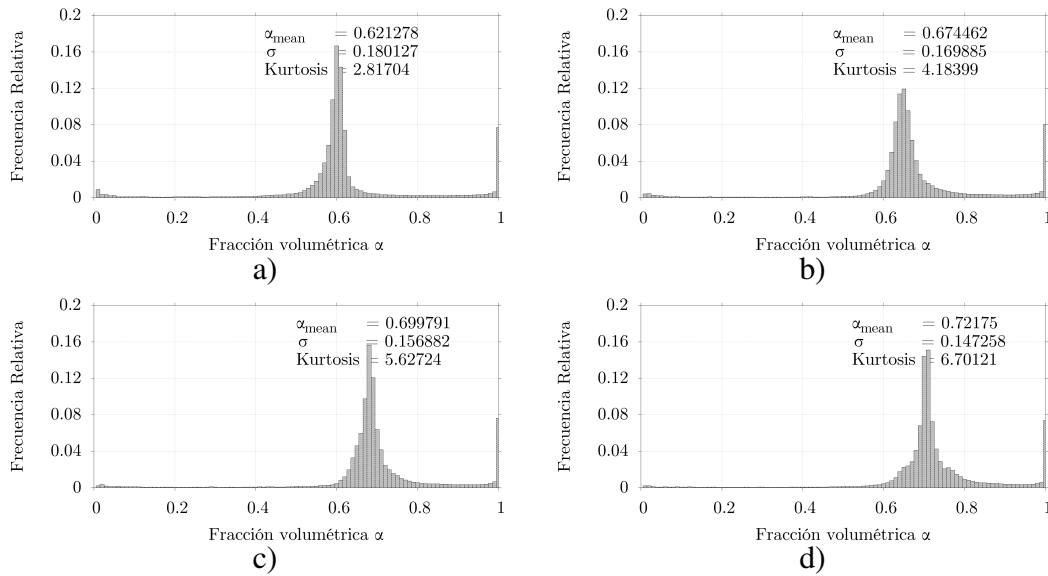


Figura 6.44: Distribución de la fracción volumétrica de petróleo al interior de la cámara de mezcla para $Re_{\text{agua}} = 3000$ y a) $Re_{\text{petroleo}} = 10000$, b) $Re_{\text{petroleo}} = 15000$, c) $Re_{\text{petroleo}} = 20000$, d) $Re_{\text{petroleo}} = 25000$

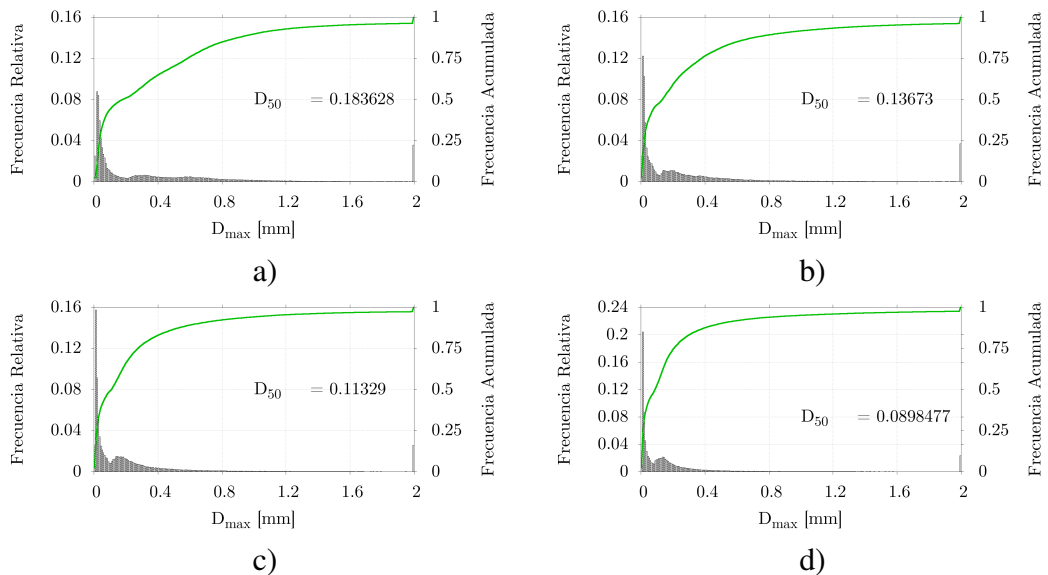


Figura 6.45: Distribución del tamaño máximo de gotas de la fase dispersa según el criterio del número de Weber crítico para la simulación con $Re_{\text{agua}} = 3000$ y a) $Re_{\text{petroleo}} = 10000$, b) $Re_{\text{petroleo}} = 15000$, c) $Re_{\text{petroleo}} = 20000$, d) $Re_{\text{petroleo}} = 25000$

6.3.2. Variación flujo agua

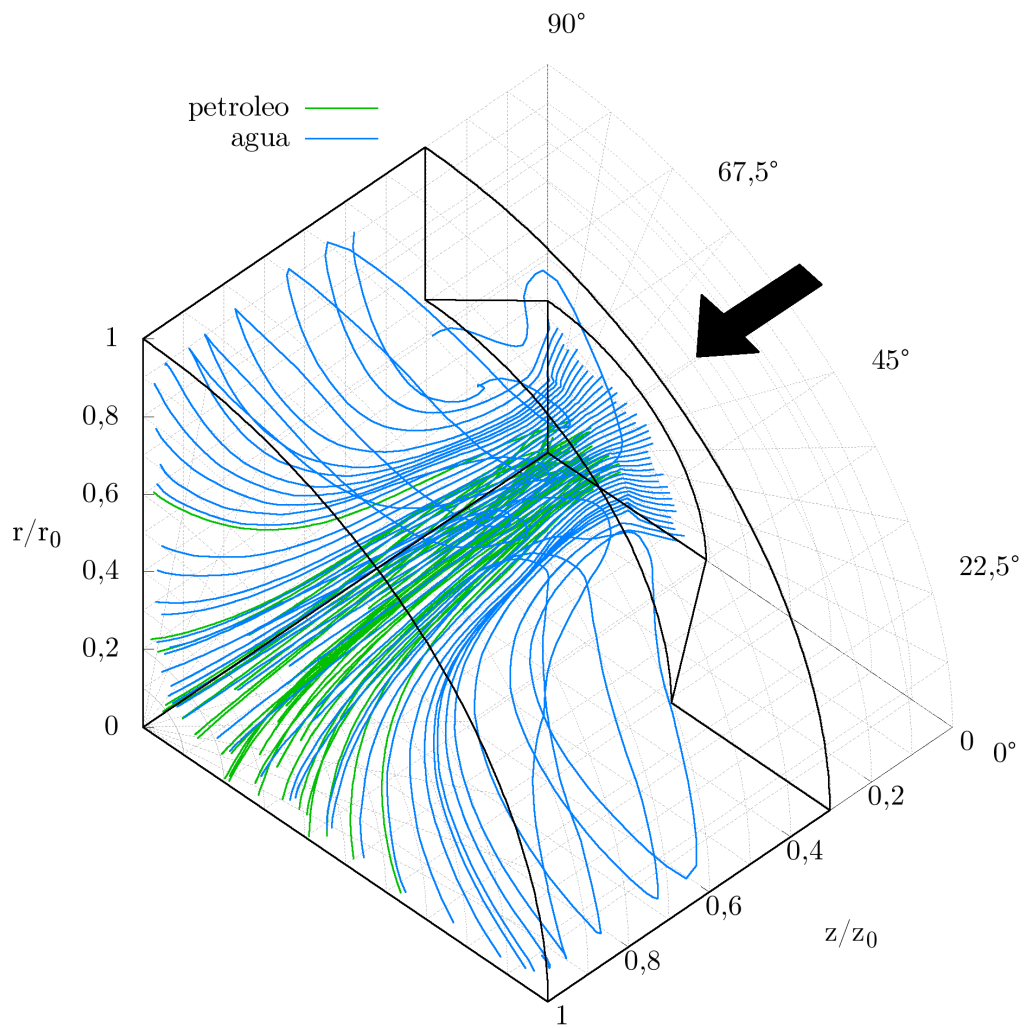


Figura 6.46: Líneas de flujo al interior de la cámara para las simulación con $Re_{petroleo} = 15000$ y $Re_{agua} = 2000$

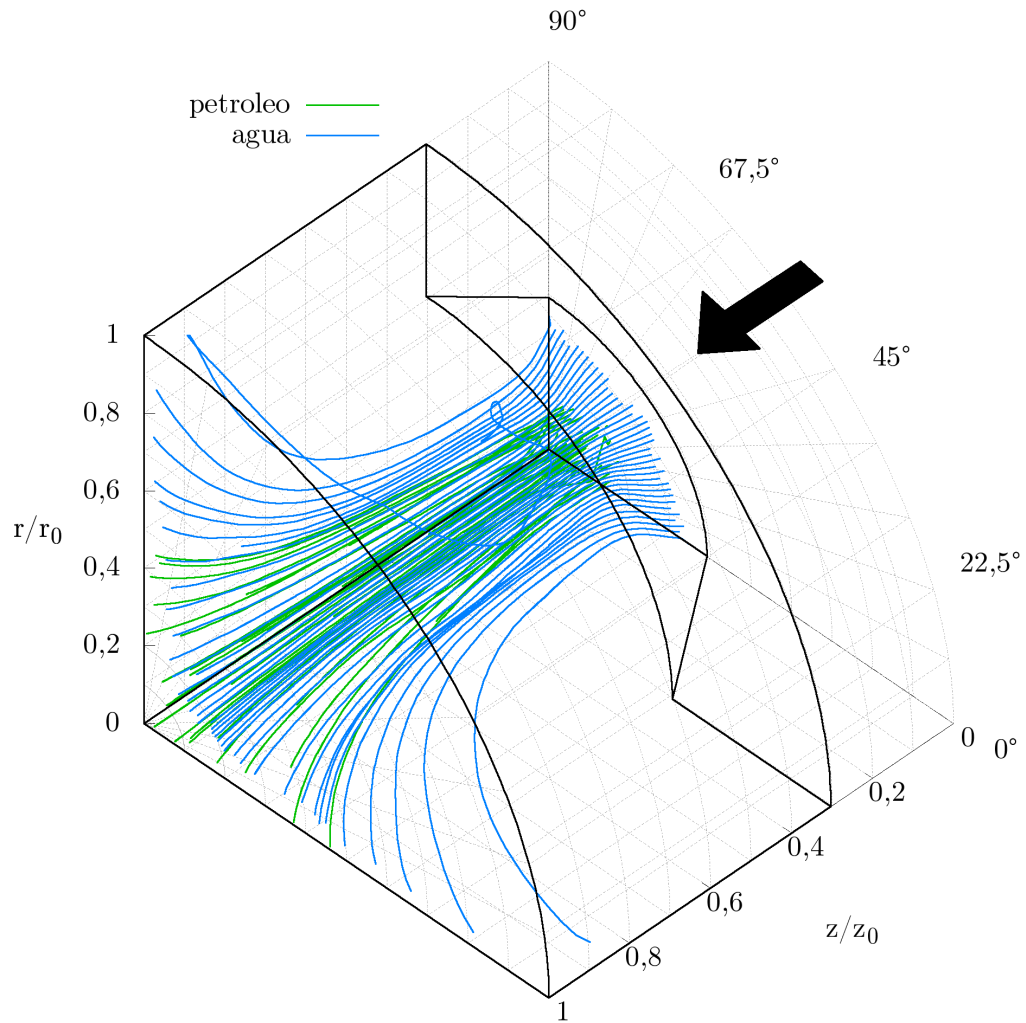


Figura 6.47: Líneas de flujo al interior de la cámara para las simulación con $Re_{petroleo} = 15000$ y $Re_{agua} = 4000$

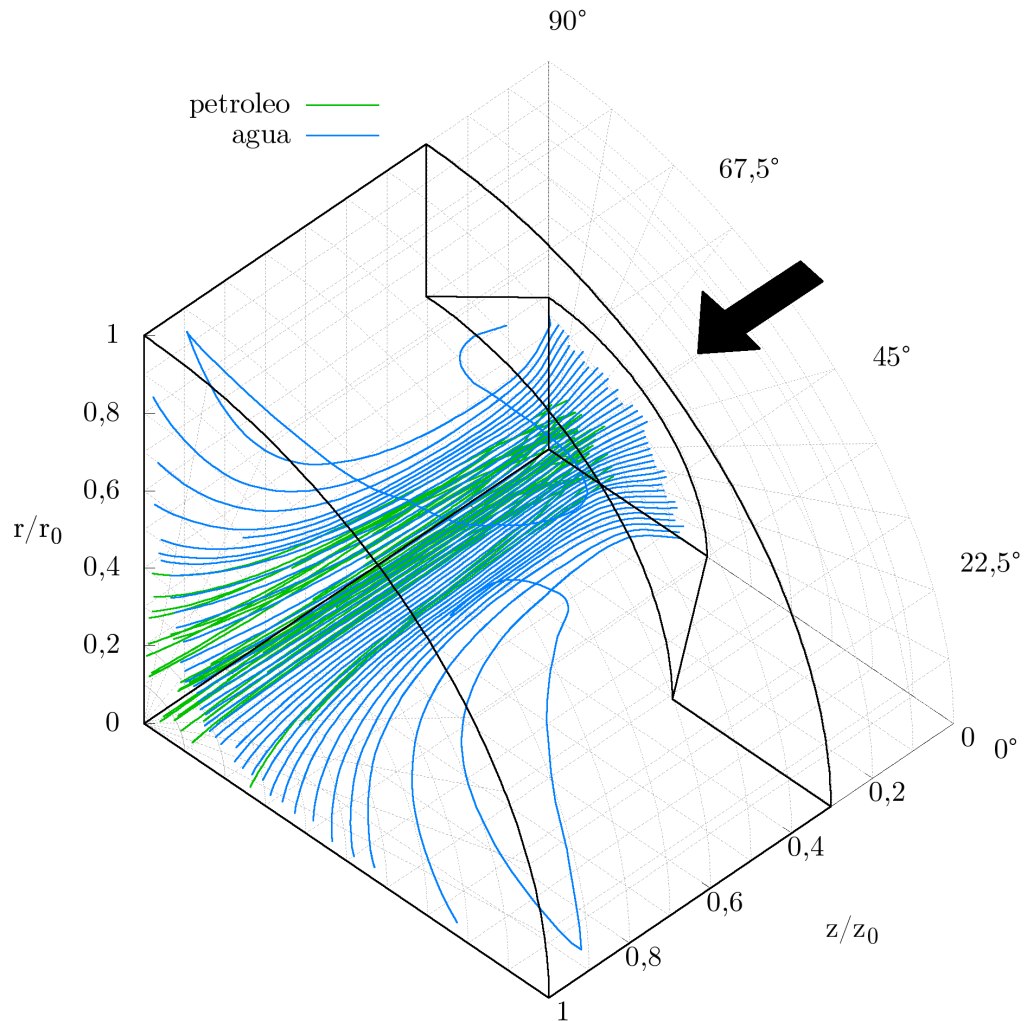


Figura 6.48: Líneas de flujo al interior de la cámara para las simulación con $Re_{petróleo} = 15000$ y $Re_{agua} = 5000$

Las figuras 6.46, 6.47 y 6.48 corresponden a las líneas de corriente promedio del flujo al interior de la cámara de mezcla para los casos en que se variaron los caudales de agua. Se puede observar de manera análoga al estudio realizado con la variación del caudal de petróleo, coherencia en las estructuras internas desarrolladas en los tres casos, existiendo variaciones exclusivamente en las intensidades del fenómeno. A medida que se incrementa el caudal de agua se produce un efecto constrictivo sobre el flujo de petróleo,

confinándolo a una zona mas cercana al eje axial de la cámara. Este efecto es mayor cuando se incrementa el numero de Reynolds para el flujo de agua desde $Re_{agua} = 2000$ a $Re_{agua} = 4000$. Sin embargo este fenómeno es menos intenso al incrementar el flujo, pasando de un numero de Reynolds de $Re_{agua} = 4000$ a $Re_{agua} = 5000$. Respecto a las velocidades máximas desarrolladas en el interior de la cámara, no se aprecian diferencias considerables al variar los caudales de agua. Las variaciones principales ocurren en la posición de los vórtices de re-circulación, el cual tiende a desplazarse aguas abajo en la cámara. Existe una zona con velocidades relativamente bajas ubicada por encima de la inyección del agua en la cámara en la que se genera un vórtice de forma anular, producto del angulo con el que ingresa el agua a la cámara. Este vórtice anular crece en dirección axial al aumentar la velocidad con que hace ingreso el agua a la cámara.

Los vórtices tipo “swirl” que se generan en la zona cercana al eje axial del inyector tienden a retraerse a medida que se ve incrementado el caudal de agua y estos tienden a posicionarse en los planos mas extremos correspondiente a los ángulos $\theta = 0^\circ$ y $\theta = 90^\circ$ mientras que los vórtices axiales de recirculación se desplazan aguas abajo en la cámara e incrementan su zona de influencia hacia los planos mas cercanos al plano medio $\theta = 45^\circ$ como se puede apreciar en las figuras 6.56 y 6.57. Al analizar la estructura de los vórtices mediante el criterio Q y Lambda 2 sobre los planos axiales se observa en primera instancia un crecimiento de los vórtices en los planos externos $\theta = 0$ y $\theta = 90$, ademas de una disminución en la contribución del flujo de petroleo en estos, pasando así a incrementar la fracción volumétrica de agua que se encuentra presente en estos como se puede apreciar en la imagen 6.29.

La distribución de la temperatura se observa en las imágenes 6.60,6.61,6.62,6.63,6.64,6.65 y 6.66. La temperatura tiende a incrementarse en la zona donde se presenta el vórtice anular debido a que el fluido atrapado en él tiene un tiempo medio de residencia en la cámara más alto que el resto. La temperatura en todas las zonas que no se ven afectadas directamente por este vórtice anular tiende a disminuir al incrementarse el flujo másico de agua.

Respecto a la composición de la mezcla para los casos simulados, se sigue observando una zona de acumulación para la fase acuosa en la región cercana a la pared de la cámara como se puede apreciar en las imágenes 6.67, 6.68, 6.69. En cuanto a la distribución de las fases (figuras 6.70, 6.71 y 6.72), para el primer caso correspondiente a un $Re_{agua} = 2000$

se observa claramente que el agua corresponde a la fase dispersa y el petróleo a la fase continua, sin embargo a medida que se incrementa el flujo de agua, se tiende hacia una configuración más simétrica entre ambas fases como puede observar en el histograma de la dispersión de la fracción volumétrica sobre el dominio 6.73.

En cuanto al tamaño máximo de gotas de la fase dispersa representado en la figura 6.74 se aprecia una coherencia en el tipo de distribución que se presenta en todas las simulaciones y un descenso en el valor del quinto decil de dichas distribuciones a medida que se incrementa el caudal de petróleo, variando desde una diámetro de gota de 0,15 a 0,12 milímetros.

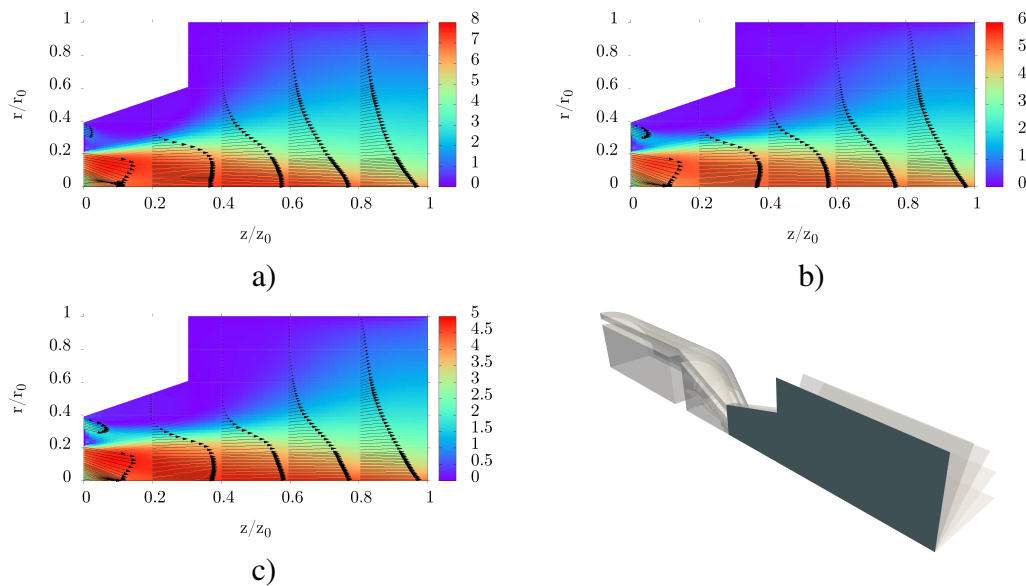


Figura 6.49: Campo de velocidad promedio en corte longitudinal a 0 grados para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

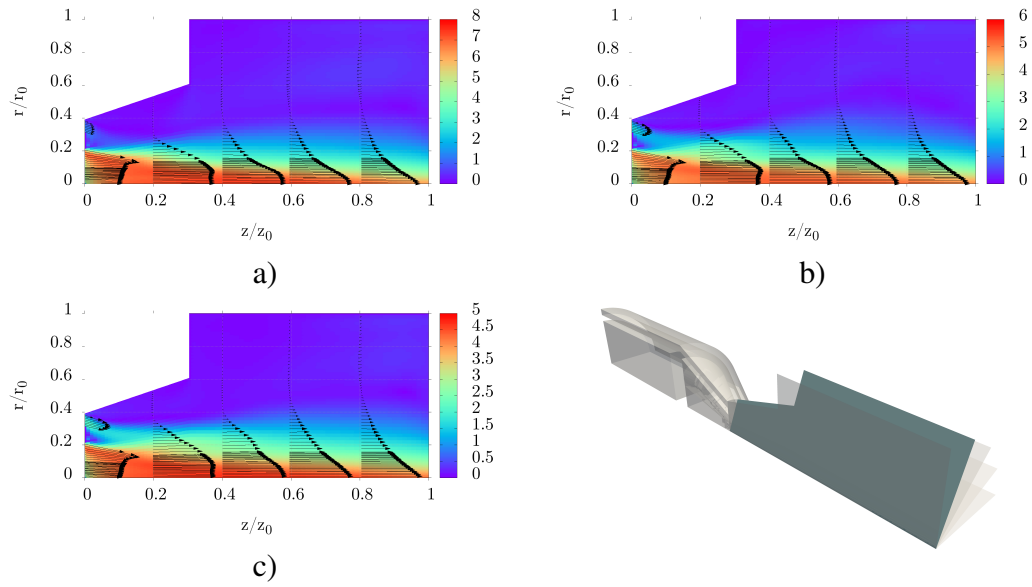


Figura 6.50: Campo de velocidad promedio en corte longitudinal a 22,5 grados para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

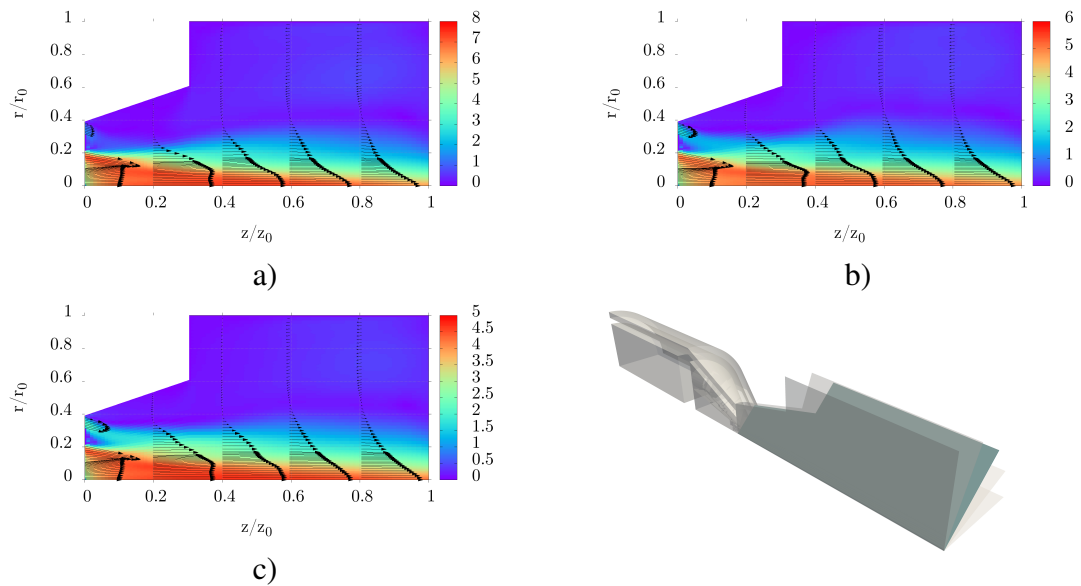


Figura 6.51: Campo de velocidad promedio en corte longitudinal a 45 grados para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

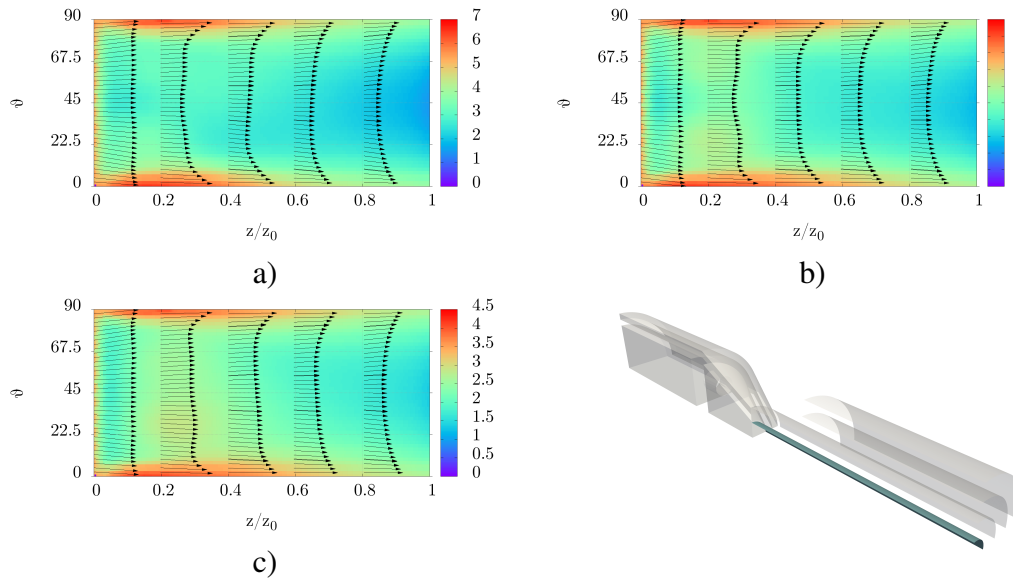


Figura 6.52: Campo de velocidad promedio en corte longitudinal sobre el radio 0.2 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

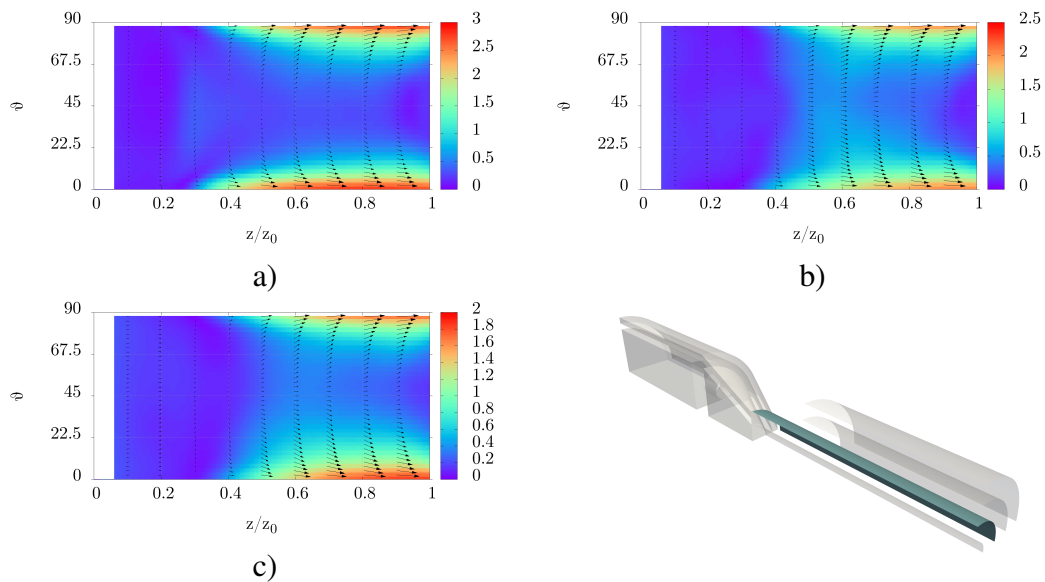


Figura 6.53: Campo de velocidad promedio en corte longitudinal sobre el radio 0.4 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

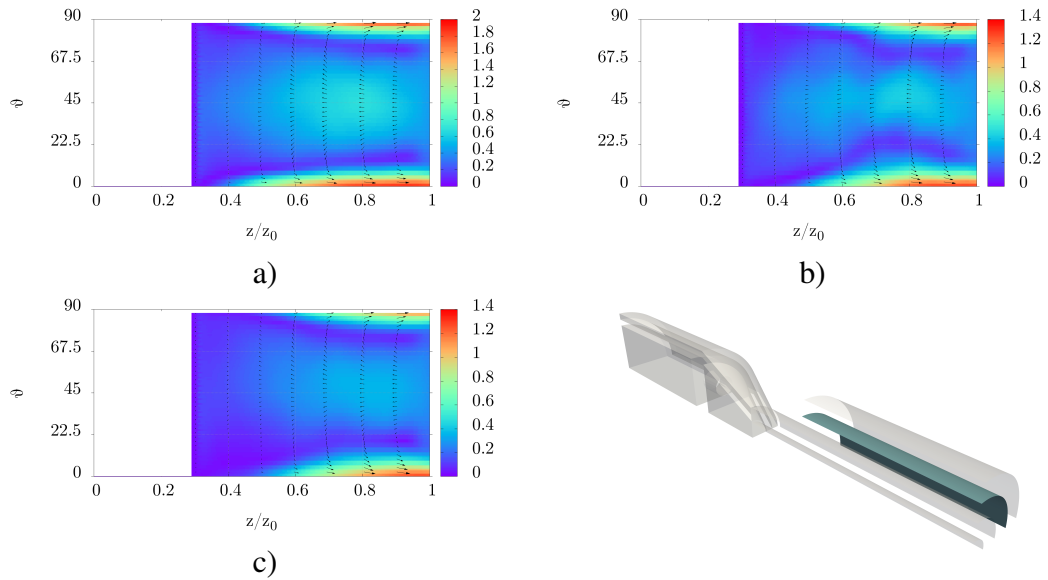


Figura 6.54: Campo de velocidad promedio en corte longitudinal sobre el radio 0.6 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

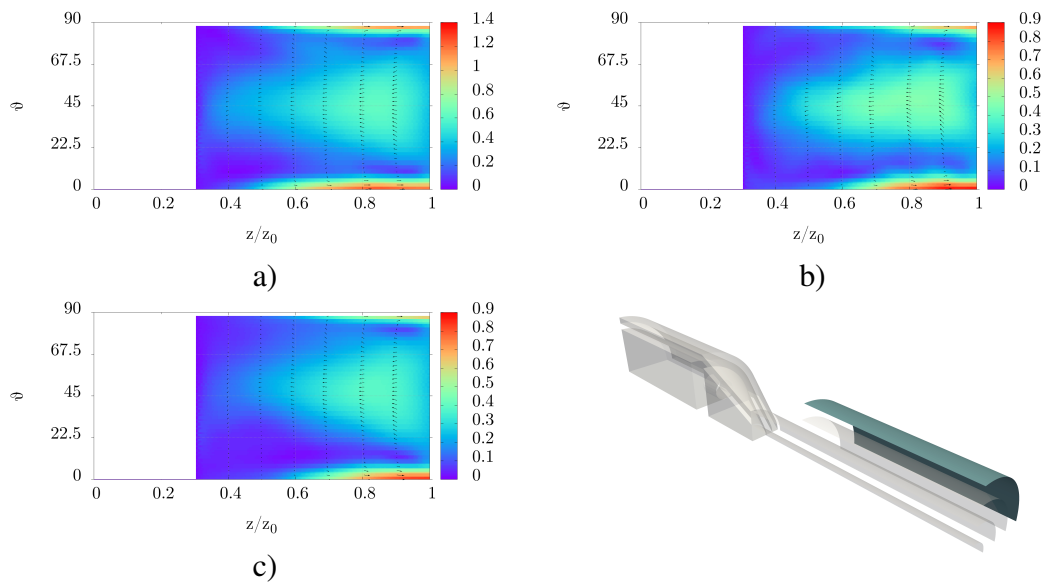
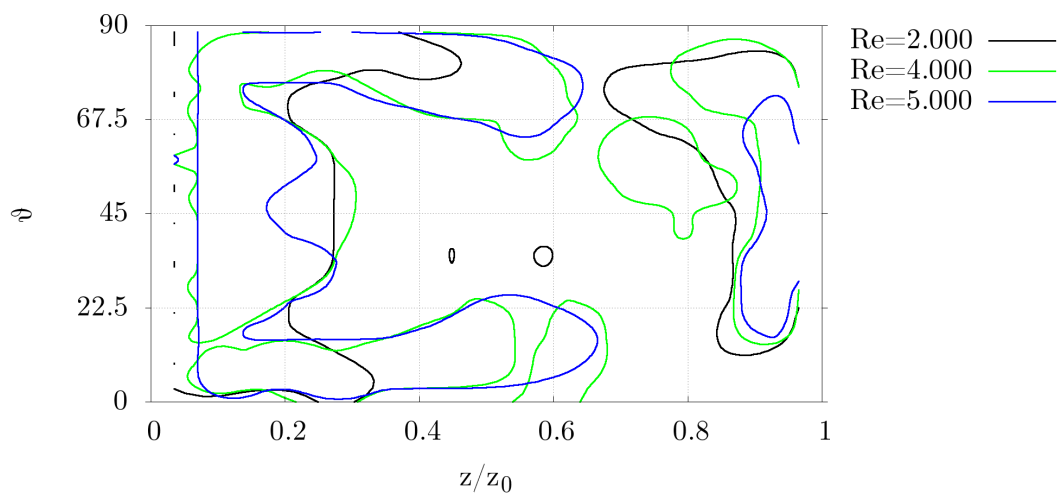
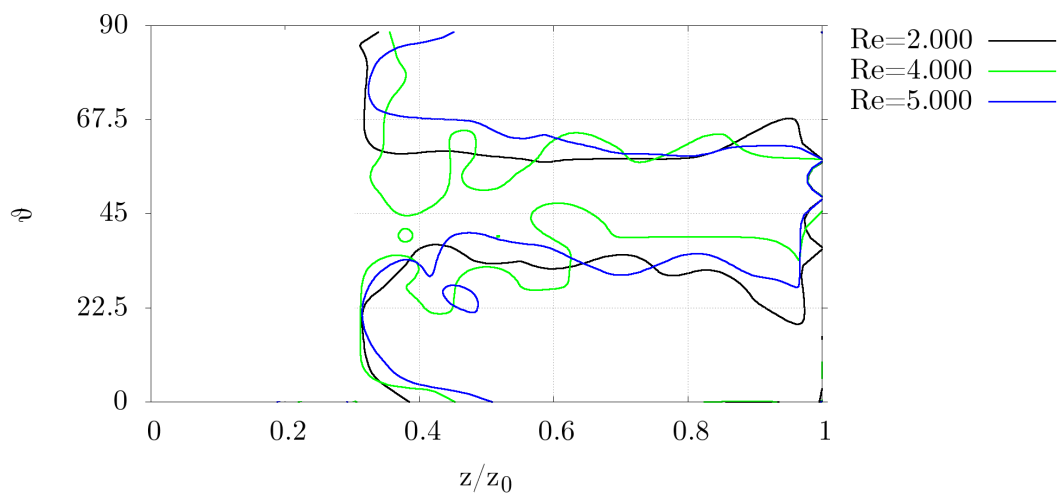


Figura 6.55: Campo de velocidad promedio en corte longitudinal sobre el radio 0.8 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$



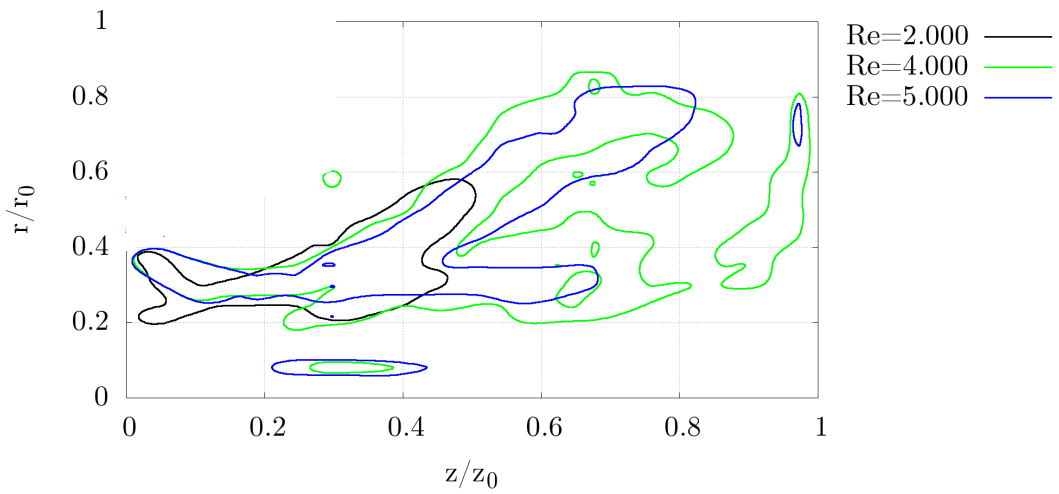
a)

Figura 6.56: Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de agua sobre el plano $r = 0,2$



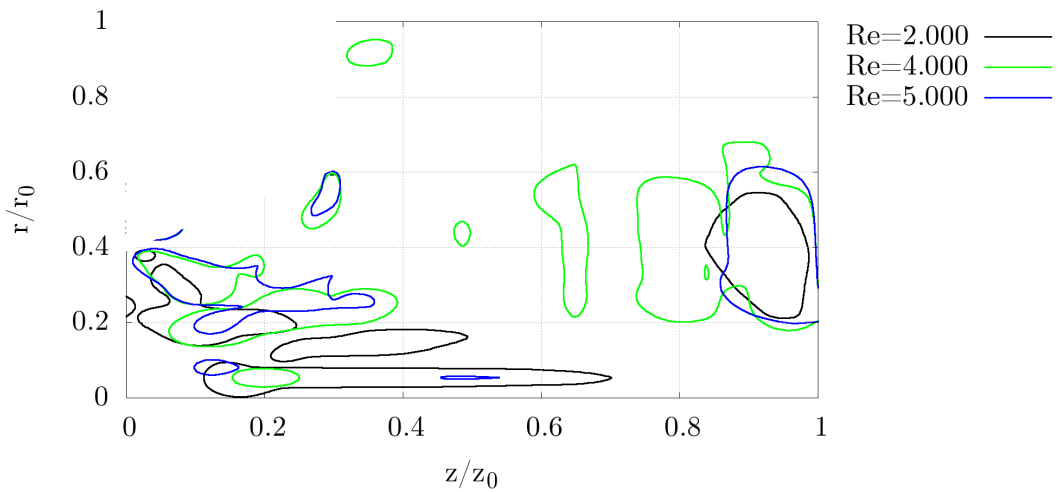
a)

Figura 6.57: Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de agua sobre el plano $r = 0,8$



a)

Figura 6.58: Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de agua sobre el plano $\theta = 0$



a)

Figura 6.59: Visualización de vórtices mediante el criterio Q para las simulaciones variando el caudal de agua sobre el plano $\theta = 45$

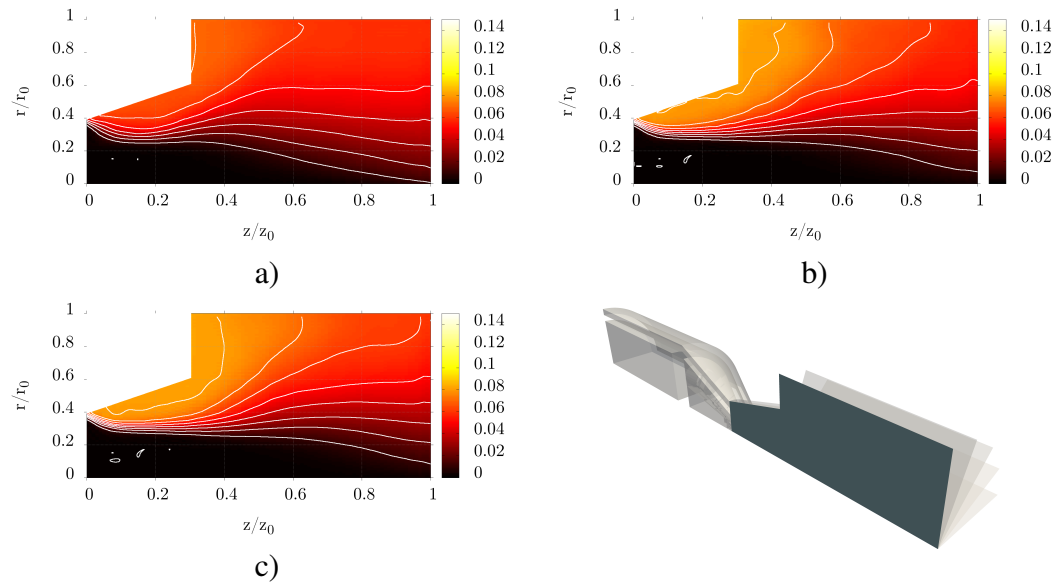


Figura 6.60: Campo de Temperatura adimensionalizada promedio en corte longitudinal a 0 grados para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

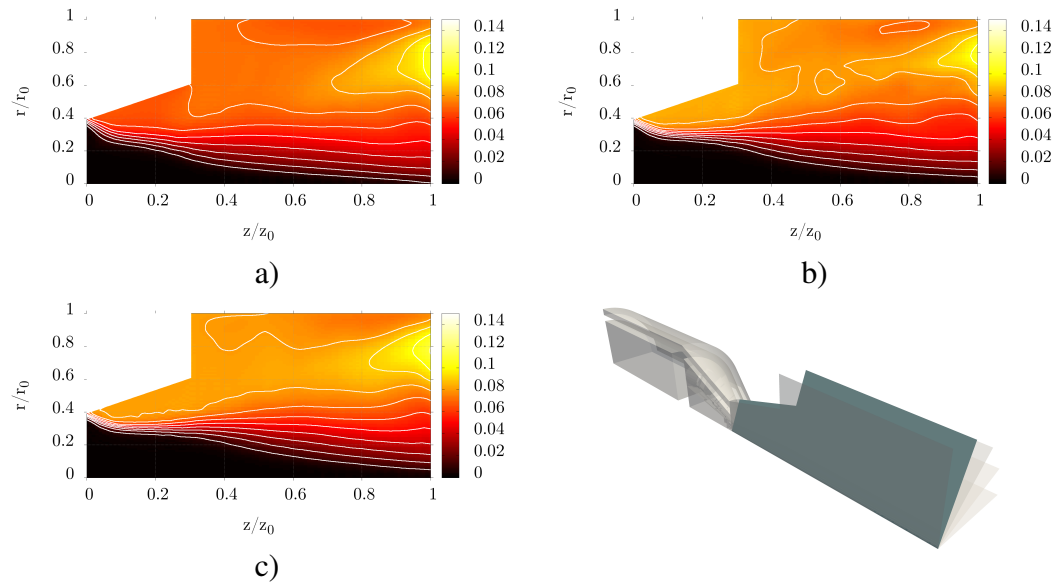


Figura 6.61: Campo de Temperatura adimensionalizada promedio en corte longitudinal a 22,5 grados para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

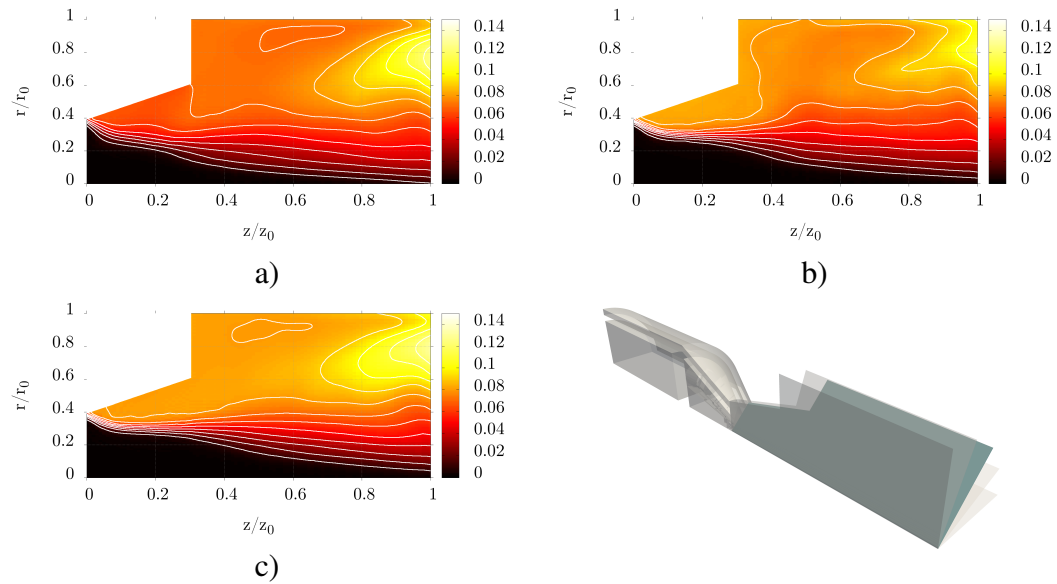


Figura 6.62: Campo de Temperatura adimensionalizada promedio en corte longitudinal a 45 grados para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

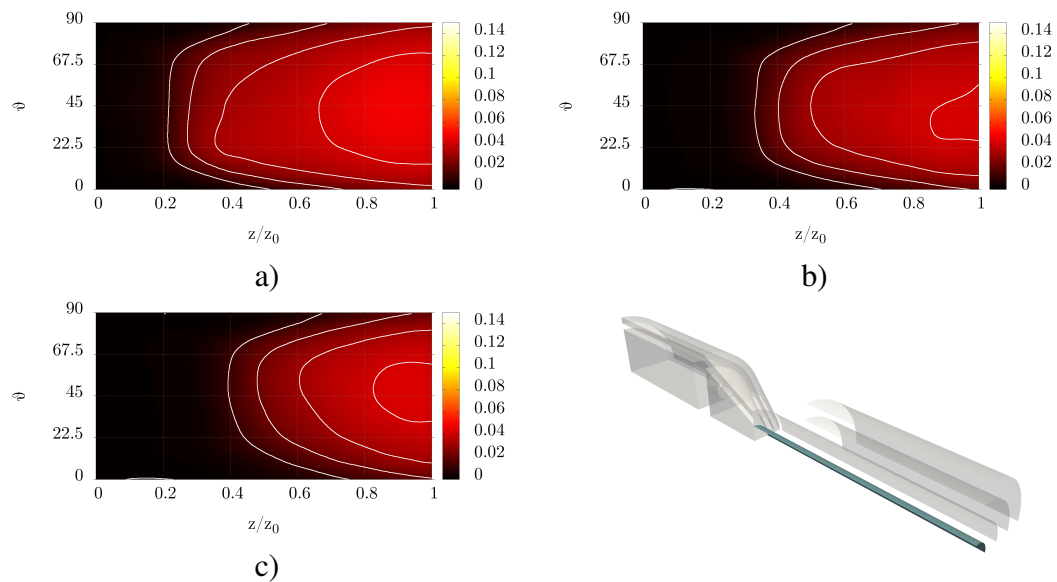


Figura 6.63: Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.2 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

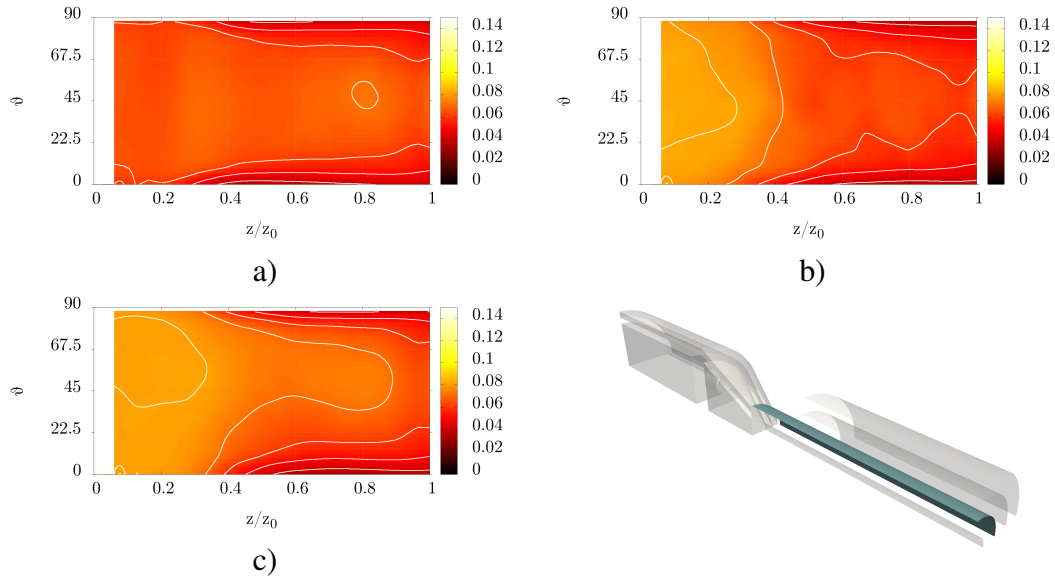


Figura 6.64: Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.4 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

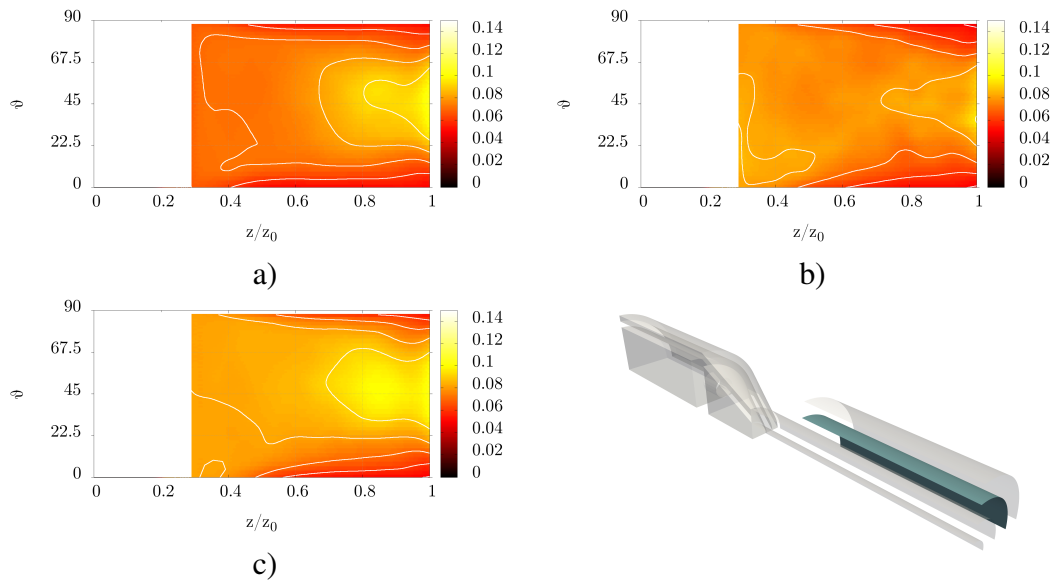


Figura 6.65: Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.6 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

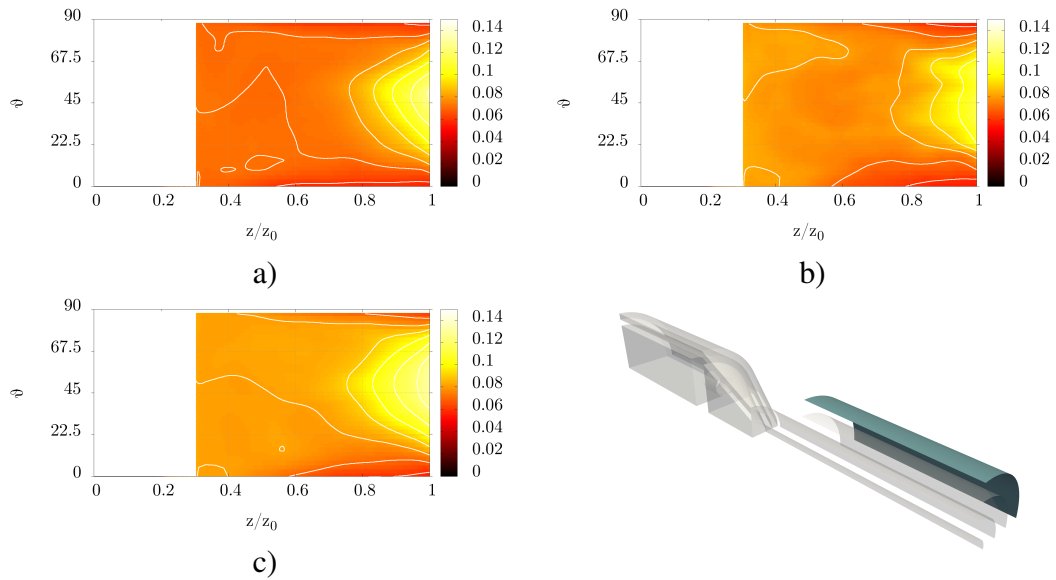


Figura 6.66: Campo de temperatura adimensionalizada promedio en corte longitudinal sobre el radio 0.8 para las simulaciones con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

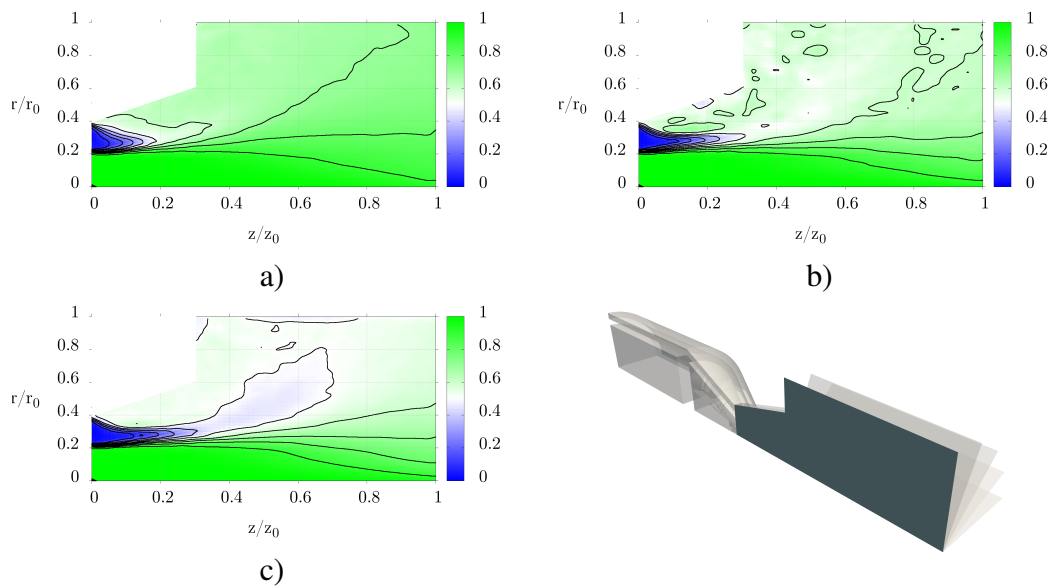


Figura 6.67: Campo de fracción volumétrica promedio de petróleo en corte longitudinal a 0 grados para la simulación con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$, c) $Re_{agua} = 5000$

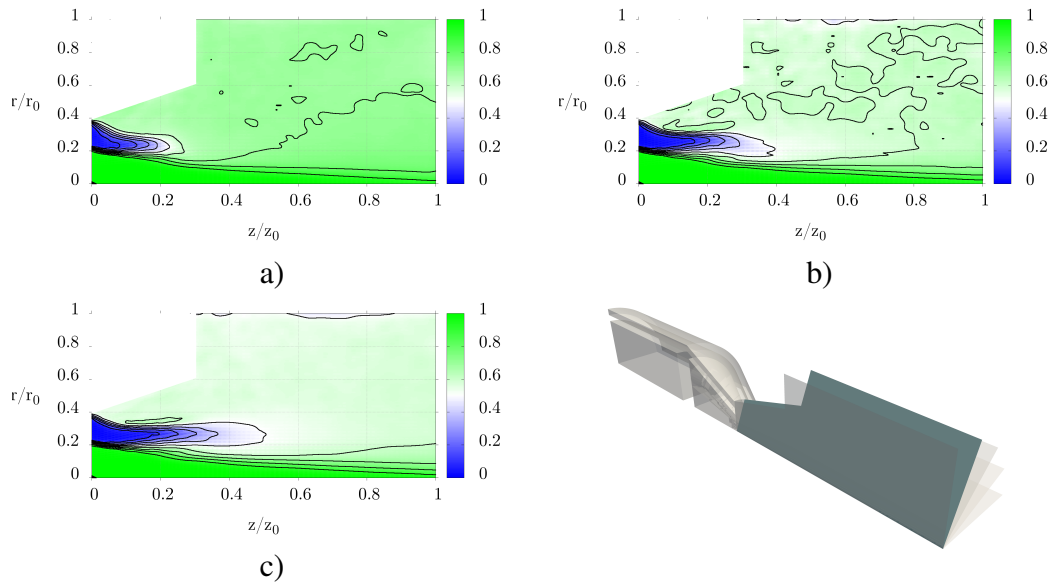


Figura 6.68: Campo de fracción volumétrica promedio de petróleo en corte longitudinal a 22,5 grados para la simulación con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$, c) $Re_{agua} = 5000$

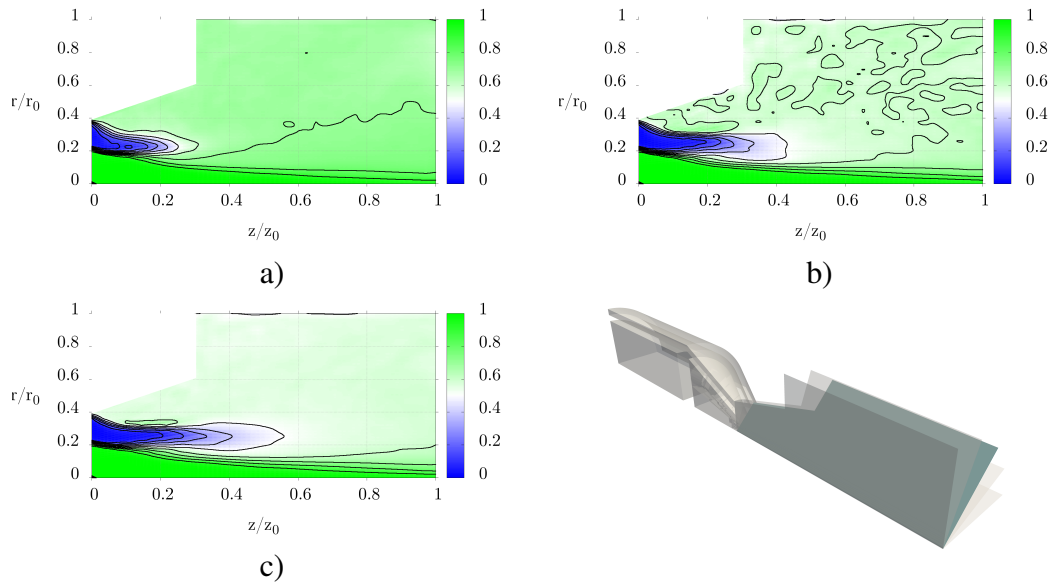


Figura 6.69: Campo de fracción volumétrica promedio de petróleo en corte longitudinal a 45 grados para la simulación con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$, c) $Re_{agua} = 5000$

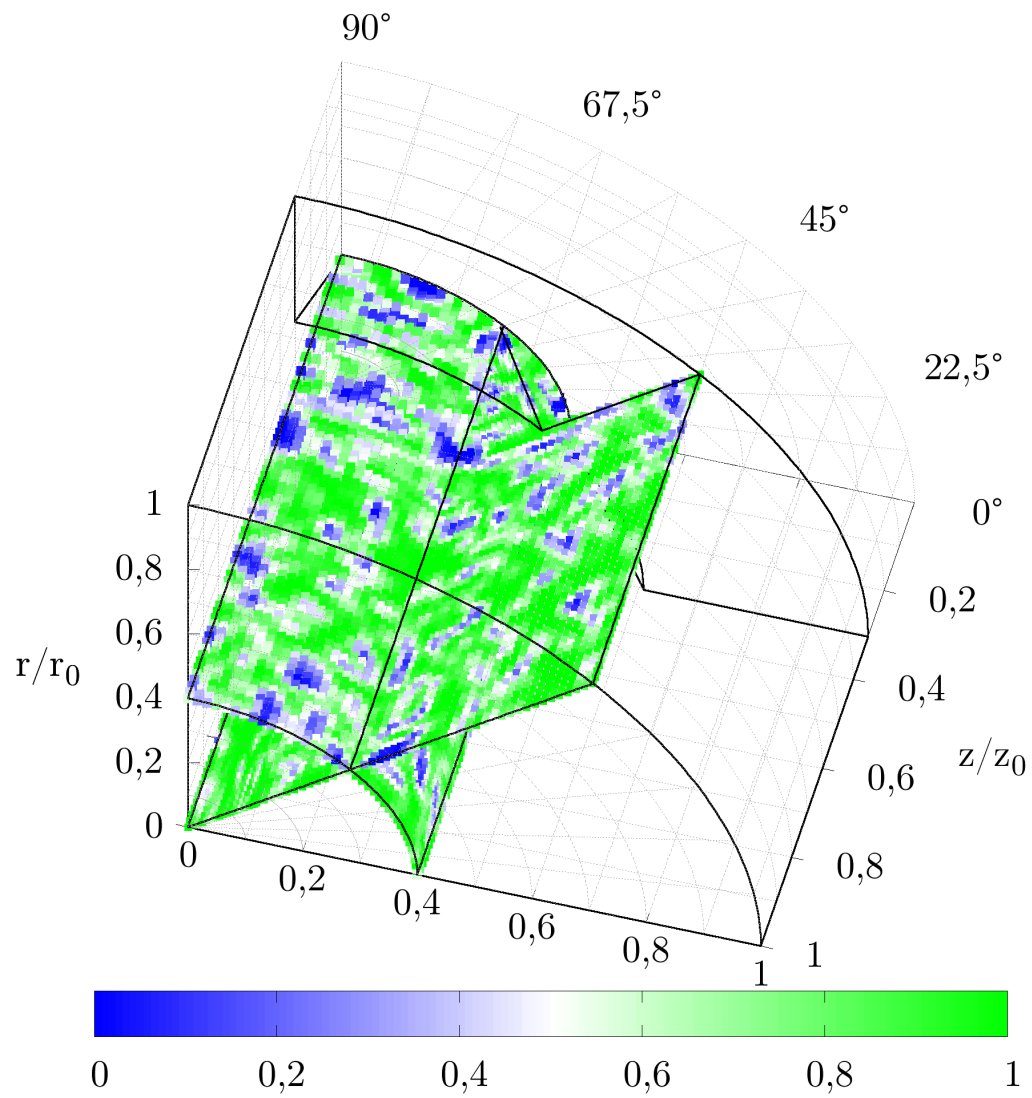


Figura 6.70: Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 2000$ y $Re_{petroleo} = 15000$

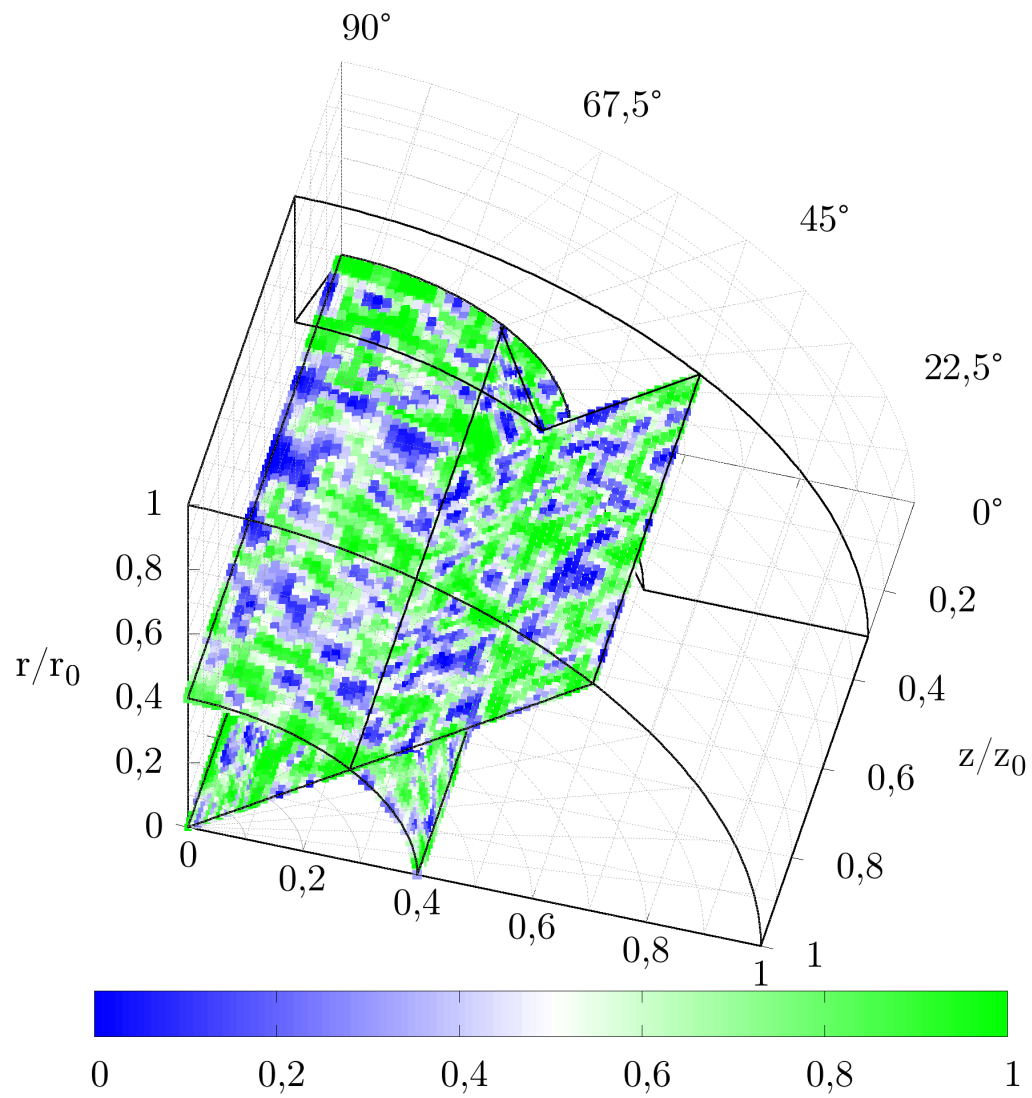


Figura 6.71: Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 4000$ y $Re_{petroleo} = 15000$

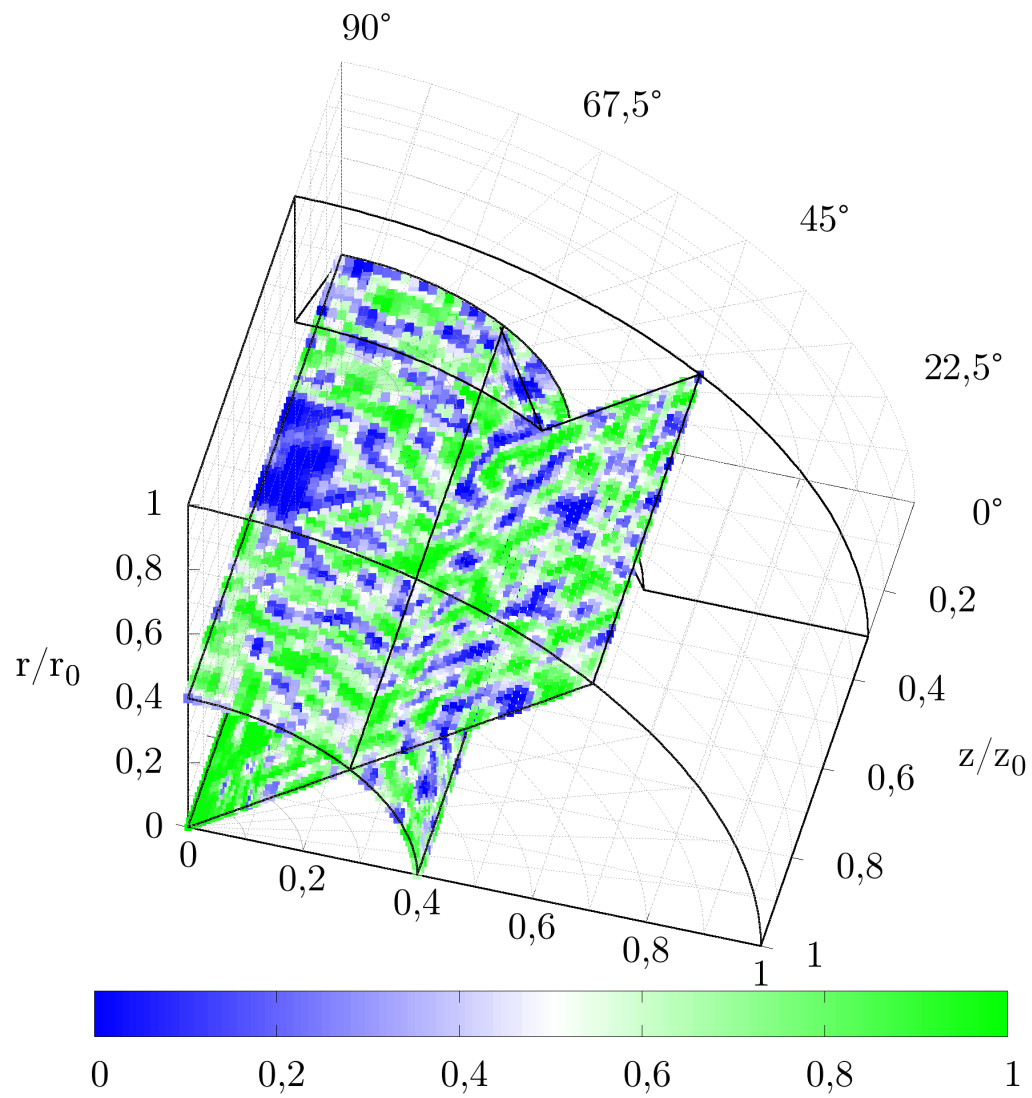


Figura 6.72: Campo de fracción volumétrica para el tiempo adimensionalizado $t^* = 30$ para la simulación con $Re_{agua} = 5000$ y $Re_{petroleo} = 15000$

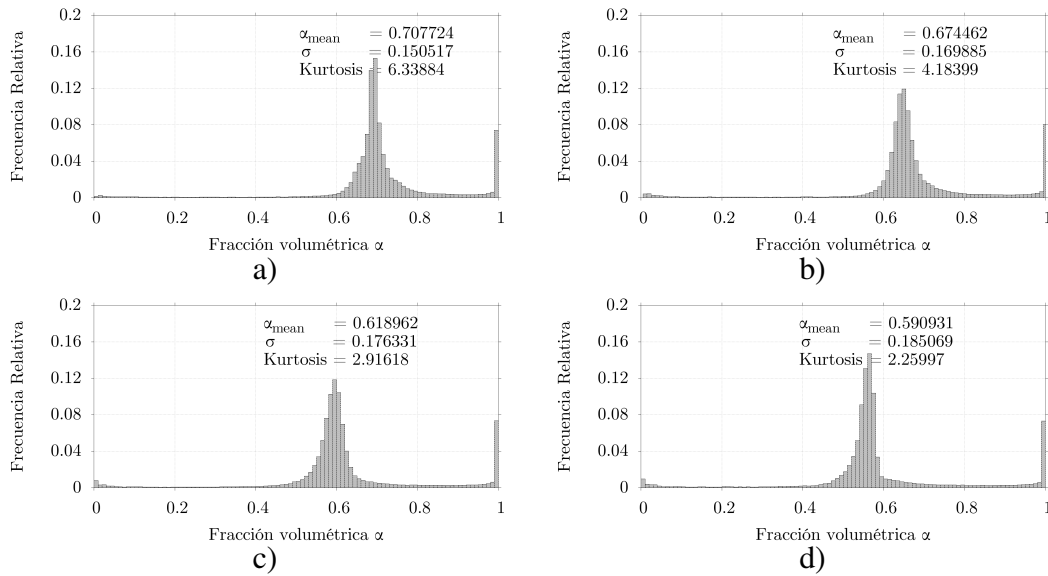


Figura 6.73: Distribución de la fracción volumétrica de petróleo al interior de la cámara de mezcla para $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 3000$, c) $Re_{agua} = 4000$, d) $Re_{agua} = 5000$

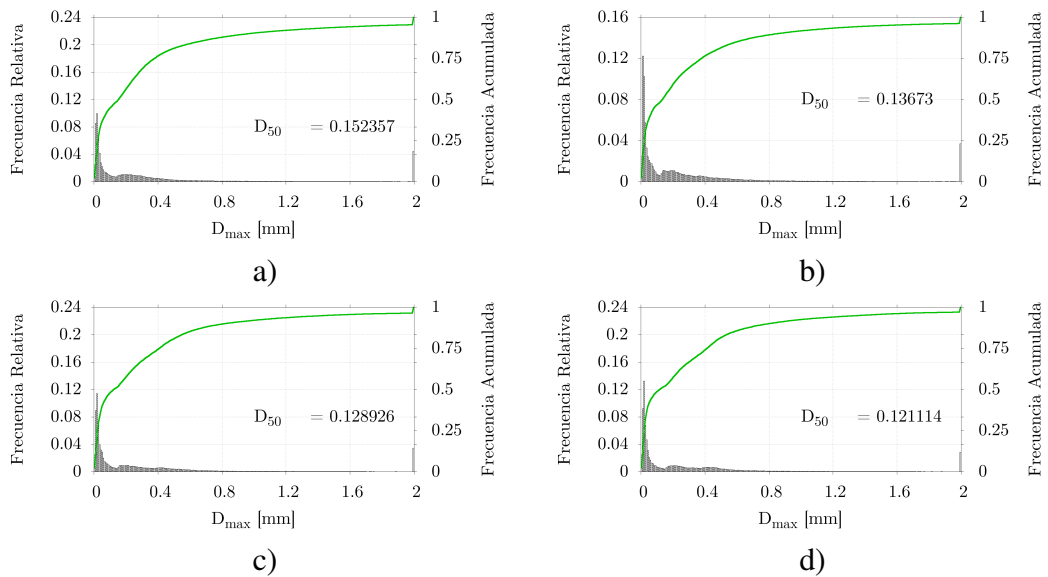


Figura 6.74: Distribución del tamaño máximo de gotas de la fase dispersa según el criterio del numero de Weber critico para la simulación con $Re_{petroleo} = 15000$ y a) $Re_{agua} = 2000$, b) $Re_{agua} = 3000$, c) $Re_{agua} = 4000$, d) $Re_{agua} = 5000$

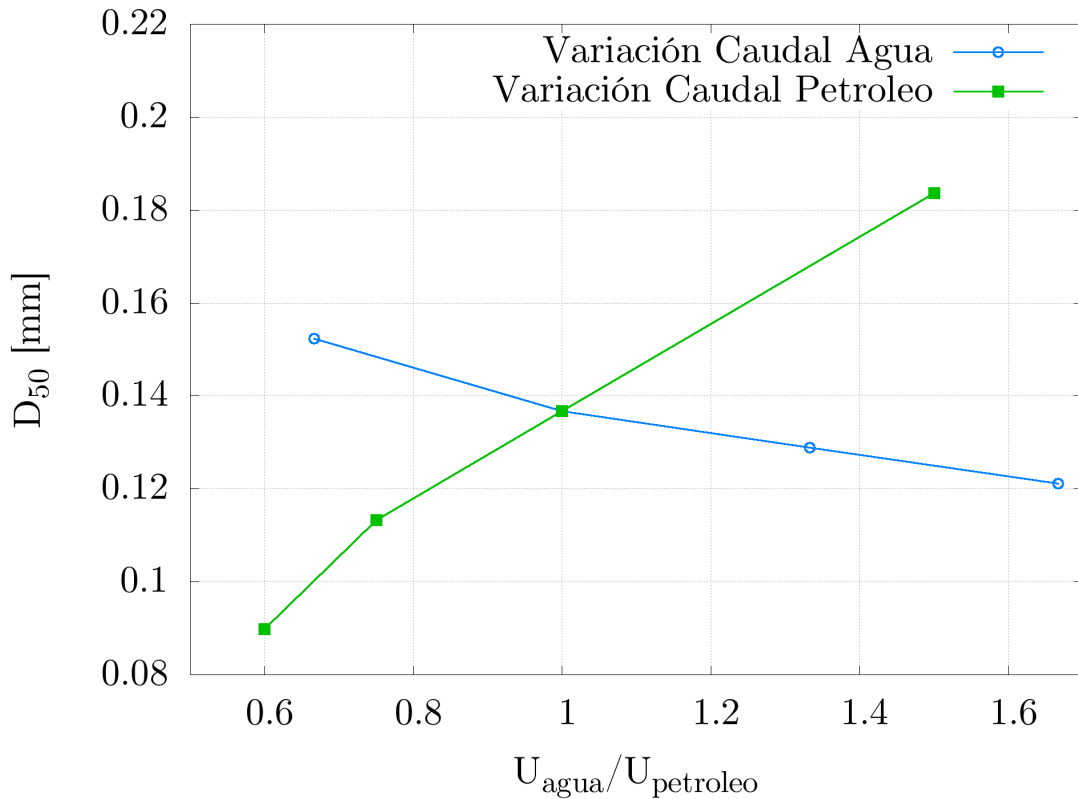


Figura 6.75: Variación del tamaño de gotas de la fase dispersa a distinta relación entre las velocidades del caudal de agua y petróleo

Del análisis de la figura 6.75 se puede observar que tanto al incrementar el caudal de agua como el de petróleo se produce una disminución en el diámetro máximo de las gotas de la fase dispersa que se estabilizan al interior de la cámara de mezcla. Dicha variación sigue una tendencia de decrecimiento lineal lo cual se explica debido a que a mayor caudal, las velocidades alcanzadas al interior de la cámara de premezclado aumentan, incrementando en el esfuerzo de corte sobre la interfaz elevando el número de Weber de las gotas por sobre el número de Weber crítico generando que estas se desestabilicen y se fracturen.

6.4. Análisis espectral

Debido a la posibilidad de que exista el fenómeno de combustión inestable en el quemador, es decir, la generación de ondas acústicas, se debe analizar las fluctuaciones en el campo de la presión. Ocurre dicho fenómeno cuando estas oscilaciones de la presión entran en resonancia con la tasa de calor fluctuante entregada por la combustión. El espectro de frecuencias para la presión a la salida del inyector para los distintos casos simulados se presenta en las imágenes 6.76 y 6.77. Al analizar la distribución de las frecuencias no es posible determinar alguna frecuencia característica predominante en el proceso, lo cual se debe a que para los números de Reynolds estudiados, no existe una regularidad en el desprendimiento de vórtices. Sin embargo se pueden determinar ciertas frecuencias que corresponden a máximos locales en el rango de 0 a 1000 Hertz. En particular, la simulación con $Re_{agua} = 4000$ presenta un peak en las frecuencias cercanas a los 1000 Hertz, la cual se desplaza hasta los 450 Hertz cuando se incrementa el caudal de agua hasta un número de Reynolds del agua de 5000. Respecto al análisis espectral para las simulaciones variando el caudal de petróleo, ocurre un comportamiento similar al observado previamente. Para los caudales más bajos, no es posible determinar una frecuencia predominante en la oscilación de la presión. Sin embargo al aumentar el número de Reynolds, comienzan a hacerse predominantes las frecuencias del orden de los 480 y 1000 [Hertz].

Al analizar los espectros de energía para el flujo desarrollado al interior de la cámara de combustión, se puede observar que estos cumplen con lo predicho por la teoría y asemejan su comportamiento al espectro de Kolmogorov, siguiendo la tendencia de un decaimiento de potencia $-5/3$. Es importante destacar que el espectro de energía calculado difiere del espectro clásico pues, en este solo se toma en consideración la velocidad del fluido. Pero al tratarse de un flujo multifásicos, la energía en un punto depende tanto de la velocidad como de la densidad de la fase presente en dicho punto. Por este motivo se calculó el espectro de energía utilizando el campo de la cantidad de movimiento. En cuanto a la discretización espacial y el tamaño de la malla, esta permite simular las escalas principales del fluido y el comienzo del subrango inercial.

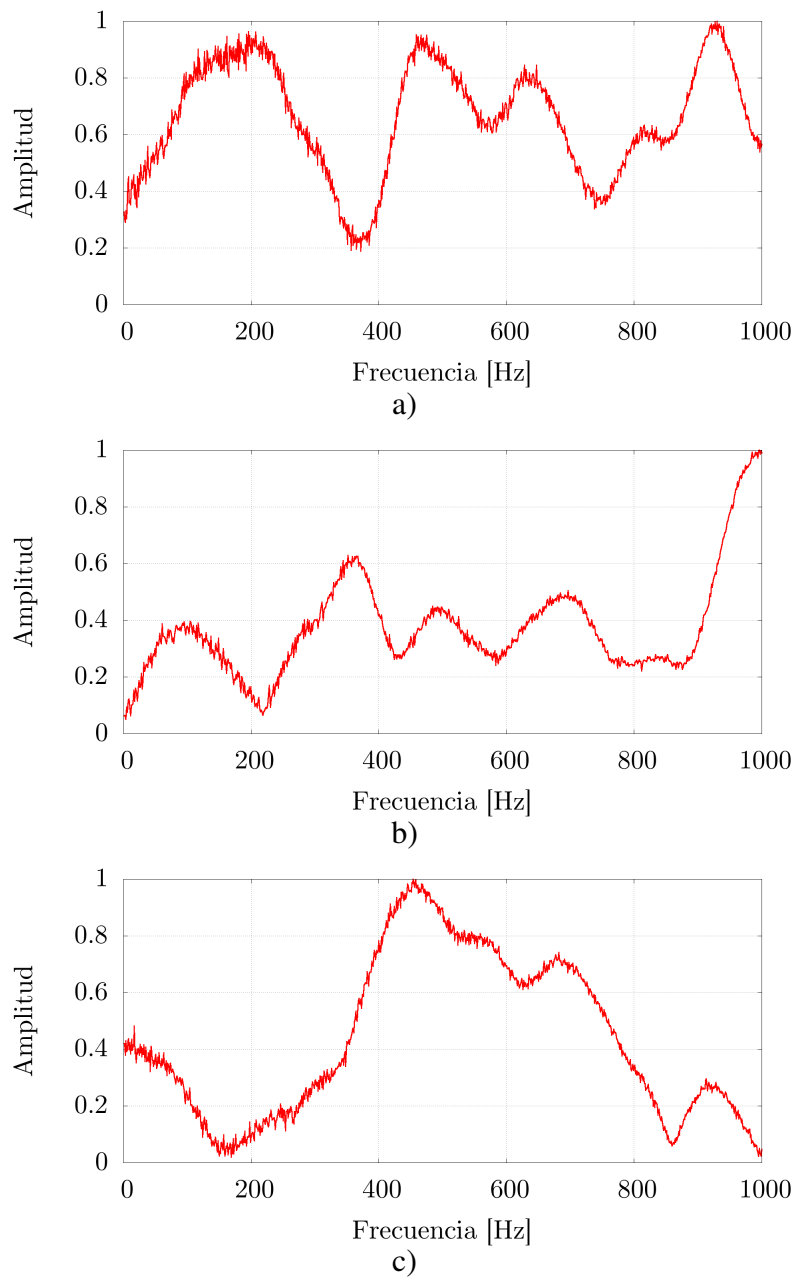


Figura 6.76: Espectro de frecuencias para la oscilación de la presión a la salida del inyector en la posición axial para las distintas simulaciones: a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

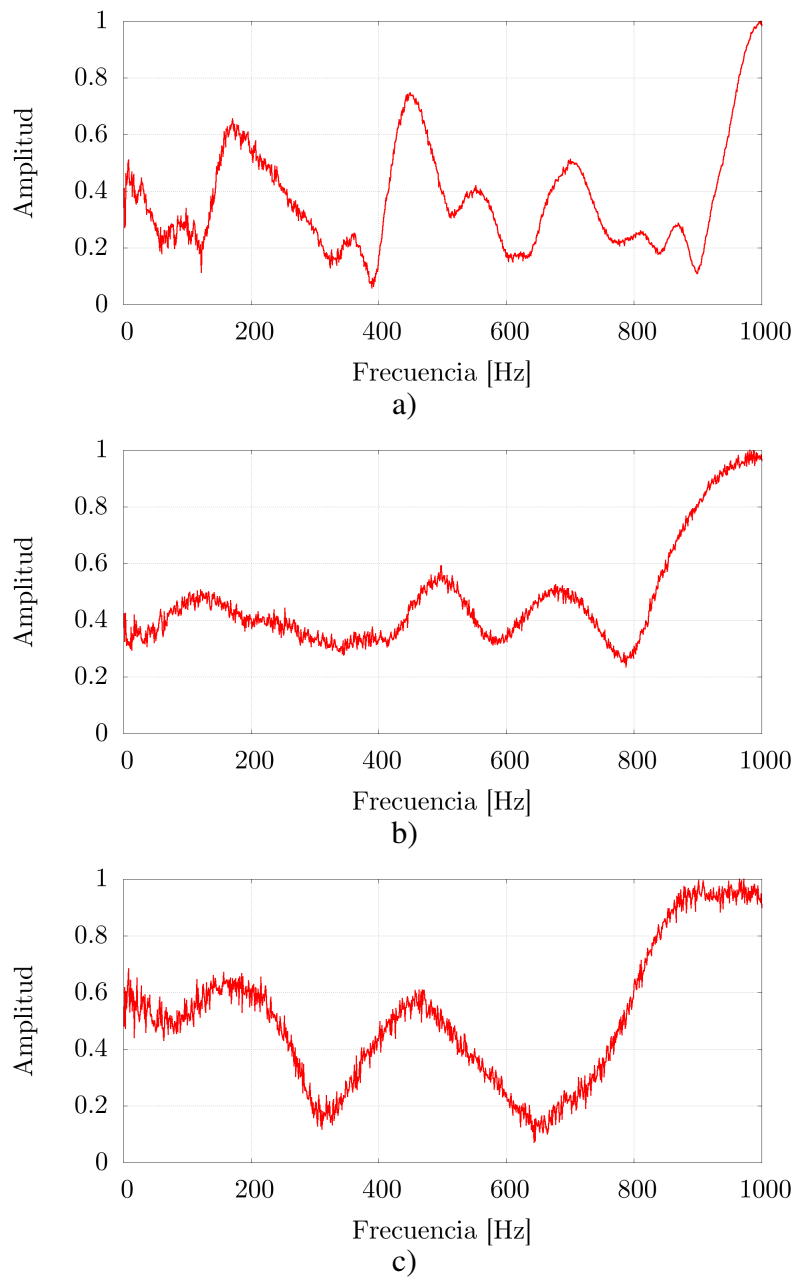


Figura 6.77: Espectro de frecuencias para la oscilación de la presión a la salida del inyector en la posición axial para las distintas simulaciones: a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$

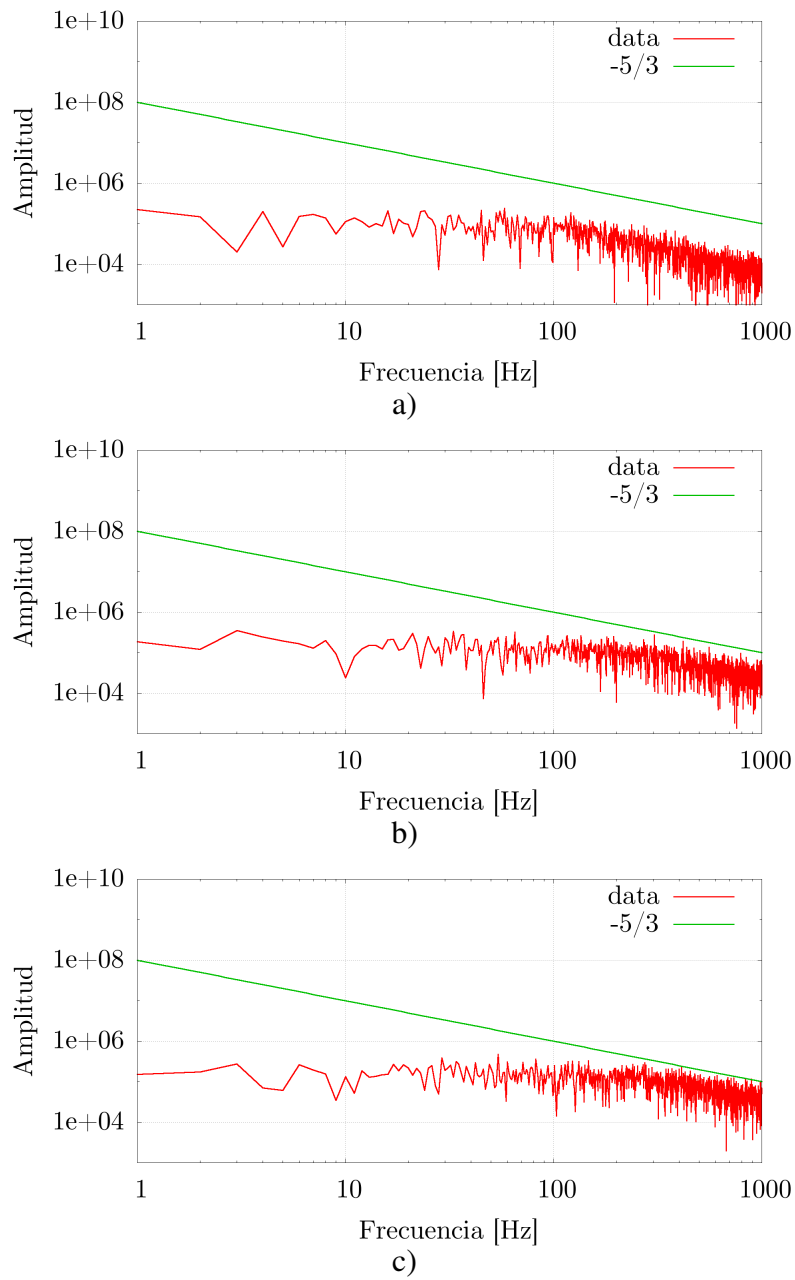


Figura 6.78: Espectro de energía del campo turbulento al interior de la cámara de mezclado para los distintos casos simulados: a) $Re_{petroleo} = 10000$, b) $Re_{petroleo} = 20000$ y c) $Re_{petroleo} = 25000$

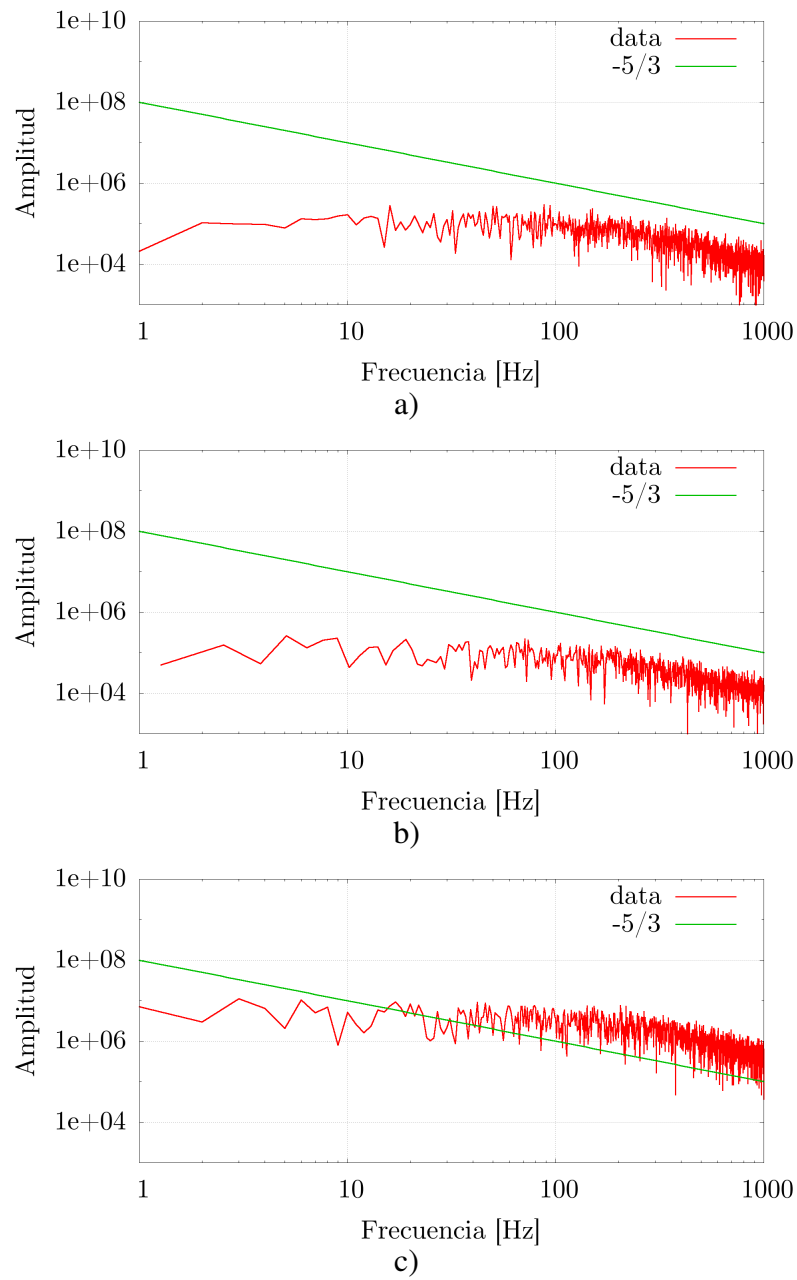


Figura 6.79: Espectro de energía del campo turbulento al interior de la cámara de mezclado para los distintos casos simulados: a) $Re_{agua} = 2000$, b) $Re_{agua} = 4000$ y c) $Re_{agua} = 5000$

6.5. Presión en la alimentación de agua

Al analizar el comportamiento de la presión dentro de la línea de alimentación del agua con respecto a la presión obtenida del caso de control, se puede observar un incremento con una tendencia cuadrática en el cual un incremento en el caudal del agua cercano a un 65 % genera un incremento de la presión cercano a 350 %. Para el caso en que se varía el caudal de petróleo, se observa el efecto contrario en el cual la presión experimentada en la línea de alimentación del agua disminuye a medida que se incrementa el caudal de petróleo, pero en este caso las variaciones experimentadas no son tan grandes si se comparan con las obtenidas en el caso previo.

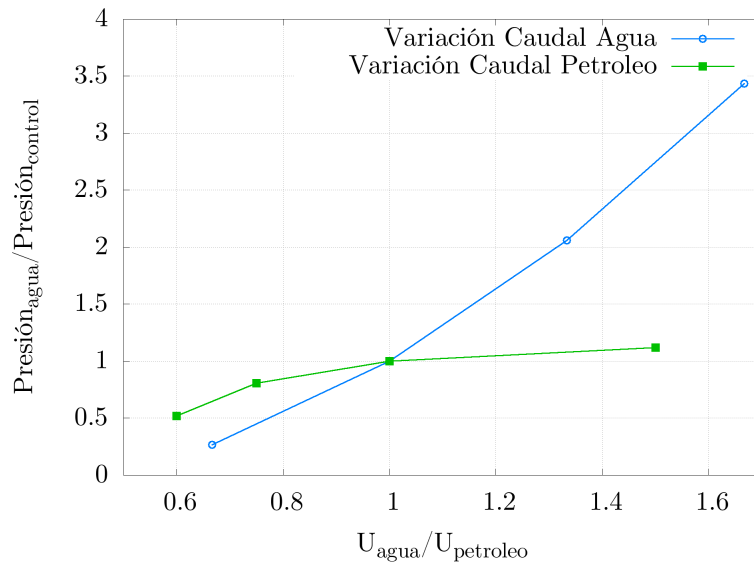


Figura 6.80: Variación de la presión en la línea de alimentación del agua con respecto al caso de control para distintas relaciones de velocidad entre las fases.

Capítulo 7

Conclusiones y trabajos futuros

A continuación se presentan las conclusiones obtenidas del trabajo realizado y se plantean las líneas de investigación a futuro para desarrollar en torno al mismo con tal de poder entender y comprender de mejor manera el fenómeno analizado.

En primera instancia se logró implementar y validar el código libre OpenFOAM que permita simular flujos turbulentos multifásicos incompresibles, con transporte de energía y fuerzas de flotación mediante la aproximación de Boussinesq.

Cuando se analizó el flujo sin presencia de agua se pudo observar la presencia de dos vórtices con vorticidad opuesta alineados axialmente dentro de la cámara de mezcla y un tercer vórtice anular ubicado en la base el cual se produce por el desprendimiento de la capa límite al ingreso de la cámara. Estos vórtices generan zonas de recirculación del fluido aguas arriba en el inyector generando zonas con un tiempo de residencia mayor y por consiguiente un mayor incremento en la temperatura.

La inyección de un caudal de agua en el inyector modifica tanto el perfil de velocidades de salida como la distribución de las temperaturas al interior del inyector. Cuando se agrega el caudal de agua al inyector se puede apreciar las mismas estructuras internas en el fluido variando únicamente la intensidad de los vórtices. Respecto a la composición de la mezcla se observó una predominancia de la fase de petróleo, con gotas de agua dispersas en él. La presencia de los vórtices axiales genera una zona de acumulación para el agua en la zona más distal al eje del inyector, debido a la diferencia entre las fuerzas centrífugas que experimentan ambas fases dado su diferencia en las densidades.

Al incrementar el caudal de petróleo que ingresa a la cámara de mezcla se observa un

incremento en la intensidad de los vórtices descritos previamente, un desplazamiento paulatino hacia las paredes del inyector y un incremento en el ángulo total abarcado por dichos vórtices. Se observó además una disminución en las temperaturas obtenidas producto de una disminución del orden del 50% en los tiempos de residencia del flujo. Respecto a la composición de la mezcla, la fracción volumétrica de petróleo promedio incrementa, decrece la dispersión y se genera una distribución más leptocúrtica.

Al incrementar el caudal de agua que ingresa a la cámara de mezcla, se observa un incremento en la intensidad de los vórtices descritos previamente. El vórtice anular ve incrementada su intensidad y aumenta su zona de influencia en dirección al eje del inyector. Los vórtices axiales sufren un desplazamiento aguas abajo de la cámara. Se observó además un incremento en las temperaturas obtenidas en la zona influenciada por el vórtice anular producto de un incremento en los tiempos de residencia del flujo, y un desplazamiento de la zona afectada térmicamente. Respecto a la composición de la mezcla, la fracción volumétrica de petróleo promedio disminuye, incrementa la dispersión y se genera una distribución menos leptocúrtica.

La inestabilidad presente en el quemador de la turbina a gas no puede ser explicada exclusivamente por las condiciones hidrodinámicas del problema por lo que se hace necesario extender la modelación para incluir el fenómeno de combustión y poder analizar el acoplamiento de las ondas termo-acústicas que provocan una combustión inestable.

La geometría del inyector permite que se genere un buen proceso de mezclado de las fases, generándose un flujo bastante homogéneo a la salida del inyector. La placa con perforaciones al interior de la tubería de petróleo hace que se genere una elevada velocidad en la zona más alejada del eje axial al interior del cono que inyecta el petróleo a la cámara de mezclado de la tubería de petróleo. Este perfil de velocidades con que posteriormente se inyecta el petróleo en dicha cámara, interactúa con la velocidad más baja con que hace ingreso el agua, generando una zona de mezclado altamente turbulenta facilitando este proceso. Debido a las dimensiones del inyector y los gradientes de temperatura alcanzados, el fenómeno de flotación es prácticamente despreciable en comparación con las fuerzas inerciales que se generan en el fluido y podrían no ser considerados en futuros estudios con el fin de acelerar y simplificar los tiempos de cálculo.

Lineas de trabajo futuro Respecto a consideraciones y mejoramientos del modelo y del caso analizado se proponen las siguientes líneas de investigación:

- Agregar el fenómeno de combustión de la mezcla en la cámara de combustión con el fin de poder simular y analizar el fenómeno de manera global y poder estudiar la inestabilidad de la combustión producida.
- Estudiar la influencia de la geometría del quemador en el comportamiento de la mezcla de los fluidos y el efecto de estos sobre la combustión inestable.
- Estudiar el proceso con el dominio completo de quemador.
- Implementación de un modelo de sub-malla para el transporte de la fracción volumétrica que permita estudiar con mas detalle la distribución y el tamaño de las micelas que se generan en la emulsión.
- Implementar un modelo para la transferencia de masa que permita considerar la posible evaporación de las gotas de agua presentes en la mezcla.
- Implementar un código compresible.
- Estudiar el comportamiento de los fluidos variando el combustible utilizado en el inyector.
- Estudiar la propagación de una perturbación acústica al interior de la mezcla.

Apéndice A

Código desarrollado

MTBoussinesqIFoam.C

```
1  /*-----*\
2  ===== |
3  \\      / F i e l d      | OpenFOAM: The Open Source
4  \\      / O p e r a t i o n |       CFD Toolbox
5  \\      / A n d          | Copyright (C) 2011–2014
6  \\ \    M a n i p u l a t i o n |       OpenFOAM Foundation
7  -----*
8  License
9      This file is part of OpenFOAM.
10
11     OpenFOAM is free software: you can redistribute it and/or
12     modify it under the terms of the GNU General Public
13     License as published by the Free Software Foundation,
14     either version 3 of the License, or (at your option)
15     any later version.
16
17     OpenFOAM is distributed in the hope that it will be
18     useful, but WITHOUT ANY WARRANTY; without even the implied
19     warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
20     PURPOSE. See the GNU General Public License for more
21     details.
22
23     You should have received a copy of the GNU General Public
24     License along with OpenFOAM. If not, see
25     <http://www.gnu.org/licenses/>.
26
27 Application
28     multiphaseInterFoam
29
30 Description
```


createFields.C

```
1 Info<< "Reading field p_rgh\n" << endl;
2 volScalarField p_rgh
3 (
4     IOobject
5     (
6         "p_rgh",
7         runTime.timeName(),
8         mesh,
9         IOobject::MUST_READ,
10        IOobject::AUTO_WRITE
11    ),
12    mesh
13);
14
15 Info<< "Reading field U\n" << endl;
16 volVectorField U
17 (
18     IOobject
19     (
20         "U",
21         runTime.timeName(),
22         mesh,
23         IOobject::MUST_READ,
24         IOobject::AUTO_WRITE
25     ),
26     mesh
27);
28
29 #include "createPhi.H"
30
31 multiphaseMixture mixture(U, phi);
32
33 // Need to store rho for ddt(rho, U)
34 volScalarField rho
35 (
36     IOobject
37     (
38         "rho",
39         runTime.timeName(),
40         mesh,
41         IOobject::READ_IF_PRESENT,
42         IOobject::AUTO_WRITE
43     ),
44     mixture.rho()
45);
46 rho.oldTime();
47
```



```
97         runTime.timeName() ,
98         mesh ,
99         IOobject::MUST_READ,
100        IOobject::AUTO_WRITE
101    ),
102    mixture.kappa()
103 );
104
105 Info<< "Reading field Pr\n" << endl;
106 volScalarField Pr
107 (
108     IOobject
109     (
110         "Pr" ,
111         runTime.timeName() ,
112         mesh ,
113         IOobject::MUST_READ,
114         IOobject::AUTO_WRITE
115     ),
116     mixture.Pr()
117 );
118
119
120
121 Info<< "Reading field Beta\n" << endl;
122 volScalarField Beta
123 (
124     IOobject
125     (
126         "Beta" ,
127         runTime.timeName() ,
128         mesh ,
129         IOobject::MUST_READ,
130         IOobject::AUTO_WRITE
131     ),
132     mixture.Beta()
133 );
134
135 Info<< "Reading transportProperties\n" << endl;
136
137 IOdictionary transportProperties
138 (
139     IOobject
140     (
141         "transportProperties" ,
142         runTime.constant() ,
143         mesh ,
144         IOobject::MUST_READ_IF_MODIFIED,
145         IOobject::NO_WRITE
```


TEqn.H

```
1 {
2     fvScalarMatrix TEqn
3     (
4         //fvm:: ddt(rhoCp, T)
5         fvm:: ddt(T)
6         //+ fvm:: div(rhoCpPhi, T)
7         + fvm:: div(phi, T)
8         - fvm:: laplacian(kappa/rhoCp, T)
9     );
10
11     TEqn.solve();
12
13     rhok = (1.0 - Beta*(T - Tref));
14 }
```

multiphaseMixture.H

```

1  /*-----*/
2  ===== |
3  \\      /  F i e l d      |  OpenFOAM: The Open Source CFD Toolbox
4  \\      /  O p e r a t i o n      |
5  \\      /  A n d      |  Copyright (C) 2011–2014 OpenFOAM Foundation
6  \\      /  M a n i p u l a t i o n      |
7  -----
8  License
9      This file is part of OpenFOAM.
10
11      OpenFOAM is free software: you can redistribute it and/or modify it
12      under the terms of the GNU General Public License as published by
13      the Free Software Foundation, either version 3 of the License, or
14      (at your option) any later version.
15
16      OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
17      ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
18      FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
19      for more details.
20
21      You should have received a copy of the GNU General Public License
22      along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.
23
24  Class
25      Foam::multiphaseMixture
26
27  Description
28      Incompressible multi-phase mixture with built in solution for the
29      phase fractions with interface compression for interface-capturing.
30
31      Derived from transportModel so that it can be used in conjunction with
32      the incompressible turbulence models.
33
34      Surface tension and contact-angle is handled for the interface
35      between each phase-pair.
36
37  SourceFiles
38      multiphaseMixture.C
39
40  /*-----*/
41
42  #ifndef multiphaseMixture_H
43  #define multiphaseMixture_H
44
45  #include "incompressible/transportModel/transportModel.H"
46  #include "IOdictionary.H"
47  #include "phase.H"

```

```

48 #include "PtrDictionary.H"
49 #include "volFields.H"
50 #include "surfaceFields.H"
51
52 // * * * * * //
53
54 namespace Foam
55 {
56
57 /*-----*\
58                      Class multiphaseMixture Declaration
59 \*-----*/
60
61 class multiphaseMixture
62 :
63     public IOdictionary ,
64     public transportModel
65 {
66 public:
67
68     class interfacePair
69     :
70         public Pair<word>
71     {
72     public:
73
74         class hash
75         :
76             public Hash<interfacePair >
77         {
78         public:
79
80             hash ()
81             {}
82
83             label operator()(const interfacePair& key) const
84             {
85                 return word::hash()(key.first()) + word::hash()(key.second());
86             }
87         };
88
89         // Constructors
90
91         interfacePair ()
92         {}
93
94         interfacePair(const word& alpha1Name , const word& alpha2Name)
95         :
96

```


multiphaseMixture.C

```

1  /*-----*/
2  ===== |
3  \\      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
4  \\      /  O p e r a t i o n      |
5  \\      /  A n d      | Copyright (C) 2011–2014 OpenFOAM Foundation
6  \\      /  M a n i p u l a t i o n      |
7  -----
8  License
9  This file is part of OpenFOAM.
10
11  OpenFOAM is free software: you can redistribute it and/or modify it
12  under the terms of the GNU General Public License as published by
13  the Free Software Foundation, either version 3 of the License, or
14  (at your option) any later version.
15
16  OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
17  ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
18  FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
19  for more details.
20
21  You should have received a copy of the GNU General Public License
22  along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.
23
24  /*-----*/
25
26  #include "multiphaseMixture.H"
27  #include "alphaContactAngleFvPatchScalarField.H"
28  #include "Time.H"
29  #include "subCycle.H"
30  #include "MULES.H"
31  #include "surfaceInterpolate.H"
32  #include "fvcGrad.H"
33  #include "fvcSnGrad.H"
34  #include "fvcDiv.H"
35  #include "fvcFlux.H"
36
37  // * * * * * Static Member Data * * * * * //
38
39  const Foam::scalar Foam::multiphaseMixture::convertToRad =
40  Foam::constant::mathematical::pi/180.0;
41
42
43  // * * * * * Private Member Functions * * * * * //
44
45  void Foam::multiphaseMixture::calcAlphas ()
46  {
47  scalar level = 0.0;

```

```

48     alphas_ == 0.0;
49
50     forAllIter (PtrDictionary<phase>, phases_, iter)
51     {
52         alphas_ += level*iter();
53         level += 1.0;
54     }
55
56     alphas_.correctBoundaryConditions();
57 }
58
59
60 // * * * * * Constructors * * * * * //
61
62 Foam::multiphaseMixture::multiphaseMixture
63 (
64     const volVectorField& U,
65     const surfaceScalarField& phi
66 )
67 :
68     IOdictionary
69     (
70         IOobject
71         (
72             "transportProperties",
73             U.time().constant(),
74             U.db(),
75             IOobject::MUST_READ_IF_MODIFIED,
76             IOobject::NO_WRITE
77         )
78     ),
79
80     phases_(lookup("phases"), phase::iNew(U, phi)),
81
82     mesh_(U.mesh()),
83     U_(U),
84     phi_(phi),
85
86     rhoPhi_
87     (
88         IOobject
89         (
90             "rhoPhi",
91             mesh_.time().timeName(),
92             mesh_,
93             IOobject::NO_READ,
94             IOobject::NO_WRITE
95         ),
96     mesh_,

```



```

146     (
147         IObject
148         (
149             "nu",
150             mesh_.time().timeName(),
151             mesh_
152         ),
153         mu()/rho()
154     ),
155
156     sigmas_(lookup("sigmas")),
157     dimSigma_(1, 0, -2, 0, 0),
158     deltaN_
159     (
160         "deltaN",
161         1e-8/pow(average(mesh_.V()), 1.0/3.0)
162     )
163 {
164     calcAlphas();
165     alphas_.write();
166 }
167
168 // * * * * * Member Functions * * * * * //
169
170
171 Foam::tmp<Foam::volScalarField>
172 Foam::multiphaseMixture::rho() const
173 {
174     PtrDictionary<phase>::const_iterator iter = phases_.begin();
175
176     tmp<volScalarField> trho = iter()*iter().rho();
177
178     for (++iter; iter != phases_.end(); ++iter)
179     {
180         trho() += iter()*iter().rho();
181     }
182
183     return trho;
184 }
185
186
187 Foam::tmp<Foam::scalarField>
188 Foam::multiphaseMixture::rho(const label patchi) const
189 {
190     PtrDictionary<phase>::const_iterator iter = phases_.begin();
191
192     tmp<scalarField> trho = iter().boundaryField()[patchi]*iter().rho().value();
193
194     for (++iter; iter != phases_.end(); ++iter)

```



```

244     }
245
246     return trhoCp;
247 }
248
249
250 Foam::tmp<Foam::scalarField>
251 Foam::multiphaseMixture::rhoCp(const label patchi) const
252 {
253     PtrDictionary<phase>::const_iterator iter = phases_.begin();
254
255     tmp<scalarField> trhoCp = iter().boundaryField()[patchi]*iter().rho().value()
256     *iter().Cp().value();
257
258     for (++iter; iter != phases_.end(); ++iter)
259     {
260         trhoCp += iter().boundaryField()[patchi]*iter().rho().value()
261         *iter().Cp().value();
262     }
263
264     return trhoCp;
265 }
266
267 Foam::tmp<Foam::volScalarField>
268 Foam::multiphaseMixture::kappa() const
269 {
270     PtrDictionary<phase>::const_iterator iter = phases_.begin();
271
272     tmp<volScalarField> tkappa = iter()*iter().nu()*iter().rho()*iter().Cp()
273     /iter().Pr();
274
275     for (++iter; iter != phases_.end(); ++iter)
276     {
277         tkappa += iter()*iter().nu()*iter().rho()*iter().Cp()/iter().Pr();
278     }
279
280     return tkappa;
281 }
282
283
284 Foam::tmp<Foam::scalarField>
285 Foam::multiphaseMixture::kappa(const label patchi) const
286 {
287     PtrDictionary<phase>::const_iterator iter = phases_.begin();
288
289     tmp<scalarField> tkappa = iter().boundaryField()[patchi]*iter().nu(patchi)
290     *iter().rho().value()*iter().Cp().value()/iter().Pr().value();
291
292     for (++iter; iter != phases_.end(); ++iter)

```

```

293     {
294         tkappa() += iter().boundaryField()[patchi]*iter().nu(patchi)
295         *iter().rho().value()*iter().Cp().value()/iter().Pr().value();
296     }
297
298     return tkappa;
299 }
300
301 Foam::tmp<Foam::volScalarField>
302 Foam::multiphaseMixture::Pr(const) const
303 {
304     PtrDictionary<phase>::const_iterator iter = phases_.begin();
305
306     tmp<volScalarField> tPr = iter()*iter().Pr();
307
308     for (++iter; iter != phases_.end(); ++iter)
309     {
310         tPr() += iter()*iter().Pr();
311     }
312
313     return tPr;
314 }
315
316
317 Foam::tmp<Foam::scalarField>
318 Foam::multiphaseMixture::Pr(const label patchi) const
319 {
320     PtrDictionary<phase>::const_iterator iter = phases_.begin();
321
322     tmp<scalarField> tPr = iter().boundaryField()[patchi]*iter().Pr().value();
323
324     for (++iter; iter != phases_.end(); ++iter)
325     {
326         tPr() += iter().boundaryField()[patchi]*iter().Pr().value();
327     }
328
329     return tPr;
330 }
331
332 Foam::tmp<Foam::volScalarField>
333 Foam::multiphaseMixture::Beta(const) const
334 {
335     PtrDictionary<phase>::const_iterator iter = phases_.begin();
336
337     tmp<volScalarField> tBeta = iter()*iter().Beta();
338
339     for (++iter; iter != phases_.end(); ++iter)
340     {
341         tBeta() += iter()*iter().Beta();

```



```

440 Foam::tmp<Foam::volScalarField>
441 Foam::multiphaseMixture::nu() const
442 {
443     return nu_;
444 }
445
446
447 Foam::tmp<Foam::scalarField>
448 Foam::multiphaseMixture::nu(const label patchi) const
449 {
450     return nu_.boundaryField()[patchi];
451 }
452
453
454 Foam::tmp<Foam::surfaceScalarField>
455 Foam::multiphaseMixture::muf() const
456 {
457     return muf()/fvc::interpolate(rho());
458 }
459
460
461 Foam::tmp<Foam::surfaceScalarField>
462 Foam::multiphaseMixture::surfaceTensionForce() const
463 {
464     tmp<surfaceScalarField> tstf
465     (
466         new surfaceScalarField
467         (
468             IOobject
469             (
470                 "surfaceTensionForce",
471                 mesh_.time().timeName(),
472                 mesh_
473             ),
474             mesh_,
475             dimensionedScalar
476             (
477                 "surfaceTensionForce",
478                 dimensionSet(1, -2, -2, 0, 0),
479                 0.0
480             )
481         )
482     );
483
484     surfaceScalarField& stf = tstf();
485
486     forAllConstIter(PtrDictionary<phase>, phases_, iter1)
487     {
488         const phase& alpha = iter1();

```

```

489
490     PtrDictionary<phase>::const_iterator iter2 = iter1;
491     ++iter2;
492
493     for (; iter2 != phases_.end(); ++iter2)
494     {
495         const phase& alpha2 = iter2();
496
497         sigmaTable::const_iterator sigma =
498             sigmas_.find(interfacePair(alpha1, alpha2));
499
500         if (sigma == sigmas_.end())
501         {
502             FatalErrorIn("multiphaseMixture::surfaceTensionForce() const")
503                 << "Cannot find interface " << interfacePair(alpha1, alpha2)
504                 << " in list of sigma values"
505                 << exit(FatalError);
506         }
507
508         stf += dimensionedScalar("sigma", dimSigma_, sigma())
509             *fvc::interpolate(K(alpha1, alpha2))*
510             (
511                 fvc::interpolate(alpha2)*fvc::snGrad(alpha1)
512                 - fvc::interpolate(alpha1)*fvc::snGrad(alpha2)
513             );
514     }
515 }
516
517 return tstf;
518 }
519
520
521 void Foam::multiphaseMixture::solve()
522 {
523     correct();
524
525     const Time& runTime = mesh_.time();
526
527     volScalarField& alpha = phases_.first();
528
529     const dictionary& alphaControls = mesh_.solverDict("alpha");
530     label nAlphaSubCycles(readLabel(alphaControls.lookup("nAlphaSubCycles")));
531     scalar cAlpha(readScalar(alphaControls.lookup("cAlpha")));
532
533     if (nAlphaSubCycles > 1)
534     {
535         surfaceScalarField rhoPhiSum
536         (
537             IOobject

```



```

587     }
588
589
590     void Foam::multiphaseMixture::correct()
591     {
592         forAllIter (PtrDictionary<phase>, phases_, iter)
593         {
594             iter().correct();
595         }
596     }
597
598
599     Foam::tmp<Foam::surfaceVectorField> Foam::multiphaseMixture::nHatf
600     (
601         const volScalarField& alpha1,
602         const volScalarField& alpha2
603     ) const
604     {
605         /*
606          // Cell gradient of alpha
607          volVectorField gradAlpha =
608              alpha2*fvc::grad(alpha1) - alpha1*fvc::grad(alpha2);
609
610          // Interpolated face-gradient of alpha
611          surfaceVectorField gradAlphaf = fvc::interpolate(gradAlpha);
612          */
613
614         surfaceVectorField gradAlphaf
615         (
616             fvc::interpolate(alpha2)*fvc::interpolate(fvc::grad(alpha1))
617             - fvc::interpolate(alpha1)*fvc::interpolate(fvc::grad(alpha2))
618         );
619
620         // Face unit interface normal
621         return gradAlphaf/(mag(gradAlphaf) + deltaN_);
622     }
623
624
625     Foam::tmp<Foam::surfaceScalarField> Foam::multiphaseMixture::nHatf
626     (
627         const volScalarField& alpha1,
628         const volScalarField& alpha2
629     ) const
630     {
631         // Face unit interface normal flux
632         return nHatfv(alpha1, alpha2) & mesh_.Sf();
633     }
634
635

```

```

636 // Correction for the boundary condition on the unit normal nHat on
637 // walls to produce the correct contact angle.
638
639 // The dynamic contact angle is calculated from the component of the
640 // velocity on the direction of the interface, parallel to the wall.
641
642 void Foam::multiphaseMixture::correctContactAngle
643 (
644     const phase& alpha1,
645     const phase& alpha2,
646     surfaceVectorField::GeometricBoundaryField& nHatb
647 ) const
648 {
649     const volScalarField::GeometricBoundaryField& gbf
650         = alpha1.boundaryField();
651
652     const fvBoundaryMesh& boundary = mesh_.boundary();
653
654     forAll(boundary, patchi)
655     {
656         if (isA<alphaContactAngleFvPatchScalarField>(gbf[patchi]))
657         {
658             const alphaContactAngleFvPatchScalarField& acap =
659                 refCast<const alphaContactAngleFvPatchScalarField>(gbf[patchi]);
660
661             vectorField& nHatPatch = nHatb[patchi];
662
663             vectorField AfHatPatch
664             (
665                 mesh_.Sf().boundaryField()[patchi]
666                 /mesh_.magSf().boundaryField()[patchi]
667             );
668
669             alphaContactAngleFvPatchScalarField::thetaPropsTable::
670                 const_iterator tp =
671                 acap.thetaProps().find(interfacePair(alpha1, alpha2));
672
673             if (tp == acap.thetaProps().end())
674             {
675                 FatalErrorIn
676                 (
677                     "multiphaseMixture::correctContactAngle"
678                     "(const phase& alpha1, const phase& alpha2, "
679                     "fvPatchVectorFieldField& nHatb) const"
680                 ) << "Cannot find interface " << interfacePair(alpha1, alpha2)
681                 << "\n    in table of theta properties for patch "
682                 << acap.patch().name()
683                 << exit(FatalError);
684             }
685         }
686     }
687 }

```

```

685
686     bool matched = (tp.key().first() == alpha1.name());
687
688     scalar theta0 = convertToRad*tp().theta0(matched);
689     scalarField theta(boundary[patchi].size(), theta0);
690
691     scalar uTheta = tp().uTheta();
692
693     // Calculate the dynamic contact angle if required
694     if (uTheta > SMALL)
695     {
696         scalar thetaA = convertToRad*tp().thetaA(matched);
697         scalar thetaR = convertToRad*tp().thetaR(matched);
698
699         // Calculated the component of the velocity parallel to the wall
700         vectorField Uwall
701         (
702             U_.boundaryField()[patchi].patchInternalField()
703             - U_.boundaryField()[patchi]
704         );
705         Uwall -= (AfHatPatch & Uwall)*AfHatPatch;
706
707         // Find the direction of the interface parallel to the wall
708         vectorField nWall
709         (
710             nHatPatch - (AfHatPatch & nHatPatch)*AfHatPatch
711         );
712
713         // Normalise nWall
714         nWall /= (mag(nWall) + SMALL);
715
716         // Calculate Uwall resolved normal to the interface parallel to
717         // the interface
718         scalarField uwall(nWall & Uwall);
719
720         theta += (thetaA - thetaR)*tanh(uwall/uTheta);
721     }
722
723
724     // Reset nHatPatch to correspond to the contact angle
725
726     scalarField a12(nHatPatch & AfHatPatch);
727
728     scalarField b1(cos(theta));
729
730     scalarField b2(nHatPatch.size());
731
732     forAll(b2, facei)
733     {

```

```

734         b2[facei] = cos(acos(a12[facei]) - theta[facei]);
735     }
736
737     scalarField det(1.0 - a12*a12);
738
739     scalarField a((b1 - a12*b2)/det);
740     scalarField b((b2 - a12*b1)/det);
741
742     nHatPatch = a*AfHatPatch + b*nHatPatch;
743
744     nHatPatch /= (mag(nHatPatch) + deltaN_.value());
745 }
746 }
747 }
748
749
750 Foam::tmp<Foam::volScalarField> Foam::multiphaseMixture::K
751 (
752     const phase& alpha1,
753     const phase& alpha2
754 ) const
755 {
756     tmp<surfaceVectorField> tnHatfv = nHatfv(alpha1, alpha2);
757
758     correctContactAngle(alpha1, alpha2, tnHatfv().boundaryField());
759
760     // Simple expression for curvature
761     return -fvc::div(tnHatfv & mesh_.Sf());
762 }
763
764
765 Foam::tmp<Foam::volScalarField>
766 Foam::multiphaseMixture::nearInterface() const
767 {
768     tmp<volScalarField> tnearInt
769     (
770         new volScalarField
771         (
772             IOobject
773             (
774                 "nearInterface",
775                 mesh_.time().timeName(),
776                 mesh_
777             ),
778             mesh_,
779             dimensionedScalar("nearInterface", dimless, 0.0)
780         )
781     );
782

```

```

783     forAllConstIter (PtrDictionary<phase>, phases_, iter)
784     {
785         tnearInt () = max (tnearInt (), pos (iter () - 0.01)*pos (0.99 - iter ()));
786     }
787
788     return tnearInt;
789 }
790
791
792 void Foam::multiphaseMixture::solveAlphas
793 (
794     const scalar cAlpha
795 )
796 {
797     static label nSolves=-1;
798     nSolves++;
799
800     word alphaScheme ("div(phi, alpha)");
801     word alphasScheme ("div(phirb, alpha)");
802
803     surfaceScalarField phic (mag (phi_ / mesh_.magSf ());
804     phic = min (cAlpha*phic, max (phic));
805
806     PtrList<surfaceScalarField> phiAlphaCorrs (phases_.size ());
807     int phasei = 0;
808
809     forAllIter (PtrDictionary<phase>, phases_, iter)
810     {
811         phase& alpha = iter ();
812
813         phiAlphaCorrs.set
814         (
815             phasei,
816             new surfaceScalarField
817             (
818                 "phi" + alpha.name () + "Corr",
819                 fvc::flux
820                 (
821                     phi_,
822                     alpha,
823                     alphaScheme
824                 )
825             )
826         );
827
828         surfaceScalarField& phiAlphaCorr = phiAlphaCorrs [phasei];
829
830         forAllIter (PtrDictionary<phase>, phases_, iter2)
831         {

```



```
930     bool readOK = true;
931
932     PtrList<entry> phaseData(lookup("phases"));
933     label phasei = 0;
934
935     forAllIter(PtrDictionary<phase>, phases_, iter)
936     {
937         readOK &= iter().read(phaseData[phasei++].dict());
938     }
939
940     lookup("sigmas") >> sigmas_;
941
942     return readOK;
943 }
944 else
945 {
946     return false;
947 }
948 }
949
950
951 // ***** //
```

phase.H

```

1  /*-----*/
2  ===== |
3  \\      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
4  \\      /  O p e r a t i o n      |
5  \\      /  A n d      | Copyright (C) 2011 OpenFOAM Foundation
6  \\      /  M a n i p u l a t i o n      |
7  -----
8  License
9      This file is part of OpenFOAM.
10
11      OpenFOAM is free software: you can redistribute it and/or modify it
12      under the terms of the GNU General Public License as published by
13      the Free Software Foundation, either version 3 of the License, or
14      (at your option) any later version.
15
16      OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
17      ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
18      FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
19      for more details.
20
21      You should have received a copy of the GNU General Public License
22      along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.
23
24  Class
25      Foam::phase
26
27  Description
28      Single incompressible phase derived from the phase-fraction.
29      Used as part of the multiPhaseMixture for interface-capturing multi-phase
30      simulations.
31
32  SourceFiles
33      phase.C
34
35  /*-----*/
36
37  #ifndef phase_H
38  #define phase_H
39
40  #include "volFields.H"
41  #include "dictionaryEntry.H"
42  #include "incompressible/viscosityModels/viscosityModel/viscosityModel.H"
43
44  // * * * * *
45
46  namespace Foam
47  {

```



```

97         const surfaceScalarField& phi
98     )
99     :
100     U_(U) ,
101     phi_(phi)
102     {}
103
104     autoPtr<phase> operator()(Istream& is) const
105     {
106         dictionaryEntry ent(dictionary::null, is);
107         return autoPtr<phase>(new phase(ent.keyword(), ent, U_, phi_));
108     }
109 };
110
111
112 // Member Functions
113
114 const word& name() const
115 {
116     return name_;
117 }
118
119 const word& keyword() const
120 {
121     return name();
122 }
123
124 // Return const-access to phase1 viscosityModel
125 const viscosityModel& nuModel() const
126 {
127     return nuModel_();
128 }
129
130 // Return the kinematic laminar viscosity
131 tmp<volScalarField> nu() const
132 {
133     return nuModel_>nu();
134 }
135
136 // Return the laminar viscosity for patch
137 tmp<scalarField> nu(const label patchi) const
138 {
139     return nuModel_>nu(patchi);
140 }
141
142 // Return const-access to phase1 density
143 const dimensionedScalar& rho() const
144 {
145     return rho_;

```


phase.C

```

1  /*-----*/
2  ===== |
3  \\      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
4  \\      /  O p e r a t i o n      |
5  \\      /  A n d      | Copyright (C) 2011–2013 OpenFOAM Foundation
6  \\      /  M a n i p u l a t i o n      |
7  -----
8  License
9      This file is part of OpenFOAM.
10
11      OpenFOAM is free software: you can redistribute it and/or modify it
12      under the terms of the GNU General Public License as published by
13      the Free Software Foundation, either version 3 of the License, or
14      (at your option) any later version.
15
16      OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
17      ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
18      FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
19      for more details.
20
21      You should have received a copy of the GNU General Public License
22      along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.
23
24  /*-----*/
25
26  #include "phase.H"
27
28  // * * * * * Constructors * * * * * //
29
30  Foam::phase::phase
31  (
32      const word& phaseName,
33      const dictionary& phaseDict,
34      const volVectorField& U,
35      const surfaceScalarField& phi
36  )
37  :
38      volScalarField
39      (
40          IOobject
41          (
42              IOobject::groupName("alpha", phaseName),
43              U.mesh().time().timeName(),
44              U.mesh(),
45              IOobject::MUST_READ,
46              IOobject::AUTO_WRITE
47          ),

```


alphaContactAngleFvPatchScalarField.H

```

1  /*-----*/
2  ===== |
3  \\      /  F i e l d      |  OpenFOAM: The Open Source CFD Toolbox
4  \\      /  O p e r a t i o n      |
5  \\      /  A n d      |  Copyright (C) 2011 OpenFOAM Foundation
6  \\      /  M a n i p u l a t i o n      |
7  -----
8  License
9      This file is part of OpenFOAM.
10
11      OpenFOAM is free software: you can redistribute it and/or modify it
12      under the terms of the GNU General Public License as published by
13      the Free Software Foundation, either version 3 of the License, or
14      (at your option) any later version.
15
16      OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
17      ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
18      FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
19      for more details.
20
21      You should have received a copy of the GNU General Public License
22      along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.
23
24  Class
25      Foam::alphaContactAngleFvPatchScalarField
26
27  Description
28      Contact-angle boundary condition for multi-phase interface-capturing
29      simulations. Used in conjunction with multiphaseMixture.
30
31  SourceFiles
32      alphaContactAngleFvPatchScalarField.C
33
34  /*-----*/
35
36  #ifndef alphaContactAngleFvPatchScalarField_H
37  #define alphaContactAngleFvPatchScalarField_H
38
39  #include "zeroGradientFvPatchFields.H"
40  #include "multiphaseMixture.H"
41
42  // * * * * * //
43
44  namespace Foam
45  {
46
47  /*-----*/

```

```

48                                     Class alphaContactAngleFvPatch Declaration
49  /*-----*/
50
51  class alphaContactAngleFvPatchScalarField
52  :
53      public zeroGradientFvPatchScalarField
54  {
55      public:
56
57          class interfaceThetaProps
58          {
59              //-- Equilibrium contact angle
60              scalar theta0_;
61
62              //-- Dynamic contact angle velocity scale
63              scalar uTheta_;
64
65              //-- Limiting advancing contact angle
66              scalar thetaA_;
67
68              //-- Limiting receding contact angle
69              scalar thetaR_;
70
71
72      public:
73
74          // Constructors
75          interfaceThetaProps ()
76          {}
77
78          interfaceThetaProps (Istream&);
79
80
81          // Member functions
82
83          //-- Return the equilibrium contact angle theta0
84          scalar theta0 (bool matched=true) const
85          {
86              if (matched) return theta0_;
87              else return 180.0 - theta0_;
88          }
89
90          //-- Return the dynamic contact angle velocity scale
91          scalar uTheta () const
92          {
93              return uTheta_;
94          }
95
96          //-- Return the limiting advancing contact angle

```

```

97         scalar thetaA (bool matched=true) const
98     {
99         if (matched) return thetaA_;
100        else return 180.0 - thetaA_;
101    }
102
103    //- Return the limiting receding contact angle
104    scalar thetaR (bool matched=true) const
105    {
106        if (matched) return thetaR_;
107        else return 180.0 - thetaR_;
108    }
109
110
111    // IO functions
112
113    friend Istream& operator >>(Istream&, interfaceThetaProps&);
114    friend Ostream& operator <<(Ostream&, const interfaceThetaProps&);
115 };
116
117 typedef HashTable
118 <
119     interfaceThetaProps ,
120     multiphaseMixture::interfacePair ,
121     multiphaseMixture::interfacePair::hash
122 > thetaPropsTable ;
123
124
125 private :
126
127     // Private data
128
129     thetaPropsTable thetaProps_ ;
130
131
132 public :
133
134     //- Runtime type information
135     TypeName("alphaContactAngle");
136
137
138     // Constructors
139
140     //- Construct from patch and internal field
141     alphaContactAngleFvPatchScalarField
142     (
143         const fvPatch&,
144         const DimensionedField<scalar , volMesh>&
145     );

```

```

146
147     //– Construct from patch, internal field and dictionary
148     alphaContactAngleFvPatchScalarField
149     (
150         const fvPatch&,
151         const DimensionedField<scalar , volMesh>&,
152         const dictionary&
153     );
154
155     //– Construct by mapping given alphaContactAngleFvPatchScalarField
156     // onto a new patch
157     alphaContactAngleFvPatchScalarField
158     (
159         const alphaContactAngleFvPatchScalarField&,
160         const fvPatch&,
161         const DimensionedField<scalar , volMesh>&,
162         const fvPatchFieldMapper&
163     );
164
165     //– Construct and return a clone
166     virtual tmp<fvPatchScalarField> clone() const
167     {
168         return tmp<fvPatchScalarField>
169         (
170             new alphaContactAngleFvPatchScalarField(*this)
171         );
172     }
173
174     //– Construct as copy setting internal field reference
175     alphaContactAngleFvPatchScalarField
176     (
177         const alphaContactAngleFvPatchScalarField&,
178         const DimensionedField<scalar , volMesh>&
179     );
180
181     //– Construct and return a clone setting internal field reference
182     virtual tmp<fvPatchScalarField> clone
183     (
184         const DimensionedField<scalar , volMesh>& iF
185     ) const
186     {
187         return tmp<fvPatchScalarField>
188         (
189             new alphaContactAngleFvPatchScalarField(*this , iF)
190         );
191     }
192
193
194     // Member functions

```

```
195
196     //– Return the contact angle properties
197     const thetaPropsTable& thetaProps() const
198     {
199         return thetaProps_;
200     }
201
202     //– Write
203     virtual void write(Ostream&) const;
204 };
205
206
207 // * * * * * //
208
209 } // End namespace Foam
210
211 // * * * * * //
212
213 #endif
214
215 // * * * * * //
```

alphaContactAngleFvPatchScalarField.C

```

1  /*-----*/
2  ===== |
3  \\      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
4  \\      /  O p e r a t i o n      |
5  \\      /  A n d      | Copyright (C) 2011 OpenFOAM Foundation
6  \\      /  M a n i p u l a t i o n      |
7  -----
8  License
9      This file is part of OpenFOAM.
10
11      OpenFOAM is free software: you can redistribute it and/or modify it
12      under the terms of the GNU General Public License as published by
13      the Free Software Foundation, either version 3 of the License, or
14      (at your option) any later version.
15
16      OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
17      ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
18      FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
19      for more details.
20
21      You should have received a copy of the GNU General Public License
22      along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.
23
24  /*-----*/
25
26  #include "alphaContactAngleFvPatchScalarField.H"
27  #include "addToRunTimeSelectionTable.H"
28  #include "fvPatchFieldMapper.H"
29
30  // *****
31
32  namespace Foam
33  {
34
35  alphaContactAngleFvPatchScalarField::interfaceThetaProps::interfaceThetaProps
36  (
37      Istream& is
38  )
39  :
40      theta0_( readScalar( is ) ),
41      uTheta_( readScalar( is ) ),
42      thetaA_( readScalar( is ) ),
43      thetaR_( readScalar( is ) )
44  {}
45
46  Istream& operator >>

```

```

48 (
49     Istream& is ,
50     alphaContactAngleFvPatchScalarField::interfaceThetaProps& tp
51 )
52 {
53     is >> tp.theta0_ >> tp.uTheta_ >> tp.thetaA_ >> tp.thetaR_;
54     return is;
55 }
56
57
58 Ostream& operator <<
59 (
60     Ostream& os ,
61     const alphaContactAngleFvPatchScalarField::interfaceThetaProps& tp
62 )
63 {
64     os << tp.theta0_ << token::SPACE
65         << tp.uTheta_ << token::SPACE
66         << tp.thetaA_ << token::SPACE
67         << tp.thetaR_;
68
69     return os;
70 }
71
72
73 // * * * * * Constructors * * * * * //
74
75 alphaContactAngleFvPatchScalarField::alphaContactAngleFvPatchScalarField
76 (
77     const fvPatch& p ,
78     const DimensionedField<scalar , volMesh>& iF
79 )
80 :
81     zeroGradientFvPatchScalarField(p, iF)
82 {}
83
84
85 alphaContactAngleFvPatchScalarField::alphaContactAngleFvPatchScalarField
86 (
87     const alphaContactAngleFvPatchScalarField& gcpsf ,
88     const fvPatch& p ,
89     const DimensionedField<scalar , volMesh>& iF ,
90     const fvPatchFieldMapper& mapper
91 )
92 :
93     zeroGradientFvPatchScalarField(gcpsf, p, iF, mapper),
94     thetaProps_(gcpsf.thetaProps_)
95 {}
96

```

```

97
98 alphaContactAngleFvPatchScalarField::alphaContactAngleFvPatchScalarField
99 (
100     const fvPatch& p,
101     const DimensionedField<scalar, volMesh>& iF,
102     const dictionary& dict
103 )
104 :
105     zeroGradientFvPatchScalarField(p, iF),
106     thetaProps_(dict.lookup("thetaProperties"))
107 {
108     evaluate();
109 }
110
111
112 alphaContactAngleFvPatchScalarField::alphaContactAngleFvPatchScalarField
113 (
114     const alphaContactAngleFvPatchScalarField& gcpsf,
115     const DimensionedField<scalar, volMesh>& iF
116 )
117 :
118     zeroGradientFvPatchScalarField(gcpsf, iF),
119     thetaProps_(gcpsf.thetaProps_)
120 {}
121
122
123 // * * * * * Member Functions * * * * * //
124
125 void alphaContactAngleFvPatchScalarField::write(Ostream& os) const
126 {
127     fvPatchScalarField::write(os);
128     os.writeKeyword("thetaProperties")
129         << thetaProps_ << token::END_STATEMENT << nl;
130     writeEntry("value", os);
131 }
132
133
134 // * * * * * //
135
136 makePatchTypeField
137 (
138     fvPatchScalarField,
139     alphaContactAngleFvPatchScalarField
140 );
141
142 // * * * * * //
143 } // End namespace Foam
144 // * * * * * //

```

Apéndice B

Malla quemador tridimensional

```
1 //#####
2 //#####
3 //#####
4 //          Malla 3D con simetria axial
5 //#####
6 //#####
7 //#####
8
9
10 //#####
11 //      Variables auxiliares
12 a=1; // Estructurado de la malla
13 //a=0.0001;
14 // Seleccionar unidades de distancia (Descomentar valor
15 //                               que corresponda)
16 //      milimetros
17      x_to_m=0.001;
18 //      Centimetros
19 //      x_to_m=0.01;
20 //      Metros
21 //      x_to_m=1;
22 //
23 pi=3.14159265358979323846;
24 //#####
25
26
27 //#####
28 //#####
29 //          1.— Base tubería de petróleo
30 //#####
31 //#####
```

```

32
33 //a=0.0001;
34 //a=1;
35 //#####
36 // 1.1- Parametros Geometricos
37 //#####
38 rpetroleo=7.5*x_to_m;
39 lpetroleo=20*x_to_m;
40
41 //#####
42 // 1.2- Parametros Geometricos Auxiliares
43 //#####
44 rauxint=rpetroleo/2;
45
46 //#####
47 // 1.3- Geometria Basica
48 //#####
49 Point(0)={0,0,0};
50 Point(1)={rpetroleo/2,0,0};
51 Point(2)={rauxint*cos(pi/4),rauxint*sin(pi/4),0};
52 Point(3)={0,rauxint,0};
53 Point(4)={rpetroleo,0,0};
54 Point(5)={rpetroleo*cos(pi/4),rpetroleo*sin(pi/4),0};
55 Point(6)={0,rpetroleo,0};
56 Line(1) = {0, 1};
57 Line(2) = {0, 3};
58 Line(3) = {1, 2};
59 Line(4) = {2, 3};
60 Line(5) = {1, 4};
61 Line(6) = {2, 5};
62 Line(7) = {3, 6};
63 Circle(8) = {4, 0, 5};
64 Circle(9) = {5, 0, 6};
65 Line Loop(10) = {1, 3, 4, -2};
66 Ruled Surface(11) = {10};
67 Line Loop(12) = {5, 8, -6, -3};
68 Ruled Surface(13) = {12};
69 Line Loop(14) = {9, -7, -4, 6};
70 Ruled Surface(15) = {14};
71
72 //#####
73 // 1.4- Estructurado
74 //#####
75
76 n_vol_theta_petroleo=10;
77 n_vol_radial_petroleo=10;
78 Transfinite Line {1, 2, 3, 4, 8, 9} = n_vol_theta_petroleo
79 Using Progression 1;
80 Transfinite Line {5, 6, 7} = n_vol_radial_petroleo Using

```

```

81         Progression 1;
82         Transfinite Surface {11} = {0, 1, 2, 3};
83         Transfinite Surface {13} = {1, 4, 5, 2};
84         Transfinite Surface {15} = {2, 5, 6, 3};
85         Recombine Surface {11,13,15};
86
87         //#####
88         // 1.5- Extrucion
89         //#####
90
91         n_vol_largo_petroleo=35;
92         Extrude {0, 0, lpetroleo*a} {
93             Surface {11, 13, 15};
94             Layers {n_vol_largo_petroleo};
95             Recombine;
96         }
97
98
99         //#####
100        //#####
101        //          2.- Perforacion Tuberia Petroleo
102        //#####
103        //#####
104
105        //a=0.0001;
106        //a=1;
107        //#####
108        // 2.1- Parametros Geometricos
109        //#####
110        rperfpetroleo=1.5*x_to_m;
111        rposperfpetroleo=4.5*x_to_m;
112        lperforacion=3*x_to_m;
113
114        //#####
115        // 2.2- Parametros Geometricos Auxiliares
116        //#####
117        rauxperf=rperfpetroleo /2.5;
118
119        //#####
120        // 2.3- Geometria Basica
121        //#####
122        Point (27)={ rposperfpetroleo*Cos(pi/4) , rposperfpetroleo
123            *Sin(pi/4) , lpetroleo };
124        Point (28)={ rposperfpetroleo*Cos(pi/4)+rauxperf
125            *Cos(pi/4) , rposperfpetroleo*Sin(pi/4)+rauxperf
126            *Cos(pi/4) , lpetroleo };
127        Point (29)={ rposperfpetroleo*Cos(pi/4)-rauxperf
128            *Cos(pi/4) , rposperfpetroleo*Sin(pi/4)+rauxperf
129            *Cos(pi/4) , lpetroleo };

```

```

130 Point(30)={ rposperfpetroleo*cos(pi/4)-rauxperf
131           *cos(pi/4), rposperfpetroleo*sin(pi/4)-rauxperf
132           *cos(pi/4), lpetroleo };
133 Point(31)={ rposperfpetroleo*cos(pi/4)+rauxperf
134           *cos(pi/4), rposperfpetroleo*sin(pi/4)-rauxperf
135           *cos(pi/4), lpetroleo };
136 Point(32)={ rposperfpetroleo*cos(pi/4)+rperfpetroleo
137           *cos(pi/4), rposperfpetroleo*sin(pi/4)
138           +rperfpetroleo*cos(pi/4), lpetroleo };
139 Point(33)={ rposperfpetroleo*cos(pi/4)-rperfpetroleo
140           *cos(pi/4), rposperfpetroleo*sin(pi/4)
141           +rperfpetroleo*cos(pi/4), lpetroleo };
142 Point(34)={ rposperfpetroleo*cos(pi/4)-rperfpetroleo
143           *cos(pi/4), rposperfpetroleo*sin(pi/4)
144           -rperfpetroleo*cos(pi/4), lpetroleo };
145 Point(35)={ rposperfpetroleo*cos(pi/4)+rperfpetroleo
146           *cos(pi/4), rposperfpetroleo*sin(pi/4)
147           -rperfpetroleo*cos(pi/4), lpetroleo };
148 Line(82) = {28, 29};
149 Line(83) = {29, 30};
150 Line(84) = {30, 31};
151 Line(85) = {31, 28};
152 Line(86) = {28, 32};
153 Line(87) = {29, 33};
154 Line(88) = {30, 34};
155 Line(89) = {31, 35};
156 Circle(90) = {32, 27, 33};
157 Circle(91) = {33, 27, 34};
158 Circle(92) = {34, 27, 35};
159 Circle(93) = {35, 27, 32};
160 Line Loop(94) = {86, 90, -87, -82};
161 Ruled Surface(95) = {94};
162 Line Loop(96) = {87, 91, -88, -83};
163 Ruled Surface(97) = {96};
164 Line Loop(98) = {88, 92, -89, -84};
165 Ruled Surface(99) = {98};
166 Line Loop(100) = {89, 93, -86, -85};
167 Ruled Surface(101) = {100};
168 Line Loop(102) = {82, 83, 84, 85};
169 Ruled Surface(103) = {102};
170
171 // #####
172 // 2.4- Estructurado
173 // #####
174 n_vol_theta_perforacion=11;
175 n_vol_radial_perforacion=11;
176 Transfinite Line {82, 83, 84, 85, 90, 91, 92, 93} =
177           n_vol_theta_perforacion Using Progression 1;
178 Transfinite Line {86, 87, 88, 89} =

```

```

179         n_vol_radial_perforacion Using Progression 1;
180         Transfinite Surface {103} = {28, 29, 30, 31};
181         Transfinite Surface {95} = {32, 33, 29, 28};
182         Transfinite Surface {97} = {33, 34, 30, 29};
183         Transfinite Surface {99} = {34, 35, 31, 30};
184         Transfinite Surface {101} = {35, 32, 28, 31};
185         Recombine Surface {103, 101, 95, 97, 99};
186
187         // #####
188         // 2.5- Extrucion
189         // #####
190         n_vol_largo_perforacion=10;
191         Extrude {0, 0, lperforacion*a} {
192             Surface {103,95,97,99,101};
193             Layers {n_vol_largo_perforacion };
194             Recombine;
195         }
196
197
198         // #####
199         // #####
200         // 3.- cono de petroleo
201         // #####
202         // #####
203
204         //a=0.0001;
205         //a=1;
206         // #####
207         // 3.1- Parametros Geometricos
208         // #####
209         r_menor_cono_p=2.5*x_to_m;
210         l_cono_p=10*x_to_m;
211
212         // #####
213         // 3.2- Parametros Geometricos Auxiliares
214         // #####
215         rel_cono_p=r_menor_cono_p/rpetroleo;
216
217         // #####
218         // 3.3- Geometria Basica
219         // #####
220         Point(300)={0,0,lpetroleo+lperforacion };
221         Point(301)={ rpetroleo/2,0,lpetroleo+lperforacion };
222         Point(302)={ rauxint*cos(pi/4),rauxint*sin(pi/4),
223             lpetroleo+lperforacion };
224         Point(303)={0,rauxint,lpetroleo+lperforacion };
225         Point(304)={ rpetroleo,0,lpetroleo+lperforacion };
226         Point(305)={ rpetroleo*cos(pi/4),rpetroleo*sin(pi/4),
227             lpetroleo+lperforacion };

```

```

228 Point(306)={0,rpetroleo ,lpetroleo+lperforacion };
229 Line(301) = {300, 301};
230 Line(302) = {300, 303};
231 Line(303) = {301, 302};
232 Line(304) = {302, 303};
233 Line(305) = {301, 304};
234 Line(306) = {302, 305};
235 Line(307) = {303, 306};
236 Circle(308) = {304, 300, 305};
237 Circle(309) = {305, 300, 306};
238 Line Loop(310) = {301, 303, 304, -302};
239 Ruled Surface(311) = {310};
240 Line Loop(312) = {305, 308, -306, -303};
241 Ruled Surface(313) = {312};
242 Line Loop(314) = {309, -307, -304, 306};
243 Ruled Surface(315) = {314};
244
245 Point(350)={0,0,lpetroleo+lperforacion+(l_cono_p)*a };
246 Point(351)={rpetroleo/2*rel_cono_p ,0,lpetroleo
247 +lperforacion+(l_cono_p)*a };
248 Point(352)={rauxint*Cos(pi/4)*rel_cono_p ,rauxint
249 *Sin(pi/4)*rel_cono_p ,lpetroleo+lperforacion
250 +(l_cono_p)*a };
251 Point(353)={0,rauxint*rel_cono_p ,lpetroleo+lperforacion
252 +(l_cono_p)*a };
253 Point(354)={rpetroleo*rel_cono_p ,0,lpetroleo+lperforacion
254 +(l_cono_p)*a };
255 Point(355)={rpetroleo*Cos(pi/4)*rel_cono_p ,rpetroleo
256 *Sin(pi/4)*rel_cono_p ,lpetroleo+lperforacion
257 +(l_cono_p)*a };
258 Point(356)={0,rpetroleo*rel_cono_p ,lpetroleo+lperforacion
259 +(l_cono_p)*a };
260 Line(351) = {350, 351};
261 Line(352) = {350, 353};
262 Line(353) = {351, 352};
263 Line(354) = {352, 353};
264 Line(355) = {351, 354};
265 Line(356) = {352, 355};
266 Line(357) = {353, 356};
267 Circle(358) = {354, 350, 355};
268 Circle(359) = {355, 350, 356};
269 Line Loop(360) = {351, 353, 354, -352};
270 Ruled Surface(361) = {360};
271 Line Loop(362) = {355, 358, -356, -353};
272 Ruled Surface(363) = {362};
273 Line Loop(364) = {359, -357, -354, 356};
274 Ruled Surface(365) = {364};
275 Line(366) = {300, 350};
276 Line(367) = {301, 351};

```

```

277     Line(368) = {302, 352};
278     Line(369) = {303, 353};
279     Line(370) = {304, 354};
280     Line(371) = {305, 355};
281     Line(372) = {306, 356};
282     Line Loop(373) = {301, 367, -351, -366};
283     Ruled Surface(374) = {373};
284     Line Loop(375) = {303, 368, -353, -367};
285     Ruled Surface(376) = {375};
286     Line Loop(377) = {304, 369, -354, -368};
287     Ruled Surface(378) = {377};
288     Line Loop(379) = {302, 369, -352, -366};
289     Ruled Surface(380) = {379};
290     Line Loop(381) = {305, 370, -355, -367};
291     Ruled Surface(382) = {381};
292     Line Loop(383) = {308, 371, -358, -370};
293     Ruled Surface(384) = {383};
294     Line Loop(385) = {306, 371, -356, -368};
295     Ruled Surface(386) = {385};
296     Line Loop(387) = {309, 372, -359, -371};
297     Ruled Surface(388) = {387};
298     Line Loop(389) = {307, 372, -357, -369};
299     Ruled Surface(390) = {389};
300     Surface Loop(391) = {311, 374, 361, 380, 378, 376};
301     Volume(9) = {391};
302     Surface Loop(393) = {363, 382, 313, 384, 376, 386};
303     Volume(10) = {393};
304     Surface Loop(395) = {388, 315, 390, 365, 386, 378};
305     Volume(11) = {395};
306
307     // #####
308     // 3.4- Estructurado
309     // #####
310
311     n_vol_theta_petroleo=20;
312     n_vol_radial_petroleo=20;
313     n_vol_largo_cono_p=35;
314
315     Transfinite Line {301, 302, 303, 304, 308, 309} =
316         n_vol_theta_petroleo Using Progression 1;
317     Transfinite Line {305, 306, 307} = n_vol_radial_petroleo
318         Using Progression 1;
319     Transfinite Surface {311} = {300, 301, 302, 303};
320     Transfinite Surface {313} = {301, 304, 305, 302};
321     Transfinite Surface {315} = {302, 305, 306, 303};
322     Recombine Surface {311,313,315};
323
324     Transfinite Line {351, 352, 353, 354, 358, 359} =
325         n_vol_theta_petroleo Using Progression 1;

```

```

326         Transfinite Line {355, 356, 357} = n_vol_radial_petroleo
327             Using Progression 1;
328         Transfinite Surface {361} = {350, 351, 352, 353};
329         Transfinite Surface {363} = {351, 354, 355, 352};
330         Transfinite Surface {365} = {352, 355, 356, 353};
331         Recombine Surface {361,363,365};
332
333         Transfinite Line {366, 367, 368, 369, 370, 371, 372} =
334             n_vol_largo_cono_p Using Progression 1;
335         Transfinite Surface {374} = {300, 350, 351, 301};
336         Transfinite Surface {376} = {301, 351, 352, 302};
337         Transfinite Surface {378} = {302, 303, 353, 352};
338         Transfinite Surface {380} = {303, 300, 350, 353};
339         Transfinite Surface {382} = {301, 351, 354, 304};
340         Transfinite Surface {384} = {304, 354, 355, 305};
341         Transfinite Surface {386} = {302, 352, 355, 305};
342         Transfinite Surface {388} = {355, 305, 306, 356/*, 355*/};
343         Transfinite Surface {390} = {306, 303, 353, 356};
344         Recombine Surface {374,376,378,380,382,384,386,388,390};
345
346         Transfinite Volume{9} = {300, 301, 302, 303, 350, 351,
347             352, 353};
348         Transfinite Volume{10} = {304, 305, 302, 301, 354, 355,
349             352, 351};
350         Transfinite Volume{11} = {305, 306, 303, 302, 355, 356,
351             353, 352};
352
353         //#####
354         // 3.5- Extrucion
355         //#####
356
357         /*n_vol_largo_petroleo=10;
358         Extrude {0, 0, lpetroleo*a} {
359             Surface{311, 313, 315};
360             Layers{n_vol_largo_petroleo};
361             Recombine;
362         }*/
363
364
365         //#####
366         //#####
367         //      4.- tuberia de agua
368         //#####
369         //#####
370
371         //a=0.0001;
372         //a=1;
373         //#####
374         // 4.1- Parametros Geometricos

```

```

375 // #####
376     r_agua=9.5*x_to_m;
377     e_agua=1*x_to_m;
378     l_agua=lpetroleo+lperforacion;
379
380 // #####
381 // 4.2- Parametros Geometricos Auxiliares
382 // #####
383
384 // #####
385 // 4.3- Geometria Basica
386 // #####
387     Point(401)={r_agua-e_agua,0,0};
388     Point(402)={r_agua,0,0};
389     Point(403)={(r_agua-e_agua)*Cos(pi/4),(r_agua-e_agua)
390         *Sin(pi/4),0};
391     Point(404)={r_agua*Cos(pi/4),r_agua*Sin(pi/4),0};
392     Point(405)={0,r_agua-e_agua,0};
393     Point(406)={0,r_agua,0};
394     Line(400) = {401, 402};
395     Line(401) = {403, 404};
396     Line(402) = {405, 406};
397     Circle(403) = {401, 0, 403};
398     Circle(404) = {403, 0, 405};
399     Circle(405) = {402, 0, 404};
400     Circle(406) = {404, 0, 406};
401     Line Loop(407) = {400, 405, -401, -403};
402     Ruled Surface(408) = {407};
403     Line Loop(409) = {406, -402, -404, 401};
404     Ruled Surface(410) = {409};
405
406 // #####
407 // 4.4- Estructurado
408 // #####
409     n_vol_theta_agua=20;
410     n_vol_radial_agua=10;
411     n_vol_largo_agua=30;
412     Transfinite Line {403, 405, 404, 406} = n_vol_theta_agua
413         Using Progression 1;
414     Transfinite Line {400, 401, 402} = n_vol_radial_agua
415         Using Progression 1;
416     Transfinite Surface {408} = {401, 402, 404, 403};
417     Transfinite Surface {410} = {403, 404, 406, 405};
418     Recombine Surface {408,410};
419
420 // #####
421 // 4.5- Extrucion
422 // #####
423     Extrude {0, 0, l_agua*a} {

```

```

424         Surface {408,410};
425         Layers {n_vol_largo_agua };
426         Recombine;
427     }
428
429     //#####
430     //#####
431     //      5.- cono de agua
432     //#####
433     //#####
434
435     //a=0.0001;
436     //a=1;
437     //#####
438     // 5.1- Parametros Geometricos
439     //#####
440     r_menor_cono_agua=4.5*x_to_m;
441     l_cono_agua=l_cono_p;
442
443     //#####
444     // 5.2- Parametros Geometricos Auxiliares
445     //#####
446     rel_cono_a=r_menor_cono_agua/r_agua;
447
448     //#####
449     // 5.3- Geometria Basica
450     //#####
451     Point (501)={r_agua*rel_cono_a-e_agua ,0 ,(l_agua
452         +l_cono_agua)*a };
453     Point (502)={r_agua*rel_cono_a ,0 ,(l_agua+l_cono_agua)*a };
454     Point (503)={({r_agua*rel_cono_a-e_agua}*Cos(pi/4) ,(r_agua
455         *rel_cono_a-e_agua)*Cos(pi/4) ,(l_agua+l_cono_agua)
456         *a };
457     Point (504)={({r_agua*rel_cono_a}*Cos(pi/4) ,(r_agua
458         *rel_cono_a)*Cos(pi/4) ,(l_agua+l_cono_agua)*a };
459     Point (505)={0,r_agua*rel_cono_a-e_agua ,(l_agua+l_cono_agua)
460         *a };
461     Point (506)={0,r_agua*rel_cono_a ,(l_agua+l_cono_agua)*a };
462     Line (500) = {501, 502};
463     Line (501) = {503, 504};
464     Line (502) = {505, 506};
465     Circle (503) = {501, 350, 503};
466     Circle (504) = {503, 350, 505};
467     Circle (505) = {502, 350, 504};
468     Circle (506) = {504, 350, 506};
469     Line Loop(507) = {500, 505, -501, -503};
470     Ruled Surface (508) = {507};
471     Line Loop(509) = {506, -502, -504, 501};
472     Ruled Surface (510) = {509};

```

```

473 Line(511) = {407, 501};
474 Line(512) = {408, 502};
475 Line(513) = {424, 505};
476 Line(514) = {420, 506};
477 Line(515) = {417, 503};
478 Line(516) = {413, 504};
479 Line Loop(517) = {412, 512, -500, -511};
480 Ruled Surface(518) = {517};
481 Line Loop(519) = {413, 516, -505, -512};
482 Ruled Surface(520) = {519};
483 Line Loop(521) = {414, 515, 501, -516};
484 Ruled Surface(522) = {521};
485 Line Loop(523) = {415, 511, 503, -515};
486 Ruled Surface(524) = {523};
487 Line Loop(525) = {434, 514, -506, -516};
488 Ruled Surface(526) = {525};
489 Line Loop(527) = {435, 513, 502, -514};
490 Ruled Surface(528) = {527};
491 Line Loop(529) = {436, 515, 504, -513};
492 Ruled Surface(530) = {529};
493 Surface Loop(531) = {432, 518, 520, 508, 524, 522};
494 Volume(14) = {531};
495 Surface Loop(533) = {522, 510, 526, 528, 530, 454};
496 Volume(15) = {533};
497
498 //#####
499 // 5.4- Estructurado
500 //#####
501 /* n_vol_theta_agua=3;
502 n_vol_radial_agua=2;*/
503 n_vol_largo_cono_agua=30;
504 Transfinite Line {415, 413, 436, 434, 503, 504, 505, 506,
505 503, 504, 505, 506} = n_vol_theta_agua Using
506 Progression 1;
507 Transfinite Line {412, 414, 435, 500, 501, 502} =
508 n_vol_radial_agua Using Progression 1;
509 Transfinite Line {511, 512, 515, 516, 513, 514} =
510 n_vol_largo_cono_agua Using Progression 1;
511 Transfinite Surface {432} = {407, 408, 413, 417};
512 Transfinite Surface {454} = {417, 413, 420, 424};
513 Transfinite Surface {518} = {407, 501, 502, 408};
514 Transfinite Surface {522} = {417, 503, 504, 413};
515 Transfinite Surface {528} = {424, 505, 506, 420};
516 Transfinite Surface {520} = {408, 502, 504, 413};
517 Transfinite Surface {526} = {413, 420, 506, 504};
518 Transfinite Surface {524} = {407, 501, 503, 417};
519 Transfinite Surface {530} = {417, 424, 505, 503};
520 Transfinite Surface {508} = {501, 502, 504, 503};
521 Transfinite Surface {510} = {503, 504, 506, 505};

```

```

522         Recombine Surface {432, 454, 528, 522, 518, 530, 526, 524,
523             520, 508, 510};
524         Transfinite Volume{14} = {407, 408, 413, 417, 501, 502, 504,
525             503};
526         Transfinite Volume{15} = {417, 413, 420, 424, 503, 504, 506,
527             505};
528
529         // #####
530         // 5.5- Extrucion
531         // #####
532         /* Extrude {0, 0, l_agua*a} {
533             Surface {411};
534             Layers {n_vol_largo_agua};
535             Recombine;
536         }*/
537
538
539         // #####
540         // #####
541         // 6.- cono de mezcla
542         // #####
543         // #####
544
545         //a=0.0001;
546         //a=1;
547         // #####
548         // 6.1- Parametros Geometricos
549         // #####
550         r_cono_mezcla=7*x_to_m;
551         l_cono_mezcla=10*x_to_m;
552
553         // #####
554         // 6.2- Parametros Geometricos Auxiliares
555         // #####
556         rel_cono_mezcla=r_cono_mezcla/r_menor_cono_agua;
557
558         // #####
559         // 6.3- Geometria Basica
560         // #####
561         Point (650)={0,0,l_agua+l_cono_agua+l_cono_mezcla*a};
562         Point (651)={(r_agua*rel_cono_a-e_agua)*rel_cono_mezcla,0,
563             l_agua+l_cono_agua+l_cono_mezcla*a};
564         Point (652)={(r_agua*rel_cono_a)*rel_cono_mezcla,0,l_agua
565             +l_cono_agua+l_cono_mezcla*a};
566         Point (653)={(r_agua*rel_cono_a-e_agua)*Cos(pi/4)
567             *rel_cono_mezcla,(r_agua*rel_cono_a-e_agua)
568             *Cos(pi/4)*rel_cono_mezcla,l_agua+l_cono_agua
569             +l_cono_mezcla*a};
570         Point (654)={(r_agua*rel_cono_a)*Cos(pi/4)*rel_cono_mezcla,

```

```

571         (r_agua*rel_cono_a)*Cos(pi/4)*rel_cono_mezcla ,
572         l_agua+l_cono_agua+l_cono_mezcla*a};
573 Point(655)={0,(r_agua*rel_cono_a-e_agua)*rel_cono_mezcla ,
574         l_agua+l_cono_agua+l_cono_mezcla*a};
575 Point(656)={0,(r_agua*rel_cono_a)*rel_cono_mezcla ,l_agua
576         +l_cono_agua+l_cono_mezcla*a};
577 Point(6510)={(rpetroleo/2*rel_cono_p)*rel_cono_mezcla ,0 ,
578         lpetroleo+lperforacion+l_cono_p+(l_cono_mezcla)*a};
579 Point(6520)={(rauxint*Cos(pi/4)*rel_cono_p)*rel_cono_mezcla
580         ,(rauxint*Sin(pi/4)*rel_cono_p)*rel_cono_mezcla ,
581         lpetroleo+lperforacion+l_cono_p+(l_cono_mezcla)*a};
582 Point(6530)={0,(rauxint*rel_cono_p)*rel_cono_mezcla ,
583         lpetroleo+lperforacion+l_cono_p+(l_cono_mezcla)*a};
584 Point(6540)={(rpetroleo*rel_cono_p)*rel_cono_mezcla ,0 ,
585         lpetroleo+lperforacion+l_cono_p+(l_cono_mezcla)*a};
586 Point(6550)={(rpetroleo*Cos(pi/4)*rel_cono_p)
587         *rel_cono_mezcla ,(rpetroleo*Sin(pi/4)*rel_cono_p)
588         *rel_cono_mezcla ,lpetroleo+lperforacion+l_cono_p
589         +(l_cono_mezcla)*a};
590 Point(6560)={0,(rpetroleo*rel_cono_p)*rel_cono_mezcla ,
591         lpetroleo+lperforacion+l_cono_p+(l_cono_mezcla)*a};
592 Line(534) = {354, 501};
593 Line(535) = {355, 503};
594 Line(536) = {356, 505};
595 Line Loop(537) = {358, 535, -503, -534};
596 Ruled Surface(538) = {537};
597 Line Loop(539) = {359, 536, -504, -535};
598 Ruled Surface(540) = {539};
599
600 Line(541) = {350, 650};
601 Line(542) = {650, 6510};
602 Line(543) = {6510, 351};
603 Line(544) = {6510, 6520};
604 Line(545) = {6520, 352};
605 Line(546) = {6520, 6530};
606 Line(547) = {6530, 353};
607 Line(548) = {6530, 650};
608 Line Loop(549) = {351, -543, -542, -541};
609 Ruled Surface(550) = {549};
610 Line Loop(551) = {353, -545, -544, 543};
611 Ruled Surface(552) = {551};
612 Line Loop(553) = {354, -547, -546, 545};
613 Ruled Surface(554) = {553};
614 Line Loop(555) = {352, -547, 548, -541};
615 Ruled Surface(556) = {555};
616 Line Loop(557) = {542, 544, 546, 548};
617 Ruled Surface(558) = {557};
618 Surface Loop(559) = {558, 550, 552, 554, 556, 361};
619 Volume(560) = {559};

```

```
620 Line(561) = {6510, 6540};
621 Line(562) = {6540, 354};
622 Circle(563) = {6540, 650, 6550};
623 Line(564) = {6550, 355};
624 Line(565) = {6550, 6520};
625 Line Loop(566) = {355, -562, -561, 543};
626 Ruled Surface(567) = {566};
627 Line Loop(568) = {358, -564, -563, 562};
628 Ruled Surface(569) = {568};
629 Line Loop(570) = {356, -564, 565, 545};
630 Ruled Surface(571) = {570};
631 Line Loop(572) = {561, 563, 565, -544};
632 Ruled Surface(573) = {572};
633 Surface Loop(574) = {363, 567, 569, 571, 573, 552};
634 Volume(575) = {574};
635 Circle(576) = {6550, 650, 6560};
636 Line(577) = {6560, 6530};
637 Line(578) = {6560, 356};
638 Line Loop(579) = {359, -578, -576, 564};
639 Ruled Surface(580) = {579};
640 Line Loop(581) = {357, -578, 577, 547};
641 Ruled Surface(582) = {581};
642 Line Loop(583) = {546, -577, -576, 565};
643 Ruled Surface(584) = {583};
644 Surface Loop(585) = {365, 554, 571, 580, 582, 584};
645 Volume(586) = {585};
646 Line(587) = {501, 651};
647 Line(588) = {651, 6540};
648 Line(589) = {503, 653};
649 Circle(590) = {653, 650, 651};
650 Line(591) = {6550, 653};
651 Line Loop(592) = {534, 587, 588, 562};
652 Ruled Surface(593) = {592};
653 Line Loop(594) = {503, 589, 590, -587};
654 Ruled Surface(595) = {594};
655 Line Loop(596) = {535, 589, -591, 564};
656 Ruled Surface(597) = {596};
657 Line Loop(598) = {588, 563, 591, 590};
658 Ruled Surface(599) = {598};
659 Surface Loop(600) = {538, 593, 595, 597, 599, 569};
660 Volume(601) = {600};
661 Circle(602) = {653, 650, 655};
662 Line(603) = {655, 505};
663 Line(604) = {655, 6560};
664 Line Loop(605) = {504, -603, -602, -589};
665 Ruled Surface(606) = {605};
666 Line Loop(607) = {536, -603, 604, 578};
667 Ruled Surface(608) = {607};
668 Line Loop(609) = {576, -604, -602, -591};
```

```

669         Ruled Surface(610) = {609};
670         Surface Loop(611) = {540, 608, 606, 610, 580, 597};
671         Volume(612) = {611};
672         Line(613) = {651, 652};
673         Line(614) = {652, 502};
674         Circle(615) = {652, 650, 654};
675         Line(616) = {654, 504};
676         Line(617) = {654, 653};
677         Line Loop(618) = {500, -614, -613, -587};
678         Ruled Surface(619) = {618};
679         Line Loop(620) = {505, -616, -615, 614};
680         Ruled Surface(621) = {620};
681         Line Loop(622) = {501, -616, 617, -589};
682         Ruled Surface(623) = {622};
683         Line Loop(624) = {590, 613, 615, 617};
684         Ruled Surface(625) = {624};
685         Surface Loop(626) = {508, 619, 621, 623, 625, 595};
686         Volume(627) = {626};
687         Circle(628) = {654, 650, 656};
688         Line(629) = {656, 506};
689         Line(630) = {656, 655};
690         Line Loop(631) = {506, -629, -628, 616};
691         Ruled Surface(632) = {631};
692         Line Loop(633) = {502, -629, 630, 603};
693         Ruled Surface(634) = {633};
694         Line Loop(635) = {602, -630, -628, 617};
695         Ruled Surface(636) = {635};
696         Surface Loop(637) = {510, 606, 623, 632, 634, 636};
697         Volume(638) = {637};
698
699         // #####
700         // 6.4- Estructurado
701         // #####
702         n_vol_largo_cono_mezcla=45;
703         n_vol_radial_pared=15;
704         n_vol_theta_cono_mezcla=n_vol_theta_petroleo;
705         n_vol_radial_cono_mezcla=n_vol_radial_petroleo;
706         Transfinite Line {351, 352, 353, 354, 358, 359, 503, 504,
707             505, 506, 542, 544, 546, 548, 563, 576, 590, 602,
708             615, 628} = n_vol_theta_cono_mezcla Using
709             Progression 1;
710         Transfinite Line {355, 356, 357, 561, 565, 577} =
711             n_vol_radial_cono_mezcla Using Progression 1;
712         Transfinite Line {500, 501, 502, 613, 617, 630} =
713             n_vol_radial_agua Using Progression 1;
714         Transfinite Line {534, 535, 536, 588, 591, 604} =
715             n_vol_radial_pared Using Progression 1;
716         Transfinite Line {541, 543, 545, 547, 562, 564, 578, 587,
717             589, 603, 614, 616, 629} = n_vol_largo_cono_mezcla

```

```

718         Using Progression 1;
719         /*Transfinite Surface {361} = {350, 351, 352, 353};
720         Transfinite Surface {363} = {351, 354, 355, 352};
721         Transfinite Surface {365} = {352, 355, 356, 353};*/
722         Transfinite Surface {508} = {501, 502, 504, 503};
723         Transfinite Surface {510} = {503, 504, 506, 505};
724         Transfinite Surface {538} = {354, 501, 503, 355};
725         Transfinite Surface {540} = {355, 503, 505, 356};
726         Transfinite Surface {550} = {350, 650, 6510, 351};
727         Transfinite Surface {552} = {351, 6510, 6520, 352};
728         Transfinite Surface {554} = {352, 6520, 6530, 353};
729         Transfinite Surface {556} = {353, 6530, 650, 350};
730         Transfinite Surface {558} = {650, 6510, 6520, 6530};
731         Transfinite Surface {567} = {351, 6510, 6540, 354};
732         Transfinite Surface {569} = {354, 6540, 6550, 355};
733         Transfinite Surface {571} = {355, 6550, 6520, 352};
734         Transfinite Surface {573} = {6510, 6540, 6550, 6520};
735         Transfinite Surface {580} = {355, 356, 6560, 6550};
736         Transfinite Surface {582} = {353, 6530, 6560, 356};
737         Transfinite Surface {584} = {6520, 6550, 6560, 6530};
738         Transfinite Surface {593} = {354, 501, 651, 6540};
739         Transfinite Surface {595} = {501, 651, 653, 503};
740         Transfinite Surface {597} = {355, 6550, 653, 503};
741         Transfinite Surface {599} = {6540, 651, 653, 6550};
742         Transfinite Surface {606} = {503, 653, 655, 505};
743         Transfinite Surface {608} = {505, 655, 6560, 356};
744         Transfinite Surface {610} = {6550, 653, 655, 6560};
745         Transfinite Surface {619} = {501, 651, 652, 502};
746         Transfinite Surface {621} = {502, 652, 654, 504};
747         Transfinite Surface {623} = {504, 654, 653, 503};
748         Transfinite Surface {625} = {651, 652, 654, 653};
749         Transfinite Surface {632} = {504, 506, 656, 654};
750         Transfinite Surface {634} = {506, 505, 655, 656};
751         Transfinite Surface {636} = {653, 654, 656, 655};
752         Recombine Surface {361,363,365,508,510,538,540,550,552,
753             554,556,558, 567, 569, 571, 573, 580, 582, 584,
754             593, 595, 597, 599, 606, 608, 610, 619, 621, 623,
755             625, 632, 634, 636};
756         Transfinite Volume{560} = {350, 351, 352, 353, 650, 6510,
757             6520, 6530};
758         Transfinite Volume{575} = {351, 354, 355, 352, 6510, 6540,
759             6550, 6520};
760         Transfinite Volume{586} = {353, 352, 355, 356, 6530, 6520,
761             6550, 6560};
762         Transfinite Volume{601} = {354, 501, 503, 355, 6540, 651,
763             653, 6550};
764         Transfinite Volume{612} = {355, 503, 505, 356, 6550, 653,
765             655, 6560};
766         Transfinite Volume{627} = {501, 502, 504, 503, 651, 652,

```

```

767         654, 653};
768         Transfinite Volume{638} = {503, 504, 506, 505, 653, 654,
769         656, 655};
770         Recombine Volume {560,575,586,601,612,627, 638};
771
772         // #####
773         // 6.5- Extrucion
774         // #####
775
776         /* n_vol_largo_petroleo=10;
777         Extrude {0, 0, lpetroleo*a} {
778         Surface{311, 313, 315};
779         Layers{n_vol_largo_petroleo};
780         Recombine;
781         }*/
782
783         // #####
784         // #####
785         // 7.- camara de atomizacion
786         // #####
787         // #####
788
789         //a=0.0001;
790         //a=1;
791         // #####
792         // 7.1- Parametros Geometricos
793         // #####
794         r_camara_atomizacion=11.5*x_to_m;
795         l_camara_atomizacion=23*x_to_m;
796
797         // #####
798         // 7.2- Parametros Geometricos Auxiliares
799         // #####
800
801         // #####
802         // 7.3- Geometria Basica
803         // #####
804         Point(700)={r_camara_atomizacion , 0, lpetroleo
805         +lperforacion+l_cono_p+l_cono_mezcla};
806         Point(701)={r_camara_atomizacion*Cos(pi/4) ,
807         r_camara_atomizacion*Sin(pi/4) ,lpetroleo
808         +lperforacion+l_cono_p+l_cono_mezcla};
809         Point(702)={0, r_camara_atomizacion , lpetroleo
810         +lperforacion+l_cono_p+l_cono_mezcla};
811         Line(639) = {652, 700};
812         Line(640) = {654, 701};
813         Line(641) = {656, 702};
814         Circle(642) = {700, 650, 701};
815         Circle(643) = {701, 650, 702};

```

```

816         Line Loop(644) = {615, 640, -642, -639};
817         Ruled Surface(645) = {644};
818         Line Loop(646) = {628, 641, -643, -640};
819         Ruled Surface(647) = {646};
820
821         // #####
822         // 7.4- Estructurado
823         // #####
824         // n_vol_theta_cono_mezcla=30;
825         n_vol_radial_camara_atomizacion=30;
826         n_vol_largo_camara_atomizacion=45;
827
828         Transfinite Line {642,643} = n_vol_theta_cono_mezcla Using
829             Progression 1;
830         Transfinite Line {639,640,641} =
831             n_vol_radial_camara_atomizacion Using Progression 1;
832         Transfinite Surface {645} = {652, 700, 701, 654};
833         Transfinite Surface {647} = {654, 701, 702, 656};
834         Recombine Surface {645, 647};
835
836         // #####
837         // 7.5- Extrucion
838         // #####
839         n_vol_largo_camara_atomizacion=150;
840         Extrude {0, 0, l_camara_atomizacion*a} {
841             Surface {558,573,584,599,610,625,636,645,647};
842             Layers {n_vol_largo_camara_atomizacion};
843             Recombine;
844         }
845
846
847         // #####
848         // #####
849         // 8.- Grupos Fisicos
850         // #####
851         // #####
852
853         // #####
854         // 8.1- Volumenes Fisicos (Condicion Inicial)
855         // #####
856
857         // #####
858         // 8.1.1- Petroleo (listo)
859         // #####
860         // Volumenes dentro de la tuberia de petroleo
861         // 1,2,3
862         // Volumenes dentro de la perforacion
863         // 4,5,6,7,8
864         // Volumenes dentro del cono de petroleo

```

```

865 //9,10,11
866 Physical Volume("oil")={1,2,3,4,5,6,7,8,9,10,11};
867
868 //#####
869 // 8.1.2- Agua (listo)
870 //#####
871 //Volumenes dentro de la tuberia de agua
872 //12,13
873 Physical Volume("water")={12,13,14,15};
874
875 //#####
876 // 8.1.3- Aire (listo)
877 //#####
878 //Volumenes dentro de la camara de mezcla
879 //14,15,16 (listo)
880 //Volumenes dentro de la camara de atomizacion
881 //17,18,19,20,21 (listo)
882 Physical Volume("air")={560,586,575,612,601,638,627,
883 639,641,640,643,642,645,644,647,646};
884
885 //#####
886 // 8.2- Superficies Fisicas (Condiciones de Contorno)
887 //#####
888
889 //#####
890 // 8.2.1- Entrada Petroleo (listo)
891 //#####
892 //Superficies en la entrada de la tuberia de
893 //petroleo
894 Physical Surface("oilIn") = {11, 13, 15};
895
896 //#####
897 // 8.2.2- Entrada Agua (listo)
898 //#####
899 //Superficies en la entrada de la tuberia de agua
900 Physical Surface("waterIn") = {410,408};
901
902 //#####
903 // 8.2.3- Pared (listo)
904 //#####
905
906 //Superficies en la pared de la tuberia de petroleo
907 //50,68 (listo)
908 //Superficies en la pared de la tuberia de agua
909 //432,424,523,519 (listo)
910 //Superficies en la pared de la placa perforada
911 //Cara interior de la tuberia de petroleo
912 //37,59,81 (combinar con base perforacion) (listo)
913 //Caras laterales de la placa perforada (listo)

```

```

914 //138,160,182,204
915 //Caras interior cono petroleo
916 //311,315,313
917 //Superficies en la pared del cono de petroleo
918 //384,388
919 //Superficies en el cono de mezcla
920 //Base radio menor
921 //611,613,615
922 //Cara lateral
923 //688,684
924 //Superficies en la camara de atomizacion
925 //Base
926 //702,704
927 //Lateral
928 //791,801
929 Physical Surface("wall") = {68, 50, 431, 449, 160,
930 182, 134, 204, 138, 530, 524, 423, 441,
931 520, 526, 540, 538, 621, 632, 647, 645,
932 840, 818, 388, 384};
933 Physical Surface("wall_combinar1") = {37,59,81};
934 Physical Surface("wall_combinar2") = {311,315,313};
935
936 #####
937 // 8.2.4- Simetria (listo)
938 #####
939 //Superficies dentro de la tuberia de petroleo
940 //24,36,46,72
941 //Superficies dentro del cono de petroleo
942 //374,380,390,382
943 //Superficies dentro de la tuberia de agua
944 //428,420
945 //Superficies dentro del cono de agua
946 //517,521
947 //Superficies dentro del cono de mezcla
948 //680,690,674,682
949 //Superficies dentro de la camara de atomizacion
950 //725,761,805,713,735,779
951 Physical Surface("symmetry_1") = {24, 46, 374, 382,
952 419, 518, 550, 567, 593, 619, 656, 678, 722,
953 770, 822};
954 Physical Surface("symmetry_2") = {72, 445, 390, 380,
955 36, 528, 556, 582, 608, 634, 668, 704, 748,
956 792, 836};
957
958 #####
959 // 8.2.5- Caras a combinar (listo)
960 #####
961 //Superficies de entrada a la perforacion
962 //103,95,101,99,97

```

```
963 //Superficies de entrada al cono de petroleo
964 //125,213,147,169,191
965 //Superficies de entrada al cono de mezcla
966 //361,363,365,511
967 Physical Surface("combinar_1") = {103,95,101,99,97};
968 Physical Surface("combinar_2") = {125,213,147,169,
969 191};
970
971 //#####
972 // 8.2.6- Salida
973 //#####
974 //Superficies en la salida
975 //726,748,770,792,814
976 Physical Surface("out") = {669,691,713,735,757,
977 779,801,823,845};
978
979 Coherence;
```

Apéndice C

Casos simulados

En el presente anexo se presentan los archivos necesarios para las simulaciones realizadas en el estudio actual, en primera instancia se presenta el caso base denominado “control” el cual sirve de punto inicial para los demás casos simulados. Para los casos posteriores, se presentan exclusivamente aquellos archivos que fueron modificados, los archivos que no se muestran son iguales a los presentados en el caso de control

C.1. Caso Control

C.1.1. 0/alpha.oil

```
1  /*----- C++ -----*/
2  | ===== |
3  | \\      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \\      /  O p e r a t i o n | Version: 2.3.1 |
5  | \\      /  A n d          | Web: www.OpenFOAM.org |
6  | \\      /  M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volScalarField;
13     location      "0";
14     object        alpha.oil;
15 }
16 // ***** //
```

```
17
18 dimensions      [0 0 0 0 0 0 0];
19
20 internalField    uniform 0;
21
22
23 boundaryField
24 {
25     oilIn
26     {
27         type      fixedValue;
28         value     uniform 1;
29     }
30     symmetry_1
31     {
32         type      symmetryPlane;
33     }
34     symmetry_2
35     {
36         type      symmetryPlane;
37     }
38     wall_combinar1
39     {
40         type      zeroGradient;
41     }
42     wall
43     {
44         type      zeroGradient;
45     }
46     combinar_1
47     {
48         type      zeroGradient;
49     }
50     combinar_2
51     {
52         type      zeroGradient;
53     }
54     wall_combinar2
55     {
56         type      zeroGradient;
57     }
58     waterIn
59     {
60         type      fixedValue;
61         value     uniform 0;
62     }
63     out
64     {
65         type      zeroGradient;
```

```
66     }
67   }
68
69
70 // ***** //
```

C.1.2. 0/alpha.water

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      /  O p e r a t i o n  | Version: 2.3.1 |
5  | \ \      /  A n d      | Web: www.OpenFOAM.org |
6  |  \ \ /      M a n i p u l a t i o n  | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volScalarField;
13     location      "0";
14     object        alpha.water;
15 }
16 // ***** //
17
18 dimensions      [0 0 0 0 0 0 0];
19
20 internalField    uniform 0;
21
22 boundaryField
23 {
24     oilIn
25     {
26         type      fixedValue;
27         value      uniform 0;
28     }
29     symmetry_1
30     {
31         type      symmetryPlane;
32     }
33     symmetry_2
34     {
35         type      symmetryPlane;
36     }
37     wall_combinar1
38     {
39         type      zeroGradient;
40     }
41     wall
42     {
43         type      zeroGradient;
44     }
45     combinar_1
46     {

```

```
47     type          zeroGradient;
48   }
49   combinar_2
50   {
51     type          zeroGradient;
52   }
53   wall_combinar2
54   {
55     type          zeroGradient;
56   }
57   waterIn
58   {
59     type          fixedValue;
60     value         uniform 1;
61   }
62   out
63   {
64     type          zeroGradient;
65   }
66 }
67
68
69 // ***** //
```

C.1.3. 0/alphas

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O p e r a t i o n | Version: 2.3.0 |
5  | \ \      / A n d      | Web: www.OpenFOAM.org |
6  | \ \ /      M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volScalarField;
13     location      "0";
14     object        alphas;
15 }
16 // ***** //
17
18 dimensions      [0 0 0 0 0 0 0];
19
20 internalField    uniform 1;
21
22 boundaryField
23 {
24     symmetry_1
25     {
26         type      symmetryPlane;
27     }
28
29     symmetry_2
30     {
31         type      symmetryPlane;
32     }
33
34     oilIn
35     {
36         type      zeroGradient;
37     }
38
39     wall
40     {
41         type      zeroGradient;
42     }
43
44     waterIn
45     {
46         type      zeroGradient;

```

```
47     }
48
49     wall_combinar1
50     {
51         type          zeroGradient;
52     }
53
54     wall_combinar2
55     {
56         type          zeroGradient;
57     }
58
59     out
60     {
61         type          zeroGradient;
62     }
63
64     combinar_1
65     {
66         type          zeroGradient;
67     }
68
69     combinar_2
70     {
71         type          zeroGradient;
72     }
73 }
74
75
76 // ***** //
```

C.1.4. 0/k

```
1  /*----- C++ -----*/
2  | ===== |
3  | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O p e r a t i o n | Version: 2.3.0 |
5  | \ \ / A n d | Web: www.OpenFOAM.org |
6  | \ \ / M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volScalarField;
13     location      "0";
14     object        k;
15 }
16 // ***** //
17
18 dimensions      [0 2 -2 0 0 0 0];
19
20 internalField    uniform 1e-11;
21
22 boundaryField
23 {
24     symmetry_1
25     {
26         type      symmetryPlane;
27     }
28
29     symmetry_2
30     {
31         type      symmetryPlane;
32     }
33
34     oilIn
35     {
36         type      fixedValue;
37         value     1e-05;
38     }
39
40     wall
41     {
42         type      zeroGradient;
43     }
44
45     waterIn
46     {
```

```
47     type      fixedValue;
48     value     1e-05;
49 }
50
51 wall_combinar1
52 {
53     type      zeroGradient;
54 }
55
56 wall_combinar2
57 {
58     type      zeroGradient;
59 }
60
61 out
62 {
63     type      zeroGradient;
64 }
65
66 combinar_1
67 {
68     type      zeroGradient;
69 }
70
71 combinar_2
72 {
73     type      zeroGradient;
74 }
75 }
76
77
78 // ***** //
```

C.1.5. 0/nuSgs

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \ / Field | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / Operation | Version: 2.3.0 |
5  | \ \ / And | Web: www.OpenFOAM.org |
6  | \ \ / Manipulation | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volScalarField;
13     location      "0";
14     object        nuSgs;
15 }
16 // ***** //
17
18 dimensions      [0 2 -1 0 0 0 0];
19
20 internalField    uniform 1e-11;
21
22 boundaryField
23 {
24     symmetry_1
25     {
26         type      symmetryPlane;
27     }
28
29     symmetry_2
30     {
31         type      symmetryPlane;
32     }
33
34     oilIn
35     {
36         type      zeroGradient;
37     }
38
39     wall
40     {
41         type      zeroGradient;
42     }
43
44     waterIn
45     {
46         type      zeroGradient;

```

```
47     }
48
49     wall_combinar1
50     {
51         type          zeroGradient;
52     }
53
54     wall_combinar2
55     {
56         type          zeroGradient;
57     }
58
59     out
60     {
61         type          zeroGradient;
62     }
63
64     combinar_1
65     {
66         type          zeroGradient;
67     }
68
69     combinar_2
70     {
71         type          zeroGradient;
72     }
73 }
74
75
76 // ***** //
```

C.1.6. 0/p_rgh

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O p e r a t i o n | Version: 2.3.0 |
5  | \ \      / A n d           | Web: www.OpenFOAM.org |
6  | \ \ /      M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volScalarField;
13     location      "0";
14     object        p_rgh;
15 }
16 // ***** //
17
18 dimensions      [1 -1 -2 0 0 0 0];
19
20 internalField    uniform 0;
21
22 boundaryField
23 {
24     symmetry_1
25     {
26         type      symmetryPlane;
27     }
28
29     symmetry_2
30     {
31         type      symmetryPlane;
32     }
33
34     oilIn
35     {
36         /* type      zeroGradient;
37         type      fixedValue;
38         value      uniform 10; */
39
40         type      fixedFluxPressure;
41         value      uniform 0;
42     }
43 }
44
45 wall
46 {

```

```
47     /* type          zeroGradient; */
48
49     type          fixedFluxPressure;
50     value         uniform 0;
51 }
52
53 waterIn
54 {
55     /* type          zeroGradient; */
56
57     type          fixedValue;
58     value         uniform 10; */
59
60     type          fixedFluxPressure;
61     value         uniform 0;
62 }
63
64 wall_combinar1
65 {
66     /* type          zeroGradient; */
67
68     type          fixedFluxPressure;
69     value         uniform 0;
70 }
71
72 wall_combinar2
73 {
74     /* type          zeroGradient; */
75
76     type          fixedFluxPressure;
77     value         uniform 0;
78 }
79
80 out
81 {
82     /* type          fixedValue; */
83     value         uniform 0;
84
85     type          fixedFluxPressure;
86     value         uniform 0; */
87
88     type          totalPressure;
89     p0            uniform 0;
90     U             U;
91     phi           phi;
92     rho           rho;
93     psi           none;
94     gamma         1;
95     value         uniform 0;
```

```
96
97     }
98
99     combinar_1
100    {
101        type          zeroGradient;
102    }
103
104     combinar_2
105    {
106        type          zeroGradient;
107    }
108 }
109
110
111 // ***** //
```

C.1.7. 0/T

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O p e r a t i o n | Version: 2.3.1 |
5  | \ \ / A n d | Web: www.OpenFOAM.org |
6  | \ \ / M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volScalarField;
13     location      "0";
14     object        T;
15 }
16 // ***** //
17
18 dimensions      [0 0 0 1 0 0 0];
19
20 internalField    uniform 273.15;
21
22 boundaryField
23 {
24     oilIn
25     {
26         type      fixedValue;
27         value      uniform 293.15;
28     }
29     symmetry_1
30     {
31         type      symmetryPlane;
32     }
33     symmetry_2
34     {
35         type      symmetryPlane;
36     }
37     wall_combinar1
38     {
39         type      fixedGradient;
40         gradient  uniform 3000;
41     }
42     wall
43     {
44         type      fixedGradient;
45         gradient  uniform 3000;
46     }

```

```
47     combinar_1
48     {
49         type          zeroGradient;
50     }
51     combinar_2
52     {
53         type          zeroGradient;
54     }
55     wall_combinar2
56     {
57         type          fixedGradient;
58         gradient      uniform 3000;
59     }
60     waterIn
61     {
62         type          fixedValue;
63         value         uniform 293.15;
64     }
65     out
66     {
67         type          fixedGradient;
68         gradient      uniform 3000;
69     }
70 }
71
72
73 // ***** //
```

C.1.8. 0/U

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O p e r a t i o n | Version: 2.3.1 |
5  | \ \      / A n d      | Web: www.OpenFOAM.org |
6  | \ \ /      M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volVectorField;
13     location      "0";
14     object        U;
15 }
16 // ***** //
17
18 dimensions      [0 1 -1 0 0 0 0];
19
20 internalField    uniform (0 0 0);
21
22 boundaryField
23 {
24     oilIn
25     {
26         type      fixedValue;
27         value      uniform (0 0 3.3287);
28     }
29     symmetry_1
30     {
31         type      symmetryPlane;
32     }
33     symmetry_2
34     {
35         type      symmetryPlane;
36     }
37     wall_combinar1
38     {
39         type      fixedValue;
40         value      uniform (0 0 0);
41     }
42     wall
43     {
44         type      fixedValue;
45         value      uniform (0 0 0);
46     }

```

```
47     combinar_1
48     {
49         type          zeroGradient;
50     }
51     combinar_2
52     {
53         type          zeroGradient;
54     }
55     wall_combinar2
56     {
57         type          fixedValue;
58         value         uniform (0 0 0);
59     }
60     waterIn
61     {
62         type          fixedValue;
63         value         uniform (0 0 3.0093);
64     }
65     out
66     {
67         type          zeroGradient;
68     }
69 }
70
71
72 // ***** //
```

C.1.9. constant/LESProperties

```

1  /*-----* C++ *-----*/
2  | ===== |
3  | \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \ / O p e r a t i o n | Version: 2.3.0 |
5  | \ \ / A n d | Web: www.OpenFOAM.org |
6  | \ \ / M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "constant";
14     object        LESProperties;
15 }
16 // * * * * * //
17
18 LESModel        oneEqEddy;
19 //LESModel      dynOneEqEddy;
20 //LESModel      Smagorinsky;
21 //LESModel      homogeneousDynSmagorinsky;
22
23 //delta         cubeRootVol;
24 //vanDriest se debe utilizar para modelos no dinamicos
25 delta           vanDriest;
26
27 printCoeffs     on;
28
29 cubeRootVolCoeffs
30 {
31     deltaCoeff    1;
32 }
33
34 PrandtlCoeffs
35 {
36     delta         cubeRootVol;
37     cubeRootVolCoeffs
38     {
39         deltaCoeff    1;
40     }
41
42     smoothCoeffs
43     {
44         delta         cubeRootVol;
45         cubeRootVolCoeffs
46         {

```

```
47     deltaCoeff    1;
48   }
49
50   maxDeltaRatio  1.1;
51 }
52
53 Cdelta          0.158;
54 }
55
56 vanDriestCoeffs
57 {
58   delta          cubeRootVol;
59   cubeRootVolCoeffs
60   {
61     deltaCoeff    1;
62   }
63
64   smoothCoeffs
65   {
66     delta          cubeRootVol;
67     cubeRootVolCoeffs
68     {
69       deltaCoeff    1;
70     }
71
72     maxDeltaRatio  1.1;
73   }
74
75   Aplus          26;
76   Cdelta         0.158;
77 }
78
79 smoothCoeffs
80 {
81   delta          cubeRootVol;
82   cubeRootVolCoeffs
83   {
84     deltaCoeff    1;
85   }
86
87   maxDeltaRatio  1.1;
88 }
89
90 homogeneousDynSmagorinskyCoeffs
91 {
92   filter simple;
93   ce 1.05;
94 }
95
```

```
96  dynOneEqEddyCoeffs
97  {
98      cubeRootVolCoeffs
99      {
100         deltaCoeff    1;
101     }
102     filter    simple;
103 }
104
105 // ***** //
```

C.1.10. constant/transportProperties

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      /  O p e r a t i o n  | Version:  2.3.0 |
5  | \ \      /  A n d      | Web:      www.OpenFOAM.org |
6  | \ \ /      M a n i p u l a t i o n  | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "constant";
14     object       transportProperties;
15 }
16 // ***** //
17
18 Tref           Tref [0 0 0 1 0 0 0] 273.15;
19
20 phases
21 (
22     water
23     {
24         transportModel Newtonian;
25         nu nu [ 0 2 -1 0 0 0 0 ] 1.002e-06;
26         rho rho [ 1 -3 0 0 0 0 0 ] 1000;
27         Cp Cp [ 0 2 -2 -1 0 0 0 ] 4185;
28         Pr Pr [ 0 0 0 0 0 0 0 ] 5;
29         Beta Beta [ 0 0 0 -1 0 0 0 ] 0.05;
30     }
31
32     oil //Diesel
33     {
34         transportModel Newtonian;
35         nu nu [ 0 2 -1 0 0 0 0 ] 3e-06;
36         rho rho [ 1 -3 0 0 0 0 0 ] 850;
37         Cp Cp [ 0 2 -2 -1 0 0 0 ] 2130;
38         Pr Pr [ 0 0 0 0 0 0 0 ] 5;
39         Beta Beta [ 0 0 0 -1 0 0 0 ] 0.05;
40     }
41
42     air
43     {
44         transportModel Newtonian;
45         nu nu [ 0 2 -1 0 0 0 0 ] 1.48e-05;
46         rho rho [ 1 -3 0 0 0 0 0 ] 1;

```

```
47     Cp Cp [0 2 -2 -1 0 0 0] 1000;
48     Pr Pr [ 0 0 0 0 0 0 0 ] 0.7;
49     Beta Beta [ 0 0 0 -1 0 0 0] 0.03;
50   }
51 );
52
53 sigmas
54 (
55   (air water) 0.065
56   (air oil) 0.03264
57   (water oil) 0.03264
58 );
59
60
61 // ***** //
```

C.1.11. system/controlDict

```

1  /*-----* C++ *-----*/
2  | ===== | |
3  | \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O p e r a t i o n | Version: 2.3.0 |
5  | \ \      / A n d      | Web: www.OpenFOAM.org |
6  | \ \ /      M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        controlDict;
15 }
16 // * * * * * //
17
18 // application compressibleMultiphaseInterFoam;
19 application     MTBoussinesqIFoam;
20 // application  MTIFoam;
21
22
23 // startFrom      startTime;
24 startFrom        latestTime;
25 // startTime      0;
26
27 stopAt           endTime;
28 // stopAt         writeNow;
29
30 endTime          1;
31
32 deltaT           0.00001;
33
34 writeControl     adjustableRunTime;
35
36 writeInterval    0.001;
37
38 purgeWrite       0;
39
40 writeFormat      ascii;
41
42 writePrecision   6;
43
44 writeCompression uncompressed;
45
46 timeFormat       general;

```

```
47
48 timePrecision 6;
49
50 runTimeModifiable yes;
51
52 adjustTimeStep yes;
53
54 maxCo 0.5;
55 maxAlphaCo 0.5;
56 maxFo 0.2;
57
58 maxDeltaT 0.001;
59
60 libs ("libmyBCs.so");
61
62 functions
63 {
64     fieldAverage1
65     {
66         // Type of functionObject
67         type fieldAverage;
68
69         // Where to load it from (if not already in solver)
70         functionObjectLibs ("libfieldFunctionObjects.so");
71
72         // Function object enabled flag
73         enabled true;
74
75         // When to output the average fields
76         outputControl outputTime;
77
78         // Fields to be averaged – runTime modifiable
79         fields
80         (
81             U
82             {
83                 mean on;
84                 prime2Mean on;
85                 base time;
86                 window 0.100; // optional averaging window
87             }
88
89             p
90             {
91                 mean on;
92                 prime2Mean on;
93                 base time;
94                 window 0.100; // optional averaging window
95             }
96         )
97     }
98 }
```

```
96
97     T
98     {
99         mean      on;
100        prime2Mean on;
101        base      time;
102        window    0.100; // optional averaging window
103    }
104
105    alpha.air
106    {
107        mean      on;
108        prime2Mean on;
109        base      time;
110        window    0.100; // optional averaging window
111    }
112
113    alpha.water
114    {
115        mean      on;
116        prime2Mean on;
117        base      time;
118        window    0.100; // optional averaging window
119    }
120
121    alpha.oil
122    {
123        mean      on;
124        prime2Mean on;
125        base      time;
126        window    0.100; // optional averaging window
127    }
128
129    alphas
130    {
131        mean      on;
132        prime2Mean on;
133        base      time;
134        window    0.100; // optional averaging window
135    }
136
137    );
138 }
139
140 probes
141 {
142     // Where to load it from
143     functionObjectLibs ( "libsampling.so" );
144 }
```

```
145     type          probes ;
146
147     // Name of the directory for probe data
148     name          probes ;
149
150     // Write at same frequency as fields
151     outputControl  timeStep ;
152     outputInterval 1 ;
153
154     // Fields to be probed
155     fields
156     (
157         p
158         U
159         alpha.air
160         alpha.water
161         alpha.oil
162         T
163     );
164
165     probeLocations
166     (
167         ( 0.00375 0.00375 0.01 ) // tuberia de petroleo
168         ( 0.0025 0.0025 0.028 ) // cono de petroleo
169         //( 0.00375 0.00375 0.043 ) // cono de mezcla
170         ( 0.006 0.006 0.06 ) // camara de mezcla
171     );
172 }
173
174 }
175 // ***** //
```

C.1.12. system/fvSchemes

```

1  /*-----* C++ *-----*/
2  | ===== | |
3  | \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O p e r a t i o n | Version: 2.3.0 |
5  | \ \      / A n d      | Web: www.OpenFOAM.org |
6  |  \ \ /      M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        fvSchemes;
15 }
16 // * * * * * //
17
18 ddtSchemes
19 {
20     default        CrankNicolson 1/* 0.8 */;
21 }
22
23 gradSchemes
24 {
25     default        Gauss linear;
26     grad(U)        Gauss linear;
27     grad(gamma)    Gauss linear;
28 }
29
30 divSchemes
31 {
32     // div(rhoPhi,U)    Gauss QUICK;
33     // div(rhoPhi,U)    Gauss linear;
34     // div(rhoPhi,U)    Gauss MUSCL;
35     div(rhoPhi,U)    Gauss vanLeer;
36     // div(rhoPhi,U)    Gauss upwind;
37     // div(rhoPhi,U)    Gauss limitedLinear 1;
38
39     // div(rhoCpPhi,T)   Gauss QUICK;
40     // div(rhoCpPhi,T)   Gauss linear;
41     // div(rhoCpPhi,T)   Gauss MUSCL;
42     div(rhoCpPhi,T)   Gauss vanLeer;
43     // div(rhoCpPhi,T)   Gauss upwind;
44     // div(rhoCpPhi,T)   Gauss limitedLinear 1;
45
46

```

```
47     div(phi,k)      Gauss limitedLinear 1;
48     div(phi,B)      Gauss limitedLinear 1;
49     div(B)          Gauss linear;
50     div(phi,nuTilda) Gauss limitedLinear 1;
51     div(phi,alpha)  Gauss linear;
52     div(phirb,alpha) Gauss linear;
53     div((muEff*dev(T(grad(U)))) Gauss linear;
54     div(phi,T)      Gauss vanLeer;
55 }
56
57 laplacianSchemes
58 {
59     default          Gauss linear corrected;
60 }
61
62 interpolationSchemes
63 {
64     default          linear;
65 }
66
67 snGradSchemes
68 {
69     default          corrected;
70 }
71
72 fluxRequired
73 {
74     default          no;
75     pcorr;
76     p_rgh;
77     "alpha.*";
78 }
79
80
81 // ***** //
```

C.1.13. system/fvSolution

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O p e r a t i o n | Version: 2.3.1 |
5  | \ \      / A n d           | Web: www.OpenFOAM.org |
6  | \ \ /      M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        fvSolution;
15 }
16 // ***** //
17
18 solvers
19 {
20     alpha
21     {
22         nAlphaSubCycles 2;
23         cAlpha          2;
24     }
25
26     pcorr
27     {
28         solver          PCG;
29         preconditioner
30         {
31             preconditioner GAMG;
32             tolerance      1e-05;
33             relTol         0;
34             smoother       GaussSeidel;
35             nPreSweeps     0;
36             nPostSweeps    2;
37             nFinestSweeps  2;
38             cacheAgglomeration on;
39             nCellsInCoarsestLevel 10;
40             agglomerator   faceAreaPair;
41             mergeLevels    1;
42         }
43         tolerance      1e-05;
44         relTol         0;
45         maxIter        100;
46     }

```

```
47
48 p_rgh
49 {
50     solver          GAMG;
51     tolerance       1e-07;
52     relTol          0.05;
53     smoother        GaussSeidel;
54     nPreSweeps      0;
55     nPostSweeps     2;
56     nFinestSweeps   2;
57     cacheAgglomeration on;
58     nCellsInCoarsestLevel 10;
59     agglomerator     faceAreaPair;
60     mergeLevels      1;
61 }
62
63 p_rghFinal
64 {
65     solver          PCG;
66     preconditioner
67     {
68         preconditioner GAMG;
69         tolerance       1e-07;
70         relTol          0;
71         nVcycles        2;
72         smoother        GaussSeidel;
73         nPreSweeps      0;
74         nPostSweeps     2;
75         nFinestSweeps   2;
76         cacheAgglomeration on;
77         nCellsInCoarsestLevel 10;
78         agglomerator     faceAreaPair;
79         mergeLevels      1;
80     }
81     tolerance       1e-07;
82     relTol          0;
83     maxIter         20;
84 }
85
86 U
87 {
88     solver          smoothSolver;
89     smoother        GaussSeidel;
90     tolerance       1e-08;
91     relTol          0.1;
92     nSweeps         1;
93 }
94
95 UFinal
```

```
96     {
97         $U;
98         tolerance      1e-08;
99         relTol         0;
100     }
101
102     T
103     {
104         solver          BICCG;
105         preconditioner  DILU;
106         tolerance      1e-7;
107         relTol         0;
108     }
109
110     TFinal
111     {
112         $T;
113         tolerance      1e-07;
114         relTol         0;
115     }
116
117     "(TIUik)"
118     {
119         solver          smoothSolver;
120         smoother        GaussSeidel;
121         tolerance      1e-06;
122         relTol         0.1;
123         nSweeps        1;
124     }
125
126     "(TIUik)Final"
127     {
128         solver          smoothSolver;
129         smoother        symGaussSeidel;
130         tolerance      1e-08;
131         relTol         0;
132     }
133 }
134
135 PIMPLE
136 {
137     nCorrectors      2;
138     nNonOrthogonalCorrectors 4;
139 }
140
141 relaxationFactors
142 {
143     fields
144     {
```

```
145     }
146     equations
147     {
148         "U.*"      0.8;
149         "T.*"      0.8;
150         "k.*"      0.8;
151     }
152 }
153
154 // ***** //
```

C.1.14. system/setFieldsDict

```

1  /*-----* C++ *-----*/
2  | ===== |
3  | \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O p e r a t i o n | Version: 2.3.0 |
5  | \ \      / A n d      | Web: www.OpenFOAM.org |
6  |  \ \ /      M a n i p u l a t i o n | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "system";
14     object        setFieldsDict;
15 }
16 // ***** //
17
18 defaultFieldValues
19 (
20     volScalarFieldValue alpha.air 0
21     volScalarFieldValue alpha.water 0
22     volScalarFieldValue alpha.oil 0
23     volVectorFieldValue U ( 0 0 0 )
24     volScalarFieldValue T 293
25 );
26
27 regions
28 (
29     zoneToCell
30     {
31         name air;
32         fieldValues
33         (
34             volScalarFieldValue alpha.water 0
35             volScalarFieldValue alpha.oil 1
36             volScalarFieldValue alpha.air 0
37             volScalarFieldValue T 293
38         );
39     }
40     zoneToCell
41     {
42         name oil;
43         fieldValues
44         (
45             volScalarFieldValue alpha.water 0
46             volScalarFieldValue alpha.oil 1

```

```
47         volScalarFieldValue alpha.air 0
48         volScalarFieldValue T 293.15
49     );
50 }
51 zoneToCell
52 {
53     name water;
54     fieldValues
55     (
56         volScalarFieldValue alpha.water 1
57         volScalarFieldValue alpha.oil 0
58         volScalarFieldValue alpha.air 0
59         volScalarFieldValue T 293.15
60     );
61 }
62 );
63
64
65 // ***** //
```

C.2. Sin flujo de agua

C.2.1. 0/U

```
1  /*-----* C++ *-----*/
2  | ===== |
3  | \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O p e r a t i o n      | Version: 2.3.0 |
5  | \ \      / A n d      | Web: www.OpenFOAM.org |
6  | \ \ /      M a n i p u l a t i o n      | |
7  /*-----*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format      ascii;
12     class      volVectorField;
13     location     "0";
14     object      U;
15 }
16 // ***** //
17
18 dimensions      [0 1 -1 0 0 0 0];
19
20 internalField    uniform (0 0 0);
21
22 boundaryField
23 {
24     oilIn
25     {
26         type      fixedValue;
27         value     uniform (0 0 3.96);
28     }
29     symmetry_1
30     {
31         type      symmetryPlane;
32     }
33     symmetry_2
34     {
35         type      symmetryPlane;
36     }
37     wall_combinar1
38     {
39         type      fixedValue;
40         value     uniform (0 0 0);
41     }
42     wall
43     {
```

```
44     type      fixedValue;
45     value     uniform (0 0 0);
46 }
47 combinar_1
48 {
49     type      zeroGradient;
50 }
51 combinar_2
52 {
53     type      zeroGradient;
54 }
55 wall_combinar2
56 {
57     type      fixedValue;
58     value     uniform (0 0 0);
59 }
60 waterIn
61 {
62     type      fixedValue;
63     value     uniform (0 0 0);
64 }
65 out
66 {
67     type      zeroGradient;
68 }
69 }
70
71
72 // ***** //
```

C.3. Variación flujo petroleo: Re 10.000

C.3.1. 0/U

```

1  /*-----*-- C++ --*-----*/
2  | ===== | |
3  | \ \      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      /  O p e r a t i o n  | Version:  2.3.0 |
5  | \ \      /  A n d      | Web:      www.OpenFOAM.org |
6  | \ \ \    /  M a n i p u l a t i o n  | |
7  /*-----*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volVectorField;
13     location      "0";
14     object        U;
15 }
16 // ***** //
17
18 dimensions      [0 1 -1 0 0 0 0];
19
20 internalField    uniform (0 0 0);
21
22 boundaryField
23 {
24     oilIn
25     {
26         type      fixedValue;
27         value      uniform (0 0 2);
28     }
29     symmetry_1
30     {
31         type      symmetryPlane;
32     }
33     symmetry_2
34     {
35         type      symmetryPlane;
36     }
37     wall_combinar1
38     {
39         type      fixedValue;
40         value      uniform (0 0 0);
41     }
42     wall
43     {

```

```
44     type      fixedValue;
45     value     uniform (0 0 0);
46 }
47 combinar_1
48 {
49     type      zeroGradient;
50 }
51 combinar_2
52 {
53     type      zeroGradient;
54 }
55 wall_combinar2
56 {
57     type      fixedValue;
58     value     uniform (0 0 0);
59 }
60 waterIn
61 {
62     type      fixedValue;
63     value     uniform (0 0 3.006);
64 }
65 out
66 {
67     type      zeroGradient;
68 }
69 }
70
71
72 // ***** //
```

C.4. Variación flujo petroleo: Re 20.000

C.4.1. 0/U

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      /  O p e r a t i o n  | Version:  2.3.0 |
5  | \ \      /  A n d      | Web:      www.OpenFOAM.org |
6  | \ \      /  M a n i p u l a t i o n  | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volVectorField;
13     location      "0";
14     object        U;
15 }
16 // ***** //
17
18 dimensions      [0 1 -1 0 0 0 0];
19
20 internalField    uniform (0 0 0);
21
22 boundaryField
23 {
24     oilIn
25     {
26         type      fixedValue;
27         value      uniform (0 0 4);
28     }
29     symmetry_1
30     {
31         type      symmetryPlane;
32     }
33     symmetry_2
34     {
35         type      symmetryPlane;
36     }
37     wall_combinar1
38     {
39         type      fixedValue;
40         value      uniform (0 0 0);
41     }
42     wall
43     {

```

```
44     type      fixedValue;
45     value     uniform (0 0 0);
46 }
47 combinar_1
48 {
49     type      zeroGradient;
50 }
51 combinar_2
52 {
53     type      zeroGradient;
54 }
55 wall_combinar2
56 {
57     type      fixedValue;
58     value     uniform (0 0 0);
59 }
60 waterIn
61 {
62     type      fixedValue;
63     value     uniform (0 0 3.006);
64 }
65 out
66 {
67     type      zeroGradient;
68 }
69 }
70
71
72 // ***** //
```

C.5. Variación flujo petroleo: Re 25.000

C.5.1. 0/U

```

1  /*----- C++ -----*/
2  | ===== |
3  | \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O p e r a t i o n      | Version: 2.3.0 |
5  | \ \      / A n d      | Web: www.OpenFOAM.org |
6  | \ \ \    / M a n i p u l a t i o n      | |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volVectorField;
13     location      "0";
14     object        U;
15 }
16 // ***** //
17
18 dimensions      [0 1 -1 0 0 0 0];
19
20 internalField    uniform (0 0 0);
21
22 boundaryField
23 {
24     oilIn
25     {
26         type      fixedValue;
27         value      uniform (0 0 5);
28     }
29     symmetry_1
30     {
31         type      symmetryPlane;
32     }
33     symmetry_2
34     {
35         type      symmetryPlane;
36     }
37     wall_combinar1
38     {
39         type      fixedValue;
40         value      uniform (0 0 0);
41     }
42     wall
43     {

```

```
44     type      fixedValue;
45     value     uniform (0 0 0);
46 }
47 combinar_1
48 {
49     type      zeroGradient;
50 }
51 combinar_2
52 {
53     type      zeroGradient;
54 }
55 wall_combinar2
56 {
57     type      fixedValue;
58     value     uniform (0 0 0);
59 }
60 waterIn
61 {
62     type      fixedValue;
63     value     uniform (0 0 3.006);
64 }
65 out
66 {
67     type      zeroGradient;
68 }
69 }
70
71
72 // ***** //
```

C.6. Variación flujo agua: Re 2.000

C.6.1. 0/U

```

1  /*-----*-- C++ --*-----*/
2  | ===== | |
3  | \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O p e r a t i o n | Version: 2.3.0 |
5  | \ \      / A n d | Web: www.OpenFOAM.org |
6  | \ \ /      M a n i p u l a t i o n | |
7  /*-----*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volVectorField;
13     location      "0";
14     object        U;
15 }
16 // ***** //
17
18 dimensions      [0 1 -1 0 0 0 0];
19
20 internalField    uniform (0 0 0);
21
22 boundaryField
23 {
24     oilIn
25     {
26         type      fixedValue;
27         value      uniform (0 0 3);
28     }
29     symmetry_1
30     {
31         type      symmetryPlane;
32     }
33     symmetry_2
34     {
35         type      symmetryPlane;
36     }
37     wall_combinar1
38     {
39         type      fixedValue;
40         value      uniform (0 0 0);
41     }
42     wall
43     {

```

```
44     type      fixedValue;
45     value     uniform (0 0 0);
46 }
47 combinar_1
48 {
49     type      zeroGradient;
50 }
51 combinar_2
52 {
53     type      zeroGradient;
54 }
55 wall_combinar2
56 {
57     type      fixedValue;
58     value     uniform (0 0 0);
59 }
60 waterIn
61 {
62     type      fixedValue;
63     value     uniform (0 0 2.004);
64 }
65 out
66 {
67     type      zeroGradient;
68 }
69 }
70
71
72 // ***** //
```

C.7. Variación flujo agua: Re 4.000

C.7.1. 0/U

```

1  /*-----*-- C++ --*-----*/
2  | ===== | |
3  | \\      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \\      / O p e r a t i o n      | Version: 2.3.0 |
5  | \\      / A n d      | Web: www.OpenFOAM.org |
6  |  \\/      M a n i p u l a t i o n      | |
7  /*-----*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format      ascii;
12     class      volVectorField;
13     location      "0";
14     object      U;
15 }
16 // ***** //
17
18 dimensions      [0 1 -1 0 0 0 0];
19
20 internalField      uniform (0 0 0);
21
22 boundaryField
23 {
24     oilIn
25     {
26         type      fixedValue;
27         value      uniform (0 0 3);
28     }
29     symmetry_1
30     {
31         type      symmetryPlane;
32     }
33     symmetry_2
34     {
35         type      symmetryPlane;
36     }
37     wall_combinar1
38     {
39         type      fixedValue;
40         value      uniform (0 0 0);
41     }
42     wall
43     {

```

```
44     type      fixedValue;
45     value     uniform (0 0 0);
46   }
47   combinar_1
48   {
49     type      zeroGradient;
50   }
51   combinar_2
52   {
53     type      zeroGradient;
54   }
55   wall_combinar2
56   {
57     type      fixedValue;
58     value     uniform (0 0 0);
59   }
60   waterIn
61   {
62     type      fixedValue;
63     value     uniform (0 0 4.008);
64   }
65   out
66   {
67     type      zeroGradient;
68   }
69 }
70
71
72 // ***** //
```

C.8. Variación flujo agua: Re 5.000

C.8.1. 0/U

```

1  /*-----* C++ *-----*/
2  | ===== |
3  | \ \      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
4  | \ \      / O p e r a t i o n      | Version: 2.3.0 |
5  | \ \      / A n d      | Web: www.OpenFOAM.org |
6  | \ \ \    / M a n i p u l a t i o n      | |
7  /*-----*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format        ascii;
12     class         volVectorField;
13     location      "0";
14     object        U;
15 }
16 // ***** //
17
18 dimensions      [0 1 -1 0 0 0 0];
19
20 internalField    uniform (0 0 0);
21
22 boundaryField
23 {
24     oilIn
25     {
26         type      fixedValue;
27         value      uniform (0 0 3);
28     }
29     symmetry_1
30     {
31         type      symmetryPlane;
32     }
33     symmetry_2
34     {
35         type      symmetryPlane;
36     }
37     wall_combinar1
38     {
39         type      fixedValue;
40         value      uniform (0 0 0);
41     }
42     wall
43     {

```

```
44     type      fixedValue;
45     value     uniform (0 0 0);
46 }
47 combinar_1
48 {
49     type      zeroGradient;
50 }
51 combinar_2
52 {
53     type      zeroGradient;
54 }
55 wall_combinar2
56 {
57     type      fixedValue;
58     value     uniform (0 0 0);
59 }
60 waterIn
61 {
62     type      fixedValue;
63     value     uniform (0 0 5.01);
64 }
65 out
66 {
67     type      zeroGradient;
68 }
69 }
70
71
72 // ***** //
```

Bibliografía

Patrik Andersson. Tutorial multiphaseInterFoam. *Report*, pág. 21, 2010.

a. M. Ardekani, D. D. Joseph, D. Dunn-Rankin, y R. H. Rangel. Particle-wall collision in a viscoelastic fluid. *Journal of Fluid Mechanics*, 633:475, 2009. ISSN 0022-1120. doi:10.1017/S0022112009990632.

I a Inform Atica. Algebra Numérica. 2004.

Sethuraman Babu. Simulation of three-dimensional incompressible flows with free-surfaces including fluid-structure interaction and microgravity flows. 1994.

S. Balachandar y John K. Eaton. Turbulent Dispersed Multiphase Flow. *Annual Review of Fluid Mechanics*, 42(1):111–133, 2010. ISSN 0066-4189. doi:10.1146/annurev.fluid.010908.165243.

Tim Behrens. OpenFOAM's basic solvers for linear systems of equations. *Report*, pág. 18, 2009.

Eberhard Bodenschatz, Werner Pesch, y Guenter Ahlers. RECENT DEVELOPMENTS IN RAYLEIGH -BENARD CONVECTION. págs. 709–778, 2000.

Richard L. Burden y John Douglas Faires. *Numerical Analysis*. 2011. ISBN 0538733519.
URL [http://www.lavoisier.fr/livre/notice.asp?ouvrage=1248244\\$%5Cdelimiter"026E30F\\$nhhttp://books.google.es/books/about/Numerical_Analysis.html?id=zXnSxY9G2JgC&pgis=1](http://www.lavoisier.fr/livre/notice.asp?ouvrage=1248244$%5Cdelimiter).

Goong Chen, Qingang Xiong, Phillip J. Morris, Eric G. Paterson, Alexey Sergeev, y Yi-Ching Wang. OpenFOAM for Computational Fluid Dynamics. *Notices of the*

- American Mathematical Society*, 61(4):354, 2014. ISSN 0002-9920. doi:10.1090/noti1095. URL <http://www.ams.org/jourcgi/jour-getitem?pii=noti1095>.
- Clayton T Crowe, Martin Sommerfeld, y Yutaka Tsuji. *Multiphase Flows with Droplets and Particles*. 1998. ISBN 0849394694. doi:10.1201/b11103-4. URL <http://books.google.de/books?id=CioXotlGMiYC>.
- Primer Cuatrimestre. Método de descenso más rápido y del gradiente conjugado para la resolución de sistemas lineales. págs. 1–2, 2007.
- Santiago Márquez Damián. Description and utilization of interFoam multiphase solver. *Report*, págs. 1–64, 2004.
- Ministerio del Medio Ambiente. Norma de emisión para centrales termoeléctricas. 2013.
- Wendt J F. *Computational Fluid Dynamics*. 1992. ISBN 978-3-540-85055-7. doi:10.1007/978-3-540-85056-4.
- J. Farmer, L. Martinelli, y A. Jameson. A Fast Multigrid Method for Solving Incompressible Hydrodynamic Problems with Free Surfaces. 1993.
- Joel H. Ferziger y Milovan Peric. *Computational Methods for Fluid Dynamics*. 2002. ISBN 3540420746.
- Rodney O. Fox. Large-Eddy-Simulation Tools for Multiphase Flows. *Annual Review of Fluid Mechanics*, 44(1):47–76, 2012. ISSN 0066-4189. doi:10.1146/annurev-fluid-120710-101118.
- Christophe Geuzaine y Jean François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. ISSN 00295981. doi:10.1002/nme.2579.
- Martin Gramlich. Numerical Investigations of the Unsteady Flow in the Stuttgart Swirl Generator with OpenFOAM. *Thesis*, págs. 1–75, 2012.
- User Guide. *Open FOAM*. December. FOAM Foundation, 2014.

- C.W Hirt y B.D Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, 1981. ISSN 00219991. doi:10.1016/0021-9991(81)90145-5. URL <http://www.sciencedirect.com/science/article/pii/0021999181901455>.
- Frank P Incropera, David P DeWitt, Theodore L Bergman, y Adrienne S Lavine. *Fundamentals of Heat and Mass Transfer*, tomo 6th de *Dekker Mechanical Engineering*. John Wiley & Sons, 2007. ISBN 9780471457282. doi:10.1016/j.applthermaleng.2011.03.022. URL http://www.osti.gov/energycitations/product.biblio.jsp?osti_{_}id=6008324.
- Hrvoje Jasak y Hrvoje Jasak. Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows. *Fluid Dynamics*, M(June), 1996.
- R. Johnson. Fluid Mechanics of Compound Multiphase Drops and Bubbles. *Annual Review of Fluid Mechanics*, 17(1):298–320, 1985. ISSN 00664189. doi:10.1146/annurev.fluid.17.1.298.
- Nina Gall Jørgensen. Implementation of a turbulent inflow boundary condition for LES based on a vortex method. *Report*, págs. 1–26, 2012.
- Djamel Lakehal. Turbulence et Ecoulements Multiphasiques. *Interfaces*.
- Djamel Lakehal. LES of a Vertical Mixing Layer Laden with Bubbles. *Journal of Turbulence*, 2002.
- Delphine Leroux. Getting started with OpenFOAM 1.4. *Report*, pág. 59, 2007.
- Detlef Lohse y Ke-Qing Xia. Small-Scale Properties of Turbulent Rayleigh-Bénard Convection. *Annual Review of Fluid Mechanics*, 42(1):335–364, 2010. ISSN 0066-4189. doi:10.1146/annurev.fluid.010908.165152.
- H Lomax, TH Pulliam, DW Zingg, y TA Kowalewski. *Fundamentals of Computational Fluid Dynamics*. 2002. doi:10.1115/1.1483340.
- Nasreddine Ouertatani, Nader Ben Cheikh, Brahim Ben Beya, y Taieb Lili. Numerical simulation of two-dimensional Rayleigh-Bénard convection in an enclosure. *El Sevier*, págs. 464–470, 2008.

- Olle Penttinen. A pimpleFoam tutorial for channel flow, with respect to different LES models. *Change*, págs. 1–23, 2011.
- S.B. Pope. *Turbulent flows*. 2000. ISBN 0521598869. doi:10.1017/CBO9780511840531.
- Jordi Casacuberta Puig. *OpenFOAM GUIDE FOR BEGINNERS*. 2014.
- I. H. Revilla. Un Análisis del Método de Gradiente Conjugado. *Desconhecido*, 1(2):1–10, 1971.
- Henrik Rusche. Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions. *Direct*, (December), 2002. doi:10.1145/1806799.1806850.
- J. a. Sethian y Peter Smereka. Level Set Methods for Fluid Interfaces. *Annual Review of Fluid Mechanics*, 35(1):341–372, 2003. ISSN 0066-4189. doi:10.1146/annurev.fluid.35.101101.161105. URL <http://www.annualreviews.org/doi/abs/10.1146/annurev.fluid.35.101101.161105>.
- Mark Sussman y Elbridge Gerry Puckett. A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. *Journal of Computational Physics*, 162(2):301–337, 2000. ISSN 00219991. doi:10.1006/jcph.2000.6537. URL <http://www.sciencedirect.com/science/article/pii/S0021999100965379>.
- Introduction To y Computational Fluid. *Introduction to Computational Fluid Dynamics*, tomo 44. 2005. ISBN 9780131274983. doi:10.2514/1.22547. URL <http://www.ncbi.nlm.nih.gov/pubmed/6686412>.
- O Ubbink. Numerical prediction of two fluid systems with sharp interfaces. *Splash*, (January 1997):69, 1997. ISSN 00219991. doi:10.1145/1774088.1774119. URL <http://powerlab.fsb.hr/ped/kturbo/OpenFOAM/docs/OnnoUbbinkPhD.pdf>.
- Javier Achury Varila, Carlos Duque Daza, y Carlos Galeano Urueña. Two-Dimensional Simulation of an Unstable Combustion System. 18:105–119, 2010.

- H.K: Versteeg y W. Malalasekera. An Introduction to Computational Fluid Dynamics - The Finite Volume Method. 1995. URL <http://www.mie.utoronto.ca/labs/mussl/cfd20.pdf> \delimiter"026E30F\$nh[http://books.google.com/books?hl=en&lr=&id=RvBZ-UMpGzIC&oi=fnd&pg=PP12&dq=An+Introduction+to+Computational+Fluid+Dynamics+-+The+Finite+Volume+Method&ots=ulvXu9CjGd&sig=hliwjbhQnS5p48PHdFs-NsFQnfU\\$](http://books.google.com/books?hl=en&lr=&id=RvBZ-UMpGzIC&oi=fnd&pg=PP12&dq=An+Introduction+to+Computational+Fluid+Dynamics+-+The+Finite+Volume+Method&ots=ulvXu9CjGd&sig=hliwjbhQnS5p48PHdFs-NsFQnfU$) \delimiter"026E30F\$nh<http://bo>.
- H.K: Versteeg y W. Malalasekera. *Introduction to Computational Fluid Dynamics*, tomo 44. 2005. ISBN 9780131274983. doi:10.2514/1.22547. URL <http://www.ncbi.nlm.nih.gov/pubmed/6686412>.
- H G Weller y G Tabor. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6):620–631, 1998. ISSN 08941866. doi:10.1063/1.168744.
- A Wirth. *A Guided Tour Through Buoyancy Driven Flows and Mixing*. 2014.
- et al Yunus A. *Transferencia de calor y masa*. 2007. ISBN 9780521191333. doi:10:0-8400-5444-0.
- L. X. Zhou. Advances in studies on two-phase turbulence in dispersed multiphase flows. *International Journal of Multiphase Flow*, 36(2):100–108, 2010. ISSN 03019322. doi:10.1016/j.ijmultiphaseflow.2009.02.011. URL <http://dx.doi.org/10.1016/j.ijmultiphaseflow.2009.02.011>.