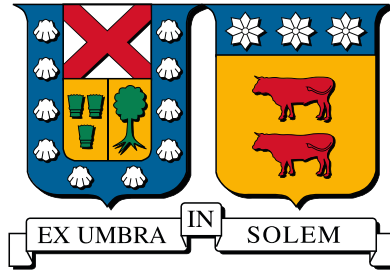


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE OBRAS CIVILES
SANTIAGO - CHILE



Simulación Numérica de Transporte de Fondo con Doble Acoplamiento en un Canal Turbulento

Memoria de titulación presentada por
Joaquín Neira Jaeger

Como requisito parcial para optar al título de
Ingeniero Civil

Profesor Guía
Patricio Moreno Casas

Profesora Correferente
Vivian Aranda Núñez

Mayo de 2021

Dedicatoria

A mis padres y hermanos, por quererme y educarme.

A Rocío, por apoyarme en este largo proceso.

A mis amigos, por todo lo compartido.

Al profesor Patricio, por su dedicación, motivación y disposición.

Y al pueblo de Chile, que quiere cambios.

Agradecimientos

Agradezco a la Universidad de Los Andes, por permitirme utilizar sus instalaciones y equipos durante la realización de esta memoria.

Al profesor Patricio Moreno, por haberme dado la oportunidad de trabajar con él, y aprender de él.

Al profesor José Abell, por enseñarme todo lo relacionado con Linux y programación de alto rendimiento.

Powered@NLHPC: Esta investigación fue parcialmente apoyada por la infraestructura de supercómputo del NLHPC (ECM-02)

Resumen

El transporte de sedimentos en canales y ríos es el proceso mediante el cual partículas de distintos tamaños, inmersas en un fluido, son desplazadas producto de las fuerzas ejercidas por este último. El estudio de este fenómeno durante las últimas décadas ha tenido un enfoque mayormente experimental, dejando la teoría en un segundo plano, por lo cual el proceso aún no es comprendido en su totalidad.

Existen dos formas de transporte de sedimentos. La primera es el transporte en suspensión, en el cual las partículas de menor tamaño se mueven suspendidas en el fluido, lejos del lecho. La segunda es el transporte de fondo, en el cual las partículas de mayor tamaño, presentes en el fondo del cauce, son movidas deslizando, rodando, o incluso dando saltos, lo que se conoce como saltación. Esta última forma de transporte es el objeto de estudio de este trabajo.

Para realizar el estudio del arrastre de fondo, se ha desarrollado un modelo computacional en OpenFOAM, consistente de una fase continua (fluido), que simula un flujo turbulento, y una fase dispersa (partículas), que es dividida en tres submodelos: vuelo libre (aplicación de fuerzas hidrodinámicas ejercidas por el flujo sobre la partícula), colisión con el fondo, y colisión entre partículas. Se ha considerado, además, el acoplamiento de dos vías entre las fases continua y dispersa, por lo cual las partículas inmersas en el fluido también afectan su comportamiento.

El modelo se ha validado utilizando valores experimentales presentes en la literatura, y ha permitido obtener resultados respecto al efecto, tanto en el flujo como en las partículas, de aumentar la concentración de partículas presentes en el flujo. Por último, se han estudiado los recursos computacionales utilizados en cada caso simulado.

Palabras clave: arrastre de fondo, doble acoplamiento, flujo turbulento, simulación.

Abstract

The transport of sediments in canals and rivers is the process by which particles of different sizes, immersed in a fluid, are displaced due to the forces exerted by the latter. The study of this phenomenon during the last decades has had a mostly experimental approach, leaving the theory in the background, so the process is not yet fully understood.

There are two forms of sediment transport. The first is suspended load, in which smaller particles move suspended in the fluid, away from the bed. The second is the bed load, in which the larger particles, present in the bottom of the channel, are moved by sliding, rolling, or even hopping, what is known as saltation. This last form of transport is the object of study of this work.

To carry out the study of bed load, a computational model has been developed in OpenFOAM, consisting of a continuous phase (fluid), which simulates a turbulent flow, and a dispersed phase (particles), which is divided into three submodels: free flight (application of hydrodynamic forces exerted by the flow on the particle), collision with the bottom, and collision between particles. In addition, the two-way coupling between the continuous and dispersed phases has been considered, whereby the particles immersed in the fluid also affect their behavior.

The model has been validated using experimental values present in the literature, and has allowed us to obtain results regarding the effect, both in the flow and in the particles, of increasing the concentration of particles present in the flow. Finally, the computational resources used in each simulated case have been studied.

Keywords: bed load, two-way coupling, turbulent flow, simulation.

Glosario

Los símbolos en negrita corresponden a vectores

Símbolos romanos (minúsculas)

d_p	Diámetro de la partícula
e	Coeficiente de restitución
f	Coeficiente de fricción
g	Aceleración de gravedad
h	Altura del canal
$\tilde{k}$	Energía turbulenta
k_n	Rigidez en la dirección normal
k_t	Rigidez en la dirección tangencial
m	Masa de la partícula
m^*	Masa añadida
$\hat{n}$	Vector unitario normal al plano de colisión
n	Vector unitario normal a la superficie
p	Presión
t	Vector unitario en la dirección tangencial
t	Tiempo
t_{final}	Tiempo final de la simulación
u	Velocidad del fluido
u_0	Velocidad inicial del fluido
u_p	Velocidad de la partícula
u_r	Velocidad relativa de la partícula respecto al flujo
u_s	Velocidad del sólido

u'	Raíz cuadrática media de las fluctuaciones de velocidad del flujo
u^*	Velocidad de fricción
u^+	Velocidad promedio del flujo, normalizada por la velocidad de fricción
u'_i	Fluctuación de velocidad del flujo en la dirección i
$\mathbf{u}_{pc}$	Velocidad del punto de contacto de la partícula durante colisión
$\mathbf{u}_{pct}$	Componente tangencial de la velocidad del punto de contacto de la partícula durante colisión
w	Ley de la estela
y	Distancia desde la pared
y^+	Distancia desde la pared, normalizada por la escala de longitud viscosa

Símbolos romanos (mayúsculas)

B	Constante en la ley logarítmica de la pared
C_1, C_2, C_3	Coeficientes para el cálculo de C_T (ver Tabla A.1)
C_d	Coeficiente de arraste
C_l	Coeficiente de sustentación
C_M	Coeficiente de Magnus
C_m	Coeficiente de masa virtual
C_p	Concentración de partículas
C_T	Coeficiente de torque
Co	Número de Courant
E	Módulo de Young
$\mathbf{F}$	Fuerza aplicada sobre la partícula
$\mathbf{F}_{BS}$	Fuerza de Basset o fuerza histórica de Basset
$\mathbf{F}_{DR}$	Fuerza de arrastre o <i>drag</i>
$\mathbf{F}_{FA}$	Fuerza de aceleración del fluido
$\mathbf{F}_{LF}$	Fuerza de sustentación o <i>lift</i>

$\mathbf{F}_{MG}$	Fuerza de Magnus
$\mathbf{F}_{nij}$	Fuerza normal ejercida por la partícula j sobre la partícula i
$\mathbf{F}_{partícula}$	Sumatoria de fuerzas aplicadas sobre la partícula
$\mathbf{F}_{SW}$	Fuerza de peso sumergido
$\mathbf{F}_{tij}$	Fuerza tangencial ejercida por la partícula j sobre la partícula i
$\mathbf{F}_{VM}$	Fuerza de masa virtual
$\mathbf{F}_{VM}^*$	Primer término del lado derecho la Ec. (19)
$\mathbf{G}$	Velocidad relativa entre partículas
$\mathbf{G}_{ct}$	Velocidad relativa en el punto de contacto, en la dirección tangencial
H	Módulo de corte
H	Altura de salto de partícula
$\mathbf{I}$	Tensor identidad de segundo orden
$\tilde{I}$	Intensidad de turbulencia
$\tilde{I}_{máx}$	Intensidad de turbulencia máxima
I	Momento de inercia de la partícula
$\mathbf{J}$	Fuerza de impulso sobre la partícula
J_n	Fuerza de impulso sobre la partícula en la dirección normal
J_t	Fuerza de impulso sobre la partícula en la dirección tangencial
L	Longitud de salto de partícula
Q_m	Flujo másico
Re	Número de Reynolds
Re_R	Número de Reynolds de rotación
S	Superficie del sólido
$\mathbf{T}$	Torque aplicado sobre la partícula
U	Magnitud de la velocidad promedio del flujo en el tiempo

V Volumen del sólido

Símbolos griegos

α Coeficiente asociado al coeficiente de restitución, e , modelo de colisión de esfera blanda

δ_v Escala de longitud viscosa

δ_n Desplazamiento en la dirección normal

δ_t Desplazamiento en la dirección tangencial

η_{nj} Amortiguamiento en la dirección normal

η_{tj} Amortiguamiento en la dirección tangencial

θ_b Ángulo de rotación del vector unitario normal

$\tilde{\kappa}$ Número de onda, relacionado con la escala de longitud del vortice

κ Constante de von Kármán

μ Viscosidad dinámica del fluido

ν Viscosidad cinemática del fluido

Π Parámetro de fuerza de la estela

ρ Densidad del fluido

ρ_p Densidad de la partícula

ρ_s Densidad del sólido

σ Razón de Poisson

τ Esfuerzo de corte

τ_b Esfuerzo de corte sobre la partícula

τ_w Esfuerzo de corte en la pared

τ_{cr} Esfuerzo de corte crítico

θ_{in} Ángulo de colisión

ξ Variable auxiliar

ω_p Velocidad angular de la partícula

ω_r Velocidad angular relativa de la partícula respecto al flujo

Operadores matemáticos

Δ Operador incremento

$\frac{\partial}{\partial t}, \frac{d}{dt}$ Operador diferencial respecto al tiempo

$\frac{D}{Dt}$ Operador de derivada material

∇ Operador diferencial vectorial

Índice

Agradecimientos	I
Resumen	II
Abstract	III
Glosario	IV
1. Introducción	1
1.1. Objetivo general	2
1.2. Objetivos específicos	2
2. Marco Teórico	3
2.1. Modelamiento de la fase continua o transportadora	3
2.1.1. Ley de la pared	6
2.1.2. Intensidad de turbulencia	7
2.2. Modelamiento de la fase dispersa	8
2.3. Submodelo de fuerzas sobre la fase dispersa	8
2.3.1. Fuerza de peso sumergido	9
2.3.2. Fuerza de arrastre o <i>drag</i>	9
2.3.3. Fuerza de sustentación o <i>lift</i>	10
2.3.4. Efecto de masa virtual o masa añadida	10
2.3.5. Fuerza de aceleración del fluido	10
2.3.6. Fuerza de Magnus	11
2.3.7. Fuerza de Basset	11
2.4. Submodelo de vuelo libre	11
2.5. Submodelos de colisión	12
2.5.1. Colisión con el fondo	13
2.5.1.1. Modelo de esfera dura	13

2.5.1.2.	Modelo estocástico de colisión o de rugosidad del lecho	14
2.5.2.	Colisión entre partículas	15
2.5.2.1.	Modelo de esfera blanda (DSEM)	15
2.6.	Acoplamiento de la fase transportadora con la fase dispersa	17
2.7.	Información experimental para validación	18
3.	Simulaciones	21
3.1.	Hardware utilizado	21
3.2.	Simulación en OpenFOAM	21
3.2.1.	Descripción del <i>solucionador pimpleLPTFoam</i>	22
3.2.2.	Simulación numérica del flujo	23
3.2.3.	Propiedades de las partículas	25
3.3.	Obtención de resultados y análisis	25
4.	Resultados	27
4.1.	Validación de la fase continua	27
4.2.	Validación del modelo de saltación	27
4.3.	Estudio con concentraciones mayores	28
4.3.1.	Efecto del doble acoplamiento sobre la velocidad del flujo	29
4.3.2.	Efecto del doble acoplamiento sobre la turbulencia	31
4.3.3.	Flujo másico	33
4.3.4.	Costo computacional	35
5.	Discusión y Análisis	36
5.1.	Validación de la fase continua	36
5.2.	Validación del modelo de saltación	36
5.3.	Estudio con concentraciones mayores	37
5.3.1.	Efecto del doble acoplamiento sobre la velocidad	37
5.3.2.	Efecto del doble acoplamiento sobre la turbulencia	38
5.3.3.	Flujo másico	38

5.3.4. Costo computacional	39
6. Conclusiones	40
Referencias	42
Anexos	44
A. Tablas	A.1
B. Modelos de Colisión	B.1
B.1. Modelo de esfera dura	B.1
B.2. Modelo de esfera blanda	B.4
B.2.1. Determinación de la rigidez	B.5
B.2.2. Determinación del amortiguamiento	B.6
C. Algoritmo PIMPLE	C.1
C.1. Algoritmo SIMPLE	C.2
C.2. Algoritmo PISO	C.2
C.3. Algoritmo PIMPLE	C.2
D. Archivos OpenFOAM	D.1
D.1. Solver <code>pimpleLPTFoam</code>	D.1
D.2. Simulaciones	D.2
D.3. Código de partículas	D.3
D.4. Código fuente	D.5
D.4.1. Archivos <i>solucionador</i> <code>pimpleLPTFoam</code>	D.5
D.4.2. Archivos de simulación	D.12

Índice de Figuras

Figura 2.1. Esquema de desarrollo del modelo de transporte de fondo.	3
Figura 2.2. Esquema de los flujos (a) laminar y (b) turbulento.	4
Figura 2.3. Esquema del método LES en términos de energía. κ es el número de onda, relacionado con la escala de longitud del vórtice, y $E(\kappa)$ la energía asociada. Con este modelo, las escalas energéticas mayores son calculadas, mientras que aquellas menores son modeladas.	5
Figura 2.4. Esquema de la Ley de la pared.	6
Figura 2.5. Esquema de la intensidad de turbulencia.	7
Figura 2.6. Esquema de la etapa de vuelo libre en el modelo de saltación. u_i es la velocidad lineal de la partícula en la dirección i	12
Figura 2.7. Esquema del modelo de colisión estocástica de García y Niño (1992).	15
Figura 2.8. Esquema del modelo de esfera blanda.	15
Figura 2.9. Vías de acoplamiento entre la fase continua y la fase dispersa.	17
Figura 2.10. Interacción entre fases de acuerdo a concentración de partículas y etapa de transporte según Elghobashi (1994). El acoplamiento de dos vías puede producir o disipar momentum según la <i>etapa de transporte</i>	18
Figura 3.1. Diagrama de flujo del <i>solucionador</i> pimpleLPTFoam.	23
Figura 3.2. Detalle de la malla utilizada en el estudio. En (a) se puede observar el aumento del refinamiento hacia las paredes superior e inferior. En (b) se ve una esquina de la malla, donde se puede apreciar el espaciado equidistante de las celdas en los ejes horizontales. Las paredes superior e inferior se utilizan condiciones de no deslizamiento, mientras que en las demás se imponen condiciones cíclicas (entre paredes opuestas).	24
Figura 3.3. Flujo turbulento simulado en OpenFOAM. Se puede ver el efecto de la turbulencia generada por las paredes superior e inferior, la cual es capturada gracias al mayor refinamiento existente en esa zona. El flujo circula en sentido del eje x , y los colores mostrados corresponden a la magnitud de la velocidad del flujo.	25
Figura 3.4. Esquema de los datos obtenidos en un salto de partícula. H_i y L_i corresponden a la altura máxima y la longitud del salto i , respectivamente. Además, se obtiene la velocidad de la partícula $\mathbf{u}_{p,t}$ en cada punto de la trayectoria.	26
Figura 4.1. Comparación del flujo turbulento simulado en OpenFOAM con la Ley de la Pared. La línea negra representa la altura en que ocurre el arrastre de fondo ($y = 4d_p$).	27

Figura 4.2. Efecto de doble acoplamiento sobre la velocidad del flujo. No se observa un patrón definido con el aumento de la concentración de partículas.	29
Figura 4.3. Velocidad promedio del flujo en $y = 2d_p$ para cada concentración estudiada.	30
Figura 4.4. Velocidad promedio del flujo para cada concentración estudiada.	31
Figura 4.5. Perfil de intensidad de turbulencia para cada concentración de partículas. Se ha limitado el eje de las abscisas para mejorar la resolución del gráfico.	32
Figura 4.6. Intensidad de turbulencia máxima para cada concentración de partículas estudiada. . .	33
Figura 4.7. Comparación del flujo másico en el canal para cada concentración de partículas estudiada, para las situaciones con y sin doble acoplamiento. Se indica, además, la diferencia (en porcentaje) para cada par de datos considerando como base la situación con acoplamiento.	34
Figura 4.8. Fracción del tiempo utilizado por las etapas de resolución de flujo, movimiento de partículas y colisiones para cada concentración estudiada. Se indica, además, el tiempo total de cada simulación, en minutos.	35
Figura B.1. Relación entre el coeficiente α y el coeficiente de restitución e	B.6
Figura D.1. Árbol de archivos utilizados para las simulaciones del modelo de saltación implementado en OpenFOAM.	D.1
Figura D.2. Árbol de archivos utilizados para la implementación del <i>solucionador</i> <code>pimpleLPTFoam</code> en OpenFOAM.	D.2
Figura D.3. Árbol de archivos de la librería <code>intermediate</code> de OpenFOAM que han sido modificados para la implementación del modelo de saltación. Solo se muestran los archivos creados y/o modificados.	D.3

Índice de Tablas

Tabla 2.1. Configuración de los experimentos sobre saltación realizados por distintos autores. H y L corresponden a las altura y longitud de salto, respectivamente, y $ \mathbf{u}_p $ es la velocidad de la partícula.	19
Tabla 2.2. Valores para el coeficiente de restitución (e) y fricción (f) sugeridos por distintos autores.	20
Tabla 3.1. Características del hardware de la partición SLIMS del clúster Leftrarú.	21
Tabla 4.1. Comparación de los resultados obtenidos en la simulación en OpenFOAM con los obtenidos por Niño y García (1998). Los resultados de altura y longitud se han normalizado con d_p , y los de velocidad con u^* . Se han obtenido resultados de más de 100 saltos para 9 partículas.	28
Tabla 4.2. Cantidad de partículas según concentración.	28
Tabla A.1. Coeficientes para el cálculo de C_T	A.1
Tabla A.2. Promedio y desviación estándar de altura, longitud y velocidad de saltos obtenidos para diez partículas simuladas en OpenFOAM. Los resultados de altura y longitud se han normalizado con d_p , y los de velocidad con u^* . Para cada partícula se han considerado más de 100 saltos.	A.1

CAPÍTULO 1

INTRODUCCIÓN

El transporte de sedimentos es el proceso mediante el cual partículas de distintos tamaños que se encuentran inmersas en un fluido son desplazadas debido a las fuerzas ejercidas por este último. Este mecanismo es de especial interés para muchas y muy distintas disciplinas: geología, hidrología, oceanografía e ingeniería civil, entre otros (Moreno, 2014; Wiberg y Smith, 1985). En ingeniería civil, en particular, el fenómeno se puede observar en distintos ámbitos. Desde el punto de vista físico, el movimiento de sedimentos puede generar cambios en los lechos de ríos, la disminución de la capacidad de embalses, la socavación de pilas de puentes y la erosión de pavimentos, entre otros (Julien, 2010).

Además, el transporte de sedimentos lleva consigo el movimiento de agentes químicos que pueden afectar ecosistemas y la salud de la población en general (Julien, 2010). Una mejor comprensión del fenómeno de transporte permite contener mejor estos efectos, sobre todo cuando conllevan un elevado costos en términos económicos y de tiempo (González, 2008).

El transporte de sedimentos se puede dividir en dos categorías principales según el tamaño de partícula y la turbulencia del flujo (Julien, 2010). En flujos de alta turbulencia, las partículas de menor tamaño como limos y arcillas se transportan en *suspensión*, lejos del lecho, mientras las partículas de mayor tamaño como arena y grava se mueven deslizando, rodando y saltando en una capa delgada cercana al fondo, lo que se denomina *transporte de fondo*.

En el transporte de fondo, cada partícula pasará de rodar y deslizarse a seguir trayectorias balísticas según la magnitud de las fuerzas ejercidas por el fluido, realizando pequeños saltos durante todo su movimiento, lo que se conoce como *saltación* (Wiberg y Smith, 1985).

Para el estudio del transporte de sedimentos, se han desarrollado modelos desde tres puntos de vista distintos: empíricos, conceptuales y físicos. El uso de cada uno de ellos depende del tipo de resultados que se desee obtener (Merritt et al., 2003). En esta memoria se estudia el proceso de saltación con un enfoque físico, es decir, se centra en la física alrededor de cada partícula de sedimento.

Lo anterior se debe a que a pesar de existir modelos físicos complejos y con resultados cercanos a la observación experimental en cuanto a altura y longitud promedio de los saltos de partículas individuales (Moreno y Bombardelli, 2012; Niño y García, 1994; Wiberg y Smith, 1985), pocos de ellos consideran la interacción entre las fases transportadora y dispersa (Moreno, 2014), con lo cual el conocimiento del proceso sigue siendo limitado.

Con las herramientas actuales, es posible desarrollar modelos tridimensionales con flujos turbulentos resueltos numéricamente, mediante un enfoque euleriano (utilizando volúmenes de control), y no modelados con representaciones como la ley de la pared, lo cual permite obtener resultados más cercanos a la realidad. Para la fase dispersa (partículas), en cambio, se utiliza un enfoque lagrangiano que permite el seguimiento de cada partícula independientemente, aplicando los submodelos necesarios que permitan simular de buena manera su trayectoria,

considerando colisiones con el fondo y también entre partículas.

Junto con lo anterior, se definen cuatro vías de acoplamiento entre las fases involucradas: la primera vía considera que sólo el flujo ejerce fuerzas sobre las partículas, la segunda vía considera que tanto el flujo como las partículas se afectan entre sí, la tercera vía incluye los efectos hidrodinámicos entre partículas y la cuarta vía incorpora la interacción directa (colisiones) entre partículas. Como regla general, la inclusión de cada vía de acoplamiento dependerá de la concentración de partículas existente en el flujo.

En adelante, se presenta el modelamiento, resultados y conclusiones obtenidos de la simulación de un flujo turbulento con acoplamiento de cuatro vías, implementado en OpenFOAM, con partículas correspondientes al rango de las arenas.

1.1 Objetivo general

Implementar un modelo computacional tridimensional que permita el acoplamiento de cuatro vías entre las fases continua (flujo) y dispersa (partículas), a través de un enfoque euleriano-lagrangiano. El modo de transporte se restringe a la saltación, siendo este el proceso de mayor interés en el transporte de sedimentos en ríos.

1.2 Objetivos específicos

- i. Identificar un *solucionador* adecuado para la resolución de un canal turbulento a través de un modelo matemático LES (Large Eddy Simulation), que permite resolver las escalas más energéticas y modelar la interacción con la pared.
- ii. Modificar el *solucionador* identificado en el punto anterior para agregar la fase dispersa, implementando además el doble acoplamiento entre ambas fases.
- iii. Validar los resultados obtenidos con los datos experimentales existentes disponibles en la literatura.
- iv. Analizar el efecto del doble acoplamiento en el transporte de partículas.
- v. Analizar el efecto de la turbulencia en el transporte de fondo de partículas sedimentarias.
- vi. Estudiar el costo computacional de cada uno de los submodelos conforme se incrementa la concentración de partículas.

CAPÍTULO 2

MARCO TEÓRICO

En el esquema de la Figura 2.1 se muestra un esquema del desarrollo del modelo de transporte de fondo implementado en este trabajo.

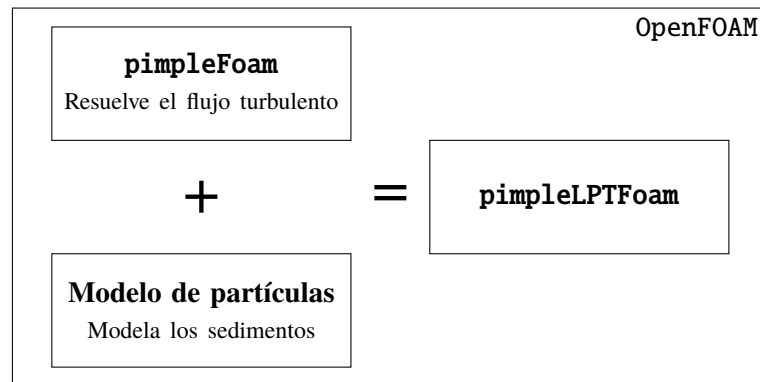


Figura 2.1: Esquema de desarrollo del modelo de transporte de fondo.

Fuente: Elaboración propia.

En primer lugar, se escoge el *solucionador* `pimpleFoam`, parte de OpenFOAM, que permite resolver flujos turbulentos. Luego, se implementa el modelo de partículas que incluye todos los submodelos necesarios para el desarrollo de la saltación.

Por último, se agrega el modelo de partículas al *solucionador* y se genera uno nuevo que es capaz de desarrollar el transporte de sedimentos cuyo nombre es `pimpleLPTFoam`.

En adelante se presenta la teoría que compone las fundaciones de la resolución del flujo, así como también los modelos y submodelos que permiten simulación de partículas. Además, se muestran los valores experimentales utilizados para la validación del modelo.

2.1 Modelamiento de la fase continua o transportadora

En mecánica de fluidos se pueden identificar flujos laminares y turbulentos. Los primeros son aquellos en que las fuerzas viscosas predominan sobre las fuerzas inerciales, y se puede observar que el fluido se mueve de forma ordenada, como si se tratase de láminas, pudiendo identificar las llamadas *líneas de corriente*, tal como se muestra en la Figura 2.2a.

Por otra parte, en el flujo turbulento se observa que las fuerzas inerciales tienen mayor relevancia que las fuerzas viscosas, lo que se traduce en la aleatoriedad del flujo: existen grandes variaciones de velocidad y presión y existe una mezcla constante (vórtices), debido a lo cual no se pueden identificar líneas de corriente definidas (Figura

2.2b), y por lo mismo son imposibles de resolver analíticamente (Pope, 2000).

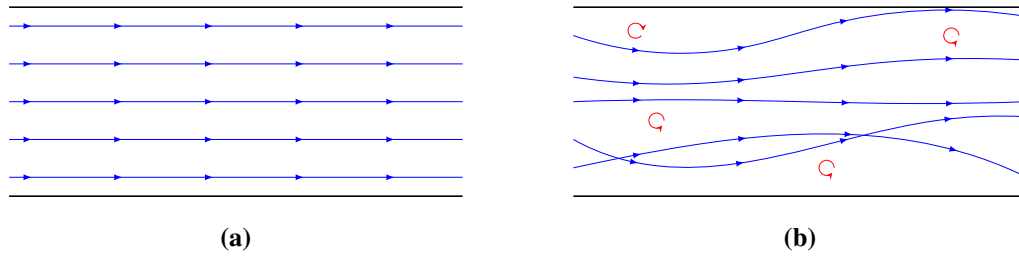


Figura 2.2: Esquema de los flujos (a) laminar y (b) turbulento.

Fuente: Elaboración propia.

El flujo simulado en este estudio es de tipo turbulento, debido a que este es el que se encuentra en los ríos y canales.

Para simular el flujo, el dominio computacional se divide en celdas o volúmenes. De esta forma, en cada celda se resuelven las ecuaciones de Navier-Stokes:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{g} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

Donde $\mathbf{u}$ es la velocidad del fluido, $\frac{\partial}{\partial t}$ es el operador diferencial respecto al tiempo, ∇ es el operador diferencial vectorial, ρ es la densidad del fluido, p es la presión y ν es la viscosidad cinemática del fluido.

En un caso ideal, el flujo sería resuelto completamente utilizando un método de simulación numérica directa (DNS) (Pope, 2000). Sin embargo, el costo computacional de este método es muy alto debido a que requiere discretizaciones muy finas, para capturar los vórtices más pequeños. De esta forma, es necesario recurrir a métodos que permitan reducir los costos de cálculo, pero que no perjudiquen la calidad de la simulación.

El método LES (Large Eddy Simulation) permite simular la turbulencia calculando sólo aquellos vórtices que sean mayores a una escala de longitud dada (filtro espacial), mientras que aquellos menores serán modelados (Figura 2.3). De esta forma, el costo computacional es notablemente menor, al requerir mallas menos refinadas (Mukha y Liefvendahl, 2015; Pope, 2000).

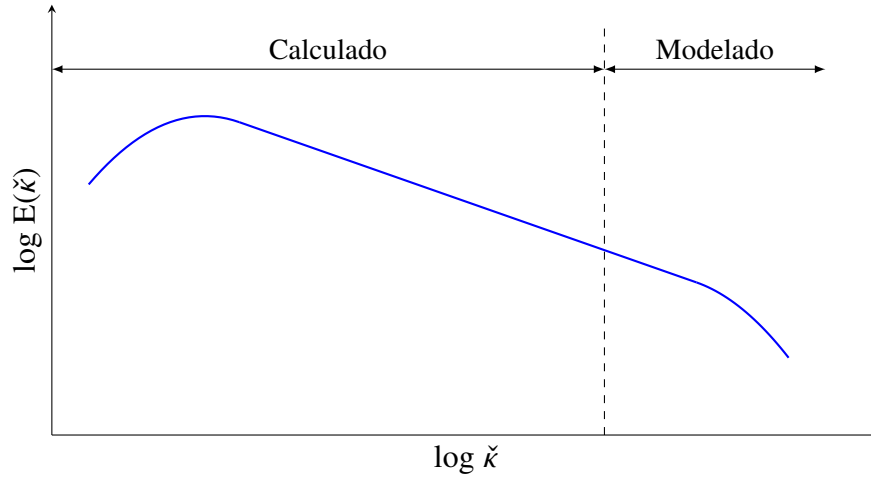


Figura 2.3: Esquema del método LES en términos de energía. k es el número de onda, relacionado con la escala de longitud del vórtice, y $E(k)$ la energía asociada. Con este modelo, las escalas energéticas mayores son calculadas, mientras que aquellas menores son modeladas.

Fuente: Adaptación propia de Pope (2000).

En flujos turbulentos inducidos por condiciones de no deslizamiento del fluido con la pared (Wall-bounded turbulence), como es este caso, se introduce el concepto de *velocidad de fricción* (Bailly y Comte-Bellot, 2015):

$$u^* = \sqrt{\frac{\tau_w}{\rho}} \quad (3)$$

Donde τ_w es el esfuerzo de corte cerca de la pared. La velocidad de fricción contiene toda la información necesaria respecto del flujo turbulento, por lo que se utiliza para caracterizarlo (Bailly y Comte-Bellot, 2015). Con lo anterior, se define la *escala de longitud viscosa*:

$$\delta_v = \frac{\nu}{u^*} \quad (4)$$

que caracteriza los efectos viscosos inducidos por la pared. De esta forma, se introducen las variables adimensionales:

$$y^+ = \frac{y}{\delta_v} \quad (5)$$

$$u^+ = \frac{|\mathbf{u}|}{u^*} \quad (6)$$

Donde y^+ es la distancia a la pared normalizada por la escala de longitud viscosa, u^+ es la velocidad media normalizada por la velocidad de fricción y y es la distancia desde la pared. Estas variables permiten determinar

la importancia relativa entre los procesos viscosos y turbulentos (Pope, 2000).

2.1.1. Ley de la pared

Aunque el flujo turbulento no puede ser resuelto analíticamente, sí es posible aproximar el perfil de velocidades promedio a través de la ley de la pared (Pope, 2000), cuya forma básica esta dada por:

$$u^+ = f(y^+) \quad (7)$$

Donde $f(\cdot)$ es una función que depende sólo de y^+ .

En la Figura 2.4 se muestra un esquema de las principales componentes de la ley de la pared.

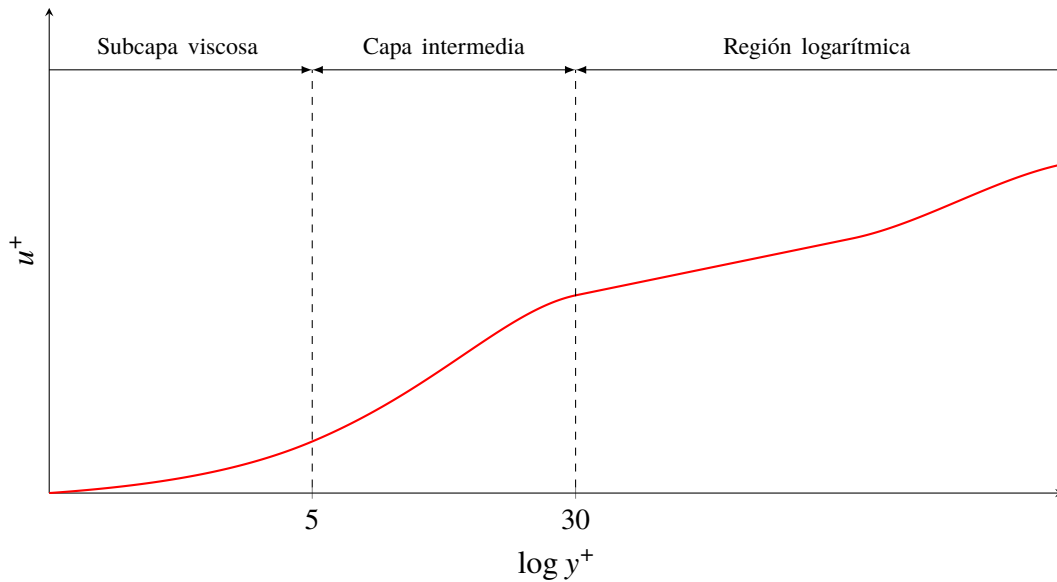


Figura 2.4: Esquema de la Ley de la pared.

Fuente: Elaboración propia.

Se observan tres zonas principales: la más cercana a la pared se denomina subcapa viscosa, y llega hasta una altura aproximada de y^+5 . En esta zona la velocidad es igual a la altura. La zona siguiente recibe el nombre de capa intermedia, y la velocidad se puede calcular interpolando entre los puntos extremos.

Por último, la región logarítmica, que comienza a una altura aproximada de $y^+ = 30$, la velocidad se puede calcular según la ecuación (8).

$$u^+ = \frac{1}{\kappa} \log y^+ + B \quad (8)$$

Dónde κ es la constante de von Kármán, y cuyo valor es aproximadamente $\kappa = 0,41$.

En esta última zona, se ha observado que existe un aumento de la velocidad en la zona más alta, lo que se conoce como Ley de la estela o capa de defecto de velocidad.

Con la ley de la pared definida, es posible validar los flujos turbulentos simulados a través de la comparación de los perfiles de velocidad promedio.

2.1.2. Intensidad de turbulencia

La intensidad de turbulencia es un indicador de la desviación típica de la velocidad del flujo respecto de la velocidad promedio en el tiempo. En términos prácticos, permite medir el grado de *desorden* del flujo.

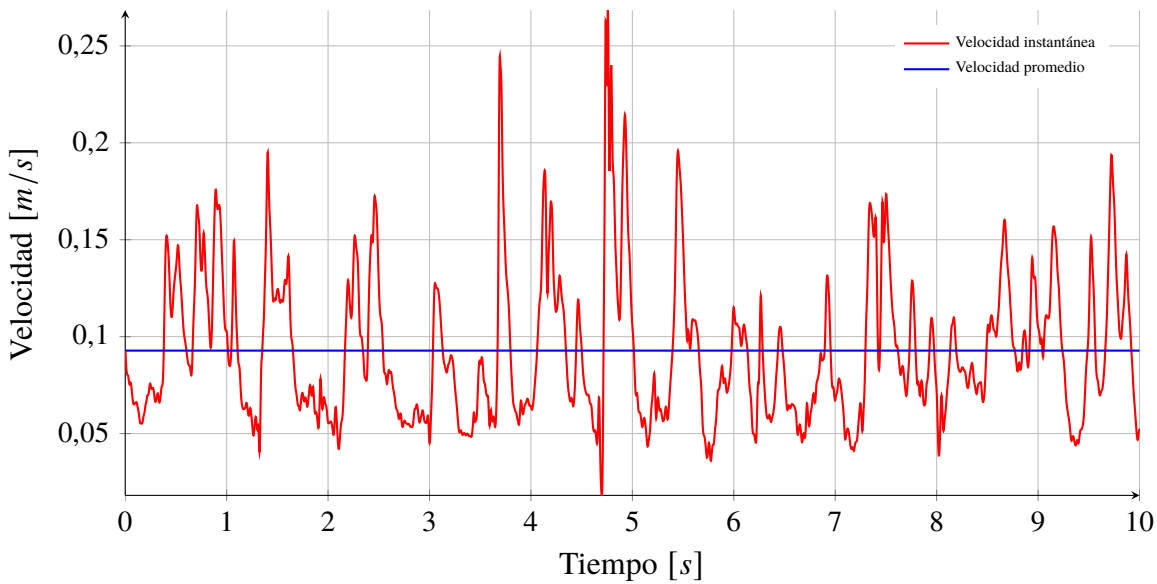


Figura 2.5: Esquema de la intensidad de turbulencia.
Fuente: Elaboración propia.

La Figura 2.5 muestra la velocidad instantánea del flujo en comparación con la velocidad promedio, con lo que es posible calcular la intensidad de turbulencia como se muestra a continuación:

$$\tilde{I} = \frac{u'}{U} \quad (9)$$

Donde I es la intensidad de turbulencia, U es la velocidad promedio del flujo en el tiempo y u' es la raíz cuadrática media de las fluctuaciones de velocidad.

$$u' = \frac{1}{\sqrt{3}} \left(u_x'^2 + u_y'^2 + u_z'^2 \right) \quad (10)$$

Donde u'_i es la diferencia entre la velocidad instantánea en dirección i y la velocidad promedio en la misma

dirección.

2.2 Modelamiento de la fase dispersa

Idealmente, la fase dispersa (partículas) se debería tratar como un continuo sometido a fuerzas de cuerpo y superficie, de acuerdo a la segunda ley de Newton al aplicar el Teorema de Transporte de Reynolds (Ec. (11)). De esta manera existiría un acoplamiento perfecto entre ambas fases (Wiberg y Smith, 1985).

$$\frac{d}{dt} \int_V \rho_s \mathbf{u}_s dV = - \int_V \rho_s \mathbf{g} dV - \int_S p \mathbf{n} \cdot \mathbf{I} dS + \int_S \mathbf{n} \cdot \boldsymbol{\tau} dS \quad (11)$$

Donde $\frac{d}{dt}$ es el operador diferencial temporal, ρ_s es la densidad del sólido, $\mathbf{u}_s$ la velocidad del sólido, $\mathbf{g}$ es la gravedad, $\mathbf{n}$ es un vector unitario normal a la superficie, $\mathbf{I}$ es el tensor identidad de segundo orden y $\boldsymbol{\tau}$ es el esfuerzo de corte. V y S corresponden al volumen y superficie del sólido, respectivamente.

Sin embargo, un método de esta naturaleza no es factible con la capacidad computacional disponible en la actualidad. En cambio, se utiliza un enfoque lagrangiano, específicamente el método de elementos discretos (DEM), en el cual se considera que cada elemento es un punto a la que se le asignan propiedades tales como diámetro, densidad, velocidad y otras características y submodelos pertinentes (Prosperetti y Tryggvason, 2007). Esto permite que se puedan realizar simulaciones con hasta cientos de miles de partículas, de acuerdo a la capacidad computacional disponible en la actualidad.

En el modelamiento del movimiento de las partículas se pueden identificar tres procesos: el vuelo libre (movimiento en suspensión de la partícula entre colisiones), la colisión con el fondo y la colisión entre partículas. De esta forma, se debe utilizar un submodelo de vuelo libre, en el cual la Ec. (11) sea simplificada mediante la utilización de fuerzas calculadas de acuerdo a las propiedades de la misma y las características del flujo.

2.3 Submodelo de fuerzas sobre la fase dispersa

El enfoque lagrangiano aplicado a la fase dispersa implica que la Ec. (11) debe ser simplificada a fin de poder calcular las fuerzas implicadas en el fenómeno de saltación.

De esta manera, para cada partícula se consideran las fuerzas de peso sumergido (F_{SW}), arrastre (F_{DR}), sustentación (F_{LF}), masa virtual (F_{VM}), aceleración del fluido, (F_{FA}), Magnus (F_{MG}) y Basset (F_{BS}). Así, como indica la Segunda Ley de Newton, las ecuaciones del movimiento lineal y angular de la partícula serán, respectivamente:

$$m \frac{d\mathbf{u}_p}{dt} = \sum_i \mathbf{F}_i \quad (12)$$

$$I \frac{d\omega_p}{dt} = \sum_i \mathbf{T}_i \quad (13)$$

Donde m es la masa de la partícula, I su momento de inercia, $\mathbf{u}_p$ y ω_p corresponden a las velocidades lineal y angular de la partícula, respectivamente, $\mathbf{F}_i$ es la fuerza i y $\mathbf{T}_i$ es el torque i . Estas ecuaciones deberán ser resueltas para cada paso de tiempo, dt .

En las ecuaciones (12) y (13) no se consideran los efectos de colisión, puesto que se trata de un caso particular durante el proceso de saltación.

2.3.1. Fuerza de peso sumergido

Corresponde a la diferencia entre la fuerza de gravedad y el peso del fluido desplazado por la partícula. También se conoce como el principio de Arquímedes.

$$\mathbf{F}_{SW} = m\mathbf{g} \left(1 - \frac{\rho}{\rho_p} \right) \quad (14)$$

Donde ρ_p es la densidad de la partícula.

2.3.2. Fuerza de arrastre o *drag*

Es la fuerza producida por los esfuerzos de corte que genera el fluido sobre la partícula, generando el efecto de arrastre (Crowe et al., 2012).

$$\mathbf{F}_{DR} = -\frac{3}{4}mC_d \frac{\rho}{\rho_p} \frac{\mathbf{u}_r |\mathbf{u}_r|}{d_p} \quad (15)$$

Donde C_d es el coeficiente de arrastre, $\mathbf{u}_r$ es la velocidad relativa de la partícula respecto al flujo ($\mathbf{u}_r = \mathbf{u}_p - \mathbf{u}$) y d_p es el diámetro de la partícula

C_d se calcula según la expresión dada por Yen (1992):

$$C_d = \frac{24}{Re} \left(1 + 0,15\sqrt{Re} + 0,017Re \right) - \frac{0,208}{1 + 10^4\sqrt{Re}} \quad (16)$$

Donde Re es el número de Reynolds de la partícula, calculado como $Re = \frac{|\mathbf{u}_r|d_p}{\nu}$.

2.3.3. Fuerza de sustentación o *lift*

Corresponde a la fuerza generada por la diferencia de velocidad entre puntos opuestos de la partícula, lo que genera una diferencia de presión, y por tanto un levantamiento o sustentación de esta misma (Crowe et al., 2012).

$$\mathbf{F}_{LF} = mC_l \frac{\rho}{\rho_p} (-\mathbf{u}_r \times (\nabla \times \mathbf{u})) \quad (17)$$

Donde C_l es el coeficiente de sustentación (se asume igual a 0,2 según Wiberg y Smith (1985)).

2.3.4. Efecto de masa virtual o masa añadida

Cuando un cuerpo es acelerado a través de un fluido, existe una aceleración correspondiente de este último. Esto es conocido como el efecto de masa virtual (Crowe et al., 2012).

$$\mathbf{F}_{VM} = -mC_m \frac{\rho}{\rho_p} \frac{d\mathbf{u}_r}{dt} \quad (18)$$

Donde C_m es el coeficiente de masa virtual (su valor se asume igual a 0,5, según Niño y García (1994)).

Se puede observar que esta ecuación se puede separar en dos términos:

$$\mathbf{F}_{VM} = \mathbf{F}_{VM}^* - m^* \frac{d\mathbf{u}_p}{dt} \quad (19)$$

Donde el primer término del lado izquierdo corresponde a:

$$\mathbf{F}_{VM}^* = mC_m \frac{\rho}{\rho_p} \frac{d\mathbf{u}}{dt} \quad (20)$$

El segundo término se puede trasladar al lado izquierdo de la Ec. (12), por lo que se conoce como masa añadida:

$$m^* = mC_m \frac{\rho}{\rho_p} \quad (21)$$

Se puede ver, entonces, que el efecto de masa aparente toma ese nombre porque es equivalente a añadirle masa a la partícula.

2.3.5. Fuerza de aceleración del fluido

Es la fuerza que toma en cuenta las fuerzas que *existirían* en ausencia de la partícula, tanto debido al gradiente de presiones como al gradiente de presión hidrostática (González, 2008).

$$\mathbf{F}_{FA} = m \frac{\rho}{\rho_p} \frac{D\mathbf{u}}{Dt} \quad (22)$$

Donde $\frac{D}{Dt}$ es el operador de derivada material.

2.3.6. Fuerza de Magnus

Es otra forma de fuerza de sustentación, producida por la diferencia de velocidad entre dos puntos opuestos de la partícula generados por la velocidad de rotación de esta misma (Crowe et al., 2012).

$$\mathbf{F}_{MG} = m C_M \frac{\rho}{\rho_p} (\boldsymbol{\omega}_p \times \mathbf{u}_r) \quad (23)$$

Donde C_M es el coeficiente de Magnus.

El coeficiente de Magnus esta dado por (Lukerchenko et al., 2009):

$$C_M = 0,0844 \frac{Re}{Re_R} + \left(0,75 - 0,0844 \frac{Re}{Re_R} \right) e^{-0,1 Re^{0,3} Re_R^{0,4}} \quad (24)$$

Re_R es el número de Reynolds de rotación, dado por $Re_R = \frac{\rho d_p^2 |\boldsymbol{\omega}_p - (\nabla \times \mathbf{u})|}{4 \rho_p \nu}$.

2.3.7. Fuerza de Basset

La fuerza de Basset o fuerza histórica de Basset considera la demora en la formación de la capa límite en torno a la partícula en movimiento a medida que la velocidad relativa cambia en el tiempo (Crowe et al., 2012).

$$\mathbf{F}_{BS} = -\frac{3}{2} d_p^2 \sqrt{\pi \rho \mu} \int_0^t \frac{d\mathbf{u}_r}{d\xi} \frac{d\xi}{\sqrt{t-\xi}} \quad (25)$$

Donde μ es la viscosidad dinámica del fluido, t es el tiempo y ξ es una variable auxiliar.

La Ec. (25) es difícil de resolver puesto que se genera una singularidad en el límite superior. Para su resolución se han desarrollado técnicas numéricas como las mencionadas en Moreno y Bombardelli (2016), entre las que se cuenta el método de van Hinsberg et al. (2011), que se ha utilizado en este trabajo.

2.4 Submodelo de vuelo libre

Para el período entre colisiones (Figura 2.6), se resuelven las ecuaciones (12) y (13) en cada paso de tiempo, calculando así las nuevas aceleraciones, velocidades y posiciones.

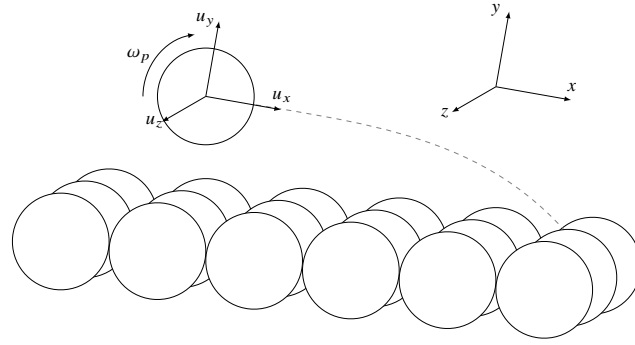


Figura 2.6: Esquema de la etapa de vuelo libre en el modelo de saltación. u_i es la velocidad lineal de la partícula en la dirección i .

Fuente: Adaptación propia de Moreno (2014).

Considerando las ecuaciones para cada fuerza presentadas en la sección 2.3, la Ec. (12) se reduce a:

$$(m + m^*) \frac{d\mathbf{u}_p}{dt} = \mathbf{F}_{SW} + \mathbf{F}_{DR} + \mathbf{F}_{LF} + \mathbf{F}_{VM}^* + \mathbf{F}_{FA} + \mathbf{F}_{MG} + \mathbf{F}_{BS} \quad (26)$$

Por otra parte, para la Ec. (13) no existen estudios sobre fuerzas que permitan realizar este seguimiento. En su lugar, se utiliza la expresión entregada por Yamamoto et al. (2001):

$$I \frac{d\omega_p}{dt} = -C_T \frac{\rho_p}{2} \left(\frac{d_p}{2} \right)^5 |\omega_r| \omega_r \quad (27)$$

Donde ω_r la velocidad angular relativa de la partícula respecto a la vorticidad flujo. C_T es un coeficiente adimensional dependiente del número de Reynolds de rotación, denominado *coeficiente de torque*, y se calcula como:

$$C_T = \frac{C_1}{\sqrt{Re_R}} + \frac{C_2}{Re_R} + C_3 Re_R \quad (28)$$

Donde los coeficientes C_1 , C_2 y C_3 dependen del número de Reynolds de rotación y son presentados en la Tabla A.1.

2.5 Submodelos de colisión

En un flujo con sedimentos, es común observar que las partículas chocan entre ellas debido a sus trayectorias irregulares, además del choque con el fondo del canal. Este es un fenómeno importante y parte fundamental del proceso de saltación, sobre todo cuando existen altas concentraciones de partículas, por lo que se hace necesario utilizar modelos que capturen este efecto.

Existen dos modelos predominantes: el modelo de esfera dura (Hard Sphere Model), que calcula el momentum lineal y angular de las partículas post colisión a partir de las condiciones previas al choque y de parámetros como el coeficiente de restitución e y el coeficiente de fricción f . Sin embargo, este modelo sólo permite colisiones binarias, por lo que su uso se restringe a flujos con baja densidad de partículas (donde prácticamente no existen colisiones entre partículas), o para la colisión partícula-pared (Crowe et al., 2012).

El modelo de esfera blanda (Soft Sphere Model), por otra parte, considera un sistema de resorte-amortiguador para representar las colisiones, y el proceso se resuelve mediante la integración numérica de la ecuación de movimiento. Aunque este modelo es más costoso en términos de tiempo computacional, permite la colisión de múltiples partículas en un mismo instante, por lo que su uso es adecuado para altas concentraciones de partículas (mayores al 5 % de concentración por volumen) (Crowe et al., 2012).

2.5.1. Colisión con el fondo

2.5.1.1 Modelo de esfera dura

La colisión con el fondo se modela mediante un *modelo de esfera dura*, el cual se basa en la fuerza de impulso aplicada sobre la partícula en el momento de la colisión. Además, se hacen los siguientes supuestos (Crowe et al., 2012):

1. No existe deformación de la partícula, por lo que la distancia del centro de la partícula al punto de colisión es constante,
2. La fricción durante el deslizamiento obedece la ley de fricción de Coulomb y,
3. Cuando una partícula deja de deslizar, ya no ocurrirá más deslizamiento.

De esta manera, se tienen las siguientes ecuaciones de impulso:

$$m(\mathbf{u}_p - \mathbf{u}_p^{(0)}) = \mathbf{J} \quad (29)$$

$$I(\boldsymbol{\omega}_p - \boldsymbol{\omega}_p^{(0)}) = \frac{d_p}{2} \hat{\mathbf{n}} \times \mathbf{J} \quad (30)$$

Donde $\mathbf{J}$ es la fuerza de impulso sobre la partícula y $\hat{\mathbf{n}}$ es el vector unitario en dirección del centro de la partícula a la pared. El superíndice (0) indica el valor antes de la colisión.

Con lo anterior, e introduciendo los coeficientes de restitución, e y de fricción, f , asumidos conocidos, se obtienen las siguientes velocidades de salida a partir de las velocidades de entrada (ver desarrollo en el Anexo B.1). Si se satisface la condición:

$$\frac{\hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)}}{|\mathbf{u}_{p_{ct}}^{(0)}|} < \frac{2}{7} \frac{1}{f(1+e)} \quad (31)$$

Donde $\mathbf{u}_{p_{ct}}$ es la velocidad tangencial en el punto de contacto de la partícula, entonces la partícula desliza, y las velocidades de salida son:

$$\mathbf{u}_p = \mathbf{u}_p^{(0)} - (1+e)(\hat{\mathbf{n}} + f\mathbf{t})(\hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)}) \quad (32)$$

$$\omega_p = \omega_p^{(0)} - \frac{5}{d_p}(1+e)(\hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)})f(\hat{\mathbf{n}} \times \mathbf{t}) \quad (33)$$

Donde $\mathbf{t}$ es un vector unitario perpendicular a $\mathbf{n}$. En caso contrario, la partícula rebota, y las velocidades de salida están dadas por:

$$\mathbf{u}_p = \mathbf{u}_p^{(0)} - \left\{ (1+e)(\hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)})\hat{\mathbf{n}} + \frac{2}{7}|\mathbf{u}_{p_{ct}}^{(0)}|\mathbf{t} \right\} \quad (34)$$

$$\omega_p = \omega_p^{(0)} - \frac{10}{7d_p}|\mathbf{u}_{p_{ct}}^{(0)}|(\hat{\mathbf{n}} \times \mathbf{t}) \quad (35)$$

2.5.1.2 Modelo estocástico de colisión o de rugosidad del lecho

Para representar la rugosidad del lecho, se incluye un modelo estocástico de colisión elaborado por García y Niño (1992), que establece una relación geométrica entre el ángulo de entrada y el ángulo de salida de la partícula. Esta relación es:

$$\frac{r}{d_p} = \frac{1}{2} [\cos(\theta_b) - \tan(\theta_{in}) \cos(\theta_b)] \quad (36)$$

Donde $\frac{r}{d_p}$ varía aleatoriamente entre 0 y 0,5. El ángulo θ_{in} es el ángulo de colisión de la partícula, y queda definido por las velocidades de entrada. θ_b es el ángulo de rotación de la normal para calcular las velocidades de salida (Figura 2.7).

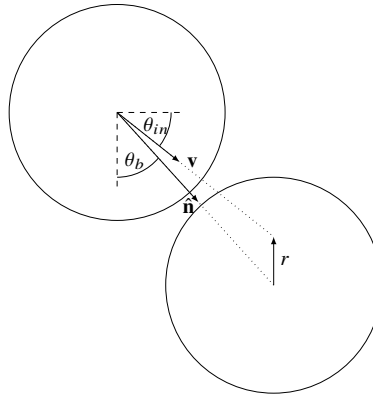


Figura 2.7: Esquema del modelo de colisión estocástica de García y Niño (1992).

Fuente: Adaptación propia de Moreno (2014).

2.5.2. Colisión entre partículas

2.5.2.1 Modelo de esfera blanda (DSEM)

Este modelo reemplaza las deformaciones que produciría un choque entre partículas con un solapamiento entre las mismas, y mediante un modelo de resorte-amortiguador como los de la Figura 2.8, se calculan las fuerzas repulsivas sobre cada partícula, tanto normales (Figura 2.8a) como tangenciales (Figura 2.8b) (Crowe et al., 2012). A diferencia del modelo de esfera dura, el modelo de esfera blanda permite colisiones entre más de dos partículas simultáneamente, por lo que es ideal para flujo con altas concentraciones de sedimentos.

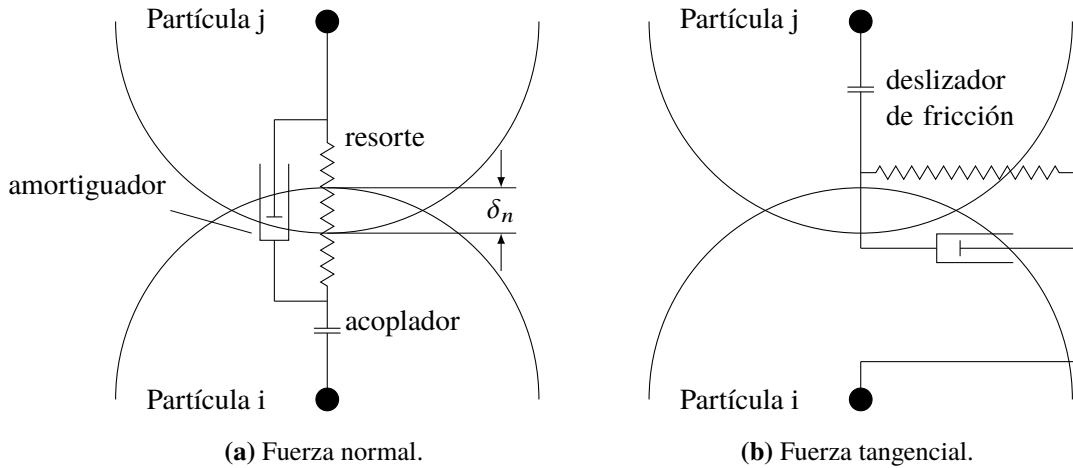


Figura 2.8: Esquema del modelo de esfera blanda.

Fuente: Adaptación propia de Crowe et al. (2012).

El modelo de esfera blanda se basa en tres parámetros principales:

- Rigidez, k

- Amortiguamiento, η
- Coeficiente de fricción, f (obtenido empíricamente)

Así, la fuerza normal $\mathbf{F}_{nij}$ ejercida por la partícula j sobre la partícula i está dada por:

$$\mathbf{F}_{nij} = (-k_n\delta_n - \eta_{nj}\mathbf{G} \cdot \hat{\mathbf{n}}) \hat{\mathbf{n}} \quad (37)$$

Donde δ_n es el desplazamiento en dirección de $\hat{\mathbf{n}}$, $\mathbf{G}$ es la velocidad relativa entre las partículas y $\hat{\mathbf{n}}$ es un vector unitario en la dirección de la línea entre los centros de ambas partículas (perpendicular al plano de colisión).

La fuerza tangencial $\mathbf{F}_{tij}$ ejercida por la partícula j sobre la partícula i está dada por:

$$\mathbf{F}_{tij} = -k_t\delta_t - \eta_{tj}\mathbf{G}_{ct} \quad (38)$$

Donde $\mathbf{G}_{ct}$ es la velocidad de deslizamiento entre las partículas. Si se cumple la condición:

$$|\mathbf{F}_{tij}| > f|\mathbf{F}_{nij}| \quad (39)$$

La partícula i deslizará y por lo tanto en vez de la Ec. (38), la fuerza tangencial está dada por:

$$\mathbf{F}_{tij} = -f|\mathbf{F}_{nij}|\mathbf{t} \quad (40)$$

Donde $\mathbf{t}$ es un vector unitario dado por:

$$\mathbf{t} = \frac{\mathbf{G}_{ct}}{|\mathbf{G}_{ct}|} \quad (41)$$

De esta forma, en cada paso de tiempo la fuerza y el torque total ejercida sobre una partícula será:

$$\mathbf{F}_i = \sum_j (\mathbf{F}_{nij} + \mathbf{F}_{tij}) \quad (42)$$

$$\mathbf{T}_i = \sum_j (r\hat{\mathbf{n}} \times \mathbf{F}_{tij}) \quad (43)$$

Es de notar que para resolver esta ecuación de movimiento se hace necesario discretizar el tiempo de colisión en intervalos más pequeños que los utilizados para resolver el vuelo libre de la partícula, y por lo mismo este modelo se hace más costoso en tiempo computacional (Crowe et al., 2012). El desarrollo matemático completo del modelo se puede ver en el Anexo B.2.

2.6 Acoplamiento de la fase transportadora con la fase dispersa

En un modelo de interacción fluido-partícula existen cuatro vías de acoplamiento posibles (Figura 2.9).

La primera vía, como se observa en la Figura 2.9a, sólo considera el efecto de la fase continua sobre la partícula, no existiendo ningún efecto adicional sobre el flujo ni entre partículas.

En la segunda vía se integra la ley de acción y reacción, por lo cual la presencia de la partícula tendrá efecto sobre el momentum del flujo, como se ve en la Figura 2.9b. Para lograr este efecto es necesario agregar un último término en la ecuación de Navier-Stokes (Ec. (1)), que corresponde a las fuerzas aplicadas sobre la partícula.

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{g} + \sum F_{partícula} \quad (44)$$

En la tercera vía, tal como indica la Figura 2.9c, se considera que la estela dejada por una partícula que ya ha pasado afectará a una que pase posteriormente por aquel lugar, generándose fuerzas hidrodinámicas adicionales.

Por último, la cuarta vía incorpora la interacción directa entre partículas, como se muestra en la Figura 2.9d. Esta vía de acoplamiento es importante cuando se trabaja con flujos con alta densidad de sedimentos (concentración por volumen de partículas mayor al 5 %).

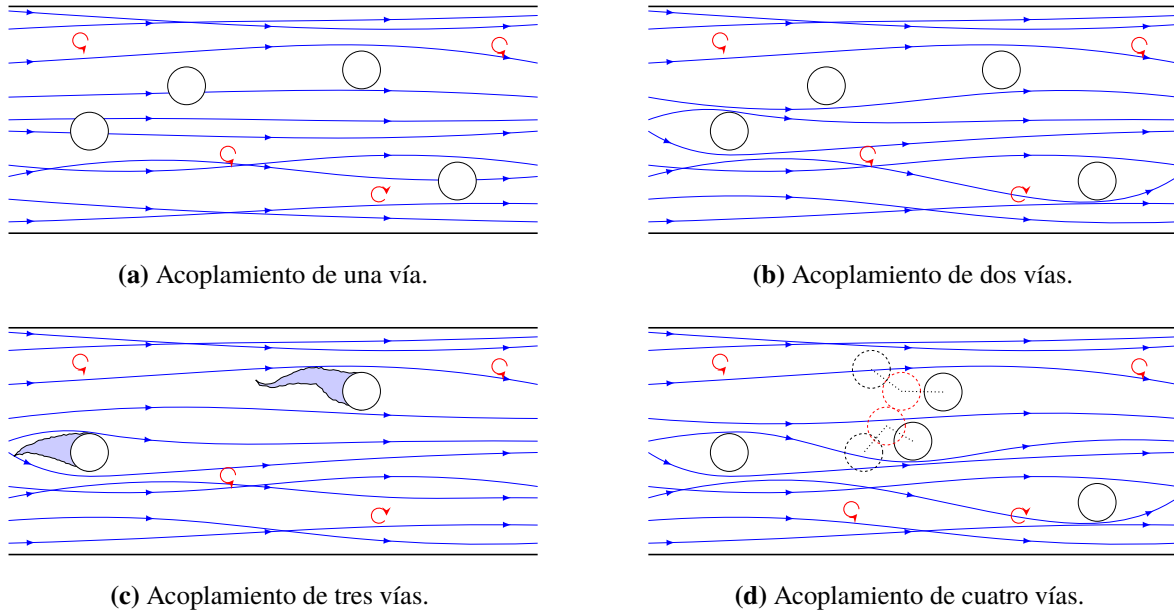


Figura 2.9: Vías de acoplamiento entre la fase continua y la fase dispersa.

Fuente: Elaboración propia.

Según Elghobashi (1994), la incorporación de cada una de estas vías se hace necesaria a medida que aumenta la concentración de partículas y su efecto depende de la *etapa de transporte*, que se define como la relación entre el esfuerzo cortante τ_b y el esfuerzo de corte crítico τ_{cr} que inicia el movimiento de la partícula, y es una forma

de medir la intensidad del flujo:

$$\text{etapa de transporte} = \tau_b / \tau_{cr} \quad (45)$$

De esta forma, existe movimiento de partículas cuando la etapa de transporte excede la unidad, es decir, cuando $\tau_b > \tau_{cr}$

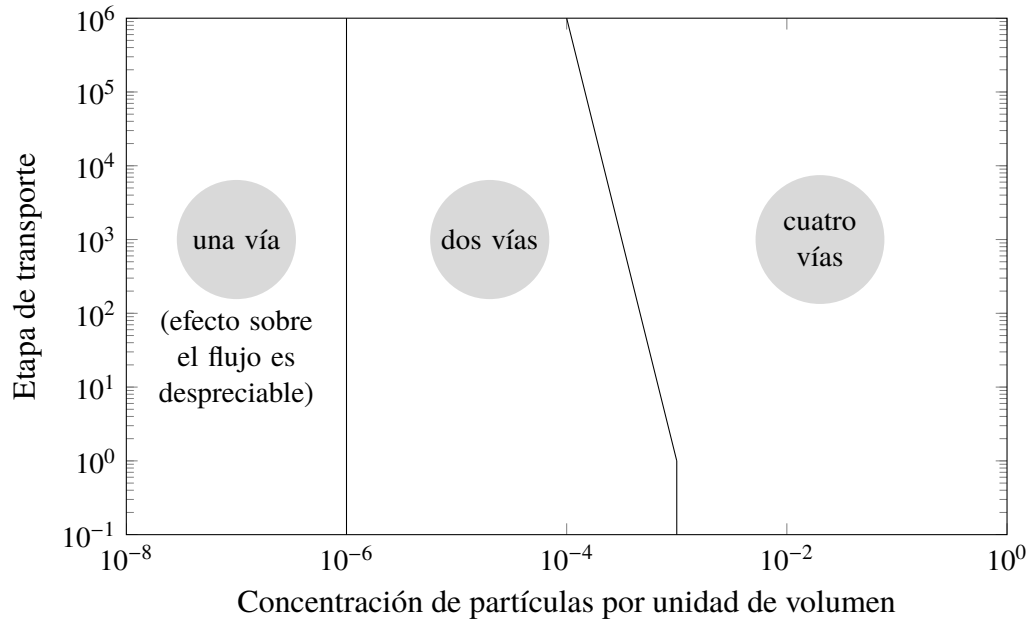


Figura 2.10: Interacción entre fases de acuerdo a concentración de partículas y etapa de transporte según Elghobashi (1994). El acoplamiento de dos vías puede producir o disipar momentum según la *etapa de transporte*.

Fuente: Adaptación propia de Elghobashi (1994).

2.7 Información experimental para validación

Para realizar la validación del modelo computacional, es necesario que los resultados obtenidos sean coherentes con los datos experimentales presentes en la literatura. Entre estos, los artículos que cuentan con la información más detallada son los de Niño y García (1994), Lee y Hsu (1994), Niño y García (1998) y Lee et al. (2006), cuyas configuraciones se resumen en la Tabla 2.1

Tabla 2.1: Configuración de los experimentos sobre saltación realizados por distintos autores. H y L corresponden a las altura y longitud de salto, respectivamente, y $|u_p|$ es la velocidad de la partícula.

Autor	Método de grabación	Tamaño de partícula [mm]	Número de saltos considerados	Velocidad de fricción, u^* [cm/s]	Variables obtenidas
Niño y García (1994)	Cámara de video estándar	15 – 31	80	14 – 23	Promedio y desviación estándar de H, L y $ u_p $
Lee y Hsu (1994)	Cámara de video estándar	1,36 – 2,47	<i>No disponible</i>	3,6 – 10,5	Promedio de H, L y $ u_p $
Niño y García (1998)	Cámara de video de alta velocidad	0,5 – 0,8	1 – 2 saltos por cada 100 partículas	2,1 – 2,6	Promedio y desviación estándar de H, L y $ u_p $
Lee (2000)	Cámara de video estándar	6	<i>No disponible</i>	3,8 – 5,4	Promedio de H, L y $ u_p $
Lee et al. (2006)	Cámara de video estándar	0,6	<i>No disponible</i>	3,9 – 6,8	Promedio de H, L y $ u_p $

Fuente: Elaboración propia.

En particular, los valores a utilizar para la validación son los obtenidos por Niño y García (1998), debido principalmente a que es el que utiliza los parámetros de flujo y tamaño de partícula considerados en este estudio.

En cuanto a los valores para los coeficientes de restitución (e) y fricción (f), los valores utilizados por distintos investigadores varían ampliamente según el material de la partícula y las condiciones del flujo, por lo que no existen valores universales (Moreno, 2014). En la Tabla 2.2 se resumen algunos de los valores sugeridos por diferentes autores.

Tabla 2.2: Valores para el coeficiente de restitución (e) y fricción (f) sugeridos por distintos autores.

Autor	Coeficiente de restitución e	Coeficiente de fricción f
Niño y García (1994)	$e = 0,75 - 0,25\tau^*/\tau_c^*$	0,89
Schmeeckle et al. (2001)	0,65	0,1
Tsuji et al. (1987)	0,8	0,4

Fuente: Elaboración propia.

CAPÍTULO 3

SIMULACIONES

El trabajo se divide en dos etapas: primero se realiza validación del modelo de saltación con los datos experimentales para arenas de Niño y García (1998). Posteriormente se simulan canales con distintas concentraciones de partículas, con el fin de estudiar el fenómeno de arrastre de sedimentos desde el punto de vista teórico. Todas las simulaciones se ejecutan con el *solucionador* `pimpleLPTFoam`, implementado en OpenFOAM, que une el algoritmo para resolución del flujo, PIMPLE, con el modelo de partículas.

3.1 Hardware utilizado

Las simulaciones se han realizado en Leftraru, el clúster de computación de alto rendimiento del NLHPC¹.

Para cada simulación se han utilizado 40 núcleos de la partición SLIMS de Leftraru, cuyas características se muestran en la Tabla 3.1.

Tabla 3.1: Características del hardware de la partición SLIMS del clúster Leftraru.

Nodo	Modelo	Cantidad	Procesador	Ram	Infiniband
Nodos delgados	HP ProLiant SL230s Gen8	128	2 x Intel Xeon E5-2660 10 cores C./U.	48 GB	Infiniband FDR 4X Mellanox
Nodos gruesos	HP ProLiant SL250s Gen8	2	2 x Intel Xeon E5-2660 10 cores C./U.	64 GB	Infiniband FDR 4X Mellanox

Fuente: NLHPC, 2019.

3.2 Simulación en OpenFOAM

OpenFOAM es un entorno de trabajo para el desarrollo de aplicaciones que utilizan las funcionalidades contenidas en aproximadamente 100 librerías escritas en C++. OpenFOAM cuenta con casi 250 aplicaciones que se dividen en dos categorías: *solucionadores*, que permiten resolver problemas de mecánica de fluidos; y *utilidades* que permiten la manipulación de datos (*OpenFOAM User Guide*, 2017).

Los *solucionadores* existentes, en su mayoría corresponden a distintas modificaciones de los métodos PISO y SIMPLE, que permiten resolver problemas para casos transientes o estacionarios, respectivamente. Además, existen *solucionadores* que implementan el método PIMPLE, que es una combinación de los algoritmos PISO y SIMPLE, y cuya ventaja es que permite aumentar la discretización temporal sin perder estabilidad ni precisión (Holzman, 2017).

¹National Laboratory of High Performance Computing

La ventaja en el uso de OpenFOAM es que permite hacer modificaciones a los *solucionadores* existentes y de esta manera resolver un problema específico. A continuación, se describen tanto el *solucionador* como la simulación del flujo y las partículas utilizadas en el estudio.

3.2.1. Descripción del *solucionador* `pimpleLPTFoam`

El *solucionador* utilizado corresponde a una modificación de `pimpleFoam`, cuya base de funcionamiento es el algoritmo PIMPLE (ver Anexo C), al cual se le ha incorporado una nube de partículas (modificación de la librería `intermediate` de OpenFOAM).

La nube de partículas implementada permite la utilización de todas las fuerzas mencionadas en la Sección 2.3, además de los modelos de esfera dura para las colisiones con el fondo y de esfera blanda para las colisiones entre partículas. Para el acoplamiento entre las fases dispersa y continua, se agrega el término de las fuerzas sobre las partículas en la ecuación de Navier-Stokes, obteniendo la Ec. (44).

El *solucionador* `pimpleLPTFoam` permite variar tanto parámetros del flujo como de las partículas, tales como las fuerzas aplicadas, la densidad, velocidad inicial, número, entre otros, aumentando así su versatilidad para ser utilizado en flujos de diversas características, o con otros tipos de partículas. La ventaja de la implementación del modelo de saltación de este estudio en OpenFOAM, es que permite ser extendido con relativa facilidad, pudiendo agregar modelos de colisión o fuerzas. En el Anexo D.1 se puede ver el código fuente del *solucionador* y las adiciones realizadas por el autor en el marco de esta memoria.

Las condiciones iniciales necesarias son la velocidad del flujo y sus propiedades, junto con las posiciones iniciales de la nube de partículas y los parámetros de todos los submodelos implementados.

De esta manera, la simulación sigue el proceso mostrado en la Figura 3.1. En primera instancia, se calculan las nuevas velocidades y presiones en el flujo resolviendo la Ec. (44) a través del algoritmo PIMPLE, considerando el momentum aportado por las partículas dentro del flujo (en la primera iteración será igual a 0). Luego, conocidas las nuevas velocidades del flujo, se calculan las fuerzas sobre las partículas y se realiza la integración numérica para el movimiento, comprobando si existe algún tipo de colisión (de haberlo se activa la subrutina). Esta operación se repite hasta que $t = t_{final}$.

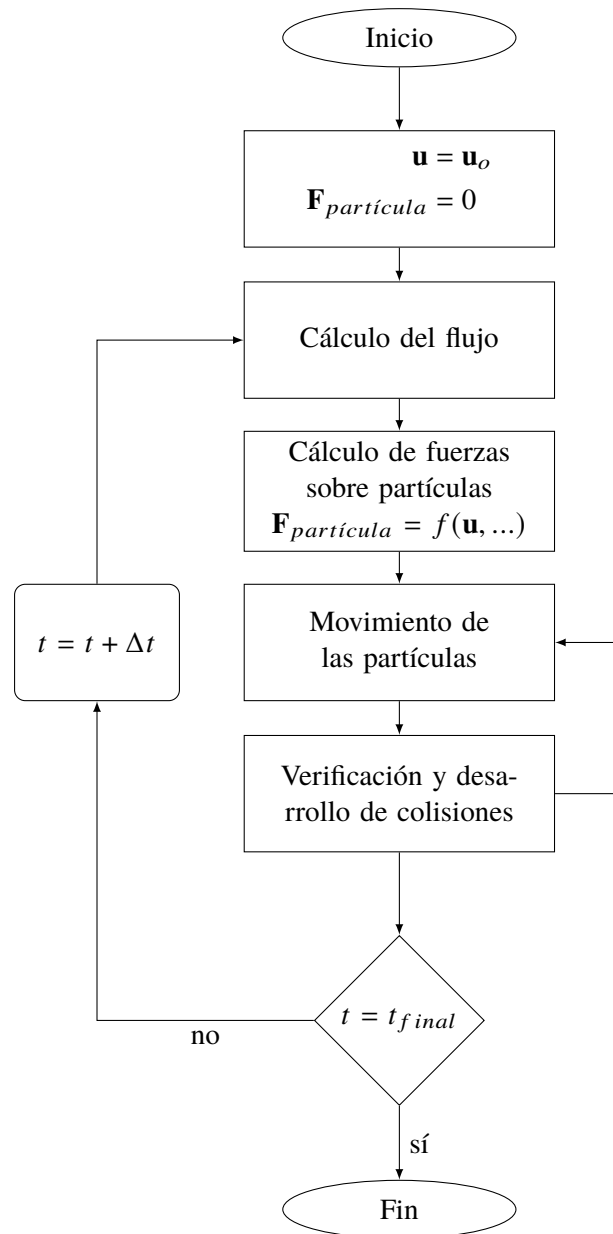


Figura 3.1: Diagrama de flujo del *solucionador* pimpleLPTFoam.

3.2.2. Simulación numérica del flujo

El estudio se realiza en un volumen de dimensiones $2\pi h \times 2h \times \pi h$, donde h es la altura del canal, con condiciones cíclicas en las paredes laterales y condiciones de no deslizamiento en las paredes superior e inferior para inducir la turbulencia. Además, como condición inicial se utiliza una velocidad promedio uniforme en todo el dominio, tal que se obtenga la velocidad de fricción u^* deseada.

Las condiciones cíclicas consisten en que entre paredes opuestas, los valores de velocidad y presión son iguales, por lo que en términos prácticos el canal tiene una extensión infinita.

Para la modelación de turbulencia, se utiliza el método LES (Large Eddy Simulation), cuyo principio de funcionamiento se explica en la sección 2.1. Así, el refinamiento de la malla debe ser lo suficientemente fino para capturar los vórtices de mayor tamaño, pero también lo suficientemente grande para modelar los vórtices más pequeños y resolver el problema en un tiempo menor. En la Figura 3.2 se muestra el detalle de la malla utilizada en el estudio, que toma en cuenta las condiciones mencionadas anteriormente.

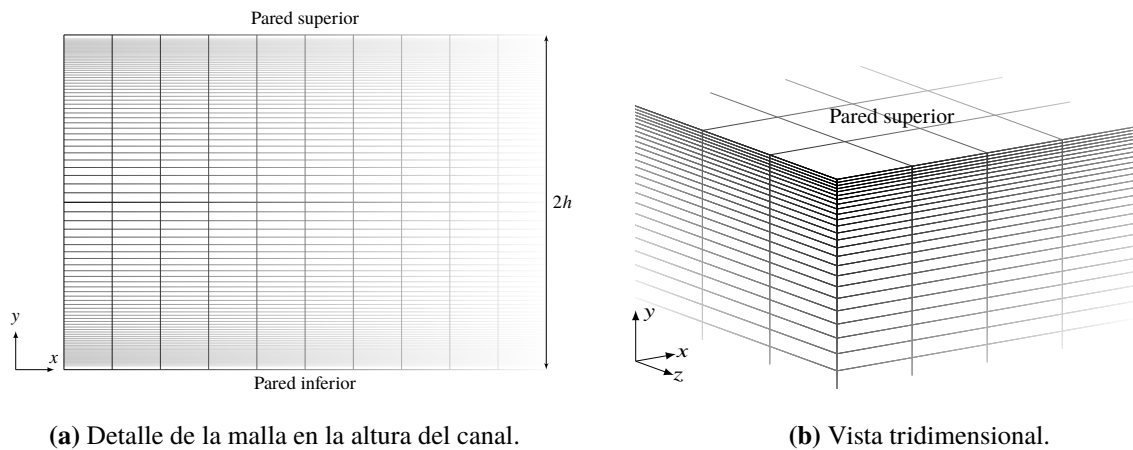


Figura 3.2: Detalle de la malla utilizada en el estudio. En (a) se puede observar el aumento del refinamiento hacia las paredes superior e inferior. En (b) se ve una esquina de la malla, donde se puede apreciar el espaciamiento equidistante de las celdas en los ejes horizontales. Las paredes superior e inferior se utilizan condiciones de no deslizamiento, mientras que en las demás se imponen condiciones cíclicas (entre paredes opuestas).

Fuente: Elaboración propia.

En la Figura 3.3 se muestra el flujo turbulento completamente desarrollado, dadas las condiciones de borde establecidas en el modelo. Se puede ver que el mayor refinamiento en las paredes permite capturar de mejor forma la turbulencia.

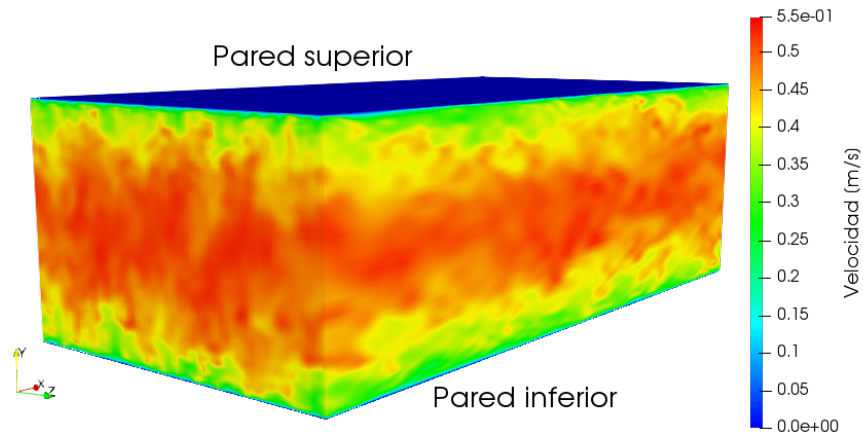


Figura 3.3: Flujo turbulento simulado en OpenFOAM. Se puede ver el efecto de la turbulencia generada por las paredes superior e inferior, la cual es capturada gracias al mayor refinamiento existente en esa zona. El flujo circula en sentido del eje x , y los colores mostrados corresponden a la magnitud de la velocidad del flujo.

Fuente: Elaboración propia con Paraview.

En el Anexo D.2 se pueden ver los archivos utilizados para el modelamiento del flujo. En este trabajo, en particular, se trabaja con una altura $h = 0,0347$ [m] y una velocidad de fricción de $u^* = 0,024$ [m/s], que son los valores utilizados en los experimentos de Niño y García (1998).

3.2.3. Propiedades de las partículas

Las partículas simuladas corresponden a arena de diámetro uniforme $d_p = 0,69$ [mm] y densidad de $\rho_p = 2650$ [kg/m³]. La inyección de las partículas se hace de forma manual, indicando la posición inicial de cada una individualmente. Además, se establecen los modelos de colisión, fuerzas, y otras funciones auxiliares para el modelamiento. En el anexo D.3 se puede ver el detalle de los modelos y parámetros utilizados en las simulaciones.

3.3 Obtención de resultados y análisis

En primera instancia, se valida la fase transportadora comparando el perfil de velocidades simulado con el perfil dado por la Ley de la Pared (Pope, 2000), para las condiciones de turbulencia utilizadas por Niño y García (1998).

Luego, es necesario validar el modelo computacional de saltación. Para esto, se realiza la simulación con 10 partículas, y se calcula el promedio y la desviación estándar de cada parámetro estudiado, según se ve en la Figura 3.4. Estos valores estadísticos permiten comparar el modelo con los resultados experimentales de Niño y García (1998), y de esta forma validar el modelo.

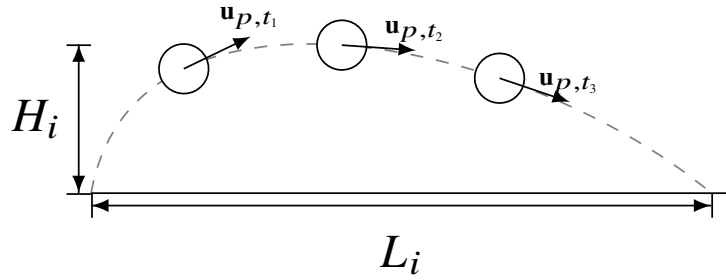


Figura 3.4: Esquema de los datos obtenidos en un salto de partícula. H_i y L_i corresponden a la altura máxima y la longitud del salto i , respectivamente. Además, se obtiene la velocidad de la partícula $u_{p,t}$ en cada punto de la trayectoria.

Fuente: Adaptación propia de Moreno (2014).

Una vez validada la fase continua y el modelo de saltación, se realizan simulaciones con concentraciones entre 0 y 15 % por unidad de volumen, con el fin de analizar diversos efectos. En cuanto al flujo, se desea cuantificar la variación de velocidad respecto a la situación sin partículas, y también el efecto sobre la turbulencia. Por otra parte, sobre las partículas, se calcula el flujo másico a través del canal y se compara con la situación sin doble acoplamiento.

Por último, se quiere calcular la fracción del tiempo total que requiere cada etapa del programa: cálculo del flujo, movimiento de las partículas y colisiones. Esto, para obtener una noción de qué etapa predomina conforme la concentración de partículas varía.

CAPÍTULO 4

RESULTADOS

Los resultados de las simulaciones mencionadas en el Capítulo 3 se muestran a continuación.

4.1 Validación de la fase continua

Se compara el perfil de velocidades promedio del flujo simulado para una velocidad de fricción de $u^* = 0,024 \text{ [m/s]}$, en un canal de altura $h = 0,0347 \text{ [m]}$ (Niño y García, 1998), con el perfil teórico calculado con la Ley de la Pared (Pope, 2000), el que se puede observar en la Figura 4.1. La comparación se realiza utilizando valores normalizados de velocidad (u^+) y distancia (y^+), a modo de simplificar el análisis.

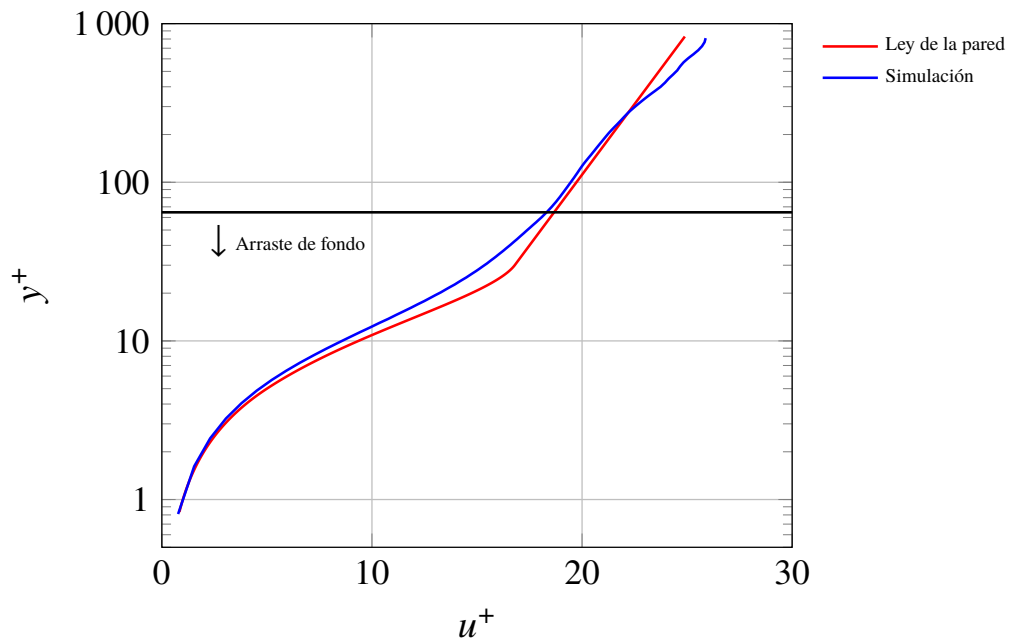


Figura 4.1: Comparación del flujo turbulento simulado en OpenFOAM con la Ley de la Pared. La línea negra representa la altura en que ocurre el arrastre de fondo ($y = 4d_p$).

Fuente: Elaboración propia.

4.2 Validación del modelo de saltación

Los resultados obtenidos para nueve partículas (Tabla A.2), son promediados y comparados con los datos experimentales de Niño y García (1998). En la Tabla 4.1 se resumen los valores obtenidos para velocidad, altura y longitud de salto, y se calcula el error respecto a los valores experimentales.

Tabla 4.1: Comparación de los resultados obtenidos en la simulación en OpenFOAM con los obtenidos por Niño y García (1998). Los resultados de altura y longitud se han normalizado con d_p , y los de velocidad con u^* . Se han obtenido resultados de más de 100 saltos para 9 partículas.

Resultado	Altura [H/d_p]		Longitud [L/d_p]		Velocidad [u_p/u^*]	
	Promedio	Desv. Est.	Promedio	Desv. Est.	Promedio	Desv. Est.
Simulación	0,94	1,34	11,51	11,70	6,64	5,53
Experimental (Niño y García, 1998)	1,07	0,37	9,06	4,63	5,67	1,48
Error porcentual	11,8 %	263,8 %	27,0 %	152,6 %	17,2 %	273,4 %

Fuente: Elaboración propia.

Los valores para los coeficientes de restitución y fricción utilizados son:

$$e = 0,7$$

$$f = 0,2$$

4.3 Estudio con concentraciones mayores

Se obtuvieron resultados para simulaciones con concentraciones de partículas entre 0 y 15 % por unidad de volumen. Dicha concentración ha sido calculada considerando que el arrastre de fondo se produce hasta una altura de cuatro diámetros de partícula desde el lecho del canal. De esta forma, se obtienen las cantidades de partículas mostradas en la Tabla 4.2.

Tabla 4.2: Cantidad de partículas según concentración.

Concentración [%]	Número de partículas
0,0	0
2,5	9 526
5,0	19 050
7,5	28 575
10,0	38 099
12,5	47 624
15,0	57 149

Fuente: Elaboración propia.

Debido a las altas concentraciones de partículas y a la discretización de la malla, es posible que se generen

inestabilidades. Esto, porque al aplicarse las fuerzas en las celdas correspondientes, en ocasiones el momentum total de una celda será mucho mayor (o menor) que las adyacentes, dando como resultado un error en la simulación.

Para evitar la situación anterior, para este estudio se ha utilizado una malla menos refinada, donde el volumen de la celda más pequeña que puede contener una partícula sea mayor que el volumen de la partícula (Norourzi et al., 2016), pero que es capaz de capturar la turbulencia del flujo.

4.3.1. Efecto del doble acoplamiento sobre la velocidad del flujo

Para cada concentración de partículas, se obtiene el promedio de velocidad para cada celda, y luego se obtiene el perfil en la altura del canal, lo cual se puede ver en la Figura 4.2.

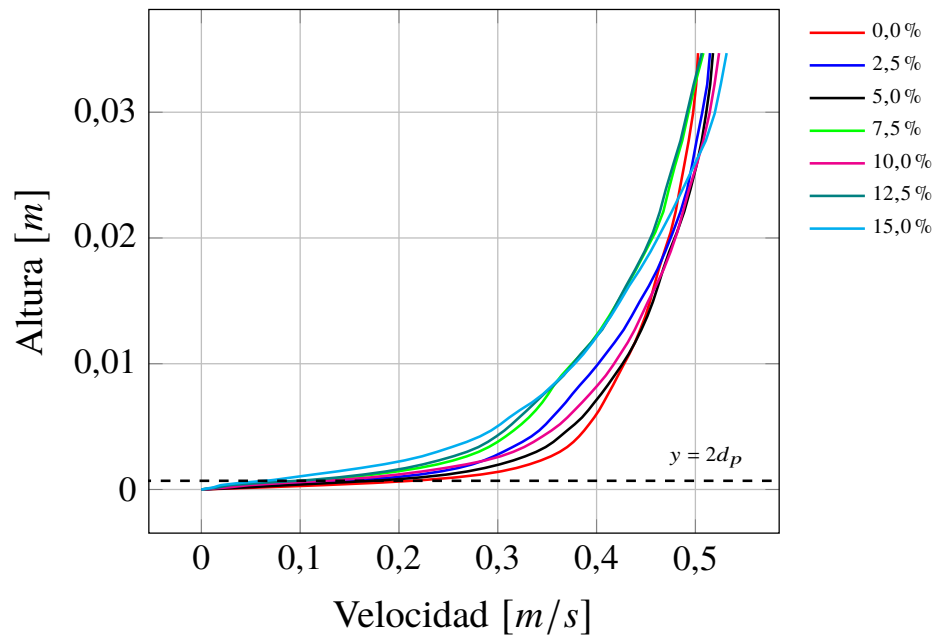


Figura 4.2: Efecto de doble acoplamiento sobre la velocidad del flujo. No se observa un patrón definido con el aumento de la concentración de partículas.

Fuente: Elaboración propia.

La velocidad promedio en $y = 2d_p$ (correspondiente a la mitad de la altura considerada para el cálculo de la concentración de partículas), para cada concentración de partículas, se muestra en la Figura 4.3

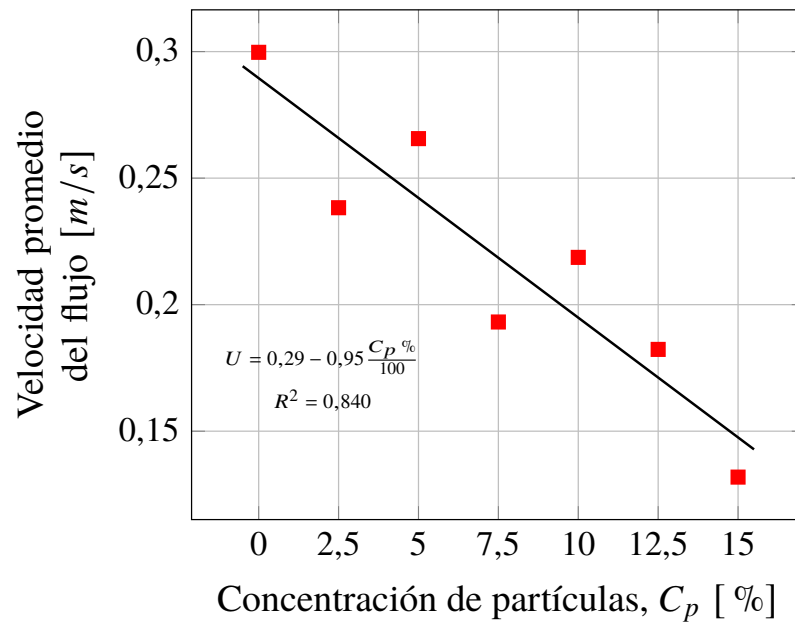


Figura 4.3: Velocidad promedio del flujo en $y = 2d_p$ para cada concentración estudiada.

Fuente: Elaboración propia.

Luego, para cada perfil se promedian los valores de velocidad en toda su altura, obteniendo las velocidades promedio de la Figura 4.4.

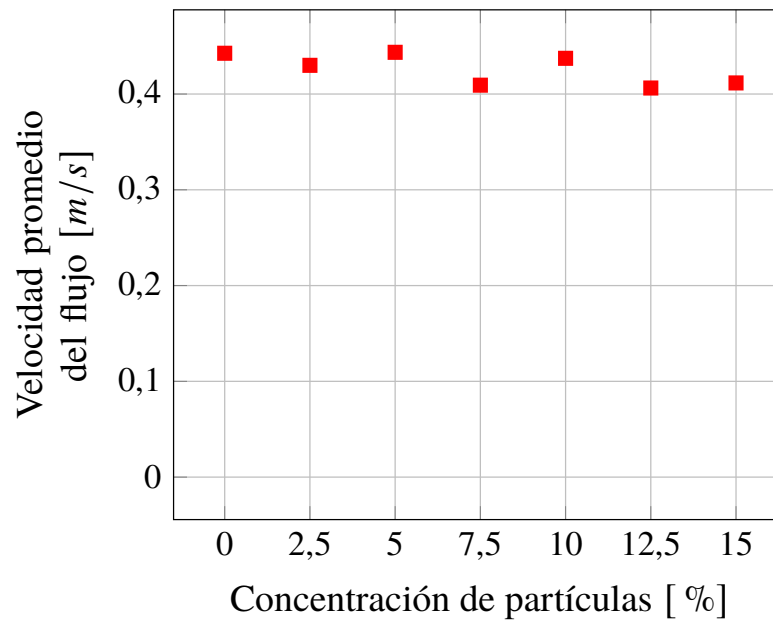


Figura 4.4: Velocidad promedio del flujo para cada concentración estudiada.

Fuente: Elaboración propia.

4.3.2. Efecto del doble acoplamiento sobre la turbulencia

Los resultados sobre la intensidad de turbulencia en el canal, para distintas concentraciones de partículas, se muestran en la Figura 4.5.

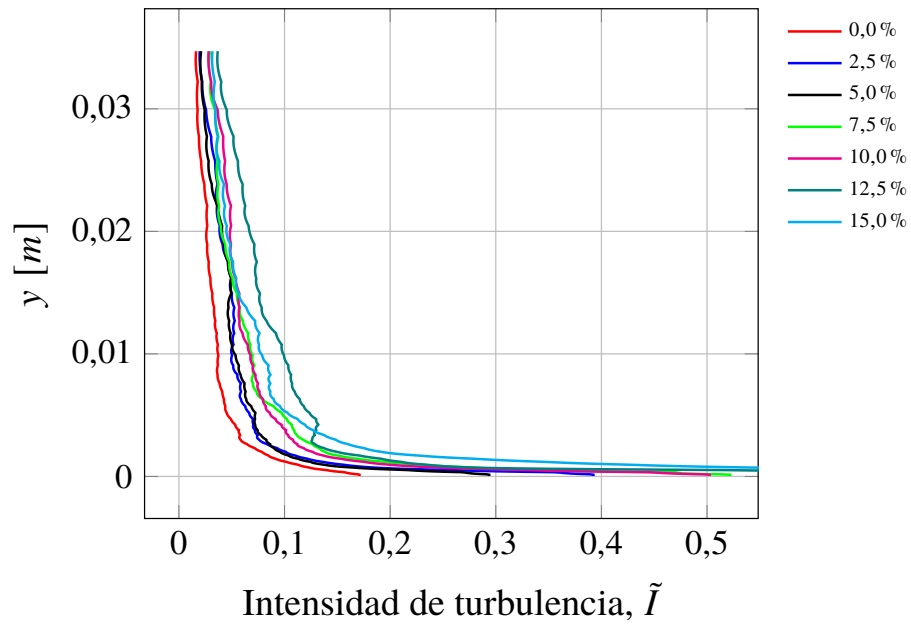


Figura 4.5: Perfil de intensidad de turbulencia para cada concentración de partículas. Se ha limitado el eje de las abscisas para mejorar la resolución del gráfico.

Fuente: Elaboración propia.

Para cada perfil, se toma el valor máximo de intensidad de turbulencia, lo cual se puede ver en la Figura 4.6 a continuación.

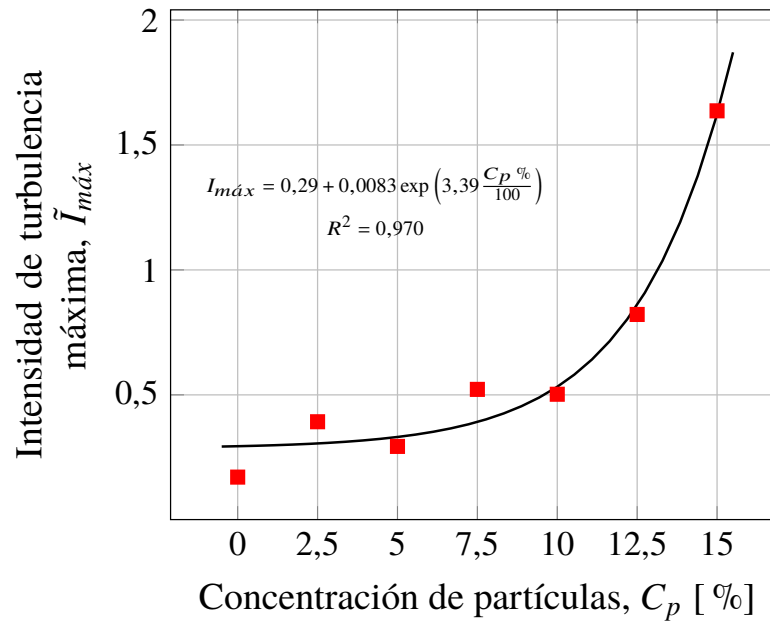


Figura 4.6: Intensidad de turbulencia máxima para cada concentración de partículas estudiada.

Fuente: Elaboración propia.

4.3.3. Flujo másico

Para cada concentración de partículas, se ha obtenido la cantidad de masa movida como arrastre de fondo que pasa por el canal en una ventana de tiempo de 1 [s], con lo cual es posible calcular el flujo másico, Q_m . En la Figura 4.7 se comparan los resultados obtenidos para las situaciones con y sin doble acoplamiento entre las fases.

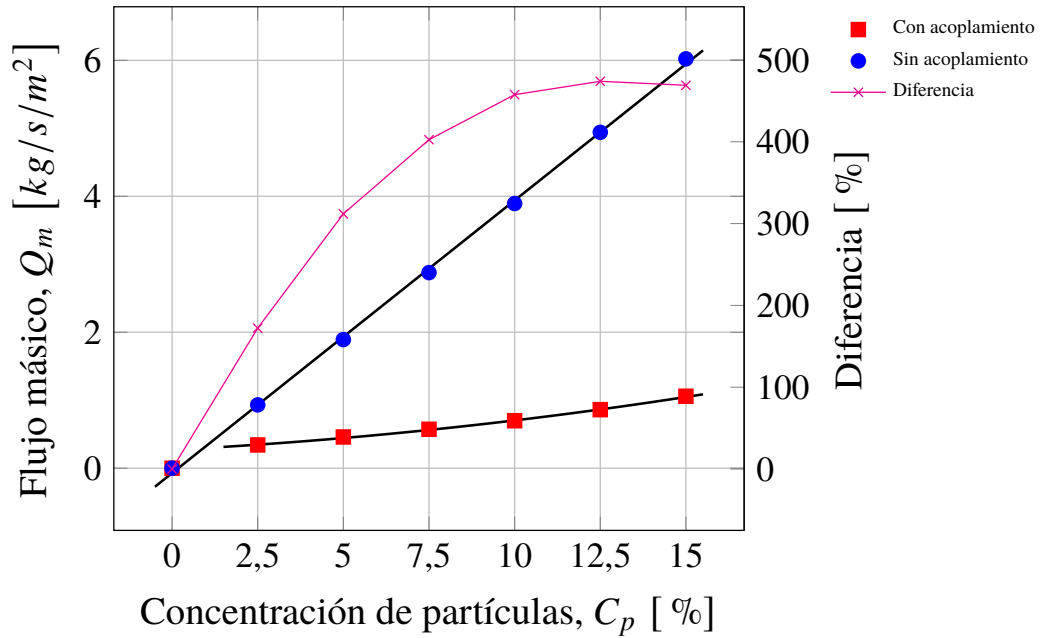


Figura 4.7: Comparación del flujo másico en el canal para cada concentración de partículas estudiada, para las situaciones con y sin doble acoplamiento. Se indica, además, la diferencia (en porcentaje) para cada par de datos considerando como base la situación con acoplamiento.

Fuente: Elaboración propia.

Las líneas de tendencia para ambos casos se presentan en negro en la Figura 4.7. La ecuación y el coeficiente de correlación para el caso con acoplamiento es la siguiente:

$$Q_m = 0,27 + 2,62 \frac{C_p}{100} + 17,08 \left(\frac{C_p}{100} \right)^2 \quad (46)$$

$$R^2 = 0,999 \quad (47)$$

Por otra parte, la ecuación y coeficiente de correlación para el caso sin acoplamiento es la siguiente:

$$Q_m = -0,07 + 40,12 \frac{C_p}{100} \quad (48)$$

$$R^2 = 0,999 \quad (49)$$

4.3.4. Costo computacional

Se ha obtenido el costo computacional de cada una de las etapas más importantes de la simulación: resolución del flujo, vuelo libre de partículas y colisiones. En la Figura 4.8 se muestran los resultados en fracción del tiempo total utilizado.

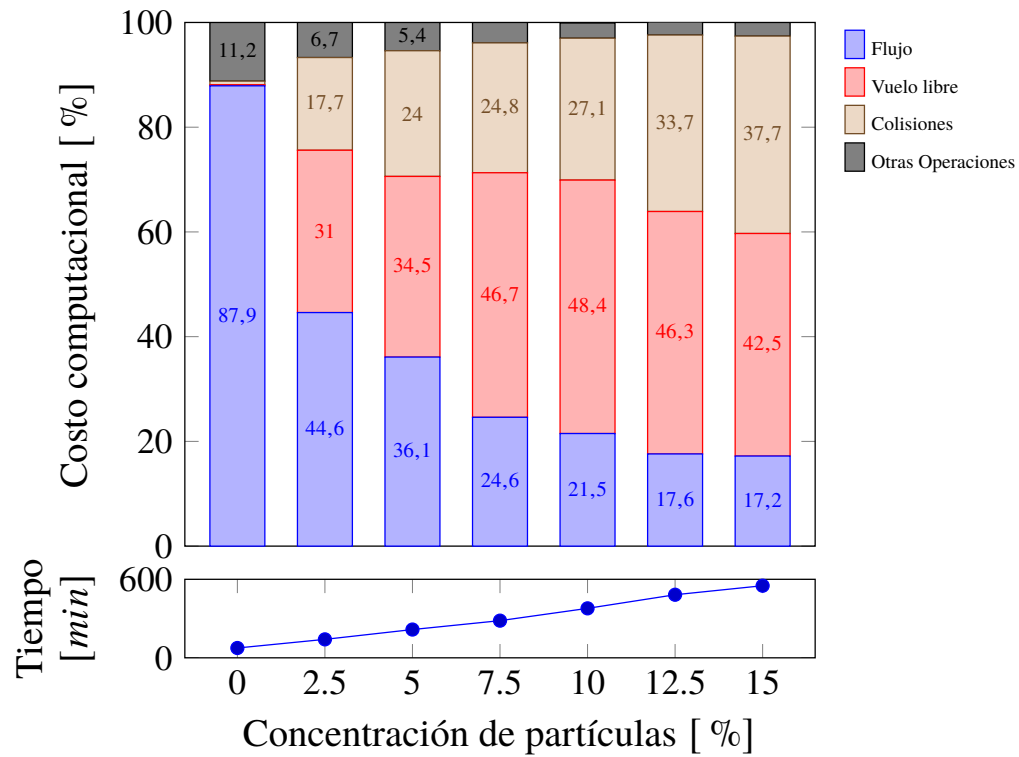


Figura 4.8: Fracción del tiempo utilizado por las etapas de resolución de flujo, movimiento de partículas y colisiones para cada concentración estudiada. Se indica, además, el tiempo total de cada simulación, en minutos.

Fuente: Elaboración propia.

CAPÍTULO 5

DISCUSIÓN Y ANÁLISIS

Las simulaciones realizadas utilizando el modelo de saltación implementado en OpenFOAM, han arrojado los resultados que se presentan en el Capítulo 4. Ahora, dichos resultados son analizados.

5.1 Validación de la fase continua

En la Figura 4.1 se compara el perfil dado por la Ley de la Pared para la velocidad de fricción objetivo de $u^* = 0,024 [m/s]$ y una altura de $h = 0,0347 [m]$, con el calculado en OpenFOAM para las mismas condiciones.

En la subcapa viscosa los perfiles son prácticamente coincidentes, mientras que en la capa intermedia los perfiles se separan, lo cual es aceptable puesto que no existe una expresión analítica que permita predecir el valor de la velocidad del flujo en ese tramo.

Por último, en la región correspondiente a la Ley Logarítmica, los perfiles son nuevamente coincidentes, observándose un aumento de la velocidad en la zona más alejada de la pared, lo cual es capturado por la Ley de la estela.

Con lo anterior, vista la coincidencia existente entre los perfiles, se establece que el flujo simulado es efectivamente turbulento y cumple con las condiciones $h = 0,0347 [m]$ y $u^* = 0,024 [m/s]$.

Por último, es de especial interés notar que en la zona en que ocurre el arrastre de fondo ($0 \leq y \leq 4d_p$) los perfiles convergen, con lo cual los subsiguientes análisis se realizan sobre la base de un flujo que simula la realidad.

5.2 Validación del modelo de saltación

El modelo de saltación implementado en OpenFOAM se ha comparado con los valores experimentales de Niño y García (1998), para una velocidad de fricción de $u^* = 0,024 [m/s]$. Para obtener los resultados, y dado que se quiere validar el modelo en términos estadísticos, se ha utilizado una velocidad de flujo menor que aquella utilizada para la validación de la fase continua. De esta manera, se han obtenido los valores promedio y desviación estándar de altura, longitud y velocidad de salto de partícula.

En particular, para la altura se ha obtenido un error en el promedio de 11,8 %, mientras que en la desviación estándar el error es de 263,8 %. En cuanto a la longitud, los errores en los parámetros respectivos son de 27,0 % y 152,6 % y finalmente en la velocidad se encuentran errores de 17,2 % y 273,4 %, respectivamente. Se puede ver, entonces, que mientras los promedios muestran valores que se condicen con los datos experimentales, las desviaciones estándar evidencian una mayor dispersión en los resultados.

Durante la realización de las simulaciones y la calibración del modelo, se han utilizado todos los valores de

los coeficientes de restitución y fricción propuestos en la Tabla 2.2, obteniendo resultados poco satisfactorios. En consecuencia, se ha encontrado que los valores $e = 0,7$ y $f = 0,2$ son los que entregan los resultados más cercanos a aquellos obtenidos en Niño y García (1998).

Con todo lo anterior, los resultados obtenidos del modelo de saltación implementado en OpenFOAM han sido efectivamente validados con los datos experimentales de Niño y García (1998), permitiendo así extender el modelo a situaciones más complejas, tanto en términos de geometría como en condiciones de flujo.

5.3 Estudio con concentraciones mayores

Se ha utilizado el modelo de saltación implementado en OpenFOAM, para estudiar el efecto de aumentar la concentración en un canal turbulento, analizando distintas variables de interés tanto físico como computacional. Las concentraciones utilizadas, y el número de partículas correspondiente se muestran en la Tabla 4.2.

5.3.1. Efecto del doble acoplamiento sobre la velocidad

Los perfiles de velocidad promedio del flujo para cada concentración de partículas estudiada se pueden ver en la Figura 4.2. Es posible ver que para todos los casos en que existen partículas en el flujo, el perfil de velocidades en la zona de arrastre es menor que aquel sin partículas, lo que es un indicador del efecto del doble acoplamiento entre las fases continua y dispersa.

En concreto, la velocidad promedio el flujo en $y = 2d_p$ (Figura 4.3) tiene un comportamiento decreciente según aumenta la concentración de partículas en el flujo, lo cual se ve reflejado en el ajuste lineal de los datos, dado por la Ec. (50).

$$U = 0,29 - 0,95 \frac{C_p \text{‰}}{100} [m/s] \quad (50)$$

De esta ecuación es posible desprender que para una concentración de aproximadamente $C_p = 30 \text{‰}$, la velocidad del fluido en $y = 2d_p$ es nula, lo que quiere decir que el flujo no genera los esfuerzos de corte suficientes para mover partículas.

Al calcular la velocidad promedio del flujo en toda la altura del canal para cada concentración, se obtienen los resultados de la Figura 4.4. En ella, se puede ver que la velocidad es prácticamente constante para todas las concentraciones (existe una variación de 8,4 % respecto a la más alta), es decir, el caudal del canal se mantiene constante independientemente de la cantidad de partículas que existan en su lecho.

Dado lo anterior, como las velocidades en la parte baja del canal tienen un comportamiento decreciente según aumenta la concentración de partículas, y siendo el caudal aproximadamente constante, es necesario que las velocidades en la parte más alta se vean aumentadas, lo cual se cumple según se puede ver en la Figura 4.2.

Finalmente, se observa que el acoplamiento de dos vías no tiene un impacto en la velocidad media del flujo, pero

sí tiene un efecto en el perfil de velocidades: en las zonas más cercanas a la pared la velocidad del flujo se ve reducida, mientras que en las zonas más alejadas se ve aumentada.

5.3.2. Efecto del doble acoplamiento sobre la turbulencia

La intensidad de turbulencia es una medida de la fluctuación de velocidad del flujo respecto de su valor promedio en el tiempo, con lo cual se puede tener una referencia del comportamiento de este.

En la Figura 4.5 se pueden ver los perfiles de intensidad de turbulencia para cada concentración de partículas. Para todos los casos, en $y = 0$ la intensidad de turbulencia es 0 (la velocidad es nula) para luego aumentar rápidamente a su mayor valor y finalmente decaer gradualmente a su valor mínimo. Por último, se puede ver que en general aquellas concentraciones para las que la intensidad de turbulencia máxima es mayor, su descenso es más lento.

Hay dos motivos por los que se produce el aumento de intensidad de turbulencia en las zonas más cercanas a la pared: El primero es que en la parte baja del canal es donde se genera la turbulencia, por lo que hay una mayor cantidad de remolinos y aleatoriedad en el flujo; el segundo, es que la presencia de partículas inevitablemente ejerce sobre el flujo fuerzas que se traducen en mayores fluctuaciones en las velocidades, y por tanto una mayor intensidad de turbulencia.

Lo anteriormente expuesto se grafica mejor en la Figura 4.6, donde se puede ver que la intensidad de turbulencia máxima tiene un crecimiento exponencial conforme aumenta la concentración de partículas, lo cual se refleja en la ecuación de mejor ajuste de los datos:

$$I_{max} = 0,29 + 0,0083 \exp\left(3,39 \frac{C_p \%}{100}\right) \quad (51)$$

Es evidente que la presencia de partículas en el flujo tiene un efecto importante en la intensidad de turbulencia, aumentándola de manera exponencial y propagándola hacia niveles superiores.

5.3.3. Flujo másico

El flujo de masa a través del canal se ha cuantificado promediando el número de partículas que pasan a través de tres superficies de control, equiespaciadas en la longitud del canal. En la Figura 4.7 se pueden ver los resultados obtenidos para las situaciones con y sin acoplamiento.

En cuanto a la situación con acoplamiento, se puede observar un crecimiento cuadrático del movimiento de masa conforme la concentración aumenta (Ec. (46)), llegando hasta un valor de $Q_m = 1,06 [kg/s/m^2]$ en la concentración más alta. Para el caso sin acoplamiento, el crecimiento es lineal (Ec. (48)), con un valor máximo de $Q_m = 6,02 [kg/s/m^2]$ para la concentración más alta.

Si bien ambos casos son estrictamente crecientes, para los casos físicamente posibles el flujo másico de la

situación sin acoplamiento es siempre mayor. Es más, para las concentraciones de partículas más altas de este estudio, la diferencia relativa del caso sin acoplamiento respecto a aquel con acoplamiento llega hasta un valor de 474 %.

La situación anterior es un reflejo del efecto del doble acoplamiento entre las fases continua y dispersa: La disminución de velocidad en las zonas bajas del canal, debido a la resistencia opuesta por las partículas, incide directamente en la capacidad del flujo de mover sedimentos.

5.3.4. Costo computacional

Se ha calculado el tiempo utilizado en cada fase de la simulación para cada concentración de partículas estudiada. Los resultados se resumen en la Figura 4.8.

Para el caso sin partículas, se tiene que el 87,9 % del costo computacional está asociado a la resolución del flujo, y un 11,2 % está dedicado a otras operaciones tales como la escritura de datos y funciones auxiliares. Se observa, además que a pesar de que no hay partículas participando de la simulación, de igual manera el programa pasa por las rutinas correspondientes, sumando hasta un 0,9 % del tiempo total utilizado.

Cuando la concentración aumenta a 2,5 %, tan sólo un 44,6 % del tiempo es utilizado para la resolución del flujo (es decir, el mismo tiempo, en minutos, que el caso anterior). En cambio, las rutinas de colisión y vuelo libre toman mayor relevancia, alcanzando valores de 17,7 % y 31,0 % del costo total, respectivamente.

Para los casos siguientes, las rutinas asociadas a la fase dispersa aumentan gradualmente su participación, llegando a utilizar hasta un 80,2 % del total del tiempo de la simulación, mientras que la resolución del flujo disminuye su costo, llegando a un 17,2 %.

En los casos con mayor concentración de partículas (10,0 %, 12,5 % y 15,0 %), se ve que el tiempo de cálculo del flujo aumenta gradualmente. Este aumento se debe a que, al existir una mayor cantidad de partículas, y a que las fases se encuentran acopladas entre sí (en las cuatro vías posibles), se generan fuerzas (y por consiguiente, mayores velocidades), sobre celdas individuales que obligan al programa a reducir la discretización del tiempo, manteniendo la estabilidad pero aumentando el tiempo de ejecución del programa.

CAPÍTULO 6

CONCLUSIONES

El modelo de saltación de partículas implementado en OpenFOAM muestra concordancia con los resultados experimentales de Niño y García (1998) para partículas de arena de 0,69 [mm] de diámetro en un canal turbulento con velocidad de fricción de $u^* = 0,024$ [m/s]. En efecto, los resultados muestran una alta correlación entre los valores experimentales y los simulados en los tres parámetros estudiados: altura, longitud y velocidad de salto. Además, se han implementado satisfactoriamente las cuatro vías de acople entre las fases continua y dispersa.

Los resultados obtenidos para la altura, longitud y velocidad de salto en un canal turbulento de características $h = 0,0347$ [m] y $u^* = 0,024$ [m/s] se presentan en la Tabla 4.1. Teniendo en cuenta que el problema modelado es complejo y que existe la propagación de errores debido a las simplificaciones realizadas para reducir las variables y cálculos implicados, se puede decir que el modelo de saltación implementado muestra un comportamiento acorde a la realidad, y por lo tanto es posible utilizarlo como herramienta para el estudio del fenómeno de arrastre de fondo con distintas condiciones de flujo, geometrías y granulometrías, permitiendo mejorar el entendimiento de la física del problema a microescala.

Sumado a lo anterior, para la interacción entre partículas se ha implementado el modelo de esfera blanda, lo que permite múltiples colisiones simultáneamente, pudiendo así trabajar con altas concentraciones de partículas en el flujo estudiado. También, y junto con lo anterior, se ha incluido el acoplamiento entre las fases continua y dispersa, a través del término incorporado a la ecuación de Navier-Stokes (Ec. (44)).

Con todo lo anterior, se hizo el estudio del efecto de aumentar la concentración de partículas en el flujo, para concentraciones entre 0 y 15 % (Tabla 4.2), lo que abarca desde la ausencia de partículas hasta los flujos hiperconcentrados.

Con respecto al efecto del doble acoplamiento sobre el flujo, se observó que, al aumentar la concentración de partículas, la velocidad del flujo en la zona en que estas se mueven disminuye, encontrándose una relación lineal entre ambas variables, que se presenta en la Ec. (50).

La intensidad de turbulencia también fue analizada, encontrándose que el aumento de la concentración de partículas afecta de manera exponencial (Ec. 51) la intensidad de turbulencia máxima. Además, fue posible ver que el aumento de la intensidad de turbulencia en la parte más baja del canal se propaga hacia las zonas más altas.

No obstante los puntos anteriores, se pudo ver que el doble acoplamiento tiene, sobre el flujo, un efecto más bien local, modificando los perfiles de velocidad y aumentando la intensidad de turbulencia, pero manteniendo el caudal constante, lo que queda de manifiesto en la Figura 4.4.

En cuanto al movimiento de sólidos dentro del canal, se observó que este tiene un crecimiento cuadrático conforme aumenta la concentración cuando se considera la situación con acoplamiento, mientras que el crecimiento es lineal para el caso sin acoplamiento. También, se pudo ver que para los casos físicamente posibles ($C_p < 100$ [%]), el

acoplamiento entre fases genera una disminución importante del flujo másico, llegando a una diferencia relativa del 474 %.

El doble acoplamiento, entonces, tiene dos aristas: En el flujo no tiene efecto sobre el caudal total, pero sí tiene efecto sobre la turbulencia y el perfil de velocidades; en el movimiento de partículas su efecto es más importante, ya que existe si bien para cada concentración el flujo másico aumenta (Ec. (46)), este flujo es siempre menor que en el caso sin acoplamiento, llegando a una diferencia de $Q_m = 4,96 \text{ [kg/m}^2\text{/s]}$ para la concentración más alta de sedimentos.

El costo computacional del modelo de saltación implementado en OpenFOAM fue calculado para cada concentración de partículas (Figura 4.8): se encontró que conforme esta aumenta, las subrutinas asociadas al vuelo libre y desarrollo de colisiones cobran más relevancia frente al tiempo utilizado en el cálculo de la fase continua. Además, se pudo ver que el tiempo total utilizado, en minutos, aumenta en forma lineal a medida que también lo hace la concentración de partículas.

Finalmente, gracias al desarrollo del modelo de saltación de partículas en un software optimizado, de alto desempeño y además extensible como OpenFOAM, es posible realizar diversos estudios que permitan entregar mejores conocimientos y mejores herramientas tanto a la academia como la industria.

REFERENCIAS

- Bailly, C. & Comte-Bellot, G. (2015). *Turbulence* (1.^a ed.). Springer.
- CFD Direct Ltd. (2017). *OpenFOAM User Guide version 5.0*. Ver. 5.0.
- Crowe, C., Schwarzkopf, J., Sommerfeld, M. & Tsuji, Y. (2012). *Multiphase Flows with Droplets and Particles* (2.^a ed.). CRC Press.
- Elghobashi, S. (1994). On predicting particle-laden turbulent flows. *Applied Scientific Research*, 52(4), 309-329
- García, M. & Niño, Y. (1992). Lagrangian description of bedload transport by saltating particles. *J.T. Kuo and G.F. Lin (eds.), Stochastic Hydraulics: Proc. 6th IAHR International Symposium*, 259-266.
- González, A. E. (2008). *Coupled numerical modeling of sediment transport near the bed using a two-phase flow approach* (Ph.D. Thesis, University of California, Davis).
- Holzmann, T. (2017). *Mathematics, Numerics, Derivations and OpenFOAM* (1.^a ed.). Holzmann CFD.
- Julien, P. Y. (2010). *Erosion and Sedimentation* (2.^a ed.). Cambridge University Press.
- Lee, H.-Y. (2000). Investigations of Continuous Bed Load Saltating Process. *Journal of Hydraulic Engineering-asce - J HYDRAUL ENG-ASCE*, 126.
- Lee, H.-Y. et al. (2006). On three-dimensional continuous saltating process of sediment particles near the channel bed. *Journal of Hydraulic Research*, 44(3), 374-389
- Lee, H.-Y. & Hsu, I. (1994). Investigation of Saltating Particle Motions. *Journal of Hydraulic Engineering*, 120(7), 831-845
- Lukerchenko, N., Piatsevich, S., Chara, Z. & Vlasak, P. (2009). Numerical model of spherical particle saltation in a channel with a transversely tilted rough bed. *Journal of Hydrology and Hydromechanics*, 57.
- Merritt, W., Letcher, R. & Jakeman, A. (2003). A Review of Erosion and Sediment Transport Models. *Environmental Modelling & Software*, 18, 761-799.
- Moreno, P. (2014). *Two-phase flow, three-dimensional approach for saltating particles near the bed at finite reynolds numbers* (Ph.D. Thesis, University of California, Davis).
- Moreno, P. & Bombardelli, F. (2012). 3D numerical simulation of particle-particle collisions in saltation mode near stream beds. *Acta Geophysica*, 60, 1661-1688.
- Moreno, P. & Bombardelli, F. (2016). Computation of the Basset force: recent advances and environmental flow applications. *Environmental Fluid Mechanics*, 16(1), 193-208
- Mukha, T. & Liefvendahl, M. (2015). *Large-Eddy Simulation of Turbulent Channel Flow*.
- Niño, Y. & García, M. (1994). Gravel Saltation: 2. Modeling. *Water Resources Research*, 30(6), 1915-1924.
- Niño, Y. & García, M. (1998). Using Lagrangian particle saltation observations for bedload sediment transport modelling. *Hydrological Processes*, 12(8), 1197-1218.
- NLHPC. (2019). Recursos Computacionales NLHPC. [Internet; descargado 4-Noviembre-2019]. Recuperado desde http://usuarios.nlhpc.cl/index.php/Recursos_Computacionales_NLHPC
- Norourzi, H., Zarghami, R., Sotudeh-Gharebagh, R. & Mostoufi, N. (2016). *Coupled CFD-DEM Modeling: Formulation, Implementation and Application to Multiphase Flows* (1.^a ed.).

- Pope, S. B. (2000). *Turbulent Flows* (1.^a ed.). Cambridge University Press.
- Prosperetti, A. & Tryggvason, G. (2007). *Computational Methods for Multiphase Flow* (1.^a ed.). Cambridge University Press.
- Schmeeckle, M. W., Nelson, J. M., Pitlick, J. & Bennett, J. P. (2001). Interparticle collision of natural sediment grains in water. *Water Resources Research*, 37(9), 2377-2391
- Tsuji, Y., Morikawa, Y., Tanaka, T., Nakatsukasa, N. & Nakatani, M. (1987). Numerical simulation of gas-solid two-phase flow in a two-dimensional horizontal channel. *International Journal of Multiphase Flow*, 13(5), 671-684
- van Hinsberg, M., ten Thije Boonkamp, J. & Clercx, H. (2011). An efficient, second order method for the approximation of the Basset history force. *Journal of Computational Physics*, 230(4), 1465-1478
- Wiberg, P. L. & Smith, J. D. (1985). A theoretical model for saltating grains in water. *Journal of Geophysical Research: Oceans*, 90(C4), 7341-7354.
- Yamamoto, Y., Potthoff, M., Tanaka, T., Kajishima, T. & Tsuji, Y. (2001). Large-eddy simulation of turbulent gas-particle flow in a vertical channel: Effect of considering inter-particle collisions. *Journal of Fluid Mechanics*, 442, 303-334.
- Yen, B. (1992). Sediment Fall Velocity in Oscillating Flow. *Water resources and environmental engineering research report*.
-

ANEXOS

ANEXO A
TABLAS**Tabla A.1:** Coeficientes para el cálculo de C_T .

Re_R	C_1	C_2	C_3
0 – 1	0	50,27	0
1 – 10	0	50,27	0,0418
10 – 20	5,32	37,20	5,32
20 – 50	6,44	32,20	6,44
50+	6,45	32,10	6,45

Fuente: Yamamoto et al. (2001).

Tabla A.2: Promedio y desviación estándar de altura, longitud y velocidad de saltos obtenidos para diez partículas simuladas en OpenFOAM. Los resultados de altura y longitud se han normalizado con d_p , y los de velocidad con u^* . Para cada partícula se han considerado más de 100 saltos.

Partícula	Nº de saltos	Altura		Longitud		Velocidad	
		Promedio	Desv. Est.	Promedio	Desv. Est.	Promedio	Desv. Est.
1	113	0.86	1.33	10.69	12.97	6.94	8.08
2	104	1.05	1.26	12.72	10.08	6.76	4.61
3	116	0.80	0.95	10.50	9.91	6.07	4.13
4	113	1.12	2.09	12.35	15.39	6.93	5.2
5	109	0.99	1.20	12.41	11.76	6.99	4.9
6	121	0.86	0.95	10.87	9.86	6.83	3.94
7	113	0.93	1.29	11.09	11.18	6.23	3.8
8	117	0.99	1.80	11.22	12.31	6.04	9.7
9	124	0.90	1.17	11.74	11.80	6.98	5.2

Fuente: Elaboración propia.

ANEXO B

MODELOS DE COLISIÓN

B.1. Modelo de esfera dura

Este modelo, utilizado para la colisión con el fondo, se basa en la fuerza de impulso aplicada sobre la partícula en el momento de la colisión (Crowe et al., 2012). Como se trata de una partícula rígida, las ecuaciones de impulso son las siguientes:

$$m(\mathbf{u}_p - \mathbf{u}_p^{(0)}) = \mathbf{J} \quad (\text{B.1})$$

$$I(\boldsymbol{\omega}_p - \boldsymbol{\omega}_p^{(0)}) = \frac{d_p}{2} \hat{\mathbf{n}} \times \mathbf{J} \quad (\text{B.2})$$

Donde m es la masa, $\mathbf{u}_p$ la velocidad, $\boldsymbol{\omega}_p$ la velocidad angular, I es la inercia y d_p el diámetro de la partícula. $\hat{\mathbf{n}}$ es el vector unitario normal al plano de colisión, $\mathbf{J}$ es la fuerza de impulso sobre la partícula. El superíndice (0) indica el valor antes de la colisión. En el sistema anterior se identifican dos incógnitas: $\mathbf{u}_p$ y $\mathbf{J}$. Para resolver el problema se hacen las siguientes suposiciones:

1. No existe deformación de la partícula, por lo que la distancia del centro de la partícula al punto de colisión es constante,
2. La fricción durante el deslizamiento obedece la ley de fricción de Coulomb y,
3. Cuando una partícula deja de deslizar, ya no ocurrirá más deslizamiento.

La velocidad del punto de contacto antes de la colisión es:

$$\mathbf{u}_{p_c}^{(0)} = \mathbf{u}_p^{(0)} + \frac{d_p}{2} \boldsymbol{\omega}_p^{(0)} \times \hat{\mathbf{n}} \quad (\text{B.3})$$

Cuya componente tangencial está dada por:

$$\mathbf{u}_{p_{ct}}^{(0)} = \mathbf{u}_{p_c}^{(0)} - (\mathbf{u}_{p_c}^{(0)} \cdot \hat{\mathbf{n}}) \hat{\mathbf{n}} + \frac{d_p}{2} \boldsymbol{\omega}_p^{(0)} \times \hat{\mathbf{n}} \quad (\text{B.4})$$

La fuerza de impulso se puede expresar en sus componentes normal y tangencial:

$$\mathbf{J} = J_n \hat{\mathbf{n}} + J_t \mathbf{t} \quad (\text{B.5})$$

El vector unitario $\mathbf{t}$ en la dirección tangencial esta dado por:

$$\mathbf{t} = \frac{\mathbf{u}_{pct}^{(0)}}{|\mathbf{u}_{pct}^{(0)}|} \quad (\text{B.6})$$

La velocidad antes y después de la colisión se relaciona a través del coeficiente de restitución.

$$\hat{\mathbf{n}} \cdot \mathbf{u}_p = -e(\hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)}) \quad (\text{B.7})$$

La ecuación anterior se considera la definición del coeficiente de restitución e . Multiplicando la Ec. (B.1) por el vector $\mathbf{n}$, se obtiene:

$$\hat{\mathbf{n}} \cdot \mathbf{u}_p = \hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)} + \frac{\hat{\mathbf{n}} \cdot \mathbf{J}}{m} \quad (\text{B.8})$$

Sustituyendo la Ec. (B.7) en el lado izquierdo de la ecuación anterior, se obtiene:

$$-e(\hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)}) = \hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)} + \frac{J_n}{m} \quad (\text{B.9})$$

De esta ecuación se puede obtener J_n de manera explícita:

$$J_n = -m(1 + e)(\hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)}) < 0 \quad (\text{B.10})$$

Si se asume que la partícula desliza, la componente tangencial de la fuerza impulsiva, J_t , viene dada por:

$$J_t = fJ_n < 0 \quad (\text{B.11})$$

Según la ley de fricción de Coulomb, el coeficiente de fricción f y el coeficiente de restitución e se consideran como parámetros conocidos. Reemplazando la fuerza impulsiva $\mathbf{J}$ de las ecuaciones (B.10) y (B.11) en (B.1) y (B.2):

$$\mathbf{u}_p = \mathbf{u}_p^{(0)} - (1 + e)(\hat{\mathbf{n}} + f\mathbf{t})(\hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)}) \quad (\text{B.12})$$

$$\omega_p = \omega_p^{(0)} - \frac{5}{d_p}(1 + e)(\hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)})f(\hat{\mathbf{n}} \times \mathbf{t}) \quad (\text{B.13})$$

Estas ecuaciones corresponden al caso en que la partícula desliza. A partir del valor de la componente tangencial de la velocidad se puede determinar si la partícula sigue deslizando o no. Considerando el caso en que desliza, la velocidad después de la colisión está dada por:

$$\mathbf{u}_{p_c} = \mathbf{u}_p + \frac{d_p}{2} \omega_p \times \hat{\mathbf{n}} \quad (\text{B.14})$$

Usando las ecuaciones (B.1), (B.2) y (B.14):

$$\mathbf{u}_{p_c} = \mathbf{u}_{p_c}^{(0)} + \frac{\mathbf{J}}{m} + \frac{d_p}{2} \omega_p \times \hat{\mathbf{n}} \quad (\text{B.15})$$

$$\mathbf{u}_{p_c} = \mathbf{u}_{p_c}^{(0)} + \left(J_n \hat{\mathbf{n}} + \frac{7}{2} J_t \mathbf{t} \right) \frac{1}{m} \quad (\text{B.16})$$

Tomando la componente tangencial de la ecuación anterior, y reemplazando por la Ec. (B.6):

$$\mathbf{u}_{p_{ct}} = \frac{\mathbf{u}_{p_{ct}}^{(0)}}{|\mathbf{u}_{p_{ct}}^{(0)}|} \left(|\mathbf{u}_{p_{ct}}^{(0)}| + \frac{7}{2} J_t \frac{1}{m} \right) \quad (\text{B.17})$$

El caso en que la partícula desliza, viene dado por la condición en que el signo del paréntesis anterior es positivo, esto es:

$$J_t > -\frac{2}{7} m |\mathbf{u}_{p_{ct}}^{(0)}| \quad (\text{B.18})$$

El impulso J_t se puede expresar en términos de condiciones antes de la colisión a través de las ecuaciones (B.10) y (B.11). Así:

$$\frac{\hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)}}{|\mathbf{u}_{p_{ct}}^{(0)}|} < \frac{2}{7} \frac{1}{f(1+e)} \quad (\text{B.19})$$

Si esta relación no se satisface, la velocidad en el punto del contacto posterior a la colisión es cero, esto es, el lado izquierdo de la Ec. (B.17) es cero. Por lo tanto:

$$J_t = -\frac{2}{7} m |\mathbf{u}_{p_{ct}}^{(0)}| \quad (\text{B.20})$$

Reemplazando las ecuaciones (B.10) y (B.20) en las ecuaciones (B.1) y (B.2), se obtienen las velocidades para el caso en que no hay deslizamiento:

$$\mathbf{u}_p = \mathbf{u}_p^{(0)} - \left\{ (1 + e)(\hat{\mathbf{n}} \cdot \mathbf{u}_p^{(0)})\hat{\mathbf{n}} + \frac{2}{7}|\mathbf{u}_{p_{ct}}^{(0)}|\mathbf{t} \right\} \quad (\text{B.21})$$

$$\omega_p = \omega_p^{(0)} - \frac{10}{7d_p}|\mathbf{u}_{p_{ct}}^{(0)}|(\hat{\mathbf{n}} \times \mathbf{t}) \quad (\text{B.22})$$

B.2. Modelo de esfera blanda

Este modelo reemplaza las deformaciones que produciría un choque entre partículas con un solapamiento entre las mismas, y mediante un modelo de resorte-amortiguador como los de la Figura 2.8, se calculan las fuerzas repulsivas sobre cada partícula, tanto normales como tangenciales (Crowe et al., 2012). A diferencia del modelo de esfera dura, el modelo de esfera blanda permite colisiones entre más de dos partículas simultáneamente, por lo que es ideal para flujo con altas concentraciones de sedimentos.

El modelo de esfera blanda se basa en tres parámetros principales:

- Rigidez, k
- Amortiguamiento, η
- Coeficiente de fricción, f (obtenido empíricamente)

Así, la fuerza normal ejercida por la partícula j sobre la partícula i está dada por:

$$\mathbf{F}_{nij} = (-k_n\delta_n - \eta_{nj}\mathbf{G} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} \quad (\text{B.23})$$

Donde δ_n es el desplazamiento en dirección de la fuerza, $\mathbf{G}$ es la velocidad relativa entre las partículas y $\hat{\mathbf{n}}$ es un vector unitario en la dirección de la línea entre los centros de ambas partículas (perpendicular al plano de colisión).

La fuerza tangencial ejercida por la partícula j sobre la partícula i está dada por:

$$\mathbf{F}_{tij} = -k_t\delta_t - \eta_{tj}\mathbf{G}_{ct} \quad (\text{B.24})$$

Donde $\mathbf{G}_{ct}$ es la velocidad de deslizamiento entre las partículas, dado por:

$$\mathbf{G}_{ct} = \mathbf{G} - (\mathbf{G} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} + \frac{d_i}{2}\omega_i \times \hat{\mathbf{n}} + \frac{d_j}{2}\omega_j \times \hat{\mathbf{n}} \quad (\text{B.25})$$

Donde ω es la velocidad angular y d es el diámetro de la partícula. En las ecuaciones anteriores, los subíndices $\hat{\mathbf{n}}$ y $\mathbf{t}$ denotan las componentes en la dirección normal y tangencial, respectivamente.

Si se cumple la condición:

$$|\mathbf{F}_{tij}| > f|\mathbf{F}_{nij}| \quad (\text{B.26})$$

La partícula i deslizará y por lo tanto en vez de la Ec. (B.24), la fuerza tangencial está dada por:

$$\mathbf{F}_{tij} = -f|\mathbf{F}_{nij}|\mathbf{t} \quad (\text{B.27})$$

Que corresponde a la ley de fricción de Coulomb. $\mathbf{t}$ es un vector unitario dado por:

$$\mathbf{t} = \frac{\mathbf{G}_{ct}}{|\mathbf{G}_{ct}|} \quad (\text{B.28})$$

De esta forma, la fuerza y el torque total ejercida sobre una partícula i será:

$$\mathbf{F}_i = \sum_j (\mathbf{F}_{nij} + \mathbf{F}_{tij}) \quad (\text{B.29})$$

$$\mathbf{T}_i = \sum_j \left(\frac{d_i}{2} \hat{\mathbf{n}} \times \mathbf{F}_{tij} \right) \quad (\text{B.30})$$

Es de notar que se hace necesario discretizar el tiempo de colisión en intervalos muy pequeños para representar de forma adecuada todo el proceso, y por lo mismo este modelo se hace más costoso en tiempo computacional.

B.2.1 Determinación de la rigidez

La rigidez k_n se puede determinar mediante la teoría de contacto Hertziano cuando se conocen tanto el módulo de Young E como la razón de Poisson σ . Así, la rigidez se calcula como:

$$k_n = \frac{4}{3} \left(\frac{1 - \sigma_i^2}{E_i} + \frac{1 - \sigma_j^2}{E_j} \right)^{-1} \left(\frac{1}{r_i} + \frac{1}{r_j} \right)^{-1/2} \quad (\text{B.31})$$

Donde r es el radio de la partícula y los subíndices i y j denotan las partículas i y j , respectivamente.

La rigidez k_t , por su parte, se puede determinar utilizando la teoría de Mindlin (Crowe et al., 2012):

$$k_t = 8 \left(\frac{1 - \sigma_i^2}{H_i} + \frac{1 - \sigma_j^2}{H_j} \right)^{-1} \left(\frac{1}{r_i} + \frac{1}{r_j} \right)^{-1/2} \delta_n^{1/2} \quad (\text{B.32})$$

Donde H es el módulo de corte y está dado por:

$$H = \frac{E}{2(1 + \sigma)} \quad (\text{B.33})$$

B.2.2 Determinación del amortiguamiento

El amortiguamiento se calcula como (Crowe et al., 2012):

$$\eta_n = \alpha \sqrt{mk} \delta^{1/4} \quad (\text{B.34})$$

Donde m es la masa de la partícula y α se relaciona con el coeficiente de restitución e , según se muestra en la Figura B.1. Para el amortiguamiento en dirección tangencial se utiliza el mismo valor.

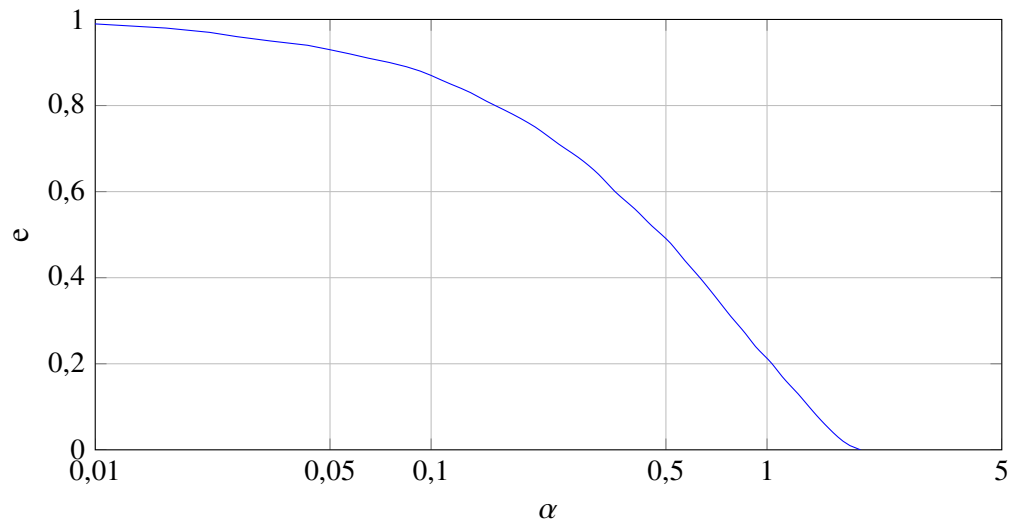


Figura B.1: Relación entre el coeficiente α y el coeficiente de restitución e .

Fuente: Adaptación de Crowe et al. (2012).

ANEXO C

ALGORITMO PIMPLE

Resolver la ecuación de Navier-Stokes requiere de técnicas numéricas para el acoplamiento de la presión y el momentum del flujo (Holzman, 2017). Para entender el problema, es necesario ver las dificultades que se presentan al resolver las ecuaciones. Tomando la ecuación general de momentum

$$\frac{\partial}{\partial t} \rho \mathbf{u} = -\nabla \cdot (\rho \mathbf{u} \times \mathbf{u}) - \nabla \cdot \boldsymbol{\tau} - \nabla p + \rho \mathbf{g} \quad (\text{C.1})$$

Aplicando la condición de incompresibilidad (densidad constante):

$$\rho \frac{\partial}{\partial t} \mathbf{u} = -\rho \nabla \cdot (\mathbf{u} \times \mathbf{u}) - \nabla \cdot \boldsymbol{\tau} - \nabla p + \rho \mathbf{g} \quad (\text{C.2})$$

Dividiendo la ecuación anterior por la densidad ρ , se obtiene:

$$\frac{\partial}{\partial t} \mathbf{u} = -\nabla \cdot (\mathbf{u} \times \mathbf{u}) - \nabla \cdot 2\nu \mathbf{D} - \nabla \frac{p}{\rho} + \mathbf{g} \quad (\text{C.3})$$

Donde $\mathbf{D}$ es el tensor de razón de deformación $\mathbf{D} = \frac{1}{2} [\nabla \times \mathbf{u} + (\nabla \times \mathbf{u})^T]$.

Como la densidad es constante, no hay necesidad de resolver ninguna ecuación de energía. Se puede ver que existen cuatro incógnitas, la presión p y las tres componentes de velocidad. Sin embargo, sólo se tienen tres ecuaciones (la ecuación de momentum para cada dirección), por lo que es necesario utilizar una cuarta que permita resolver el problema. Esta ecuación es la de conservación de masa.

$$\nabla \cdot \mathbf{u} = 0 \quad (\text{C.4})$$

Esta ecuación no incluye la presión, por lo que se deben utilizar técnicas especiales para resolver la ecuación de momentum. Esto se conoce como el problema de *acoplamiento presión-momentum*.

Se resuelve el problema aplicando el operador divergencia en la ecuación de momentum (C.3). Luego, se realiza la discretización de la derivada temporal, con lo cual se puede usar la ecuación de conservación de masa para eliminar términos y terminar con la ecuación de Poisson para presión.

$$\nabla^2 p = -\mathbf{g} \quad (\text{C.5})$$

Ahora contamos con una ecuación para el momentum y la presión. La estrategia, ahora, es encontrar una presión y momentum al que se satisfagan la conservación de masa y las condiciones del caso. Esto se logra con los algoritmos de acoplamiento presión-momentum PISO, SIMPLE y PIMPLE.

C.1. Algoritmo SIMPLE

El algoritmo SIMPLE (Semi Implicit Method of Pressure Linked Equations) permite calcular la solución para un instante t fijo, eliminando las derivadas temporales, por lo cual se utiliza en problemas estacionarios. Dado que se elimina un término del problema, las ecuaciones se deben subrelajar, a modo obtener estabilidad.

C.2. Algoritmo PISO

El algoritmo PISO (Pressure Implicit Split Operator) incluye la derivada temporal, por lo que permite resolver problemas transientes. Sin embargo, se debe cumplir un criterio para mantener la estabilidad de la simulación.

El número de Courant, dado por:

$$Co = \frac{\mathbf{u}\Delta t}{\Delta x} \quad (C.6)$$

Debe ser menor que la unidad. Esta condición restringe el paso de información a celdas contiguas, para obtener la continuidad deseada en el flujo.

C.3. Algoritmo PIMPLE

El algoritmo PIMPLE combina los algoritmos SIMPLE y PISO, y permite calcular problemas transientes sin la restricción de usar un número de Courant bajo ($Co \leq 1$). El principio de funcionamiento es el siguiente: Para cada paso de tiempo, se busca una solución estacionaria con subrelajación, para una tolerancia dada, y se continua al siguiente instante $t_{n+1} = t_n + \Delta t$. Este proceso se repite hasta llegar al tiempo final.

ANEXO D

ARCHIVOS OPENFOAM

A continuación se muestran los archivos para la simulación y el *solucionador* `pimpleLPTFoam` implementado en OpenFOAM en su versión 5.0. También se presenta el árbol de archivos modificados de la librería `intermediate`. Tanto el *solucionador* como la librería modificada se encuentran alojados en el repositorio <https://github.com/cojua8/SaltationSolvers>

D.1. Solver `pimpleLPTFoam`

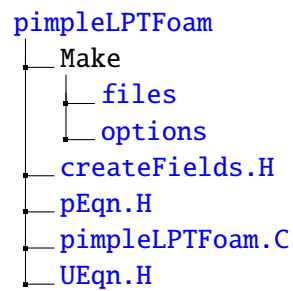


Figura D.1: Árbol de archivos utilizados para las simulaciones del modelo de saltación implementado en OpenFOAM.

Fuente: Elaboración propia.

D.2. Simulaciones

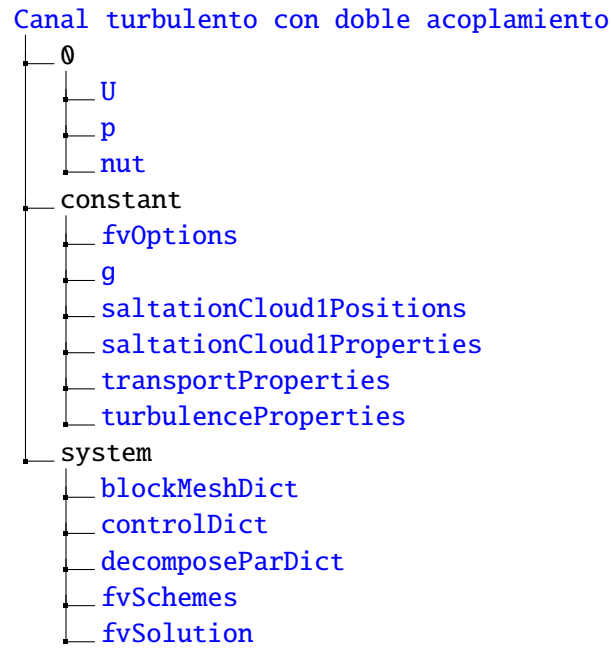
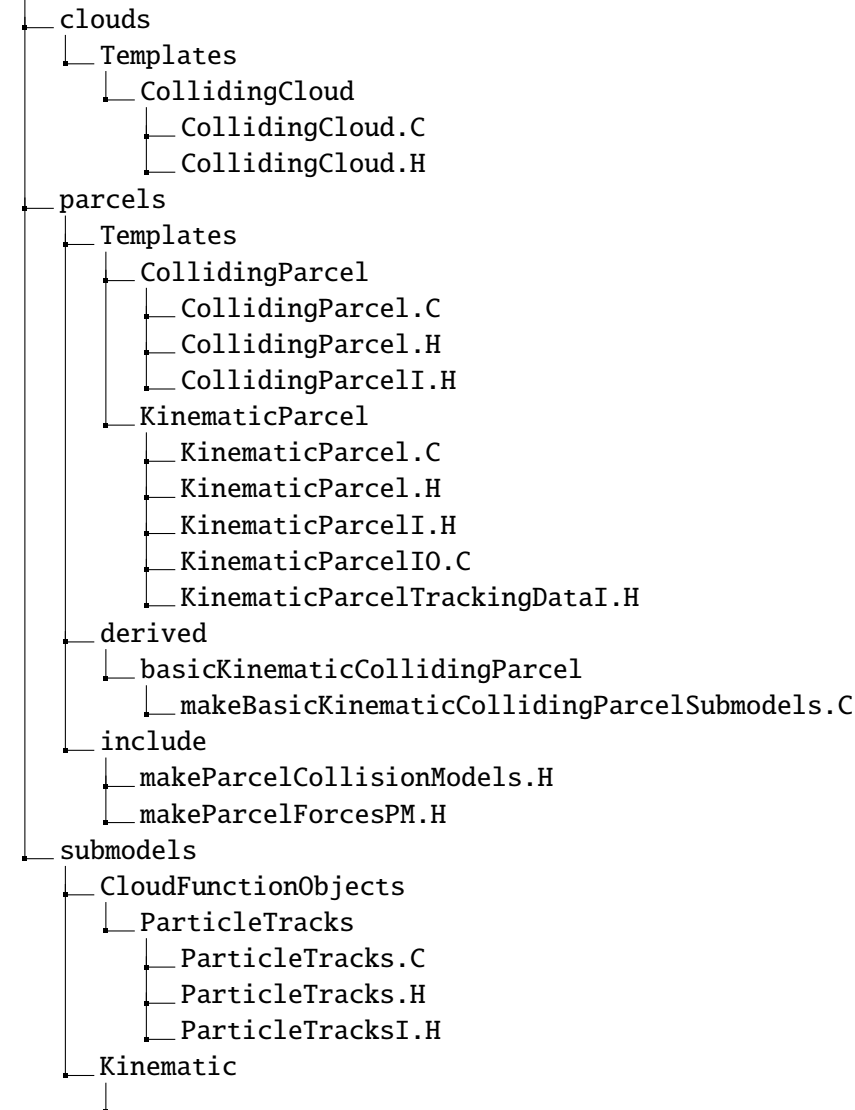


Figura D.2: Árbol de archivos utilizados para la implementación del *solucionador* *pimpleLPTFoam* en OpenFOAM.

Fuente: Elaboración propia.

D.3. Código de partículas

Código de partículas



Continúa...

Figura D.3: Árbol de archivos de la librería intermediate de OpenFOAM que han sido modificados para la implementación del modelo de saltación. Solo se muestran los archivos creados y/o modificados.

Fuente: Elaboración propia.

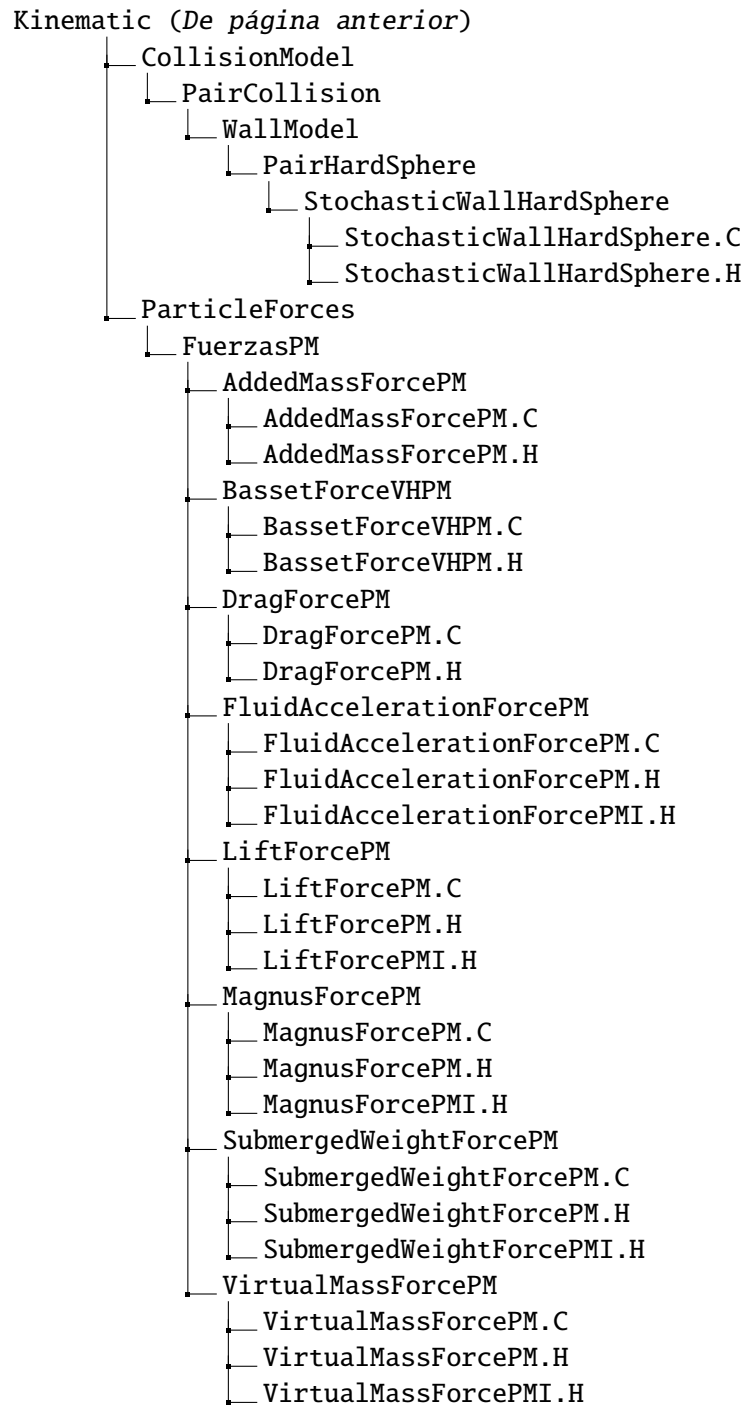


Figura D.3 (cont.): Árbol de archivos de la librería intermediate de OpenFOAM que han sido modificados para la implementación del modelo de saltación. Solo se muestran los archivos creados y/o modificados.

Fuente: Elaboración propia.

D.4. Código fuente

D.4.1 Archivos solucionador *pimpleLPTFoam*

files.

```
1 pimpleLPTFoam.C
2
3 EXE = $(FOAM_USER_APPBIN)/pimpleLPTFoam
```

options.

```
1 EXE_INC = \
2     -I../src/intermediate/lnInclude \
3     -I$(LIB_SRC)/TurbulenceModels/turbulenceModels/lnInclude \
4     -I$(LIB_SRC)/TurbulenceModels/incompressible/lnInclude \
5     -I$(LIB_SRC)/transportModels \
6     -I$(LIB_SRC)/transportModels/incompressible/singlePhaseTransportModel \
7     -I$(LIB_SRC)/finiteVolume/lnInclude \
8     -I$(LIB_SRC)/meshTools/lnInclude \
9     -I$(LIB_SRC)/sampling/lnInclude \
10    -I$(LIB_SRC)/lagrangian/basic/lnInclude \
11    -I$(LIB_SRC)/regionModels/surfaceFilmModels/lnInclude \
12    -I$(LIB_SRC)/regionModels/regionModel/lnInclude
13
14 EXE_LIBS = \
15     -L$(FOAM_USER_LIBBIN) \
16     -lCustomLagrangianIntermediate \
17     -llagrangian \
18     -lturbulenceModels \
19     -lincompressibleTurbulenceModels \
20     -lincompressibleTransportModels \
21     -lfiniteVolume \
22     -lmeshTools \
23     -lfvOptions \
24     -lsampling
```

createFields.H

```
1 Info<< "Reading field p\n" << endl;
2 volScalarField p
3 (
4     IOobject
5     (
6         "p",
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
32 label pRefCell = 0;
33 scalar pRefValue = 0.0;
34 setRefCell(p, pimple.dict(), pRefCell, pRefValue);
35 mesh.setFluxRequired(p.name());
36
37
38 singlePhaseTransportModel laminarTransport(U, phi);
39
40 autoPtr<incompressible::turbulenceModel> turbulence
41 (
42     incompressible::turbulenceModel::New(U, phi, laminarTransport)
43 );
44
45 #include "createMRF.H"
46
47 Info<< "Reading transportProperties\n" << endl;
48 IOdictionary transportProperties
49 (
50     IOobject
51     (
52         "transportProperties",
53         runTime.constant(),
54         mesh,
```

```
55     IOobject::MUST_READ_IF_MODIFIED,
56     IOobject::NO_WRITE
57 )
58 );
59
60 dimensionedScalar rhoInfValue
61 (
62     transportProperties.lookup("rhoInf")
63 );
64
65 dimensionedScalar invrhoInf("invrhoInf",(1.0/rhoInfValue));
66
67 volScalarField rhoInf
68 (
69     IOobject
70     (
71         "rho",
72         runTime.timeName(),
73         mesh,
74         IOobject::NO_READ,
75         IOobject::AUTO_WRITE
76     ),
77     mesh,
78     rhoInfValue
79 );
80
81 volScalarField mu
82 (
83     IOobject
84     (
85         "mu",
86         runTime.timeName(),
87         mesh,
88         IOobject::NO_READ,
89         IOobject::AUTO_WRITE
90     ),
91     laminarTransport.nu()*rhoInfValue
92 );
93
94 Info<< "Constructing kinematicCloud " << "saltationCloud1" << endl;
95 basicKinematicCollidingCloud saltationCloud1
96 (
97     "saltationCloud1",
98     rhoInf,
99     U,
100    mu,
```

```
101     g
102 );

pEqn.H

1 volScalarField rAU(1.0 / UEqn.A());
2 volVectorField HbyA(constrainHbyA(rAU *UEqn.H(), U, p));
3
4 surfaceScalarField phiHbyA(
5     "phiHbyA",
6     fvc::flux(HbyA) + fvc::interpolate(rAU) * fvc::ddtCorr(U, phi));
7
8 MRF.makeRelative(phiHbyA);
9
10 adjustPhi(phiHbyA, U, p);
11
12 tmp<volScalarField> rAtU(rAU);
13
14 if (pimple.consistent())
15 {
16     rAtU = 1.0 / max(1.0 / rAU - UEqn.H1(), 0.1 / rAU);
17     phiHbyA +=
18         fvc::interpolate(rAtU() - rAU) * fvc::snGrad(p) * mesh.magSf();
19     HbyA -= (rAU - rAtU()) * fvc::grad(p);
20 }
21
22 if (pimple.nCorrPISO() <= 1)
23 {
24     tUEqn.clear();
25 }
26
27 // Update the pressure BCs to ensure flux consistency
28 constrainPressure(p, U, phiHbyA, rAtU(), MRF);
29
30 // Non-orthogonal pressure corrector loop
31 while (pimple.correctNonOrthogonal())
32 {
33     // Pressure corrector
34     fvScalarMatrix pEqn(
35         fvm::laplacian(rAtU(), p) == fvc::div(phiHbyA));
36
37     pEqn.setReference(pRefCell, pRefValue);
38
39     pEqn.solve(mesh.solver(p.select(pimple.finalInnerIter())));
40
41     if (pimple.finalNonOrthogonalIter())
```

```
42     {
43         phi = phiHbyA - pEqn.flux();
44     }
45 }
46
47 #include "continuityErrs.H"
48
49 // Explicitly relax pressure for momentum corrector
50 p.relax();
51
52 U = HbyA - rAtU() * fvc::grad(p);
53 U.correctBoundaryConditions();
54 fvOptions.correct(U);
```

pimpleLPTFoam.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) 2011–2016 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  Application
25      pimpleLPTFoam
26
27  Description
28      Large time-step transient solver for incompressible, turbulent flow, using
29      the PIMPLE (merged PISO–SIMPLE) algorithm, with kinematicCollidingCloud
30      attached
```

```
31
32     Sub-models include :
33     - turbulence modelling , i.e. laminar , RAS or LES
34     - run-time selectable MRF and finite volume options , e.g. explicit porosity
35
36  \*-----*/
37
38  #include "fvCFD.H"
39  #include "singlePhaseTransportModel.H"
40  #include "turbulentTransportModel.H"
41  #include "pimpleControl.H"
42  #include "fvOptions.H"
43  #include "basicKinematicCollidingCloud.H"
44
45  // * * * * *
46
47  int main(int argc , char *argv[])
48  {
49  #include "setRootCase.H"
50  #include "createTime.H"
51  #include "createMesh.H"
52  #include "readGravitationalAcceleration.H"
53  #include "createControl.H"
54  #include "createTimeControls.H"
55  #include "createFields.H"
56  #include "createFvOptions.H"
57  #include "initContinuityErrs.H"
58
59      turbulence->validate();
60
61      // * * * * *
62
63      Info << "\nStarting time loop\n"
64            << endl;
65
66      // Time spent solving the carrier phase
67      scalar tCarrier = 0.0;
68      // Time spent solving the dispersed phase
69      scalar tDispersed = 0.0;
70      // Auxiliar
71      scalar tCPU = 0.0;
72      while (runTime.run())
73      {
74  #include "readTimeControls.H"
75  #include "CourantNo.H"
76  #include "setDeltaT.H"
```

```
77
78     runTime++;
79
80     Info << "Time = " << runTime.timeName() << nl << endl;
81
82     tCPU = runTime.elapsedCpuTime();
83     // --- Pressure-velocity PIMPLE corrector loop
84     while (pimple.loop())
85     {
86     #include "UEqn.H"
87
88         // --- Pressure corrector loop
89         while (pimple.correct())
90         {
91     #include "pEqn.H"
92         }
93
94         if (pimple.turbCorr())
95         {
96             laminarTransport.correct();
97             turbulence->correct();
98         }
99     }
100     tCarrier += runTime.elapsedCpuTime() - tCPU;
101
102     tCPU = runTime.elapsedCpuTime();
103     saltationCloud1.evolve();
104     tDispersed += runTime.elapsedCpuTime() - tCPU;
105
106     runTime.write();
107
108     Info << "tCarrier = " << tCarrier << " s"
109         << " tDispersed = " << tDispersed << " s"
110         << nl << endl;
111
112     Info << "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
113         << " ClockTime = " << runTime.elapsedClockTime() << " s"
114         << nl << endl;
115 }
116
117 Info << "End\n"
118     << endl;
119
120 return 0;
121 }
122
```

```
123 // ***** //
```

UEqn.H

```
1 // Solve the Momentum equation
2
3 MRF.correctBoundaryVelocity(U);
4
5 tmp<fvVectorMatrix> tUEqn(
6     fvm::ddt(U) + fvm::div(phi, U) + MRF.DDt(U) + turbulence->divDevReff(U) ==
7     fvOptions(U)
8     //se agregan las fuerzas de la particula sobre el fluido
9     + invrhoInf * saltationCloud1.SU(U));
10
11 fvVectorMatrix &UEqn = tUEqn.ref();
12
13 UEqn.relax();
14
15 fvOptions.constrain(UEqn);
16
17 if (pimple.momentumPredictor())
18 {
19     solve(UEqn == -fvc::grad(p));
20
21     fvOptions.correct(U);
22 }
```

D.4.2 Archivos de simulación

nut.

```
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: 5.0-20180524 |
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        nut;
15 }
16 // ***** //
```

```
17
18 dimensions      [0 2 -1 0 0 0 0];
19
20 internalField    uniform 0;
21
22 boundaryField
23 {
24     bottomWall
25     {
26         type      zeroGradient;
27     }
28     topWall
29     {
30         type      zeroGradient;
31     }
32     sides1_half0
33     {
34         type      cyclic;
35     }
36     sides1_half1
37     {
38         type      cyclic;
39     }
40     sides2_half0
41     {
42         type      cyclic;
43     }
44     sides2_half1
45     {
46         type      cyclic;
47     }
48     inout1_half0
49     {
50         type      cyclic;
51     }
52     inout1_half1
53     {
54         type      cyclic;
55     }
56     inout2_half0
57     {
58         type      cyclic;
59     }
60     inout2_half1
61     {
62         type      cyclic;
```

```

63     }
64 }
65
66
67 // *****

```

p.

```

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: 5.0-20180524 |
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;
15 }
16 // *****
17
18 dimensions      [0 2 -2 0 0 0 0];
19
20 internalField    uniform 0;
21
22 boundaryField
23 {
24     bottomWall
25     {
26         type      zeroGradient;
27     }
28     topWall
29     {
30         type      zeroGradient;
31     }
32     sides1_half0
33     {
34         type      cyclic;
35     }
36     sides1_half1
37     {
38         type      cyclic;

```

```
39     }
40     sides2_half0
41     {
42         type            cyclic;
43     }
44     sides2_half1
45     {
46         type            cyclic;
47     }
48     inout1_half0
49     {
50         type            cyclic;
51     }
52     inout1_half1
53     {
54         type            cyclic;
55     }
56     inout2_half0
57     {
58         type            cyclic;
59     }
60     inout2_half1
61     {
62         type            cyclic;
63     }
64 }
65
66
67 // ***** //
```

U.

```
1  /*-----*-- C++ -*-----*\
2  | ===== |
3  |  \ \      /  F ield      | OpenFOAM: The Open Source CFD Toolbox |
4  |  \ \      /  O peration   | Version:  5.0-20180524                |
5  |   \ \      /  A nd         | Web:      www.OpenFOAM.org           |
6  |    \ \ /    M anipulation  | |
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 (0.44532 0 0);
21
22 boundaryField
23 {
24     bottomWall
25     {
26         type      fixedValue;
27         value      uniform (0 0 0);
28     }
29     topWall
30     {
31         type      fixedValue;
32         value      uniform (0 0 0);
33     }
34     sides1_half0
35     {
36         type      cyclic;
37     }
38     sides1_half1
39     {
40         type      cyclic;
41     }
42     sides2_half0
43     {
44         type      cyclic;
45     }
46     sides2_half1
47     {
48         type      cyclic;
49     }
50     inout1_half0
51     {
52         type      cyclic;
53     }
54     inout1_half1
55     {
56         type      cyclic;
57     }
58     inout2_half0
59     {
60         type      cyclic;
```

```

61     }
62     inout2_half1
63     {
64         type            cyclic;
65     }
66 }
67
68
69 // *****

```

fvOptions.

```

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: 5.0-20180524 |
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       fvOptions;
15 }
16 // *****
17
18 momentumSource
19 {
20     type          meanVelocityForce;
21     active        on;           // on/off switch
22     selectionMode all;          // cellSet / points / cellZone
23     fields        (U);         // Apply force to all cells
24     Ubar          (0.44532 0 0); // Desired mean velocity 0.44532
25     // relaxation  0.2;         // Optional relaxation factor
26 }
27
28 // *****

```

g.

```

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: 5.0-20180524 |

```

```

5 |   \ \   /   A nd           | Web:      www.OpenFOAM.org           |
6 |   \ \ /   M anipulation   |           |
7 /*-----*/
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        uniformDimensionedVectorField;
13     location     "constant";
14     object       g;
15 }
16 // *****
17
18 dimensions      [0 1 -2 0 0 0 0];
19 value           (0 -9.81 0 );
20
21
22 // *****

```

saltationCloud1Positions.

```

1 /*-----* C++ -*-----*\
2 | ===== |
3 | \ \   /   F ield           | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \   /   O peration       | Version:  5                           |
5 |   \ \   /   A nd           | Web:      www.OpenFOAM.org           |
6 |   \ \ /   M anipulation   |           |
7 /*-----*/
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        vectorField;
13     object       saltationCloud1Positions;
14 }
15 // *****
16
17 (
18     // Lista de posiciones para las particulas en la forma: (x y z);
19     (0.00036 0.00690 -0.05412)
20     (0.00036 0.01023 -0.05274)
21     (0.00036 0.01109 -0.05136)
22     (0.00036 0.00901 -0.04998)
23     // ...
24 )
25 // *****

```

saltationCloud1Properties.

```

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: 5.0-20180108 |
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;
11     format        ascii;
12     class         dictionary;
13     location      "constant";
14     object        saltationCloud1Properties;
15 }
16 // *****
17
18 solution
19 {
20     active        true;
21     coupled       true;
22     transient     yes;
23     cellValueSourceCorrection on;
24     maxCo         0.3;
25     interpolationSchemes
26     {
27         rho       cell;
28         U         cellPoint;
29         mu        cell;
30         DUcdt     cellPoint;
31         dUcdt     cellPoint;
32         curlUcdt  cellPoint;
33     }
34     integrationSchemes
35     {
36         U         Euler;
37     }
38     sourceTerms
39     {
40         schemes
41         {
42             U         semiImplicit 1;
43         }
44     }

```

```
45 }
46
47 constantProperties
48 {
49     rho0          2650;
50     youngsModulus 130000;
51     poissonsRatio 0.35;
52     NB            0;
53     dtB           0.0001;
54 }
55
56 subModels
57 {
58     particleForces
59     {
60         submergedWeightForcePM
61         {
62             theta          0;
63             printInfo      no;
64         }
65         dragForcePM
66         {
67             printInfo      no;
68         }
69         liftForcePM
70         {
71             Cl             0.11;
72             printInfo      no;
73         }
74         addedMassForcePM
75         {
76             Cm             0.5;
77             printInfo      no;
78         }
79         virtualMassForcePM
80         {
81             Cm             0.5;
82             printInfo      no;
83         }
84         fluidAccelerationForcePM
85         {
86             printInfo      no;
87         }
88         magnusForcePM
89         {
90             printInfo      no;
```

```
91     }
92     bassetForceVHPM
93     {
94         tolerance      1e5;
95         printInfo      no;
96     }
97 }
98 injectionModels
99 {
100     model1
101     {
102         massFlowRate    0;
103         type             manualInjection;
104         patchName       bottomWall;
105         duration        0.7;
106         parcelsPerSecond 15000;
107         massTotal       0;
108         SOI             0.000;
109         parcelBasisType fixed;
110         flowRateProfile constant 1;
111         nParticle       1;
112         positionsFile    saltationCloud1Positions;
113         U0              ( 0 0 0 );
114         sizeDistribution
115         {
116             type         fixedValue;
117             fixedValueDistribution
118             {
119                 value     0.00069;
120             }
121         }
122     }
123 }
124 dispersionModel          none;
125 patchInteractionModel    none;
126 surfaceFilmModel        none;
127 stochasticCollisionModel none;
128 collisionModel           pairCollision;
129 pairCollisionCoeffs
130 {
131     maxInteractionDistance 0.00069;
132     writeReferredParticleCloud no;
133     pairModel              pairSpringSliderDashpot;
134     pairSpringSliderDashpotCoeffs
135     {
136         alpha              0.3;
```

```
137         b            1.5;
138         mu            0.1;
139         cohesionEnergyDensity 0;
140         collisionResolutionSteps 12;
141         useEquivalentSize no;
142     }
143     wallModel          stochasticWallHardSphere;
144     stochasticWallHardSphereCoeffs
145     {
146         e              0.7;
147         f              0.2;
148         angMax          20;
149         angMin          0;
150         printInfo       no;
151     }
152 }
153 }
154
155 cloudFunctions
156 {
157     particleTracks1
158     {
159         type            particleTracks;
160         trackInterval   1;
161         maxSamples       10000000;
162         resetOnWrite     no;
163         maxTracked       1;
164     }
165     particleCollector1
166     {
167         type            particleCollector;
168         mode            polygon;
169         polygons
170         (
171             // Poligono en 1/4 del dominio
172             (
173                 (0.054507 0 -0.054507)
174                 (0.054507 0.0694 -0.054507)
175                 (0.054507 0.0694 0.054507)
176                 (0.054507 0 0.054507)
177             )
178             // Poligono en 2/4 del dominio
179             (
180                 (0.109014 0 -0.054507)
181                 (0.109014 0.0694 -0.054507)
182                 (0.109014 0.0694 0.054507)
```

```

183         (0.109014 0 0.054507)
184     )
185     // Poligono en 3/4 del dominio
186     (
187         (0.163521 0 -0.054507)
188         (0.163521 0.0694 -0.054507)
189         (0.163521 0.0694 0.054507)
190         (0.163521 0 0.054507)
191     )
192 );
193 normal          (-1 0 0);
194 negateParcelsOppositeNormal no;
195 removeCollected no;
196 surfaceFormat   vtk;
197 resetOnWrite     yes; // si no tira error, arreglar en el postproceso
198 log              yes;
199 }
200 }
201
202
203 // *****

```

transportProperties.

```

1 /*-----*- C++ -*-----*/
2 | ===== |
3 | \ \      / F ield      | OpenFOAM: The Open Source CFD Toolbox |
4 | \ \      / O peration  | Version: 2.2.2 |
5 | \ \      / A nd        | Web: www.OpenFOAM.org |
6 | \ \ /      M anipulation | |
7 /*-----*-*/
8 FoamFile
9 {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "constant";
14     object         transportProperties;
15 }
16 // *****
17
18 Ubar          Ubar [ 0 1 -1 0 0 0 0 ] (0.44532 0 0 );
19
20 transportModel Newtonian;
21
22 nu            nu [ 0 2 -1 0 0 0 0 ] 1.004e-06;

```

```

23
24 rhoInf          rhoInf [ 1 -3 0 0 0 0 0 ] 1000.0;
25
26 // *****

turbulenceProperties.

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: 2.2.2 |
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         turbulenceProperties;
15 }
16 // *****
17
18 simulationType LES;
19
20 LES
21 {
22     LESModel      WALE;
23
24     turbulence     on;
25
26     printCoeffs    on;
27
28     delta          cubeRootVol;
29
30     cubeRootVolCoeffs
31     {
32         deltaCoeff      1;
33     }
34 }
35 // *****

```

blockMeshDict.

```

1 /*-----*- C++ -*-----*\
2 | =====|

```



```
49     type          wall;
50     faces          ((0 1 7 6));
51 }
52 topWall
53 {
54     type          wall;
55     faces          ((4 10 11 5));
56 }
57
58 sides1_half0
59 {
60     type          cyclic;
61     neighbourPatch sides1_half1;
62     faces          ((0 2 3 1));
63 }
64 sides1_half1
65 {
66     type          cyclic;
67     neighbourPatch sides1_half0;
68     faces          ((6 7 9 8));
69 }
70
71 sides2_half0
72 {
73     type          cyclic;
74     neighbourPatch sides2_half1;
75     faces          ((2 4 5 3));
76 }
77 sides2_half1
78 {
79     type          cyclic;
80     neighbourPatch sides2_half0;
81     faces          ((8 9 11 10));
82 }
83
84 inout1_half0
85 {
86     type          cyclic;
87     neighbourPatch inout1_half1;
88     faces          ((1 3 9 7));
89 }
90 inout1_half1
91 {
92     type          cyclic;
93     neighbourPatch inout1_half0;
94     faces          ((0 6 8 2));
```

```

95     }
96
97     inout2_half0
98     {
99         type            cyclic;
100        neighbourPatch  inout2_half1;
101        faces           ((3 5 11 9));
102    }
103    inout2_half1
104    {
105        type            cyclic;
106        neighbourPatch  inout2_half0;
107        faces           ((2 8 10 4));
108    }
109 );
110
111 mergePatchPairs
112 (
113 );
114
115 // *****

```

controlDict.

```

1  /*-----*-- C++ --*-----*\
2  | =====|
3  |  \ \    /  F ield      | OpenFOAM: The Open Source CFD Toolbox |
4  |  \ \    /  O peration  | Version:  2.2.2                      |
5  |   \ \  /  A nd         | Web:      www.OpenFOAM.org           |
6  |    \ \ /  M anipulation |                                     |
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     pimpleLPTFoam;
19
20 startTime       0;
21
22 stopAt          endTime;

```

```
23
24 endTime          10;
25
26 deltaT           0.0005;
27
28 writeControl      adjustableRunTime;
29
30 writeInterval     0.05;
31
32 purgeWrite        60;
33
34 writeFormat       ascii;
35
36 writePrecision    6;
37
38 writeCompression  off;
39
40 timeFormat        general;
41
42 timePrecision     6;
43
44 runTimeModifiable true;
45
46 adjustTimeStep    true;
47
48 maxCo 0.3;
49
50 functions
51 {
52     #includeFunc residuals
53
54     fieldAverager1
55     {
56         type          fieldAverage;
57         functionObjectLibs ( "libfieldFunctionObjects.so" );
58         enabled        true;
59         writeControl    writeTime;
60         restartOnOutput false;
61         restarOnRestart true;
62         periodicRestart true;
63         restartPeriod   5;
64
65         fields
66         (
67             U
68             {
```

```
69         mean      on;
70         prime2Mean off;
71         base       time;
72     }
73 );
74 }
75
76 yPlus1
77 {
78     type          yPlus;
79     functionObjectLibs ("libfieldFunctionObjects.so");
80     writeControl    writeTime;
81 }
82 }
83
84 // ***** //
```

decomposeParDict.

```
1 /*-----* C++ -*-----*\
2 | ===== |
3 | \\      / F ield      | OpenFOAM: The Open Source CFD Toolbox |
4 | \\      / O peration   | Version: 5.0-20180524                |
5 |  \\     / A nd         | Web:      www.OpenFOAM.org            |
6 |   \\//   M anipulation  |                                     |
7 /*-----*/
8 FoamFile
9 {
10     version      2;
11     format        ascii;
12     class         dictionary;
13     note          "mesh decomposition control dictionary";
14     object         decomposeParDict;
15 }
16
17 numberOfSubdomains 40;
18
19 method            simple;
20
21 simpleCoeffs
22 {
23     n              ( 5 2 4 );
24     delta          0.001;
25 }
26
27 scotchCoeffs
```

```

28 {
29     processorWeights ( 1 1 1 1 );
30     writeGraph      true;
31     strategy        "b";
32 }
33
34
35 // *****

```

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: 2.2.2 |
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        backward;
21 }
22
23 gradSchemes
24 {
25     default        Gauss linear;
26     grad(p)        Gauss linear;
27     grad(U)        Gauss linear;
28 }
29
30 divSchemes
31 {
32     default        none;
33     div(phi,U)     Gauss linear;
34     div(phi,k)     Gauss limitedLinear 0.5;
35     div(phi,B)     Gauss limitedLinear 0.5;

```

```
36     div(B)          Gauss linear;
37     div(phi,nuTilda) Gauss limitedLinear 1;
38     div((nuEff*dev2(T(grad(U)))) Gauss linear;
39 }
40
41 laplacianSchemes
42 {
43     default          none;
44     laplacian(nuEff,U) Gauss linear corrected;
45     laplacian(rAUf,p) Gauss linear corrected;
46     laplacian((1|A(U)),p) Gauss linear corrected;
47     laplacian(DkEff,k) Gauss linear corrected;
48     laplacian(DBEff,B) Gauss linear corrected;
49     laplacian(DnuTildaEff,nuTilda) Gauss linear corrected;
50 }
51
52 interpolationSchemes
53 {
54     default          linear;
55     interpolate(U)   linear;
56 }
57
58 snGradSchemes
59 {
60     default          corrected;
61 }
62
63 wallDist
64 {
65     method meshWave;
66 }
67
68
69 // ***** //
```

fvSolution.

```
1 /*-----*-- C++ --*-----*/
2 | ===== |
3 | \\      /  F ield      | OpenFOAM: The Open Source CFD Toolbox |
4 | \\      /  O peration  | Version: 2.2.2 |
5 |  \\    /  A nd         | Web:      www.OpenFOAM.org |
6 |   \\//   M anipulation | |
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     p
21     {
22         solver      GAMG;
23         tolerance    0;
24         relTol      0.1;
25         smoother     GaussSeidel;
26     }
27
28     pFinal
29     {
30         $p;
31         smoother     DICGaussSeidel;
32         tolerance     1e-06;
33         relTol        0;
34     }
35
36     "(U|k|nuTilda)"
37     {
38         solver      smoothSolver;
39         smoother     symGaussSeidel;
40         tolerance    1e-05;
41         relTol       0.1;
42     }
43
44     "(U|k|nuTilda) Final"
45     {
46         $U;
47         tolerance    1e-05;
48         relTol        0;
49     }
50 }
51
52 PIMPLE
53 {
54     nOuterCorrectors 1;
55     nCorrectors      2;
```

```
56     nNonOrthogonalCorrectors 0;
57     pRefCell          1;
58     pRefValue          0;
59 }
60
61
62 // ***** //
```

