



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA CIVIL

MODELACIÓN NUMÉRICA TRIDIMENSIONAL DE LOS PROCESOS
DE SOCAVACIÓN EN TORNO A PILAS CIRCULARES

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA
INGENIERÍA, MENCIÓN RECURSOS Y MEDIO AMBIENTE HÍDRICO

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL

JOAQUÍN ANTONIO MORENO UMAÑA

PROFESOR GUÍA:
ALDO TAMBURRINO TAVANTZIS

MIEMBROS DE LA COMISIÓN:
YARKO NIÑO CAMPOS
MATÍAS QUEZADA LABRA
LUIS ZAMORANO RIQUELME

SANTIAGO DE CHILE
2021

RESUMEN DE LA TESIS PARA OPTAR AL TÍTULO DE INGENIERO CIVIL Y GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA MENCIÓN RECURSOS Y MEDIO AMBIENTE HÍDRICO

POR: JOAQUÍN ANTONIO MORENO UMAÑA

FECHA: 2021

PROF. GUÍA: ALDO TAMBURRINO TAVANTZIS

MODELACIÓN NUMÉRICA TRIDIMENSIONAL DE LOS PROCESOS DE SOCAVACIÓN EN TORNO A PILAS CIRCULARES

La socavación local que se produce alrededor de pilas circulares es una de las principales causas de fallas en puentes. Esto es producto de la alteración de la hidrodinámica natural del cauce, pues el obstáculo genera una modificación en el esfuerzo de corte en el fondo, induciendo erosión local. Así, es necesario un estudio de los efectos de las crecidas en la socavación en torno a pilas, para poder fundarlas de forma adecuada y prevenir fallas.

En el presente estudio, se realizan modelaciones numéricas tridimensionales de fenómenos erosivos en torno a pilas circulares frente a diferentes tipos de flujo, siendo estos: permanente e hidrogramas de crecida, teniendo como condición de transporte de sedimentos en régimen de aguas claras.

Las modelaciones se realizan mediante el software OpenFOAM, utilizando el paquete SedFoam, el cual permite la representación de estos fenómenos a través de la resolución de las ecuaciones RANS para fluido y sedimento, utilizando el modelo $k - \epsilon$ o $k - \omega$ para cerrar la turbulencia y la incorporación de la teoría cinética de flujos granulares o la reología de flujos granulares densos para los esfuerzos intergranulares, se evalúa su funcionalidad en este tipo de fenómenos, generando un contraste con modelaciones experimentales y de REEF3D, un modelo numérico que resuelve las ecuaciones RANS para una fase, utilizando una expresión empírica para los esfuerzos de corte sobre el lecho y la ecuación de Exner para resolver el nivel del lecho.

Con los modelos anteriores, se determina la socavación en el tiempo y campos de velocidad, para los casos en estudio, comparándose con los resultados disponibles en la literatura. Además, se analizan las ventajas y desventajas de modelar considerando una o dos fases en el flujo, teniendo en cuenta la representación de fenómenos físicos y costo computacional.

Finalmente se concluye que, dado el costo computacional y las inestabilidades del modelo en torno a la interfaz observadas, actualmente no es factible utilizar en la industria ingenieril el modelo de dos fases SedFoam para estudiar socavación en el tiempo. Desde este punto de vista, REEF3D permite obtener resultados en una escala de tiempo mucho menor, que se adecua de mejor manera a los tiempos requeridos en ingeniería.

“Lo importante es nunca dejar de hacerse preguntas”

-Albert Einstein

Agradecimientos

En primer lugar, agradecer a mis padres, que dieron la posibilidad de estudiar en un ambiente propicio y sin preocupaciones. A mi familia por el cariño y la preocupación desde siempre.

A todo mi círculo de apoyo, no hace falta nombrarlos uno por uno, pues el cariño y las experiencias compartidas lo dicen todo.

A mis profesores y guías, que me apoyaron en este proceso con sus ideas y consejos. En particular a Aldo Tamburrino, quien despertó el gusto por la hidráulica en mí, me acogió en el laboratorio hace varios años, llevándome a vivir muy bonitas experiencias y me abrió el camino para aprender sobre OpenFoam con Aldo Muñoz, al cual agradezco también por la paciencia y disposición para siempre responder la infinidad de dudas que tuve en el camino.

A Roberto Bravo y Álvaro Torres que durante la pandemia me apoyaron manteniéndome conectado a los computadores de la universidad, cada vez que fallaban los programas o se cortaba la luz. A la Jacque por siempre apoyar y hacer las gestiones cuando fue necesario.

Este trabajo ha sido financiado por la Agencia Nacional de Investigación y Desarrollo / Programa Becas / BECA MAGISTER NACIONAL 2020 – 22200637.

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

Tabla de contenido

1.Introducción.....	1
2.Objetivos.....	5
3.Modelos Conceptuales.....	6
3.1.Modelo de una fase.....	6
3.2.Modelo de dos fases	11
4.Modelos Numéricos.....	20
4.1.REEF3D	20
4.2.OpenFOAM: SedFoam.....	22
5.Casos en estudio	26
5.1.Caso Permanente	26
5.1.1.Numérico	26
5.1.2.REEF3D	26
5.1.3.OpenFOAM.....	27
5.2.Casos Frente a Hidrogramas de Crecida.....	27
5.2.1.Experimental.....	27
5.2.2.REEF3D	28
5.2.3.OpenFOAM.....	29
6.Resultados.....	30
6.1.Caso Permanente	30
6.2.Casos Frente a Hidrogramas de Crecida.....	34
7.Discusión	37
8.Conclusiones.....	40
9.Bibliografía.....	41
Anexos	47
Anexo A: Ficheros del modelo REEF3D	48
Control.txt.....	48
Ctrl.txt.....	49
Anexo B: Código generación archivo de malla.....	50
Anexo C.....	57
Carpeta 0.....	58

Carpeta constant	67
Carpeta system.....	71

Índice de Figuras

Figura 1.1 Falla puente Cancura, río Rahue, Chile (2018). Fuente: www.uchile.cl/noticias/	1
Figura 1.2 Estructuras características del flujo. Fuente: Baykal (2017).....	2
Figura 4.1 Malla generada con DIVEMesh para REEF3D.	20
Figura 4.2 Esquema de la geometría de la malla utilizada con SedFoam. Fuente: Nagel et al. (2018)..	23
Figura 4.3 Vista en planta de malla generada para SedFoam.....	23
Figura 4.4 Corte Longitudinal de malla generada para SedFoam	24
Figura 5.1 Instalación experimental utilizada por Link et al. (2017).	27
Figura 5.2 Hidrograma 1. Fuente: Link et al (2017)..	28
Figura 6.1 Evolución temporal de la socavación máxima en el tiempo, aguas arriba del centro de la pila para el caso de flujo permanente $Q = 0.02 \text{ m}^3 / \text{s}$ estudiado por Baykal et al. (2015)	31
Figura 6.2 Evolución temporal de la socavación máxima en el tiempo, aguas abajo del centro de la pila para el caso de flujo permanente $Q = 0.02 \text{ m}^3 / \text{s}$ estudiado por Baykal et al. (2015)	31
Figura 6.3 Evolución de la socavación adimensional (S/D) en la región cercana a la pila. .	32
Figura 6.4 Perfil de velocidad longitudinal (m/s) en el centro de la pila modelado con REEF3D. Caso Baykal et al. (2015), flujo permanente $Q = 0.02 \text{ m}^3 / \text{s}$, $D = 0.04 \text{ m}$. Considera desde en x desde 4D aguas arriba del centro de la pila hasta 4D aguas abajo del centro de la pila.	33
Figura 6.5 Perfil de velocidad en planta (m/s) de la superficie libre modelado con REEF3D. Caso Baykal et al. (2015), flujo permanente $Q = 0.02 \text{ m}^3 / \text{s}$, $D = 0.04 \text{ m}$. Considera desde en x desde 2D aguas arriba del centro de la pila hasta 10D aguas abajo del centro de la pila y 8D de extensión en el eje y.....	33
Figura 6.6 Estructuras de flujo en el caso de Baykal et al. (2015), flujo permanente $Q = 0.02 \text{ m}^3 / \text{s}$, $D = 0.04 \text{ m}$ modelado con REEF3D ($t=120 \text{ s}$).....	34
Figura 6.7 Evolución temporal de la socavación máxima en el tiempo. Caso con flujo permanente, hidrograma de crecida propuesto por Link et al. (2017). $D = 0.15 \text{ m}$	35
Figura 6.8 Acercamiento a la evolución temporal de la socavación máxima en el tiempo, a los tiempos en que fue posible modelar con SedFoam. Caso con flujo permanente, hidrograma de crecida propuesto por Link et al. (2017). $D = 0.15 \text{ m}$	35
Figura 6.9 Líneas de corriente alrededor de la pila. Caso hidrograma de crecida propuesto por Link et al. (2017). Simulación con SedFoam. $T=150 \text{ s}$. Los colores de las líneas de corriente dan cuenta de la velocidad (m/s) en el eje x.....	36
Figura 6.10 Líneas de corriente alrededor de la pila. Caso hidrograma de crecida propuesto por Link et al. (2017). Simulación con REEF3D. $T=150 \text{ s}$. Los colores de las líneas de corriente dan cuenta de la velocidad (m/s) en el eje x.....	36

Índice de Tablas

Tabla 6.1 Tiempos de cálculo empleados en cada simulación para cada software.	1
------------------------------------------------------------------------------------	---

1.Introducción

Tanto el flujo, como la socavación en torno a pilas verticales, han sido ampliamente estudiados las últimas décadas, con especial detalle en las pilas de puentes en ríos y en fundaciones de pilas en las costas (Quezada *et al.*, 2018). Esto pues la socavación local que se produce en torno a estas es una de las principales causas de fallas en las estructuras (Ver Figura 1.1).



Figura 1.1 Falla puente Cancura, río Rahue, Chile (2018). Fuente: www.uchile.cl/noticias/

Acorde a lo señalado por Quezada *et al.* (2019), la hidrodinámica en ambientes costeros resulta de la interacción de diferentes forzantes como olas, mareas y vientos, los cuales actúan en diferentes escalas tanto temporales como espaciales, gobernando la circulación. Mientras que, en ríos, esta se debe principalmente a la acción gravitacional que mueve el agua producto del derretimiento de nieve o por lluvia que fluye hacia el canal aluvional. La mezcla entre los ambientes señalados se da en los estuarios, donde la hidrodinámica resulta de una mezcla compleja entre todas estas forzantes.

Esta hidrodinámica se ve afectada al presentarse un obstáculo, pues el esfuerzo de corte de fondo se ve modificado, lo cual induce una erosión local en torno a este. Sumer *et al.* (1997) proponen que las estructuras características de flujo que se manifiestan son el vórtice de herradura (Horseshoe vortex), el cual se presenta en el frente de la pila, y los vórtices de sotavento (Lee-wake vórtices), en la parte trasera. El primero se produce por la rotación hacia el fondo del flujo entrante a la pila, el cual se debe a la disminución de presión, mientras que el segundo se produce debido a que en los lados del cilindro las capa límites generadas que se separan de este, se enrollan hacia el centro. Estas estructuras se exponen en la Figura 1.2.

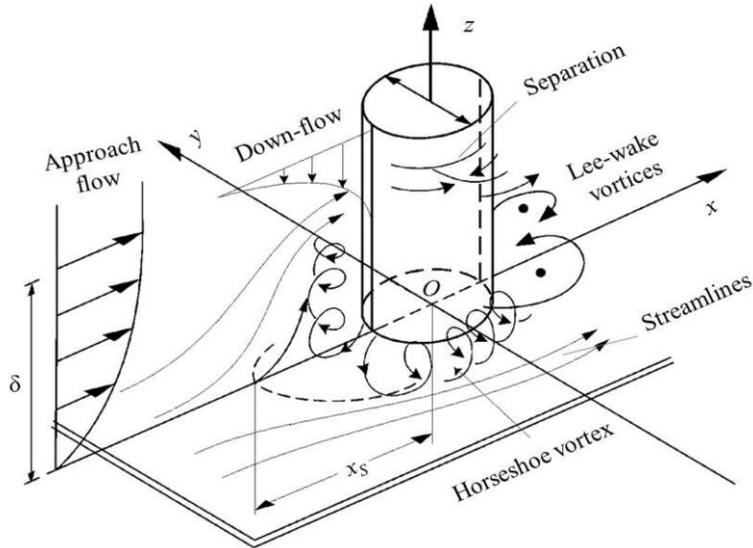


Figura 1.2 Estructuras características del flujo. Fuente: Baykal (2017).

En el estudio de la erosión en torno a estructuras se distinguen dos condiciones, de agua clara y de lecho vivo. La primera es aquella donde la velocidad en la cercanía del fondo (o el esfuerzo de corte sobre el lecho) es menor a la de movimiento incipiente, y por tanto solo existe transporte en la vecindad de la pila, y la segunda donde la velocidad en la cercanía del fondo es mayor a la incipiente, y, por tanto, a diferencia de la primera condición, existe un arrastre de sedimentos debido al flujo de aproximación a la pila.

Dada la complejidad de los fenómenos presentes en la socavación en torno a pilas, tanto para una pila, como para un conjunto de estas, en su mayoría los trabajos reportados a la fecha son experimentales (Link *et al.*, 2018; Yagci *et al.*, 2017; Chen *et al.*, 2017; Wang *et al.*, 2016; Moreno *et al.*, 2016; Lança *et al.*, 2013; Amini *et al.*, 2012; Link, 2006; Ataie *et al.*, 2006; Sheppard *et al.*, 2004; entre otros).

A pesar de esto, los últimos años se han realizado avances en la modelación numérica de estos procesos, mediante la resolución de las ecuaciones de Navier-Stokes promediadas de Reynolds (RANS), en conjunto con un modelo de turbulencia para cerrar el problema o en menor medida con métodos más complejos, como DNS (direct numerical solutions), LES (large-eddy simulations) o DES (detached-eddy simulations). (Baykal *et al.*, 2017; Baykal *et al.*, 2015; Afzal *et al.*, 2015; Mohamed, 2012; Aghaee & Hakimzadeh, 2010; Liu & García, 2008; Roulund *et al.*, 2005)

Existen una serie de herramientas numéricas computacionales para la dinámica de fluidos (CFD) que han sido utilizados para la modelación de socavación en torno a pilas, tales como OpenFOAM (<https://www.openfoam.com>), REEF3D (<https://reef3d.wordpress.com>), Fluidity (<https://fluidityproject.github.io>), Flow-3D (<https://www.flow3d.com>) y SPHysics (<https://dual.sphysics.org>), entre otros.

Flow-3D es un software de simulación de flujos de fluidos con superficie libre en régimen transitorio que permite realizar análisis completos de interacción fluido-estructura para obtener las fuerzas, tensiones y deformaciones resultantes de los cuerpos y ha sido aplicado para estudiar la influencia de diferentes modelos de turbulencia en la socavación de pilas de puentes (*Man et al., 2019*).

Fluidity es un CFD multifásico y ha sido utilizado para estudiar la implementación del acoplamiento de algoritmos de adaptabilidad de malla anisotrópica considerando transporte turbulento de sedimentos suspendidos y de fondo (*Nunez et al., 2018*).

SPHysics se basa en un modelo “smoothed particle hydrodynamics” y está diseñado para estudiar fenómenos de superficie libre donde los métodos eulerianos pueden ser difíciles de aplicar. *Fourtakas & Rogers (2016)* mediante este, llevaron a cabo simulaciones de procesos de socavación, considerando sedimentación y resuspensión.

REED3D está diseñado para utilizar una gran cantidad de procesadores. Posee un comportamiento numérico estable y preciso debido a sus esquemas de discretización temporal y espacial de alto orden. Así, es capaz de resolver las ecuaciones gobernantes en todas las escalas relevantes. Este resuelve las ecuaciones RANS para una fase, utilizando una expresión empírica para los esfuerzos de corte sobre el lecho y para el nivel del lecho la ecuación de Exner en su versión generalizada por *Paola & Voller (2005)*. *Quezada et al (2018)* utilizó este CFD para investigar el efecto de flujos oscilatorios e hidrogramas en pilas circulares y *Ahmad et al. (2015)* estudió la erosión en torno a un arreglo triangular de pilas.

SedFoam es un solver tridimensional de OpenFOAM, desarrollado por *Chauchat et al. (2017)*, capaz de resolver la erosión local en torno a una pila, mediante la resolución de las ecuaciones promediadas turbulentas de Navier-Stokes para dos fases, utilizando el modelo $k - \epsilon$ o $k - \omega$ para cerrar la turbulencia y la incorporación de la teoría cinética de flujos granulares o la reología de flujos granulares densos para los esfuerzos integranulares. *Nagel et al. (2017)* aplicó el modelo para simular socavación frente a flujos permanentes.

Según lo observado por *Melville (2008)* en las modelaciones realizadas para encontrar socavación máxima, en general se opta por utilizar la condición de agua clara en estudios en torno a pilas, lo cual maximiza la socavación local.

Las escalas de tiempo en procesos de socavación para alcanzar el equilibrio dependen del tipo de flujo en estudio. En el caso permanente puede variar de días a semanas acorde a lo sugerido por *Melville & Chiew (1999)* y *Oliveto & Hager (2002)*, mientras que para flujos oscilatorios según *Kobayashi & Oda (1994)* este debiera ser del orden de 1000 veces el período de la onda. Por otra parte, las crecidas u olas asociadas a lluvias o eventos extremos tienen una duración acotada, por lo que, para poder notar el efecto de estas en la erosión en torno a pilas, algunos autores han observado experimentalmente la dependencia de la erosión frente a hidrogramas (*Oliveto and Hager, 2002; Chang et al., 2004; Gjunsburgs et al., 2010; Hager & Unger, 2010; Link et al., 2017*).

En el caso de los flujos oscilatorios, la determinación de la socavación en torno a pilas es escasa y la inclusión de escalas temporales se reduce a sólo algunos trabajos experimentales (*Sumer et al., 1992; Kobayashi & Oda, 1994; Sumer & Fredsoe, 2002*) y numéricos (*Quezada et al., 2018; Quezada et al., 2019*).

El presente estudio apunta a modelar numéricamente la socavación del sedimento en torno a pilas, frente a condiciones permanentes de flujo e hidrogramas de crecida mediante el software OpenFOAM, utilizando el solver SedFoam, se busca evaluar su funcionalidad en este tipo de fenómenos, generar un contraste con modelaciones a realizar en REEF3D, con el fin de analizar las ventajas y desventajas de modelar considerando una o dos fases en el flujo y aumentar el espectro de la modelación numérica en torno a pilas.

2.Objetivos

General:

-El trabajo de investigación tiene como objeto la modelación numérica tridimensional de la socavación que se desarrolla en torno a pilas circulares mediante los softwares OpenFOAM y REEF3D.

Específicos:

-Definir cómo se modelará la socavación usando los softwares mencionados, esto es, definir el modelo de cierre de turbulencia, como se modelarán los esfuerzos intergranulares, entre fluido y sedimento, en el caso de dos fases y las expresiones para modelar el transporte de sedimento.

-Analizar las ventajas y desventajas de utilizar un software que resuelva una o dos fases.

-Modelar problemas relacionados con la socavación en torno a una pila circular haciendo uso de OpenFOAM.

-Modelar problemas relacionados con la socavación en torno a una pila circular haciendo uso de REEF3D.

-Determinar la socavación máxima y campo de velocidades obtenidos en la modelación y compararlos con la información experimental y numérica que se encuentre disponible en la literatura.

3. Modelos Conceptuales

Acorde a *Chauchat et al. (2017)*, la modelación clásica del transporte de sedimentos consiste en dividir el dominio físico en dos subcapas. La superior corresponde a la de fluido y sedimento en suspensión, donde se resuelven las ecuaciones de Navier-Stokes promediadas, se considera una concentración diluida del sedimento, y los sedimentos son tratados como un escalar pasivo con una velocidad de sedimentación, mientras que para la subcapa inferior se resuelve el arrastre de sedimentos de fondo usando una expresión empírica para el flujo de sedimento en conjunto con la ecuación de conservación de masa de sedimento (ecuación de Exner). Estas dos capas interactúan mediante flujos verticales de sedimento, los que entran en suspensión y los que se transportan por el fondo.

En los últimos años ha surgido interés por desarrollar modelos de flujo de dos fases, donde la dinámica se resuelve, tanto para la fase del fluido, como la de las partículas sólidas, siendo esta última considerada como una fase continua dispersa en el fluido. Esto podría permitirnos incluir la mayoría de los fenómenos físicos involucrados en el transporte de sedimento, como las interacciones fluido-partícula, interacciones partícula-partícula y turbulencia (*Sumer & Serife, 2008*).

Desde un punto de vista de modelación, la dificultad recae en resolver un sistema de ecuaciones no lineales acopladas del fluido (RANS, DES, LES, DNS, etc) y del sedimento, mientras que, desde el punto de vista teórico, es complejo incorporar la gran cantidad de procesos físicos que tienen lugar en escalas más pequeñas que las escalas promediadas sobre la turbulencia, que son las usadas para derivar las ecuaciones. La clave en este tipo de modelo son los cierres utilizados para los esfuerzos intergranulares y la turbulencia.

A continuación, se describe el modelo conceptual detrás de los softwares a utilizar, en el caso del modelo de una fase REEF3D y en el caso de dos fases OpenFOAM: SedFoam.

3.1. Modelo de una fase

Para describir la hidrodinámica de un fluido incompresible, isótropo y homogéneo, se utilizan las ecuaciones de Navier-Stokes promediadas de Reynolds, las cuales se basan en la conservación de momentum y continuidad de masa. Estas se exponen a continuación en las ecuaciones 3.1.1 y 3.1.2, donde u_i es el vector de velocidad promediado sobre la turbulencia, x_i es la componente geométrica espacial, ρ es la densidad del fluido, ν es la viscosidad cinemática, p es la presión promediada sobre la turbulencia, g_i es la componente de gravedad y ν_t es la viscosidad cinemática de remolinos.

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (3.1.1)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[(v + v_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] + g_i \quad (3.1.2)$$

Para resolver las ecuaciones anteriores, es necesario un modelo de cierre para la turbulencia. A continuación, se muestran dos modelos utilizados comúnmente, $k - \epsilon$ (Launder & Spalding, 1974) y $k - \omega$ (Wilcox, 1998). En la ecuación 3.1.3 se expone como se define la viscosidad de remolinos para el primer modelo, donde k es la energía cinética turbulenta, ϵ es la tasa de disipación de energía cinética turbulenta y c_μ es una constante empírica igual a 0.09.

$$v_t = c_\mu \frac{k^2}{\epsilon} \quad (3.1.3)$$

Las incógnitas recién planteadas se determinan a partir de las ecuaciones de transporte que se muestran en las ecuaciones 3.1.4, 3.1.5 y 3.1.6, donde σ_k , σ_ϵ , $c_{\epsilon 1}$ y $c_{\epsilon 2}$ son iguales a 1, 1.3, 1.44 y 1.92 respectivamente.

$$\frac{\partial k}{\partial t} + u_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(v + \frac{v_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + P_k - \epsilon \quad (3.1.4)$$

$$\frac{\partial \epsilon}{\partial t} + u_j \frac{\partial \epsilon}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(v + \frac{v_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right] + c_{\epsilon 1} P_k - c_{\epsilon 2} \frac{\epsilon^2}{k} \quad (3.1.5)$$

$$P_k = v_t \frac{\partial u_i}{\partial x_j} \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] \quad (3.1.6)$$

Para el segundo modelo, la viscosidad de remolinos se define acorde a la ecuación 3.1.7, donde ω representa la disipación específica de energía cinética turbulenta.

$$\nu_t = \frac{k}{\omega} \quad (3.1.7)$$

En este caso la disipación específica turbulenta y la energía cinemática turbulenta se encuentran utilizando las ecuaciones de transporte señaladas en las ecuaciones 3.1.8 y 3.1.9, donde c_μ , $c_{1\omega}$, $c_{2\omega}$, σ_k y σ_ω son iguales a $\frac{9}{100}$, $\frac{5}{9}$, $\frac{3}{40}$, 2 y 2, mientras que P_k se define de igual forma que en el modelo anterior.

$$\frac{\partial k}{\partial t} + u_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + P_k - c_\mu k \omega \quad (3.1.8)$$

$$\frac{\partial \omega}{\partial t} + u_j \frac{\partial \omega}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\omega} \right) \frac{\partial \omega}{\partial x_j} \right] + \frac{\omega}{k} c_{1\omega} P_k - c_{2\omega} \omega^2 \quad (3.1.9)$$

En el presente trabajo, se utilizará el modelo $k - \omega$ en un caso con condición de entrada de hidrograma de crecida para ser consistente con la modelación en OpenFoam, y $k - \epsilon$ para mostrar la funcionalidad de este en un caso permanente. Esto último dado que *Quezada et al. (2018)* obtuvo una evolución de la socavación más certera con el modelo de turbulencia $k - \epsilon$. El transporte de sedimento que es posible modelar en la versión actual de REEF3D es de fondo y deslizamientos. Para el transporte de fondo existen diversas ecuaciones semi empíricas, como la de *Meyer-Peter & Müller (1948)*, *Engelund & Fredsoe (1976)* y *Van Rijn (1948)*, las cuales fueron deducidas para flujo permanente. En las ecuaciones 3.1.10 y 3.1.11 se exponen las dos primeras, siendo la primera la utilizada en el presente estudio.

$$q_b^* = 8(\tau^* - \tau_c^*)^{\frac{3}{2}} \quad (3.1.10)$$

$$q_b^* = 18.74(\tau^* - \tau_c^*)(\tau^{*1/2} - 0.7\tau_c^{*1/2}) \quad (3.1.11)$$

Donde q_b^* es el transporte de fondo adimensional, τ^* es el esfuerzo de corte de fondo adimensional y τ_c^* es el esfuerzo de corte crítico adimensional, los cuales se definen como se muestra en las ecuaciones 3.1.12, 3.1.13 y 3.1.14.

$$\tau^* = \frac{\tau}{(\rho_s - \rho)gd} \quad (3.1.12)$$

$$\tau_c^* = \frac{\tau_c}{(\rho_s - \rho)gd} \quad (3.1.13)$$

$$q_b^* = \frac{q_b}{\sqrt{\frac{(\rho_s - \rho)gd^3}{\rho}}} \quad (3.1.14)$$

Donde τ , τ_c , q_b , ρ_s y d son el esfuerzo de corte de fondo, el esfuerzo de corte de fondo crítico, el transporte de fondo, la densidad del sedimento y el diámetro uniforme del sedimento, respectivamente. El esfuerzo de corte crítico se estima a partir de la curva de Shields parametrizada por *Yalin (1972)*, y se corrige por la pendiente según lo propuesta por *Dey (2001)*. Esta última consiste en una ponderación del esfuerzo de corte crítico por el coeficiente r definido en la ecuación 3.1.15, donde ϕ es el ángulo de reposo, θ el ángulo de la pendiente longitudinal y α el ángulo de la pendiente transversal.

$$r = 0.954 \left(1 - \frac{\theta}{\phi}\right)^{0.745} \left(1 - \frac{\alpha}{\phi}\right)^{0.372} \quad (3.1.15)$$

Además, se incluye un coeficiente de relajación (K_r) como el propuesto por *Quezada et al. (2018)*, el cual permite la regulación de la movilidad de las partículas debido a la acción hidrodinámica, este se pondera por el esfuerzo de corte crítico en una zona 2 diámetros aguas arriba y abajo de la pila. Esto implica una disminución en el esfuerzo de corte crítico, lo cual se traduce en un aumento en el transporte de fondo en la zona en la cual se aplica. La zona de aplicación se define en base a una predicción del tamaño de la fosa que se genera en el equilibrio en torno a la pila acorde a *Link et al. (2008)* y *Diab et al. (2010)*.

Para determinar las variaciones en el nivel del lecho, se utiliza la ecuación de Exner, considerando que no existe transporte en suspensión dado que se consideran aguas claras (ecuación 3.1.16).

$$\frac{\partial z_b}{\partial t}(1 - \eta) + \left[\frac{\partial q_{b,x}}{\partial x} + \frac{\partial q_{b,y}}{\partial y} \right] = 0 \quad (3.1.16)$$

Donde z_b , η , $q_{b,x}$ y $q_{b,y}$ corresponden al nivel del lecho, la porosidad y el transporte de fondo en las direcciones x e y respectivamente. Estos últimos se estiman de acuerdo con las ecuaciones 3.1.17, 3.1.18, 3.1.19, 3.1.20 y 3.1.21.

$$q_{b,x} = q_b \frac{\tau_{bx}}{|\tau_b|} \quad (3.1.17)$$

$$q_{b,y} = q_b \frac{\tau_{by}}{|\tau_b|} \quad (3.1.18)$$

$$|\tau_b|^2 = \tau_{bx}^2 + \tau_{by}^2 \quad (3.1.19)$$

$$\tau_{bx} = \frac{1}{2} \rho \sqrt{c_{\mu 2}} \left(\sqrt{u_b^2 + v_b^2 + w_b^2} \right) u_b \quad (3.1.20)$$

$$\tau_{by} = \frac{1}{2} \rho \sqrt{c_{\mu 2}} \left(\sqrt{u_b^2 + v_b^2 + w_b^2} \right) v_b \quad (3.1.21)$$

Donde τ_{bx} y τ_{by} son las componentes x e y del esfuerzo de corte total τ_b , c_{μ} es una constante igual a 0.09 y u_b y v_b son las componentes del vector velocidad en la cercanía del lecho. El modelo también es capaz de reproducir el esfuerzo de corte a partir de la relación logarítmica entre la velocidad de corte y variación de velocidad con la altura, o en base a la viscosidad de remolinos, entre otras descritas por *Saud (2013)*.

Por último, una vez estimadas las variaciones en el nivel del lecho, se verifica la condición de deslizamiento en la dirección transversal y longitudinal de acuerdo con la metodología descrita por *Burkow (2010)*.

3.2. Modelo de dos fases

La formulación del modelo de flujo euleriano de dos fases, según lo señalado por *Chauchat et al. (2017)* y *Hsu et al. (2003)*, se obtiene promediando de forma local e instantánea las ecuaciones de conservación de masa y momentum del fluido y las partículas. Además, para aplicar estas ecuaciones en flujos turbulentos, se utiliza un promedio turbulento. En las ecuaciones 3.2.1 y 3.2.2 se muestran las ecuaciones de conservación de masa para la fase de partículas y de fluido, donde α y $\beta = 1 - \alpha$ son las concentraciones volumétricas de la fase de partículas y sedimento respectivamente y u_i^s , u_i^f son las velocidades de la fase de sedimento y fluido.

$$\frac{\partial \alpha}{\partial t} + \frac{\partial \alpha u_i^s}{\partial x_i} = 0 \quad (3.2.1)$$

$$\frac{\partial \beta}{\partial t} + \frac{\partial \beta u_i^f}{\partial x_i} = 0 \quad (3.2.2)$$

Las expresiones de momentum se detallan en las ecuaciones 3.2.3 y 3.2.4, donde ρ , ρ_s son las densidades del fluido y del sedimento, p es la presión del fluido, τ_{ij}^f es el esfuerzo de corte del fluido (el cual considera los esfuerzos viscosos y de Reynolds) y \tilde{p}^s , τ_{ij}^s son el esfuerzo normal y de corte de la partícula.

$$\begin{aligned} \frac{\partial \rho_s \alpha u_i^s}{\partial t} + \frac{\partial \rho_s \alpha u_i^s u_j^s}{\partial x_j} = & -\alpha \frac{\partial p}{\partial x_i} - \frac{\partial \tilde{p}^s}{\partial x_i} + \frac{\partial \tau_{ij}^s}{\partial x_j} + \alpha \rho_s g_i + \alpha \beta K (u_i^f - u_i^s) \\ & - SUS \beta K \nu_t \frac{\partial \alpha}{\partial x_i} \end{aligned} \quad (3.2.3)$$

$$\begin{aligned} \frac{\partial \rho \beta u_i^f}{\partial t} + \frac{\partial \rho \beta u_i^f u_j^f}{\partial x_j} = & -\beta \frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}^f}{\partial x_j} + \beta \rho g_i - \alpha \beta K (u_i^f - u_i^s) \\ & + SUS \beta K \nu_t \frac{\partial \alpha}{\partial x_i} \end{aligned} \quad (3.2.4)$$

Los dos últimos términos de estas ecuaciones representan el acoplamiento de momentum que existe entre la fase de sedimento y fluido a través de la fuerza de arrastre, donde K es el parámetro de arrastre. En particular, el último representa el efecto de la dispersión del transporte de partículas por la turbulencia del flujo, donde ν_t es la viscosidad de remolinos del flujo y SUS es el inverso del número de Schmidt, el cual es el cociente entre la viscosidad y la difusividad turbulentas. El término anterior a este reproduce la fuerza de arrastre promediada debido a la velocidad relativa entre ambas fases. El número de Schmidt se calcula en base a la proporción de la velocidad de sedimentación y la velocidad de corte acorde a lo propuesto por *Nagel et al. (2020)*.

Otras fuerzas como la de sustentación podrían jugar un rol en el transporte de sedimento, acorde a *Jha & Bombardelli (2010)*, esta fuerza en el transporte de sedimentos suspendidos diluidos representa solo un 4% de la fuerza de arrastre, la masa agregada puede ser del orden del 10% en la región cercana al lecho, y según *Bombardelli, González & Niño (2008)*, despreciar la fuerza de Basset en arenas, puede traer errores grandes en cálculos computacionales debido a errores del orden del 40% en la predicción del largo de los saltos de una partícula y del 15% en la altura de los saltos. Estas últimas requieren más investigación y escapan por el momento del CFD de dos fases a utilizar, por lo que se omiten en las ecuaciones.

El parámetro de arrastre se modela según lo propuesto por *Schiller & Naumann (1933)* y es posible visualizarlo en la ecuación 3.2.5, donde d es el diámetro de la partícula y $\beta^{-h_{Exp}}$ representa el incremento de arrastre cuando la concentración de partículas aumenta. El exponente de esta última función depende del número de Reynolds, pero por simplicidad se supone constante e igual a 2.65, lo cual es válido para números de Reynolds de partícula menores a la unidad o mayores 300, mientras que para el rango intermedio se considera igual 2 (*Chauchat et al, 2017*).

$$K = 0.75C_d \frac{\rho}{d} \|u^f - u^s\| \beta^{-h_{Exp}} \quad (3.2.5)$$

El coeficiente de arrastre se calcula como se indica en la ecuación 3.2.6, donde el número de Reynolds de partícula se define como $Re_p = \beta \|u^f - u^s\| d / \nu$, siendo ν la viscosidad cinemática del fluido.

$$C_d = \begin{cases} \frac{24}{Re_p} (1 + 0.15 Re_p^{0.687}) & Re_p \leq 1000 \\ 0.44 & Re_p > 1000 \end{cases} \quad (3.2.6)$$

El esfuerzo de corte τ_{ij}^f es igual a la suma de dos componentes, la de Reynolds R_{ij}^{ft} (ecuación 3.2.7) y la de escala granular r_{ij}^f (ecuación 3.2.8), la cual incluye los efectos viscosos y un efecto debido a la interacción entre fluido y partícula a esta escala, donde S_{ij}^f es la parte deviatorica del tensor de tasa de deformación del fluido y k es la energía cinética turbulenta.

$$R_{ij}^{ft} = \rho\beta \left[2\nu_t S_{ij}^f - \frac{2}{3} k \delta_{ij} \right] \quad (3.2.7)$$

$$r_{ij}^f = 2\rho\beta\nu^{mix} S_{ij}^f \quad (3.2.8)$$

Para la viscosidad equivalente de la mezcla entre el fluido y los sedimentos ν^{mix} , se utiliza el modelo empírico propuesto por *Boyer et al. (2011)*, el cual se muestra en la ecuación 3.2.9, donde α_{max} corresponde a la concentración máxima. Este es válido solo para el modelo de reología de flujos granulares densos, en caso de usarse el modelo de la teoría cinética de flujos granulares, la viscosidad de mezcla es igual a la cinemática.

$$\frac{\nu^{mix}}{\nu} = 1 + 2.5\alpha \left(1 - \frac{\alpha}{\alpha_{max}} \right)^{-1} \quad (3.2.9)$$

A continuación, se presentan los modelos de turbulencia disponibles en el software a utilizar, $k - \epsilon$ (*Cheng & Hsu, 2014*) y $k - \omega$ (*Chauchat et al., 2017*). En el primero, la viscosidad turbulenta se estima de igual forma que la presentada en el modelo análogo de una fase, mientras que la ecuación de transporte se diferencia en sus dos últimos términos (ecuación 3.2.10 y 3.2.11), los cuales toman en cuenta la amortiguación de los sedimentos en la turbulencia del flujo a través del arrastre y el flujo estratificado. En el término de amortiguamiento, el parámetro t_{mf} caracteriza el grado de correlación entre las fluctuaciones de la velocidad de las partículas y el fluido en función del número de Stokes (ecuación 3.2.12), el cual relaciona el tiempo de respuesta de las partículas con la escala de tiempo característica de remolinos.

$$\begin{aligned} \frac{\partial k}{\partial t} + u_j^f \frac{\partial k}{\partial x_j} &= \frac{R_{ij}^{ft}}{\rho} \frac{\partial u_i^f}{\partial x_j} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] - \epsilon - \frac{2K(1 - t_{mf})\alpha k}{\rho} \\ &\quad - \frac{SUS}{\beta} \nu_t \frac{\partial \alpha}{\partial x_j} \left(\frac{\rho_s}{\rho} - 1 \right) g_j \quad (3.2.10) \end{aligned}$$

$$\begin{aligned} \frac{\partial \epsilon}{\partial t} + u_j^f \frac{\partial \epsilon}{\partial x_j} = c_{\epsilon 1} \frac{\epsilon}{k} \frac{R_{ij}^{ft}}{\rho} \frac{\partial u_i^f}{\partial x_j} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right] - c_{\epsilon 2} \frac{\epsilon^2}{k} - c_{\epsilon 3} \frac{\epsilon}{k} \frac{2K(1-t_{mf})\alpha k}{\rho} \\ - c_{\epsilon 4} \frac{SUS\epsilon}{\beta k} \nu_t \frac{\partial \alpha}{\partial x_j} \left(\frac{\rho_s}{\rho} - 1 \right) g_j \quad (3.2.11) \end{aligned}$$

$$t_{mf} = e^{-Bst} \quad (3.2.12)$$

Donde B es un coeficiente empírico. El último término representa la flotabilidad, lo cual para concentraciones de sedimento típicas con un perfil que decae hacia arriba representa una estratificación estable, lo cual nos entrega otra fuente de atenuación para la turbulencia. *Hsu et al (2003)* propone que ha falta de datos experimentales comprensivos, los coeficientes experimentales de las ecuaciones anteriores son los mismos utilizados en el modelo de una fase. Además, $c_{\epsilon 3}$ se escoge igual a 1.2 y $c_{\epsilon 4}$ es igual a 0 para una condición de estratificación estable y 1 para la inestable.

En el modelo $k - \omega$, se define la viscosidad turbulenta de igual forma que en su caso análogo de una fase, mientras que las ecuaciones de transporte (ecuación 3.2.13 y 3.2.14) se diferencian en la adición de términos que denotan la influencia del arrastre de partículas y la suspensión turbulenta (de forma similar al modelo $k - \epsilon$ recién expuesto).

$$\begin{aligned} \frac{\partial k}{\partial t} + u_j^f \frac{\partial k}{\partial x_j} = R_{ij}^{ft} \frac{\partial u_i^f}{\partial x_j} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] - c_\mu k \omega - \frac{2K(1-t_{mf})\alpha k}{\rho} \\ - \frac{SUS}{\beta} \nu_t \frac{\partial \alpha}{\partial x_j} \left(\frac{\rho_s}{\rho} - 1 \right) g_j \quad (3.2.13) \end{aligned}$$

$$\begin{aligned} \frac{\partial \omega}{\partial t} + u_j^f \frac{\partial \omega}{\partial x_j} = c_{\omega 1} \frac{\omega}{k} R_{ij}^{ft} \frac{\partial u_i^f}{\partial x_j} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\omega} \right) \frac{\partial \omega}{\partial x_j} \right] - c_{\omega 2} \omega^2 - c_{\omega 3} \frac{2K(1-t_{mf})\alpha \omega}{\rho} + \\ CD_{k\omega} - c_{\omega 4} \frac{SUS\omega}{\beta k} \nu_t \frac{\partial \alpha}{\partial x_j} \left(\frac{\rho_s}{\rho} - 1 \right) g_j \quad (3.2.14) \end{aligned}$$

Para los coeficientes análogos al modelo de una fase se escogen los mismos valores, mientras que $c_{\omega 3} = 0.35$ y $c_{\omega 4}$ es igual a 1 para condiciones de estratificación inestable y 0 para

estables. El penúltimo término de la ecuación 3.2.14 es un término de difusión cruzada que se define acorde a la ecuación 3.2.15.

$$CD_{k\omega} = \frac{A}{\omega} \sigma_{d0} \mathcal{H} \left(\frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \right) \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \quad (3.2.15)$$

Donde $\sigma_{d0} = 1/8$ según *Wilcox (2008)*, \mathcal{H} es la función de escalón que toma el valor 1 si el argumento es no negativo y 0 si el argumento es negativo. Este término incrementa la disipación de energía cinética turbulenta en el flujo lejos de paredes potenciando la producción de disipación específica en esa región. El coeficiente $A = (1 + \tanh(-40(\alpha - 0.1)))/2$ es un término que suaviza la transición donde el término de difusión cruzada es válido y donde no (cerca de las paredes y pila). Además, $c_{2\omega} = c_{2\omega 0}(1 + 85\chi_\omega)/(1 + 100\chi_\omega)$ acorde a la corrección de *Pope (1988)*, donde $c_{2\omega 0} = 0.0708$, $\chi_\omega = |\Omega_{ij}\Omega_{jk}S_{ki}^f / (c_\mu \omega)^3|$ y $\Omega_{ij} = 0.5(\partial u_i^f / \partial x_j - \partial u_j^f / \partial x_i)$.

El tensor de esfuerzos de las partículas puede dividirse en dos, la presión sobre la partícula \tilde{p}^s y el esfuerzo de corte de la partícula $\tilde{\tau}_{ij}^s$. En general, la presión de la partícula puede deberse a dos contribuciones, una componente colisional o inducida por corte p^s y otra por contacto permanente p^{ff} . Esta última se calcula según la ecuación 3.2.16, donde $\alpha_{min}^{Fric} = 0.57$, $\alpha_{max} = 0.635$, $Fr = 0.05$, $\eta_0 = 3$ y $\eta_1 = 5$, son coeficientes empíricos.

$$p^{ff} = \begin{cases} 0 & \alpha < \alpha_{min}^{Fric} \\ Fr \frac{(\alpha - \alpha_{min}^{Fric})^{\eta_0}}{(\alpha_{max} - \alpha)^{\eta_1}} & \alpha \geq \alpha_{min}^{Fric} \end{cases} \quad (3.2.16)$$

Para la descripción de la presión colisional o inducida por corte de la partícula p^s y el esfuerzo de corte de la partícula $\tilde{\tau}_{ij}^s$, existen dos líneas principalmente para modelar el cierre de estos esfuerzos intergranulares, la teoría cinética de flujos granulares y la reología de flujos granulares densos, ambas se encuentran disponibles para ser utilizadas en el software a utilizar por lo que serán descritas a continuación.

En la teoría cinética de flujos granulares, las interacciones intergranulares se suponen dominadas por colisiones binarias para concentraciones bajas a moderadas de sedimento y los esfuerzos de corte colisionales son cuantificados por fluctuaciones de las velocidades de las partículas representadas por la temperatura granular $\Theta [L^2 T^{-2}]$. El modelo por describir es el propuesto por *Ding & Gidaspow (1990)*. En la ecuación 3.2.17 se muestra la ecuación de balance para la temperatura granular, donde q_j es el flujo de temperatura granular, γ es la

tasa de disipación de energía por colisiones inelásticas y J_{int} es la producción/disipación de energía por interacciones con el fluido. Además, se expone en la ecuación 3.2.18 el cierre para p^s , donde e es el coeficiente de restitución durante la colisión y g_{s0} es la función de distribución radial que describe el hacinamiento de la partícula (ecuación 3.2.19).

$$\frac{3}{2} \left[\frac{\partial \alpha \rho^s}{\partial t} + \frac{\partial \alpha \rho^s u_j^s \Theta}{\partial x_j} \right] = (-p^s \delta_{ij} + \tau_{ij}^s) \frac{\partial u_i^s}{\partial x_j} - \frac{\partial q_j}{\partial x_j} - \gamma + J_{int} \quad (3.2.17)$$

$$p^s = \rho_s \alpha [1 + 2(1 + e) \alpha g_{s0}] \Theta \quad (3.2.18)$$

$$g_{s0} = \frac{2 - \alpha}{2(1 - \alpha)^3} \quad (3.2.19)$$

El esfuerzo colisional de partícula τ_{ij}^s se define como se expone en la ecuación 3.2.20, donde S_{ij}^s es el tensor de esfuerzos deviatorico de partículas, μ^s es la viscosidad asociada a la interacción granular de partículas, definida en la ecuación 3.2.21, y λ es la viscosidad global la cual se calcula según la ecuación 3.2.22.

$$\tau_{ij}^s = 2\mu^s S_{ij}^s + \lambda \frac{\partial u_k^s}{\partial x_k} \delta_{ij} \quad (3.2.20)$$

$$\mu^s = \rho_s d \sqrt{\Theta} \left[\frac{4}{5} \frac{\alpha^2 g_{s0} (1 + e)}{\pi} + \frac{\sqrt{\pi} g_{s0} (1 + e) (3e - 1) \alpha^2}{15(3 - e)} + \frac{\sqrt{\pi} \alpha}{6(3 - e)} \right] \quad (3.2.21)$$

$$\lambda = \frac{4}{3} \alpha^2 \rho_s d g_{s0} (1 + e) \sqrt{\frac{\Theta}{\pi}} \quad (3.2.22)$$

Para el flujo de temperatura granular, se supone un comportamiento análogo a la ley de conducción de Fourier, donde la conductividad de la temperatura granular se define como se muestra en la ecuación 3.2.23. Por otro lado, γ y J_{int} se definen acorde a las ecuaciones 3.2.24 y 3.2.25.

$$D_\Theta = \rho_s d \sqrt{\Theta} \left[\frac{2\alpha^2 g_{s0}(1+e)}{\sqrt{\pi}} + \frac{9\sqrt{\pi} g_{s0}(1+e)^2(2e-1)\alpha^2}{2(49-33e)} + \frac{5\sqrt{\pi}\alpha}{2(49-33e)} \right] \quad (3.2.23)$$

$$\gamma = 3(1-e^2)\alpha^2 \rho_s g_{s0} \Theta \left[\frac{4}{d} \left(\frac{\Theta}{\pi} \right)^{\frac{1}{2}} - \frac{\partial u_j^s}{\partial x_j} \right] \quad (3.2.24)$$

$$J_{int} = \alpha K (2t_{mf} k - 3\Theta) \quad (3.2.25)$$

Se define el esfuerzo de corte de la partícula asociado al contacto friccional τ_{ij}^{ff} (ecuación 3.2.26), donde v_{Fr}^s es la viscosidad friccional, la cual se define en la ecuación 3.2.27, en la cual θ_f es el ángulo constante de fricción interna, $\|S^s\|$ es la norma del tensor de esfuerzos y $D_{small}^2 = 10^{-6} s^{-1}$ es una constante, la cual se añade para asegurar estabilidad en el modelo.

$$\tau_{ij}^{ff} = 2\rho_s v_{Fr}^s S_{ij}^s \quad (3.2.26)$$

$$v_{Fr}^s = \frac{p^{ff} \text{sen}(\theta_f)}{\rho_s (\|S^s\|^2 + D_{small}^2)^{1/2}} \quad (3.2.27)$$

Por último, el esfuerzo de corte total de la partícula $\tilde{\tau}_{ij}^s$ es igual a la suma de la componente colisional y friccional (ecuación 3.2.28).

$$\tilde{\tau}_{ij}^s = \tau_{ij}^s + \tau_{ij}^{ff} \quad (3.2.28)$$

Pasando a la otra alternativa, la modelación de la reología de flujos granulares densos es fenomenológica y se basa en el análisis dimensional. En esta, el esfuerzo de corte total de partícula $\tilde{\tau}_{ij}^s$ se calcula como la suma de la componente de Reynolds R_{ij}^s y la contribución proveniente de interacción entre partículas $r_{ij}^s = v_{Fr}^s S_{ij}^s$. La primera componente se estima con la ecuación 3.2.29, acorde a *Rusche (2003)* y C_t se considera igual a 1 (*Nagel et al., 2020*).

$$R_{ij}^s = \rho_s \alpha \left(C_t^2 v_t S_{ij}^s - \frac{2}{3} C_t^2 k \delta_{ij} \right) \quad (3.2.29)$$

Para ser consistentes con la notación de la primera alternativa, se seguirá introduciendo la viscosidad asociada a la interacción intergranular de las partículas μ^s y la viscosidad de corte friccional ν_{Fr}^s , fijando en este caso $\mu^s = 0$ y la viscosidad friccional como se expone en la ecuación 3.2.30. El coeficiente $\mu(I)$, depende del parámetro adimensional I , donde según el número local de Stokes y Reynolds de partícula se tiene un régimen que puede variar de inercial a viscoso, del cual depende la forma de definir el parámetro adimensional.

$$\nu_{Fr}^s = \frac{\mu(I)\tilde{p}^s}{\rho_s(\|S^s\|^2 + D_{small}^2)^{\frac{1}{2}}} \quad (3.2.30)$$

En régimen viscoso, el coeficiente de fricción depende del número viscoso $I_v = \|\nabla u^s\|v/(\rho\tilde{p}^s)$, donde u^s es el vector de velocidad de la partícula, y la viscosidad se define acorde a la ecuación 3.2.31, donde μ_3 , μ_2 e I_0 son iguales a 0.32, 0.7 y 0.005. Este régimen ocurre para números de Stokes menores a la unidad, según la definición de *Cassar et al., 2005*.

$$\mu(I_v) = \mu_3 + \frac{\mu_2 - \mu_3}{\frac{I_0}{I_v} + 1} \quad (3.2.31)$$

La presión inducida por corte de la partícula se estima con la ecuación (3.2.32). Con esto, es posible calcular la presión total de la partícula \tilde{p}^s como la suma de la presión inducida por corte y la inducida por fricción.

$$p^s = \left(\frac{\alpha}{\alpha_{max} - \alpha}\right)^2 \nu\|S^s\| \quad (3.2.32)$$

En régimen inercial, el coeficiente de fricción depende del número inercial $I = \|\nabla u^s\|d\sqrt{\rho_s/\tilde{p}^s}$ y se calcula como se expone en la ecuación 3.2.32, donde μ_4 , μ_5 e I_1 son iguales 0.64, 0.38 y 0.3.

$$\mu(I) = \mu_5 + \frac{\mu_4 - \mu_5}{\frac{I_1}{I} + 1} \quad (3.2.33)$$

La presión inducida por corte de la partícula se estima con la ecuación (3.2.34). Con esto, es posible calcular la presión total de la partícula \tilde{p}^s de manera análoga al régimen viscoso.

$$p^s = \left(\frac{0.66\alpha}{\alpha_{max} - \alpha} \right)^2 \rho_s d^2 \|S^s\|^2 \quad (3.2.34)$$

Para estabilizar el modelo se añade una relajación para la presión inducida por corte: $p_{nuevo}^s = p^s relax + (1 - relax)p_{antiguo}^s$. Esto es equivalente a suponer un tiempo de relajación para el efecto dilatante, así el material granular no se dilata instantáneamente ante una tasa de deformación, lo cual está justificado físicamente.

En el presente estudio se utilizará la reología de flujos granulares densos para modelar los esfuerzos intergranulares acorde al análisis realizado por *Nagel et al. (2018)*, debido a que se ajusta mejor en la parte con mayor densidad de partículas, y el modelo $k - \omega$ para el cierre de la turbulencia al igual que en el modelo de una fase, debido a la estabilidad observada del modelo SedFoam con este tipo de cierre respecto a la variante $k - \epsilon$, ya que con $k - \epsilon$ el modelo es mucho más inestable y difícil de tratar que $k - \omega$ actualmente.

4. Modelos Numéricos

A continuación, se describen los softwares con los respectivos métodos numéricos utilizados para resolver los modelos conceptuales planteados anteriormente.

4.1. REEF3D

Para el modelo de una fase, se hace uso del software de código abierto REEF3D presentado por *Bihs (2011)*, el cual resuelve la hidrodinámica y transporte de sedimento planteados. Además, es capaz de reproducir la superficie libre y el nivel del lecho. La capacidad para resolver la socavación en el tiempo en casos de erosión en torno a una pila circular para casos permanentes y frente a hidrogramas de crecida fue verificada por *Quezada et al. (2018)*.

Para discretizar las ecuaciones gobernantes, el presente modelo utiliza métodos de diferencias finitas conservativos y como en los problemas en estudio se utiliza una malla estructurada, las diferencias finitas son simples y efectivas. Sumado a esto, también es sencillo obtener esquemas de orden alto en mallas regulares. La malla además es de tipo escalonada, lo cual quiere decir que las variables están definidas en nodos diferentes. Acorde a *Saud (2013)*, una malla de este tipo genera un gran acoplamiento entre la velocidad y la presión. La generación de la malla mencionada se lleva a cabo con el software DIVEMesh, donde el criterio para la relación del tamaño de las celdas es de igual forma a la mencionada por *Quezada et al. (2018)* ($\Delta x = \Delta y = \Delta z$). En la Figura 4.1 se observa el tipo de grilla generada.

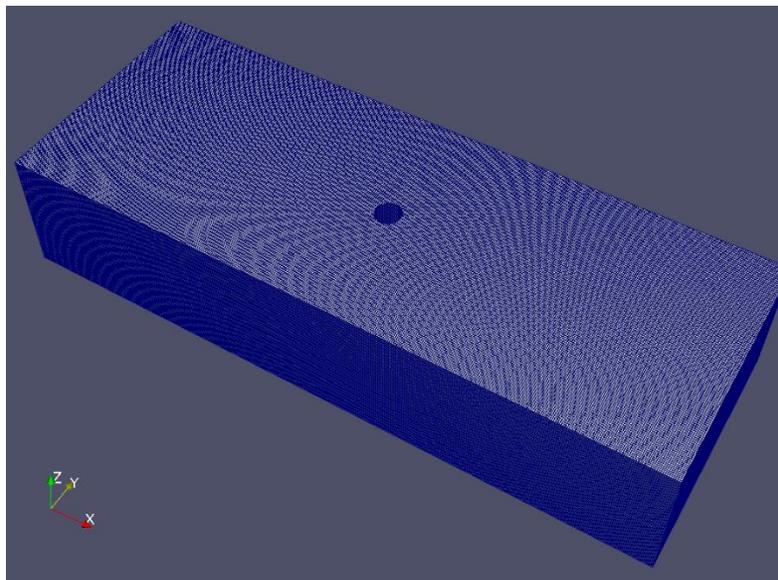


Figura 4.1 Malla generada con DIVEMesh para REEF3D.

Para tratar con el flujo en la cercanía de la pila se utiliza el método de celdas fantasmas (*Berthelsen, 2004*) para tener en cuenta la condición de borde que esta impone, el cual extrapola la solución hasta la región sólida (pila). Estas celdas que se generan en la región sólida son ficticias y se les llama celdas fantasmas. Esto asegura que las condiciones de borde se impongan implícitamente y que la discretización numérica no tenga la necesidad de tenerlas en cuenta explícitamente.

El esquema WENO de quinto orden (*Jiang & Shu, 1996*) es utilizado para discretizar los términos convectivos. En el caso de la discretización temporal, esta se lleva a cabo aplicando el método de Runge-Kutta de tercer orden (*Shu & Gottlieb, 1998*). Los intervalos de tiempo se estiman automáticamente mediante el criterio de Courant-Friedrichs-Lewy en el software. Para tratar con la presión se utiliza el método de proyección propuesto por *Chorin (1968)* y la ecuación de Poisson es resuelta usando el solver incorporado en REEF3D, BICGStab (*Van der Vorst, 1992*).

Las condiciones de borde para las simulaciones hidrodinámicas son no deslizamiento para las velocidades. Para las variables de la turbulencia k y ϵ u ω (dependiendo del caso) funciones puente (leyes de pared). Un perfil logarítmico para el flujo entrante, corrección de la presión en la entrada de flujo igual a 0 en la zona superior y un gradiente igual a cero en la salida del flujo. Para inicializar el flujo, se utiliza flujo potencial para las velocidades e hidrostática para la presión.

REEF3D es capaz de reproducir, tanto la superficie libre como el nivel del lecho, haciendo uso del método de ajuste de nivel propuesto por *Osher & Sethian (1988)*. En este, la ubicación de las interfases se representa implícitamente mediante el cero de una función de distancia suavizada, la cual entrega el valor de distancia mínimo a la interfaz en cada punto del dominio. Así, la interfaz se distingue por el cambio de signo de la función. Con la evolución del tiempo esta función deja de representar poco a poco esta distancia, y es necesario reinicializarla cada cierto tiempo, por lo que la calidad de esta reinicialización tiene una fuerte influencia en la precisión del método.

Idealmente, en el cálculo de la hidrodinámica y la evolución del lecho, estos debiesen estar acoplados, utilizando el mismo intervalo de tiempo, pero este acoplamiento es muy costoso computacionalmente. En este caso, se considera un modelo totalmente desacoplado, suponiendo que no existe interacción del flujo con el transporte de sedimento o cambios en el lecho. Así, se utiliza un intervalo de tiempo morfológico mucho mayor al intervalo de tiempo hidrodinámico, lo cual es válido para flujos en los cuales la suspensión de sedimento es baja de tal manera que no afecta el flujo formando un efecto retroalimentador.

El coeficiente de relajación (K_r) se utiliza como parámetro de calibración en el transporte de sedimentos de los modelos, dada la sensibilidad que este produce en los resultados.

REEF3D está diseñado para hacer uso de múltiples procesadores en virtud de aumentar la eficiencia de la simulación numérica, descomponiendo el problema en partes pequeñas y resolviéndolos de manera simultánea. Así, en el presente se realizan las simulaciones haciendo uso de 88 procesadores (2 núcleos) del supercomputador Leftraru perteneciente al

NLHPC (National Laboratory for High Performance Computing Chile), los cuales cuentan con 192 GB de RAM por cada 44 procesadores (1 núcleo).

Por último, el código de la versión 20.02 debió ser modificado para incorporar la de velocidad vertical en la estimación de los esfuerzos de corte tal como se expone en la ecuación 3.1.20, pues esta no se encontraba bien definida. Además, se debió corregir algunas variables que no se encontraban bien referenciadas en el código, siendo las más importantes, la porosidad y esfuerzos de corte, los cuales, si se variaban en los archivos de definición del modelo, no afectaban en el cálculo de las simulaciones. Realizar estas correcciones involucró mucho tiempo, pues se notó que existían estos problemas tras realizar muchas simulaciones. En el Anexo A se detalla un ejemplo de los archivos utilizados en las modelaciones.

4.2.OpenFOAM: SedFoam

Para la implementación numérica del modelo de dos fases descrito anteriormente, se utiliza el solver SedFoam de OpenFOAM, el cual es una modificación del solver twoPhaseEulerFoam. Este utiliza el método de volúmenes finitos para discretizar las ecuaciones diferenciales del problema, almacenando todas las variables en el centro de las celdas y utiliza el mismo volumen de control para todas las variables disminuyendo el esfuerzo computacional. La capacidad para la aplicación de este en problemas de socavación fue estudiada por *Nagel et al. (2018)*.

La metodología numérica se describe en detalle en *Chauchat et al. (2017)*, donde la velocidad y presión se resuelven de manera acoplada usando el método de *Rhie & Show (1983)*. El algoritmo PISO es utilizado para resolver las velocidades de fluidos y partículas (*Rusche, 2003*) y la ecuación de Poisson es resuelta mediante un solver GAMG. Para la discretización temporal se utiliza el esquema de Euler implícito, mientras que para los demás operadores se utilizan esquemas de Gauss. Los esquemas pueden verse detalladamente en el Anexo C.

La única diferencia respecto a la metodología expuesta por *Chauchat et al (2017)*, es el solucionador utilizado para resolver la presión, pues, en el presente se utiliza la metodología de *Nagel et al. (2020)*, donde el solucionador itera sobre la presión reducida p^* , la cual es igual a la resta entre la presión total y la hidrostática. Según *Nagel et al. (2020)*, este solucionador es más estable que el descrito en *Chauchat et al. (2017)*, permitiendo incrementar el criterio de Courant, reduciendo así el costo computacional.

Para ajustar el intervalo de tiempo, se utilizan dos números de Courant-Friedrichs-Lewy, uno basado en la velocidad local del flujo, y otro basado en la velocidad relativa, siendo el intervalo de tiempo más restringido el utilizado. En el presente trabajo se utilizará estos igual a 0.3 y 0.1 respectivamente (*Chauchat et al, 2017*), debido a la inestabilidad del modelo para números de Courant-Fridrichs-Lewy mayores.

En este caso, dado el costo computacional, el dominio de modelación difiere del modelado con REEF3D y se compone de un canal rectangular con un cilindro bajo la base del canal

donde se emplaza la pila, el cual tiene un radio igual al doble del diámetro de la pila y una profundidad igual a 5 veces el diámetro. Además, el largo es igual a 13 veces el diámetro y el ancho 8 veces el diámetro (ver Figura 4.2). Es importante mencionar que la pila no se encuentra centrada en la malla, si no que está desplazada hacia la entrada del flujo, en la misma proporción que en la modelación realizada por Nagel *et al.* (2020).

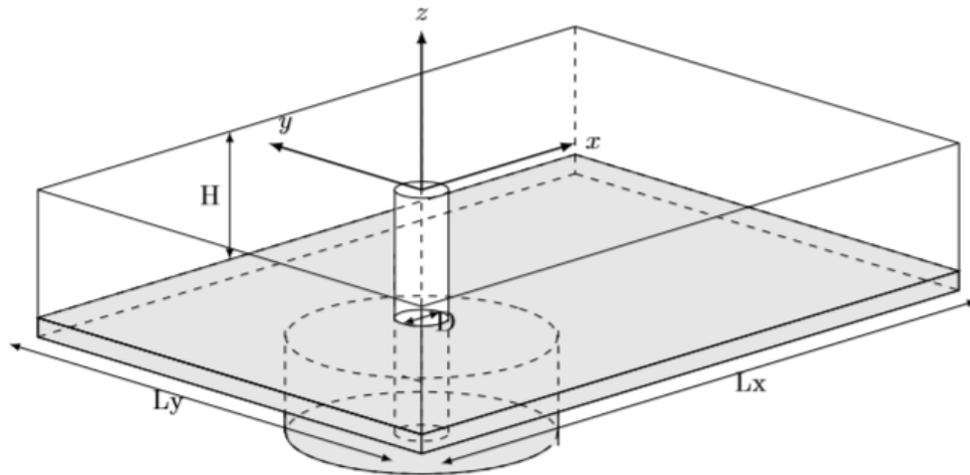


Figura 4.2 Esquema de la geometría de la malla utilizada con SedFoam. Fuente: Nagel *et al.* (2018).

El dominio computacional se discretiza utilizando una malla no estructurada, donde se realiza una refinación alrededor de la pila (axisimétrica, ver Figura 4.3) y en torno a la interfase (ver Figura 4.4). Esta malla se genera con un código que genera un archivo blockMesh, el cual se muestra explícitamente en el Anexo B.

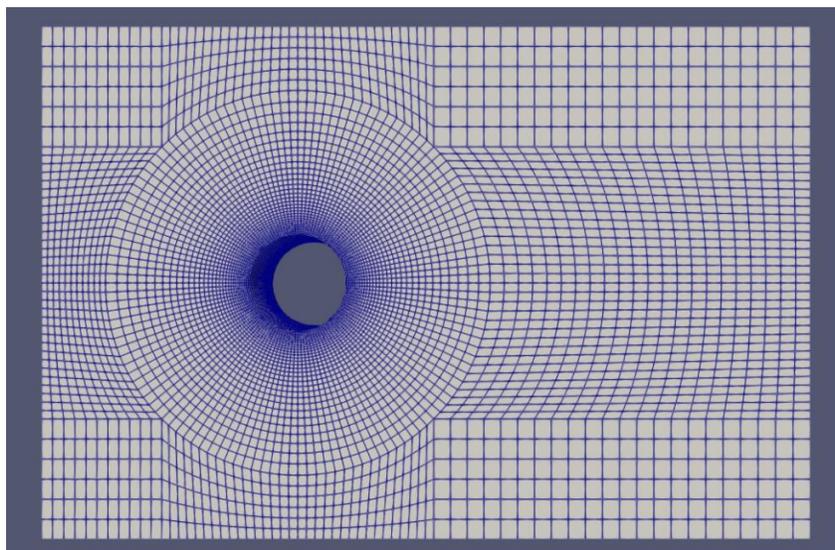


Figura 4.3 Vista en planta de malla generada para SedFoam.

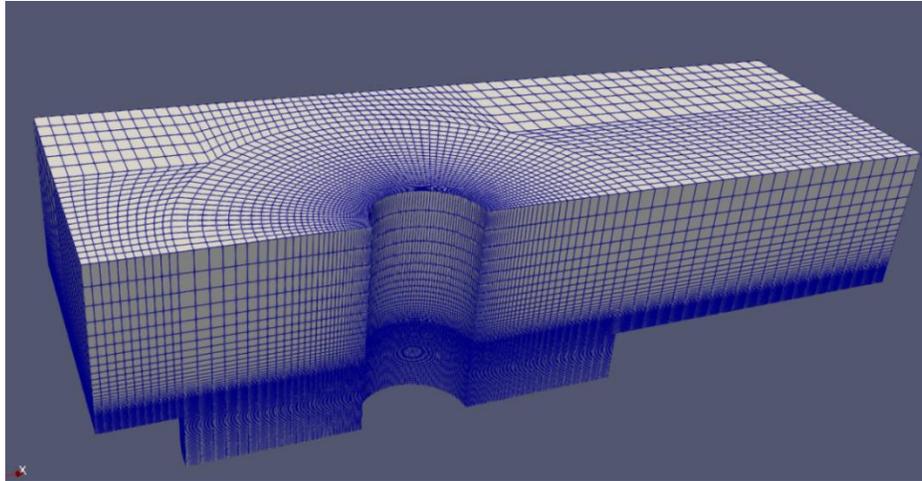


Figura 4.4 Corte Longitudinal de malla generada para SedFoam.

Las condiciones de borde en la entrada es un perfil logarítmico y velocidades transversales igual a 0, las variables turbulentas se estiman en función de la velocidad ($k = 1.5(I * u^f)^2$, donde I es la intensidad turbulenta (la cual se considera constante en todo el dominio) y $\omega = \sqrt{k}/l$, donde k es la energía cinética turbulenta, ω es la disipación específica de energía cinética turbulenta y l es la escala de longitud turbulenta) y en el caso de la concentración, se impone un perfil de concentración de sedimentos con una tangente hiperbólica (Anexo C) para tener una transición suave de la concentración de sedimentos en la interfaz (esto no implica una condición de lecho vivo, pues es no es tan suave, pero si lo suficiente como para no tener problemas numéricos), ya que se observó que las transiciones fuertes generan inestabilidades en la velocidad de los sedimentos. Además, es importante mencionar, que los sedimentos se sitúan no solo en la zona del cilindro de socavación, sino que se sitúan también en el resto de la malla, con un espesor de capa de 1/6 del diámetro de la pila. En la salida se utiliza la condición de gradiente igual a 0 (Neumann) para todas las variables excepto para la presión reducida, la cual se fija igual 0 (Dirichlet), y las velocidades en el caso de no ser en la dirección de la salida se imponen igual a 0. En la tapa se impone una condición de Neumann para la velocidad y las variables representativas de la turbulencia. En las murallas, incluyendo el cilindro, se impone la condición de no deslizamiento para la velocidad y funciones puente para las variables turbulentas predefinidas en OpenFoam (kqRWallFunction, omegaWallFunction y nutkWallFunction). Al igual que el perfil impuesto en la entrada, como condición inicial, se imponen los perfiles obtenidos de la simulación sin sedimentos.

El parámetro de calibración es el coeficiente B presente en la función que describe el parámetro t_{mf} , el cual caracteriza el grado de correlación entre las fluctuaciones de la velocidad de las partículas y el fluido. Según Nagel (2018) un aumento de 0.25 a 1.25 en el coeficiente B implica un aumento del transporte de sedimento en un 20% en un caso de transporte de sedimento simple sin obstáculos.

OpenFOAM al igual que REEF3D está condicionado para poder utilizar varios procesadores en virtud de aumentar la eficiencia de cómputo, por lo que se utilizaron 120 procesadores en cada simulación del supercomputador Leftraru, dado que el paquete SedFoam demanda más recursos computacionales que REEF3D.

5. Casos en estudio

5.1. Caso Permanente

5.1.1. Numérico

La modelación numérica con la cual se comparará para el caso permanente es la realizada por *Baykal et al. (2015)*, el cual proporciona resultados de la socavación aguas arriba y aguas abajo en el tiempo en torno a una pila circular de diámetro $D = 0.04 \text{ m}$, para un tiempo de simulación de 120 s, en un canal rectangular de 0.8, 0.6 y 0.2 m de largo, ancho y alto, donde 0.12 m corresponden a una capa de sedimento y 0.08 m a la capa de agua. La descarga es constante e igual a $0.02 \text{ m}^3/\text{s}$, el diámetro del sedimento es igual a 0.17 mm , $\rho_s = 2650 \text{ kg/m}^3$ y el ángulo de reposo 30° .

La elección de este caso en vez de uno experimental es el alto costo computacional que demanda el software de dos fases, el cual restringe el tiempo del experimento a comparar, lo cual limita los potenciales casos disponibles en la literatura pues la escala de tiempo en que se alcanza un cuasi equilibrio morfodinámico es mucho mayor a los recursos computacionales disponibles. A pesar de esto, ambos softwares han probado su capacidad para reproducir este tipo de casos. En el caso de REEF3D *Quezada et al. (2018)* realizó comparaciones con *Link (2006)*, mientras que con OpenFOAM *Nagel et al. (2020)* comparó parcialmente la socavación en el tiempo con *Roulund et al. (2005)*.

5.1.2. REEF3D

Para representar el caso planteado, se modela un canal de 0.8 m de largo, 0.6 m de ancho y 0.2 m de profundidad, donde 0.12 m corresponden a una capa de sedimento y 0.08 m a una capa de agua. La pila de 0.04 m de diámetro se sitúa en el centro del dominio y el transporte de sedimento se estima en una zona que comprende todo ancho del canal, desde 2 veces el diámetro de la pila tanto aguas arriba, como aguas abajo. Es esta zona donde se aplica el coeficiente K_r , el cual es igual a 1, es decir no se aplica corrección, en contraste a lo estudiado por *Quezada et al. (2018)*, donde declaran $K_r = 0.8$. La porosidad es igual a 0.45, el esfuerzo de corte crítico es igual a 0.06 sin las correcciones por pendiente ni el coeficiente de relajación y el ancho de las celdas es igual a 0.005 m. Además, es importante señalar que el modelo hidrodinámico se deja correr 10 veces la escala de tiempo $t^* = l/u_m$ antes de comenzar a correr el modelo morfológico para que el flujo se desarrolle, donde l es el largo del dominio y u_m la velocidad media del flujo.

Por último, en este caso se utilizó el modelo $k - \epsilon$, debido a que *Quezada et al. (2018)* sugiere que este cierre de turbulencia representa mejor la socavación, en este caso tanto aguas arriba, como aguas abajo de la pila (en ese trabajo se utilizó $K_r = 0.8$). Este modelo tiene alrededor de 1 millón de celdas, aunque gran parte de estas no se encuentran activas, esto es fuera del dominio del fluido, ya sea donde se encuentran los sedimentos o sobre la superficie libre.

5.1.3. OpenFOAM

Dados los recursos computacionales que el paquete SedFoam demanda, no fue factible realizar modelaciones para todos los casos propuestos inicialmente, por lo que se optó por dar prioridad al caso con resultados experimentales disponibles para comparar.

5.2. Casos Frente a Hidrogramas de Crecida

5.2.1. Experimental

Los experimentos se llevaron a cabo en el laboratorio de hidráulica de la Universidad de Concepción por *Link et al. (2017)*. El canal utilizado posee una sección transversal rectangular con una longitud de 26 m de largo, 1.5 m de ancho y 0.74 m de profundidad (Figura 5.1). En este se obtuvieron resultados para la socavación en torno a una pila en el tiempo para una condición de aguas claras frente a diversos hidrogramas de crecida, de los cuales, dados los recursos disponibles, se reproduce el fenómeno de socavación de un hidrograma de crecida, el cual se expone en la Figura 5.2.

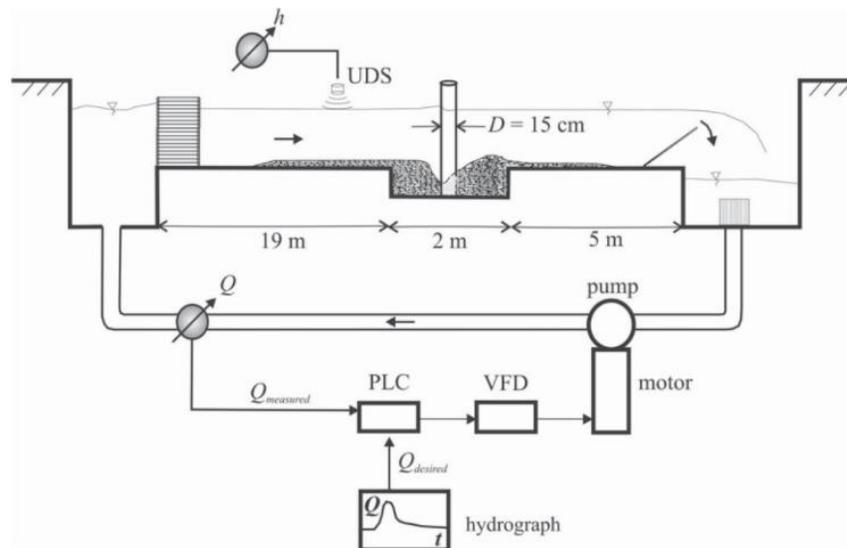


Figura 5.1 Instalación experimental utilizada por *Link et al. (2017)*.

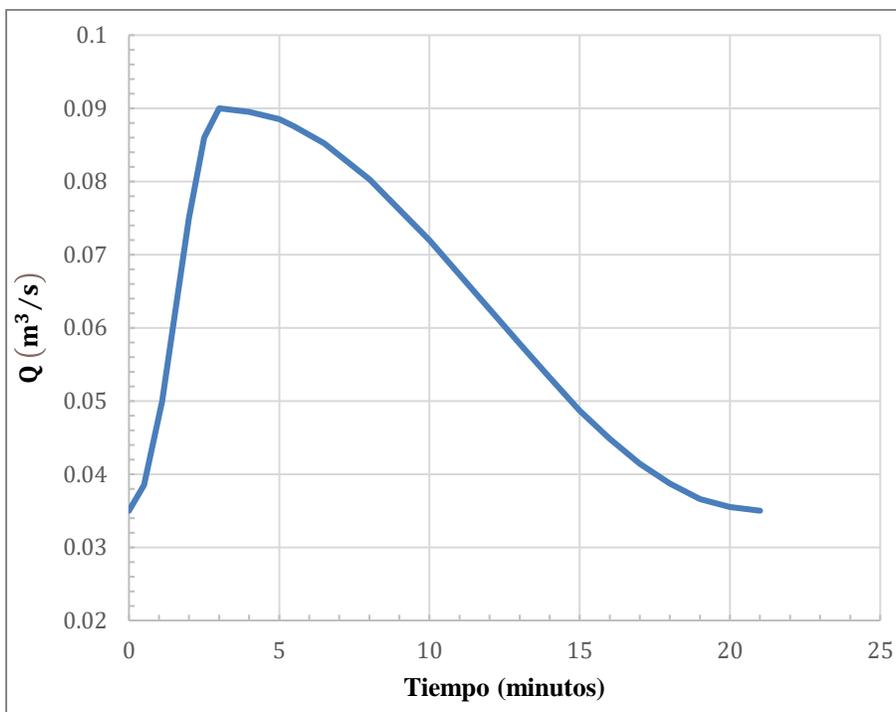


Figura 5.2 Hidrograma 1. Fuente: Link et al (2017).

La pila, cuyo diámetro es $D = 0.15 \text{ m}$, se sitúa en el centro de la sección transversal. El sedimento utilizado es arena con $d_{50} = 0.36 \text{ mm}$, $\rho_s = 2650 \text{ kg/m}^3$ y un ángulo de reposo de 30° . El arrastre incipiente ocurre a una velocidad igual a 0.32 m/s acorde a Link et al (2017).

5.2.2.REEF3D

Para representar el caso experimental planteado, se modela un canal de 4 m de largo, 1.5 m de ancho y 1 m de profundidad, donde 0.5 m corresponden a una capa de sedimento y 0.2 m a una capa de agua. La pila de 0.15 m de diámetro se sitúa en el centro del dominio y el transporte de sedimento se estima en una zona que comprende todo ancho del canal, desde 2 veces el diámetro de la pila aguas arriba, hasta la misma distancia aguas abajo.

Es esta zona donde se aplica el coeficiente K_r , el cual también se considera igual a 1 en este caso, mientras que Quezada et al (2018) declara $K_r = 0.8$. La porosidad es igual a 0.45, el esfuerzo de corte crítico es igual a 0.032 sin las correcciones por pendiente ni el coeficiente de relajación y el ancho de las celdas es igual a 0.01 m . Este modelo tiene cerca de 7 millones de celdas, aunque gran parte de estas no se encuentran activas, esto es fuera del dominio del fluido, ya sea donde se encuentran los sedimentos o sobre la superficie libre. En esta simulación se utiliza el modelo $k - \omega$, para poder comparar con OpenFoam.

Al igual que en el caso permanente, el modelo hidrodinámico se deja correr 10 veces la escala de tiempo antes de comenzar a correr el modelo morfológico para que la hidrodinámica se estabilice en su condición inicial.

5.2.3.OpenFOAM

Tal y como se mencionó en la sección de métodos numéricos, se consideró una malla con un dominio más reducido en honor a los recursos disponibles, utilizando una malla que tiene un largo de 13 veces el diámetro de la pila ($D=0.15$ m), un ancho de 8 veces el diámetro de la pila y un cilindro bajo la base del canal, el cual tiene un radio igual al doble del diámetro de la pila y una profundidad igual a 5 veces el diámetro. La capa de agua es igual a 0.2 m. El tamaño mínimo de celda generada es del orden del tamaño de los sedimentos en la zona cercana a la interfaz, y el detalle de la forma en que se definen los espaciamientos en cada eje, puede observarse en el Anexo B.

Es importante mencionar que, para acelerar el proceso de simulación, fue necesario dividir el dominio de la grilla a la mitad, mediante un corte longitudinal en el centro de la malla e imponiendo condiciones de simetría plana en esta zona. Para lograr esto, se hizo uso de las utilidades `topoSet`, `subsetMesh` y `createPatch` de OpenFoam (como se implementan, se expone en el Anexo C). Con esto la malla tiene alrededor medio millón de celdas.

El parámetro de calibración B se fijó igual a 1 para la simulación y los números de Courant-Friedrichs-Lewy señalados en la sección de Modelos Numéricos, con el fin de conseguir mayor velocidad de cómputo, fueron relajados en la medida de lo posible (el máximo alcanzado sin generar inestabilidades fue de 0.5).

Por último, para comenzar a representar el hidrograma, primero se debió estabilizar una condición inicial, la cual se generó comenzando con el fluido del reposo, y se fue aumentando el caudal paulatinamente para no tener problemas de inestabilidades, hasta llegar al caudal inicial ($Q = 0.035 \text{ m}^3/s$).

6.Resultados

En el presente capítulo se exponen los resultados obtenidos de las modelaciones descritas en la sección de Casos en Estudio, con los modelos expuestos en el capítulo de Modelos Numéricos. En la Tabla 6.1 se detallan los tiempos de simulación y los tiempos de computación de los casos estudiados en cada software. Adicionalmente, se señala un tiempo equivalente estimado de cuanto demoraría realizar la simulación utilizando solamente 1 núcleo, considerando un comportamiento lineal al aumentar o disminuir el número de núcleos. Para el caso de Link et al. (2017) con SedFoam se señala un tiempo proyectado si se hubiese corrido todo el hidrograma con el mismo rendimiento observado.

Tabla 6.1 Tiempos de cálculo empleados en cada simulación para cada software.

Caso	Software	Núcleos	Tiempo de simulación (s)	Tiempo de cómputo (hrs)	Tiempo de cálculo equivalente (días)
Link et al. (2017)	REEF3D	88	1260	20.5	75.2
	SedFoam	120		1080	5400
Baykal et al. (2015)	REEF3D	88	120	10.3	37.8

6.1.Caso Permanente

A continuación, se detalla la evolución de la socavación máxima adimensional en función del tiempo para el caso de *Baykal et al. (2015)* estimada mediante REEF3D, comparada con la evolución estimada mediante modelos numéricos señalada por *Baykal et al. (2015)* y *Quezada et al. (2018)*. De este último trabajo, se muestran los resultados obtenidos utilizando el modelo de turbulencia $k - \epsilon$ y $K_r = 0.8$.

En la Figura 6.1, se expone la socavación en la zona aguas arriba de la pila, y en la Figura 6.2, la socavación en la zona aguas abajo de la pila. En estas figuras se puede apreciar que la socavación máxima adimensional para el estado final en la zona aguas arriba es de 0.85 y aguas abajo es de 0.71, mientras que *Baykal et al. (2015)* señala valores de 0.87 y 0.7 y *Quezada et al. (2018)* obtuvo valores de 0.98 y 0.73.

Los resultados obtenidos, denotan un mejor ajuste a los de *Baykal et al. (2015)*, respecto a los obtenidos por *Quezada et al. (2018)*, en particular en la socavación máxima final y en el inicio del proceso de socavación observado. Esto podría atribuirse a las correcciones en el código de REEF3D realizadas, en particular incorporar la componente vertical de la velocidad en el cálculo de los esfuerzos de corte lo cual incrementa directamente el transporte de sedimentos, pudiendo prescindir del coeficiente de relajación, además la versión utilizada

por *Quezada et al. (2018)* es una más antigua a la utilizada en la presente tesis. Por otro lado, en *Quezada et al. (2018)* no se declara la porosidad utilizada, lo cual afecta en la velocidad con que se socava el lecho (afecta directamente en la ecuación de Exner), lo cual puede explicar la diferencia en la velocidad con que se socava el lecho al comienzo de la simulación.

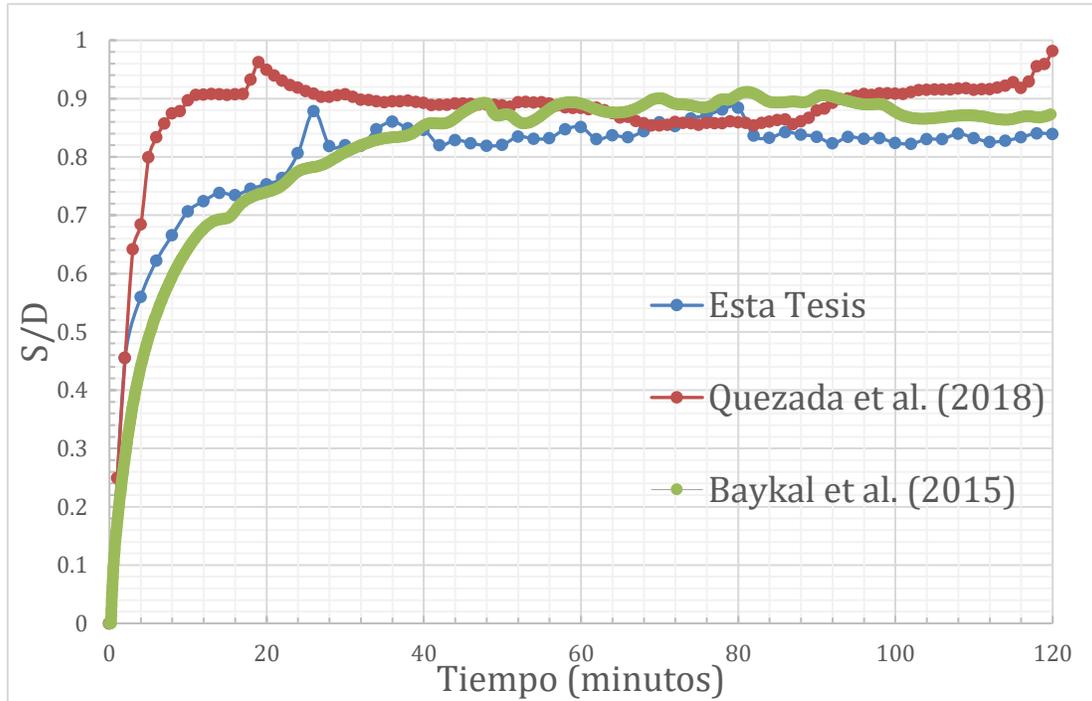


Figura 6.1 Evolución temporal de la socavación máxima en el tiempo, aguas arriba del centro de la pila para el caso de flujo permanente $Q = 0.02 \text{ m}^3/\text{s}$ estudiado por Baykal et al. (2015).

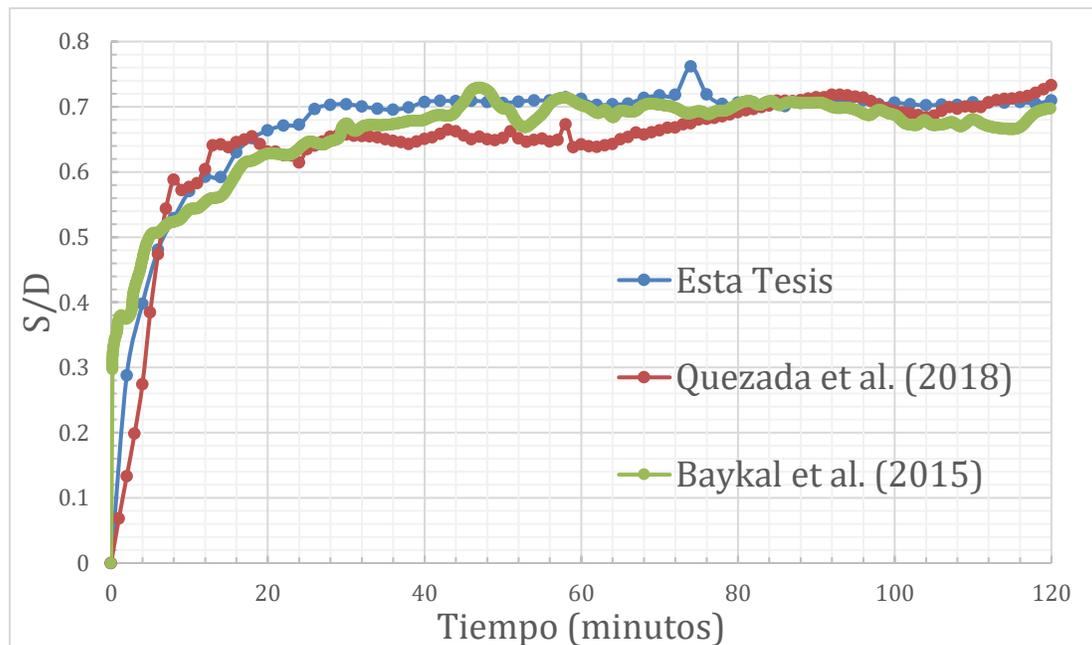


Figura 6.2 Evolución temporal de la socavación máxima en el tiempo, aguas abajo del centro de la pila para el caso de flujo permanente $Q = 0.02 \text{ m}^3/\text{s}$ estudiado por Baykal et al. (2015).

En la Figura 6.3 se muestra la serie de tiempo de la evolución de la socavación adimensional (S/D), en una zona de $4D$ de largo y ancho alrededor de la pila obtenida de la simulación en REEF3D. Donde se observa que se generan dos bulbos a los costados por la zona aguas arriba de la pila que se van juntando a medida que aumenta la socavación, lo cual difiere de lo que se observa generalmente en los experimentos donde se observa el máximo del bulbo en el punto central de la pila aguas arriba. Rápidamente el bulbo de socavación alcanza un diámetro de aproximadamente veces el diámetro de la pila (levemente más ancho que largo), estando centrado respecto a la pila si dividimos respecto al eje x y con un largo aproximado de $1.5D$ aguas arriba desde el centro de la pila y $3.5D$ hacia aguas abajo.

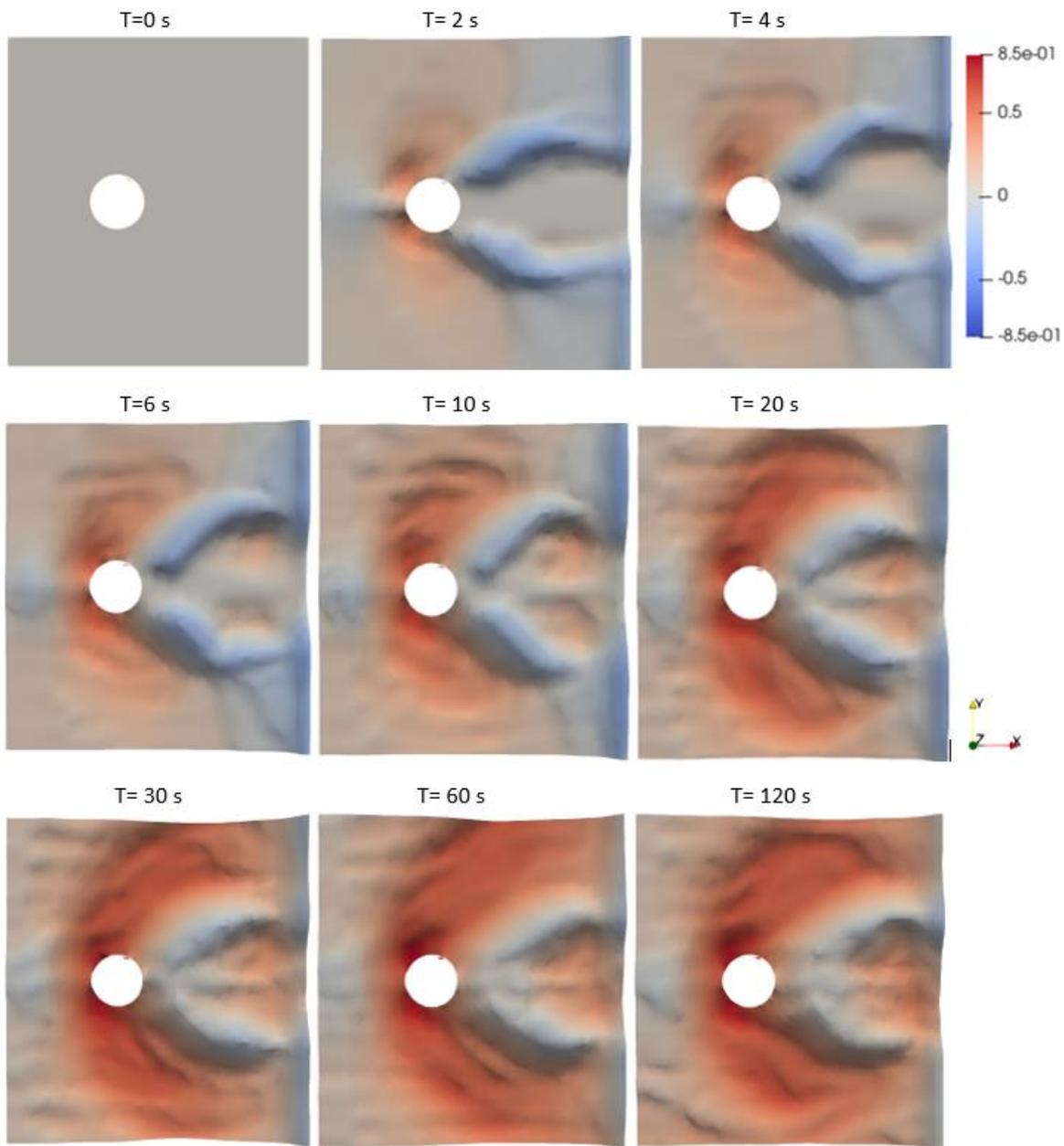


Figura 6.3 Evolución de la socavación adimensional (S/D) en la región cercana a la pila.

El campo de la componente de la velocidad en el eje x, en el momento final de la simulación, se expone en la Figura 6.4, junto al lecho deformado, y en la Figura 6.5, se muestra el campo de velocidad en planta. En estas, se observa que existen velocidades negativas en torno a la pila, lo cual hace sentido debido a la presencia del obstáculo en el flujo, lo cual se asocia a la disminución de presión en el caso aguas arriba de la pila y aguas abajo, debido a que las capas límite generadas a los costados de la pila, se enrollan hacia el centro.

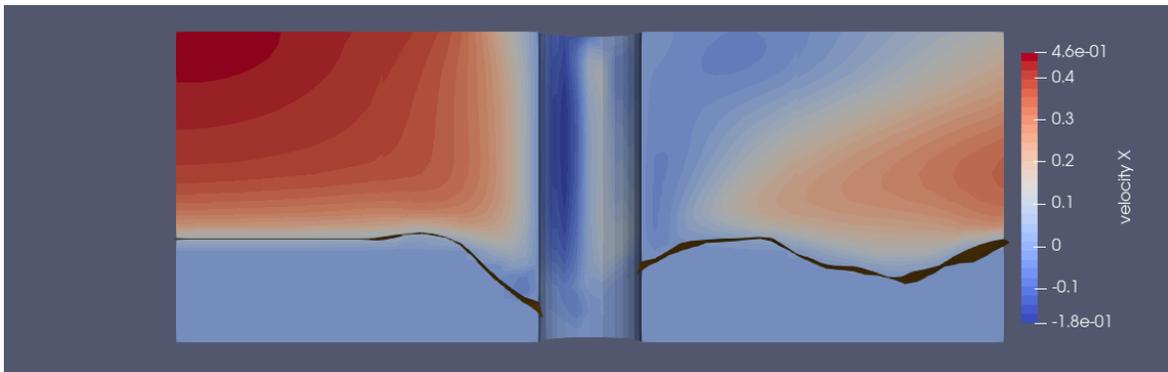


Figura 6.4 Perfil de velocidad longitudinal (m/s) en el centro de la pila modelado con REEF3D. Caso Baykal et al. (2015), flujo permanente $Q = 0.02 \text{ m}^3/\text{s}$, $D = 0.04 \text{ m}$. Considera desde en x desde $4D$ aguas arriba del centro de la pila hasta $4D$ aguas abajo del centro de la pila.

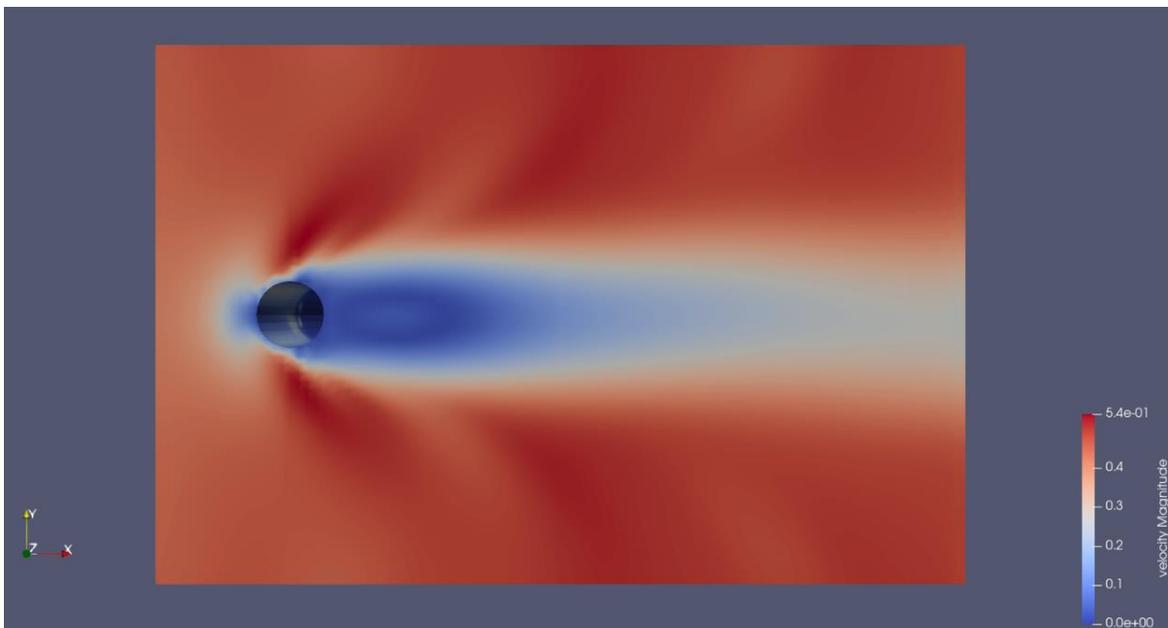


Figura 6.5 Perfil de velocidad en planta (m/s) de la superficie libre modelado con REEF3D. Caso Baykal et al. (2015), flujo permanente $Q = 0.02 \text{ m}^3/\text{s}$, $D = 0.04 \text{ m}$. Considera desde en x desde $2D$ aguas arriba del centro de la pila hasta $10D$ aguas abajo del centro de la pila y $8D$ de extensión en el eje y.

Las líneas de corriente del flujo alrededor de la pila para el tiempo final de la modelación pueden observarse en la Figura 6.6. En la zona anterior a la pila se distingue la rotación hacia el fondo del flujo entrante a la pila (vórtice de herradura), y se distingue un leve enrollamiento de las líneas de corriente en la zona justo aguas abajo, pero dista de la forma esperada del vórtice de estela.

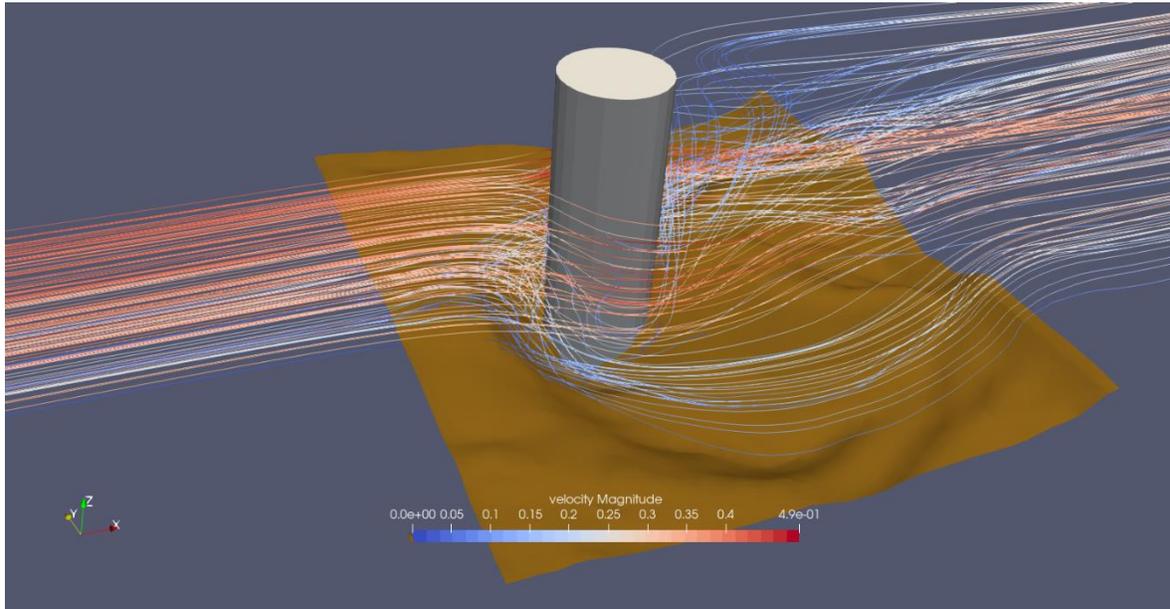


Figura 6.6 Estructuras de flujo en el caso de Baykal et al. (2015), flujo permanente $Q = 0.02 \text{ m}^3/\text{s}$, $D = 0.04 \text{ m}$ modelado con REEF3D ($t=120 \text{ s}$).

6.2.Casos Frente a Hidrogramas de Crecida

La evolución de la socavación máxima en el tiempo para el caso experimental en el estudio de Link (2017) se expone en las Figuras 6.7 y 6.8, comparada con la evolución obtenida mediante la simulación realizada con SedFoam y REEF3D. La evolución de la socavación máxima en el caso de la simulación con SedFoam es parcial debido a que no se logró estabilidad en la interfaz del modelo para velocidades más altas que las que alcanza el flujo para el tiempo máximo presentado (aproximadamente $0.3 \text{ m}^3/\text{s}$).

La inestabilidad observada se atribuye a problemas del modelo Euler-Euler con velocidades altas, en particular en la zona de la interfaz del lecho, donde se genera la inestabilidad, siendo muy difícil durante la modelación encontrar una condición de borde (e inicial) estable para la concentración de sedimentos. Esto pues al aumentar la velocidad en el fondo, se genera en ciertos puntos una concentración muy alta, cercana a la máxima tolerada por el modelo, por lo que la presión por contacto permanente de la partícula alcanza valores muy altos,

provocando una divergencia de la velocidad en esos puntos, generando inestabilidad en el modelo.

En el caso de los resultados realizados en esta investigación con REEF3D, se observa una socavación máxima adimensional, tras la simulación del hidrograma de 0.279, mientras que Link et al. (2018) declara 0.274. Respecto al comienzo de la socavación, se observa un mejor ajuste al caso experimental por parte de la simulación con SedFoam, respecto a la realizada con REEF3D.

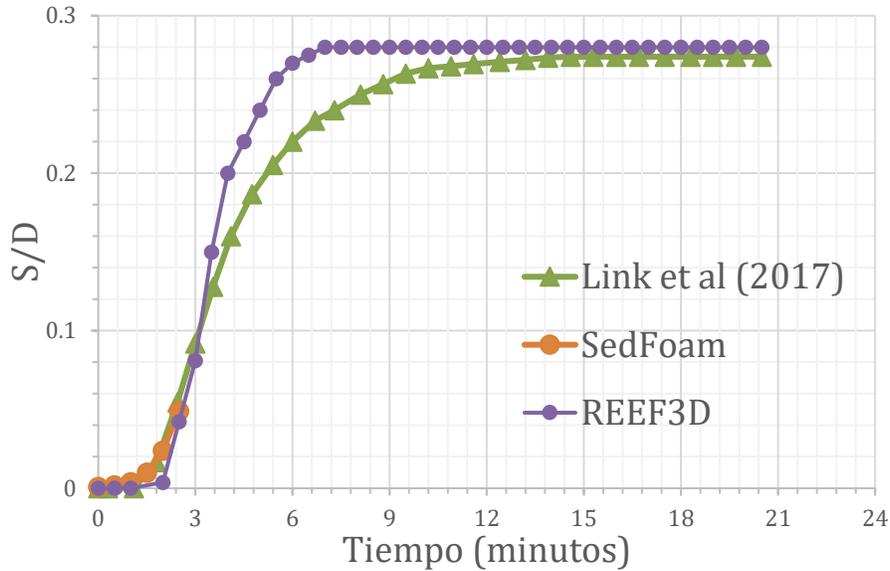


Figura 6.7 Evolución temporal de la socavación máxima en el tiempo. Caso con flujo permanente, hidrograma de crecida propuesto por Link et al. (2017). $D = 0.15$ m.

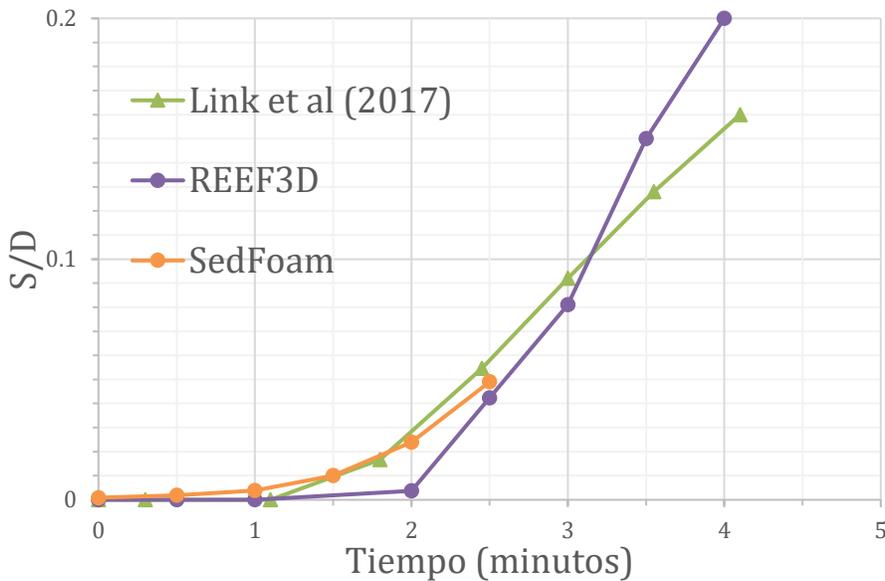


Figura 6.8 Acercamiento a la evolución temporal de la socavación máxima en el tiempo, a los tiempos en que fue posible modelar con SedFoam. Caso con flujo permanente, hidrograma de crecida propuesto por Link et al. (2017). $D = 0.15$ m.

Las estructuras propias de los flujos alrededor de la pila se observan en ambas simulaciones realizadas, tanto aguas arriba (vórtice de herradura), como aguas abajo (vórtice de estela) en las Figuras 6.9 y 6.10.

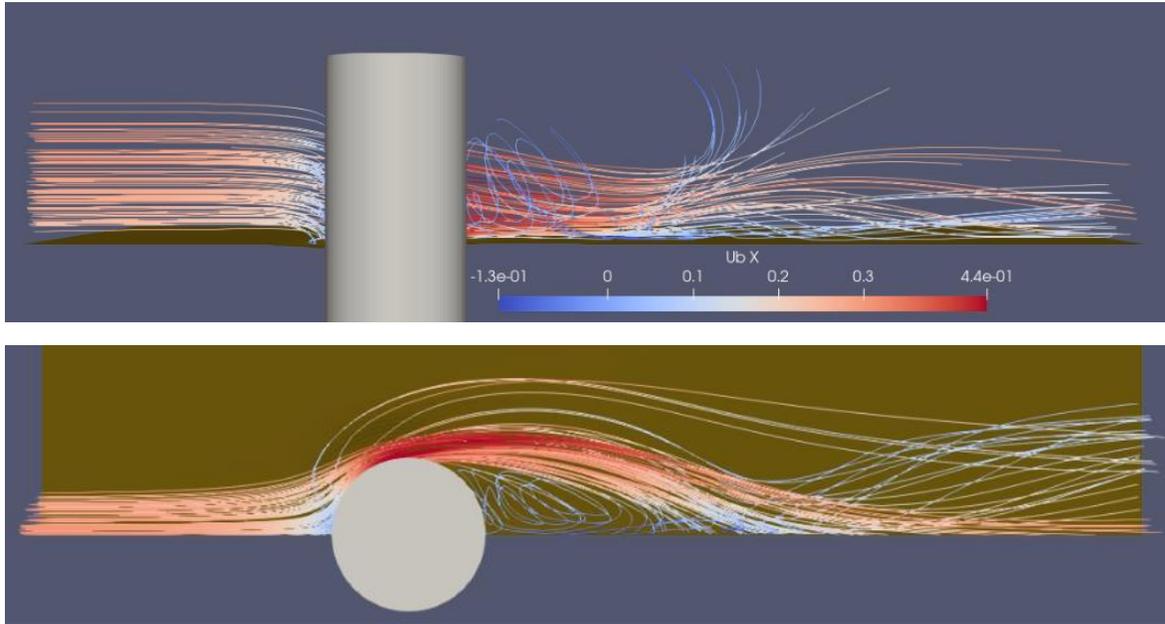


Figura 3.9 Líneas de corriente alrededor de la pila. Caso hidrograma de crecida propuesto por Link et al. (2017). Simulación con SedFoam. $T=150$ s. Los colores de las líneas de corriente dan cuenta de la velocidad (m/s) en el eje x.

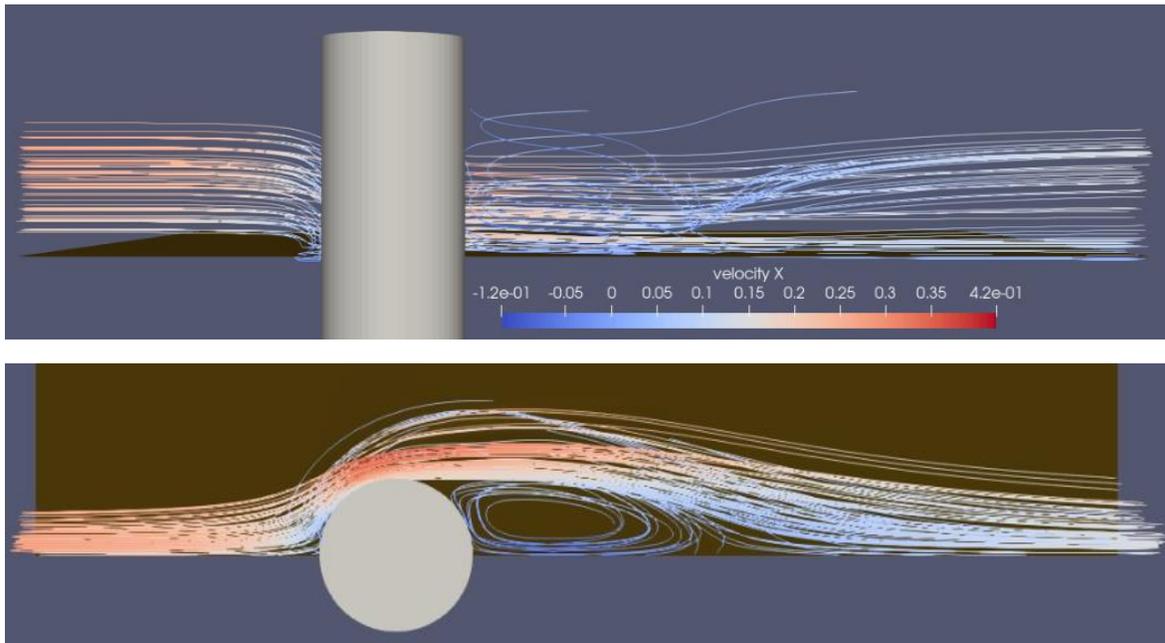


Figura 6.10 Líneas de corriente alrededor de la pila. Caso hidrograma de crecida propuesto por Link et al. (2017). Simulación con REEF3D. $T=150$ s. Los colores de las líneas de corriente dan cuenta de la velocidad (m/s) en el eje x.

7. Discusión

En relación con el caso permanente en estudio, se observa un ajuste con menor diferencia en general en la socavación máxima por parte de la simulación realizada en el presente (2% aguas arriba y 1% aguas debajo de la pila en la socavación final), respecto a la simulación realizada por *Quezada et al. (2018)* (12% aguas arriba y 4% aguas abajo de pila en la socavación máxima final).

La mejora en la solución propuesta en esta investigación, en el sentido de la predicción de la evolución de la socavación máxima en el tiempo para este caso, se observa sobre todo al comienzo de la socavación, donde los resultados obtenidos en todo momento poseen una diferencia menor al presentado por *Quezada et al. (2018)* respecto a los datos experimentales de *Link et al. (2017)*. Además, es importante recalcar, que tras corregir en el código de la versión utilizada de REEF3D (20.02), la forma en que se estiman los esfuerzos de corte (agregar componente de la velocidad vertical) y algunos problemas de referenciación (no se utilizaban datos definidos en el modelo), no fue necesario realizar una disminución en el esfuerzo de corte mediante el coeficiente de relajación utilizado por *Quezada et al. (2018)* en su modelación, en la cual cabe destacar se utilizó una versión anterior de REEF3D.

Por último, en relación con este caso, si bien las estructuras propias de este de flujos se observan aguas arriba de la pila (vórtice de herradura), aguas abajo no se observa claramente la formación de los vórtices de estela, lo cual podría atribuirse al cierre de turbulencia escogido para el caso permanente ($k - \epsilon$). *Mathieu et al. (2019)* señala que la erosión en la zona aguas arriba es probablemente mejor descrita por el modelo $k - \epsilon$ y la socavación en la zona aguas abajo se predice de mejor manera con el modelo $k - \omega$. De hecho, acorde a *Roulund et al. (2005)*, en el caso de flujos alrededor de una pila, el modelo $k - \omega$ es el único modelo de turbulencia capaz de reproducir los vórtices de estela y tener en cuenta el gradiente adverso de presión aguas arriba de la pila.

Para el caso experimental impermanente, se observa un buen ajuste en la socavación máxima en el tiempo por parte de la simulación realizada con REEF3D (2% de diferencia en la relación a los datos experimentales de *Link et al. (2017)*), en términos de la predicción del comienzo del movimiento y la socavación máxima final, pero desde el minuto 3 hasta el minuto 12, existe una sobrestimación considerable en algunos momentos del hidrograma (15% respecto a los datos experimentales).

Por otro lado, con SedFoam solo se pudo reproducir un par de minutos del hidrograma debido a la inestabilidad observada en la interfaz con velocidades altas, pero en estos minutos simulados, se aprecia un buen ajuste en la predicción del comienzo del movimiento de los sedimentos y la forma de la curva de la evolución de la socavación máxima en el tiempo, con un error menor al observado en la modelación con REEF3D. En este caso, mediante la modelación con ambos softwares, se observan claramente las estructuras de flujo características descritas previamente, tanto aguas arriba, como aguas abajo de la pila.

Como se mencionó anteriormente, no se fue capaz de realizar la simulación de todo el hidrograma con el paquete SedFoam del software OpenFoam. Esto debido a las inestabilidades observadas en la zona de la interfaz entre donde existe puramente fluido y la zona donde existe fluido y sedimentos. La cual se manifiesta generando un aumento en la velocidad de los sedimentos en la interfaz, provocando el máximo empaquetamiento posible de los sedimentos justo en la interfaz, generando una presión muy alta debido a la gran compresión de los sedimentos, lo cual genera una divergencia en la velocidad.

Dadas las pruebas realizadas, se atribuye esta inestabilidad a problemas del modelo Euler-Euler con las velocidades altas (en este caso aproximadamente $0.3 \text{ m}^3/\text{s}$, lo cual es casi la velocidad de arrastre incipiente acorde a *Link et al. (2017)*, $0.32 \text{ m}^3/\text{s}$), pues se realizaron pruebas, como por ejemplo seguir corriendo el modelo a una velocidad menor a la cual se comenzó a observar la inestabilidad y no se observó nuevamente esta inestabilidad. Además, se intentó atenuar el gradiente de velocidad modificando el hidrograma, para evaluar si el problema era el gradiente de velocidad, pero llegada la velocidad en la que se observó esta inestabilidad, ocurrió lo mismo, por lo que se piensa que el gradiente de velocidad no es el problema. Otras opciones probadas, fueron refinaciones de malla y variar la cantidad de correctores utilizados en el solucionador PIMPLE, pensando en posibles problemas de no ortogonalidad en la malla, pero no se obtuvieron resultados positivos con ninguna de estas alternativas.

De todas formas, dejando de lado esta inestabilidad, el tiempo computacional necesario para simulaciones con el software OpenFoam con el paquete SedFoam lo hace infactible para una aplicación en el ámbito ingenieril, pues en el caso impermanente demandó alrededor de 50 veces el tiempo empleado en simular con el software REEF3D (alrededor de 20 horas con REEF3D).

A pesar de esto último, sigue siendo interesante ahondar en la investigación de las aplicaciones del paquete de SedFoam que considera las ecuaciones de momentum para sólidos y fluidos, debido a las bondades que brinda respecto a la modelación utilizando un software que considere las ecuaciones de momentum solo para una fase. *Yu et al. (2012)* señala que las fórmulas empíricas para transporte de sedimento utilizadas en los modelos de una fase son deducidas para flujos permanentes, luego su utilización en flujos impermanentes es cuestionable, sumado a la presencia de un obstáculo. Además, la erosión/depositación de sedimento en el lecho y su impacto en la distribución de concentración son ampliamente despreciadas. Esto en contraste al modelo de dos fases, donde las ecuaciones de momentum son las que dictan si existe transporte de sedimento y permiten incluir potencialmente las interacciones fluido-sólido, sólido-sólido y fluido-fluido, teniendo la posibilidad de describir de forma más completa los fenómenos físicos del problema. SedFoam aún tiene como limitante poder describir la superficie libre del flujo, lo cual si es posible representar en REEF3D. En este caso, esto no es muy relevante acorde a la justificación dada por *Baykal et al. (2015)*, donde se señala que la variación de la superficie en torno a la pila para números

de Froude bajos no es tan significativa, por lo que la representación de los fenómenos alrededor de la pila serán representados relativamente bien.

Se propone, como recomendación a futuro, estudiar la forma en que se impone la condición de borde de los sedimentos (concentración), y cómo ir variando esta a medida que el caudal aumenta, pues encontrar una condición de borde para la concentración de sedimentos que no generara inestabilidades casi instantáneas en la interfaz del modelo fue uno de los desafíos más grandes del presente estudio.

8. Conclusiones

Como resultado de la presente investigación, se construyen modelos que simulan la socavación que se desarrolla alrededor de pilas circulares, utilizando dos modelos de cierre de turbulencia en el caso de REEF3D ($k - \epsilon$ y $k - \omega$), y el modelo $k - \omega$ para dos fases con SedFoam.

Tanto utilizando $k - \epsilon$, como $k - \omega$, con REEF3D se logró representar la evolución de la socavación máxima en torno a una pila. El modelo de OpenFoam, no fue capaz de representar totalmente el hidrograma de crecida propuesto, debido a inestabilidades en la zona de la interfaz observada en respuesta a velocidades altas. De todas formas, se evidenció una buena descripción de la evolución de la socavación máxima al comienzo del movimiento de los sedimentos y mientras fue posible modelar (más certera que la simulación con REEF3D, al comparar con los resultados experimentales disponibles).

Independiente de estas inestabilidades, el alto costo computacional de utilizar SedFoam, lo cual fue una de las principales dificultades para generar avances, hace infactible su uso en la industria ingenieril, por lo que, desde este punto de vista, sigue siendo lo más conveniente utilizar modelos con ecuaciones de momentum para una fase, a pesar de la limitación de la representación de fenómenos físicos, y la cuestionabilidad de las fórmulas empíricas utilizadas para describir el transporte de sedimentos en los modelos de una fase, pues los resultados obtenidos sugieren que es posible obtener una socavación máxima con un error aceptable en ingeniería.

Se realizaron modificaciones en el código del software REEF3D, como la incorporación de la componente de velocidad vertical en el cálculo de los esfuerzos de corte y corrección de referenciación de variables, las cuales permitieron prescindir del coeficiente de relajación utilizado por *Quezada et al. (2018)*, y en el caso de *Baykal et al (2015)*, se disminuyó la diferencia de los resultados de socavación respecto a los obtenidos por *Quezada et al. (2018)* al compararlos con los presentados por *Baykal et al (2015)*.

A pesar de lo mencionado en el párrafo anterior, se cree que pueden existir más errores en la programación del código de la versión utilizada de REEF3D, pues los encontrados fueron tras probar muchas simulaciones y notar que no existían cambios en los resultados al cambiar configuraciones, lo cual involucró mucho tiempo perdido, por lo que se recomienda probar otros softwares disponibles de código abierto como TELEMAC-3D (<https://www.opentelemac.org/>), con el cual se han realizado un par de trabajos de socavación en torno a estructuras (*Yiannoukos et al., 2020; Mehrzad, 2021*).

9. Bibliografía

- Aghaee, Y., & Hakimzadeh, H. (2010). Three-dimensional numerical modeling of flow around bridge piers using LES and RANS. In Dittrich, Koll, Aberle Geisenhainer (Eds.), *River Flow*, (pp. 211–218). Karlsruhe: Bundesanstalt für Wasserbau, Germany.
- Ahmad, N., Bihs, H., Kamath, A., & Arntsen Ø, A. (2015). Three-dimensional CFD modeling of wave scour around side-by-side and triangular arrangement of piles with REEF3D. *Procedia Engineering*, 116, 683–690.
- Amini, A., Melville, B. W., Ali, T. M., & Ghazali, A. H. (2012). Clear-water local scour around pile groups in shallow water flow. *Journal of Hydraulic Engineering*, 138(2), 177–185.
- Ataie-Ashtiani, B., & Beheshti, A. A. (2006). Experimental investigation of clear-water local scour at pile groups. *Journal of Hydraulic Engineering*, 132(10), 1100–1104.
- Afzal, M. S., Bihs, H., Kamath, A., & Arntsen Ø, A. (2015). Three-dimensional numerical modeling of pier scour under current and waves using level-set method. *Journal of Offshore Mechanics and Arctic Engineering*, 137(3): 032001.
- Baykal, C., Sumer, B. M., Fuhrman, D. R., Jacobsen, N. G., & Fredsøe, J. (2015). Numerical investigation of flow and scour around a vertical circular cylinder. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 373(2033), 20140104.
- Baykal, C., Sumer, B. M., Fuhrman, D. R., Jacobsen, N. G., & Fredsøe, J. (2017). Numerical simulation of scour and backfilling processes around a circular pile in waves. *Coastal Engineering*, 122, 87–107.
- Berthelsen, P. A. (2004). A decomposed immersed interface method for variable coefficient elliptic equations with nonsmooth and discontinuous solutions. *Journal of Computational Physics*, 197(1), 364–386.
- Bihs, H. (2011). Three-dimensional numerical modeling of local scouring in open channel flow. (PhD thesis), Norwegian University of Science and Technology, Trondheim, Norway.
- Boyer, F., Guazzelli, E. & Pouliquen, O. (2011). Unifying Suspension and Granular Rheology, *Phys. Rev. Lett.*, 107, 188301.
- Burkow, M. (2010). Numerische Simulation stromungsbedingten Sedimenttransports und der entstehenden Gerinnebettformen. (Master's thesis). Bonn, Germany: Mathematisch Naturwissenschaftlichen Fakultät der Rheinischen Friedrich Wilhelms-Universität Bonn.
- Cassar, C., Nicolas, M. & Pouliquen, O. (2005). Submarine granular flows down inclined planes, *Phys. Fluids*, 17, 103301.

- Chang, W., Lai, J., & Yen, C. (2004). Evolution of scour depth at circular bridge piers. *Journal of Hydraulic Engineering*, 130(9), 905–913.
- Chauchat, J., Cheng, Z., Nagel, T., Bonamy, C., & Hsu, T. (2017). SedFoam-2.0: A 3-D two-phase flow numerical model for sediment transport. *Geoscientific Model Development Discussions*, 10(12), 4367-4392.
- Cheng, Z. and Hsu, T.-J. (2014). A Multi-dimensional Two-Phase Eulerian Model for Sediment Transport-TwoPhaseEulerSedFoam, Research report CACR-14-08, Center for Applied Coastal Research – University of Delaware.
- Chen, Q., Qi, M., Zhong, Q., & Li, D. (2017). Experimental study on the multimodal dynamics of the turbulent horseshoe vortex system around a circular cylinder. *Physics of Fluids*, 29(1), 1–12.
- Chorin, A. (1968). Numerical solution of the Navier Stokes equations. *Mathematics of Computation*, 22, 745–762.
- Dey, S. (2001). Experimental studies on incipient motion of sediment particles on generalized sloping fluvial beds. *Journal of Sediment Research*. 16 (3). pp. 391-398.
- Diab, R., Link, O., & Zanke, U. (2010). Geometry of developing and equilibrium scour holes at bridge piers in gravel. *Canadian Journal of Civil Engineering*, 37(4), 544–552.
- Ding, J. & Gidaspow, D. (1990). A bubbling fluidization model using kinetic theory of granular flow, *American Institute of Chemical Engineers Journal*, 36, 523–538.
- Engelund, F., & Fredsøe, J. (1976). A sediment transport model for straight alluvial channels. *Nordic Hydrology*, 7(5), 293–306.
- Fourtakas, G. & Rogers, B. (2016). Modelling multi-phase liquid-sediment scour and resuspension induced by rapid flows using Smoothed Particle Hydrodynamics (SPH) accelerated with a Graphics Processing Unit (GPU). *Advances in Water Resources*. 92(10). 186-199.
- Gjunsburgs, B., Jaudzems, G., & Govša, J. (2010). Hydrograph shape impact on the scour development with time at engineering structures in river flow. *Scientific Journal of Riga Technical, University Construction Science*, 11, 6–12.
- Hager, W. H., & Unger, J. (2010). Bridge pier scour under flood waves. *Journal of Hydraulic Engineering*, 136(10), 842–847.
- Hsu, T., Jenkins, J. T., and Liu, L. F. (2003). On two-phase sediment transport: Dilute flow, *J. Geophys. Res.*, 108, 14.

- Jha, S. K. & Bombardelli, F. A. (2010). Toward two-phase flow modeling of non-dilute sediment transport in open channels, *J. Geophys. Res.*, 115, F03015.
- Jiang, G.-S., & Shu, C.-W. (1996). Efficient implementation of weighted weno schemes. *Journal of Computational Physics*, 126(1), 202–228.
- Kobayashi, T., & Oda, K. (1994). Experimental study on developing process of local scour around a vertical cylinder. *Coastal Engineering Proceedings*, 1(24), 1284–1297.
- Lança, R., Fael, C., Maia, R., Pêgo, J. P., & Cardoso, A. H. (2013). Clear-Water Scour at Pile Groups. *Journal of Hydraulic Engineering*, 139(10), 1089–1098.
- Lauder, B. E., & Spalding, D. B. (1974). The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 3(2), 269–289.
- Link, O. (2006). Untersuchung der Kolkung an einem schlanken zylindrischen Pfeiler in sandigem Boden. (PhD thesis) Inst. für Wasserbau u. Wasserwirtschaft.
- Link, O., Pflieger, F., & Zanke, U. (2008). Characteristics of developing scour-holes at a sand-embedded cylinder. *International Journal of Sediment Research*, 23(3), 258–266.
- Link, O., Castillo, C., Pizarro, A., Rojas, A., Ettmer, B., Escauriaza, C., & Manfreda, S. (2017). A model of bridge pier scour during flood waves. *Journal of Hydraulic Research*, 55(3), 310–323.
- Link, O.; Henríquez, S.; Ettmer, B. (2018). Physical scale modelling of scour around bridge piers. *Journal of Hydraulic Research*. 57, 227–237.
- Liu, X., & García, M. H. (2008). Three-dimensional numerical model with free water surface and mesh deformation for local sediment scour. *Journal of Waterway, Port, Coastal, and Ocean Engineering*. 134:203–217.
- Man, C., Zhang, G.G., Hong, V., Zhou, S. and Feng, Y.L. (2019) Assessment of Turbulence Models on Bridge-Pier Scour Using Flow-3D. *World Journal of Engineering and Technology*, 7, 241-255.
- Mathieu, A.; Chauchat, J.; Bonamy, C.; Nagel, T. (2019) Two-Phase Flow Simulation of Tunnel and Lee-Wake Erosion of Scour below a Submarine Pipeline. *Water*, 11, 1727.
- Mehrzhad, R. (2021). Localized scour around structures under transient flow conditions. Ph.D. Thesis. University of Ottawa, Canada.
- Melville, B. (2008). The physics of local scour at bridge piers. In *Fourth International Conference on Scour and Erosion*. Tokyo, Japan, 28–38.
- Melville, B. W., & Chiew, Y.-M. (1999). Time scale for local scour at bridge piers. *Journal of Hydraulic Engineering*, 125(1), 59–65.

- Meyer-Peter, E., & Müller, R. (1948). Formulas for bed-load transport. In Proceedings, Second Congress IAHR, Stockholm, Sweden, 1948.
- Mohamed, H. I. (2012). Numerical simulation of flow and local scour at two submerged emergent tandem cylindrical piers. *Journal of Engineering Sciences*, 41(1), 1–19.
- Moreno, M., Maia, R., & Couto, L. (2016). Prediction of equilibrium local scour depth at complex bridge piers. *Journal of Hydraulic Engineering*, 142(11), 1–13.
- Nagel, T., Chauchat, J., Cheng, Z., Bonamy, C., Liu, X., Hsu, T., & Bertrand, O. (2017). Two-phase flow simulation of scour around a cylindrical pile, *Coastal Dynamics 2017 Proceedings*.
- Nagel, T. (2018). Numerical Study of Multi-Scale Flow-Sediment-Structure Interactions Using a Multiphase Approach. Ph.D. Thesis, Université de Grenoble-Alpes, Grenoble, France.
- Nagel, T., Chauchat, J., Bonamy, C., Liu, X., Cheng, Z., & Hsu, T. (2020). Three-dimensional scour simulations with a two-phase flow model. *Advances in Water Resources*, 138(C3), 103544.
- Nunez Rattia, J., Percival, J., Neethling, S., & Piggott, M. (2018). Modelling local scour near structures with combined mesh movement and mesh optimisation. *Journal of Computational Physics*, 375, 1220-1237.
- Oliveto, G. & Hager, W. H. (2002). Temporal evolution of clear-water pier and abutment scour. *Journal of Hydraulic Engineering*, 128(9), 811–820.
- Osher, S. & Sethian, J. (1988). Fronts propagating with curvature- dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79, 12–49.
- Paola, C., & Voller, V. (2005). A generalized Exner equation for sediment mass balance. *Journal of Geophysical Research: Earth Surface*, 110(F4), F04014.
- Pope, S., 1988. An explanation of the turbulent round jet/plane jet anomaly. *AIAA Journal* 16 (3), 279–281.
- Quezada, M., Tamburrino, A., & Niño, Y. (2018). Numerical simulation of scour around circular piles due to unsteady currents and oscillatory flows. *Engineering Applications of Computational Fluid Mechanics*, 12(1), 354-374.
- Quezada, M., Tamburrino, A., Niño, Y. (2019). Numerical Study of the Hydrodynamics of Waves and Currents and Their Effects in Pier Scouring. *Water*. 11. 2256.

- Rhie, C. and Chow, W. (1983). Numerical study of the turbulent flow past an air foil with trailing edge separation, *AIAAJ.*, 21, 1525–1532.
- Roulund, A., Sumer, B. M., Fredsøe, J., & Michelsen, J. (2005). Numerical and experimental investigation of flow and scour around a circular pile. *Journal of Fluid Mechanics*, 534, 351–401.
- Rusche, H. (2003). Computational fluid dynamics of dispersed two-phase flows at high phase fractions, PhD thesis, Imperial College London, University of London.
- Saud, M. (2013). 3D numerical modelling of sediment transport under current and waves. (MSc-Thesis) Coastal and Marine Engineering and Management, Norwegian University of Science and Technology.
- Schiller, L. and Naumann, A. (1933). *Über die Grundlegenden Berechnungen bei der Schwerkraftaufbereitung*, Ver. Deut. Ing., 77, 318–320.
- Sheppard, D. M., Odeh, M., & Glasser, T. (2004). Large scale clear-water local pier scour experiments. *Journal of Hydraulic Engineering*, 130(10), 957–963.
- Shu, C. & Gottlieb, S. (1998). Total variation diminishing runge-kutta schemes. *Mathematics of Computation*, 67, 73–85.
- Sumer, B. M., Fredsøe, J., & Christiansen, N. (1992). Scour around vertical pile in waves. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 118(1), 15–31.
- Sumer, B.M.; Christiansen, N.; Fredsøe, J. (1997). The horseshoe vortex and vortex shedding around a vertical wall-mounted cylinder exposed to waves. *Journal of Fluid Mechanics*. 332. 41-70.
- Sumer, B. M., & Fredsøe, J. (2002). *The mechanics of scour in the marine environment*. Jurong East, Singapore: World Scientific Publishing Co. Pte. Ltd.
- Sumer, M. & Serife, S. (2008). Motion of Particles in Fluids. *Solid-Liquid Two Phase Flow*. 245-289.
- Van der Vorst H. (1992). BI-GStab: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13, 631–644.
- Van Rijn, L. C. (1984). Sediment transport, Part I: Bed load transport. *Journal of Hydraulic Engineering*, 110(10), 1431–1456.
- Wang, H., Tang, H., Liu, Q., & Wang, Y. (2016). Local Scouring around Twin Bridge Piers in Open-Channel Flows. *Journal of Hydraulic Engineering*, 142(9).
- Wilcox, D. C. (1998). *Turbulence modeling for CFD, Volume 2*. DCW industries La Canada, CA.

-Wilcox, D. C. (2008). Formulation of the k–w Turbulence Model Revisited, *AIAA J.*, 46, 2823–2838.

-Yagci, O., Yildirim, I., Celik, M. F., Kitsikoudis, V., Duran, Z., & Kirca, V. S. O. (2017). Clear water scour around a finite array of cylinders. *Applied Ocean Research*, 68, 114–129.

-Yalin, M. S. (1972). *Mechanics of Sediment Transport*. New York: Pergamon Press.

-Yiannoukos, I., Benson, T., Van Landeghem, K., Couldrey, A., Whitehouse, R., McCarron, C., Quinn, R., Morgan, J., Roberts, M., & Clayton-Smith, B. (2020). Modelling scour around submerged objects with TELEMAC3D - GAIA. In W. A. Breugem, L. Frederickx, T. Koutrouveli, R. Kulkarni, K. Chu, & B. Decrop (Eds.), *Online proceedings of the papers submitted to the 2020 TELEMAC-MASCARET User Conference* (pp. 68-73). International Marine & Dredging Consultants.

- Yu, X., Hsu, T.-J., Jenkins, J. T., and Liu, P. L.-F. (2012). Predictions of vertical sediment flux in oscillatory flows using a two-phase, sheetflow model, *Adv. Water Resour.*, 48, 2–17.

Anexos

Anexo A: Ficheros del modelo REEF3D

En el presente anexo, se entrega un ejemplo de los archivos utilizados para la definición de las simulaciones del modelo REEF3D, los cuales son: Control.txt, donde se imponen las características de la malla, que se genera con DIVEMesh, y Ctrl.txt, donde se definen las características de la simulación con REEF3D.

Adicionalmente, en el caso de necesitarse hidrogramas de crecida como entrada de la simulación, es necesario incluir un archivo llamado hydrograph.dat con una columna con el tiempo y otra con el caudal asociado.

Control.txt

```
C 11 1 // Tipo de condicion de borde de la malla
C 12 21
C 13 21
C 14 2
C 15 21
C 16 3

B 1 0.005

B 10 0.0 0.8 0.0 0.6 0.0 0.24 // Dimensiones malla

O 33 0.4 0.3 0.02 // Pila

M 10 80 // Numero de nucleos a utilizar
```

Ctrl.txt

```
B 10 1 // Características de las condiciones de borde
B 50 0.0001
B 60 1
B 63 2
B 76 1

D 10 4 // Discretizacion
D 20 2
D 30 1

F 30 3 // Superficie libre
F 40 3
F 50 2
F 53 0.12
F 60 0.2

I 10 1 // Características de las inicialización del modelo
I 20 1500

N 10 14 // Opciones numericas
N 11 11
N 40 3
N 41 140
N 47 0.3

M 10 80 // Numero de núcleos a utilizar

P 10 1 // Opciones de salida de la simulacion
P 35 20 140 2
P 66 1
P 67 0.7
P 75 1
P 79 1
P 180 1

S 10 1 // Características del transporte de sedimentos
S 11 2
S 14 0.3
S 15 0
S 16 2
S 20 0.00017
S 21 3
S 22 2650.0
S 24 0.5
S 25 30.0
S 30 0.048
S 41 2
S 42 1
S 44 10
S 45 20.0
S 50 4
S 57 0.12
S 71 0.32
S 72 0.54
S 80 2
S 90 1
S 91 3
S 93 -2.0

T 10 1 // Modelo de turbulencia

W 10 0.02 // Propiedades del flujo y gravedad
W 22 -9.81
```

Anexo B: Código generación archivo de malla

En el presente anexo se expone el código utilizado para generar el archivo blockMesh, el cual es el input de la malla utilizada en OpenFoam.

```
#!/usr/bin/perl
# Generar blockMeshDict
#
#   16           17           18           19
#   y ^         0 - - - - - 1 - - - - - 2 - - - - - 3
#   |           |           |           |
#   --> x       |           |           |           |           ny1
#               |20        |21        |22        |23
#               4 - - - - - 5 - - - - - 6 - - - - - 7
#
#               |           |32/36 33/37 |           |
#
#               |           |34/38 35/39 |           |           ny2
#
#               |24        |25        |26        |27
#   8 - - - - - 9 - - - - - 10 - - - - - 11
#   |           |           |           |
#   |           |           |           |           ny1
#   |           |           |           |
#   |28        |29        |30        |31
#   12 - - - - - 13 - - - - - 14 - - - - - 15
#
#               nx1           nx2=ny2           nx3
use strict;

if ($#ARGV != 7)
{
    usage();
    exit;
}

my @input = @ARGV;
my $D = $ARGV[0];
my $H0 = $ARGV[1];
my $H1 = $ARGV[2];
my $H2 = $ARGV[3];
my $H3 = $ARGV[4];
my $L1 = $ARGV[5];
my $L2 = $ARGV[6];
my $W = $ARGV[7];

my $w = 0.5*$W;
my $rext = 0.5*$D;
my $r = $rext/sqrt(2.);
my $r0 = $rext/sqrt(2.);
my $rinner1 = $D/8.;
my $rinner = $rinner1/sqrt(2.);
my $xe = $L2-$L1;

# aspect ratio para celdas de bloques diferentes
my $g1 = 1.05710412557;
my $g2 = 1.00394312014;
my $g3 = 1.05710412557;
my $gC = 10;
my $gZ = 100;
my $gZ1 = 0.39576022153092771;
my $gZ1b = 0.29844065970697592;

my $nx1 = 10;
```

```

my $nx2 = 48;
my $nx3 = 33;

my $ny1 = 9;
my $ny2 = $nx2;

my $nz = 48;
my $nz1 = 75;
my $nzlb = 75;

my $gli = 1.0/$g1;
my $g2i = 1.0/$g2;
my $g3i = 1.0/$g3;
my $gCi = 1.0/$gC;
my $gZi = 1.0/$gZ;
my $gZi1 = $gZ1;
my $gZi1b = $gZ1b;

printHeader("2.0");
printConvertToMeters("1.0");
printVertices();
printBlocks();
printEdges();
printPatches();
printMerge();

#####

sub printHeader
{
    print "FoamFile\n";
    print "{\n";
    print "    version\t $_[0];\n";
    print "    format\t ascii;\n";
    print "    class\t dictionary;\n";
    print "    object\t blockMeshDict;\n";
    print "}\n\n";

    print "/*\n";
    print "Automatically generated with $0 @input\n";
    print "*/\n\n";
}

sub printConvertToMeters
{
    print "convertToMeters\t $_[0];\n\n";
}

sub printVertices
{
    print "vertices\n";
    print "( \n";
    print "    ( -$L1 $w $H1 ) //0\n";
    print "    ( -$r $w $H1 ) //1\n";
    print "    ( $r $w $H1 ) //2\n";
    print "    ( $xe $w $H1 ) //3\n";

    print "    ( -$L1 $r $H1 ) //4\n";
    print "    ( -$r $r $H1 ) //5\n";
    print "    ( $r $r $H1 ) //6\n";
    print "    ( $xe $r $H1 ) //7\n";

    print "    ( -$L1 -$r $H1 ) //8\n";
    print "    ( -$r -$r $H1 ) //9\n";
    print "    ( $r -$r $H1 ) //10\n";
    print "    ( $xe -$r $H1 ) //11\n";

    print "    ( -$L1 -$w $H1 ) //12\n";
    print "    ( -$r -$w $H1 ) //13\n";
    print "    ( $r -$w $H1 ) //14\n";
    print "    ( $xe -$w $H1 ) //15\n";
}

```

```

print "    ( -$L1 $w $H2 ) //16\n";
print "    ( -$r $w $H2 ) //17\n";
print "    ( $r $w $H2 ) //18\n";
print "    ( $xe $w $H2 ) //19\n";

print "    ( -$L1 $r $H2 ) //20\n";
print "    ( -$r $r $H2 ) //21\n";
print "    ( $r $r $H2 ) //22\n";
print "    ( $xe $r $H2 ) //23\n";

print "    ( -$L1 -$r $H2 ) //24\n";
print "    ( -$r -$r $H2 ) //25\n";
print "    ( $r -$r $H2 ) //26\n";
print "    ( $xe -$r $H2 ) //27\n";

print "    ( -$L1 -$w $H2 ) //28\n";
print "    ( -$r -$w $H2 ) //29\n";
print "    ( $r -$w $H2 ) //30\n";
print "    ( $xe -$w $H2 ) //31\n";

# vertices en el cilindro
print "    ( -$rinner $rinner $H1 ) //32\n";
print "    ( $rinner $rinner $H1 ) //33\n";
print "    ( -$rinner -$rinner $H1 ) //34\n";
print "    ( $rinner -$rinner $H1 ) //35\n";

print "    ( -$rinner $rinner $H2 ) //36\n";
print "    ( $rinner $rinner $H2 ) //37\n";
print "    ( -$rinner -$rinner $H2 ) //38\n";
print "    ( $rinner -$rinner $H2 ) //39\n";

# Bloques inferiores
print "    ( -$L1 $w $H0 ) //40\n";
print "    ( -$r $w $H0 ) //41\n";
print "    ( $r $w $H0 ) //42\n";
print "    ( $xe $w $H0 ) //43\n";

print "    ( -$L1 $r $H0 ) //44\n";
print "    ( -$r $r $H0 ) //45\n";
print "    ( $r $r $H0 ) //46\n";
print "    ( $xe $r $H0 ) //47\n";

print "    ( -$L1 -$r $H0 ) //48\n";
print "    ( -$r -$r $H0 ) //49\n";
print "    ( $r -$r $H0 ) //45\n";
print "    ( $xe -$r $H0 ) //51\n";

print "    ( -$L1 -$w $H0 ) //52\n";
print "    ( -$r -$w $H0 ) //53\n";
print "    ( $r -$w $H0 ) //54\n";
print "    ( $xe -$w $H0 ) //55\n";

print "    ( -$rinner $rinner $H0 ) //56\n";
print "    ( $rinner $rinner $H0 ) //57\n";
print "    ( -$rinner -$rinner $H0 ) //58\n";
print "    ( $rinner -$rinner $H0 ) //59\n";

# Base del cilindro de socavacion
print "    ( -$r $r $H3 ) //60\n";
print "    ( $r $r $H3 ) //61\n";

print "    ( -$r -$r $H3 ) //62\n";
print "    ( $r -$r $H3 ) //63\n";

print "    ( -$rinner $rinner $H3 ) //64\n";
print "    ( $rinner $rinner $H3 ) //65\n";
print "    ( -$rinner -$rinner $H3 ) //66\n";
print "    ( $rinner -$rinner $H3 ) //67\n";

```

```

    print ");\n\n";
}

sub printBlocks
{
    print "blocks\n";
    print "(\n";
    print "    hex (4 5 1 0 20 21 17 16) ($nx1 $ny1 $nz) simpleGrading ($g1i $g2 $gZ)
//0\n";
    print "    hex (5 6 2 1 21 22 18 17) ($nx2 $ny1 $nz) simpleGrading (1 $g2 $gZ) //1\n";
    print "    hex (6 7 3 2 22 23 19 18) ($nx3 $ny1 $nz) simpleGrading ($g3 $g2 $gZ) //2\n";

    print "    hex (8 9 5 4 24 25 21 20) ($nx1 $ny2 $nz) simpleGrading ($g1i 1 $gZ) //3\n";
    print "    hex (10 11 7 6 26 27 23 22) ($nx3 $ny2 $nz) simpleGrading ($g3 1 $gZ) //4\n";

    print "    hex (12 13 9 8 28 29 25 24) ($nx1 $ny1 $nz) simpleGrading ($g1i $g2i
$gZ) //5\n";
    print "    hex (13 14 10 9 29 30 26 25) ($nx2 $ny1 $nz) simpleGrading (1 $g2i
$gZ) //6\n";
    print "    hex (14 15 11 10 30 31 27 26) ($nx3 $ny1 $nz) simpleGrading ($g3 $g2i
$gZ) //7\n";

    print "    hex (32 33 6 5 36 37 22 21) ($nx2 $ny2 $nz) simpleGrading (1 $gC $gZ) //8\n";
    print "    hex (35 10 6 33 39 26 22 37) ($nx2 $ny2 $nz) simpleGrading ($gC 1 $gZ) //9\n";
    print "    hex (9 10 35 34 25 26 39 38) ($nx2 $ny2 $nz) simpleGrading (1 $gCi
$gZ) //10\n";
    print "    hex (9 34 32 5 25 38 36 21) ($nx2 $ny2 $nz) simpleGrading ($gCi 1
$gZ) //11\n";

    print "    hex (44 45 41 40 4 5 1 0) ($nx1 $ny1 $nz1) simpleGrading ($g1i $g2 $gZ1)
//12\n";
    print "    hex (45 46 42 41 5 6 2 1) ($nx2 $ny1 $nz1) simpleGrading (1 $g2 $gZ1)
//13\n";
    print "    hex (46 47 43 42 6 7 3 2) ($nx3 $ny1 $nz1) simpleGrading ($g3 $g2 $gZ1)
//14\n";

    print "    hex (48 49 45 44 8 9 5 4) ($nx1 $ny2 $nz1) simpleGrading ($g1i 1 $gZ1)
//15\n";
    print "    hex (50 51 47 46 10 11 7 6) ($nx3 $ny2 $nz1) simpleGrading ($g3 1
$gZ1) //16\n";

    print "    hex (52 53 49 48 12 13 9 8) ($nx1 $ny1 $nz1) simpleGrading ($g1i $g2i
$gZ1) //17\n";
    print "    hex (53 54 50 49 13 14 10 9) ($nx2 $ny1 $nz1) simpleGrading (1 $g2i
$gZ1) //18\n";
    print "    hex (54 55 51 50 14 15 11 10) ($nx3 $ny1 $nz1) simpleGrading ($g3 $g2i
$gZ1) //19\n";

    print "    hex (56 57 46 45 32 33 6 5) ($nx2 $ny2 $nz1) simpleGrading (1 $gC
$gZ1) //20\n";
    print "    hex (59 50 46 57 35 10 6 33) ($nx2 $ny2 $nz1) simpleGrading ($gC 1
$gZ1) //21\n";
    print "    hex (49 50 59 58 9 10 35 34) ($nx2 $ny2 $nz1) simpleGrading (1 $gCi
$gZ1) //22\n";
    print "    hex (49 58 56 45 9 34 32 5) ($nx2 $ny2 $nz1) simpleGrading ($gCi 1
$gZ1) //23\n";

    print "    hex (64 65 61 60 56 57 46 45) ($nx2 $ny2 $nz1b) simpleGrading (1 $gC
$gZ1b) //24\n";
    print "    hex (67 63 61 65 59 50 46 57) ($nx2 $ny2 $nz1b) simpleGrading ($gC 1
$gZ1b) //25\n";
    print "    hex (62 63 67 66 49 50 59 58) ($nx2 $ny2 $nz1b) simpleGrading (1 $gCi
$gZ1b) //26\n";
    print "    hex (62 66 64 60 49 58 56 45) ($nx2 $ny2 $nz1b) simpleGrading ($gCi 1
$gZ1b) //27\n";

    print ");\n\n";
}

```

```

sub printEdges
{
  print "edges\n";
  print "(\n";
  print "  arc 32 33 ( 0 $rinner1 $H1 )\n";
  print "  arc 33 35 ( $rinner1 0 $H1 )\n";
  print "  arc 34 35 ( 0 -$rinner1 $H1 )\n";
  print "  arc 34 32 ( -$rinner1 0 $H1 )\n";

  print "  arc 36 37 ( 0 $rinner1 $H2 )\n";
  print "  arc 39 37 ( $rinner1 0 $H2 )\n";
  print "  arc 38 39 ( 0 -$rinner1 $H2 )\n";
  print "  arc 38 36 ( -$rinner1 0 $H2 )\n";

  print "  arc 56 57 ( 0 $rinner1 $H0 )\n";
  print "  arc 59 57 ( $rinner1 0 $H0 )\n";
  print "  arc 58 59 ( 0 -$rinner1 $H0 )\n";
  print "  arc 58 56 ( -$rinner1 0 $H0 )\n";

  print "  arc 64 65 ( 0 $rinner1 $H3 )\n";
  print "  arc 67 65 ( $rinner1 0 $H3 )\n";
  print "  arc 66 67 ( 0 -$rinner1 $H3 )\n";
  print "  arc 66 64 ( -$rinner1 0 $H3 )\n";

  print "  arc 5 6 ( 0 $rext $H1 )\n";
  print "  arc 6 10 ( $rext 0 $H1 )\n";
  print "  arc 9 10 ( 0 -$rext $H1 )\n";
  print "  arc 9 5 ( -$rext 0 $H1 )\n";

  print "  arc 21 22 ( 0 $rext $H2 )\n";
  print "  arc 26 22 ( $rext 0 $H2 )\n";
  print "  arc 25 26 ( 0 -$rext $H2 )\n";
  print "  arc 25 21 ( -$rext 0 $H2 )\n";

  print "  arc 45 46 ( 0 $rext $H0 )\n";
  print "  arc 46 50 ( $rext 0 $H0 )\n";
  print "  arc 49 50 ( 0 -$rext $H0 )\n";
  print "  arc 49 45 ( -$rext 0 $H0 )\n";

  print "  arc 60 61 ( 0 $rext $H3 )\n";
  print "  arc 63 61 ( $rext 0 $H3 )\n";
  print "  arc 62 63 ( 0 -$rext $H3 )\n";
  print "  arc 62 60 ( -$rext 0 $H3 )\n";

  print ");\n\n";
}

sub printPatches
{
  print "patches\n";
  print "(\n";
  print "  wall cylinder\n";
  print "  (\n";
  print "    (32 36 37 33)\n";
  print "    (33 37 39 35)\n";
  print "    (35 39 38 34)\n";
  print "    (34 38 36 32)\n";
  print "    (56 32 33 57)\n";
  print "    (57 33 35 59)\n";
  print "    (59 35 34 58)\n";
  print "    (58 34 32 56)\n";
  print "    (64 56 57 65)\n";
  print "    (65 57 59 67)\n";
  print "    (67 59 58 66)\n";
  print "    (66 58 56 64)\n";
  print "  )\n\n";

  print "  patch inlet\n";
  print "  (\n";
  print "    (4 20 16 0)\n";

```

```

print "      (8 24 20 4)\n";
print "      (12 28 24 8)\n";
print "      (44 4 0 40)\n";
print "      (48 8 4 44)\n";
print "      (52 12 8 48)\n";
print "    )\n";

print "  patch outlet\n";
print "  (\n";
print "    (3 19 23 7)\n";
print "    (7 23 27 11)\n";
print "    (11 27 31 15)\n";
print "    (43 3 7 47)\n";
print "    (47 7 11 51)\n";
print "    (51 11 15 55)\n";
print "  )\n";

print "  wall pit\n";
print "  (\n";
print "    (45 46 61 60)\n";
print "    (62 63 50 49)\n";
print "    (61 46 50 63)\n";
print "    (62 49 45 60)\n";
print "  )\n";

print "  patch frontAndBackPlanes\n";
print "  (\n";
print "    (12 13 29 28)\n";
print "    (13 14 30 29)\n";
print "    (14 15 31 30)\n";
print "    (52 53 13 12)\n";
print "    (53 54 14 13)\n";
print "    (54 55 15 14)\n";
print "    (16 17 1 0)\n";
print "    (17 18 2 1)\n";
print "    (18 19 3 2)\n";
print "    (0 1 41 40)\n";
print "    (1 2 42 41)\n";
print "    (2 3 43 42)\n";
print "  )\n";

print "  wall bottom\n";
print "  (\n";
print "    (40 41 45 44)\n";
print "    (41 42 46 45)\n";
print "    (42 43 47 46)\n";
print "    (44 45 49 48)\n";
print "    (46 47 51 50)\n";
print "    (48 49 53 52)\n";
print "    (49 50 54 53)\n";
print "    (50 51 55 54)\n";
print "    (60 61 65 64)\n";
print "    (61 63 67 65)\n";
print "    (66 67 63 62)\n";
print "    (60 64 66 62)\n";
print "  )\n";

print "  patch top\n";
print "  (\n";
print "    (20 21 17 16)\n";
print "    (21 22 18 17)\n";
print "    (22 23 19 18)\n";
print "    (24 25 21 20)\n";
print "    (26 27 23 22)\n";
print "    (28 29 25 24)\n";
print "    (29 30 26 25)\n";
print "    (30 31 27 26)\n";
print "    (36 37 22 21)\n";
print "    (39 26 22 37)\n";
print "    (25 26 39 38)\n";
print "    (25 38 36 21)\n";

```

```

    print "    )\n";
    print ");\n\n";
}

sub printMerge
{
    print "mergPatchPairs\n";
    print "(\n";
    print ");\n";
}

sub usage
{
    print "usage:\n";
    print "\t $0 <D> <H0> <H1> <H2> <H3> <L1> <L2> <W> <delta>\n";
    print "\t where (recommended value):\n";
    print "\t D = Ancho zona de refinación en torno a la pila. (4)\n";
    print "\t H0 = Elevacion de la base de los sedimentos fuera del cilindro de socavacion.
(-1)\n";
    print "\t H1 = Elevacion de la interfaz fluido-sedimento. (0.)\n";
    print "\t H2 = Elevacion de la zona superior del modelo. (2.)\n";
    print "\t H3 = Elevacion de la base del cilindro de socavacion. (2.)\n";
    print "\t L1 = Largo desde el inlet hasta el centro de la pila. (5.)\n";
    print "\t L2 = Largo del dominio. (10.)\n";
    print "\t W = Ancho del dominio(10)\n";
    print "\t delta = tamaño de celda al lado de la pila. (0.001)\n";
}

```

Anexo C

En el presente anexo, se entrega un ejemplo de los principales archivos utilizados para la definición de las simulaciones de OpenFoam, SedFoam. Las librerías del modelo se separa en 3 carpetas. La carpeta 0, donde se definen las condiciones iniciales y de borde del problema, la carpeta constant, donde se fijan las propiedades de la física del modelo (del fluido, del sedimento, cierre de turbulencia, esfuerzos intergranulares) y la carpeta system, en la cual se detallan la malla, los esquemas numéricos utilizados, las tolerancias y los archivos para dividir la malla y descomponer los archivos para trabajar con varios núcleos en paralelo.

Además, se incluye a continuación, el archivo con el cual se genera la malla y se corre la simulación.

```
#!/bin/sh
# -----SLURM Parameters-----
#SBATCH -J socavacion
#SBATCH -p slims
#SBATCH -n 120
#SBATCH --ntasks-per-node=20
#SBATCH --mem=48000
#SBATCH --mail-user=joaquin.moreno@ing.uchile.cl
#SBATCH --mail-type=ALL
#SBATCH -o socavacioncasopermanente_%j.out
#SBATCH -e socavacioncasopermanente_%j.err

#-----Toolchain-----
# -----Mo³dulos-----

ml OpenFOAM/v2006
source $FOAM_BASH
export LANG=en_US.UTF-8
# -----Comandos-----
foamCleanPolyMesh
perl generateMeshDict_cyl_3dsedim.pl 0.6 -0.025 0.0 0.2 -0.1 0.4 1.2 0.8 >
system/blockMeshDict
blockMesh
topoSet
subsetMesh Simetria -overwrite
createPatch -overwrite
# create the intial time folder
cp -r 0_org 0
#setExprFields Se utiliza solo para imponer los sedimentos en el modelo inicial
decomposePar
# Run sedFoam in parallel
srun -n 120 sedFoam_rbgh -parallel

reconstructPar
rm -r *process*
```

Carpeta 0

Alpha

```
/*-----* C++ *-----*\
|=====|
| \\ / Field | OpenFOAM: The Open Source CFD Toolbox |
| \\ / Operation | Version: 2.4.0 |
| \\ / A nd | Web: www.OpenFOAM.org |
| \\ / M anipulation |
|-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    location     "0";
    object       alpha_a;
}
// *****
dimensions      [0 0 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type          codedFixedValue;
        value          uniform 0;
        name           alphainlet;
        code
        #{
            const fvPatch& boundaryPatch = patch();
            const vectorField& Cf = boundaryPatch.Cf();
            scalarField& field =*this;
            forAll(boundaryPatch, faceI)
            {
                scalar z = Cf[faceI].z();
                field[faceI] = 0.6128*0.5*(1.0+tanh((0.001-Cf[faceI].z())*5000))+1e-
100;
            }
        #};
    }
    cylinder
    {
        type          zeroGradient;
    }
    pit
    {
        type          zeroGradient;
    }
    outlet
    {
        type          zeroGradient;
    }
    top
    {
        type          zeroGradient;
    }
    bottom
    {
        type          zeroGradient;
    }
}
```

```

    frontAndBackPlanes
    {
        type          zeroGradient;
    }
    Mitad
    {
        type          symmetryPlane;
    }
}
// ***** //

```

k

```

/*-----* C++ -*-----*\
|=====|
|  \ \ /  /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
|  \ \ /  /  O p e r a t i o n  | Version:  2.4.0                    |
|  \ \ /  /  A n d                | Web:      www.OpenFOAM.org       |
|  \ \ /  /  M a n i p u l a t i o n  |                               |
\*-----*
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    location     "0";
    object       k;
}
// ***** //

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 1e-12;

boundaryField
{
    inlet
    {
        type          codedFixedValue;
        value          uniform 1e-6;
        redirectType   inletk;
        code
        #{
            const fvPatch& boundaryPatch = patch();
            const vectorField& Cf = boundaryPatch.Cf();
            scalarField& field = *this;
            scalar t = this->db().time().value();
            forAll(Cf, faceI)
            {
                if (t <= 60)
                {
                    field[faceI] = (0.003809523809524*t +
2.457142857142860)*(0.003809523809524*t + 2.457142857142860)*1.166e-5;
                }
            }
        }
    }
    cylinder
    {
        type          fixedValue;
        value          uniform 1e-12;
    }
    pit
    {
        type          fixedValue;
        value          uniform 1e-12;
    }
}

```

```

    }
    outlet
    {
        type            zeroGradient;
    }
    top
    {
        type            zeroGradient;
    }
    bottom
    {
        type            kqRWallFunction;
        value           uniform 1e-12;
    }
    frontAndBackPlanes
    {
        type            zeroGradient;
    }
    Mitad
    {
        type            symmetryPlane;
    }
}

// ***** //

```

omega

```

/*----- C++ -----*\
| ===== |
| \ \ / / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / / O p e r a t i o n : | Version: 2.1.0 |
| \ \ / / A n d | Web: www.OpenFOAM.org |
| \ \ / / M a n i p u l a t i o n | |
| \ \ / / | |
/*-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    location     "0";
    object       omega;
}
// ***** //

dimensions      [0 0 -1 0 0 0 0];

internalField   uniform 10;

boundaryField
{
    inlet
    {
        type            codedFixedValue;
        value           uniform 10;
        redirectType    inletw;
        code
        #{
            const fvPatch& boundaryPatch = patch();
            const vectorField& Cf = boundaryPatch.Cf();
            scalarField& field = *this;
            scalar t = this->db().time().value();
            forAll(Cf, faceI)
            {
                if (Cf[faceI].z() >= 0.003)
                {

```

```

                if (t <= 60)
                {
                    field[faceI] = 100*pow((0.003809523809524*t +
2.457142857142860)*(0.003809523809524*t + 2.457142857142860)*1.166e-5, 0.5)/(0.09*0.1*0.2);
                }
                else
                {
                    field[faceI] = 1000;
                }
            }
        #};
    }
    cylinder
    {
        type            omegaWallFunction;
        value            uniform 10;
    }
    pit
    {
        type            zeroGradient;
    }
    outlet
    {
        type            zeroGradient;
    }
    top
    {
        type            zeroGradient;
    }
    bottom
    {
        type            zeroGradient;
    }
    frontAndBackPlanes
    {
        type            zeroGradient;
    }
    Mitad
    {
        type            symmetryPlane;
    }
}

// ***** //

```

Ub

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / Field | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O peration | Version: 2.4.0 |
| \\ / A nd | Web: www.OpenFOAM.org |
| \\ / M anipulation |
|-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    location     "0";
    object       Ub;
}
// ***** //

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    inlet
    {
        type          codedFixedValue;
        value          uniform (0 0 0);

        redirectType  inletUa;

        code
        #{
            const fvPatch& boundaryPatch = patch();
            const vectorField& Cf = boundaryPatch.Cf();
            vectorField& field = *this;
            scalar t = this->db().time().value();
            forAll(Cf, faceI)
            {
                if (Cf[faceI].z() >= 0.002)
                {
                    if (t <= 60)
                    {
                        field[faceI] = (0.003809523809524*t + 2.457142857142860)
                            *vector
                            (
                                (0.0061204/0.41)
                                *log(30*(Cf[faceI].z()+0.002)
                                    /(9e-4)),
                                0,
                                0
                            );
                    }
                }
                else
                {
                    field[faceI] = vector(0,0,0);
                }
            }
        }
    };

    cylinder
    {
        type          fixedValue;
        value          uniform (0 0 0);
    }
}
}
```

```

    pit
    {
        type          fixedValue;
        value          uniform (0 0 0);
    }
    outlet
    {
        type          inletOutlet;
        inletValue     uniform (0 0 0);
        value          uniform (0 0 0);
    }
    top
    {
        type          zeroGradient;
    }
    bottom
    {
        type          fixedValue;
        value          uniform (0 0 0);
    }
    frontAndBackPlanes
    {
        type          zeroGradient;
    }
    Mitad
    {
        type          symmetryPlane;
    }
}

// *****

```

Ua

```

/*-----* C++ *-----*\
|=====|
| \\ / | F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / | O p e r a t i o n | Version: 2.4.0 |
| \\ / | A n d | Web: www.OpenFOAM.org |
| \\ / | M a n i p u l a t i o n | |
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    location     "0";
    object       Ua;
}
// *****

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    inlet
    {
        type          codedFixedValue;
        value          uniform (0 0 0);

        redirectType  inletUa;

        code
        #{

```

```

const fvPatch& boundaryPatch = patch();
const vectorField& Cf = boundaryPatch.Cf();
vectorField& field = *this;
scalar t = this->db().time().value();
forAll(Cf, faceI)
{
    if (Cf[faceI].z() >= 0.002)
    {
        if (t <= 60)
        {
            field[faceI] = (0.003809523809524*t + 2.457142857142860)
                *vector
                (
                    (0.0061204/0.41)
                    *log(30*(Cf[faceI].z()+0.002)
                        /(9e-4)),
                    0,
                    0
                );
        }
        else
        {
            field[faceI] = vector(0,0,0);
        }
    }
};

}
cylinder
{
    type          fixedValue;
    value         uniform (0 0 0);
}
pit
{
    type          fixedValue;
    value         uniform (0 0 0);
}
outlet
{
    type          inletOutlet;
    inletValue    uniform (0 0 0);
    value         uniform (0 0 0);
}
top
{
    type          zeroGradient;
}
bottom
{
    type          fixedValue;
    value         uniform (0 0 0);
}
frontAndBackPlanes
{
    type          zeroGradient;
}
Mitad
{
    type          symmetryPlane;
}
}

// ***** //

```

nut

```
/*-----* C++ -*-----*\
|=====|
| \\ / | F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / | O p e r a t i o n | Version: 2.4.0 |
| \\ / | A n d | Web: www.OpenFOAM.org |
| \\ / | M a n i p u l a t i o n | |
|-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    location     "0";
    object       nut;
}
// ***** //

dimensions      [0 2 -1 0 0 0 0];

internalField   uniform 0.;

boundaryField
{
    inlet
    {
        type      calculated;
        value     uniform 0;
    }
    cylinder
    {
        type      nutkWallFunction;
        value     uniform 0;
    }
    pit
    {
        type      calculated;
        value     uniform 0;
    }
    outlet
    {
        type      calculated;
        value     uniform 0;
    }
    top
    {
        type      zeroGradient;
    }
    bottom
    {
        type      calculated;
        value     uniform 0;
    }
    frontAndBackPlanes
    {
        type      zeroGradient;
    }
    Mitad
    {
        type      symmetryPlane;
    }
}

// ***** //
```

p_rbg

```
/*-----* C++ -*-----*\
|=====|
| \\ / | F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / | O p e r a t i o n | Version: 2.1.0 |
| \\ / | A n d | Web: www.OpenFOAM.org |
| \\ / | M a n i p u l a t i o n | |
|-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    location     "0";
    object       p_rbg;
}
// ***** //

dimensions      [1 -1 -2 0 0 0];

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type      zeroGradient;
    }
    cylinder
    {
        type      zeroGradient;
    }
    pit
    {
        type      zeroGradient;
    }
    outlet
    {
        type      fixedValue;
        value     uniform 0;
    }
    top
    {
        type      zeroGradient;
    }
    bottom
    {
        type      zeroGradient;
    }
    frontAndBackPlanes
    {
        type      zeroGradient;
    }
    Mitad
    {
        type      symmetryPlane;
    }
}

// ***** //
```

Carpeta constant

granularReologyProperties

En este archivo se active el modelo de reología de flujos granulares densos y se definen los parámetros asociados.

```
/*
| ===== |
| \\      / | F i e l d           | OpenFOAM: The Open Source CFD Toolbox |
| \\      / | O p e r a t i o n   | Version: 1.7.1                     |
| \\      / | A n d                | Web:      www.OpenFOAM.com          |
| \\      / | M a n i p u l a t i o n |                                     |
|-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       granularRheologyProperties;
}
// ***** //

granularRheology    on;

granularDilatancy   off;

alphaMaxG           alphaMaxG [ 0 0 0 0 0 0 0 ] 0.625;

mus                 mus [ 0 0 0 0 0 0 0 ] 0.63;

mu2                 mu2 [ 0 0 0 0 0 0 0 ] 1.13;

I0                  I0 [ 0 0 0 0 0 0 0 ] 0.6;

Bphi                Bphi [ 0 0 0 0 0 0 0 ] 0.66;

n                   n [ 0 0 0 0 0 0 0 ] 2.5;

relaxPa             relaxPa [ 0 0 0 0 0 0 0 ] 5e-5;

FrictionModel      MuI;

PPressureModel     MuI;

FluidViscosityModel BoyerEtAl;

DilatancyModel     none;

// ***** //
```

interfacialProperties

En el presente archivo se define el modelo de arrastre a utilizar.

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / Field | OpenFOAM: The Open Source CFD Toolbox |
| \\ / Operation | Version: 1.7.1 |
| \\ / A nd | Web: www.OpenFOAM.com |
| \\ / M anipulation |
|-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       interfacialProperties;
}
// ***** //
dragModela     GidaspowSchillerNaumann;

dragModelb     GidaspowSchillerNaumann;

dragPhase      a;
// ***** //
```

ppProperties

Se define el modelo de los esfuerzos granulares y algunos parámetros asociados.

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / Field | OpenFOAM: The Open Source CFD Toolbox |
| \\ / Operation | Version: 1.7.1 |
| \\ / A nd | Web: www.OpenFOAM.com |
| \\ / M anipulation |
|-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       ppProperties;
}
// ***** //
// parameters for pff

ppModel        JohnsonJackson;

alphaMax       alphaMax [ 0 0 0 0 0 0 ] 0.635;

alphaMinFriction alphaMinFriction [ 0 0 0 0 0 0 ] 0.57;

Fr             Fr [ 1 -1 -2 0 0 0 ] 5e-2;

eta0           eta0 [ 0 0 0 0 0 0 ] 3;

eta1           eta1 [ 0 0 0 0 0 0 ] 5;

packingLimiter no;
// ***** //
```

transportProperties

En el presente se fijan las propiedades de la fase sólida y líquida.

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / | F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / | O p e r a t i o n | Version: 1.7.1 |
| \\ / | A n d | Web: www.OpenFOAM.com |
| \\ / | M a n i p u l a t i o n | |
| ===== |
\*-----*- C++ -*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       transportProperties;
}
// * * * * * sediment properties * * * * * //
phasea
{
    rho          rho [ 1 -3 0 0 0 ] 2650;
    nu           nu [ 0 2 -1 0 0 ] 1e-6;
    d            d [ 0 1 0 0 0 0 0 ] 0.36e-3;
    sF           sF [ 0 0 0 0 0 0 0 ] 1.; // shape Factor to adjust settling
    velocity for non-spherical particles
    hExp         hExp [ 0 0 0 0 0 0 0 ] 2.65; // hindrance exponent for drag
}
phaseb
{
    rho          rho [ 1 -3 0 0 0 ] 1000;
    nu           nu [ 0 2 -1 0 0 ] 1.e-06;
    d            d [ 0 1 0 0 0 0 0 ] 10e-6;
    sF           sF [ 0 0 0 0 0 0 0 ] 1.;
    hExp         hExp [ 0 0 0 0 0 0 0 ] 2.65;
}
//***** //
alphaSmall     alphaSmall [ 0 0 0 0 0 0 0 ] 1e-4; // minimum volume fraction (phase a) for
division by alpha
transportModel Newtonian;
nu             nu [ 0 2 -1 0 0 0 0 ] 1e-06;
nuMax          nuMax [0 2 -1 0 0 0 0] 10; // viscosity limiter for the Frictional model
(required for stability)
// * * * * * //
```

turbulenceProperties

En el archivo presentado se define el cierre de turbulencia utilizado.

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / | F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / | O p e r a t i o n | Version: 2.4.0 |
| \\ / | A n d | Web: www.OpenFOAM.org |
| \\ / | M a n i p u l a t i o n | |
| ===== |
\*-----*- C++ -*-----*/
FoamFile
{
    version      2.0;
```

```

    format      ascii;
    class       dictionary;
    location    "constant";
    object      turbulenceProperties;
}
// * * * * *

simulationType RAS;
RAS{
RASModel twophasekOmega2006;
printCoeffs  on;
turbulence   on;
}

// *****

```

twophaseRASProperties

Se definen los coeficientes y parámetros del modelo de turbulencia utilizado.

```

/*-----* C++ -*-----*\
|=====|
| \\ / | F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / | O p e r a t i o n | Version: 1.7.1 |
| \\ / | A n d | Web: www.OpenFOAM.com |
| \\ / | M a n i p u l a t i o n |
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       twophaseRASProperties;
}
// * * * * *

// * * * * * definition of turbulence coefficient and others * * * * * //
SUS          SUS [ 0 0 0 0 0 0 0 ] 1.; //set to 1 to include turbulent suspension term
C3om         C3om [ 0 0 0 0 0 0 0 ] 0.35; //
C4om         C4om [ 0 0 0 0 0 0 0 ] 1;
C3ep         C3ep [ 0 0 0 0 0 0 0 ] 1.2; //
C4ep         C4ep [ 0 0 0 0 0 0 0 ] 1;
KE1          KE1 [ 0 0 0 0 0 0 0 ] 0; //density strata (Uf-Ug)
KE2          KE2 [ 0 0 0 0 0 0 0 ] 1; //turb modulation
KE3          KE3 [ 0 0 0 0 0 0 0 ] 0; //turb generation
KE4          KE4 [ 0 0 0 0 0 0 0 ] 1; //density stratification g
B            B [ 0 0 0 0 0 0 0 ] 1; //turb modulation coeff
Cmu          Cmu [ 0 0 0 0 0 0 0 ] 0.09;
Tpsmall      Tpsmall [1 -3 -1 0 0 0 0] 1e-6;
kSmall       kSmall [0 2 -2 0 0 0 0] 1e-20;
nutMax       nutMax [0 2 -1 0 0 0 0] 5e-3;

// *****

```

Carpeta system

blockMeshdict

Se define la malla utilizada.

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / Field | OpenFOAM: The Open Source CFD Toolbox |
| \\ / Operation | Version: 1.7.1 |
| \\ / And | Web: www.OpenFOAM.com |
| \\ / Manipulation |
\*-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       blockMeshDict;
}

/*
Automatically generated with generateMeshDict_cyl_3dsedim.pl 0.6 -0.025 0.0 0.2 -0.1 0.4 1.2
0.8
*/

convertToMeters      1.0;

vertices
(
    ( -0.4  0.4 0.0 ) //0
    ( -0.212132034355964  0.4 0.0 ) //1
    (  0.212132034355964  0.4 0.0 ) //2
    (  0.8  0.4 0.0 ) //3
    ( -0.4  0.212132034355964 0.0 ) //4
    ( -0.212132034355964  0.212132034355964 0.0 ) //5
    (  0.212132034355964  0.212132034355964 0.0 ) //6
    (  0.8  0.212132034355964 0.0 ) //7
    ( -0.4  -0.212132034355964 0.0 ) //8
    ( -0.212132034355964  -0.212132034355964 0.0 ) //9
    (  0.212132034355964  -0.212132034355964 0.0 ) //10
    (  0.8  -0.212132034355964 0.0 ) //11
    ( -0.4  -0.4 0.0 ) //12
    ( -0.212132034355964  -0.4 0.0 ) //13
    (  0.212132034355964  -0.4 0.0 ) //14
    (  0.8  -0.4 0.0 ) //15
    ( -0.4  0.4 0.2 ) //16
    ( -0.212132034355964  0.4 0.2 ) //17
    (  0.212132034355964  0.4 0.2 ) //18
    (  0.8  0.4 0.2 ) //19
    ( -0.4  0.212132034355964 0.2 ) //20
    ( -0.212132034355964  0.212132034355964 0.2 ) //21
    (  0.212132034355964  0.212132034355964 0.2 ) //22
    (  0.8  0.212132034355964 0.2 ) //23
    ( -0.4  -0.212132034355964 0.2 ) //24
    ( -0.212132034355964  -0.212132034355964 0.2 ) //25
    (  0.212132034355964  -0.212132034355964 0.2 ) //26
    (  0.8  -0.212132034355964 0.2 ) //27
    ( -0.4  -0.4 0.2 ) //28
    ( -0.212132034355964  -0.4 0.2 ) //29
    (  0.212132034355964  -0.4 0.2 ) //30
    (  0.8  -0.4 0.2 ) //31
    ( -0.0530330085889911  0.0530330085889911 0.0 ) //32
    (  0.0530330085889911  0.0530330085889911 0.0 ) //33
    ( -0.0530330085889911  -0.0530330085889911 0.0 ) //34
)
```

```

( 0.0530330085889911 -0.0530330085889911 0.0 ) //35
( -0.0530330085889911 0.0530330085889911 0.2 ) //36
( 0.0530330085889911 0.0530330085889911 0.2 ) //37
( -0.0530330085889911 -0.0530330085889911 0.2 ) //38
( 0.0530330085889911 -0.0530330085889911 0.2 ) //39
( -0.4 0.4 -0.025 ) //40
( -0.212132034355964 0.4 -0.025 ) //41
( 0.212132034355964 0.4 -0.025 ) //42
( 0.8 0.4 -0.025 ) //43
( -0.4 0.212132034355964 -0.025 ) //44
( -0.212132034355964 0.212132034355964 -0.025 ) //45
( 0.212132034355964 0.212132034355964 -0.025 ) //46
( 0.8 0.212132034355964 -0.025 ) //47
( -0.4 -0.212132034355964 -0.025 ) //48
( -0.212132034355964 -0.212132034355964 -0.025 ) //49
( 0.212132034355964 -0.212132034355964 -0.025 ) //45
( 0.8 -0.212132034355964 -0.025 ) //51
( -0.4 -0.4 -0.025 ) //52
( -0.212132034355964 -0.4 -0.025 ) //53
( 0.212132034355964 -0.4 -0.025 ) //54
( 0.8 -0.4 -0.025 ) //55
( -0.0530330085889911 0.0530330085889911 -0.025 ) //56
( 0.0530330085889911 0.0530330085889911 -0.025 ) //57
( -0.0530330085889911 -0.0530330085889911 -0.025 ) //58
( 0.0530330085889911 -0.0530330085889911 -0.025 ) //59
( -0.212132034355964 0.212132034355964 -0.1 ) //60
( 0.212132034355964 0.212132034355964 -0.1 ) //61
( -0.212132034355964 -0.212132034355964 -0.1 ) //62
( 0.212132034355964 -0.212132034355964 -0.1 ) //63
( -0.0530330085889911 0.0530330085889911 -0.1 ) //64
( 0.0530330085889911 0.0530330085889911 -0.1 ) //65
( -0.0530330085889911 -0.0530330085889911 -0.1 ) //66
( 0.0530330085889911 -0.0530330085889911 -0.1 ) //67
);

blocks
(
  hex (4 5 1 0 20 21 17 16) (10 9 48) simpleGrading (0.945980604759054 1.00394312014 100)
//0
  hex (5 6 2 1 21 22 18 17) (48 9 48) simpleGrading (1 1.00394312014 100) //1
  hex (6 7 3 2 22 23 19 18) (33 9 48) simpleGrading (1.05710412557 1.00394312014 100) //2
  hex (8 9 5 4 24 25 21 20) (10 48 48) simpleGrading (0.945980604759054 1 100) //3
  hex (10 11 7 6 26 27 23 22) (33 48 48) simpleGrading (1.05710412557 1 100) //4
  hex (12 13 9 8 28 29 25 24) (10 9 48) simpleGrading (0.945980604759054 0.996072366988829
100) //5
  hex (13 14 10 9 29 30 26 25) (48 9 48) simpleGrading (1 0.996072366988829 100) //6
  hex (14 15 11 10 30 31 27 26) (33 9 48) simpleGrading (1.05710412557 0.996072366988829
100) //7
  hex (32 33 6 5 36 37 22 21) (48 48 48) simpleGrading (1 10 100) //8
  hex (35 10 6 33 39 26 22 37) (48 48 48) simpleGrading (10 1 100) //9
  hex (9 10 35 34 25 26 39 38) (48 48 48) simpleGrading (1 0.1 100) //10
  hex (9 34 32 5 25 38 36 21) (48 48 48) simpleGrading (0.1 1 100) //11
  hex (44 45 41 40 4 5 1 0) (10 9 75) simpleGrading (0.945980604759054 1.00394312014
0.395760221530928) //12
  hex (45 46 42 41 5 6 2 1) (48 9 75) simpleGrading (1 1.00394312014 0.395760221530928)
//13
  hex (46 47 43 42 6 7 3 2) (33 9 75) simpleGrading (1.05710412557 1.00394312014
0.395760221530928) //14
  hex (48 49 45 44 8 9 5 4) (10 48 75) simpleGrading (0.945980604759054 1
0.395760221530928) //15
  hex (50 51 47 46 10 11 7 6) (33 48 75) simpleGrading (1.05710412557 1
0.395760221530928) //16
  hex (52 53 49 48 12 13 9 8) (10 9 75) simpleGrading (0.945980604759054 0.996072366988829
0.395760221530928) //17
  hex (53 54 50 49 13 14 10 9) (48 9 75) simpleGrading (1 0.996072366988829
0.395760221530928) //18
  hex (54 55 51 50 14 15 11 10) (33 9 75) simpleGrading (1.05710412557 0.996072366988829
0.395760221530928) //19
  hex (56 57 46 45 32 33 6 5) (48 48 75) simpleGrading (1 10 0.395760221530928) //20
  hex (59 50 46 57 35 10 6 33) (48 48 75) simpleGrading (10 1 0.395760221530928) //21
  hex (49 50 59 58 9 10 35 34) (48 48 75) simpleGrading (1 0.1 0.395760221530928) //22

```

```

hex (49 58 56 45 9 34 32 5) (48 48 75) simpleGrading (0.1 1 0.395760221530928)//23
hex (64 65 61 60 56 57 46 45) (48 48 75) simpleGrading (1 10 0.298440659706976)//24
hex (67 63 61 65 59 50 46 57) (48 48 75) simpleGrading (10 1 0.298440659706976)//25
hex (62 63 67 66 49 50 59 58) (48 48 75) simpleGrading (1 0.1 0.298440659706976)//26
hex (62 66 64 60 49 58 56 45) (48 48 75) simpleGrading (0.1 1 0.298440659706976)//27
);

edges
(
  arc 32 33 ( 0 0.075 0.0 )
  arc 33 35 ( 0.075 0 0.0 )
  arc 34 35 ( 0 -0.075 0.0 )
  arc 34 32 ( -0.075 0 0.0 )
  arc 36 37 ( 0 0.075 0.2 )
  arc 39 37 ( 0.075 0 0.2 )
  arc 38 39 ( 0 -0.075 0.2 )
  arc 38 36 ( -0.075 0 0.2 )
  arc 56 57 ( 0 0.075 -0.025 )
  arc 59 57 ( 0.075 0 -0.025 )
  arc 58 59 ( 0 -0.075 -0.025 )
  arc 58 56 ( -0.075 0 -0.025 )
  arc 64 65 ( 0 0.075 -0.1 )
  arc 67 65 ( 0.075 0 -0.1 )
  arc 66 67 ( 0 -0.075 -0.1 )
  arc 66 64 ( -0.075 0 -0.1 )
  arc 5 6 ( 0 0.3 0.0 )
  arc 6 10 ( 0.3 0 0.0 )
  arc 9 10 ( 0 -0.3 0.0 )
  arc 9 5 ( -0.3 0 0.0 )
  arc 21 22 ( 0 0.3 0.2 )
  arc 26 22 ( 0.3 0 0.2 )
  arc 25 26 ( 0 -0.3 0.2 )
  arc 25 21 ( -0.3 0 0.2 )
  arc 45 46 ( 0 0.3 -0.025 )
  arc 46 50 ( 0.3 0 -0.025 )
  arc 49 50 ( 0 -0.3 -0.025 )
  arc 49 45 ( -0.3 0 -0.025 )
  arc 60 61 ( 0 0.3 -0.1 )
  arc 63 61 ( 0.3 0 -0.1 )
  arc 62 63 ( 0 -0.3 -0.1 )
  arc 62 60 ( -0.3 0 -0.1 )
);

patches
(
  wall cylinder
  (
    (32 36 37 33)
    (33 37 39 35)
    (35 39 38 34)
    (34 38 36 32)
    (56 32 33 57)
    (57 33 35 59)
    (59 35 34 58)
    (58 34 32 56)
    (64 56 57 65)
    (65 57 59 67)
    (67 59 58 66)
    (66 58 56 64)
  )

  patch inlet
  (
    (4 20 16 0)
    (8 24 20 4)
    (12 28 24 8)
    (44 4 0 40)
    (48 8 4 44)
    (52 12 8 48)
  )

  patch outlet

```

```

(
  (3 19 23 7)
  (7 23 27 11)
  (11 27 31 15)
  (43 3 7 47)
  (47 7 11 51)
  (51 11 15 55)
)
wall pit
(
  (45 46 61 60)
  (62 63 50 49)
  (61 46 50 63)
  (62 49 45 60)
)
patch frontAndBackPlanes
(
  (12 13 29 28)
  (13 14 30 29)
  (14 15 31 30)
  (52 53 13 12)
  (53 54 14 13)
  (54 55 15 14)
  (16 17 1 0)
  (17 18 2 1)
  (18 19 3 2)
  (0 1 41 40)
  (1 2 42 41)
  (2 3 43 42)
)
wall bottom
(
  (40 41 45 44)
  (41 42 46 45)
  (42 43 47 46)
  (44 45 49 48)
  (46 47 51 50)
  (48 49 53 52)
  (49 50 54 53)
  (50 51 55 54)
  (60 61 65 64)
  (61 63 67 65)
  (66 67 63 62)
  (60 64 66 62)
)
patch top
(
  (20 21 17 16)
  (21 22 18 17)
  (22 23 19 18)
  (24 25 21 20)
  (26 27 23 22)
  (28 29 25 24)
  (29 30 26 25)
  (30 31 27 26)
  (36 37 22 21)
  (39 26 22 37)
  (25 26 39 38)
  (25 38 36 21)
)
);

mergPatchPairs
(
);
// ***** //

```

controlDict

Se definen los parámetros principales de la simulación, como el tiempo de simulación, intervalos de tiempo, criterio de Courant-Friedrichs-Lewy y cada cuanto se guardan resultados.

```
/*-----* C++ -*-----*\
|=====|
|  \ \ /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
|  \ \ /  O p e r a t i o n | Version: 1.7.1 |
|  \ \ /  A n d | Web: www.OpenFOAM.com |
|  \ \ /  M a n i p u l a t i o n |
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       controlDict;
}
// ***** //

application      sedFoam;

startFrom        latestTime;

startTime        0;

stopAt           endTime;

endTime          30;

deltaT           5e-4;

writeControl     adjustableRunTime;

writeInterval    3;

purgeWrite       0;

writeFormat      binary;

writePrecision   6;

writeCompression uncompressed;

timeFormat       general;

timePrecision    6;

runTimeModifiable on;

adjustTimeStep   yes;

maxCo            0.3;

maxAlphaCo       0.3;

maxDeltaT        5e-4;

// ***** //
```

createPatchDict

Convierte en un parche una zona definida, y se le asigna un nombre y tipo de parche para poder asignarle condiciones de borde. Este es utilizado tras la división de la malla.

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 2.3.1 |
| \\ / A n d | Web: www.OpenFOAM.org |
| \\ / M a n i p u l a t i o n |
|-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       createPatchDict;
}

// * * * * *

pointSync false;

patches
(
    {
        // Name of new patch
        name Mitad;

        // Type of new patch
        patchInfo
        {
            type symmetryPlane;
        }
        patches ("oldInternalFaces");
    }
);

// * * * * *
```

decomposeParDict

Se define como se descompone y en cuantas partes el problema para el paralelizado.

```
/*-----*- C++ -*-----*\
| ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \\ / O p e r a t i o n | Version: 1.7.1 |
| \\ / A n d | Web: www.OpenFOAM.com |
| \\ / M a n i p u l a t i o n |
|-----*\
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
}
```

```

    object      decomposeParDict;
}
// * * * * *

numberOfSubdomains 120;

//method      simple;
method        scotch;

simpleCoeffs
{
    n          (4 8 1);
    delta      0.001;
}

hierarchicalCoeffs
{
    n          ( 1 1 1 );
    delta      0.001;
    order      xyz;
}

metisCoeffs
{
    processorWeights ( 1 1 1 1 );
}

manualCoeffs
{
    dataFile    "";
}

distributed    no;

roots          ( );

// * * * * *

```

fvSchemes

En este archivo se definen los esquemas numéricos utilizados.

```

/*-----* C++ *-----*\
|=====|
| \ \ / / F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / / O p e r a t i o n | Version: 1.7.1 |
| \ \ / / A n d           | Web:      www.OpenFOAM.com |
| \ \ / / M a n i p u l a t i o n |
|=====|
\*-----*
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       fvSchemes;
}
// * * * * *

ddtSchemes
{
    default      Euler;
}

gradSchemes

```

```

{
    default          Gauss linear;
    grad(epsilon)    cellLimited Gauss linear 1;
    grad(k)          cellLimited Gauss linear 1;
    grad(omega)     cellLimited Gauss linear 1;
}

divSchemes
{
    default          none;
// alphaEqn
    div(phi,alpha)  Gauss limitedLinear01 1;
    div(phir,alpha) Gauss limitedLinear01 1;
// UEqn
    div(phia,Ua)    Gauss linearUpwind grad(Ua);
    div(phib,Ub)    Gauss linearUpwind grad(Ub);
    div(phiRa,Ua)   Gauss linearUpwind grad(Ua);
    div(phiRb,Ub)   Gauss linearUpwind grad(Ub);

// pEqn
    div(alpha,nu)   Gauss linear;

// k and EpsilonEqn
    div(phib,k)     Gauss linearUpwind grad(k);
    div(phib,epsilon) Gauss linearUpwind grad(epsilon);
    div(phib,omega) Gauss linearUpwind grad(omega);

// ThetaEqn
    div(phi,Theta)  Gauss limitedLinear 1;

// alphaPlastic
    div(phia,alphaPlastic)  Gauss upwind;

// pa
    div(phia,pa_new_value)  Gauss upwind;
}
laplacianSchemes
{
    default          Gauss linear corrected;

// alphaEqn
    laplacian(alphaPpMag,alpha) Gauss linear corrected;

// UEqn
    laplacian(nuEffa,Ua) Gauss linear corrected;
    laplacian(nuEffb,Ub) Gauss linear corrected;
    laplacian(nuFra,Ua)  Gauss linear corrected;

// pEqn
    laplacian((rho*(1|A(U))),p) Gauss linear corrected;

// k and EpsilonEqn
    laplacian(DkEff,k) Gauss linear corrected;
    laplacian(DkEff,beta) Gauss linear corrected;
    laplacian(DomegaEff,omega) Gauss linear corrected;
    laplacian(DepsilonEff,epsilon) Gauss linear corrected;
    laplacian(DepsilonEff,beta) Gauss linear corrected;

//ThetaEqn
    laplacian(kappa,Theta) Gauss linear corrected;
}
interpolationSchemes
{
    default  linear;
}
snGradSchemes
{
    default          corrected;
}
fluxRequired
{
    default          no;
}

```

```

    p_rbgh          ;
}
// ***** //

```

fvSolution

Se define la tolerancia de los solucionadores para las diferentes variables y la cantidad de correctores para PIMPLE.

```

/*-----* C++ -*-----*\
|=====|
|  \ \ /  /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox |
|  \ \ /  /  O p e r a t i o n | Version: 1.7.1 |
|  \ \ /  /  A n d      | Web: www.OpenFOAM.com |
|  \ \ /  /  M a n i p u l a t i o n |
\*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       fvSolution;
}
// ***** //

solvers
{
    p_rbgh
    {
        solver          GAMG;
        tolerance       1e-06;
        relTol          0.0001;
        smoother        DIC;
        nPreSweeps      0;
        nPostSweeps     2;
        nFinestSweeps   2;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator     faceAreaPair;
        mergeLevels      1;
    }
    p_rbghFinal
    {
        $p_rbgh;
        tolerance       1e-6;
        relTol          0;
    }
    "(k|epsilon|pa_new_value|alphaPlastic)"
    {
        solver          PBiCG;
        preconditioner   DILU;
        tolerance       1e-5;
        relTol          0;
    }
    omega
    {
        solver          PBiCG;
        preconditioner   DILU;
        tolerance       1e-9;
        relTol          0;
    }
    "(kFinal|epsilonFinal|paFinal|alphaPlasticFinal)"
    {
        solver          PBiCG;
        preconditioner   DILU;
        tolerance       1e-5;
    }
}

```

```

        relTol          0;
    }
    omegaFinal
    {
        solver           PBiCG;
        preconditioner   DILU;
        tolerance        1e-9;
        relTol           0;
    }
    "Ua|UaFinal"
    {
        solver           PBiCG;
        preconditioner   DILU;
        tolerance        1e-6;
        relTol           0;
    }
    "Ub|UbFinal"
    {
        solver           PBiCG;
        preconditioner   DILU;
        tolerance        1e-6;
        relTol           0;
    }
    alpha_a
    {
        solver           PBiCG;
        preconditioner   DILU;
        tolerance        1e-9;
        relTol           0;
    }
    alpha_aFinal
    {
        solver           PBiCG;
        preconditioner   DILU;
        tolerance        1e-9;
        relTol           0;
    }
}

relaxationFactors
{
    fields
    {
        pa              1.;
    }
}

PIMPLE
{
    nCorrectors        2;
    nNonOrthogonalCorrectors 2;
    nAlphaCorr         1;
    correctAlpha       yes;
    pRefPoint          (0.9999 0.2049 0.0);
    //pRefCell         0;
    pRefValue          0;
}

// ***** //

```

topoSetDict

Permite definir una zona para posteriormente crear un parche adicional a los definidos en blockMeshDict.

```
/*-----*- C++ -*-----*\
=====
\\      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
\\      /  O peration    | Website:  https://openfoam.org
\\      /  A nd          | Version:  7
\\      /  M anipulation |
\*-----*- C++ -*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       topoSetDict;
}
// *****

actions
(
    {
        name      Simetria;
        type      cellSet;
        action    new;
        source    boxToCell;
        sourceInfo
        {
            box (-1 0 -0.2) (1 0.5 0.3);
        }
    }
);
// *****
```