

26 de octubre de 2014



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO

FACULTAD DE INGENIERÍA

INSTITUTO DE INGENIERÍA

VALIDACIÓN DE CÓDIGO NUMÉRICO
PARA LA SIMULACIÓN DE FLUJOS
CON SUPERFICIE LIBRE

T E S I S

QUE PARA OPTAR POR EL GRADO DE
INGENIERO MECÁNICO

INGENIERÍA MECÁNICA - TERMOFLUIDOS

P R E S E N T A:

RICARDO ALBERTO MAGOS ALVA

TUTOR:

JOSÉ MANUEL CUBOS RAMÍREZ



INSTITUTO
DE INGENIERÍA
UNAM

2013

JURADO ASIGNADO

Presidente:	Dr. Rogelio Soto Ayala
Secretario:	M. I. José Esteban Barrios Bonilla
Vocal:	M. I. José Manuel Cubos Ramírez
1 ^{er} <i>Suplente</i> :	M. I. Eliseo Carrizosa Elizondo
2 ^{do} <i>Suplente</i> :	Dr. Martín Salinas Vázquez

Lugar o lugares donde se realizó la tesis:

Instituto de Ingeniería

TUTOR DE TESIS

José Manuel Cubos Ramírez

FIRMA

Agradecimientos

*“Dadas unas condiciones iniciales de un determinado sistema,
la más mínima variación en ellas puede provocar que el sistema
evolucione en formas completamente diferentes.”*

TEORÍA DEL CAOS

Con este trabajo concluye un ciclo de mi vida en el que la persona que soy ahora no es más que la suma de todas las experiencias buenas y malas que he vivido, donde nada sería igual sin alguna de las valiosas personas que se han cruzado en mi camino. No hay palabras que puedan describir el profundo agradecimiento que les tendré siempre por formar parte de mi vida.

Primeramente quiero agradecer a mi familia, a quienes amo profundamente.

A mi padre Ricardo Magos, por ser un ejemplo para mí y mis hermanos, por su apoyo y comprensión incondicional en todos los sentidos. Espero poder ser un hombre como tú algún día.

A mi madre Dora Alva, porque sé que siempre puedo contar contigo sin importar nada. Por educarme, cuidarme y por ser la principal responsable de todo lo que soy como persona. Te prometo que jamás te voy a defraudar.

A mis hermanos Luis Antonio y Juan Daniel, por su compañía, amistad y aliento; porque a pesar de ser menores que yo siempre me enseñan mucho y me motivan para ser mejor, sé que sin ustedes mi vida no sería igual. Espero poder ser una buena guía y apoyo para ustedes.

A mi director de tesis José Manuel Cubos, por su ayuda, conocimientos, paciencia y soporte para la realización de este trabajo.

A mis sinodales, por tomarse el tiempo de leer mi tesis y darme valiosos consejos para mejorar este trabajo. Con una mención especial para el Doctor Martín Salinas, quien es una persona que admiro y respeto desde que me dio clases y que influyó mucho para la

realización de este trabajo.

A mis amigos de la facultad Miguel, Edgar y José Luis; por todos los momentos, buenos y malos; las desveladas por proyectos o por las puras ganas de no dormir y porque sé que cuento con ustedes siempre.

A Jocelyn Mejía por brindarme su amistad incondicional y por su ayuda para la realización de este trabajo.

Finalmente a todas las personas que han formado parte de mi vida en todas las etapas, a los que conozco desde hace mucho y los que acabo de conocer.

Gracias a Dios y a mi Universidad.

Índice general

1. Introducción	1
1.1. Generalidades	1
1.2. Flujos multifásicos	2
1.3. Seguimiento de la superficie libre	2
1.4. Alternativas numéricas	5
1.5. Objetivo	6
2. Ecuaciones de Gobierno	7
2.1. Introducción	7
2.2. Ecuación de continuidad (Conservación de masa)	7
2.3. Ecuación de Conservación de Cantidad de Movimiento	8
2.4. Esquema numérico y modelado de la turbulencia	10
2.4.1. Modelado de la turbulencia	10
2.4.2. Simulación de grandes escalas (LES, Large-Eddy Simulation)	11
2.4.2.1. Modelo de sub-malla (Sub-Grid Scale)	12
2.4.3. Aproximación de cuasi compresibilidad	14

2.4.4.	Ecuaciones en coordenadas generalizadas.	14
2.4.5.	Esquema Numérico	15
3.	Metodos de captura de la interfase	17
3.1.	Introducción	17
3.2.	Método Level Set (LSM)	18
3.3.	Local Level Set	21
3.4.	Función distancia y Re inicialización	22
3.5.	Ghost Fluid Method	24
3.6.	Conservación de masa	26
3.6.1.	Corrección a la función LS	28
3.6.1.1.	Paso 1	29
3.6.1.2.	Paso 2	31
3.6.1.3.	Paso 3	31
3.7.	Discretizacion de las ecuaciones VOF y LS	32
3.7.1.	Esquema WENO	33
3.7.2.	Esquema Runge-Kutta	36
4.	Validación	38
4.1.	Introducción	38
4.2.	Resultados	40
4.3.	Dam-Break	40
4.3.1.	Forma de la interfase bajo influencia de la malla	46

4.3.2. Comparación con el caso experimental	48
4.4. Rotura de presa sobre cama de agua	51
4.4.1. Resultados numéricos y experimentales	53
4.5. Dam-break contra un obstáculo	57
4.5.1. Comparación con el caso experimental	58
5. Conclusiones y trabajos futuros	67

Índice de figuras

3.1. Definición de la función Level Set	19
3.2. Descripción de la función Level Set	20
3.3. Formulación del método Ghost Fluid en una dimensión las variables continuas son copiadas de la parte real del fluido rígido a la fantasma del fluido suave, mientras que las variables discontinuas son extrapoladas de la parte real a la parte fantasma del mismo fluido	26
3.4. Interfase, con fracciones de volumen	28
3.5. Volumen para $\Phi \leq 0$	31
4.1. Arreglo Experimental	41
4.2. Mallado para cada prueba	42
4.3. Movimiento del frente de onda	43
4.4. Variación de altura de la columna de agua en función del tiempo	44
4.5. Resultados Experimentales vs Resultados Numéricos	45
4.6. Evolución de la superficie libre para cada prueba	47
4.7. Comparación de resultados numéricos y experimentales	49
4.8. Desarrollo del dam-break en tres dimensiones	50
4.9. Arreglo experimental	51

4.10. Dominio computacional	52
4.11. Comparación de resultados para $d = 18mm$	54
4.12. Comportamiento de los campos de velocidad para $d = 18mm$	55
4.13. Rompimiento de columna sobre cama de agua para diferentes tiempos	56
4.14. Dominio computacional para la tercer simulación	58
4.15. Simulación numérica vs experimental	59
4.16. Historial de la simulación	61
4.17. Variación de altura en la columna de agua	63
4.18. Variación de altura antes del bloque	63
4.19. Ubicación de los sensores de presión en el obstaculo	64
4.20. Variación de la presión en cada sensor	66

Preámbulo

La ingeniería moderna tiene la necesidad de responder a las demandas de la población de una manera más efectiva, optimizando costos y acelerando procesos; se trata de una carrera contra el tiempo para producir más y mejores productos y servicios para satisfacer las necesidades de una población creciente.

Con el fin de cubrir de una mejor manera las exigencias del mercado, los procesos de diseño y producción requieren de nuevas herramientas que signifiquen una ventaja respecto a los métodos usados hasta ahora, como la experimentación o las hipótesis; por lo que las grandes empresas empiezan a invertir en la investigación de nuevas técnicas esperando resultados que se vean reflejados en su utilidad.

Una de estas técnicas es el manejo computacional de problemas de diseño, entre los que se encuentra la Dinámica de Fluidos Computacional (CFD, Computational Fluid Dynamics), una herramienta utilizada en problemas que involucran mecánica de fluidos, transferencia de calor y masa, así como reacciones químicas, arrastre de sólidos, entre otros [1] [2] [3] [4] [5]. Este método de solución es relativamente nuevo, está dando muy buenos resultados y ya es utilizado en diseños para industrias muy importantes como la aeroespacial, aeronáutica y automotriz; se encuentra aún en etapa de perfeccionamiento, buscando más y mejores métodos de resolución, tanto matemáticamente como en programación, con el fin de aprovechar mejor los recursos computacionales, mejorar tiempos y obtener soluciones cada vez más aproximadas a la realidad.

Tradicionalmente los problemas relacionados con mecánica de fluidos son resueltos con datos obtenidos experimentalmente, la desventaja de esto es que generalmente los experimentos son muy caros y entregan resultados confiables después de un largo número de pruebas, lo que lleva mucho tiempo. Además de que aún en experimentos muy detallados, se tienen regiones del dominio en las que es muy difícil realizar mediciones que suelen ser compensadas con cálculos analíticos apoyados en la teoría.

La Dinámica de Fluidos Computacional, es una herramienta que permite reforzar los puntos débiles de los métodos experimentales, al entregar resultados que no se pueden obtener en experimentos o permitir llevar los modelos a situaciones límite, como tempe-

raturas o reacciones peligrosas e inalcanzables experimentalmente; de una manera rápida y confiable.

La CFD consiste en recrear matemáticamente un proceso natural, utilizando métodos numéricos que serán resueltos por una computadora mediante la implementación de un código. Los resultados son normalmente representados gráficamente y permiten estudiar, analizar y trabajar con problemas que de otro modo resultarían muy difíciles de resolver, por ser gobernados por ecuaciones no lineales muy complejas como es el caso de la mecánica de fluidos.

Es por esta razón que siempre que se plantea una nueva metodología numérica para el modelado de fluidos, tiene que validarse usando para esto resultados obtenidos experimentalmente, por lo que las técnicas de diseño se complementan, analítica, experimental y computacionalmente. Aunque todo marcha bien de esta forma, el objetivo es reducir el uso de modelos experimentales y analíticos, diseñando técnicas de simulación que puedan reproducir de una manera más fiel la realidad.

Capítulo 1

Introducción

1.1. Generalidades

El principal problema que enfrenta la CFD es la simulación de los flujos altamente complejos, que incluyen turbulencias, interfases móviles y fluidos de diferentes clases y fases interactuando entre sí. Este tipo de sistemas son difíciles de modelar por la gran cantidad de datos que eso implica. Así mismo, estos flujos son también los de mayor interés en la ingeniería, pues son los más abundantes tanto en procesos naturales como en artificiales.

Contar con una herramienta que permita reproducir el comportamiento de estos fenómenos con una alta precisión, es de gran interés para la ingeniería ya que se tendría la posibilidad de conocer, diseñar, prevenir y trabajar con estos fenómenos con fines muy diversos, desde construcción de estructuras o dispositivos que van a estar expuestos a ciertas condiciones, hasta el aprovechamiento de flujos con fines energéticos.

Es por lo anterior que en las últimas décadas, diversas investigaciones se han esforzado por fortalecer las técnicas de simulación numérica. Los resultados de dichas investigaciones aportan nuevos códigos numéricos que antes de ser utilizados con fines prácticos necesitan probar su capacidad y sus limitaciones con el fin de evitar errores que podrían tener consecuencias graves.

Por lo tanto una parte importante en Dinámica de Fluidos Computacional es la validación. Los métodos de validación consisten en simular un sistema y comparar los datos obtenidos numéricamente con datos pertenecientes al experimento del mismo sistema, para probar el desempeño del código numérico. [17] [18].

Este trabajo se centra en la validación de un código numérico capaz de simular flujos con superficie libre. Por lo que se espera sirva como apoyo para investigaciones futuras que usen este código numérico o que se apoyen en las técnicas numéricas aquí expuestas.

1.2. Flujos multifásicos

Un flujo multifásico es un sistema donde coexisten varias fases, entendiendo por fase un estado de agregación de la materia de una o varias sustancias, que tiene cambios significativos en sus propiedades como variaciones en composición, velocidad, presión, temperatura, etcétera.

Los flujos multifásicos son muy importantes para la ingeniería, pues se presentan en la mayoría de los sistemas naturales y artificiales como el flujo sanguíneo, ríos, mares, medios atmosféricos, comportamiento y propagación de fuego en incendios, combustión, cavitación, flotación y sedimentación, por mencionar algunas. A pesar de su importancia, la implementación de métodos para simular numéricamente este tipo de flujos está en fase de desarrollo [6] [7] [8] [9] [10].

El caso más básico de un sistema multifásico es cuando tenemos solo dos fases, se llama flujo bifásico. La importancia de este sistema radica en que es muy útil para modelar fenómenos de interés ingenieril de una manera sencilla. Prueba de esto es que la mayoría de métodos de simulación numérica de flujos multifásicos, parten de un sistema bifásico para adaptarse a las limitaciones en capacidad computacional sin sacrificar la confiabilidad de los resultados [11] [12] [13].

En la práctica los sistemas que pueden tratarse como sistemas bifásicos son los flujos en canal abierto, donde se tiene un líquido abierto a la atmósfera. Se caracterizan por la presencia de una interfase líquido-gas, conocida como superficie libre.

1.3. Seguimiento de la superficie libre

Los problemas que incluyen dos o más fluidos (o fases de un mismo fluido) son una de las principales complicaciones en Dinámica de Fluidos Computacional, debido a que las condiciones de frontera son puestas arbitrariamente en una superficie en movimiento que es necesario localizar, donde propiedades como la densidad y viscosidad cambian discontinuamente y además actúan fuerzas como la tensión superficial.

Se han propuesto muchos métodos para resolver este tipo de problemas, que se dividen en dos grupos, los métodos de malla móvil (Métodos Lagrangianos) y los métodos de malla

fija (Métodos Eulerianos).

Para el caso de los métodos Lagrangianos, cada punto de la malla está fijo a un punto del material, de tal manera que si el material se deforma también se deforma la malla. Esto significa que este método sólo puede usarse para deformaciones pequeñas como en el caso de un sólido o para fluidos en régimen laminar, ya que si el fluido tiene cambios de forma de gran magnitud, se generan errores numéricos considerables.

Por otro lado, en los métodos Eulerianos, cada punto de la malla está fijo y el fluido se mueve a través de la malla, por lo que en este caso el material se deforma, pero la malla no lo hace. Esta característica vuelve a este método el ideal para el estudio de fluidos, ya que estos se distinguen por tener deformaciones muy grandes.

Existen métodos como el llamado Hidrodinámica de partículas suavizadas SPH (Smoothed Particle Hydrodynamics). Este procedimiento tiene la ventaja de no necesitar de una malla y poder conservar la masa sin cálculos adicionales, esta técnica consiste en representar el fluido con un conjunto de partículas con características propias que se comportan de acuerdo a las ecuaciones de conservación, pero el costo computacional para representar de manera realista el comportamiento de un fluido es muy alto ya que se requiere de un enorme número de partículas [47] [48] [49] [50].

Dentro de los métodos Eulerianos uno de los más usados por sus buenos resultados es el método del Volumen de Fluido (VOF, Volume OF Fluid) [31], utilizado para el seguimiento de superficies libres e interfaces móviles. Esta técnica se apoya en la idea de que los fluidos son inmiscibles, es decir, no pueden mezclarse; lo que significa que el volumen ocupado por un fluido, no puede ser ocupado por otro. Esto nos lleva a tratar con funciones continuas en espacio y tiempo llamadas fracciones de volumen, entonces en cada región de interés para todo tiempo, la suma de las fracciones es igual a uno. La principal ventaja es que cumple con las leyes de conservación de masa, pero tiene una mala estimación de la curvatura de la interfase, si la deformación es muy grande la difusión numérica no permite ver con claridad la interfase. Esta técnica es de las más aceptadas para el modelado de flujos con superficie libre, debido a que tiene buenos resultados y cumple con las leyes conservativas, cabe mencionar que son varios los trabajos donde se usa este método [32] [33] [34] [35] [36] [37].

Otra técnica importante para tratar el movimiento de interfases es la conocida como el Método Level Set (LSM, Level Set Method), propuesto inicialmente por Osher y Sethian[42], se trata de un método Euleriano muy utilizado en problemas de interfases móviles por ser muy simple y eficiente para tratar con grandes deformaciones, en esta técnica la interfase está dada por una curva de nivel cero de una función continua definida sobre el dominio computacional, dicha función se transporta con el fluido y es cero en la superficie libre, para un lado es positiva y para el otro negativa, la desventaja es su discretización, pues en presencia de difusión numérica excesiva tiene problemas con la

conservación de masa. Aunque es un método relativamente nuevo, también representa una muy buena alternativa para el manejo de la superficie libre por lo que es ampliamente usado [44] [45].

La facilidad con la que el método Level Set modela el comportamiento de la interfase, hace que esta técnica sea una de las más adecuadas para simulaciones de flujo con superficie libre. Sin embargo, la no conservación de masa es un gran obstáculo para la implementación de Level Set en los códigos de simulación numérica.

Es por lo anterior que muchos investigadores se han enfocado en minimizar esa desventaja usando diferentes alternativas que tratan de dar un carácter conservativo a la función Level Set. Estas técnicas se dividen en tres grupos fundamentales, el primero de ellos es el conocido como Level Set conservativo, este grupo de técnicas se basan en que la función Level Set es altamente sensible a la discretización espacial, por lo que si se hace uso de esquemas de discretización de alto orden para la función Level Set el grado de error inherente al método se reduce y la conservación de masa aumenta, aunque parece un enfoque simple, el costo computacional implicado es muy alto, por lo que no es una opción para flujos complejos.

Otra de las alternativas es la conocida como Adaptive Mesh Refinement [54] [55] [56], la idea principal es refinar la malla en las regiones donde está presente la superficie libre, de esta manera se reduce la difusión numérica en la zona donde la función Level Set cobra importancia y por lo tanto la conservación de masa se mejora considerablemente, el problema de esto es que en la mayoría de los casos la ubicación de la superficie libre es una incógnita, esto hace que el método sea generalmente impráctico.

Por último tenemos a los métodos acoplados, este tipo de técnicas consisten mejorar los puntos débiles de una técnica, aprovechando las fortalezas de otra. Ejemplos de estas metodologías son CLSVOF [57], su variante adaptativa ACLVOF [58], el método VOSEF [?] y el método LS/VOF [60] [61]. Estos métodos requieren de un paso extra para la reconstrucción de la interfase, que generalmente es complicado y costoso computacionalmente.

El método de MCLS (Mass-Conserving Level-Set) [62], es uno más de los métodos acoplados cuya característica principal es que no necesita un paso para la reconstrucción de interfase, ya que ésta es calculada como una relación entre la función Level Set y la fracción de volumen (Función VOF). Con el uso de MCLS, la función Level Set es el método principal y la técnica VOF es usada para realizar pequeñas correcciones de conservación de masa con un pequeño costo computacional.

1.4. Alternativas numéricas

En la actualidad se trabaja con diversos métodos para la simulación numérica de fluidos que pueden aportar mucha información, cada uno con sus ventajas y desventajas; por lo que la elección de la metodología a utilizar está íntimamente ligada a lo que se desea saber de un fenómeno en particular, evitándose cierta información que podría resultar menos útil.

Entre las alternativas que se tienen para resolver las ecuaciones que rigen el comportamiento de los fluidos, partiremos de la más fundamental: la simulación numérica directa (DNS, Direct Numerical Simulation), en donde se resuelven las ecuaciones de Navier – Stokes, sin ninguna aproximación o modelo, por lo que ofrecerían los resultados más exactos posibles, pero para el caso de flujos muy complejos, la capacidad de las computadoras actuales se ve rebasada, ya que requiere de un gran número de celdas para capturar todas las escalas espaciales y temporales de la turbulencia. No obstante se han llevado a cabo algunas simulaciones con esta técnica que se encuentra en una etapa muy temprana de desarrollo [19] [20] [21].

Por otro lado tenemos los modelos que promedian temporalmente las ecuaciones de Navier – Stokes (RANS, Reynolds Averaged Navier – Stokes Equations), esta técnica consiste en promediar las ecuaciones de transporte, resolviéndolas utilizando valores compuestos por la suma de un valor medio y un componente fluctuante, lo que genera términos adicionales que necesitan de otras ecuaciones para cerrar el sistema en función del modelo de turbulencia utilizado. Este método exige de menores recursos de cómputo, pero no ofrece una buena precisión en flujos con turbulencia en regiones alejadas de la frontera. Es quizá la técnica más estudiada y utilizada en simulación de fluidos, debido a su simplicidad y buenos resultados en problemas prácticos, por lo que existen una gran cantidad de trabajos que usan RANS [22] [23] [24].

Finalmente tenemos la simulación de grandes escalas (LES, Large Eddy Simulation). Consiste en resolver de forma directa las grandes escalas de turbulencia, modelando sobre ellas el efecto de las más pequeñas; basándose en que teóricamente los grandes remolinos son los que transportan mayor cantidad de energía, mientras que los pequeños están relacionados con efectos de disipación. Aunque este método es más costoso que RANS en términos de requerimientos computacionales, ofrece mejores resultados para flujos altamente turbulentos incluso en regiones alejadas de la frontera, además de ser viable para las computadoras con las que se cuenta.

Debido a las ventajas que presenta LES respecto a otros métodos, muchos investigadores están trabajando con esta técnica para el modelado de diversos sistemas [25] [26] [27]. La técnica LES, será utilizada en este trabajo, por lo que se tratará más a fondo en capítulos posteriores.

1.5. Objetivo

Este trabajo tiene como propósito presentar un código numérico para la simulación de un flujo tridimensional con superficie libre; dicho código fue desarrollado basándose en investigaciones previas con el fin de encontrar una forma de reproducir este tipo de sistemas de la manera eficiente.

Para llevar a cabo las simulaciones numéricas se implementarán la técnica de LES para la resolución de las ecuaciones de gobierno y el modelo acoplado MCLS para el seguimiento de la superficie libre con una discretización mediante diferencias finitas.

Para validar el código, se realizará la simulación y comparación de tres casos formados por un sistema bifásico de agua y aire en 3D, el primer caso es una rotura de presa libre (Dam-Break), el segundo es una rotura de presa en una cama de agua y el último caso es una rotura de presa con un obstáculo.

Estos ejemplos fueron seleccionados por representar un reto importante para nuestro código numérico, pues en ellos los cambios en la topología de la superficie libre y la relación de densidades en los fluidos son las características dominantes.

El motivo de proponer un modelo numérico para simulaciones de sistemas de este tipo, es porque se considera que cuentan con las características más difíciles de modelar computacionalmente como son: turbulencia y superficie libre, de tal forma que la información que puede proporcionar es muy interesante desde el punto de vista de la simulación numérica.

El comprobar la confiabilidad del método, ayudará a la búsqueda de metodologías más precisas con fines de reducir tiempo y costos en el diseño y manufactura de productos de ingeniería de alta tecnología, significando el ahorro de recursos naturales y humanos, así como la disminución en el costo de dichos productos.

Por ultimo, se pretende que este trabajo sirva como apoyo para investigaciones futuras en las que el código numérico o las técnicas utilizadas sean desarrolladas de una manera más profunda y práctica.

Capítulo 2

Ecuaciones de Gobierno

2.1. Introducción

Las leyes de conservación son los principios fundamentales de los análisis en ingeniería, de acuerdo a estas leyes las propiedades materiales como la masa, cantidad de movimiento, energía y entropía pueden cambiar solamente en equilibrio con otras propiedades físicas como fuerza, trabajo y transferencia de calor.

Debido a la naturaleza de los fluidos dichas ecuaciones se han formulado de una manera más útil para la resolución de problemas. Se trata de un conjunto acoplado de ecuaciones diferenciales parciales no lineales y no estacionarias que reciben el nombre de ecuaciones de Navier-Stokes.

Para este trabajo los principios a resolver numéricamente son en esencia la conservación de masa y la conservación de la cantidad de movimiento.

2.2. Ecuación de continuidad (Conservación de masa)

El principio más fundamental de la naturaleza es el que establece que la masa no se crea ni se destruye, para el caso de sistemas cerrados decimos que la masa permanece constante durante todo el proceso, pero en mecánica de fluidos es muy difícil trabajar con porciones de materia, por lo que utilizamos el concepto de volumen de control, que es una región del espacio en la que nos interesa la cantidad de masa que entra y sale, así que para nosotros la ecuación de la conservación de masa recibe el nombre de ecuación de continuidad y se escribe de la siguiente manera:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \quad (2.1)$$

Donde ρ es la densidad del fluido y \vec{u} es la velocidad.

El primer termino expresa la variación de la densidad respecto al tiempo y el segundo termino describe el flujo neto de masa a través de las fronteras.

En palabras sencillas esta ecuación significa que la transferencia neta de masa hacia dentro o hacia fuera del volumen de control, es igual al cambio neto en la masa total que esta dentro de ese volumen.

2.3. Ecuación de Conservación de Cantidad de Movimiento

El producto de la masa y la velocidad de un cuerpo, se llama cantidad de movimiento, en mecánica clásica este principio es explicado matemáticamente mediante la segunda ley de Newton.

La ley de conservación de la cantidad de movimiento dice que la cantidad de movimiento de un sistema permanece constante cuando la fuerza neta que actúa sobre él es cero.

La conservación de cantidad de movimiento en mecánica de fluidos se describe a través de la siguiente ecuación (conocida como ecuación de Cauchy):

$$\frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \vec{u}) = -\nabla p + \nabla \cdot \vec{\tau} + \rho \vec{f}_m \quad (2.2)$$

Donde p es la presión, \vec{f}_m son las fuerzas másicas o de cuerpo y $\vec{\tau}$ es el tensor de esfuerzos viscosos que describe los esfuerzos superficiales en un punto en el flujo, esta ecuación es valida para cualquier punto en el dominio del flujo y expresa que la variación de la cantidad de movimiento del volumen del fluido es igual a las fuerzas que actúan en él.

Las fuerzas que pueden actuar en un fluido son de dos tipos, fuerzas de cuerpo como la gravedad, la eléctrica y la magnética; y fuerzas de superficie como la fuerza viscosa, la presión y las reacciones en los puntos de contacto.

En un marco de referencia cartesiano x, y, z las ecuaciones de flujo compresible de Navier–Stokes son:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} = \mathbf{S} \quad (2.3)$$

Donde \mathbf{U} es un vector de cuatro componentes definido por:

$$U = \begin{bmatrix} \rho\lambda \\ \rho\lambda u_1 \\ \rho\lambda u_2 \\ \rho\lambda u_3 \end{bmatrix} \quad (2.4)$$

Se considera además que $\mathbf{u} = (u_1, u_2, u_3)$ es el vector velocidad, ρ es la densidad.

También el vector velocidad se escribe como $\mathbf{u} = (u, v, w)$. La ecuación (2.3) representa la evolución de la densidad (ecuación de continuidad) y cantidad de movimiento.

\mathbf{F}_i son flujos donde $\forall i \in \{1, 2, 3\}$, y para un fluido Newtoniano esta dado por:

$$\mathbf{F}_i = \left. \begin{array}{l} \left(\begin{array}{l} \rho\lambda u_i \\ \rho\lambda u_i u_1 + p\delta_{i1} - 2\mu_\lambda S_{i1} \\ \rho\lambda u_i u_2 + p\delta_{i2} - 2\mu_\lambda S_{i2} \\ \rho\lambda u_i u_3 + p\delta_{i3} - 2\mu_\lambda S_{i3} \end{array} \right) \\ \left. \begin{array}{l} \rightarrow \text{Ecuación de continuidad} \\ \text{Ecuaciones de Navier-Stokes} \end{array} \right\} \quad (2.5)$$

Donde el subíndice λ denota propiedades de dos materiales diferentes, líquido y gas, respectivamente, δ_{ij} es la delta Kronecker y S_{ij} es el componente divergencia del tensor deformación. Despreciando la viscosidad, S_{ij} se escribe:

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} (\nabla \cdot \mathbf{u}) \delta_{ij} \right) \quad (2.6)$$

2.4. Esquema numérico y modelado de la turbulencia

2.4.1. Modelado de la turbulencia

La mayoría de los flujos de interés en ingeniería se desarrollan en un estado donde el movimiento de las partículas del fluido es caótico, aleatorio y por lo tanto inestable, bajo estas condiciones la velocidad y la presión cambian continuamente con el tiempo, dicho estado es conocido como régimen turbulento.

La turbulencia provoca la aparición de remolinos en el flujo, con una amplia gama de escalas de longitud y tiempo que interactúan de una manera dinámica compleja. Las dificultades para desarrollar un modelo general para la turbulencia se deben en gran medida a los diferentes comportamientos en los remolinos grandes y los pequeños.

Las grandes escalas de la turbulencia son generalmente más energéticas debido a que están dominadas por los efectos de inercia y los efectos viscosos son insignificantes, su comportamiento está dictado por la geometría del dominio del problema, las condiciones de frontera y las fuerzas de cuerpo. El comportamiento general es anisótropo, extrae energía de la media del flujo y conservan su momento angular haciendo que la velocidad de rotación aumente disminuyendo al mismo tiempo el radio de su sección transversal, estas interacciones son las responsables de que la turbulencia crezca o se mantenga.

Por otro lado, las escalas pequeñas están dominadas por los efectos viscosos, son fuertemente influenciados por las escalas más grandes y en menor medida por el flujo medio. La energía asociada con su movimiento se disipa y se convierte en energía térmica interna, por lo que su comportamiento es disipativo e isotrópico; no tienen influencia en el transporte de propiedades y su comportamiento es universal.

Es por lo anterior que existen diferentes técnicas para modelar el efecto de la turbulencia en un flujo, siendo la más usada el promediado de valores (RANS, Reynolds Averaged Navier-Stokes) cuyos resultados son aceptables pero no tienen un buen grado de exactitud. Otra alternativa es la simulación directa (DNS, Direct Numerical Simulation) donde los resultados ofrecen un grado de exactitud muy elevado, pero que tiene la desventaja de ser muy costoso computacionalmente hablando, por lo que para flujos complejos se vuelve prácticamente imposible de implementar. Un enfoque diferente es la técnica conocida como LES (Large-Eddy Simulation), la naturaleza de este método permite superar las desventajas de RANS y DNS, sus características se mencionan enseguida.

2.4.2. Simulación de grandes escalas (LES, Large-Eddy Simulation)

La idea principal de la técnica LES se basa en la separación de escalas a través de un filtrado espacial de las ecuaciones de Navier-Stokes. Las escalas de la turbulencia que son afectadas por las condiciones de contorno (las grandes escalas); se resuelven directamente y las escalas más pequeñas se parametrizan mediante el uso de un modelo estadístico denominado modelo de sub-malla (subgrid-scale model). Partiendo de esta idea y reteniendo solo las grandes escalas del flujo, será posible reducir el costo computacional y al mismo tiempo seguir capturando las características deseables del flujo.

El método LES comienza con la operación de filtrado que consiste en separar una parte deseable que se conserva y una parte no deseada que se rechaza. Lo que deriva en una fracción filtrada libre de pequeñas escalas y una porción de sub-malla. El filtrado comienza con la selección de la función de filtrado espacial $G_\Delta(x)$ de tamaño Δ . Donde el ancho de corte Δ es el indicativo de la medida del tamaño de los remolinos que serán retenidos en los cálculos y los que serán rechazados.

Matemáticamente, la operación de filtrado corresponde a la integral de convolución de alguna cantidad $f(x, t)$ del flujo por la función filtro $G_\Delta(x)$ en la forma:

$$\bar{f}(x, t) = \int f(y, t)G_\Delta(x - y)dy \quad (2.7)$$

La variable original está formada por la suma de la parte filtrada \bar{f} más la parte sub-malla.

$$f = \bar{f} + f' \quad (2.8)$$

La aplicación del filtro a las ecuaciones compresibles de Navier – Stokes produce:

$$\frac{\partial \bar{\mathbf{U}}}{\partial t} + \frac{\bar{\mathbf{F}}_i}{\partial x_i} = S \quad (2.9)$$

Es conveniente para los propósitos de LES introducir un promediado de Favre[~], definido para una variable (ϕ) como

$$\tilde{\phi} = \frac{\overline{\rho\phi}}{\bar{\rho}} \quad (2.10)$$

Por lo que se tiene:

$$U = \begin{bmatrix} \bar{\rho} \\ \bar{\rho}\tilde{u}_1 \\ \bar{\rho}\tilde{u}_2 \\ \bar{\rho}\tilde{u}_3 \end{bmatrix} \quad (2.11)$$

Los flujos resueltos $\bar{\mathbf{F}}_i$ se escriben:

$$\bar{\mathbf{F}}_i = \begin{pmatrix} \bar{\rho}\tilde{u}_i \\ \frac{\bar{\rho}\tilde{u}_i\tilde{u}_1 + \bar{p}\delta_{i1} - 2\mu\overline{S_{i1}}}{\bar{\rho}\tilde{u}_i\tilde{u}_2 + \bar{p}\delta_{i2} - 2\mu\overline{S_{i2}}} \\ \frac{\bar{\rho}\tilde{u}_i\tilde{u}_3 + \bar{p}\delta_{i3} - 2\mu\overline{S_{i3}}}{\bar{\rho}\tilde{u}_i\tilde{u}_3 + \bar{p}\delta_{i3} - 2\mu\overline{S_{i3}}} \end{pmatrix} \quad (2.12)$$

Dado que la ecuación de la continuidad es lineal, el filtrado no cambia, pero es importante notar que en las ecuaciones de Navier-Stokes $\overline{\rho u_i u_j} \neq \bar{u}_i \bar{u}_j$ y la primer cantidad del lado izquierdo no se calcula fácilmente, una aproximación de modelado para la diferencia entre los lados de la desigualdad es la siguiente

$$\mathcal{T}_{i,j} = -\overline{\rho u_i u_j} + \overline{\rho u_i} \bar{u}_j \quad (2.13)$$

Al introducirla en la ecuación (2.12). La expresión para los flujos resueltos se convierte en

$$\bar{\mathbf{F}}_i = \begin{pmatrix} \bar{\rho}\tilde{u}_i \\ \frac{\bar{\rho}\tilde{u}_i\tilde{u}_1 + \bar{p}\delta_{i1} - \tau_{ij} - 2\mu\overline{S_{i1}}}{\bar{\rho}\tilde{u}_1\tilde{u}_2 + \bar{p}\delta_{i2} - \tau_{ij} - 2\mu\overline{S_{i2}}} \\ \frac{\bar{\rho}\tilde{u}_1\tilde{u}_3 + \bar{p}\delta_{i3} - \tau_{ij} - 2\mu\overline{S_{i3}}}{\bar{\rho}\tilde{u}_1\tilde{u}_3 + \bar{p}\delta_{i3} - \tau_{ij} - 2\mu\overline{S_{i3}}} \end{pmatrix} \quad (2.14)$$

En el entorno de LES τ_{ij} es llamado el esfuerzo de sub-malla (subgrid-scale stress). El nombre “esfuerzo” deriva de la forma en que se trata más que por su naturaleza física.

2.4.2.1. Modelo de sub-malla (Sub-Grid Scale)

Con el fin de estimar los esfuerzos τ_{ij} SGS, en este trabajo se usa el modelo de función estructurada selectiva (SSF, Selective Structure-Funtion Model), reportado por [28]. Con este modelo, el esfuerzo SGS queda como

$$\tau_{ij} \approx \bar{\rho}\nu_t \tilde{S}_{ij} \quad (2.15)$$

Por lo que la ecuación (2.14) se puede reescribir como

$$\bar{\mathbf{F}}_i = \begin{pmatrix} \bar{\rho}\tilde{u}_i \\ \bar{\rho}\tilde{u}_i\tilde{u}_1 + \bar{p}\delta_{i1} - \tau_{ij} - 2(\bar{\mu} + \bar{\rho}\nu_t)\tilde{S}_{i1} \\ \bar{\rho}\tilde{u}_i\tilde{u}_2 + \bar{p}\delta_{i2} - \tau_{ij} - 2(\bar{\mu} + \bar{\rho}\nu_t)\tilde{S}_{i2} \\ \bar{\rho}\tilde{u}_i\tilde{u}_3 + \bar{p}\delta_{i3} - \tau_{ij} - 2(\bar{\mu} + \bar{\rho}\nu_t)\tilde{S}_{i3} \end{pmatrix} \quad (2.16)$$

Esto proporciona al código numérico un fácil uso para el LES sin modificaciones severas.

Las expresiones para ν_t y Pr_t utilizadas en las siguientes simulaciones compresibles corresponden a los modelos incompresibles descritos en Métais y Lesieur (1996), la única diferencia es que aquí se utiliza un promedio de Favre, antes descrito. Nuestro modelo de submalla es el modelo de función estructurada selectiva, propuesto por David (1993) [29], la viscosidad local del remolino, esta dada por:

$$\nu_t(x, \Delta, t) = C_{ssf} \Delta \sqrt{\tilde{F}_2(x, \Delta, t)} \quad (2.17)$$

Donde C_{ssf} puede ser expresado como función de la constante de Kolmogorov C_K : $C_{ssf} = f(C_K^{-\frac{3}{2}})$. C_{ssf} toma el valor de 0,104 para $C_K = 1,4$. Δ se toma igual a $(\Delta x \Delta y \Delta z)^{\frac{1}{3}}$, donde Δx , Δy y Δz , son los tamaños de la malla locales en las tres direcciones espaciales.

$\tilde{F}_2(x, \Delta, t)$ es la función de estructura de segundo orden de la velocidad construida con el campo $\tilde{\mathbf{u}}$. \tilde{F}_2 es calculado en el punto x con un promedio estadístico local de las diferencias de la velocidad de cuatro de los seis puntos más cercanos que rodean al punto x en la malla computacional. La interpolación se basó sobre la ley de $\frac{2}{3}$ de Kolmogorov que se usa para la función estructura de la velocidad.

Según lo propuesto por David (1993) [29], la viscosidad turbulenta se apaga cuando la turbulencia no es lo suficientemente tridimensional. El criterio para tres dimensiones es definido como sigue: considérese en un momento dado que el ángulo entre el vector de vorticidad en un punto dado de la malla y su medio aritmético de los seis puntos vecinos más cercanos. La viscosidad turbulenta se cancela en los puntos donde este ángulo es más pequeño que 20° . Finalmente, el número de Prandtl turbulento se toma igual a 0,6, con lo que cierra la ecuación de la energía.

2.4.3. Aproximación de cuasi compresibilidad

Al resolver una ecuación artificial de estado es posible aproximar un flujo incompresible. Empezando con las ecuaciones de Navier-Stokes en su forma compresible (2.3) y una ecuación de estado cuasi incompresible en la forma.

$$p = f(\rho) \tag{2.18}$$

Las simulaciones se realizan con la siguiente ecuación artificial de estado.

$$p = c^2 \rho \tag{2.19}$$

Donde c es la velocidad del sonido en el medio. Siempre y cuando los flujos se mantengan en un número de Mach bajo y en condiciones isotérmicas, la solución de estas ecuaciones debe aproximarse al límite incompresible.

La velocidad del sonido debe ser elegida cuidadosamente para asegurar eficiencia y exactitud al problema dado. El valor de c debe ser lo suficientemente grande para que el comportamiento del correspondiente fluido cuasi incompresible se aproxime al fluido real, sin embargo no debe ser tan grande para no permitir que el paso de tiempo sea excesivamente pequeño. De acuerdo a [30] el número de Mach debe ser 0.1 o menos.

2.4.4. Ecuaciones en coordenadas generalizadas.

El código numérico usa coordenadas generalizadas. La adaptación a las coordenadas generalizadas se realiza introduciendo una matriz Jacobiana que transforma una geometría compleja de malla no uniforme o geometría curvilínea, en un sistema de coordenadas Cartesiano (x, y, z) , dentro de una geometría ortogonal simple con malla uniforme en el sistema de coordenadas generalizadas (ξ_1, ξ_2, ξ_3) donde las ecuaciones se pueden resolver más fácilmente. Para este caso, simplemente consiste en una transformación de una malla no uniforme en el espacio físico (x, y, z) dentro de una malla uniforme en el espacio computacional (ξ_1, ξ_2, ξ_3) . Cada término en la matriz Jacobiana inversa (J^{-1}) se expresa como funciones analíticas de las medidas $\frac{\partial x_i}{\partial \xi_j}$. Las medidas son introducidas y calculadas por el esquema interno de primer orden, entonces la matriz (J) es calculada directamente de (J^{-1}) .

La ecuación (2.3) se puede volver a escribir como:

$$\frac{\partial \hat{\mathbf{U}}}{\partial t} + \frac{\partial \hat{\mathbf{F}}}{\partial \xi_1} + \frac{\partial \hat{\mathbf{G}}}{\partial \xi_2} + \frac{\partial \hat{\mathbf{H}}}{\partial \xi_3} = \hat{\mathbf{S}} \quad (2.20)$$

Con

$$\left. \begin{aligned} \hat{\mathbf{U}} &= \frac{\mathbf{U}}{J}, \\ \hat{\mathbf{F}} &= \frac{1}{J} \left[\left(\frac{\partial \xi_1}{\partial x_1} \mathbf{F} \right) + \left(\frac{\partial \xi_1}{\partial x_2} \mathbf{G} \right) + \left(\frac{\partial \xi_1}{\partial x_3} \mathbf{H} \right) \right], \\ \hat{\mathbf{G}} &= \frac{1}{J} \left[\left(\frac{\partial \xi_2}{\partial x_1} \mathbf{F} \right) + \left(\frac{\partial \xi_2}{\partial x_2} \mathbf{G} \right) + \left(\frac{\partial \xi_2}{\partial x_3} \mathbf{H} \right) \right], \\ \hat{\mathbf{H}} &= \frac{1}{J} \left[\left(\frac{\partial \xi_3}{\partial x_1} \mathbf{F} \right) + \left(\frac{\partial \xi_3}{\partial x_2} \mathbf{G} \right) + \left(\frac{\partial \xi_3}{\partial x_3} \mathbf{H} \right) \right], \\ \hat{\mathbf{S}} &= \frac{1}{J} \mathbf{S}. \end{aligned} \right\} \quad (2.21)$$

J es el determinante de la matriz (J) y \mathbf{U} es función de las coordenadas cartesianas y del tiempo.

2.4.5. Esquema Numérico

El sistema en coordenadas generalizadas se resuelve por medio de una extensión del completo esquema explícito McCormack, de segundo orden en el tiempo y cuarto orden en el espacio, desarrollado por Gottlieb Turkel (1976). Debe observarse que cuando se usa \mathbf{U} tiende a ser reemplazada por $\bar{\mathbf{U}}$ definida por la ecuación (2.11) cuando la técnica LES es considerada. El esquema numérico es un esquema corrector – predictor definido en una dimensión por,

Predictor

$$U_j^{(1)} = U_j^{(n)} + \frac{1}{6} \lambda \left(-f_{j+2}^{(n)} + 8f_{j+1}^{(n)} - 7f_j^{(n)} \right) + (\delta t) S_j^{(n)}, \quad (2.22)$$

Corrector

$$U_j^{(n+1)} = \frac{1}{2} \left(U_j^{(n)} + \left(U_j^{(1)} \right) \right) + \frac{1}{12} \lambda \left(7f_{j-2}^{(1)} - 8f_{j-1}^{(1)} + f_j^{(1)} \right) + \frac{1}{2} (\delta t) S_j^{(1)}. \quad (2.23)$$

Los índices (n) , $(n+1)$ y (1) simbolizan respectivamente para los valores de la función al tiempo t , tiempo $t + \delta t$ y al paso – sub – tiempo. Obsérvese que las discretizaciones espaciales intermedias son esquemas no centrados de primer orden con un predictor adelantado (*upwind*) y un corrector atrasado (*downwind*). Como se especifica arriba el esquema resultante es de cuarto orden en el espacio.

La formulación generalizada en tres dimensiones se escribe,

Predictor

$$U_{i,j,k}^n = U_{i,j,k}^n - J_{i,j,k}^P \left[\begin{array}{l} \frac{\Delta t}{\Delta \xi_1} \left[\frac{7}{6} \left(\hat{F}_{i+1,j,k}^n - \hat{F}_{i,j,k}^n \right) - \frac{1}{6} \left(\hat{F}_{i+2,j,k}^n - \hat{F}_{i+1,j,k}^n \right) \right] \\ \frac{\Delta t}{\Delta \xi_2} \left[\frac{7}{6} \left(\hat{G}_{i,j+1,k}^n - \hat{G}_{i,j,k}^n \right) - \frac{1}{6} \left(\hat{G}_{i,j+2,k}^n - \hat{G}_{i,j+1,k}^n \right) \right] \\ \frac{\Delta t}{\Delta \xi_3} \left[\frac{7}{6} \left(\hat{H}_{i,j,k+1}^n - \hat{H}_{i,j,k}^n \right) - \frac{1}{6} \left(\hat{H}_{i,j,k+2}^n - \hat{H}_{i,j,k+1}^n \right) \right] \end{array} \right] \quad (2.24)$$

Corrector

$$U_{i,j,k}^{n+1} = \frac{1}{2} [U_{i,j,k}^1 + U_{i,j,k}^n] - \frac{1}{2} J_{i,j,k}^C \left[\begin{array}{l} \frac{\Delta t}{\Delta \xi_1} \left[\frac{7}{6} \left(\hat{F}_{i,j,k}^1 - \hat{F}_{i-1,j,k}^1 \right) - \frac{1}{6} \left(\hat{F}_{i-1,j,k}^1 - \hat{F}_{i-2,j,k}^1 \right) \right] \\ \frac{\Delta t}{\Delta \xi_2} \left[\frac{7}{6} \left(\hat{G}_{i,j,k}^1 - \hat{G}_{i,j-1,k}^1 \right) - \frac{1}{6} \left(\hat{G}_{i,j-1,k}^1 - \hat{G}_{i,j-2,k}^1 \right) \right] \\ \frac{\Delta t}{\Delta \xi_3} \left[\frac{7}{6} \left(\hat{H}_{i,j,k}^1 - \hat{H}_{i,j,k-1}^1 \right) - \frac{1}{6} \left(\hat{H}_{i,j,k-1}^1 - \hat{H}_{i,j,k-2}^1 \right) \right] \end{array} \right] \quad (2.25)$$

Capítulo 3

Metodos de captura de la interfase

3.1. Introducción

Los problemas que se presentan en ingeniería que están relacionados con la mecánica de fluidos son en su mayoría multifásicos, esto quiere decir que involucran flujos de fluidos que tienen mezclas inmiscibles, ya sea porque tratan con muchos fluidos o varias fases de uno mismo.

Uno de los principales problemas de trabajar con este tipo de sistemas es el manejo de la interfase, pues en este lugar se presentan discontinuidades en propiedades como la viscosidad y la densidad, por lo que la forma en que se desarrolla el flujo depende de cómo se relacionan los fluidos que lo conforman. Además, es difícil localizar la posición de la superficie libre y actualizarla en función del tiempo.

Es por todo lo anterior que una buena simulación requiere de un método eficiente para el cálculo de la evolución de la superficie libre.

El código numérico usado en esta tesis, está diseñado para trabajar con sistemas de flujo bifásico incompresible y tiene la capacidad de simular de la manera más exacta posible el comportamiento de la interfase entre dos fluidos, reduciendo al mismo tiempo problemas asociados a las simulaciones de este tipo como son la pérdida de masa, topologías físicamente no realistas en la superficie libre, errores en continuidad de variables como densidad y viscosidad cerca de la interfase, así como reducir los costos en computación y tiempo que un modelo de este tipo requiere. Para lograr el desempeño esperado se llevó a cabo la combinación de diversas técnicas numéricas que en conjunto consiguen muy buenos resultados; las técnicas utilizadas y la forma en que se relacionan se describen a continuación.

3.2. Método Level Set (LSM)

El método Level Set (Level Set Method, LSM), propuesto inicialmente por Osher y Sethian [42] es un procedimiento numérico para modelar el movimiento de curvas o superficies evolutivas como la interfase, capaz de tratar con problemas tanto estáticos como dinámicos.

El uso de esta técnica para la CFD tiene muchas ventajas, la principal es que el movimiento de la superficie libre puede ser calculado en una malla fija, por lo que resulta muy sencillo seguir cambios en la topología de la interfase. Además la superficie libre no tiene que ser encontrada explícitamente sino que está guardada en toda la información que aporta la función Level Set, es un método que se adapta fácilmente a dos y tres dimensiones [43] [44]. Por lo anterior el método LS es tomado como la técnica principal para la captura de la superficie libre para este trabajo.

El método consiste en la introducción de una función ϕ que cumple con ciertas características que permiten capturar la evolución en el tiempo de curvas o superficies y que recibe el nombre de función Level Set.

$$\phi(\bar{x}(t), t) \quad (3.1)$$

Para el caso de un flujo bifásico, la función Level Set es una función definida en ambos fluidos pero con signo opuesto en cada uno, siendo positivo en un fluido y negativo en el otro; de manera que la región de un fluido está definida por $\phi(x) > 0$, mientras que en el otro fluido se cumple que $\phi(x) < 0$. Por lo tanto la interfase está implícitamente definida por el conjunto de puntos donde $\phi(x) = 0$ Figura 3.1.

Cuando delimitamos el dominio Ω por una superficie cerrada en movimiento Γ , el resultado es la existencia de dos subdominios Ω^+ y Ω^- , donde $\Gamma_t : \{\bar{x} | \phi(\bar{x}, t) > 0\}$ delimita la parte interior de la superficie que encierra el subdominio Ω^- mientras que $\Gamma_t : \{\bar{x} | \phi(\bar{x}, t) < 0\}$ es la parte exterior, el subdominio Ω^+ como se muestra en la Figura 3.2.

Para este caso la función Level Set satisface las siguientes condiciones.

$$\begin{aligned} \phi(\bar{x}(t), t) &< 0 \quad \text{en} \quad \Omega^+(t) \\ \phi(\bar{x}(t), t) &= 0 \quad \text{en} \quad \Gamma(t) \\ \phi(\bar{x}(t), t) &> 0 \quad \text{en} \quad \Omega^-(t) \end{aligned} \quad (3.2)$$

Entonces la frontera $\partial\Omega$ del dominio Ω esta definida por todos los puntos en donde la

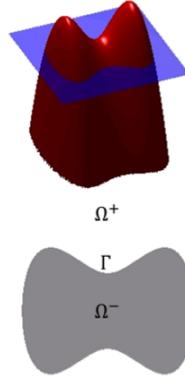


Figura 3.1: Definición de la función Level Set

función Level Set es igual a cero.

$$\phi(\bar{x}(t), t) = 0, \quad \bar{x} \in \partial\Omega \quad (3.3)$$

Otras propiedades geométricas importantes de la interfase son la curvatura $\kappa(x)$, y el campo de vectores normales $n(x)$. Las cuales solo tienen un significado físico en la frontera $\partial\Omega$ y se calculan para cada paso de tiempo como sigue.

$$n(x) = \frac{\nabla\phi(x)}{\|\nabla\phi(x)\|} \quad (3.4)$$

$$\kappa(x) = -\nabla \cdot n(x) \quad (3.5)$$

La evolución de la interfase es el resultado de la advección de la función ϕ por el campo de velocidades local $\bar{u} = \dot{x}(t)$. Por lo tanto, la ecuación de transporte para la función Level Set es,

$$\frac{\partial\phi}{\partial t} + \nabla\phi \cdot \bar{u} = 0 \quad (3.6)$$

Además, sabemos que para cada $\bar{x} \in \partial\Omega$, se cumple

$$\bar{u} \cdot \bar{n} = F(\bar{x}) \quad (3.7)$$

Con lo que tenemos,

$$\nabla\phi \cdot \bar{u} = F|\nabla\phi| \quad (3.8)$$

Finalmente obtenemos la ecuación Level Set que al combinarla con la superficie cerrada inicial Γ , obtenemos los valores iniciales del problema.

$$\begin{aligned} \phi_t + F|\nabla\phi| &= 0 \\ \Gamma(t) = \bar{x} : \phi(\bar{x}, t) &= 0 \end{aligned} \quad (3.9)$$

Por lo tanto la ecuación que sirve para determinar la evolución de la función Level Set ϕ en un campo de velocidades \bar{u} se escribe en su forma conservativa como,

$$\frac{\partial\phi}{\partial t} + \nabla \cdot (\phi\bar{u}) = 0 \quad (3.10)$$

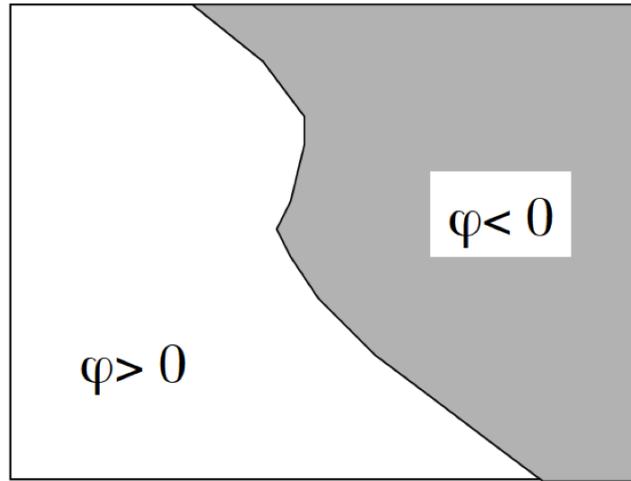


Figura 3.2: Descripción de la función Level Set

Una de las desventajas del método es que la función Level Set es una dimensión superior a la de la frontera y si la frontera está en movimiento es necesario actualizar su valor para cada instante de tiempo lo que representa un alto costo computacional, pero la mayor desventaja radica en que la discretización de la función Level Set es propensa a errores cuando la interfase sufre grandes deformaciones, lo que comúnmente se ve reflejado en una pérdida o ganancia de masa.

3.3. Local Level Set

Cuando se trabaja con el método Level Set, el costo computacional se incrementa debido que se agrega una ecuación (Ecuación Level Set, 3.6) que tiene que ser resuelta en todo el dominio computacional. Esta ecuación tiene la particularidad de ser de un orden de magnitud mayor que el problema original, por ejemplo, un problema de una dimensión es transformado en uno de dos dimensiones. De este modo en las tres dimensiones espaciales el costo computacional es muy alto para cada paso de tiempo.

El método utilizado en este trabajo es una variante del método Level Set propuesto inicialmente por Osher and Sethian en 1988 [42], que consiste en fijar unos límites determinados en los que será resuelta la ecuación Level Set, por lo tanto se trata de “Localizar” el método debido a lo cual esta alternativa recibe el nombre de Local Level Set.

En el método Local Level Set, en lugar de resolver la función Level Set en todo el dominio computacional, se restringen los cálculos a una banda estrecha con un espesor determinado en los alrededores de la interfase. Por lo que solo una fracción de la función Level Set es muestreada y actualizada en cada paso de tiempo. Cuando la interfase se mueve fuera de la banda, el cálculo se detiene y se construye una nueva banda con la interfase como centro.

Al localizar el método Level Set se tiene la ventaja de no afectar el resultado y reducir el costo computacional al no tener la necesidad de resolver la función Level Set en todo el dominio computacional, por lo que en este trabajo se adoptó el enfoque de [67] para la implementación de Local Level Set.

La técnica consiste en la construcción de bandas estrechas con un espesor finito que se actualizan siguiendo el movimiento de la interfase. El tamaño de las bandas es fijo y sus valores γ_i 's son múltiplos del tamaño de malla. Generalmente solo se requieren dos bandas, pero si el movimiento de la interfase implica cantidades delicadas como las derivadas de la curvatura, es necesario usar una tercer banda. Para este cálculo se consideran tres bandas γ_1 , γ_2 y γ_3 alrededor de la interfase donde $0 < \gamma_1 < \gamma_2 < \gamma_3$.

La elección del ancho de γ_i 's depende de los estenciles utilizados en la discretización espacial. La mayor cantidad de estenciles usados en este trabajo es de un esquema WENO de quinto orden. Por lo que los valores elegidos son: $\gamma_1 = 1,5\Delta l$, $\gamma_2 = 3\Delta l$ y $\gamma_3 = 6\Delta l$ donde Δl es el tamaño medio de paso espacial.

La idea principal en el método Local Level Set está compuesta por dos partes. La primera es actualizar la función Level Set solo en función del radio γ_3 en los alrededores del nivel cero establecido de acuerdo a la siguiente ecuación.

$$\frac{\partial \phi}{\partial t} + c(\phi)u \cdot \nabla \phi = 0 \quad (3.11)$$

Donde $c(\phi)$ es la función de corte que modifica el movimiento y se expresa por

$$c(\phi) = \begin{cases} 1 & \text{si } |\phi| \leq \gamma_2 \\ \frac{(|\phi| - \gamma_3)^2 (2|\phi| + \gamma_3 - 3\gamma_2)}{(\gamma_3 - \gamma_2)^3} & \text{si } \gamma_2 < |\phi| \leq \gamma_3 \\ 0 & \text{si } |\phi| > \gamma_3 \end{cases} \quad (3.12)$$

La función de corte c es usada para prevenir oscilaciones numéricas en la frontera de la banda y prevenir discontinuidades. El uso de una función de corte suave limita los errores en las regiones locales cerca de las discontinuidades que se propagan y da estabilidad. Este paso ayuda a mejorar el tiempo de cálculo, ya que no se necesita determinar ϕ y ϕ^1 para todas las ubicaciones en todo el dominio computacional.

El segundo paso del método requiere actualizar la banda y la evolución de la interfase en el tiempo, por lo que se regulariza la evolución mediante una ecuación de reinicialización, el valor de la función distancia de la función Level Set se puede utilizar para determinar la banda de cálculo en la próxima etapa de tiempo. Por lo que este paso no requiere de cálculos extras.

3.4. Función distancia y Re inicialización

La interfase es representada por una curva o superficie cuya evolución depende de factores como la posición espacial, la geometría y el tiempo. Esto implica que en ocasiones se generan geometrías complicadas como zonas planas o muy empinadas que dificultan el cálculo, cuando este tipo de errores ocurre el resultado es el trazo de contornos inexactos, pues la función Level Set pierde su regularidad o suavidad en regiones cercanas a la interfase, para evitar este tipo de errores se requiere reinicializar la función periódicamente.

La reinicialización significa que la función Level Set es redefinida para eliminar problemas de estabilidad numérica que pudieran provocar resultados erróneos pero manteniendo siempre la información de la interfase que va a ser transportada. Existen muchas técnicas de reinicialización [67], una de las mejores opciones para llevarla a cabo es definir a la función Level Set como una “función distancia con signo”.

La función distancia con signo cumple con todas las propiedades de la función Level Set y además tiene algunas propiedades nuevas que son muy útiles matemáticamente

hablando. De forma que redefinimos la función ϕ inicial como una distancia de \bar{x} a Γ como sigue:

$$\phi(\bar{x}, t = 0) = \pm d, \quad \bar{x} \in \partial\Omega \quad (3.13)$$

Ahora nuestra función ϕ toma una distancia desde la frontera $\partial\Omega$ con un signo positivo en el exterior, negativo en el interior y cero en la frontera de la superficie Γ .

$$\phi(\bar{x}) = \begin{cases} -d(\bar{x}) & \bar{x} \in \Omega^- \\ 0 & \bar{x} \in \partial\Omega \\ +d(\bar{x}) & \bar{x} \in \Omega^+ \end{cases} \quad (3.14)$$

Ahora definimos la distancia de un punto \bar{x} a la frontera $\partial\Omega$ como:

$$d(\bar{x}) = \min(|\bar{x} - \bar{x}_c|), \quad \bar{x}_c \in \partial\Omega \quad (3.15)$$

De la ecuación anterior se deduce que en $d(\bar{x}) = 0$ se ubica la frontera donde $\bar{x} \in \partial\Omega$. Por otra parte para un punto dado \bar{x} , encontramos el punto en la interfase definida por el conjunto $\partial\Omega$ más cercano a \bar{x} y lo llamamos \bar{x}_c .

El segmento de línea que va de \bar{x} a \bar{x}_c es la trayectoria más corta del punto \bar{x} a la interfase, por lo tanto es la trayectoria de máxima variación de la función d . Este segmento tiene una propiedad muy importante para nuestro método, pues si se evalúa $-\nabla d$ en cada punto de la trayectoria que va de \bar{x} a \bar{x}_c , el resultado es un vector que apunta de \bar{x} a \bar{x}_c . Y como d es una distancia euclidiana, entonces podemos decir que se cumple lo siguiente.

$$|\nabla d| = 1 \quad (3.16)$$

Y sabiendo que $|\phi(\bar{x})| = d(\bar{x})$ para todo \bar{x} , podemos reescribir la ecuación (3.16) como,

$$|\nabla \phi| = 1 \quad (3.17)$$

Existen tres razones importantes para mantener la función Level Set como una función distancia con signo. La primera es que los métodos que trabajan en una banda con un espesor finito, como el que se usa en este trabajo, son más exactos cuando se hace uso de

la función distancia. La segunda razón es que al hacer la función Level Set una función distancia con signo, se asegura que la interfase tiene un espesor fijo. Por último, al usar una función distancia, la función Level Set se comporta bien cerca de la interfase. No es difícil inicializar la función Level Set como una función distancia, sin embargo, cuando la función (3.6) cambia en el tiempo, no necesariamente seguirá siendo una función distancia. Por lo tanto, después de cierto tiempo, se debe reemplazar la función Level Set por una función distancia con signo sin cambiar su nivel cero. Lo anterior es conocido como el proceso de re inicialización y es una técnica diseñada para mantener la estabilidad de la función Level Set.

Con el fin de mantener ϕ como una función distancia con signo a través del tiempo se lleva a cabo un proceso de reinicialización basado en el enfoque introducido por [66] y además desarrollado en [67] donde se resuelve la siguiente ecuación pseudo-transitoria de Hamilton-Jacobi.

$$\frac{\partial \phi}{\partial t} + S(\phi_0)(|\nabla \phi| - 1) = 0 \quad (3.18)$$

En estado estable en el pseudo-tiempo τ , con la condición inicial

$$\phi(x, \tau = 0) = \phi_0 \quad (3.19)$$

Donde $S(\phi_0)$ es la función con signo definida como

$$S = \frac{\phi_0}{\sqrt{\phi_0^2 + |\nabla \phi_0|^2 \Delta x^2}} \quad (3.20)$$

Para este trabajo al igual que con la función Level Set se discretizo la ecuación (3.20) usando un esquema WENO (Weighted Essentially Non-Oscillatory) de quinto orden en el espacio y un esquema TVD-RK (Total Variation Diminishing, Runge-Kutta) de tercer orden en el pseudo-tiempo. Lo anterior se debe a la importancia de usar esquemas de discretización de alto orden con el objetivo de minimizar los errores numéricos.

3.5. Ghost Fluid Method

Al trabajar con flujos multifásicos se presentan discontinuidades en la superficie libre, debidas a que se tienen en contacto dos fluidos diferentes o en diferentes estados de agregación, por lo que propiedades como la densidad y la viscosidad son tan dispares que es

muy difícil el manejo de las ecuaciones de gobierno en esta zona. El método propuesto por Fedkiw conocido como el método del fluido fantasma (GFM, Ghost Fluid Method) [72], es una alternativa que permite capturar la interfase evitando las oscilaciones no físicas y minimizando los problemas asociados a las discontinuidades en las variables.

El método se fundamenta en la introducción de dos conjuntos de variables o “campos” para cada fluido por separado, la frontera entre cada fluido esta definida por la función Level Set. Para cada campo, el “nivel cero” de la función Level Set separa una zona “real” y una zona “fantasma” de manera que un fluido real 1 coexiste con un fluido fantasma 2 y viceversa como se muestra en la Figura 3.3.

De esta manera tenemos dos campos de variables

$$[\rho_f, u_f, P_f]$$

donde

$$f = \begin{cases} 0 & \text{aire (suave)} \\ 1 & \text{agua (rígido)} \end{cases}$$

Además se cumple que, $\phi > 0$ para la región de aire y $\phi < 0$ corresponde al agua. De esta manera dependiendo del signo podemos tener aire “real” y “fantasma” o agua “real” y “fantasma”. Donde aire “real” coexiste con agua “fantasma” y aire “fantasma” con agua “real”.

Con esto, para cada paso de tiempo se necesita rellenar las zonas “fantasma” de cada fluido con variables de campo y luego avanzar la solución en el tiempo para cada campo resolviendo por separado las ecuaciones de conservación. Para lo que existen dos algoritmos llamados el acoplamiento “suave” y el acoplamiento “rígido”.

En el acoplamiento suave la presión y velocidad son consideradas funciones “continuas” a través de la interfase, por lo que los puntos “fantasma” son llenados copiando valores del fluido real. Por otro lado, propiedades como la densidad y la viscosidad cinemática son tratadas como funciones discontinuas y sus valores se ajustan a los del campo particular, es decir, son “extrapolados” de la zona real a la zona fantasma.

Sin embargo cuando se modelan flujos con fluidos de densidades muy diferentes como agua y aire, el acoplamiento suave puede fallar. Esto es debido a que pequeños errores numéricos en presión pueden causar grandes errores en velocidad para el fluido más ligero.

Cuando tenemos flujos con densidades muy dispares el camino adecuado es el acopla-

miento rígido. Este algoritmo se basa en considerar al fluido más denso como el rígido en este caso el agua y el aire es el fluido suave pues es el que tiene la densidad más baja. La presión en la zona fantasma del fluido rígido es copiada de la parte real del fluido suave, por otro lado la presión en la zona fantasma del fluido suave es extrapolada de la región real del mismo fluido suave. La velocidad total de la zona fantasma en el fluido rígido es extrapolada de su parte real, mientras que la velocidad de la zona fantasma del fluido suave es directamente copiada de la parte real del fluido rígido como se muestra en la Figura 3.3.

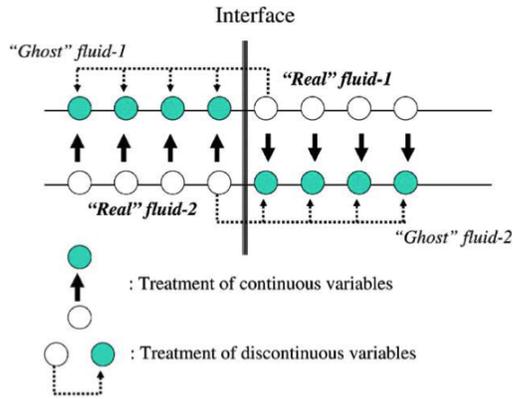


Figura 3.3: Formulación del método Ghost Fluid en una dimensión las variables continuas son copiadas de la parte real del fluido rígido a la fantasma del fluido suave, mientras que las variables discontinuas son extrapoladas de la parte real a la parte fantasma del mismo fluido

El criterio utilizado para determinar cuando se debe usar el acoplamiento suave o el rígido se basa en la relación de densidad de los fluidos involucrados en el flujo modelado. Cuando la relación de densidad es menos que 10:1 se usa un acoplamiento suave, el acoplamiento rígido se usa cuando la relación de densidad es de 1000:1, por lo que si se trabaja con flujos de alta relación de densidad es esencial usar el acoplamiento rígido para tener resultados satisfactorios.

Cabe mencionar que el uso del método Ghost Fluid es importante para eliminar los errores de divergencia asociados a saltos de densidad a través de la interfase y que el algoritmo utilizado para este trabajo es el acoplamiento rígido.

3.6. Conservación de masa

El código bifásico utilizado en este trabajo se basa principalmente en el método Level Set debido a todas las ventajas que ofrece y que ya se han mencionado, sin embargo

también existen desventajas inherentes al método que es necesario controlar. La principal desventaja es que la función Level Set no conserva la masa.

Para remediar esta situación, se utiliza el método VOF (Volume Of Fluid) [31] cuya función ψ conserva la masa de cierto fluido dentro de una celda computacional. Con la ayuda de ψ se modifica la función Level Set y se impone localmente la conservación de masa (MCLS, Mass-Conserving Level Set) [62] [63].

El método del volumen de fluido (VOF) [31], es una técnica euleriana que desde su introducción ha sido usada y estudiada en muchos trabajos. En VOF el dominio computacional es dividido en pequeñas celdas y dado que los fluidos o las fases de fluido no pueden mezclarse, podemos asignar a cada fluido una fracción de volumen.

A cada celda del dominio computacional se le asigna un valor en base a la fracción de material que hay dentro de ella. El valor igual a cero indica la presencia de un solo fluido en la celda, un valor igual a uno indica que la celda está llena del otro fluido; por último los valores en las celdas de fracciones entre cero y uno indican la presencia de la interfase y el valor de dichas celdas ofrece una indicación de las proporciones de cada fluido que ocupan el volumen de cada celda.

Matemáticamente podemos definir a la fracción volumétrica del fluido como ψ , entonces las tres posibilidades quedan de la siguiente manera [8][44].

$$\begin{array}{ll} \psi = 0 & \text{Celda llena de fluido 1} \\ \psi = 1 & \text{Celda llena de fluido 2} \\ 0 < \psi < 1 & \text{Celda con interfase} \end{array}$$

A partir de las fracciones de volumen en cada celda se puede determinar la posición de la interfase realizando una reconstrucción de la misma en las celdas atravesadas por la interfase como se muestra en la Figura 3.4, bajo la suposición de que la interfase se encuentra bien alineada con las direcciones del plano cartesiano.

Para representar la fracción de volumen ψ sólo se requiere una ecuación escalar (3.21), que tiene la misma forma de la ecuación de la continuidad y que al ser resuelta se obtiene la evolución de la interfase.

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\psi \bar{u}) = 0 \quad (3.21)$$

Debido a que la función es resuelta de manera conservativa el método conserva la masa de manera exacta. Sin embargo, aunque el método tiene excelentes propiedades conservati-

0,95	0,8	0,3	0	0
1	1	1	0,6	0
1	1	1	1	0,3
1	1	1	1	0,8
1	1	1	1	0,95

Figura 3.4: Interfase, con fracciones de volumen

vas, las aproximaciones en las propiedades geométricas en la interfase y la reconstrucción de la misma son tareas computacionalmente caras, especialmente en simulaciones 3D. Además, debido a las discontinuidades de ψ , se requiere el uso de métodos numéricos para reducir la difusión numérica que provoca manchas en la interfase, estas técnicas adicionales pueden ser esquemas de discretización de alto orden como QUICK y TVD; o correctores como el esquema Donnor Acceptor [38][39][40] [41].

3.6.1. Corrección a la función LS

El método MCLS se basa en la idea de mejorar la conservación de masa de una función explícita f que relaciona directamente la función LS con la función VOF cuyo gradiente es

$$\psi = f(\phi, \nabla\phi) \quad (3.22)$$

Las correcciones de conservación de masa se aplican a la función LS ϕ , usando la función VOF ψ . Lo anterior se logra con la aplicación de tres pasos:

1. Cálculo la función VOF a partir de la función LS $\psi^n = f(\phi^n, \nabla\phi^n)$
2. Advección de la función VOF (ψ^{n+1})
3. Corrección de la función ϕ^{n+1} con la advección de la función VOF ψ^{n+1} , de tal forma que $\phi^{n+1} = f(\psi^{n+1}, \nabla\phi^{n+1})$.

A continuación se explica a detalle cada paso.

3.6.1.1. Paso 1

La función VOF ψ en una celda computacional en términos de la función LS ϕ se define como

$$\Psi_k = \frac{1}{\text{vol}(\Omega_k)} \int_{\Omega_k} H(\Phi) d\Omega \quad (3.23)$$

Donde Ω_k es una celda en la malla, x_k corresponde al nodo central de Ω_k y H es la fusión escalón de Heaviside. La función Level Set ϕ es linealizada alrededor de x_k , lo que lleva a

$$\varphi = \phi_k + \nabla\phi_k \cdot (x - x_k) \quad (3.24)$$

Donde $\nabla\phi_k$ se aproxima por diferencias centrales. La fracción de volumen ψ_k corta la celda Ω_k por el plano definido por $\varphi = 0$ que es calculado mapeando un cubo unitario Ω con coordenadas $(\xi, \eta, \zeta) \in (1, 1)^3$. La ecuación (3.24) se puede reescribir como

$$\varphi = \phi_k + D_\xi\xi + D_\eta\eta + D_\zeta\zeta \quad (3.25)$$

Donde

$$\begin{aligned} \xi &= \frac{(x-x_k)}{\Delta x} \\ \eta &= \frac{(y-y_k)}{\Delta y} \\ \zeta &= \frac{(z-z_k)}{\Delta z} \end{aligned}$$

Los ejes son elegidos para cumplir que

$$D_\xi \geq D_\eta \geq D_\zeta \geq 0 \quad (3.27)$$

Lo que limita el numero de posibles topologías de la interfase en el cubo, como consecuencia de

$$\begin{aligned}
D_\xi &= \max(|D_x|, |D_y|, |D_z|) \\
D_\zeta &= \min(|D_x|, |D_y|, |D_z|) \\
D_\eta &= |D_x| + |D_y| + |D_z| - D_\xi - D_\zeta
\end{aligned} \tag{3.28}$$

donde

$$\begin{aligned}
d_x &= \Delta x \left. \frac{\partial \phi}{x} \right|_k \\
d_y &= \Delta y \left. \frac{\partial \phi}{y} \right|_k \\
d_z &= \Delta z \left. \frac{\partial \phi}{z} \right|_k
\end{aligned} \tag{3.29}$$

La función VOF se obtiene geoméricamente calculando el volumen relativo encerrado por el plano de corte en Ω_k definido por el nivel cero de la linealización de función LS. La topología del plano de corte cambia cuando pasa por una esquina del cubo.

Después de algunas manipulaciones matemáticas la función f es calculada geoméricamente a partir del volumen de recortado del cubo. La función f es evaluada como

$$f \begin{cases} \frac{A}{6D_\xi D_\eta D_\zeta} & \phi \leq 0 \\ 1 - f(-\phi, \Delta \phi) & \phi > 0 \end{cases} \tag{3.30}$$

donde,

$$\mathcal{A} = \max(\phi_A, 0)^3 - \max(\phi_B, 0)^3 - \max(\phi_C, 0)^3 - \max(\phi_D, 0)^3 + \max(\phi_E, 0)^3 \tag{3.31}$$

y

$$\begin{aligned}
\Phi_A &= \Phi_k + (D_\xi + D_\eta + D_\zeta) / 2 \\
\Phi_B &= \Phi_k + (D_\xi + D_\eta - D_\zeta) / 2 \\
\Phi_C &= \Phi_k + (D_\xi - D_\eta + D_\zeta) / 2 \\
\Phi_D &= \Phi_k + (-D_\xi + D_\eta + D_\zeta) / 2 \\
\Phi_E &= \Phi_k + (D_\xi - D_\eta - D_\zeta) / 2
\end{aligned} \tag{3.32}$$

Φ_A representa el valor de $\Phi(x_A)$ en la esquina A y de manera similar para las otras esquinas. La Figura 3.5 es una representación geométrica del cálculo de f .

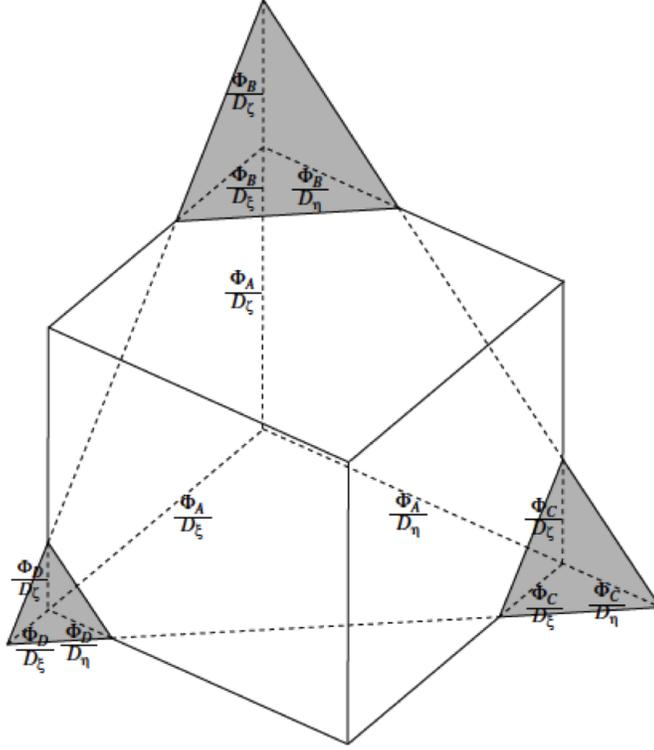


Figura 3.5: Volumen para $\Phi \leq 0$

3.6.1.2. Paso 2

La función LS ϕ corregida con la función VOF ψ es advetada por medio de los esquemas WENO y Runge-Kutta.

3.6.1.3. Paso 3

La función LS inicial ϕ^{n+1} es corregida con la función VOF ψ^{n+1} de tal forma que

$$|f(\Phi_k^{n+1}, \nabla \Phi_k^{n+1}) - \Psi_k^{n+1}| \leq \varepsilon \quad \forall k = 1, 2, \dots \quad (3.33)$$

Donde ε es una tolerancia (para este trabajo $\varepsilon = 10^{-6}$), ϕ_k^{n+1} es la función Level Set en el nuevo tiempo y f es la función VOF. Cabe resaltar que debido al comportamiento de ψ no hay una solución única para ϕ . Sin embargo, se busca una pequeña corrección a ϕ^* , donde ϕ^* viene de la advección de la función Level Set. Una solución ϕ es encontrada cuando se realiza la siguiente iteración hasta la convergencia: Dejar sin modificar ϕ en un punto de la malla cuando la restricción de VOF se satisface y hacer correcciones localmente cuando no se satisface. Esto se logra utilizando la función g que es la inversa de f en la ecuación (3.30) con respecto al argumento ω_k ,

$$f(g(\psi, \nabla\phi), \nabla\phi) = \psi \quad (3.34)$$

El proceso anterior mejora la conservación de masa.

3.7. Discretización de las ecuaciones VOF y LS

La ecuación Level Set 3.10, es una ecuación hiperbólica, por lo que durante su evolución existe la posibilidad de desarrollar zonas planas o pendientes muy empinadas que eventualmente conducirán a discontinuidades en la solución.

Para tratar correctamente la evolución de la función Level Set 3.10, los esquemas numéricos deben tener la precisión suficiente. El esquema elegido también debe limitar los problemas numéricos de difusión y dispersión, con el fin de limitar la pérdida de masa y los errores en la velocidad de advección para todas las escalas de la interfase, especialmente en presencia de grandes deformaciones.

Tomando en cuenta lo anterior y sabiendo que nuestro código va a estar expuesto a errores pequeños, pero que pueden tener influencia significativa principalmente por el hecho que se combina una gran cantidad de técnicas y se tienen procesos de reinicialización, se eligió la implementación de esquemas de discretización espacial y temporal de alto orden.

Para resolver la ecuación (3.9) se usa el esquema de quinto orden WENO (Weighted Essentially Nonoscillatory; Jiang and Peng, 2000) [69], para la discretización espacial y un esquema Runge–Kutta TVD de tercer orden (Fedkiw et al., 1998) [68] para la discretización temporal.

3.7.1. Esquema WENO

Debido a que el método Level Set es más sensible a la exactitud espacial en relación a la temporal, se necesita una discretización espacial de orden más alto que el de la discretización temporal. Por lo que la discretización espacial es llevada a cabo con un esquema de ponderaciones esencialmente no oscilatorias WENO (Weighted Essentially Non-Oscillatory) de quinto orden .

La técnica WENO [69] es una mejora a la técnica conocida como ENO que permite tener un orden de exactitud más alto en la parte suave de la función y un comportamiento al menos igual a la técnica ENO en las discontinuidades, lo que representa un notable incremento en exactitud además de eliminar o aliviar problemas de inestabilidad [71].

En la técnica ENO el objetivo es aproximar la función a un polinomio de la forma,

$$\Phi(x) = Q_0(x) + Q_1(x) + Q_2(x) + Q_3(x) \quad (3.35)$$

que puede ser diferenciado y evaluado en x_i para encontrar $(\Phi_x^-)_i$ y $(\Phi_x^+)_i$ donde el signo indica si $\Phi(x)$ se mueve para la izquierda o la derecha, además el término $Q_0(x)$ desaparece después de derivar por lo que tenemos

$$\Phi(x_i) = Q_1'(x_i) + Q_2'(x_i) + Q_3'(x_i) \quad (3.36)$$

Cuando se calcula $(\Phi_x^-)_i$ con la técnica ENO de tercer orden tenemos tres posibles aproximaciones que son,

$$\Phi_x^1 = \frac{a}{3} - \frac{7b}{6} + \frac{11c}{6}, \quad (3.37)$$

$$\Phi_x^2 = -\frac{b}{6} + \frac{5c}{6} + \frac{d}{3}, \quad (3.38)$$

$$\Phi_x^3 = \frac{c}{3} + \frac{5d}{6} - \frac{e}{6} \quad (3.39)$$

El objetivo del método ENO es escoger la mejor opción de las tres anteriores, con la que se puede construir la función más suave posible con el menor error.

Por otro lado la técnica WENO es una combinación convexa de las tres opciones del método ENO, de este modo se utiliza toda la información que se tiene, lo que resulta en un mayor orden de exactitud en las regiones suaves de la función. Al polinomio resultante de la técnica WENO se le conoce como combinación convexa y tiene la forma,

$$\Phi = \omega_1 \left(\frac{a}{3} - \frac{7b}{6} + \frac{11c}{6} \right) + \omega_2 \left(-\frac{b}{6} + \frac{5c}{6} + \frac{d}{3} \right) + \omega_3 \left(\frac{c}{3} + \frac{5d}{6} - \frac{e}{6} \right) \quad (3.40)$$

donde para el lado izquierdo Φ_0^-

$$a = \frac{\Phi_{i0-2} - \Phi_{i0-3}}{\Delta x} \quad (3.41)$$

$$b = \frac{\Phi_{i0-1} - \Phi_{i0-2}}{\Delta x} \quad (3.42)$$

$$c = \frac{\Phi_{i0} - \Phi_{i0-1}}{\Delta x} \quad (3.43)$$

$$d = \frac{\Phi_{i0+1} - \Phi_{i0}}{\Delta x} \quad (3.44)$$

$$e = \frac{\Phi_{i0+2} - \Phi_{i0+1}}{\Delta x} \quad (3.45)$$

mientras que para el lado derecho Φ_0^+

$$a = \frac{\Phi_{i0+3} - \Phi_{i0+2}}{\Delta x} \quad (3.46)$$

$$b = \frac{\Phi_{i0+2} - \Phi_{i0+1}}{\Delta x} \quad (3.47)$$

$$c = \frac{\Phi_{i0+1} - \Phi_{i0}}{\Delta x} \quad (3.48)$$

$$d = \frac{\Phi_{i0} - \Phi_{i0-1}}{\Delta x} \quad (3.49)$$

$$e = \frac{\Phi_{i0-1} - \Phi_{i0-2}}{\Delta x} \quad (3.50)$$

La clave es encontrar los pesos adecuados para combinación convexa dando más peso a los polinomios más suaves y por supuesto, si cualquiera de las tres aproximaciones interpola a través de una discontinuidad, se le da un peso mínimo en la combinación convexa con el fin de minimizar su contribución y los errores resultantes. Los pesos de la función se ponderan asignando un valor de $0 \leq \omega_k \leq 1$ en la ecuación (3.40), cumpliendo con $\omega_1 + \omega_2 + \omega_3 = 1$.

En las regiones suaves de la función las tres aproximaciones son tomadas en cuenta para mejorar la exactitud local, por lo que un método ENO de tercer orden se convierte en un método WENO de quinto orden. En estas zonas los pesos se toman típicamente como $\omega_1 = 0,1$, $\omega_2 = 0,6$ y $\omega_3 = 0,3$ con lo que se obtiene una aproximación satisfactoria de quinto orden a Φ_x .

Cuando se está en las regiones que tienen discontinuidades los valores anteriores de ω_k pueden llevar a graves errores de exactitud, por lo que es mejor dar valores de $\omega_k = 0$ o $\omega_k = 1$ igual que la técnica ENO en tanto que el orden de exactitud en este caso decrece de quinto hasta tercer orden.

Para definir los pesos de ω_k , se estima la suavidad de las ecuaciones (3.37), (3.38) y (3.39) como sigue,

$$S_1 = \frac{13}{12} (a - 2b + c)^2 + \frac{1}{4} (a - 4b + 3c)^2 \quad (3.51)$$

$$S_2 = \frac{13}{12} (b - 2c + d)^2 + \frac{1}{4} (b - d)^2 \quad (3.52)$$

$$S_3 = \frac{13}{12} (c - 2d + e)^2 + \frac{1}{4} (3c - 4d + e)^2 \quad (3.53)$$

Respectivamente. Con los tres indicadores de suavidad anteriores obtenemos,

$$\alpha_1 = \frac{0,1}{(S_1 + \epsilon)^2} \quad (3.54)$$

$$\alpha_2 = \frac{0,6}{(S_2 + \epsilon)^2} \quad (3.55)$$

$$\alpha_3 = \frac{0,3}{(S_3 + \epsilon)^2} \quad (3.56)$$

con $\epsilon = 10^{-6} \max\{a^2, b^2, c^2, d^2, e^2\} + 10^{-99}$, donde 10^{-99} evita la división entre cero de α_k [43]

Se tiene una solución suave cuando se tienen valores de S_k lo suficientemente pequeños comparados con ϵ , dado que las ecuaciones (3.54), (3.55) y (3.56) se convierten en $\alpha_1 \approx 0,1\epsilon^{-2}$, $\alpha_2 \approx 0,6\epsilon^{-2}$ y $\alpha_3 \approx 0,3\epsilon^{-2}$. Normalizando α_k tenemos los pesos,

$$\omega_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3} \quad (3.57)$$

$$\omega_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2 + \alpha_3} \quad (3.58)$$

$$\omega_3 = \frac{\alpha_3}{\alpha_1 + \alpha_2 + \alpha_3} \quad (3.59)$$

Cuando no se tiene una solución suave, los valores de S_k son muy grandes y por lo tanto las α_k se vuelven pequeñas. Cuando dos de las S_k son relativamente grandes, las α_k se vuelven pequeñas en relación a la restante, dado que el comportamiento será similar al esquema ENO, donde se escoge sólo un polinomio entre (3.37), (3.38) y (3.39).

Se tiene un caso especial cuando las tres S_k son grandes, pues la función se mantiene suave siempre y cuando las tres S_k sean aproximadamente del mismo tamaño, en caso contrario ninguna de las tres alternativas (3.37), (3.38) y (3.39) es útil lo que es un problema tanto en ENO como en WENO, afortunadamente esto sólo sucede en casos puntuales tanto en espacio como tiempo y estos métodos se autoreparan cuando esta situación desaparece.

3.7.2. Esquema Runge-Kutta

Generalmente no es necesario utilizar esquemas de discretización temporal de alto orden, pues los errores de truncado en el tiempo no producen un error significativo en la solución numérica. Pero cuando es necesario, se pueden usar esquemas numéricos de alto

orden con el fin de obtener una solución más exacta, una de estas técnicas es la conocida como discretización de variación total decreciente (TVD, Total Variation Diminishing) Runge-Kutta TVD [68] [43].

El esquema Runge-Kutta TVD de tercer orden se escribe como sigue. Si la discretización espacial del término de advección se escribe como $-(u \cdot \nabla)\phi = L(\phi)$, la ecuación (3.11) puede escribirse como

$$\frac{\partial\phi}{\partial t} = L(\phi) \quad (3.60)$$

El esquema TVD Runge-Kutta de primer orden es el método de Euler progresivo

$$\phi^{n+1} = \phi^n + \Delta t L(\phi^n) \quad (3.61)$$

El esquema TVD Runge-Kutta de segundo orden es el metodo predictor - corrector de Heun's

$$\begin{cases} \phi^* & = \phi^n + \Delta t L(\phi^n) \\ \phi^{n+1} & = \phi^n + \Delta t \left(\frac{1}{2}(\phi^n) + \frac{1}{2}(\phi^*) \right) \end{cases} \quad (3.62)$$

El esquema de tercer orden TVD Runge-Kutta esta dado por

$$\begin{cases} \phi^1 & = \phi^n + \Delta t L(\phi^n) \\ \phi^2 & = \frac{3}{4}\phi^n + \frac{1}{4}(\phi^1 + \Delta t L(\phi^1)) \\ \phi^{n+1} & = \frac{1}{3}\phi^n + \frac{2}{3}(\phi^2 + \Delta t L(\phi^2)) \end{cases} \quad (3.63)$$

Generalmente se recomienda el método TVD de tercer orden, dado que tiene la mayor exactitud y la mayor región de estabilidad de los esquemas TVD. Debido a su amplia zona de estabilidad para un paso de tiempo pequeño se garantiza que el método sera linealmente estable para una gran variedad de problemas. En contraste, los metodos de primer y segundo orden requieren algunos terminos difusivos espaciales para ser estables.

Capítulo 4

Validación

4.1. Introducción

Una vez que se tiene un código numérico surgen cuestiones como: ¿Hasta qué punto se puede confiar en la exactitud de los resultados que entrega? ¿Cómo puede validarse críticamente un modelo numérico? ¿Qué se puede deducir cuando se realiza una predicción de algún proceso físico con un modelo que ha sido validado?

Para dar respuesta a las preguntas anteriores, el Instituto Americano de Aeronáutica y Astronáutica (AIAA, American Institute of Aeronautics and Astronautics) y la Sociedad Americana de Ingenieros Mecánicos (ASME, American Society of Mechanical Engineers) han propuesto una metodología de verificación y validación de códigos numéricos que es utilizada en este trabajo [73], [74].

El proceso de validación consiste principalmente en comprobar el grado de exactitud que tiene nuestra simulación con el “mundo real” mediante la comparación de resultados obtenidos numéricamente con resultados obtenidos experimentalmente.

Consideramos a los datos experimentales como la observación del mundo real de una manera controlada, por lo que una de las partes más importantes para llevar a cabo la validación es la selección del caso experimental con el que se va a comparar el código pues este debe de cumplir con una serie de requisitos como son [75]:

- Un experimento de validación debe estar diseñado para capturar la esencia física de interés de una manera sencilla que evite al máximo caer en errores conceptuales o de medición.

- Debe de tener claramente especificados todos los datos físicos como por ejemplo condiciones iniciales, de contorno y condiciones límite; condiciones de rugosidad en las superficies, además de imperfecciones o desajustes en los componentes.
- Ubicación exacta de toda la instrumentación así como sus características como escala y rango de medición.
- Mediciones precisas de las dimensiones reales del modelo, se debe señalar claramente la complejidad de la geometría y procesos del flujo físico, así como su naturaleza temporal y espacial.
- Se debe tener un reporte detallado de los resultados obtenidos que incluya incertidumbres y porcentajes de error.

Existen una serie de experimentos que se han convertido en los puntos de partida para validar simulaciones numéricas en CFD [74], así como técnicas y consejos para desarrollar experimentos nuevos o utilizar adecuadamente los datos proporcionados por experimentos comprobados [74].

Aun con todos los apoyos anteriormente mencionados es importante tener en cuenta que todo experimento incluso los realizados con herramientas de alta precisión y de manera altamente controlada, tienen errores que no se pueden evitar. La magnitud de estos errores está relacionada con la complejidad del experimento así como con la comprensión de los conocimientos teóricos de los procesos involucrados en él. Por lo tanto se recomienda que la elección del experimento a utilizar para la validación de un código numérico este sustentada por sólidos conocimientos de los procesos físicos y matemáticos, además de optar siempre por el experimento menos complejo posible.

Una vez que se seleccionó el experimento y se va a llevar a cabo la validación, los puntos más importantes a evaluar son:

- Se debe comprobar que la solución de la simulación es coherente. Por ejemplo, se debe cumplir con todas la leyes de gobierno como conservación de masa y cantidad de movimiento.
- Debe de haber concordancia entre los datos obtenidos en la simulación y los experimentales.
- Se deben considerar los errores de incertidumbre en la simulación. Estos son errores inherentes a los métodos numéricos y se deben a los algoritmos de discretización, cuestiones de hardware y software, falta de comprensión de procesos físicos complejos como por ejemplo la turbulencia, entre otros.

Cuando el proceso de validación se ha completado debe quedar claro que los resultados solo sirven para obtener conclusiones del desempeño del código numérico evaluado.

Un código que ha sido validado es por lo tanto más confiable y permite al analista contar con datos útiles para el diseño de sistemas que involucran fluidos, pero tomando en cuenta en todo momento que se trata de una herramienta complementaria que debe usarse con precaución.

Considerando lo expuesto anteriormente se eligieron tres casos para poner a prueba el código numérico presentado en este trabajo. En cada experimento se aumenta la dificultad de simulación, lo que nos permite conocer el comportamiento del código ante diferentes situaciones.

El primero caso es una rotura de presa o dam-break sencillo, el segundo consta del rompimiento de una columna de agua sobre una cama del mismo fluido; por último se simuló el rompimiento de una columna de agua que impacta contra un bloque sólido.

Los resultados se muestran a continuación.

4.2. Resultados

4.3. Dam-Break

Un dam-break o rotura de presa, es el derrame súbito de una columna de líquido que está confinada en un tanque con paredes rígidas de dimensiones conocidas. Es uno de los modelos físicos más utilizados para la evaluación de técnicas numéricas que modelan superficies libres e interfases móviles. Ya que ha sido analizado tanto numérica [15], [16], [18], [50], [53], como experimentalmente [76], [77], se cuenta con información suficiente para poder comparar los resultados con trabajos previos y determinar el desempeño del código numérico utilizado en este trabajo.

Típicamente el punto de partida para la validación de un método numérico son los experimentos realizados por Martin Moyce (1952) [76] y Koshizuka et al. (1995) [77]. A pesar de que en estos trabajos experimentales no se obtuvo la forma exacta de la interfase, se cuenta con datos importantes como la velocidad del borde delantero de la onda y la reducción de la altura de la columna de líquido, así como fotografías que muestran el desarrollo de la interfase a través del tiempo.

Para este trabajo dimensiones del dominio computacional corresponden con la usadas en el experimento realizado por Koshizuka et al. (1995) [77] y el arreglo experimental se

muestra en la Figura 4.1.

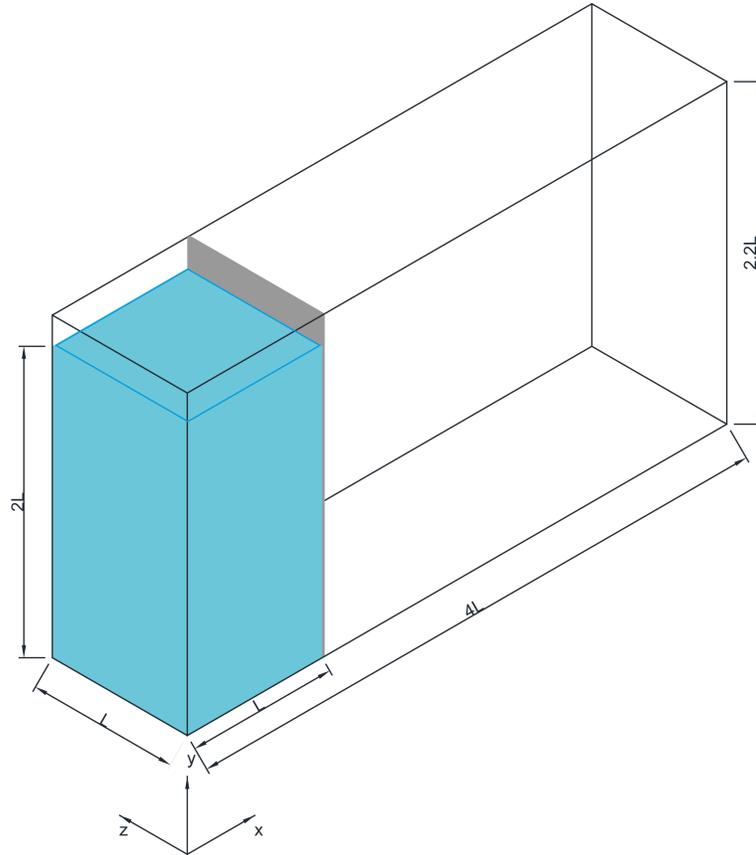


Figura 4.1: Arreglo Experimental

El dominio experimental consta de un tanque de vidrio con una longitud de $4L$, donde $L = 14,6\text{cm}$ que da un total de $0,584\text{m}$, la columna de agua tiene una base de $0,146\text{m}$ ($1L$) y una altura de $0,292\text{m}$ ($2L$). Inicialmente dicha columna es detenida por una placa vertical que se retira en un tiempo de $t = 0,0\text{s}$ (aunque en realidad es de alrededor de $t = 0,05\text{s}$).

La columna de agua parte del reposo y al retirar la placa vertical el agua cae hacia la derecha por efecto de la gravedad ocupando el fondo del tanque, por lo que eventualmente volverá al reposo. En las etapas iniciales, el flujo es dominado por los efectos de inercia, mientras que los efectos viscosos toman importancia rápidamente a medida que el agua tiende al reposo, se considera que la tensión superficial es despreciable.

El mallado está íntimamente ligado con la calidad de la simulación ya que usualmente una densidad de mallado elevada disminuye notablemente los errores normales que se generan en la simulación debido a los errores de truncado y que se presentan en todos los modelos numéricos. Por otro lado el contar con una malla muy refinada también significa un

alto costo computacional, así como un tiempo de simulación más elevado que es ocasiones es tan excesivo que se vuelve imposible llevar a la práctica.

Es por lo anterior que se deben encontrar las características ideales del mallado para cada código numérico, buscando lo que se conoce como independencia de mallado, es decir, encontrar la densidad de malla en la que los resultados dejan de cambiar significativamente. Por lo que el primer experimento va a servir para encontrar dichas características de manera sencilla por ser el experimento más simple de todos, una vez conocidos los parámetros de mallado apropiados para el código se tomaran como punto de partida para las simulaciones siguientes.

Para llevar a cabo lo anteriormente mencionado se realizaron 4 simulaciones de una rotura de presa con las dimensiones del experimento de Koshizuka et al. (1995), haciendo las pruebas hasta 0,6s con pasos de tiempo de 0,006s.

Al tratarse de una simulación tridimensional, las mallas se formaron en los ejes x, y, z; y quedaron de la siguiente manera:

- Prueba 1: $150 \times 83 \times 20 = 249,000$ [Nodos]
- Prueba 2: $200 \times 109 \times 25 = 545,000$ [Nodos]
- Prueba 3: $300 \times 165 \times 35 = 1,732,500$ [Nodos]
- Prueba 4: $350 \times 191 \times 44 = 2,941,400$ [Nodos]

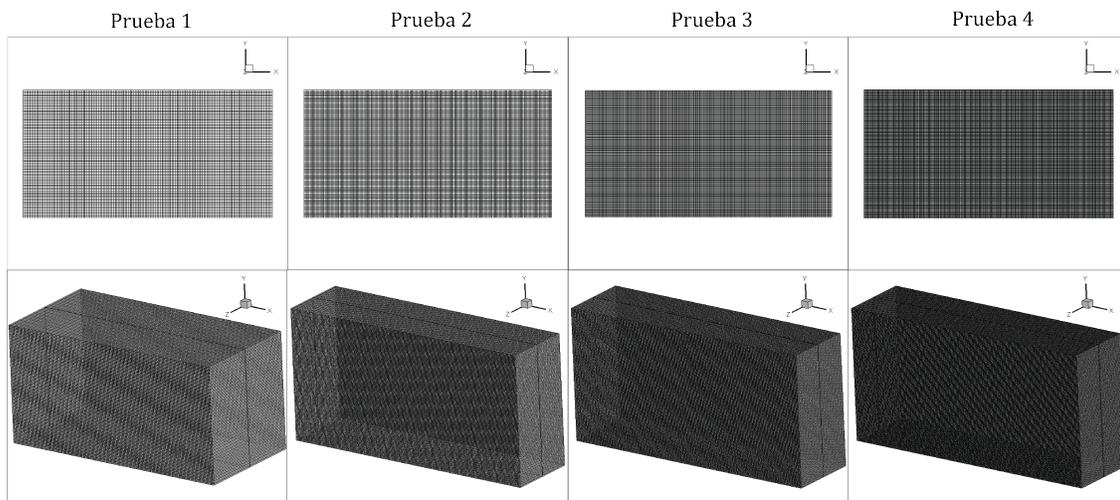


Figura 4.2: Mallado para cada prueba

La Figura 4.2 muestra los mallados correspondientes a cada prueba.

En la Figura 4.3 se muestran las gráficas comparativas adimensionales de la evolución temporal del frente del fluido desde que se retira la compuerta hasta que toca la pared derecha del dominio, para todas las mallas utilizadas. Es importante mencionar que todas las simulaciones se realizaron incluyendo el efecto de retirar la compuerta en lugar de quitarla instantáneamente; este pequeño detalle es determinante para la exactitud en los resultados como se verá más adelante.

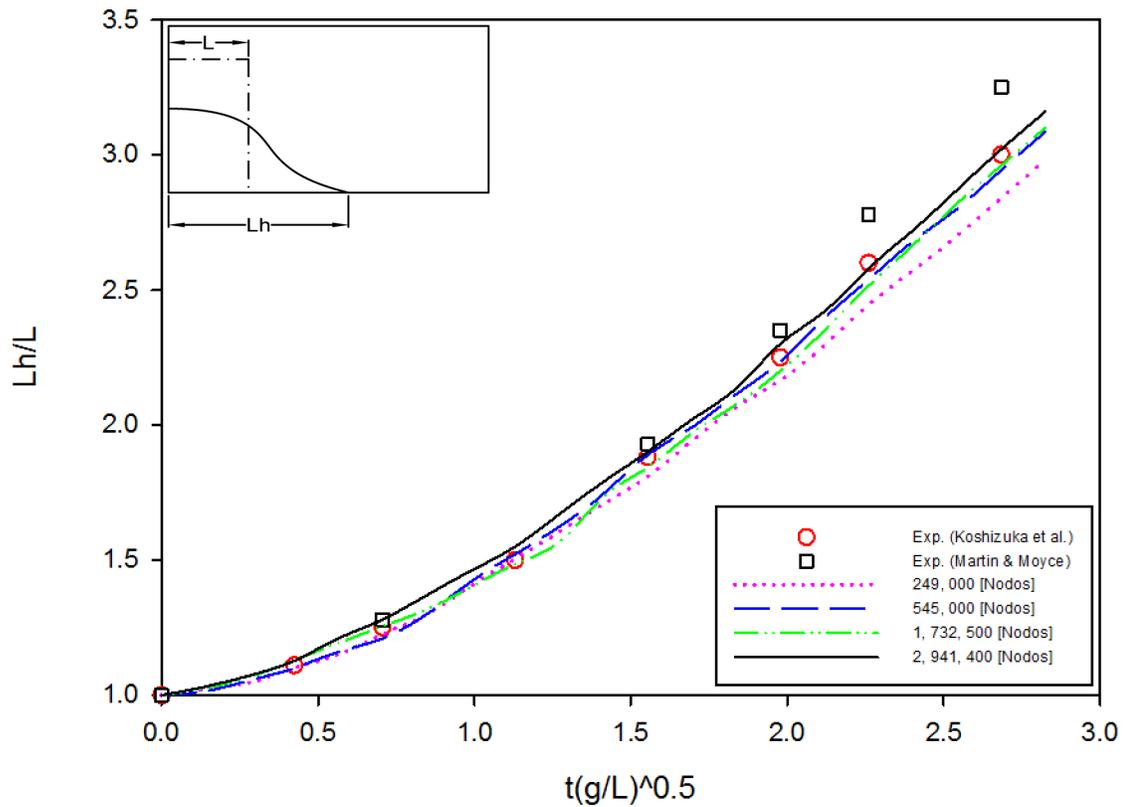


Figura 4.3: Movimiento del frente de onda

La gráfica contiene resultados obtenidos experimentalmente por Martin Moyce (1952) y Koshizuka et al. (1995) con la finalidad de compararlos con los resultados obtenidos por el método LS-VOF. Puede observarse que la densidad de mallado tiene influencia en el avance del frente de onda, sin embargo, la diferencia entre cada simulación no es tan grande y la variación entre cada grafica es menor a medida que aumenta la densidad de malla. Lo anterior confirma que para mallas refinadas el código numérico se vuelve independiente de la discretización espacial, por lo que no tendría caso trabajar con mallas más grandes

pues el resultado no mejoraría mucho y el tiempo de simulación se incrementaría. Cabe mencionar que en todos los casos se obtuvieron resultados buenos y que como era de esperarse los resultados más exactos corresponden a la simulación con la mayor densidad de mallado donde se tiene una amplia correspondencia sobre todo con el caso experimental de Koshizuka et al. (1995).

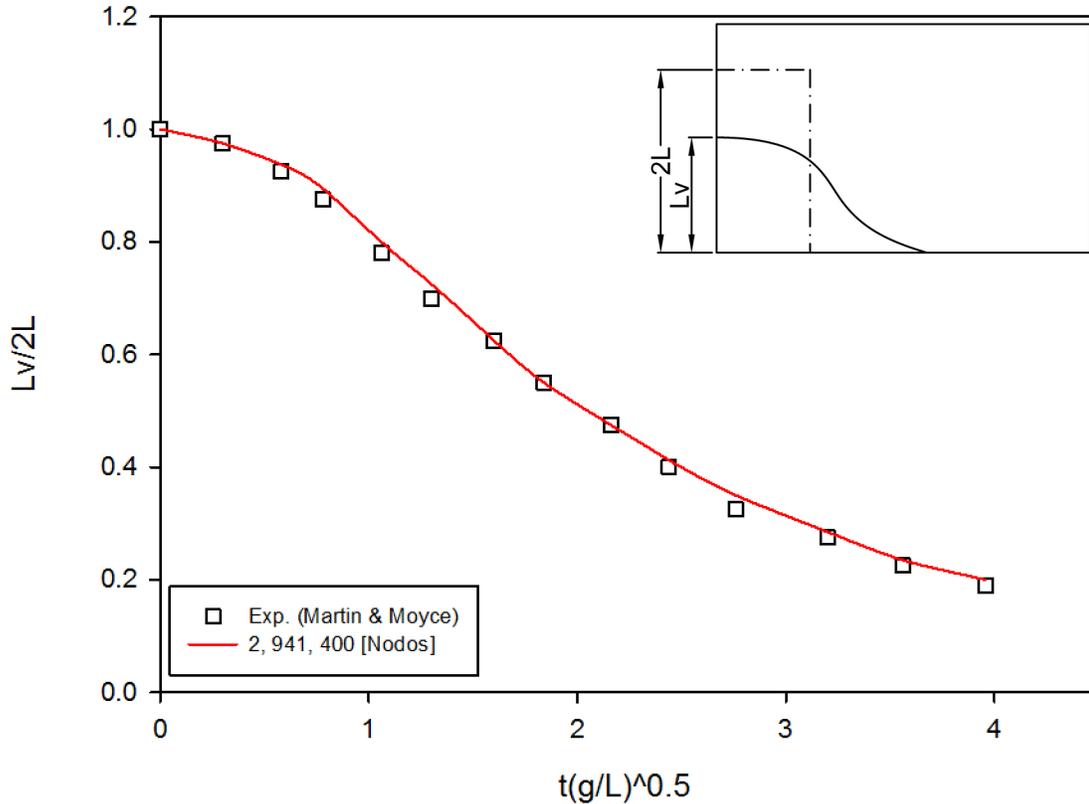


Figura 4.4: Variación de altura de la columna de agua en función del tiempo

Una medición más fue realizada para conocer el comportamiento de la columna de agua del lado izquierdo, esto se realizó registrando el valor de la altura de la columna en la pared en diferentes tiempos y se graficó en ejes adimensionales, los resultados se muestran a continuación en la Figura 4.4.

Para este caso solo se compararon los resultados obtenidos con la prueba 4, que demostraron ser los más exactos; con los obtenidos experimentalmente por Martin Moyce (1952), debido a que en el experimento de Koshizuka et al. (1995) la variación en la altura de la columna de agua no fue registrada.

Los resultados fueron muy satisfactorios para esta prueba, en donde la correspondencia de los resultados experimentales con los numéricos es mucho mejor que para la medición anterior. El hecho de que los resultados para este caso sean mejores que en la gráfica de la Figura 4.3 se debe a que las variaciones de los campos de velocidades y presiones en la pared izquierda del dominio son mínimas comparadas con las que ocurren en el fondo del tanque donde se tiene el frente que rompe de la columna de agua.

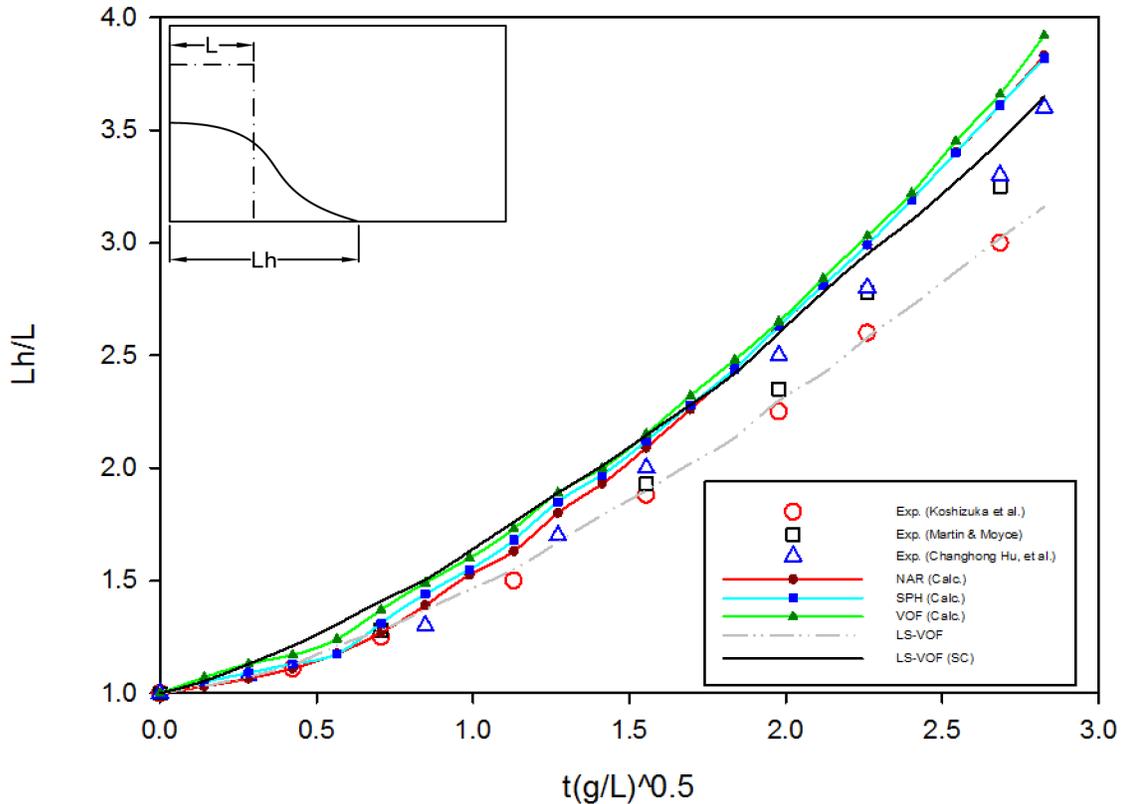


Figura 4.5: Resultados Experimentales vs Resultados Numéricos

Una vez establecida la importancia del mallado se tomó la Prueba 4 para compararla con los resultados experimentales de Koshizuka et al. (1995) [77], Martin Moyce [76]; así como con un trabajo numérico que usa la técnica NAR realizado por R.R. Nourgaliev, et al. [53], la técnica SPH de Songdong Shao, et al. [47] y el método VOF llevado a cabo por I. R. Park, et al. [32], los resultados están en la Figura 4.5.

Al analizar los resultados se nota que existe una discrepancia importante entre los resultados obtenidos por las vías numéricas con los resultados obtenidos por el LS-VOF que también es numérico. La causa puede deberse a que en los casos numéricos mostrados

en la gráfica de la Figura 4.5 no se consideró el efecto que tiene el retirar la compuerta en el caso experimental, pues a pesar de ser un movimiento súbito, la compuerta físicamente está ahí por un instante de tiempo, que aunque parece mínimo puede provocar errores.

Para comprobar lo anterior se realizó una simulación más con nuestro código numérico, con las características de la Prueba 4, pero con la diferencia de que para este caso el retiro de la compuerta no es simulado sino que simplemente desaparece, tal y como ocurrió en los otros casos numéricos.

Por lo anterior el resultado LS-VOF corresponde a los datos obtenidos con la prueba 4 y la gráfica denominada LS-VOF (SC) corresponde a los datos obtenidos con la prueba en donde no se considera el efecto de la compuerta.

Se puede observar que el comportamiento del código numérico correspondiente a este trabajo LS-VOF tiene un buen desempeño, incluso mejor que el obtenido con códigos numéricos similares, a tal grado de ser prácticamente igual al experimental realizado por Koshizuka et al. (1995), en el caso en que se considera el efecto de la compuerta y que es por lo tanto el más apegado a la realidad. Sin embargo, cuando dicho efecto no se toma en cuenta, los resultados conseguidos por el LS-VOF siguen siendo muy buenos comparados con los otros casos numéricos, pues se alcanza un alto grado de concordancia con los datos experimentales.

Por último, es importante mencionar que el hecho de considerar la compuerta o no hacerlo, tiene una alta influencia en el resultado final, este efecto es más evidente en las etapas iniciales de la simulación donde la gráfica no avanza como debería, sino que parece ir más lenta, resultando en una gráfica casi lineal.

4.3.1. Forma de la interfase bajo influencia de la malla

Una característica importante de este código es la manera en que se visualiza la interfase. Al refinar la malla la forma de la interfase tiende a ser más cercana a la realidad pues los errores inherentes a las técnicas numéricas son menores.

Con el fin de evaluar el efecto que tiene el mallado en la topología de la superficie libre, se compararon imágenes que corresponden al mismo tiempo de simulación pero con mallas diferentes, se seleccionaron la malla menos densa correspondiente a la Prueba 1 y la malla más refinada correspondiente a la Prueba 4, los resultados se muestran en la Figura 4.6.

En las imágenes se muestra que el comportamiento de la topología de la superficie libre cambia un poco con la malla y se hace evidente especialmente en la forma del frente de onda. Para la Prueba 1 que corresponde a la malla menos densa, la punta de la ola tiende a ser cuadrada, en cambio en la malla más fina correspondiente a la Prueba 4 el frente de

onda tiene una forma similar a la correspondiente al caso experimental.

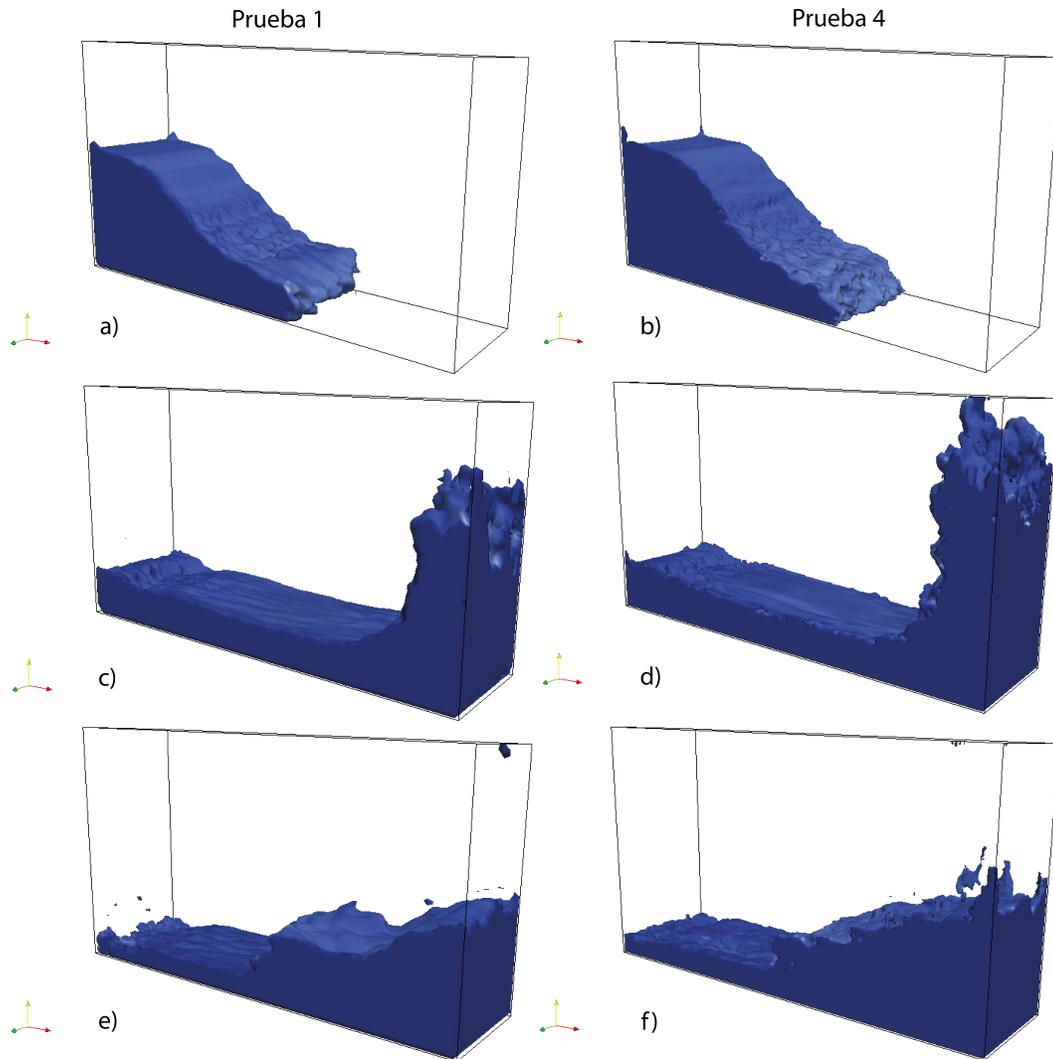


Figura 4.6: Evolución de la superficie libre para cada prueba

A medida que la malla se refina, la topología de la interfase se vuelve más exacta y también más nítida o definida; pero también se vuelve evidente que el frente de la onda tiende a ser caótico. Esto se debe a que al tener una malla más fina, también se capturan mejor los efectos de la turbulencia y por lo tanto la forma de la interfase no es lisa, sino que tiene una forma muy irregular como salpicaduras, tal y como pasa en el experimento.

En la figura anterior, la forma de la interfase cambia con la densidad de la malla y aunque los cambios parecen mínimos se presentan detalles como curvas más pronunciadas y

una ligera variación en el avance que se puede observar en los últimos cuadros comparativos donde la simulación correspondiente a la malla más densa avanzó más y tiene una forma más definida en el fondo, mientras que la imagen correspondiente a la Prueba 1 presenta formas irregulares e incluso parece no tocar el fondo del dominio.

4.3.2. Comparación con el caso experimental

De acuerdo con las fotografías obtenidas por Koshizuka et al. (1995) [77], se espera que:

- ◇ En el tiempo $t = 0,2s$: aproximadamente el 75 % de la base del tanque está cubierta por agua.
- ◇ En el tiempo $t = 0,3s$: el frente de la ola toca la pared derecha del tanque.
- ◇ En el tiempo $t = 0,4s$: Se forma un pequeño ángulo entre la horizontal de la interfase y la base del tanque, y en la esquina superior derecha del dominio un poco de agua sale del tanque.
- ◇ En el tiempo $t = 0,6s$: La horizontal de la interfase es prácticamente paralela a la base del tanque y el agua que impacto contra la pared derecha comienza a descender por efecto de la gravedad.
- ◇ En el tiempo $t = 0,8s$: la ola ha doblado hacia atrás y una porción de aire queda atrapada.
- ◇ En el tiempo $t = 1,0s$: la lengua de la ola que dobló toca la pared izquierda y queda atrapada una gran burbuja de aire.

Partiendo de lo anterior, se realizó una comparación de los perfiles obtenidos numéricamente usando el código perteneciente a esta tesis, con respecto al experimental llevado a cabo por Koshizuka et al. (1995) [77] en los tiempos mencionados anteriormente Figura 4.7.

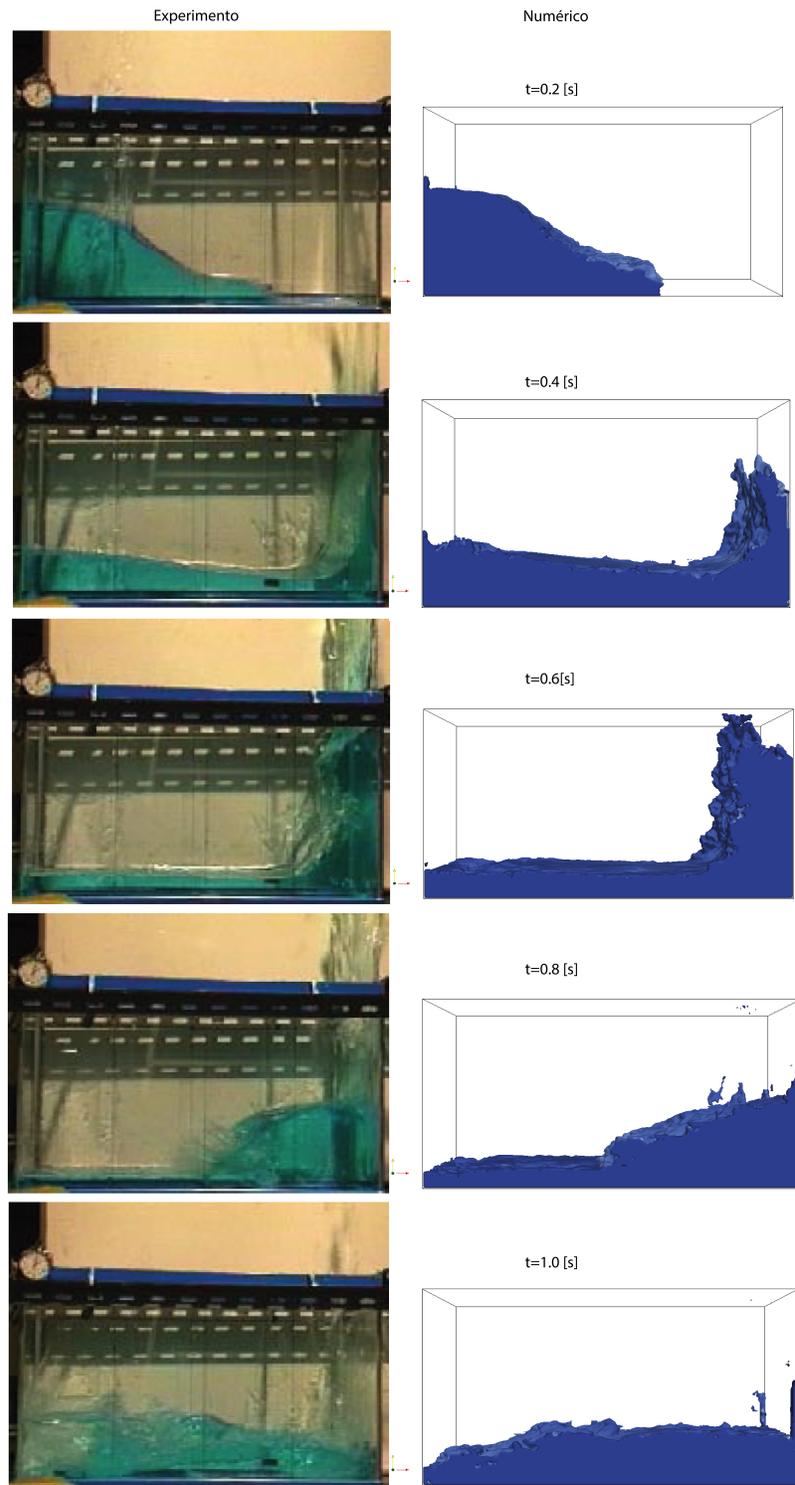


Figura 4.7: Comparación de resultados numéricos y experimentales

Se puede observar que el método se comporta como se esperaba, resaltando el pequeño ángulo formado por la cama de agua respecto a la horizontal para el tiempo $t = 4s$, así como que el flujo es prácticamente paralelo al fondo en el tiempo $t = 6s$. En el caso experimental el dominio está abierto a la atmósfera, por lo que el agua sale del contenedor en el tiempo $t = 6s$, debido a que en la simulación realizada para este trabajo el dominio está totalmente cerrado, el fluido no tiene posibilidad de salir, pero choca con la parte superior lo que nos permite pensar que tiene un comportamiento muy similar al real.

Finalmente, en la siguiente Figura 4.8. se muestra la tridimensionalidad de la simulación, mencionando que aunque la naturaleza del experimento permite tratarlo como un caso bidimensional; el código desarrollado para esta tesis trabaja con tres dimensiones, lo que permite una simulación más exacta, donde la superficie libre es simulada de una forma realista y detallada muy similar a un flujo real.

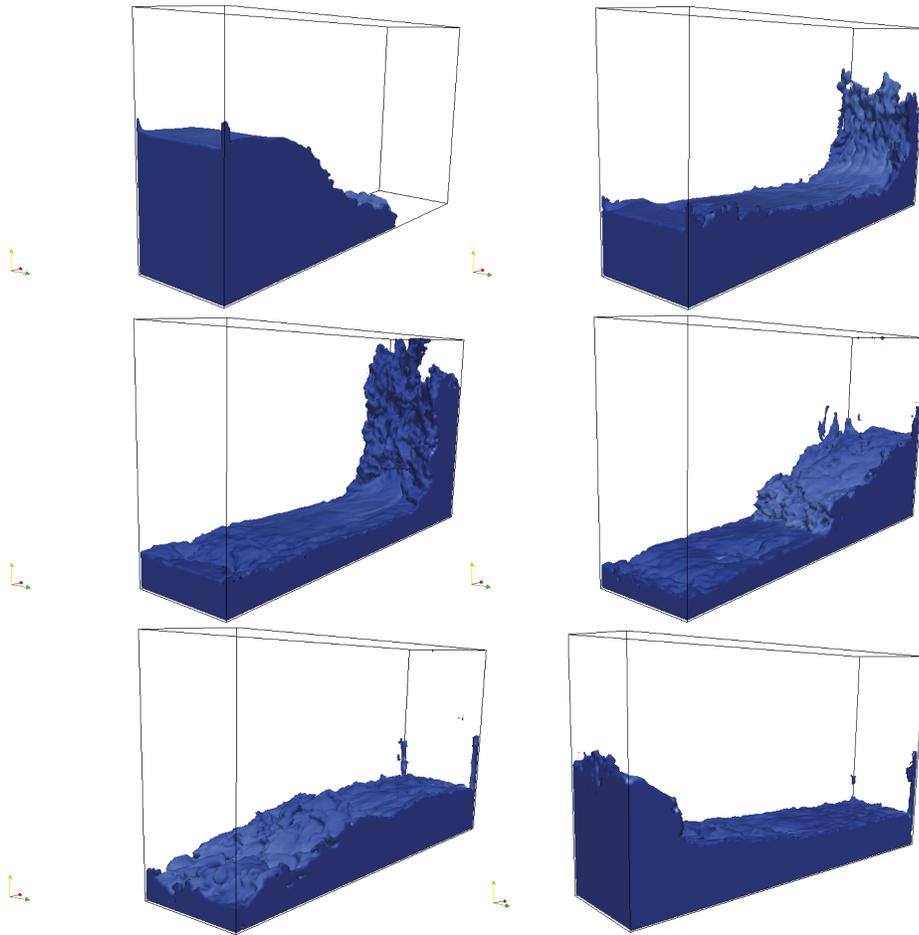


Figura 4.8: Desarrollo del dam-break en tres dimensiones

4.4. Rotura de presa sobre cama de agua

El segundo caso experimental considerado para validar el código numérico presentado en este trabajo es una variante del anterior que por su naturaleza es un poco más difícil de simular y por lo tanto es un reto importante para poner a prueba nuestro modelo.

El experimento fue realizado por Janosi et al. [78], y debido a sus interesantes características fue tomado como punto de partida para la validación de algunos códigos numéricos [48] [79] [80] [81] [82] [83].

El arreglo experimental Figura 4.9 es un tanque con paredes y fondo de vidrio dividido en dos partes, la primera de ellas va de la pared izquierda a la compuerta que mantiene la columna a una altura $11 \leq d_0 \leq 25\text{cm}$, la separación entre pared y compuerta es de $x = 38\text{cm}$. La segunda parte comprende de la compuerta a la pared derecha, separadas por una distancia de 955cm ; es en esta región donde se tiene una capa de agua cuya altura d , cambia en cada experimento.

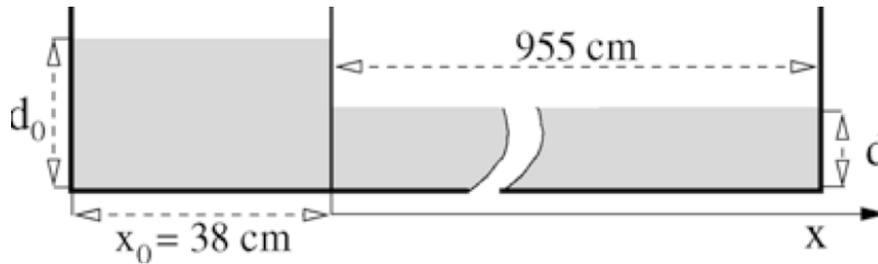


Figura 4.9: Arreglo experimental

En los experimentos se obtuvieron imágenes digitalizadas de los experimentos con los que se determinó la posición del frente de agua en función del tiempo, dichas imágenes son las que se van a ser comparadas con los resultados numéricos para llevar a cabo la validación del código.

En el trabajo de Janosi et al. [78], se realizaron una gran cantidad de experimentos con diferentes condiciones iniciales que consisten en variaciones de los valores de d_0 y d . Para este trabajo se tomó solo uno de dichos experimentos que a su vez fue elegido para validación de algunos otros códigos numéricos [48] [79] [80] [81] [82] [83].

Este experimento consta del rompimiento de una columna de agua de altura $d_0 = 15\text{cm}$, sobre una cama de agua con una altura $d = 18\text{mm}$. Es importante mencionar que en los casos numéricos [48], [79], la longitud de la segunda parte de tanque se redujo de 955cm a 66cm , pues es en esta zona donde se llevan a cabo los fenómenos más importantes para

la validación y al hacer esta reducción del dominio no se introducen errores en el código numérico y se obtiene un ahorro importante de recursos computacionales y por ende se reduce el tiempo de simulación. Esta consideración fue tomada para este trabajo por lo que el dominio computacional con el que se realizaron las simulaciones queda como en la Figura 4.10.

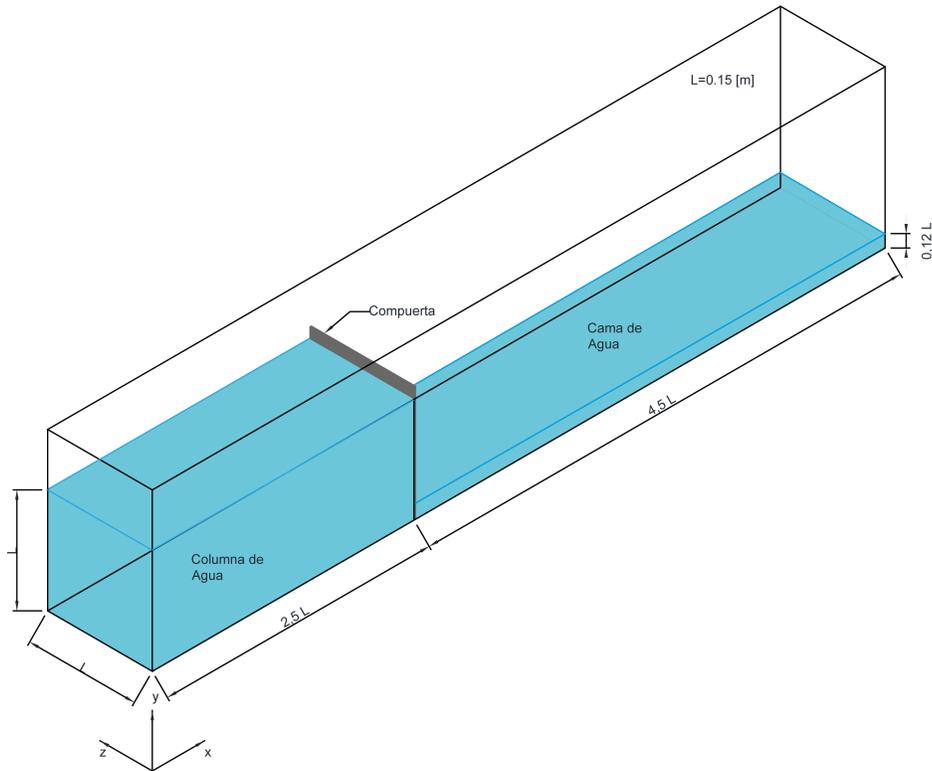


Figura 4.10: Dominio computacional

El desarrollo de una rotura de presa sobre una cama de agua es un proceso difícil de tratar que dista mucho de ser estacionario sobre todo en las etapas iniciales, donde el comportamiento del flujo es el resultado de una compleja interacción puesto que al momento de entrar en contacto el agua de la columna con el agua de la cama se genera una vorticidad que causa formas características en la interfase como frentes verticales que rompen hacia atrás y hacia adelante, así como el desarrollo de una pequeña ola que rebota y genera una segunda ola pequeña. Estas características especiales son las que permiten validar nuestro código numérico.

En el momento que la placa es removida hay un cambio instantáneo en el campo de presión debido a que el fluido es incompresible. Inicialmente la presión es discontinua entre la columna y la cama de agua a causa de la presencia de la compuerta, en el instante que se

quita la compuerta la presión se vuelve continua. Este repentino cambio causa un vórtice y una ola que se comporta de manera diferente en función de la altura de la cama de agua.

Para este experimento se tiene una altura $d = 18mm$, al retirar la compuerta se genera un vórtice debido a que el agua de la columna empuja al agua de la cama, lo que forma una ola que trata de propagarse pero que es frenada por la cama de agua, por lo que la onda en la superficie rompe hacia adelante y hacia atrás tomando una forma similar a un hongo. Al romper la ola atrapa burbujas de aire, lo que contribuye al desarrollo de turbulencias. La energía cinética se mantiene en el flujo hasta que se generan dos o tres roturas de ola más, después de lo cual el flujo se vuelve tranquilo. Ya que el efecto de las paredes laterales es despreciable, el flujo es prácticamente de dos dimensiones.

4.4.1. Resultados numéricos y experimentales

Para realizar las simulaciones numéricas para este experimento se tomó en cuenta la información acerca del código aportada por la simulación del rompimiento de columna sencillo que se presentó en este trabajo como el primer caso de validación.

Por lo tanto las simulaciones se realizaron usando una malla uniforme y fina de $1000 \times 214 \times 32$, en las direcciones x, y, z ; lo que permitió obtener resultados satisfactorios en poco tiempo, nótese que el mallado en z es mucho menor en relación al número de nodos en los otros ejes; esto no afecta de forma importante a la simulación pues por la naturaleza del experimento el fenómeno puede considerarse bidimensional, sin embargo el hecho de no mallar demasiado el eje z representa un ahorro importante de cálculos y por lo tanto de tiempo y recursos computacionales.

Las condiciones iniciales no fueron diferentes, puesto que se partió de un campo de velocidades nulo y un gradiente de presión tanto para la columna como para la cama de agua, cabe mencionar que se cuidó el hecho de que ambas partes del dominio están bien definidas y separadas, lo que significa que inicialmente se tienen dos campos de velocidad y presión diferentes, tal y como pasa en el caso experimental.

Se compararon los resultados numéricos del código correspondiente a este trabajo con los obtenidos numéricamente mediante el método SPH Crespo [48], así como con los resultados experimentales. La comparación de los perfiles obtenidos se muestra en la Figura 4.11.

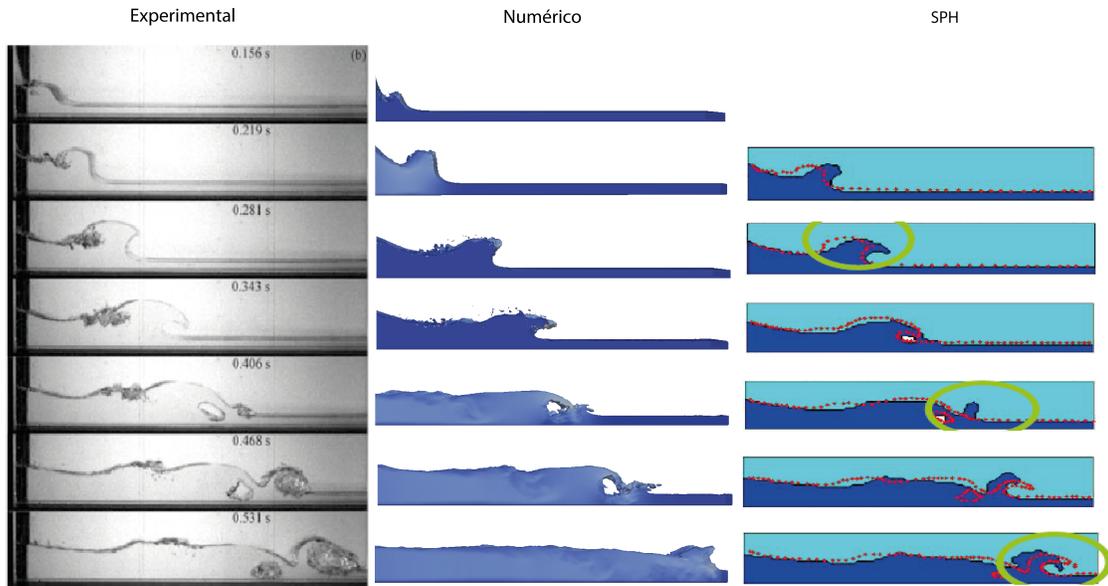


Figura 4.11: Comparación de resultados para $d = 18mm$

De acuerdo con las fotografías obtenidas por Janosi et al. [78], se espera que:

- ◇ En el tiempo $t = 0,156s$: los dos campos de fluido entran en contacto con lo que se forma un pequeño vórtice como resultado de la interacción de los dos flujos.
- ◇ En el tiempo $t = 0,219s$: se genera una ola cuyo movimiento tiende hacia atrás, producto del empuje de la columna de agua sobre la cama de agua.
- ◇ En el tiempo $t = 0,281s$: la ola empieza a tomar fuerza y la forma de lengua característica de las olas.
- ◇ En el tiempo $t = 0,343s$: la ola rompe hacia delante y hacia atrás teniendo una forma similar a un hongo.
- ◇ En el tiempo $t = 0,406s$: la ola rompió totalmente y toca la cama de agua por primera vez.
- ◇ En el tiempo $t = 0,468s$: la energía de la ola se mantiene y se genera una segunda ola mucho más pequeña que la primera.
- ◇ En el tiempo $t = 0,531s$: la segunda ola rompe y toca la cama de agua por segunda vez.

Al comparar los perfiles de la superficie libre formada es fácil notar que se tiene un comportamiento similar en el desarrollo del fenómeno, la topología de la interfase obtenida por el código numérico correspondiente a este trabajo es igual a la capturada experimentalmente; incluso se tiene un mayor grado de concordancia con el método LS-VOF, que con los perfiles obtenidos por el método numérico SPH donde no se consideró en la simulación el efecto del retiro de la compuerta. Lo anterior parece ser un detalle menor, pero la diferencia es muy grande pues al retirar la compuerta el vórtice generado es mucho mayor que el que se produce por la desaparición instantánea de la misma. Este fenómeno es el causante de la discrepancia en los resultados, este hecho fue verificado en el trabajo correspondiente al método SPH [48] con el que se comparó este trabajo.

Aunque las imágenes demuestran que el resultado obtenido con el método LS-VOF perteneciente a este trabajo tiene un buen desempeño y por lo tanto un alto grado de concordancia con los resultados experimentales; no se puede tener la certeza de que tan aproximados son los resultados por un simple análisis visual, por lo que se realizaron unas graficas de los perfiles numéricos y experimentales para algunos tiempos con el fin de poder tener una visión más clara del comportamiento del código aquí validado. Los resultados se muestran en la Figura 4.12.

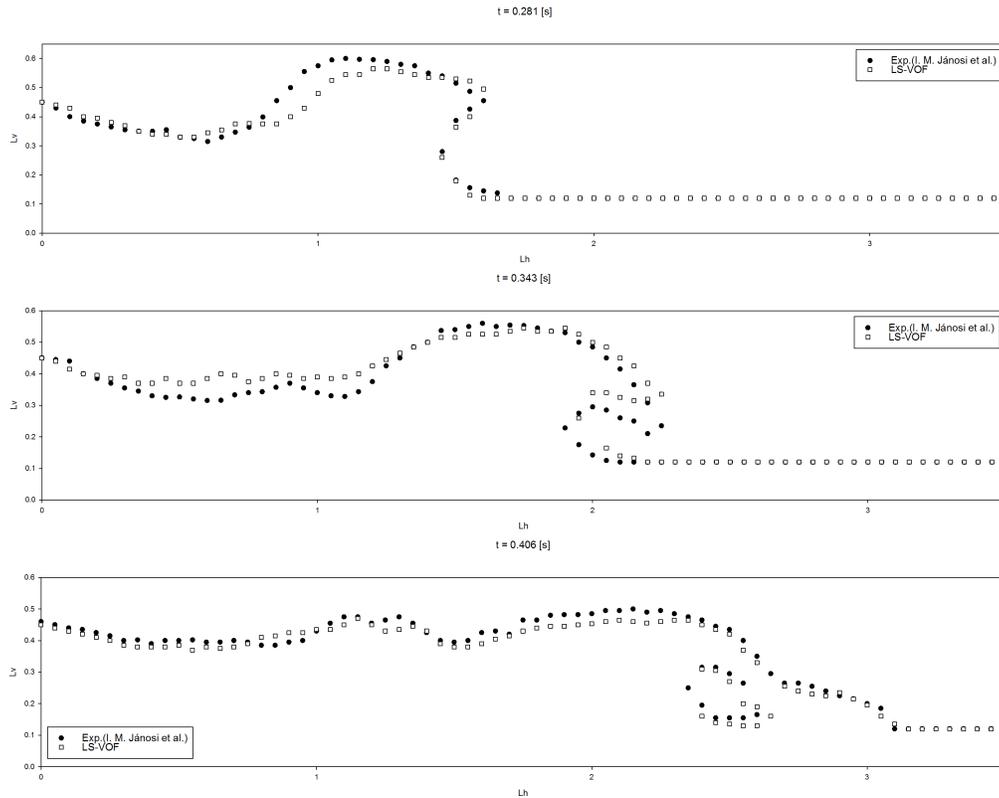


Figura 4.12: Comportamiento de los campos de velocidad para $d = 18mm$.

Los perfiles obtenidos demuestran de una manera más contundente el grado de aproximación de la simulación con el experimento para diferentes tiempos, lo que nos permite asegurar que el comportamiento del código es bueno y los resultados confiables.

La simulación realizada con el código LS-VOF es en todo momento tridimensional, por lo que los resultados son más exactos que los que podrían obtenerse con un código bidimensional. Además la forma en que se captura la superficie libre es visualmente más atractiva debido a las ventajas que ofrece la tridimensionalidad como se muestra en la Figura 4.13

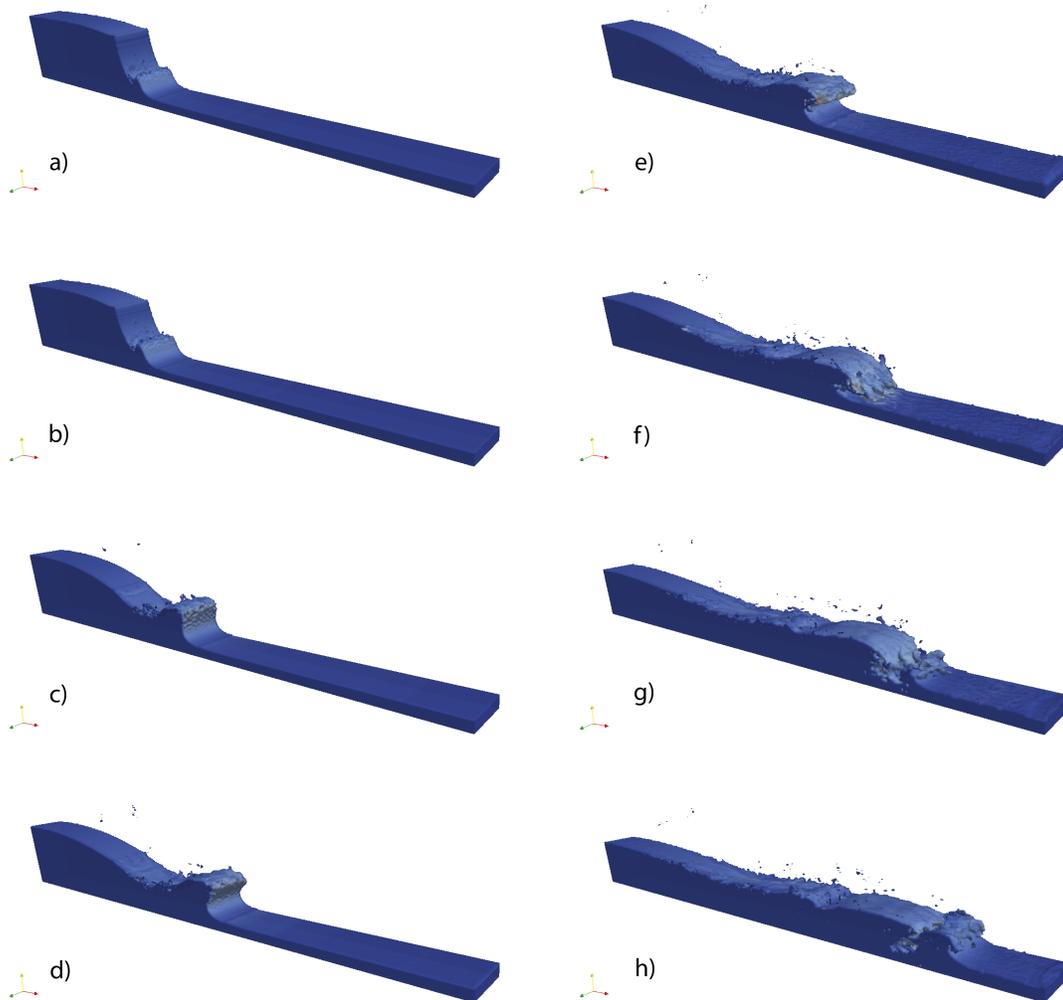


Figura 4.13: Rompimiento de columna sobre cama de agua para diferentes tiempos

La figura anterior es un historial de la simulación en tres dimensiones, donde puede observarse el comportamiento la onda en diferentes instantes de tiempo. Resulta interesante ver como aparecen rugosidades en la superficie libre conforme el tiempo avanza, lo que se debe a que el flujo adquiere más energía, por lo tanto más velocidad y más turbulencia, lo que lleva a que en las ultimas etapas (incisos g y h) se forman rugosidades en la cama de agua, antes de que la onda llegue a ella, hecho que se presenta en las playas cuando una ola se acerca a la zona de lavado.

4.5. Dam-break contra un obstáculo

Los experimentos anteriores son muy utilizados como punto de partida para la validación de códigos numéricos y aunque en este trabajo se ha simulado en 3D el comportamiento del flujo, en ambos casos puede considerarse como 2D pues las variaciones en el eje Z son mínimas comparadas con las que ocurren en los otros dos ejes.

Es por eso que el último experimento se seleccionó para poner a prueba el código LS-VOF con un caso donde el comportamiento del flujo sea muy diferente en los 3 ejes.

Para llevar a cabo esto se optó por simular el experimento llevado a cabo por Kleefsman et al. (2005) [84], se trata del rompimiento de una columna de agua que impacta contra un bloque puesto en el camino del flujo. Esta interacción incrementa significativamente la dificultad de simulación, pues el flujo se vuelve altamente turbulento y por las características del impacto, tridimensional.

El caso experimental se llevó a cabo en un taque de $3,2 \times 1,0 \times 1,0$ [m], donde se contiene una columna de agua de $0,55$ [m] de altura de la pared izquierda a una compuerta, que al ser retirada genera por la acción de la gravedad, la caída de la columna de agua sobre la parte vacía del tanque, donde se encuentra el bloque solido que obstaculiza el libre camino del flujo. Se registraron datos de altura y presión en diferentes partes del tanque, un esquema del dominio experimental se muestra en la Figura 4.14

Para esta prueba el mallado para la simulación fue uniforme y no tan refinada como para el caso anterior, quedando de $420 \times 179 \times 60$ en las direcciones x, y, z ; para un total de $4,510,800$ [nodos].

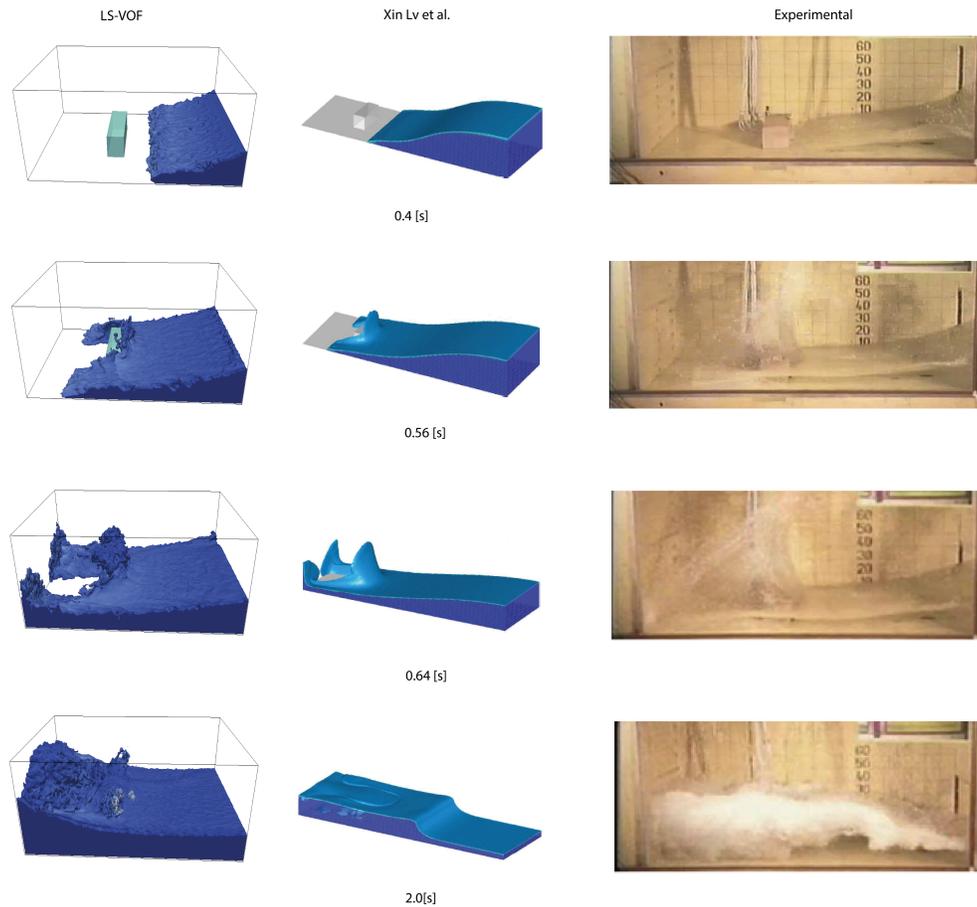


Figura 4.15: Simulación numérica vs experimental

De acuerdo con las fotografías obtenidas por Kleefsman et al. (2005) [84], se espera:

- ◇ En el tiempo $t = 0,4s$: El agua está a punto de impactar contra el bloque.
- ◇ En el tiempo $t = 0,56s$: El agua choca contra el bloque por lo que el frente de la ola se rompe, la parte que colisiona se eleva, mientras que el resto del agua avanza por los lados del bloque.
- ◇ En el tiempo $t = 0,64s$: El agua que paso por los lados del bloque choca contra la pared izquierda del contenedor; detrás del bloque se forma un espacio sin agua, mientras que el agua que impacto contra el bloque empieza a descender.
- ◇ En el tiempo $t = 2,0s$: El agua vuelve hacia el lado derecho del contenedor y el bloque queda totalmente sumergido por el agua.

La Figura 4.15 muestra imágenes del dambreak con un obstáculo, correspondientes al caso experimental de (Kleefsman, 2005) [84], una simulación numérica Xin Lv [57] y el código propuesto en esta tesis. Las imágenes corresponden a los mismos instantes de tiempo, cuando se da el impacto, y cuando el frente de onda regresa generando una ola.

Puede observarse que se tiene un buen grado de semejanza entre las imágenes del experimento y las numéricas, pero destacan principalmente dos aspectos. En la primer imagen de la Figura 4.15, correspondiente al instante en que el agua toca el bloque; sí se aprecia un salto en los casos numéricos pero en ninguno de los dos se alcanza la altura del caso experimental, al observar más detenidamente, se aprecia que además de elevarse, el frente de onda rompe en pequeñas gotas que se dispersan en muchas direcciones; cosa que no pasa en los casos numéricos. Lo anterior pareciera ser un error numérico pero en el caso del método LS-VOF no lo es; el hecho de que el agua no alcance la altura y el frente de onda no rompa se debe al mismo motivo; el mallado. Como el flujo al momento del impacto se vuelve extremadamente complejo se necesita una densidad de mallado muy alta para capturar cada aspecto del fenómeno como son las pequeñas gotas y la altura que alcanza el agua; por su naturaleza el código trata de mantener definida la forma del frente de onda en todo momento, mientras más fina es la malla, mayor es la libertad del fluido para deformarse libremente; en este caso la malla permite un nivel de deformación que no es lo suficientemente bueno para simular el impacto y en consecuencia la altura en el choque y las salpicaduras no corresponden con el caso experimental. La solución obvia es mallar más finamente, pero no es tan sencilla de ejecutar pues como ya se mencionó el elevar la densidad de mallado incrementa considerablemente el tiempo de simulación y para este caso no vale la pena por razones que se explicaran más adelante.

Por otro lado, en la última imagen los resultados son mejores. Aunque en la imagen correspondiente al experimento se observa un flujo altamente turbulento en la superficie; globalmente ya no es tan complicado como en el momento del choque y por lo tanto la simulación es más sencilla. Para este instante de tiempo los casos numéricos se asemejan al experimental en la posición de la onda que se mueve hacia la pared derecha, pero los resultados obtenidos con el método LS-VOF son más satisfactorios pues la turbulencia en la superficie se aprecia mucho mejor que la simulación correspondiente a Xin Lv [57] donde la superficie libre es prácticamente lisa en toda la simulación.

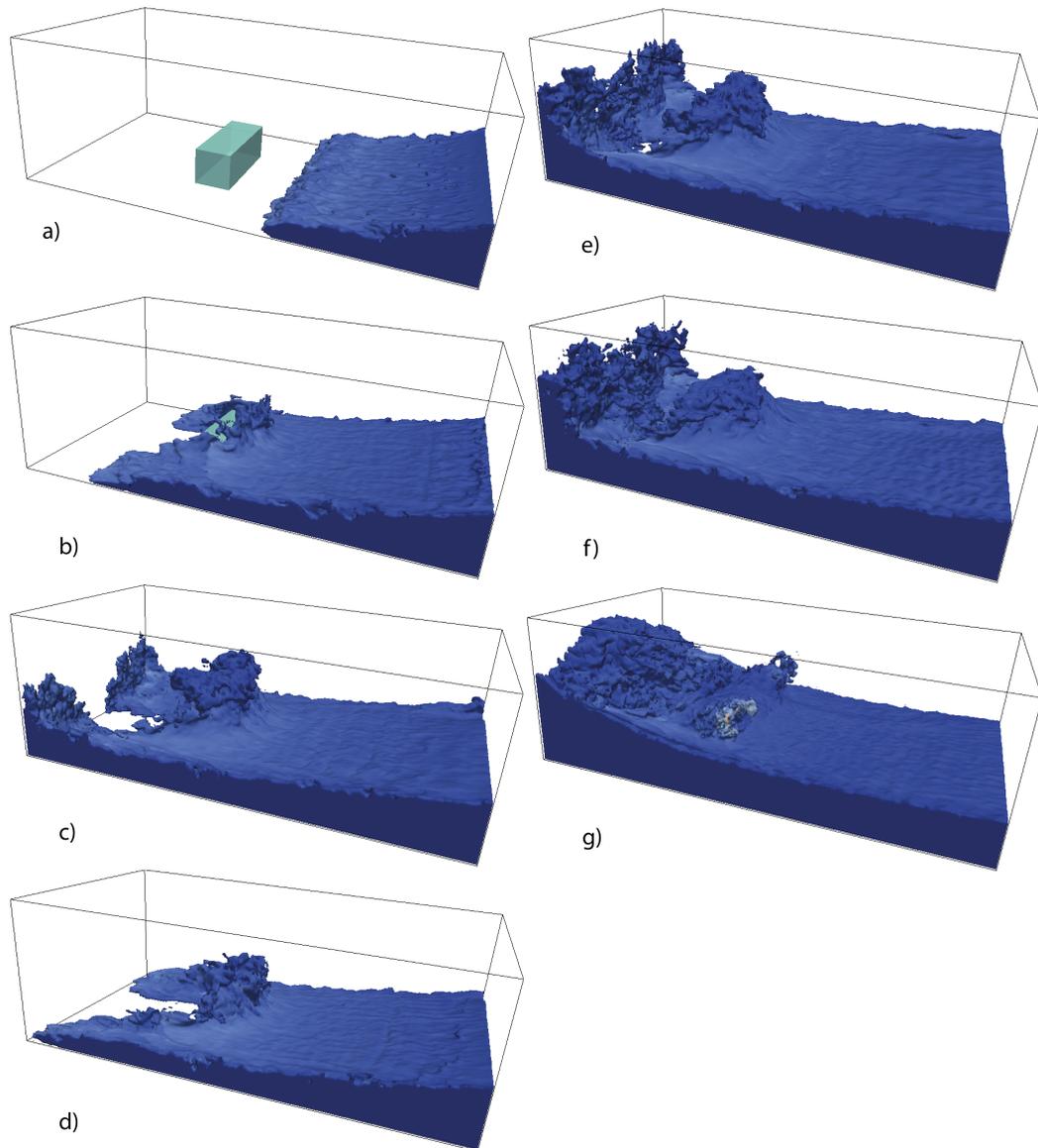


Figura 4.16: Historial de la simulación

Aunque se tienen resultados muy similares, la superficie libre tiene un aspecto más real para cada instante de tiempo comparada con la simulación numérica de Xin Lv [57]. La visualización de la superficie libre con ese grado de realismo es el resultado del trabajo en sinergia de LES, GFM, LS y VOF, además de la discretización de alto orden que se utilizó para el código numérico aquí validado; la Figura 4.16 muestra un historial del dam-break con un obstáculo donde puede observarse el comportamiento de la interfase y como se vuelve turbulenta para cada instante de tiempo.

En el caso experimental se usaron 4 sondas para medir la profundidad del flujo en diferentes zonas del dominio experimental, estas mediciones se reportaron en el trabajo de Kleefsman et al. [84], y con dos de ellas se generaron graficas que describen el comportamiento de la superficie libre. Para poder comparar mejor el desempeño de la simulación correspondiente a este trabajo, se midió el comportamiento de la superficie libre en las mismas zonas que el caso experimental, los resultados se muestran en las Figuras 4.17 y 4.18.

Con el apoyo de las gráficas se puede demostrar que la simulación obtenida con LS-VOF da buenos resultados, la curva obtenida numéricamente se ajusta a la experimental en la mayor parte del tiempo, mientras que en las partes complicadas donde los resultados no son exactos; la tendencia se mantiene de una forma muy satisfactoria, es por ello que anteriormente se mencionó que no valdría la pena aumentar el mallado pues los resultados son muy satisfactorios antes y después de la zona crítica en el flujo, por lo que se puede asegurar que el hecho de que el frente de onda no alcanzó la misma altura que el caso experimental no es un error numérico al no alterar los resultados.

La primer grafica (Figura 4.17) se obtuvo con la sonda ubicada en el depósito donde inicialmente se tenía la columna de agua, por lo cual la variación de altura tiene en las etapas iniciales el comportamiento de un dambreak sencillo, alcanzando una altura mínima adimensional de 0,2 después de la cual se tienen dos picos importantes, el primero aproximadamente en el tiempo adimensional 12, debido al retorno del flujo después de impactar contra la pared izquierda. Después de esta primer elevación el frente choca contra la pared derecha y vuelve hacia al lado izquierdo aumentando su profundidad una vez más alcanzando su máximo aproximadamente en el tiempo adimensional 16. Finalmente, el flujo toca la pared izquierda una vez más y vuelve, por lo que se tiene otro pico de menor tamaño en el tiempo 20. Los resultados obtenidos con el código LS-VOF muestran una mayor exactitud que los obtenidos por Xin Lv [57], principalmente en los dos primeros picos.

La segunda gráfica (Figura 4.18) corresponde a la sonda ubicada antes del bloque, debido a que el agua esta inicialmente contenida por la compuerta y en esta zona no se tiene agua, la gráfica empieza en una longitud adimensional inicial de 0, hasta aproximadamente el tiempo adimensional 2, que es lo que tarda el agua en llegar a esta zona, el nivel de agua empieza a aumentar hasta alcanzar un máximo aproximado de 0,7 de altura en el tiempo 7, ambos en adimensionales. El frente de onda choca contra la pared izquierda y regresa al lado derecho, por lo que se registra un mínimo en el instante en que la mayor cantidad de agua se encuentra viajando hacia la pared derecha aproximadamente en el tiempo adimensional 17, que es muy similar al tiempo de altura máxima en la gráfica anterior. Cuando la onda choca con la pared derecha y vuelve se tiene otro pico aproximadamente para el tiempo adimensional 20. Para esta gráfica la aproximación numérica con el caso experimental es un poco mejor que en la anterior teniendo una semejanza casi total.

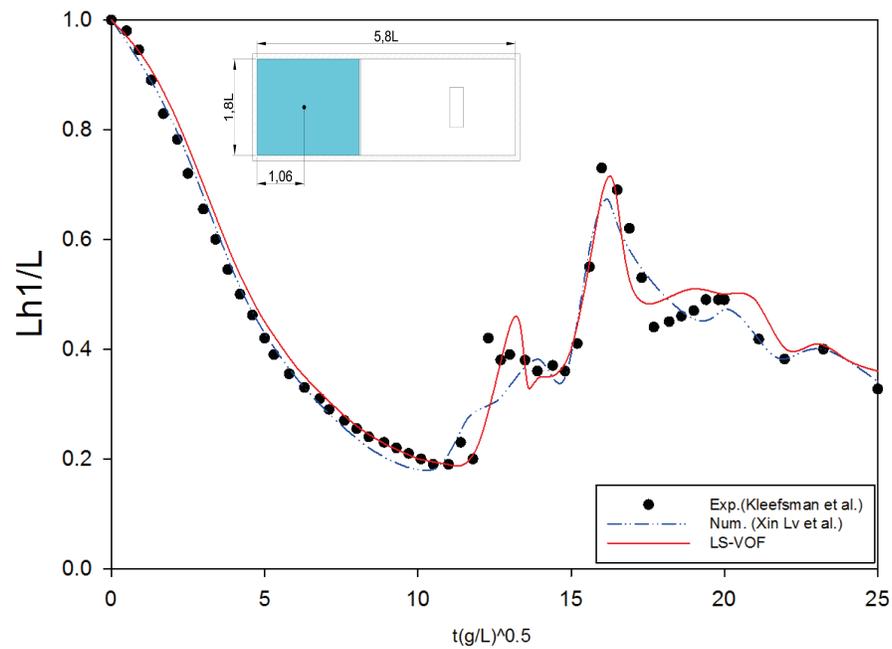


Figura 4.17: Variación de altura en la columna de agua

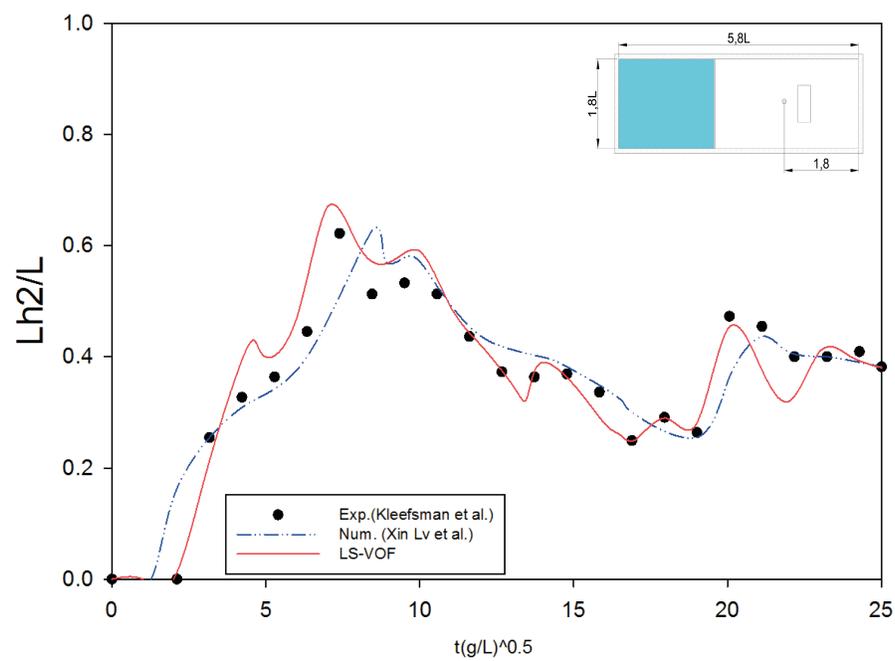


Figura 4.18: Variación de altura antes del bloque

Las gráficas anteriores representan un historial del comportamiento del flujo durante aproximadamente 6[s], el resultado de la simulación con LS-VOF es muy bueno, incluso mejor que el obtenido con la técnica numérica de Xin Lv [57]. La forma de la superficie libre es significativamente mejor, capturando los efectos de turbulencia. La mayor discrepancia se presenta al momento del choque contra el bloque debido a que el flujo se vuelve muy complejo y aunque visualmente el frente de onda en el momento del impacto no es tan similar al experimental, el resultado final no se ve afectado y sigue siendo una buena simulación. Existe la opción de refinar más el mallado para mejorar el efecto visual, pero el tiempo computacional sería mucho mayor y la ganancia sería mínima.

En el trabajo experimental se obtuvo también un historial del comportamiento de la presión en el bloque, para ello se utilizaron 8 sensores ubicados en dos caras del obstáculo; cuatro en el frente y cuatro en la superficie superior Figura 4.19

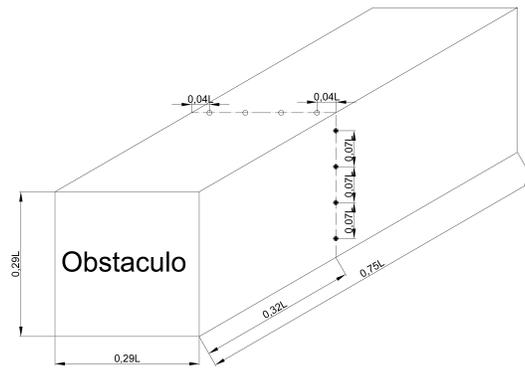


Figura 4.19: Ubicación de los sensores de presión en el obstáculo

Con los datos obtenidos por estos dispositivos se pudo conocer el comportamiento de la presión en distintos puntos del bloque, esos datos se compararon con los obtenidos por la simulación LS-VOF, los resultados se muestran en las gráficas (Figura 4.20)

Las gráficas corresponden a los puntos de presión $P1$ a $P8$, donde se puede ver que el impacto del agua con el bloque es muy bien simulado por el código LS-VOF, pues las gráficas generadas con los datos de la simulación tienen una tendencia muy similar a las obtenidas experimentalmente. La magnitud de la presión en el momento del impacto en el sensor de presión $P1$ es prácticamente la misma que la experimental, al igual que los picos obtenidos en los sensores $P2$, $P3$ y $P4$, donde las magnitudes máximas son muy cercanas a las experimentales. Después del impacto la curva de la simulación es un poco diferente de la experimental, presentando ligeras oscilaciones que pueden deberse a que el impacto violento del flujo contra el bloque genera fluctuaciones que se vuelven más pronunciadas en la simulación como resultado de la discretización o limitaciones inherentes al tamaño de la malla; pero la tendencia de la curva es la misma.

En cuanto a los sensores ubicados en la parte superior del obstáculo $P5$, $P6$, $P7$ y $P8$; el historial de la presión correspondiente a la simulación numérica es afín al obtenido experimentalmente, una vez más se presentan oscilaciones en las curvas, en este caso un poco más evidentes que en los sensores ubicados en el frente del bloque. Este historial tiene mayor dificultad para ser simulado debido a que los fenómenos que se presentan en esa región son muy complejos ya que la turbulencia es elevada como consecuencia de la interacción de agua y aire en el momento del impacto.

Por lo tanto, las gráficas de la simulación de los sensores en la parte superior del bloque son un poco menos apegadas a las experimentales comparadas con las correspondientes a los sensores del frente del bloque, nuevamente los errores asociados a la discretización y al mallado trabajan en sinergia con las complicaciones propias del experimento y en consecuencia la simulación se complica. Sin embargo, la tendencia general tanto en el caso experimental como en la simulación, tienen una buena correspondencia durante toda la simulación.

Las gráficas de presión son un buen indicador del buen comportamiento del código numérico para simulaciones de flujos a superficie libre en 3 dimensiones, pues aunque la complejidad de simular estos sistemas es elevada, los resultados son buenos y las características principales del flujo así como los puntos importantes como los máximos y mínimos de presión se capturan con un buen grado de exactitud, la ubicación de la superficie libre es también un punto fuerte de este código que puede observarse en las gráficas correspondiente a la altura del flujo donde el comportamiento concuerda con el caso experimental.

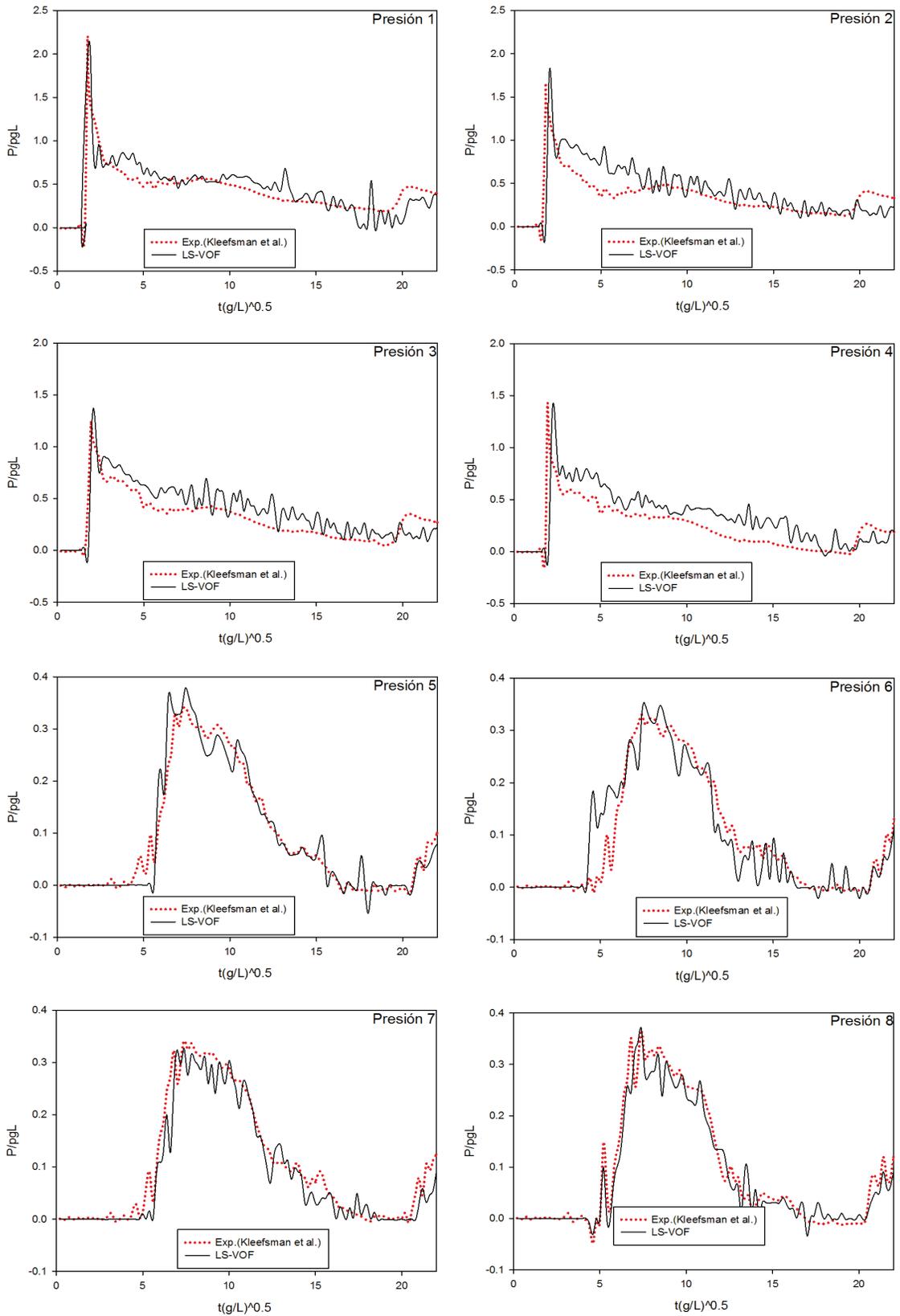


Figura 4.20: Variación de la presión en cada sensor

Capítulo 5

Conclusiones y trabajos futuros

Se realizaron simulaciones de flujos bifásicos con un código numérico que utiliza distintas técnicas para mejorar su desempeño y los resultados fueron satisfactorios.

Las mejoras son evidentes sobre todo en las gráficas de avance de frente en el primer caso de validación, puesto que los resultados del método correspondiente a esta tesis tienen una mayor concordancia con los datos experimentales que los resultados proporcionados por otros métodos numéricos.

La técnica LES en conjunto con un buen mallado da como resultado una simulación de alta calidad, hecho que puede observarse en la correspondencia casi exacta de los resultados numéricos con los experimentales en las gráficas.

Por otro lado la dependencia de la malla se observa en la topología de la superficie libre, pues esta cambia de manera drástica con la densidad y suavidad de la malla. El código cumple con la condición de anisotropía, por lo debe mallarse más densamente las zonas que presentar flujos más complejos.

El método permite que no exista un apreciable cambio en la masa del flujo, conservándose de buena manera a lo largo de toda la simulación. La pérdida de masa debida al método Level Set es evidente en las simulaciones como pequeños saltos o burbujas en la simulación en zonas que no deberían de existir, esto no ocurre en las simulaciones realizadas para esta tesis.

Además del buen manejo de la interfase, el código numérico usado en este proyecto fue diseñado para simular flujos complejos en estado estable y transitorio por lo que ofrece buenos resultados en datos como campos de presión y velocidad, asimismo es capaz de simular flujos que presentan turbulencias. Por lo que puede asegurarse que es un modelo

confiable.

Las dificultades que pueden presentarse al trabajar con este modelo están más relacionadas con la forma de plantear el problema que con el código en sí mismo, como ocurrió en el primer caso donde no se consideró en un principio el efecto de retirar la compuerta; por lo que es importante analizar cuidadosamente las condiciones iniciales y de frontera. Por otro lado, la interacción del fluido con objetos es un punto a mejorar, puesto que en el caso del dambreak con un obstáculo, se observó una baja de desempeño justo en el punto donde se da el contacto bloque-flujo.

El código permite simular de manera satisfactoria flujos bifásicos en tres dimensiones y no se tienen complicaciones siempre y cuando se tenga una malla suave, los cambios drásticos en la distribución de la malla generan notables cambios en los resultados. Por lo tanto se recomienda que en trabajos futuros se tenga esto en mente para evitar problemas asociados a la distribución de la malla, así como a su densidad.

Se demostró que este código tiene un buen comportamiento para flujos bifásicos e incompresibles sencillos y se espera que en trabajos posteriores pueda ser probado en condiciones que aumenten el grado de dificultad como simulaciones con más de dos fluidos o con cambios de fase, así como probarlo a condiciones de presión y velocidad más complejas así como con problemas que involucren transferencia de calor.

Por último es importante señalar que aun cuando parece que los resultados no son exactos en cuanto a la topología de la superficie libre, no significa que la simulación este mal y los resultados deban despreciarse. Primeramente este hecho nos obliga a analizar meticulosamente los resultados obtenidos por cualquier modelo computacional, analizar qué parte del experimento no se está tomando en cuenta y encontrar una explicación a los resultados “erróneos”, lo que nos permite comprender mejor el fenómeno en cuestión y mejorar el modelo en cuanto sea posible.

Además, aun en el caso en que algunos datos no correspondan, muchos otros pueden rescatarse, por ejemplo en el caso del dambreak sobre la cama de agua, aunque la superficie libre no se comportó exactamente igual a la original, el movimiento de la onda era prácticamente el mismo en el tiempo; esto nos aporta información valiosa del flujo, como que la fricción del fondo es el factor principal en la velocidad de la onda y la causante del rompimiento, lo cual fue corroborado por Janoshi [78]. Otro ejemplo de se encuentra en el caso del dambreak con un obstáculo donde al momento de que el flujo impactó con el obstáculo la ola de la simulación no rompió como lo hizo en el experimento, lo que introdujo un error a la simulación que se demostró que no fue tan grave, ya que la tendencia en las gráficas fue similar. Por todo lo anterior, se puede concluir que el código CFD, LS-VOF es confiable y queda validado.

Bibliografía

- [1] Tu J, Yeoh G, Liu C.(2008), “Computational Fluid Dynamics: A Practical Approach” El sevier, USA.
- [2] H. K. Versteeg and W. Malalasekera. (1995). “An introduction to computational fluid dynamics: The Finite Volume Method” Longman Group Ltd, Malaysia.
- [3] J.H. Ferziger and M. Peric. (2002). “Computational Methods for Fluid Dynamics” Springer, USA.
- [4] Fernández Oro, Jesús Manuel. (2012). “Tecnicas numericas en ingenieria de fluidos : introduccion a la dinamica de fluidos computacional (CFD) por el metodo de volu-
menes finitos” Reverte, Barcelona.
- [5] Ballesteros Tajadura, Rafael. (2003). “Tecnicas Numericas en Mecanica de Fluidos”
Universidad de Oviedo, Gijon.
- [6] Guan Heng Yeoh and Jiyuan Tu. (2010). “Computational Techniques for Multi-Phase
Flows” El sevier, USA.
- [7] Nikolay Ivanov Kolev, PhD., DrSc. (2002), “Multiphase Flow Dynamics” Springer,
Berlin.
- [8] Zhangxin Chen, Guanren Huan, Yuanle Ma.(2006), “Computational Methods for Mul-
tiphase Flows in Porous Media”. SIAM, USA.
- [9] Jean-Paul Caltagirone, Stephane Vincent, Celine Caruyer. A multiphase compressible
model for the simulation of multiphase flows” Computers Fluids 50 (2011) 24–34
- [10] James M. Osborne, Jonathan P. Whiteley. “A numerical method for the multiphase
viscous flow equations” Comput. Methods Appl. Mech. Engrg. 199 (2010) 3402–3417
- [11] Phung Dang Hieu, Tanimoto Katsutoshi, Vu Thanh Ca. “Numerical simulation of
breaking waves using a two-phase flow model” Applied Mathematical Modelling 28
(2004) 983–1005

- [12] B. Wu, L. Briens, J.-X. Zhu. “Multi-scale flow behavior in gas–solids two-phase flow systems” *Chemical Engineering Journal* 117 (2006) 187–195
- [13] L.X. Zhou. “Advances in studies on two-phase turbulence in dispersed multiphase flows” *International Journal of Multiphase Flow* 36 (2010) 100–108
- [14] P. K. Stansby, A. Chegini and T. C. D. Barnes (1998). “The initial stages of dambreak flow” *Journal of Fluid Mechanics*, 374, pp 407424
- [15] T. O’Donoghue, D. Pokrajac, L.J. Hondebrink “Laboratory and numerical study of dambreak-generated swash on impermeable slopes” *Coastal Engineering* 57 (2010) 513–530
- [16] Onno Ubbink, “Numerical prediction of two fluid systems with sharp interfaces”. Tesis, Department of Mechanical Engineering Imperial College of Science, Technology Medicine, January 1997
- [17] Sylvain L. Pigny. “Academic validation of multi-phase flow codes” *Nuclear Engineering and Design* 240 (2010) 3819–3829
- [18] A. Larese, R. Rossi, E. Oñate and S.R. Idelsohn, “Validation of the particle finite element method (PFEM) for simulation of free surface flows”, International Center for Numerical Methods in Engineering (CIMNE), Universidad Politécnic de Cataluña, Barcelona, Spain, 2008.
- [19] P. Cherukat, Y. Na, and T.J. Hanratty, J.B. McLaughlin, “Direct Numerical Simulation of a fully developed turbulent flow over a wavy wall”, *Theoretical and Computational Fluid Dynamics*, (1998) 11: 109–134
- [20] N.A. Adams, “Direct Numerical Simulation of Turbulent Compression Ramp Flow”, *Theoretical and Computational Fluid Dynamics*, (1998) 12: 109–129
- [21] Kun Luo, Jianren Fan, and Kefa Cen, “Flow Visualization of the Turbulent Jet by Direct Numerical Simulation”, *Journal of Visualization*, Vol. 7, No. 2 (2004) 110
- [22] Xin Wang, Kevin F. McNamara, “Evaluation of CFD Simulation using RANS Turbulence Models for Building Effects on Pollutant Dispersion”, *Environmental Fluid Mechanics* (2006) 6: 181–202
- [23] Tanmay Sarkar, Dracos Vassalos, “A RANS-based technique for simulation of the flow near a rolling cylinder at the free surface”, *Journal of Marine Science Technology* (2000) 5:66–77
- [24] Yulin Wu, Shuhong Liu, Hua-Shu Dou, Liang Zhang, “Simulations of unsteady cavitating turbulent flow in a Francis turbine using the RANS method and the improved mixture model of two-phase flows”, *Engineering with Computers* (2011) 27:235–250
- [25] Erik Damgaard Christensen, Rolf Deigaard. “Large eddy simulation of breaking waves” *Coastal Engineering* 42 (2001). 53–86

- [26] Dimokratidis G.E. Grigoriadis, Athanassios A. Dimas, Elias Balaras. “Large-eddy simulation of wave turbulent boundary layer over rippled bed” *Coastal Engineering* 60 (2012) 174–189
- [27] Pierre Lubin, Stephane Vincent, Stephane Abadie, Jean-Paul Caltagirone “Three-dimensional Large Eddy Simulation of air entrainment under plunging breaking waves” *Coastal Engineering* 53 (2006) 631–655
- [28] M. Salinas-Vazquez and O. Metais, “Large-eddy simulation of the turbulent flow through a heated square duct.,” *J. Fluid Mech.*, 453, pp. 202-208, 2002.
- [29] David, E., “Modélisation des écoulements compressibles et hypersoniques: une approche instationnaire. ,” PhD Grenoble Institute of Technology, 1993.
- [30] Nourgaliev R.R., Dinh T.N., Theofanous T.G. “A pseudocompressibility method for the numerical simulation of incompressible multifluid flows ,” *Int. J. Multiphase Flow*, 30, pp. 901-937, 2004.
- [31] Hirt CW, Nichols BD. “Volume of fluid method for dynamics of free boundaries.” *Journal of Computational Physics* 1981; 39:201-225.
- [32] I. R. Park, K. S. Kim, J. Kim and S. H. Van “A volume-of-fluid method for incompressible free surface flows” *Int. J. Numer. Meth. Fluids* 2009; 61:1331-1362.
- [33] Vinay R. Gopala, Berend G.M. van Wachem, “Volume of fluid methods for immiscible-fluid and free-surface flows” *Chemical Engineering Journal* 141 (2008) 204-221.
- [34] Muzaferija, S., Peric, M.: *Computation of Free Surface Flows Using Interface-Tracking and Interface- Capturing Methods*, chap. 2 in Mahrenholtz O., Markiewicz M.(eds.), *Nonlinear Water Wave Interaction*, pp. 59-100, WIT Press, Southampton, 1999
- [35] M. Moradi Larmaei, Tew-Fik Mahd, “Simulation of shallow water waves using VOF method”, *Journal of Hydro-environment Research* 3 (2010) 208-214
- [36] Nobari, M.R.H., Ketabdari, M.J., Moradi Larmaei, M., 2009. A modified volume of fluid advection method for uniform Cartesian grids. *Applied Mathematical Modelling* 33 (5), 2298-2310.
- [37] Qun Chen, Guangqing Dai and Haowu Liu, “Volume of Fluid Model for Turbulence Numerical Simulation of Stepped Spillway Overflow” *Journal Of Hydraulic Engineering*(2002)
- [38] A. Harten. “High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*”, 49:357–393, 1983.
- [39] A. Harten. “On a class of high resolution total variation stable finite difference schemes”. *SIAM J. Numer. Anal.*, 21:1–23, 1984.

- [40] A. Harten, B. Engquist, S. Osher, y S. Chakravarthy. “Uniformly high-order accurate nonoscillatory schemes” iii. *Journal of Computational Physics*, 71:231, 1987.
- [41] A. Harten y S. Osher. “Uniformly high-order accurate nonoscillatory schemes” i. *SIAM J. Numer. Anal.*, 24:279–309, 1987. R. J. Leveque. Finite volume
- [42] S. Osher and J. A. Sthian, “Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations”, *J. Comput. Phys.*, Vol. 79, 12-49 (1988)
- [43] S. Osher, and R. Fedkiw (2003). “Level Set Methods and Dynamic Implicit Surfaces” Springer-Verlag New York.
- [44] J. A. Sthian, (1996). “Level Set Methods, Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science” Cambridge University Press, USA.
- [45] A Salih,S Ghosh Moulic, “A level set formulation for the numerical simulation of impact of surge fronts” *Sadhana* Vol. 31, Part 6, December 2006, pp. 697–707.
- [46] S. McKee, M.F. Tomé, J.A. Cuminato, A. Castelo and V.G. Ferreira “Recent Advances in the Marker and Cell Method” *Archives of Computational Methods in Engineering*, Vol. 11, 2, 107-142 (2004)
- [47] Songdong Shao, Edmond Y.M. Lo, “Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface” *Advances in Water Resources* 26 (2003) 787–800
- [48] Alejandro Jacobo Cabrera Crespo, “Application of the Smoothed Particle Hydrodynamics model SPHysics to free-surface hydrodynamics” PhD Thesis, Universidad De Vigo Departamento De Física Aplicada, 2008.
- [49] Z. Ghadampour, N. Talebberydokhti, M. R. Hasheim, A. H. Nikseresht, S. P. Neill, “Numerical Simulation Of Free Surface Mudflow Using Incompressible SPH” *IJST, Transactions of Civil Engineering*, Vol. 37, No. C1, pp 77-95
- [50] B. Ataie-Ashtiani, G. Shobeyri , L. Farhadi, “Modified Incompressible SPH method for simulating free surface problems”, Department of Civil Engineering, Sharif University of Technology, Tehran, Iran.
- [51] Michał Szydłowski, Piotr Zima “Two-Dimensional Vertical Reynolds-Averaged Navier-Stokes Equations Versus One-Dimensional Saint-Venant Model for Rapidly Varied Open Channel Water Flow Modelling” *Archives of Hydro-Engineering and Environmental Mechanics* Vol. 53 (2006), No. 4, pp. 295–309
- [52] Il-Ryong Park, Kwang-SooKim, JinKim, Suak-HoVan, “Numerical investigation of the effects of turbulence intensity on dam-break flows” *Ocean Engineering* 42(2012)176–187

- [53] R.R. Nourgaliev, T.N. Dinh, and T.G. Theofanous. “The ‘Numerical Acoustic Relaxation (NAR)’ Method for Time-Dependent Incompressible Single- and Multiphase Flows” Center for Risk Studies and Safety, University of California, Santa Barbara, USA.
- [54] Berger M, Olinger J., “Adaptive mesh refinement for hyperbolic partial differential equations,” *J. Comput. Phys.*, 53, pp. 6484, 1984.
- [55] R.R. Nourgaliev *, T.N. Dinh, T.G. Theofanous. “Adaptive characteristics-based matching for compressible multifluid dynamics” Center for Risk Studies and Safety, University of California at Santa Barbara, CA 93117, USA.
- [56] Sussman M, Almgren A, Bell J, Colella P, Howell L, Welcome M., “An adaptive level set approach for incompressible two-phase flows,” *J. Comput. Phys.*, 148, pp. 81124, 1999.
- [57] Xin Lv, Qingping Zou, Yong Zhao, Dominic Reeve, “A novel coupled level set and volume of fluid method for sharp interface capturing on 3D tetrahedral grids” *Journal of Computational Physics* 229 (2010) 2573–2604
- [58] I. R. Park, K. S. Kim, J. Kim and S. H. Van, “A volume-of-fluid method for incompressible free surface flows” *International Journal For Numerical Methods In Fluids* 2009; 61:1331–1362
- [59] Sun, D.L., Tao, W.Q., “A coupled volume-of-fluid and level set (VOSET) method for computing incompressible two-phase flows,” *Int. J. Heat Mass Transf.*, 53, pp. 445-455, 2010
- [60] Z. Wang, J. Yang, B. Koo, F. Stern, “A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves,” *Int. J. Multiphase Flow*, 35, pp. 227-246, 2009
- [61] Y. Wang, S. Simakhina, M. Sussman , “A hybrid level set-volume constraint method for incompressible two-phase flow,” *J. Comput. Phys.*, 231, pp. 6438-6471, 2012
- [62] Sander Pieter Van Der Pijl, “Computation of Bubbly Flows with a Mass-Conserving Level-Set Method” Tesis, Technische Universiteit Delft.
- [63] S. P. van der Pijl, A. Segal, C. Vuik and P. Wesseling “A mass-conserving Level-Set method for modelling of multi-phase flows”, *Int. J. Numer. Meth. Fluids* 2005; 47:339–361.
- [64] Aris, R. *Vectors, Tensors and the Basic Equations of Fluid Mechanics*. Prentice Hall International Series in the Physical and Chemical Engineering Sciences. 1962.
- [65] Warnatz, J., Mass, U., Dibble, R. W. *Combustion*. Springer. 1996.
- [66] Sussman M, Smereka P, Osher S.J., “A level set approach to computing solutions to incompressible two phase flow. ,” *J. Comput. Phys.*, 59, pp. 114-146, 1994.

- [67] Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang “A PDE-Based Fast Local Level Set Method” *Journal of Computational Physics* 155, 410–438 (1999)
- [68] Fedkiw, R., Merriman, B., Donat, R., Osher, S., 1998. The penultimate scheme for systems of conservation laws: finite difference ENO with Marquina’s flux splitting. In: Hafez, M. (Ed.), *Progress in Numerical Solutions of Partial Differential Equations*, Arcachon, France, July 1998.
- [69] Jiang, G.S., Peng, D.P., 2000. “Weighted ENO schemes for Hamilton–Jacobi equations”. *SIAM Journal on Scientific Computing* 21, 2126–2143.
- [70] Jiang, G.S., and Shu, C.-W., “Efficient Implementation of Weighted ENO Schemes, *Journal of Computational Physics*”, 126, 202-228, 1996.
- [71] Belda García Ana María, 2010. “Técnicas de interpolación WENO y su aplicación al procesamiento de imágenes”. *Universitat de València*
- [72] Fedkiw, R.P., Aslam, T., Merriman, B., and Osher, S., “A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method)”, *Journal of Computational Physics*, 152, 457-492 (1999).
- [73] William L. Oberkampf and Timothy G. Trucano, “Verification and Validation in Computational Fluid Dynamics”, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550.
- [74] William L. Oberkampf and Timothy G. Trucano, “Verification and validation benchmarks”, *Nuclear Engineering and Design* 238 (2008) 716–743
- [75] Advisory Group For Aerospace Research Development, “A Selection of Experimental Test Cases for the Validation of CFD Codes”, AGARD Advisory Report No. 303, Agosto 1994.
- [76] Martin, J.C., and Moyce, W.J., “An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane”, *Philos. Trans. R. Soc. London Ser. A*, 244, 312, 1952.
- [77] Koshizuka, S., and Oka, Y., “Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid, *Nuclear Science and Engineering*”, 123, pp.421-434, 1996.
- [78] Janosi, I. M., Jan, D., Szabo, K. G., and Tel, T. 2004. “Turbulent drag reduction in dam-break flows.” *Exp. Fluids*, 37, 219–229.
- [79] Abbas Khayyer Hitoshi Gotoh, “On particle-based simulation of a dam break over a wet bed” *Journal of Hydraulic Research* Vol. 48, No. 2 (2010), pp. 238–249 doi:10.1080/00221681003726361

- [80] Colagrossi, A., and Landrini, M. “Numerical simulation of interfacial flows by smoothed particle hydrodynamics.” *J. Comput. Phys.*, 191, 448–475, 2003.
- [81] Gómez-Gesteira, M., and Dalrymple, R. A. “Using a 3D SPH method for wave impact on a tall structure.” *J. Waterway, Port, Coastal, Ocean Eng.*, 1302, 63–69, 2004.
- [82] Violeau, D., and Issa, R. “Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview.” *Int. J. Numer. Methods Fluids*, 53, 277–304, 2006.
- [83] Monaghan, J. J. “Simulating free surface flows with SPH.” *J. Comput. Phys.*, 110, 399–406, 1994.
- [84] K.M.T. Kleefsman, G. Fekken, A.E.P. Veldman, B. Iwanowski, B. Buchner “A Volume-of-Fluid based simulation method for wave impact problems.” *Journal of Computational Physics* 206 (2005) 363–393