UNIVERSIDADE DE VIGO
DEPARTAMENTO DE FÍSICA APLICADA
Environmental Physics Laboratory

PhD Thesis

Smoothed Particle Hydrodynamics model for civil and coastal engineering applications

Memoria presentada por
Anxo Barreiro Aller
para optar al título de:
DOCTOR INTERNACIONAL POR LA UNIVERSIDAD DE VIGO
Julio, 2015

A Juancho

"I was born not knowing and have had only a little time to change that ***here…***

… and there."

- Richard P. ***Feynman***

Acknowledgements/Agradecementos

Quero dar as gracias primeiro de todo a Moncho, sen a súa confianza nun rapaz que remataba a carreira fai 5 anos nada de isto sería posible, tamén agradecerlle a guía e consello durante todo este tempo. Gracias por apostar por min.

Dar as gracias o meu director de tese e compañeiro Alex, gracias pola paciencia, que sei que foi moita, polos momentos dentro e fora do traballo, sen a súa dedicación e esforzo esta tese non tería chegado a bo porto. Tamén a Jose, meu director de tese máis recente e dende sempre compañeiro de traballo, pola dedicación que mostra no traballo e o seu esforzo que fan posibles traballos como a presente tese, tamén darlle gracias polos consellos e bos momentos dentro e fora do laboratorio.

Tamén darlle gracias os meus compañeiros de traballo os que quedan e os que marcharon, Orlando, Fran, Ángel, Xurxo, Alexandre, e Isabel, polo bo ambiente de traballo, as risas, e a moita axuda que me dan.

Darlle as gracias a meus compañeiros de carreira, Serxio, Rul, Anxo, Nacho, e Bárbara, pola súa compañía e axuda xa fora en Santiago ou Ourense.

A miña familia, tanto a Barreiro como a Aller, e moi especialmente a meus pais. Gracias pai por ensinarme que sempre se pode empezar de cero que non hai barreira que non se poida superar. Gracias mama por todo o apoio e por todo o esforzo que fas día a día sempre pensado primeiro na familia antes que en ti mesma. Dedicarlle tamén este traballo a meu irmán, temos moitas diferencias apréciote moito e sabes que sempre poderás contar comigo.

A Juancho e Ana darlles as gracias polo seu apoio, pese a que non compartamos os mesmos apelidos sodes familia.

A miña “familia” de Noia por todo o cariño que me dan sen importar os anos.

Darlles as gracias a meus amigos especialmente a Sosi, Javi, Bruno, Gandalf, Tato, Sergio, María, Helena, Gabi, e David gracias por todos por estar nos bos momentos (as risas, as noites, os cafés, as cañas) e o apoio nos momentos fracos.

Por último darlle as gracias a María, polo teu apoio, paciencia e compresión, estes 5 anos non foron sinxelos pero mereceren a pena, gracias por facerme tan feliz.

This work was partially supported by Xunta de Galicia under Axudas de apoio á etapa predoutoral do Plan Galego de Investigación, Innovación e Crecemento 2011-2015, project Programa de Consolidación e Estructuración de Unidades de Investigación Competitivas (Grupos de Referencia Competitiva), funded by European Regional Development Fund (FEDER) and Ministerio de Economía y Competitividad under project BIA2012-38676-C03-03.

Resumen

Este trabajo se centra en el desarrollo del modelo *DualSPHysics*, un código basado en el método Smoothed Particle Hydrodynamics (SPH). Durante los últimos años la computación de alto rendimiento mediante el uso de tarjetas gráficas nos ha permitido desarrollar un código capaz de simular millones de partículas en un tiempo de cálculo razonable. Esto hace posible que problemas con geometrías complejas y de tamaño real se puedan simular con una alta resolución y que los tiempos de ejecución sean del orden de unas horas, por lo que se pueden ya abordar problemas reales de ingenierías. En este trabajo se muestran varios ejemplos de aplicación de *DualSPHysics* para resolver problemas de ingeniería civil y de costas.

La presente tesis tiene la siguiente estructura.

Capítulo 1, esta sección introduce el trabajo realizado y proporciona una descripción simulaciones numéricas, métodos sin malla y SPH. El código *DualSPHysics* es presentado en esta sección.

El capítulo 2 presenta la formulación matemática de SPH y una descripción de los conceptos básicos de este modelo numérico, como son, interpolandos integrales y el concepto de *kernel*.

El capítulo 3 proporciona una descripción de los diferentes pasos que se llevan a cabo en una simulación de SPH y como estos pasos se han implementado en *DualSPHysics*.

En el capítulo 4 se emplea *DualSPHysics* para la protección de costas. Se muestra la validación del modelo comparando los resultados con datos experimentales y también un caso de aplicación en el que se usa *DualSPHysics* en una configuración costera con mobiliario urbano.

El capítulo 5 trata sobre la implementación de los amarres. En esta sección se muestran las ecuaciones a resolver y la validación tanto de

los amarres como de los objetos flotantes. En la parte final se presentan dos aplicaciones de esta nueva funcionalidad.

En el capítulo 6 se presenta la integración de dos tecnologías en auge: por un lado la fotogrametría UAV, una técnica que permite obtener reconstrucciones 3D muy precisas de cualquier área y *DualSPHysics*, un modelo SPH capaz de simular geometrías muy complejas en una escala razonable de tiempo. En esta sección se explica cómo funcionan estas dos técnicas en conjunto y se muestra un ejemplo de aplicación de las mismas.

El capítulo 7 proporciona al lector las conclusiones de este trabajo, además en su parte final muestra algunas líneas de investigación que pueden surgir después de la realización de esta tesis.

A) INTRODUCCIÓN

Todos los procesos en la naturaleza atienden a una serie de leyes que pueden ser descritas con fórmulas matemáticas. Estas ecuaciones pueden ser muy complicadas o incluso casi imposibles de resolver en un tiempo razonable. Para poder afrontar este tipo de problemas aparece el modelado numérico. A la hora de resolver las ecuaciones se pueden afrontar a través de una aproximación Lagrangiana o Euleriana, y para discretizar el medio existen otras dos opciones, con o sin la ayuda de una malla. Normalmente los métodos que hacen uso de mallas son métodos Eulerianos y los que resuelven los problemas sin el uso de mallas suelen ser métodos Lagrangianos.

Los métodos basados en una malla (elementos finitos, diferencias finitas y volúmenes finitos) son muy robustos, están bien desarrollados y han sido probados a un amplio abanico de aplicaciones en las que proporcionan resultados muy precisos. Estos métodos de malla son ideales para sistemas con dominios perfectamente delimitados y para simulaciones donde los contornos permanecen fijos. Sin embargo la creación de la malla puede resultar muy ineficiente si el sistema es complejo. En este trabajo nos centraremos en métodos sin malla, los cuales se usan tanto en mecánica de sólidos como en dinámica de fluidos. Los métodos que no usan malla permiten simular más fácilmente problemas en los que se estudien geometrías muy complejas, grandes deformaciones, discontinuidades en el flujo del fluido y singularidades del material. Dentro de este gran grupo de modelos numéricos está el método llamado SPH.

SPH es un método computacional para la resolución de problemas de dinámica fluidos. Se trata de un método Lagrangiano, desarrollado en los años setenta para resolver problemas astrofísica (Gingold & Monaghan, 1977; Lucy, 1977) y en los últimos años este método se ha aplicado con éxito en numerosos campos de la dinámica de fluidos.

En el método SPH no se usa malla, eliminando así las restricciones que derivan de emplear los métodos que usan una malla. SPH discretiza el fluido como un conjunto de puntos o nodos llamados partículas. En SPH se resuelve la integral de las ecuaciones de movimiento de la dinámica de fluidos en cada punto en el formalismo Lagrangiano, calculando los valores de las variables físicas más representativas (velocidad, posición, densidad y presión) para cada partícula como interpolación de los valores de las partículas más próximas. Para representar el paso de un medio continuo, como lo es un fluido, a un medio discreto, partículas, se emplea la función kernel. Esta función tiene un soporte compacto dentro de una región que viene determinada por una distancia mucho más pequeña que la escala característica del problema. Ésta distancia, definida por el kernel, es la distancia de interacción de las partículas.

El código SPH llamado SPHysics es un código numérico desarrollado para el estudio de fluidos de superficie libre. El código está escrito en FORTRAN, es un código libre y se puede descargar gratuitamente desde la web www.sphysics.org.

A pesar de que el código SPHysics puede proporcionar una descripción precisa del fluido, su mayor limitación es el alto coste computacional al estar implementado en serie, lo cual hace imposible su uso de manera eficiente para estudiar problemas reales de ingeniería. El tiempo computacional se incrementa con el aumento de partículas que componen los sistemas, por lo que muchos problemas son inviables por la complejidad de los mismos. Por este motivo, HPC (High Performance Computing) es la herramienta necesaria para alcanzar nuevas cotas en el modelado de problemas de ingeniería, en el caso de *DualSPHysics* se usan las GPU (Graphics Processing Units). Las GPU aparecen como una alternativa accesible para acelerar los métodos SPH usando un potente modelo de programación en paralelo donde las tarjetas gráficas se usan como dispositivo de ejecución. Ésta es una tecnología nueva desarrollada por la industria de los videojuegos y que proporciona un alto rendimiento con un gasto económico razonable. Así, las GPUs hacen posible el estudio de problemas reales de ingeniería mediante modelos SPH. De este modo se desarrolló el código *DualSPHysics* escrito en C++ y CUDA (lenguaje de programación para las GPU) el cual se puede descargar en la web www.dual.sphysics.org.

Gracias a este paso adelante dado por *DualSPHysics* se puede afrontar todo un nuevo rango de casos y aplicaciones con un coste razonable de tiempo.

A continuación se presentan los diferentes casos estudiados y las implementaciones llevadas a cabo en *DualSPHysics* para poder afrontarlos satisfactoriamente.

B) PROTECCIÓN COSTERA

Las grandes olas que llegan a tierra pueden generarse por el paso periódico de tormentas por zonas cercanas a la costa, por olas de tsunami debidas a terremotos submarinos y por grandes inundaciones debidas a la entrada de los huracanes en tierra firme. Todos estos fenómenos dan lugar a grandes daños en las estructuras costeras, como diques, puertos, bahías, paseos marítimos, espigones y todo tipo de mobiliario urbano.

Para poder aplicar el código con fiabilidad se necesita una validación con datos experimentales. Esta validación se divide en tres partes, en la primera se valida con la solución analítica para un caso de presión hidrostática en el que se computa la presión contra una de sus paredes verticales. Esta validación se realiza en un tanque en 2D de un metro de ancho por un metro de alto, la solución analítica de la presión será $^{1}/_{2} \cdot \rho \cdot g \cdot H^{2} \cdot B$, donde *H=1m, B* corresponde a la superficie de la base, ignorada al ser un caso 2D. Para este caso concreto la fuerza será *4,905 N*, en la Fig. 1 se muestra la serie temporal para los valores de la fuerza usando diferentes resoluciones o *dp* (espacio inicial entre partículas). En esta figura se pueden ver unas oscilaciones, esto se debe a que a la simulación le lleva un tiempo llegar a un estado estable, también debemos tener en cuenta que cuanto menor es la resolución de nuestro sistema menores son las oscilaciones, tanto en magnitud como en la duración de las mismas.

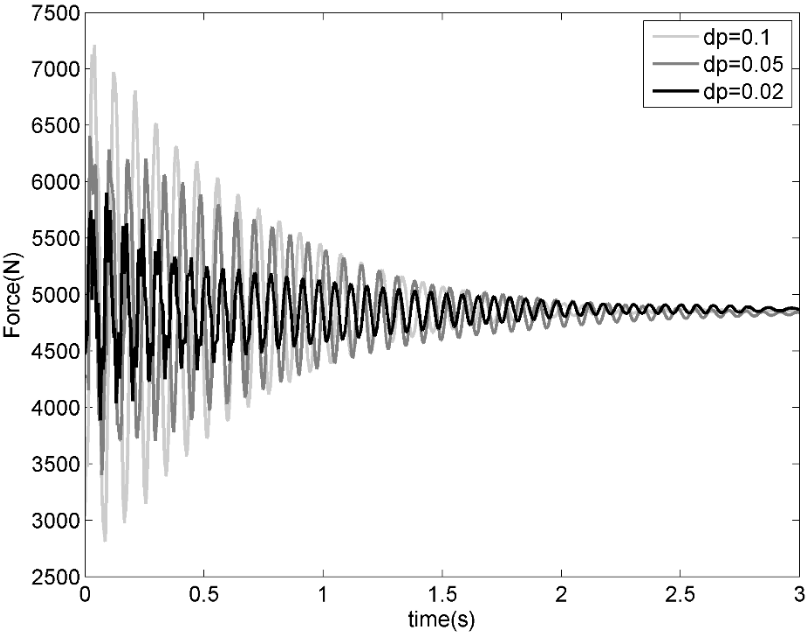


Figura 1. Serie temporal de fuerzas ejercidas sobre un muro vertical para diferentes resoluciones.

Como se puede observar en la Fig. 1, los datos numéricos convergen con el resultado teórico esperado (*4,905 N*).

La siguiente validación es la propagación de olas, gracias a la colaboración de la Universitat Politècnica de Catalunya (UPC) se puede mostrar la comparación entre olas propagadas numéricamente con las olas que se propagaron en el canal físico. En el tanque de olas CIEMito se llevó a cabo un experimento de propagación de olas regulares (0.07m de altura de ola y 1s de periodo) generadas por un pistón. La configuración del tanque es 5.05m de largo, 0.38 metros de ancho, con una altura inicial del agua de 0.25m. Los valores de altura de ola fueron medidos en tres puntos diferentes localizados en 0.38m (WG1), 1.83m (WG2) y 3.12m (WG3), estas posiciones han sido elegidas para evitar efectos de reflexión en las medidas. En la Fig. 2 podemos observar como la concordancia entre los datos experimentales y los datos proporcionados por *DualSPHysics* es muy buena, pudiendo concluir que el modelo es capaz de propagar olas de manera satisfactoria.

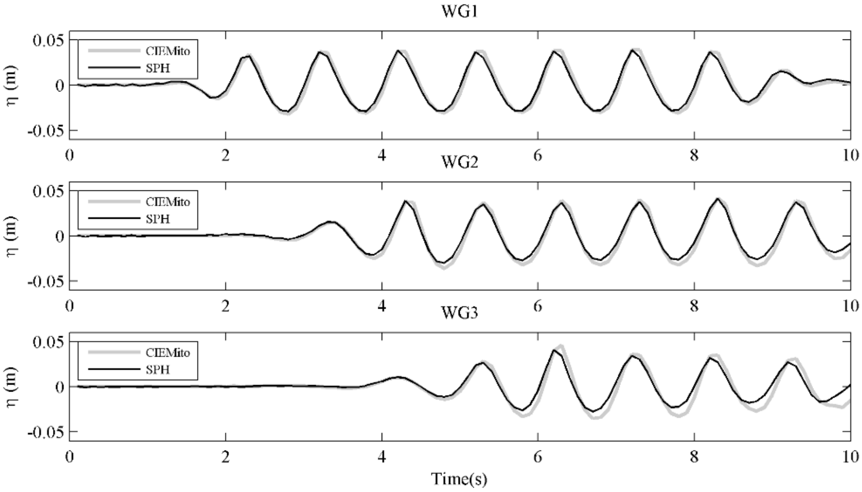


Figura 2. Comparación entre los datos experimentales y los datos proporcionados por *DualSPHysics* para la elevación del agua en tres diferentes posiciones para el caso de propagación de olas.

La última de las tres validaciones llevadas a cabo para el caso de protección de costas es la evolución de un “dambreak” que colisiona contra una estructura. En este caso se validará la fuerza ejercida por el agua contra un obstáculo inmóvil. La configuración del caso es la que sigue, una masa de agua (40cm x 61cm x 30cm) esta confinada en un lado de en un tanque de 160 cm de longitud, este volumen de agua se libera con una compuerta y evoluciona hasta colisionar con el obstáculo. Este objeto se encuentra situado a 50 cm de la compuerta, las dimensiones del obstáculo son 12 cm x 12 cm x 75 cm y está colocado en el centro del tanque.

Esta validación se ha utilizado también en Gómez-Gesteira & Dalrymple (2004), en la Fig. 3 se muestra la comparación entre los datos experimentales y los datos proporcionados por *DualSPHysics* y se puede comprobar que los datos tienen una buena concordancia entre sí.

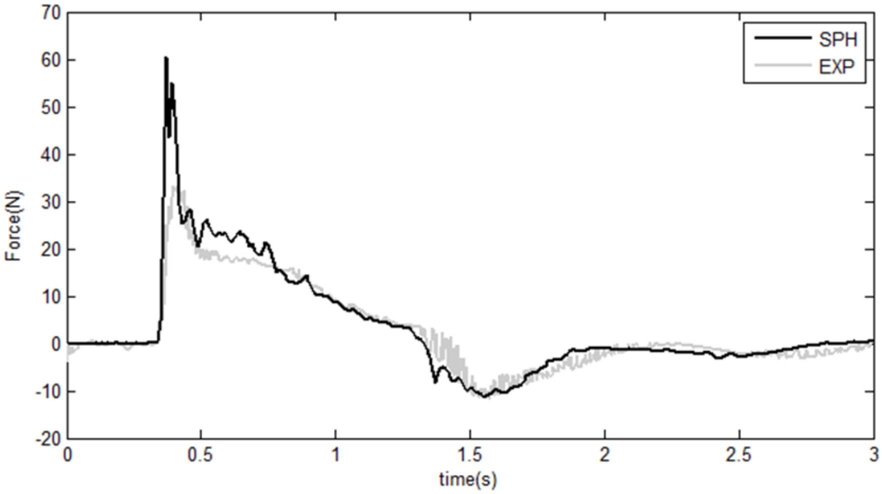


Figura 3. Comparación entre los datos proporcionados por *DualSPHysics* y los datos experimentales para la interacción fluido-estructura.

Por todo esto podemos decir que el modelo *DualSPHysics* es apto para evaluar eventos en los que el agua impacta contra una estructura y casos en los que es necesario la propagación de olas.

La aplicación que se presenta es un escenario típico en cualquier costa, una balaustrada en un paseo marítimo que protege la zona de paseo. Un esquema de este caso se muestra en la Fig. 4.

Se procede a propagar cuatro configuraciones diferentes de oleaje resumidas en la tabla 1. Para estas cuatro configuraciones del oleaje se toman las siguientes medidas, altura de ola antes del impacto en la balaustrada, fuerzas ejercidas en la balaustrada, y fuerza y momento ejercidos en la farola.

Tabla 1. Parámetros del generador de olas.

wave	frequency (Hz)	amplitude (m)	Wave length (m)
1	0.12	1.8	36.211
2	0.12	2.0	36.211
3	0.14	1.8	30.816
4	0.14	2.0	30.816

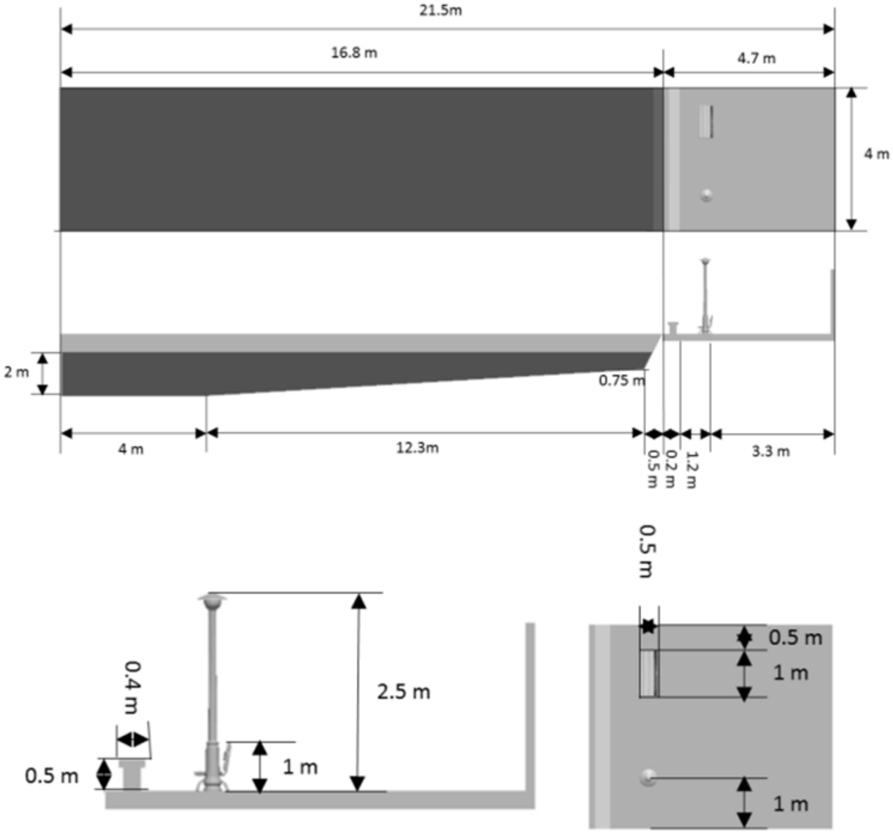


Figura 4. Configuración inicial del caso de aplicación, balaustrada y mobiliario urbano.

En la Fig. 5 se puede observar la altura que alcanzan las olas, en esta figura se analiza el máximo de altura en todo el dominio antes de que se produzca el impacto para las cuatro diferentes configuraciones de ola. Se puede observar que la cuarta ola “wave #4” es la que alcanza una mayor altura, seguida por “wave #3”, “wave #2” y “wave #1”.

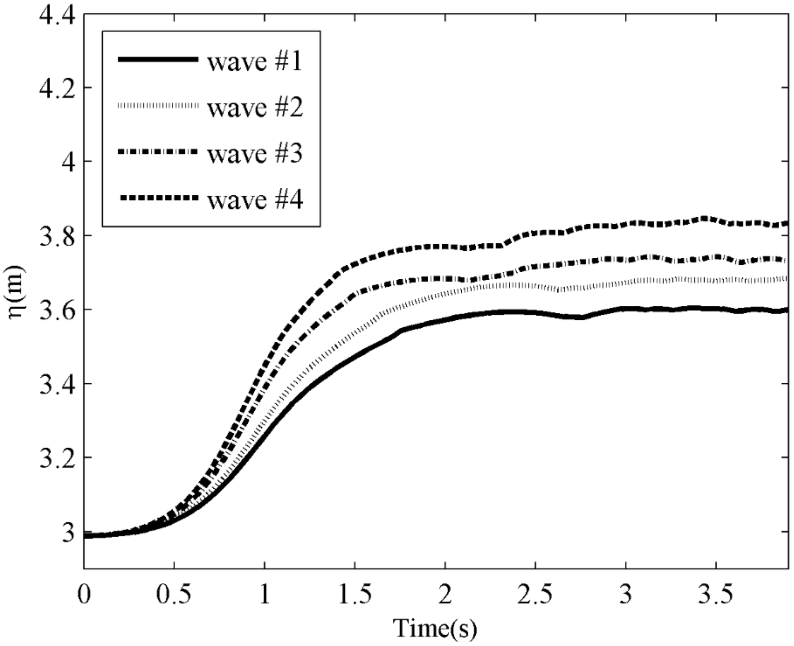


Figura 5. Serie temporal de la elevación de la superficie del agua para las cuatro configuraciones de ola.

Además de la elevación de la superficie del agua, se han estudiado las fuerzas que ejercen las cuatro olas propagadas en las diferentes estructuras presentadas en la Fig. 4. La primera de ellas en recibir el impacto es la balaustrada. En la Fig. 6 podemos observar la fuerza que experimenta este elemento con el impacto de las cuatro olas.

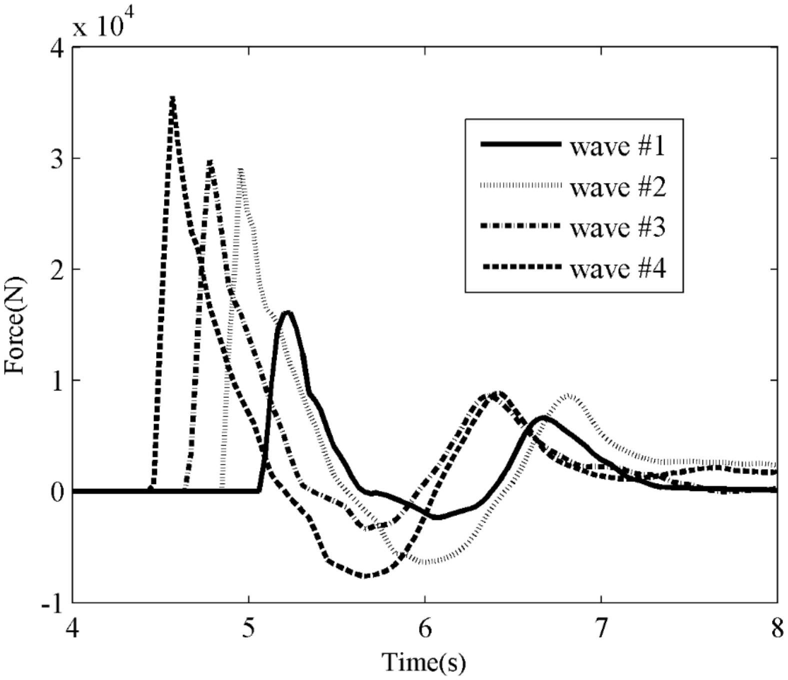


Figura 6. Serie temporal de las fuerzas experimentadas por la balaustrada.

Durante los primeros 4.4 segundos de la simulación no hay valores de fuerzas dado que las olas todavía se están propagando hacia el obstáculo. Se puede observar que "wave #4" causa los valores más altos y más bajos de la fuerza, y cómo se puede ver la intensidad de estos impactos concuerda con el orden de altura que describimos anteriormente. La aparición de valores negativos se produce cuando la ola impacta en el suelo después de superar la balaustrada y el agua se proyecta contra la parte trasera del obstáculo.

Otro objeto en el cual realizamos el estudio es la farola, en la Fig. 7 podemos ver las fuerzas ejercidas contra este objeto por las 4 olas.

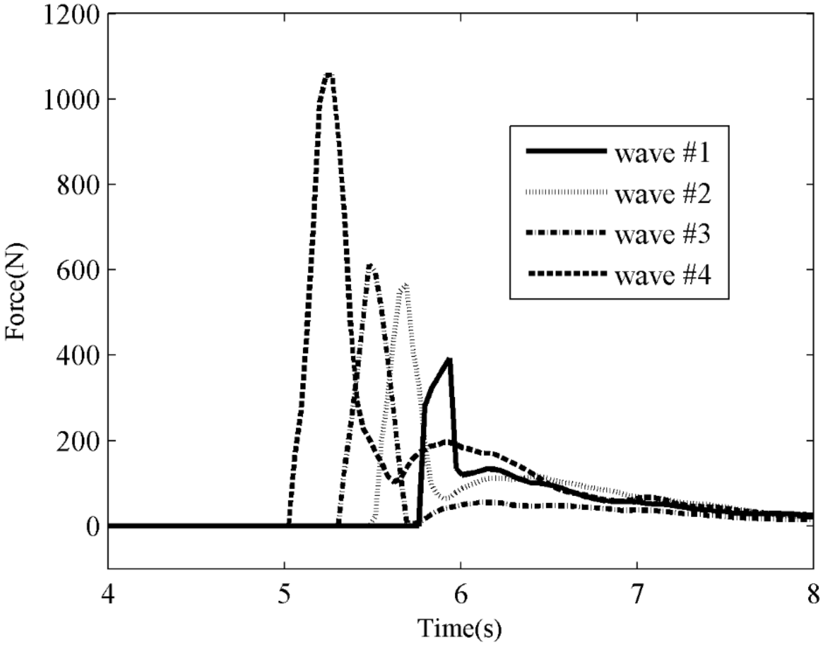


Figura 7. Fuerzas ejercidas contra la farola.

En este caso podemos ver como existe una gran diferencia entre las 4 olas propagadas, esto se debe a que las olas no consiguen superar la balaustrada de la misma forma. En la Fig. 8 se puede apreciar como el valor de la fuerza ejercida por “wave #4” supera ampliamente los valores de “wave #3”, “wave #2” y “wave #1”. Esto se debe a que al ser un objeto prácticamente bidimensional, su altura es mucho mayor que su anchura, lo cual hace que impactos más altos y que a su vez cubren mayor proporción del objeto den lugar a mayores fuerzas.

Otro test llevado a cabo con la farola ha sido el estudio del momento ejercido sobre el objeto, a veces no es solo la fuerza que se aplica lo que propicia roturas en los objetos sino también como se aplica esa fuerza. La farola es un caso excelente para este estudio por la “bidimensionalidad” que comentábamos anteriormente

La Fig. 8 muestra el momento sobre el objeto.

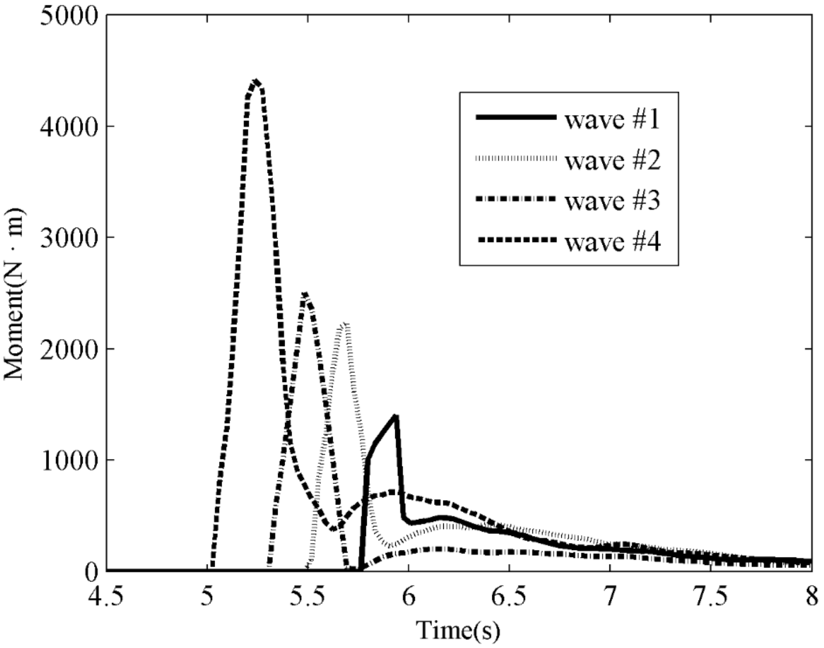


Figura 8. Serie temporal del momento ejercido sobre la farola.

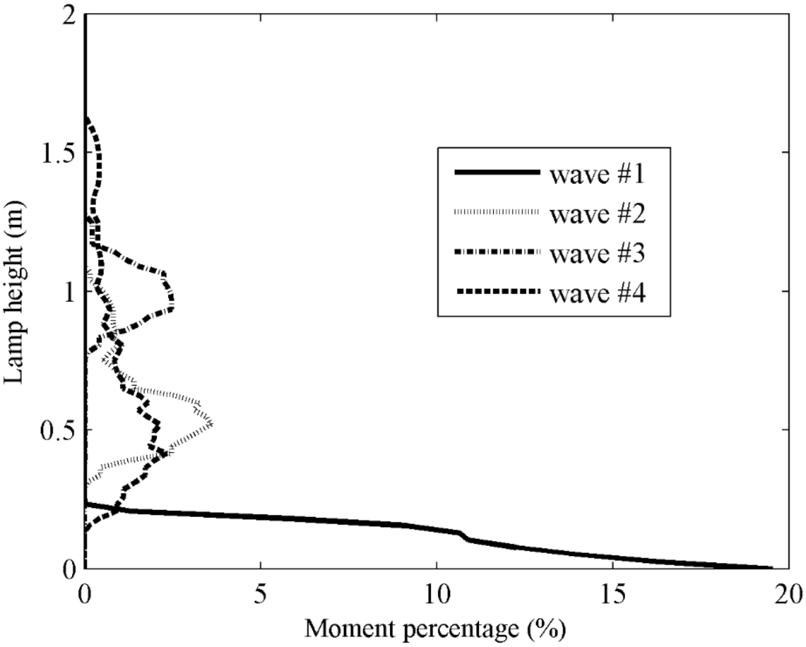


Figura 9. Distribución del momento en la farola.

En la Fig. 9 se puede observar como el momento ejercido no está homogéneamente distribuido por el objeto lo cual demuestra que este tipo de estudios son importantes dado que aunque se ejerza menos momento si este se concentra en un punto débil de la estructura esta puede verse dañada.

C) OBJETOS FLOTANTES CON AMARRES

Las estructuras situadas mar a dentro son más comunes cada día, ya sea en la extracción de gas y petróleo o en el uso de elementos para aprovechar energías renovables (energía eólica, energía de las olas, o energía de las mareas). Estos elementos se sitúan en lugares donde las condiciones del mar hacen imposible el uso de soportes fijos, ya sea por la profundidad o el oleaje. Para asegurar este tipo de estructuras se usan los amarres. El objetivo de esta implementación es realizar simulaciones más precisas de las estructuras situadas mar adentro, para ello es necesario desarrollar en *DualSPHysics* un módulo que sea capaz de simular los amarres. Se eligió la formulación presentada en Faltinsen (1995) para resolver las fuerzas causadas por los amarres.

Para validar los amarres se han elegido los trabajos Johanning et al. (2006) y Johanning et al. (2007). Las validaciones realizadas son para uno (Fig. 10 y Fig. 11) y dos (Fig. 12) amarres, en los tres casos se ha obteniendo una muy buena correlación entre los datos de referencia y los datos obtenidos por *DualSPHysics*. En todos los casos el parámetro de estudio es la tensión de la cuerda. Se estudia más de un caso para un solo amarre para comparar si la implementación da buenos resultados independientemente de la escala, lo cual es cierto. En la caso de más de un amarre además de la tensión para cada una de las cuerdas se estudia la fuerza resultante de los dos amarres.

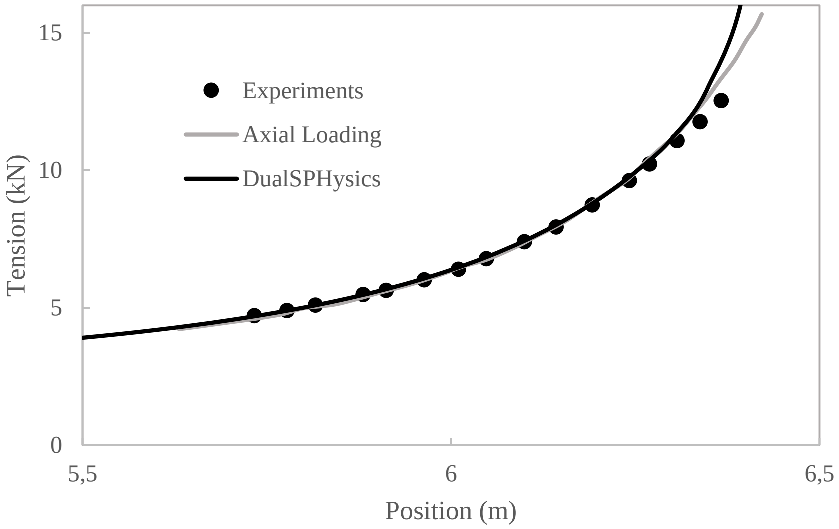


Figura 10. Comparación entre los datos experimentales, *DualSPHysics* y datos numéricos para la tensión horizontal

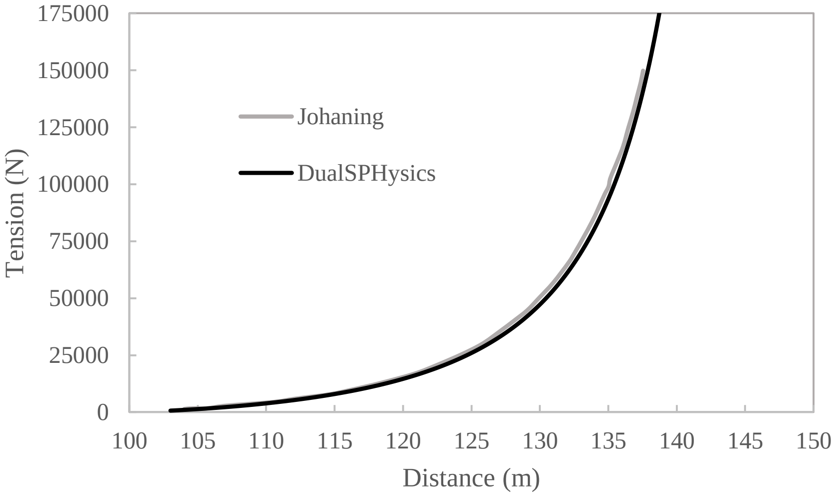


Figura 11. Comparación de la tensión horizontal para un amarre.

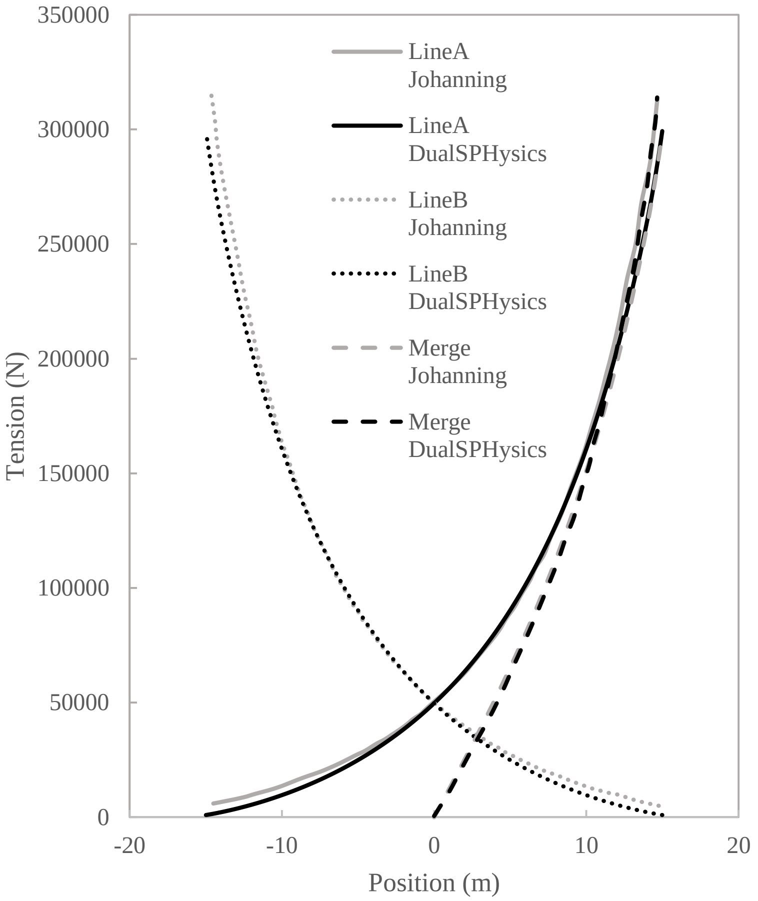


Figura 12. Datos para dos amarres enfrentados, líneas sólidas, y datos de tensión resultante, líneas punteadas, datos numéricos línea gris y *DualSPHysics* línea negra.

Estas tres validaciones nos permiten decir que *DualSPHysics* es capaz de resolver correctamente los amarres, para una cadena (Fig.10), con diferentes escalas (Fig. 11) y para un objeto con múltiples cadenas proporcionando la fuerza resultante adecuada (Fig. 12).

Como ejemplos de aplicación de esta nueva implementación tenemos dos casos diferentes. El primero se simula un barco amarrado con una sola línea en la proa con olas golpeando su costado y el segundo se compone de una base flotante de un aerogenerador amarrado con varias líneas.

En el primer caso podemos observar en la Fig. 13 como el barco se mueve por el efecto de las olas y lentamente se va girando la embarcación hasta que el amarre se estira por completo. Al alongarse completamente el amarre impide que la proa del barco se aleje más y la embarcación acaba con su proa enfrentada a la dirección de las olas. En el estado final el barco ofrece menos resistencia a las olas.

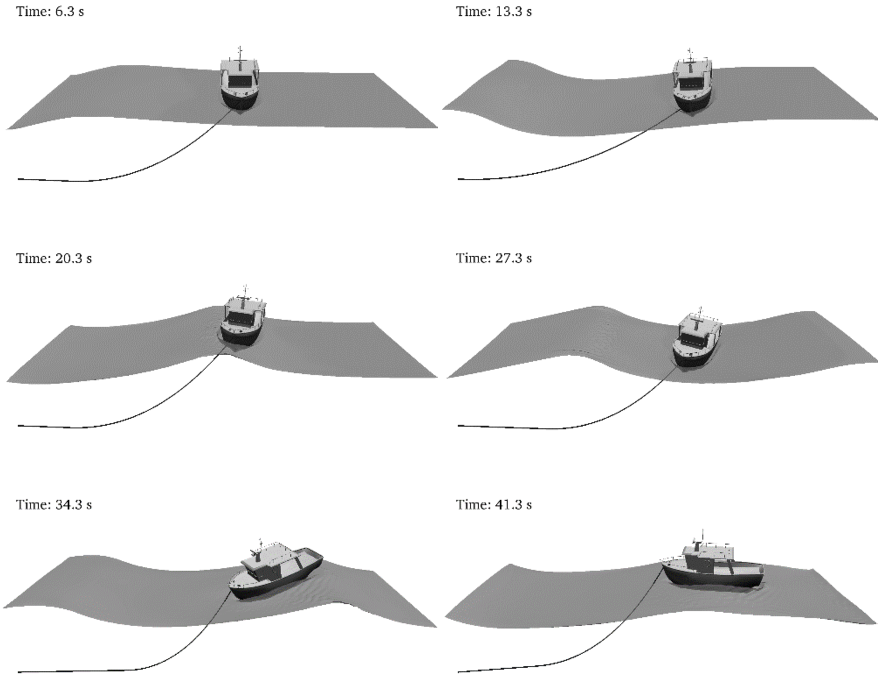


Figura 13. Diferentes momentos de un barco amarrado bajo la acción de las olas.

El segundo caso de aplicación muestra la base de un aerogenerador asegurada con tres amarres. Este es un caso típico para lugares en que la profundidad o las condiciones de oleaje no permiten usar estructuras fijas. Este objeto flotante se ha expuesto a oleaje extremo para poner a prueba la implementación llevada a cabo.

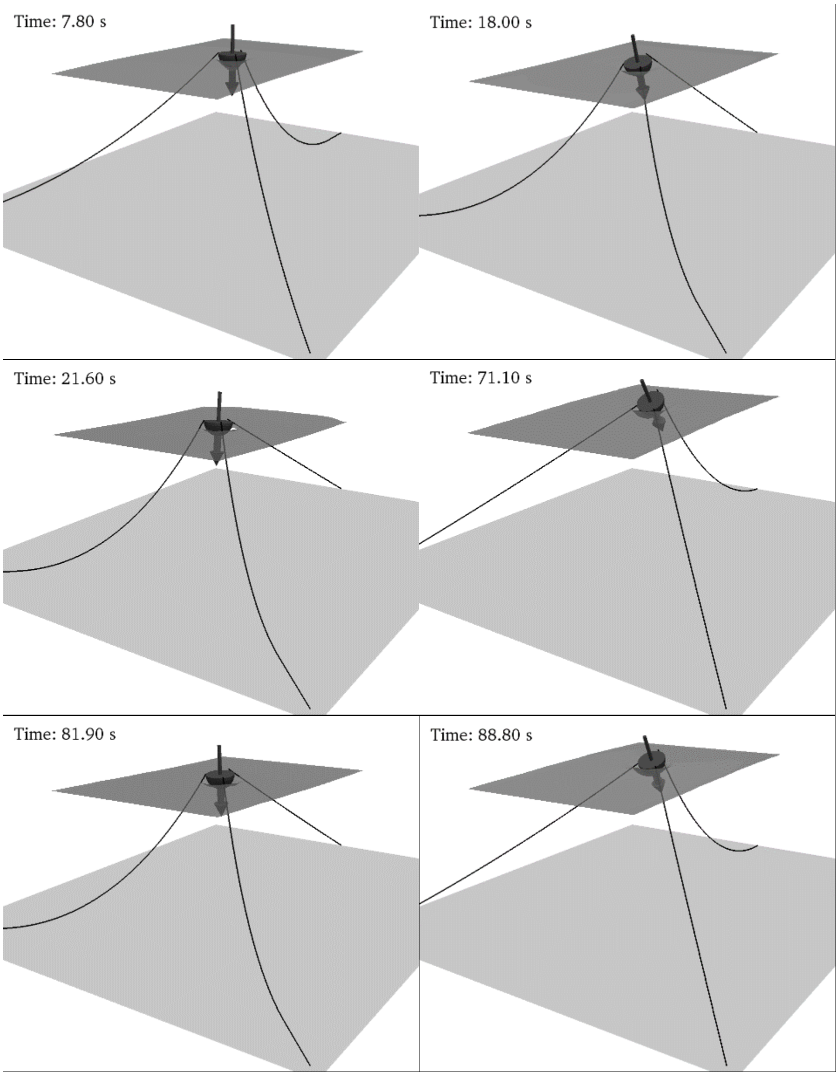


Figura 14 Diferentes momentos de la simulación de la plataforma de un aerogenerador bajo el efecto de olas.

Como se puede observar en la figura, las olas golpean la estructura y hacen que se desplace hasta que los dos amarres delanteros se extienden totalmente y tiran de la estructura para mantenerla en posición (tiempo 71.1s y 88.8s).

Estos dos ejemplos muestran que la solución implementada es capaz de lidiar adecuadamente con los amarres tanto en regímenes normales con en situaciones extremas.

D) ESCORRENTÍA EN UN TERRENO REAL

En esta sección se aplica el modelo *DualSPHysics* a un problema de ingeniería civil. El código se usa en un caso real para simular la escorrentía extrema en un terreno, la geometría de este terreno se ha obtenido usando fotogrametría UAV (Unmanned Aerial Vehicle). En este caso de aplicación se simula como el agua de una tormenta fluye hacia una carretera cercana. Para simular este tipo de efectos se ha implementado un sistema de entrada de agua que emula las condiciones meteorológicas que se pueden dar en caso de una tormenta severa. Se evaluará que cantidad de agua llega a la carretera y si las medidas de protección llevadas a cabo son efectivas, en este caso una zanja.

La fotogrametría UAV es una nueva técnica que permite obtener la geometría de cualquier área con una alta resolución. El modelo 3D de una zona se obtiene mediante la toma de fotografías desde un dron, estas fotografías son tomadas automáticamente y cubren toda la zona que se quiere estudiar. Las fotografías deben solaparse para tener puntos en común y así realizar una reconstrucción con un programa de fotogrametría (PHOTSCAN) además, el dispositivo graba las posiciones GPS de cada fotografía para una mayor precisión. En la Fig. 15 se puede ver el recorrido y las posiciones en las que el dron tomó las fotografías para generar la geometría empleada en el estudio de la escorrentía.

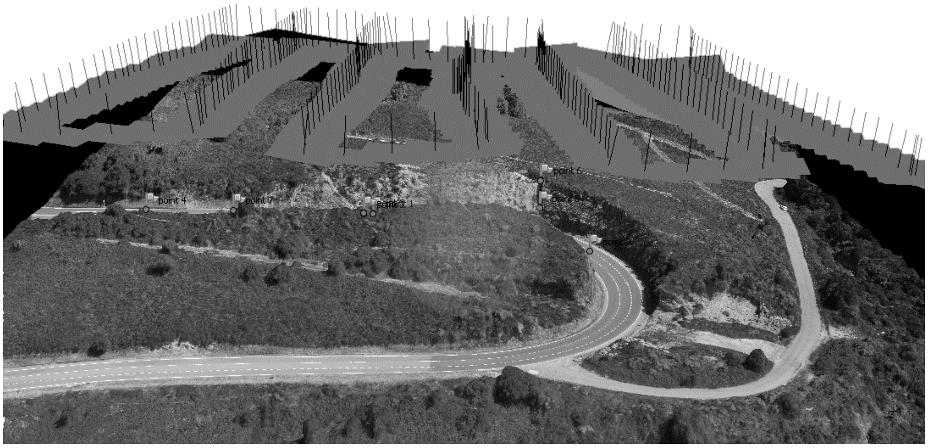


Figura 15. Recorrido del dron y posiciones de las fotografías.

El programa de fotogrametría genera una nube de puntos del área de estudio. Para poder utilizar estos datos con *DualSPHysics* se debe transformar esta nube de puntos en partículas SPH, más concretamente en partículas de contorno. Como paso intermedio de esta transformación se generará una superficie descrita por triángulos, este es un paso crucial dado que cierra posibles agujeros que se pudiesen generar en la nube de puntos.

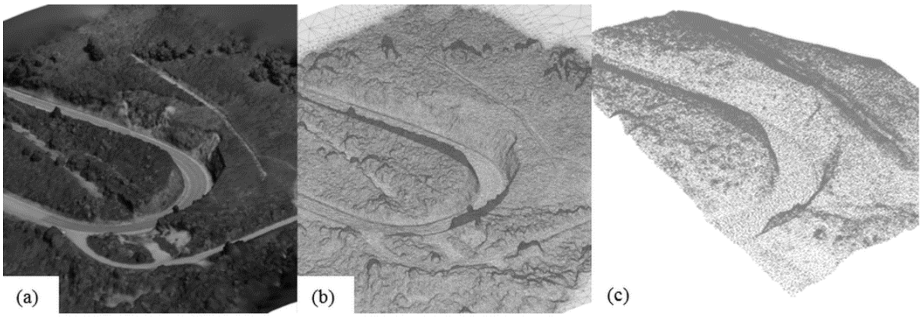


Figura 16. (a) Digital Surface Model, (b) STL y triángulos, (c) partículas SPH.

En la Fig. 16 podemos ver los tres pasos mencionados anteriormente, el panel (a) corresponde al Digital Surface Model, esta superficie es la geometría generada por el software de fotogrametría, el color en este pido de imágenes se obtiene gracias a la información radiométrica de las imágenes tomadas por la cámara. El panel central es la superficie compuesta por triángulos en formato STL, y el último panel corresponde a los puntos SPH que conforma el contorno de la simulación.

Para poder generar las partículas SPH se emplea una de las herramientas de pre-procesado de *DualSPHysics.* Esta herramienta está diseñada para crea cualquier tipo de objeto usando partículas y así poder usarlo en *DualSPHysics*. El proceso el siguiente, se crea una malla 3D, con un espaciado igual al *dp,* para localizar donde deberían ir esas partículas; después se verifican los nodos que se encuentran cerca del objeto y las partículas sólo son creadas en los nodos que componen el objeto. Se puede encontrar información más detallada de este proceso y de todas las funcionalidades de la herramienta de pre-procesado en Domínguez et al. (2011).

Las geometrías mostradas en las Fig. 15 y Fig. 16 corresponde al área de estudio. Este punto se encuentra en las coordenadas 42.2946°N, 7.5888° W situado en la provincia de Ourense, y es parte de la carretera que une Ourense con Pobra de Trives. Los daños causados por las lluvias son visibles en la orografía del lugar. En la Fig. 17(a) podemos ver una vista de satélite de la zona, en gris se indica el área captación de lluvias que afectan a la carretera. El panel (b) de la Fig. 17 muestra el área captada por el dron en gris, y se encuentra resaltada el área simulada en *DualSPHysics*, esta área está aumentada en el panel (c) de la misma figura, aquí también se muestra la disposición de los ejes, la zona de entrada del agua y la ubicación de la zanja. El dominio de estudio mide 83 m x 75 m y tiene una pendiente media de entorno al 17% en la dirección X y aproximadamente del 7% en la dirección Y. La zanja creada para paliar los efectos de la escorrentía en la carretera cercana tiene 0.8 m de anchura y 0.5 m de profundidad

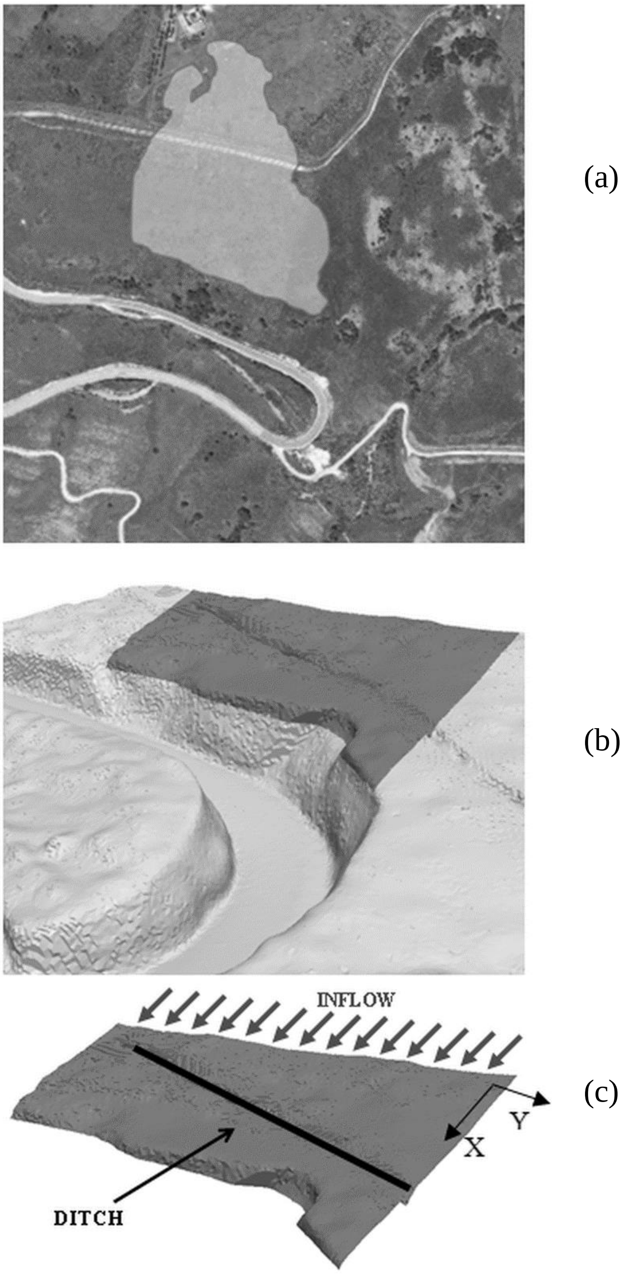


Figura 17. Dominio de la simulación SPH y localización de la zanja. El panel a muestra una visión aérea de la zona.

Las condiciones de entrada de agua han sido ajustadas para coincidir con los datos obtenidos de la Agencia Española de Meteorología (AEMET) considerando un periodo de retorno de 50 años.

Realizando los cálculos para un evento de 0.1 horas, se consigue una descarga de aproximadamente 0.5 m^3s^{-1} por lo tanto deberemos obtener ese mismo valor en *DualSPHysics*. Para conseguir este valor de descarga se han implementado condiciones especiales de entrada, nuevas partículas de fluido entran por la parta superior del dominio, como se puede ver en la Fig. 17 (c), el área por la que entra el agua se denota por flechas y la palabra INFLOW. La entrada de partículas debe ser 0.5 m^3s^{-1}, cada m^3 de fluido contiene 17,802 partículas para el *dp* de la simulación, por lo tanto cada segundo se deberán insertar 8,901 partículas.

En la Fig. 18 se puede observar la serie temporal del valor de descarga, el valor obtenido coincide con el valor calculado.

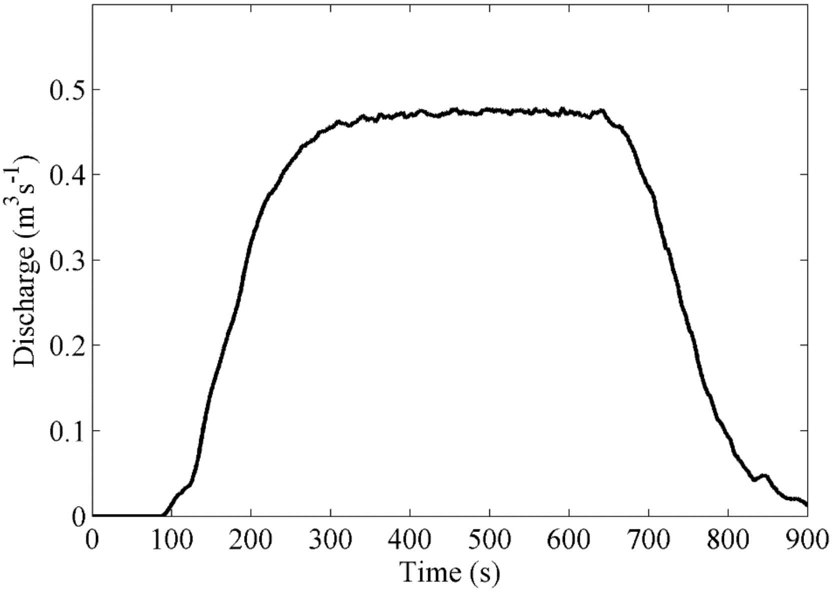


Figura 18. Entrada de agua por el contorno superior.

Debido a la naturaleza del terreno y a la cantidad de vegetación se ha tenido que ajustar el valor de la viscosidad de interacción entre partículas fluidas y partículas que pertenecen al contorno. Esto se debe a que el modelo está validado para experimentos de laboratorio donde las superficies por las que pasa el fluido suelen ser de cristal o cemento pulido. Nuestra área de estudio contiene una gran cantidad de arbustos por lo tanto se debe considerar un coeficiente de Manning igual a 0.075. El coeficiente de Manning es un coeficiente empírico usado en ingeniería el cual representa la facilidad o dificultad que tiene un flujo para descender por diferentes valores de pendiente y con diferente tipología de superficie. Para nuestro caso de estudio la velocidad que debería alcanzar el flujo es de 0.46 ms^{-1} en la Fig. 19 se puede ver como para diferentes valores de viscosidad entre partículas fluidas y de contorno se obtienen diferentes velocidades, el valor que coincide con 0.46 ms^{-1} es α_{FB}=16 α_{FF}.

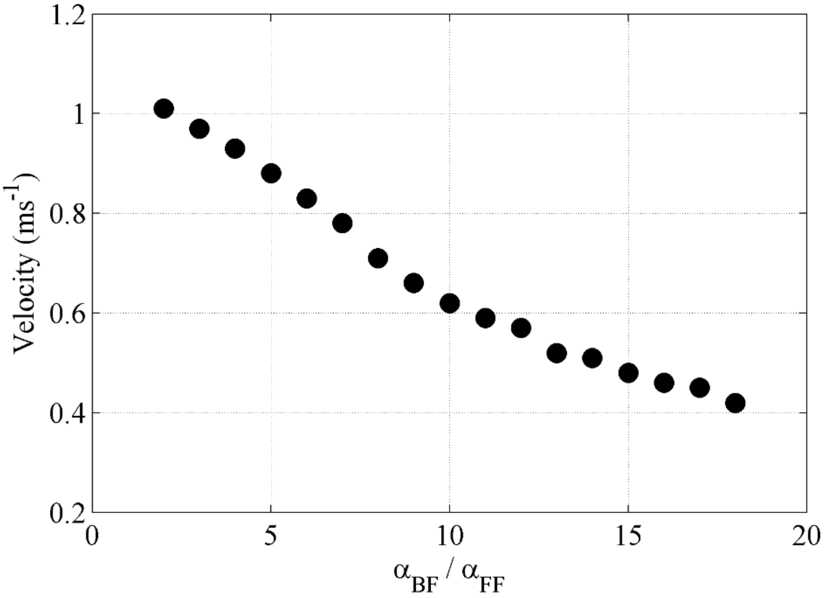


Figura 19. Velocidad de flujo dependiente de la relación entre α_{FB} y α_{FF}.

En este caso de aplicación se han realizado varias simulaciones, la primera de ellas sin zanja. En la Fig. 20 se puede observar como el agua entra en el dominio por la parte superior y va formando pequeños riachuelos que descargan el agua en la carretera. La Fig. 21 muestra la cantidad de agua que llega a la carretera y se puede ver que coincide con la Fig. 18 solo que desplazada unos segundos, este es el tiempo que tarda el fluido en viajar de la entrada hasta la carretera.

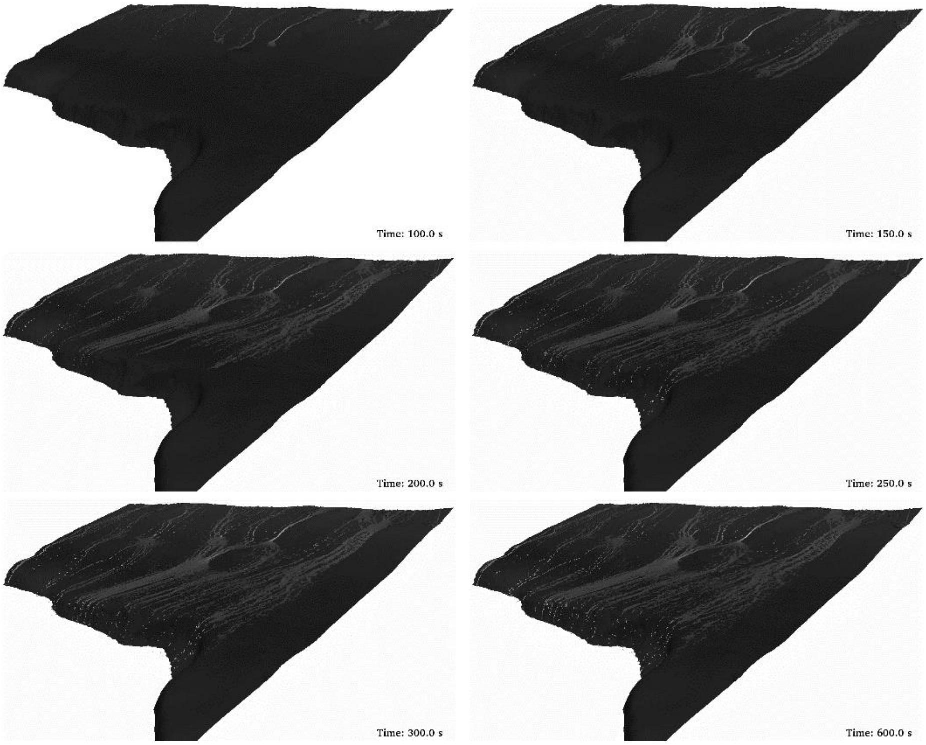


Figura 20 Diferentes instantes de la simulación sin zanja realizada con *DualSPHysics*.

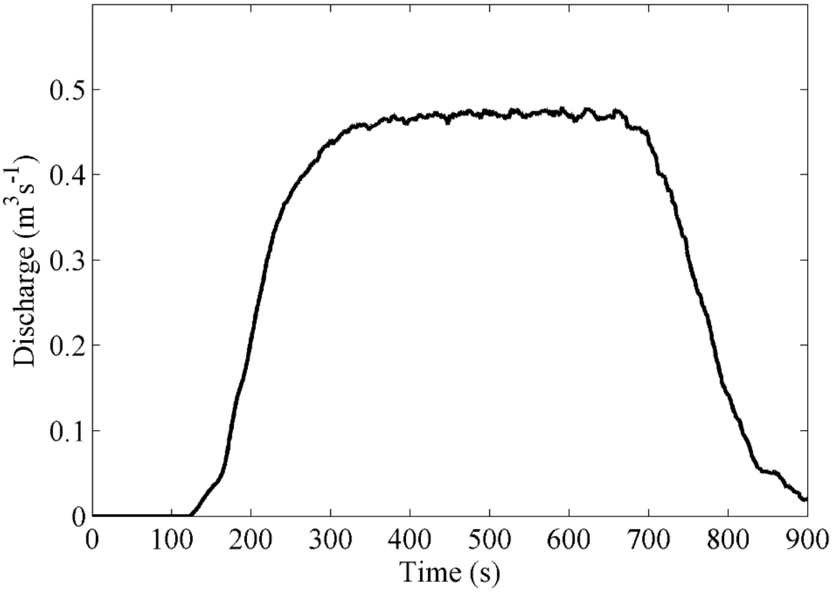


Figura 21. Flujo de agua a la carretera.

En la Fig. 22 se muestra los mismos parámetros de descarga, la única diferencia entre estas dos simulaciones es que en esta se ha añadido la zanja (80 cm de ancho y 50 cm de profundidad). En este caso podemos ver como toda el agua es desviada antes de alcanzar la carretera hacia un lugar donde los daños causados serán menores.

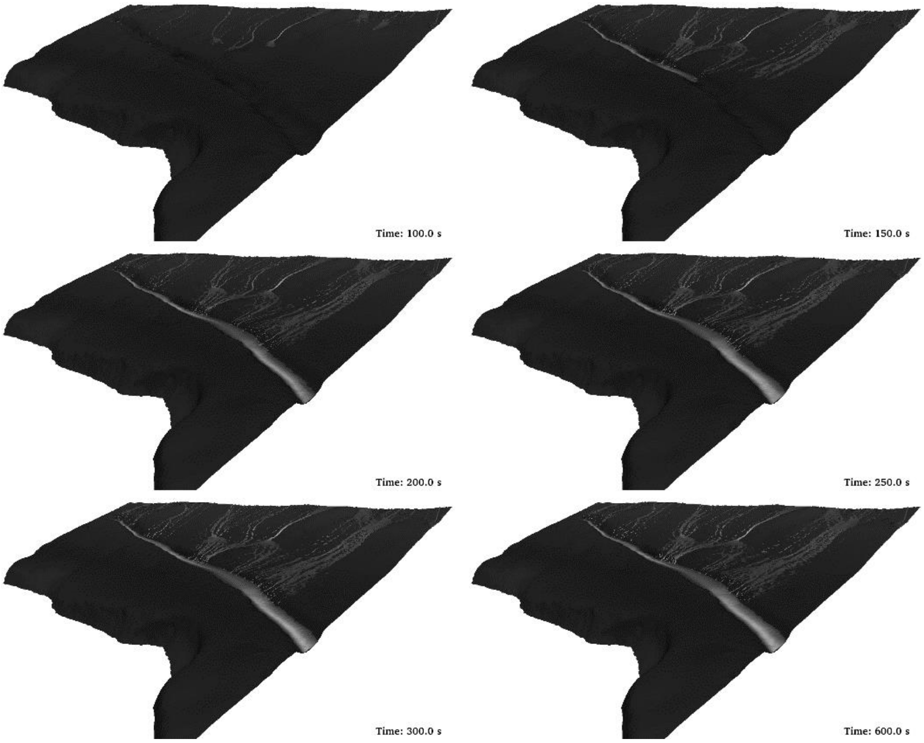


Figura 22. Diferentes instantes de la simulación con zanja de 0.8 m realizada con *DualSPHysics*.

El plan de las obras de protección era realizar una zanja que desviase el agua incluso en eventos extremos de escorrentía. El mayor problema de esta zanja es que no se trata de una construcción duradera, esta zanja puede verse obstruida o erosionada por el paso del tiempo. Teniendo en cuenta estos efectos se han realizado varias simulaciones para diferentes tamaños de zanja para así analizar como el deterioro de la zanja afectará a su capacidad para drenar el agua y proteger la carretera. En la Fig. 23 se muestran los valores de agua drenada por la zanja y agua que llega a la carretera para diferentes valores de profundidad de la zanja (de 0.0 a 0.8 m).

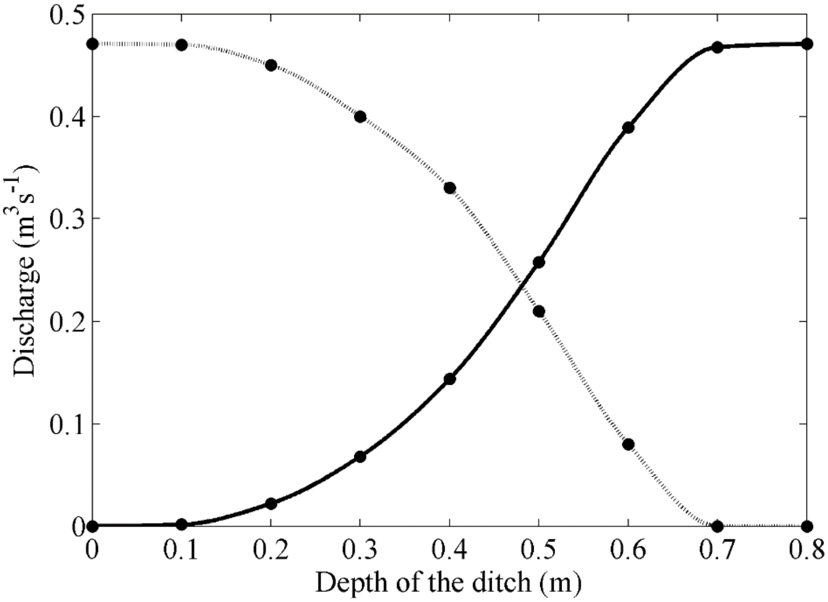


Figura 23. Descarga a la carretera (línea punteada) y drenado de la zanja (línea sólida) para diferentes profundidades de la zanja.

En la figura se puede observar que las zanjas con profundidades menores a 0.2 m son ineficientes, más del 90% del agua descargada llega a la carretera. La efectividad de las zanjas se incrementa para zanjas con profundidades mayores a 0.4 m donde el 50% del flujo es desalojado por la zanja. Finalmente, para zanjas con una profundidad superior a 0.7 m se consigue desalojar el 100% del agua a través de la zanja.

E) CONCLUSIONES

Se ha desarrollado una solución CPU-GPU llamada *DualSPHysics* para su aplicación a problemas de superficie libre. Este modelo ha sido desarrollado a partir del código SPHysics en FORTRAN, heredando las propiedades de estabilidad y precisión de su predecesor pero incrementado el rendimiento y las funcionalidades. El código *DualSPHysics* puede ser ejecutado tanto en CPU como en GPU dependiendo de la disponibilidad del hardware.

Gracias a esta implementación en GPU ahora se pueden realizar simulaciones con un gran número de partículas en ordenadores personales que dispongan de tarjetas capaces de ejecutar CUDA. *DualSPHysics* aprovecha el rendimiento y el espacio de memoria que proporciona esta nueva tecnología. Esto significa que investigaciones que antes requerían el uso de centros de cálculo (costosos y de caro mantenimiento), pueden ser llevadas a cabo con tecnología barata y de fácil acceso. Las aceleraciones conseguidas en el código permiten el estudio de problemas de ingeniería reales y complejos en un tiempo computacional razonable. Por ello se han presentado varios casos relacionados con problemas de ingeniería civil y de costas. También se han implementado nuevas funcionalidades para poder afrontar estos casos. Estas nuevas capacidades del código abren nuevos campos en los que se puede aplicar *DualSPHysics*.

Conclusiones para protección de costas

El incremento del nivel del mar aumentará el riesgo de inundación de tal manera que las medidas de protección costeras actuales deben reevaluadas. Los peligros de las inundaciones costeras son determinados por la interacción de las olas y los incrementos del nivel del mar debido a las tormentas con la batimetría y la topografía de la costa. En esta situación el fondo marino y orografía terrestre juegan un papel clave para determinar la extensión de una inundación costera. Los modelos de inundación costera no solo deben cubrir la extensión de la inundación si no también la capacidad destructiva de la misma que está directamente ligada a la fuerza ejercida por las olas en las estructuras costeras.

DualSPHysics ha demostrado ser un modelo preciso y capaz de reproducir adecuadamente la propagación de olas cerca de la costa y el impacto contra estructuras. Las capacidades del código *DualSPHysics* para reproducir la interacción ola-estructura han sido mostradas y se han computando de manera numérica las alturas de ola y las fuerzas ejercidas sobre los objetos de la costa. El caso de estudio imita de manera realista un paseo marítimo, en el que se encuentran diferentes elementos de mobiliario urbano con dimensiones y geometrías cercanas a las reales.

Concusiones para amarres y objetos flotantes

Se han implementado en *DualSPHysics* las funcionalidades necesarias para simular los amarres. Esta nueva implementación es capaz de resolver de manera correcta las fuerzas de las cadenas y los efectos de los amarres en los objetos flotantes.

Se ha presentado la formulación de gobierna los objetos flotantes en *DualSPHysics*. Se ha validado la flotabilidad de los objetos flotantes contra datos de VOF mostrando muy buena concordancia.

Se ha implementado todo un nuevo conjunto de ecuaciones para resolver de manera satisfactoria los amarres. El código cubre todos los estados posibles de un amarre, teniendo en cuenta parámetros como posición relativa entre el muerto y el punto de conexión en el cuerpo flotante, peso del amarre, forma de la ecuación de catenaria…

La implementación ha sido validada comparándola con datos experimentales y datos obtenidos de otros modelos numéricos. Esta validación ha mostrado una muy buena concordancia entre los datos numéricos y experimentales para varias escalas. También se ha mostrado la validación para un cuerpo con varios amarres proporcionando buenos resultados.

Se han presentado dos ejemplos, el primero de ellos un barco amarrado con una sola línea expuesto a olas de costado. El segundo ejemplo de aplicación consiste en la base de un aerogenerador amarrado al fondo marino con tres líneas y expuesto a grandes olas.

Por todo esto, se puede concluir que la implementación ha sido validad para problemas de uno y varios amarres ofreciendo buenos resultados. Esta nueva funcionalidad aumenta el rango de aplicaciones que *DualSPHysics* puede resolver de manera satisfactoria.

Conclusiones para escorrentía en terrenos reales

El estudio presenta un modelo capaz de resolver problemas de escorrentía en geometrías complejas con aplicación directa al campo de ingeniería civil. El estudio une dos tecnologías avanzadas, fotogrametría UAV y un código de dinámica de fluidos basado en SPH.

La fotogrametría UAV se usó para obtener la geometría del área de estudio. La nube de puntos obtenida después de la restitución fotogramétrica se transforma puntos a través de un proceso de triangulación, estos puntos son usados para crear la condición inicial para el método SPH. Se usó *DualSPHysics* para computar las trayectorias del fluido y su interacción con la geometría.

Se simularon condiciones de lluvia intensa usando una entrada de agua que imita condiciones meteorológicas extremas en la zona de estudio para un periodo de retorno de 50 años. El flujo de entrada de 0.5 m^3s^{-1} fue impuesto a través de la frontera superior del sistema.

Se analizó el efecto de una zanja (0.8 m de profundidad por 0.5 m de ancho) a la hora de prevenir el paso de agua a la carretera. Esta medida de precaución se mostró suficiente para drenar toda el agua durante un episodio extremo. También se realizó un estudio del efecto de la sedimentación en la zanja y de cómo la reducción de la profundidad afectaría a la capacidad de drenaje de la zanja perdiendo un 50% de su capacidad cuando la profundidad se reduce a 0.45 m.

Resumiendo, la combinación de las medidas tomadas con UAV y las simulaciones de *DualSPHysics* ha probado ser una herramienta capaz de describir las inundaciones en áreas cercanas a carreteras y para diseñar medidas que mitiguen ese tipo de inundaciones. La herramienta resultante puede ser aplicada a múltiples problemas de ingeniería civil y proporcionar más información que la típica solución de GIS que es incapaz de representar el comportamiento real del fluido.

Trabajo futuro

Como ya se ha comentado anteriormente, el modelo presenta un elevado rendimiento, que combinado con las nuevas funcionalidades implementadas, nos permite simular un gran número casos reales de ingeniería. Sin embargo, todavía existen varios aspectos en los que es necesario trabajar para aumentar la precisión de los resultados y poder abordar casos nuevos de mayor complejidad. Algunas de estas líneas de investigación serían

- Deformación de los contornos y los cuerpos flotantes: *DualSPHysics* cubre perfectamente la interacción fluido-contorno, pero en algunos casos extremos las estructuras costeras y los objetos flotantes pueden verse deformados, incluso rotos, debido a la acción de las olas. El reto es esta área es la implementación adecuada de las propiedades de los materiales y la capacidad de representar los esfuerzos internos. De este modo la implementación puede ser incluso más completa a la hora de diseñar estructuras costeras u objetos flotantes y puede proporcionar a los ingenieros una herramienta muy completa para facilitar su trabajo de diseño.

- Implementación de amarres complejos: en ciertos casos se pueden necesitar amarres más complejos para mantener las estructuras costeras en un sitio. Estos amarres más complejos tiene una boya en mitad de la cadena, este flotador ayuda a mitigar una parte de la fuerza a la que se somete un amarre funcionando como una especie de muelle.

- Dinámica de los amarres e interacción con el arrastre: los amarres implementados en *DualSPHysics* no interaccionan con las partículas fluidas y el código ignora la inercia de la cadena. En una descripción más completa se deben cubrir las dos situaciones. Diferentes estudios han probado que las fuerzas debidas a las corrientes arrastrando las cadenas, la elasticidad y vibración de un amarre no son despreciables.

- Erosión de sedimentos e infiltración: el estudio de escorrentía está centrado en un evento extremo, como una tormenta, pero otros efectos pueden ser causados debido a eventos de largo plazo. Fenómenos de mayor duración pude causar el movimiento de sedimentos o la penetración de agua en capas profundas del terreno. Estos dos casos pueden deformar el terreno y crear nuevas inestabilidades causando caídas de rocas y corrimientos de tierra. Para resolver estos problemas de manera adecuada deben añadir nuevas propiedades a las partículas de contorno.

Abstract

Smoothed Particle Hydrodynamics (SPH) is a Lagrangian meshless method applied successfully in the field of Computational Fluid Dynamics (CFD). Meshless methods have proven to give good results in problems involving free-surface flows. This technique can deal with complex geometries and flow discontinuities, but the serial implementation of most SPH codes makes almost impossible to face real-engineering problems. During the past years SPH codes have evolved using new powerful hardware like GPUs (Graphic Processing Units). *DualSPHysics* was developed to take advantage of GPU computing without losing the accuracy and reliability of previous SPH code, SPHysics, implemented in FORTRAN. This step forward opens a whole new range of applications but also presents a lot of new challenges.

The aim of this work is to show the research carried out to adapt *DualSPHysics* to solve new complex engineering applications. The goal of this manuscript is to present the implementations carried out in the code and how *DualSPHysics* is a tool that can facilitate the design-job in various fields.

Three different applications in the fields of coastal and civil engineering are presented in this work. The first one analyses a case of coastal protection. Climate change is forcing to re-evaluate old coastal defences and to improve the design of new ones. *DualSPHysics* is used to study wave propagation and wave-structure interaction. The code is validated with analytical and experimental data showing reliability, accuracy and efficiency. This validation proves that the method is suitable to reproduce free-surface phenomena such as breaking waves, and fluid–structure interaction. The capabilities of *DualSPHysics* to reproduce wave–structure interaction are shown where wave heights and forces exerted onto objects of the coast are numerically computed. The case of study mimics a realistic promenade including the urban furniture with dimensions and geometries close to the real ones.

The second study presents the implementation of moorings for floating structures. Each year more energy devices are placed in the sea to take advantage of unexploited resources like oil and gas platforms, wind farms, WECs (Wave Energy Converters)… Most of these off-shore structures are moored to the sea bed because of the wave conditions. A new set of equations is implemented in *DualSPHysics* to cover the behaviour of moored floating bodies. Validation is provided for both floating bodies and moored lines showing good agreement with experiments and numerical data. Two different working cases are presented to show the capabilities of the code.

The last application is related with civil engineering. The research focuses on how roads can experience runoff problems due to the intense rain discharge associated to severe storms. Two advanced tools are combined to analyse runoff phenomena in real terrains. UAV (Unmanned Aerial Vehicle) photogrammetry is used to obtain the geometry of the case and *DualSPHysics* model is applied to compute the trajectory of the water interacting with the complex geometry. The effectiveness of protective measures to palliate flood effects is also analysed. The amount of water arriving to the road is measured for different scenarios. The performance of the protective measure, a ditch, is observed to decrease when its depth is reduced.

Table of contents

List of figures

List of tables

1. Introduction

A general view of numerical models, and Smoothed Particle Hydrodynamics (SPH) in particular, is shown in this chapter. The advantages and disadvantages of the SPH method are described. Finally the *DualSPHysics* project is introduced.

1.1 Numerical modelling

Numerical modelling is used as a very useful tool in the engineering and science fields to solve complex problems. The main advantage of using numerical methods is the capability to simulate any scenario regardless its complexity. This avoids the need to build very expensive physical scaled models. Also, numerical modelling may provide physical data that could be very difficult, or even impossible, to measure in a real or scaled model. For example, the modelling of a Pelton turbine can be performed with a physical model (left snapshot in Fig. 1.1) or using the numerical model (right in Fig 1.1). In this example, the numerical simulations can be used as complementary tool providing optimizations for the physical design of the turbine. This results in savings both in time and costs, which is a key factor in industrial applications.

Figure 1.1. Real physical model (left) and numerical model.

The physical equations that govern a numerical model can be approached in two different ways; the Eulerian description or Lagrangian description. Also the domain of a simulation can be discretised with the help of a mesh or use particles to track the fluid properties. Mesh based methods (finite elements, finite differences, and finite volume) are really robust and had been used in a wide range of applications. These methods have proven to provide really accurate results and are ideal for systems with perfectly defined domains, and with static boundaries. On the other hand, the creation of the mesh sometimes could result very inefficient if the system is quite complex.

This work is focused in meshless methods. These methods are mainly used in solid mechanics and fluid mechanics. Meshless methods can easily simulate problems with very complex geometries, great deformations, flow discontinuities or material singularities. Imagine, for instance, the arduous task of creating a mesh that defines the geometry of the turbine represented on Fig. 1.1, and also needs to rotate, or the complexity of the fine layers of fluid formed after the flow impacts with the buckets (Fig. 1.1 right side).

1.2 Numerical modelling for engineering problems

Climate change is modifying the sea-level and weather drastically. During the past years, the intensity and frequency of severe storm events have increased. Old structures are not prepared to the devastating effects of storms that will lead to sea-level rise. These extreme events will push to test civil and coastal structures during the next 20-100. For example, in 2013 and 2014 winters, great storms hit the northern Europe with high winds and flooding as a result of raised sea levels, and also, the Atlantic and Cantabrian coasts of the Iberian Peninsula suffer the periodic passage of storms which give rise to severe sea states and important damage in the coast (e.g., the storm Nadja on 2nd February 2014).

Within the next century, there is a great concern on the risk of potential disasters involving multiple factors; events such as severe storms being superimposed onto sea-level rise for estuarine/river locations whose defences were designed with outdated guidelines. The design of coastal defences and civil structures are continually evolving to respond to changing needs and improved understanding of their behaviour in field conditions. To address the future requirements, many coastal and civil structures currently deployed around the world could need either replacing or retrofitting. The extent of future improvements is not currently known however, in part due to uncertainties in effects such as sea level rise and extreme storm intensities.

In the practice of coastal and civil engineering, different (simplistic) formulations are still commonly used today to estimate engineering quantities such as overtopping, run-up, run-off or other phenomena. These formulations are empirical as they are drawn from specific cases and geometries. Moreover, it is difficult to extend the use of the formulae to other scenarios different from the general starting cases or dataset from which they were derived. It is in these circumstances that, the option of using a numerical tool becomes much wholly appropriate. Nevertheless, physical and numerical modelling must be complementary since the problems under study become more and more complex to be studied in detail for a proper structural design.

Traditional computational fluid dynamics (CFD) techniques such as volume-of-fluid methods (VOF) have been used to study wave-structure interactions (Kleefsman et al., 2005) and to design breakwaters (Higuera et al., 2013; Vanneste & Troch, 2012). However, Eulerian numerical methods, such as those based on the finite volume technique, require expensive mesh generation and have severe technical challenges associated with implementing conservative multi-phase schemes that can capture the nonlinearities within rapidly changing geometries. Thus, the emergence of meshless schemes has provided a much needed alternative, and meshfree methods, such as Monte Carlo methods (Geeraerts et al., 2009) or the particle finite element method (PFEM) (Oñate et al., 2011), are becoming popular.

Actually, numerical modelling will not replace physical modelling but will make scale models fewer and more close to the final design, this will save money and time for companies and governs. The importance of models like *DualSPHysics* is the ability to perform simulations with outstanding performance. *DualSPHysics* exhibits a high precision at reasonable computational time using proper resolution.

First studies have been already conducted using SPH to study the wave run-up and overtopping in existing armour block sea breakwaters and the forces exerted by large waves onto the coastal structures comparing the numerical results with analytical solutions, with experimental data and with field data and the study of a slope and the run-off affecting a road.

1.3 SPH Method

Within the wide group of meshless methods, we can find the Smoothed Particle Hydrodynamics (SPH) model, SPH is a Lagrangian method, developed on the 70's decade to solve astrophysics problems (Gingold & Monaghan, 1977; Lucy, 1977) and in the last years it has been successfully applied to a wide range of fluid dynamics problems.

SPH discretizes the fluid as a set of points or nodes called particles, in Fig. 1.2 the top panel shows the fluid as a continuum and the bottom panel presents the discretization in SPH particles. In SPH, the continuous equation of movement is solved using the Lagrangian formalism, calculating the value of the most representative physical variables (velocity, position, density and pressure) for each particle as the interpolation of the values of the closest neighbouring particles. The *kernel* function is used to perform the transition from a continuous domain to discrete particles.

This function has a compact support inside the domain. This domain is defined by the so called smoothing length that indicates the distance of interaction between particles.

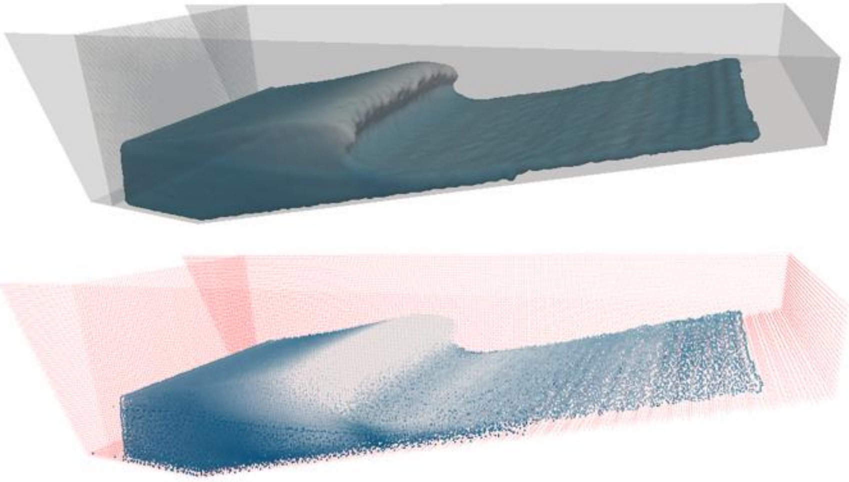


Figure 1.2.Continous fluid, top pannel, and SPH particles, bottom panel.

1.4 SPH advantages and disadvantages

The advantages and limitations of the SPH method are presented here:

Advantages

- The main advantage is given by its Lagrangian nature, this allows following and simulating highly complex flows.
- Only where particles exist, fluid or boundary, are represented. Therefore, the calculus time and the requested memory are employed in a more efficient way, avoiding the storage of data from empty places.
- There is no restriction to the system geometry, or where it could evolve from the initial conditions, the initial set up is easily programmable, without the need of complicated algorithms, like is the case of finite element method.
- The treatment of the mobile boundaries and the inclusion of different physical processes is very simple.

Disadvantages

- The implementation of the boundary conditions is a hard task, and the penetration of fluid particles through the defined boundaries has to be avoided. Boundary conditions are recognized as a topic of special interest by the SPH community. Many researchers are currently working to improve the existing ones or to develop new solutions.
- The interpolant method used on SPH is really simple, therefore it will be very affected by the particle disorder. SPH gives reasonable results to first order gradient (Bonet & Lok, 1999) recommended corrections to the gradient), but may be worse for higher order derivatives.
- The method is computational slower compared to the modern mesh methods. The reasons are that the time step is much lower, and the

number of interactions is much larger than using mesh based methods. However recent progress has been made to solve these limitations, exploring new fields, and resources provided by a whole new range of hardware.

1.5 Applications

There is a wide variety of phenomena in which the SPH technique generates satisfactory results. At the beginning of its development the application was oriented to the star rotation dynamics (Lucy, 1977). Since then, the method has been applied to diverse range of problems and has shown its reliability in different fields:

Great Waves

- Study of how submarine landslides produce great waves (Panizzo & Dalrymple, 2004; Panizzo et al., 2006; Rogers & Dalrymple, 2007).
- Great waves generated by underwater landslides (Capone et al., 2010).
- Use of non-linear swallow water equations to solve great waves (Del Guzzo & Panizzo, 2007; Vacondio et al., 2011).

Transport and suspension of sediments

- Sediment suspension under the effect of waves using the lagrangian form of the diffusion equation (Zou & Dalrymple, 2006; Zou, 2007).
- SPH Modelling of water/soil-flows using a variable resolution scheme (Ulrich et al., 2011)
- SPH Simulation of Sediment Flushing Induced by a Rapid Water Flow (Manenti et al., 2012)
- Modelling sediment resuspension in industrial tanks using SPH (Fourtakas et al., 2013).

- Modelling multi-phase flows in Nuclear Decommissioning using SPH (Fourtakas, 2014)

Swallow water

- 2D approximation of fast variation swallow water (Vacondio et al., 2010),
- Swallow water coastal flows (De Leffe et al., 2010).

Multiphase SPH

- Two phase flows; water an air (Cuomo et al., 2006).
- Interphase force calculation (Hu & Adams, 2007).
- SPH modelling of rapid multiphase flows and shockwave propagation (Manenti et al., 2011)
- GPU Acceleration of 3-D Multi-phase SPH Simulations for Violent Hydrodynamics (Mokos et al., 2013)
- Multi-phase modelling of violent hydrodynamics using Smoothed Particle Hydrodynamics (SPH) on Graphics Processing Units (GPUs) (Mokos, 2014)

Incompressible SPH

- Maximum height flooding measurements (Shao et al., 2006).
- Eliptic Poisson's equation solution for pressure (Lo & Shao, 2002; Ellero et al., 2007).
- Incompressible and lightly compressible comparison (Hughes & Graham, 2010).

Interaction between waves and coastal structures

- Dam-break impact with a structure (Gómez-Gesteira & Dalrymple, 2004).
- Interaction between a floating wall linked to the bottom (Shao & Gotoh, 2004).
- Analysis of the overtoping phenomena on horizontal decks (Gómez-Gesteira et al., 2005).

- Study of the turbulence under the breaking wave zone (Dalrymple & Rogers, 2006).
- Protection barriers to mitigate the force and momentum exerted by big waves (Crespo et al., 2007a).
- (Janosi et al., 2004) experimental data comparison with numerical results (Crespo et al., 2008).
- Smoothed Particle Hydrodynamics for coastal engineering problems (Barreiro et al., 2013).
- Numerical modelling of armour block sea breakwater with Smoothed Particle Hydrodynamics (Altomare et al., 2014).

Industrial applications

- SPH model to study a real spillway that connects the reservoir of a river dam to a valley (Lee et al., 2010).
- Hydraulic structures simulations: Hydraulic jumps (Lopez et al., 2010).
- Simulation of free-surface flows encountered in Pelton turbines (Marongiu et al., 2010).

1.6 DualSPHysics project

SPHysics is a project developed by a group of researchers from Johns Hopkins University (U.S.A), University of Vigo (Spain), and Manchester University (U.K). This group has focused its research in wave propagation and its interaction with coastal structures, in 2D (Gómez-Gesteira et al., 2005; Dalrymple & Rogers, 2006; Crespo et al., 2008) and 3D (Gómez-Gesteira & Dalrymple, 2004; Crespo et al., 2007a).

The SPH code called *SPHysics* was written in FORTRAN90, is an open-source code and it can be freely downloaded from the site www.sphysics.org. One of the purposes of this project is to foment the use of the SPH method between the researchers, and encourage them to contribute to the project sending their improvements.

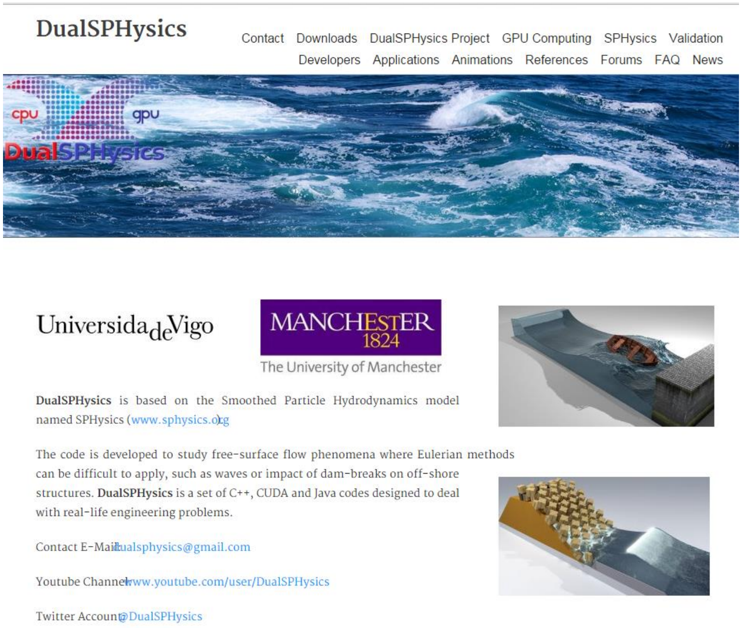
Regardless that *SPHysics* code gives an accurate description of the fluid, its major drawback is the long execution times due to the high computational cost of SPH and the serial implementation of *SPHysics*, which makes impossible its use in the study of actual engineering problems. The computational time increases with the number of particles that form the system, which is the main reason why a lot of problems cannot be faced. For this reason, High Performance Computing (HPC) is the necessary tool to achieve new goals in the modelling of engineering problems. More specifically the GPUs (Graphics Processing Units) are used as a computing device. The GPUs appear as an accessible alternative to accelerate the SPH methods employing a powerful parallel programing model. This is a new technology developed by the video game industry. Left panel in Fig. 1.3 shows a screenshot form a video game, the image is composed of millions of triangles rendered in real time. These devices provide a high performance at a reasonable cost (Fig. 1.3 right panel shows one of the latest graphics cards, the Nvidia GTX Titan X). Therefore, the GPUs make possible the study of real engineering problems using SPH methods.

Figure 1.3. An image from Elder Scrolls: Skyrim, Bethestha Studios, and the latest graphic card the Nvidia GTX Titan X.

A new code was developed starting from the former Fortran *SPHysics* model and implemented using both the C++ and CUDA programming languages. The code can then be executed either on the

CPU or on the GPU since all computations have been implemented both platforms. Most of the source code is common to CPU and GPU which makes debugging straightforward as well as the code maintenance and new extensions. This allows the code to be run on workstations without a CUDA-enabled GPU, using only the CPU implementation. On the other hand, the resulting codes should be necessarily different since code developers have considered efficient approaches for every processing unit. Thus, comparisons between the performances of both approaches are more reliable since appropriate optimisations have been considered for every architecture (CPU or GPU).

The executable package of *DualSPHysics* can be freely downloaded from the web www.dual.sphysics.org (Fig.1.4). This package includes pre-processing tools that allows creating any type of complex geometry, loading actual topographies or bathymetries and importing geometries directly from files such as .cad .3ds .max .stl .dwg etc. Post-processing tools are also included to help to visualise data or to measure interesting physical magnitudes such as vorticity, velocity, exerted forces...



DualSPHysics is based on the Smoothed Particle Hydrodynamics model named SPHysics (www.sphysics.org)

The code is developed to study free-surface flow phenomena where Eulerian methods can be difficult to apply, such as waves or impact of dam-breaks on off-shore structures. **DualSPHysics** is a set of C++, CUDA and Java codes designed to deal with real-life engineering problems.

Contact E-Mail: dualsphysics@gmail.com

Youtube Channel: www.youtube.com/user/DualSPHysics

Twitter Account: @DualSPHysics

Figure 1.4. *DualSPHysics* web (www.dual.sphysics.org).

1.7 Work outline

This manuscript is a summary of the research carried to improve *DualSPHysics* and its applicability to civil and coastal engineering cases. The thesis is organized in a total of seven chapters that are briefed as follows:

Chapter 1, introduces background knowledge of numerical simulation, meshless methods and SPH. Some general features, advantages and applications are described. The DualSPHysics code associated with this thesis is introduced.

Chapter 2 presents the mathematical formulation of SPH and a description of the basic concepts, such as, integral interpolants theory, smoothing kernels, and the momentum and mass conservation in terms of discrete notation. SPH theory chapter is based on the paper [Crespo et al., 2015] whose main responsible is A.J.C. Crespo and A. Barreiro is one of the co-authors.

Chapter 3 provides the description of the different steps of an SPH simulation, and how these steps are implemented in *DualSPHysics*. Also a performance test comparing the different version of the code (CPU vs GPU) is provided. SPH implementation is based on the paper [Crespo et al., 2015] and the candidate is one of the co-authors.

Chapter 4 describes an application in the field of coastal protection where *DualSPHysics* was used. Validation and a working case is used to simulate the interaction of large waves with urban furniture in the coast. This chapter is based on the published paper [Barreiro et al., 2013] whose main responsible is A. Barreiro.

Chapter 5 deals with the implementation of mooring lines to restrict the movement of floating objects under the action of waves. The equations to be solved are presented, and validation of the model is provided with experimental and numerical data. Two applications are also shown.

Chapter 6 presents the integration of two different technologies, UAV-photogrammetry, and *DualSPHysics*. In this section the processes are explained and an example of how these two technologies can work together. This chapter is based on the published paper [Barreiro et al., 2014] whose main responsible is A. Barreiro.

Chapter 7 draws together conclusions and ongoing research.

2. SPH method

SPH theory is presented in this section. All the formulation and mathematical background will be explained here. The main features of SPH methodology are discussed in detail in the publications by Monaghan (1982; 1992; 2005), Benz (1990), Liu & Liu (2003), Liu (2003), and Gómez-Gesteira et al. (2010; 2012a).

2.1 Interpolants

SPH is based on the interpolant theory. This principle states that any given function $A(\mathbf{r})$ can be approximated by:

$$A(\mathbf{r}) = \int_{\Omega} A(\mathbf{r}')W(\mathbf{r}-\mathbf{r}',h)d\mathbf{r}' \tag{2.1}$$

where **r** is the position vector; W the weighting function or *kernel*; h is the interaction distance called smoothing length which controls the domain **Ω** (see Fig. 2.1). The value of h should be higher than the initial particle separation. The parameter h controls the size of the area around particle a where the contribution of the rest of particles cannot be neglected.

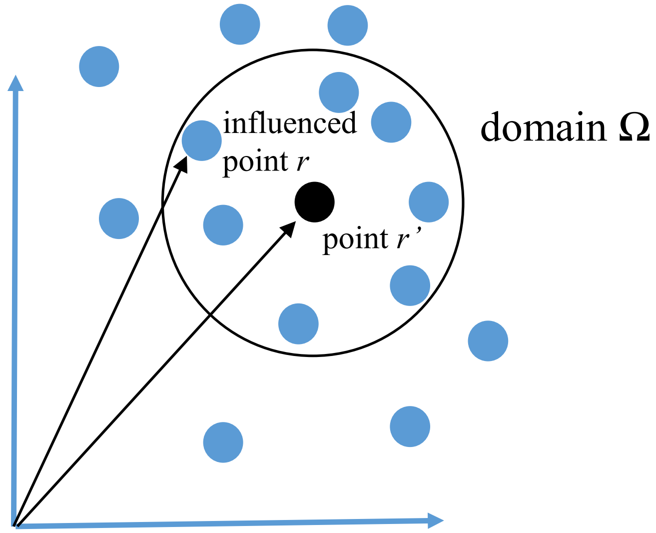


Figure 2.1. Influence domain scheme.

The Eq. 2.1, in discrete notation, leads to the following approximation function at a particle a:

$$A_a = \sum_b m_b \frac{A_b}{\rho_b} W_{ab} \tag{2.2}$$

where the summation is carried out over all the neighbour particles, *b*, that can be found in the region where the *kernel* is not zero-valued. The mass and density of the neighbour particles are denoted as m_b and ρ_b respectively, and $W_{ab}=W(\mathrm{r}_a\text{-}\mathrm{r}_b,h)$ is the weighting function or *kernel*.

One advantage of using the SPH *kernel* is that the function derivative is calculated analytically, and the derivatives can be obtained by ordinary differentiation:

$$\boldsymbol{\nabla} A_a(\mathbf{r}) = \sum_b m_b \frac{A_b}{\rho_b} \boldsymbol{\nabla} W_{ab} \tag{2.3}$$

This equation is derived by performing the integral in Eq. 2.1 for a functional derivative, and applying integration by parts.

2.2 Kernel function

The performance of SPH technique depends on the election of the weighting functions. These functions have to satisfy different conditions such as positivity, compact support, and normalization. Also, W_{ab} must be monotonically decreasing with increasing distance from the particle *a* and behaves like a delta function as *h*, the smoothing length, tends to *zero*:

$$W(\mathbf{r}_a - \mathbf{r}_b, h) \geq 0 \qquad \text{over the domain } \boldsymbol{\Omega} \tag{2.4}$$

$$W(\mathbf{r}_a - \mathbf{r}_b, h) = 0 \qquad \text{outside the domain } \boldsymbol{\Omega} \tag{2.5}$$

$$\int_{\Omega} W(\mathbf{r}_a - \mathbf{r}_b, h) d\mathbf{r}_b = 1 \qquad \text{(partition of unity)} \tag{2.6}$$

$$\lim_{h\to 0} W(\mathbf{r}_a - \mathbf{r}_b, h) d\mathbf{r}_b = \delta(\mathbf{r}_a - \mathbf{r}_b) \qquad \text{being } \delta \text{ a delta function} \tag{2.7}$$

$W(q,h)$ decreases monotonically with distance $q = \|\mathbf{r}_a - \mathbf{r}_b\|/h$ (2.8)

The *kernel* function depends on the smoothing length, h, and the nondimensional distance between particles given by $q = r_{ab}/h$, where r_{ab} is the distance between particles a and b, ($r_{ab} = r_a - r_b$).

There is a wide variety of *kernel* functions. In general, the accuracy of the SPH interpolation increases with the order of the polynomials used to define the *kernel*.

The cubic spline (Monaghan & Lattanzio, 1985) *kernel* (Fig. 2.2) has been, so far, the most widely used smoothing function in the SPH literature since it resembles a Gaussian function while having a narrower compact support:

$$W(\mathrm{r}, h) = \propto_D \exp(-q^2) \qquad (2.9)$$

where $\propto_D$ is $1/(\pi h^2)$ in 2D and $1/(\pi^{3/2} h^3)$ in 3D.

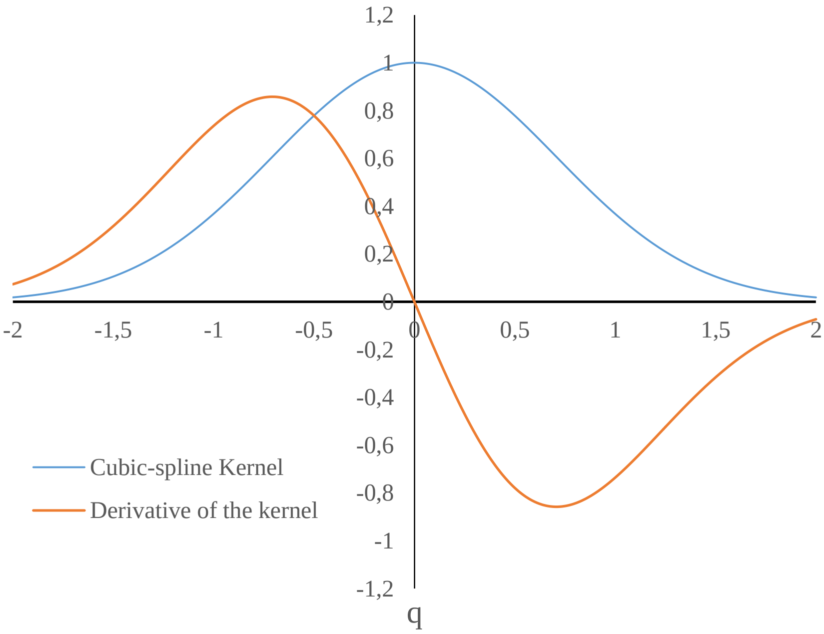


Figure 2.2. Cubic spline kernel and its derivative divided by $\propto_D$.

The Wendland (Wendland, 1995) quintic *kernel* (Fig. 2.3) is a good option because it provides a higher interpolation order.

$$W(\mathrm{r},h) = \propto_D \left(1-\frac{q}{2}\right)^4 (2\mathrm{q}+1) \qquad 0 \leq q \leq 2 \tag{2.10}$$

where $\propto_D$ is $7/(4\pi h^2)$ in 2D and $21/(16\pi h^3)$ in 3D.

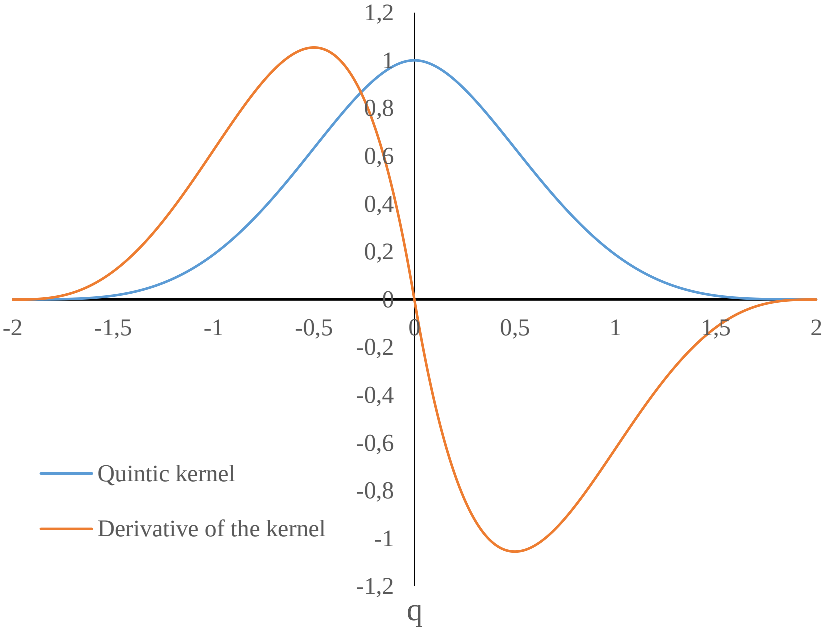


Figure 2.3. Wendland *kernel* and its derivative divided by $\propto_D$.

2.3 Momentum equation

The equation of momentum conservation in a continuous field is expressed as:

$$\frac{D\mathbf{v}_a}{Dt} = -\frac{1}{\rho}\boldsymbol{\nabla}P + \mathbf{g} + \boldsymbol{\Theta} \tag{2.11}$$

where $\mathbf{v}$ is the velocity and P and ρ are pressure and density respectively, $\mathbf{g} = (0,0,-9.81) ms^{-2}$ is the gravitational acceleration and $\boldsymbol{\Theta}$ represents the diffusion terms.

The gradient term of pressure in symmetrical form is expressed in SPH notation as:

$$-\frac{1}{\rho}\boldsymbol{\nabla}P = -\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2}\right)\boldsymbol{\nabla}_a W_{ab} \tag{2.12}$$

where P_b and ρ_b are pressure and density of neighbour particles b.

Therefore, the equation of the momentum conservation will be written in SPH notation as (Monaghan, 1992):

$$\frac{d\mathbf{v}_{\boldsymbol{a}}}{dt} = -\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2}\right)\boldsymbol{\nabla}_a W_{ab} + \mathbf{g} \tag{2.13}$$

Different approximations, based on existing formulation of diffusive terms, may be considered in the SPH method to rewrite the momentum equation.

2.4 Viscosity treatment

Two types of viscosity formulation are coded in *DualSPHysics*; (i) Artificial and (ii) Laminar + Sub-Particle Scale turbulence model.

2.4.1 Artificial viscosity

The artificial viscosity proposed by (Monaghan, 1992) has been widely applied due to its simplicity. In SPH notation, the momentum equation, including the artificial viscosity, is expressed as follows:

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b\left(\frac{P_a}{\rho_a^2}+\frac{P_b}{\rho_b^2}+\Pi_{ab}\right)\boldsymbol{\nabla}_a W_{ab}+\mathbf{g} \tag{2.14}$$

where Π_{ab} is the viscous term:

$$\Pi_{ab} = \begin{cases} \dfrac{-\alpha\overline{c_{ab}}\mu_{ab}}{\overline{\rho_{ab}}}, & si\ \mathbf{v}_{ab}\cdot\mathbf{r}_{ab}<0 \\ 0, & otherwise \end{cases} \tag{2.15}$$

with

$$\mu_{ab} = \frac{h\cdot\mathbf{v}_{ab}\cdot\mathbf{r}_{ab}}{\mathbf{r}_{ab}^2+\eta^2} \tag{2.16}$$

where $\overline{\rho_{ab}} = \frac{1}{2}(\rho_a+\rho_b)$, $\overline{c_{ab}} = \frac{1}{2}(c_a+c_b)$; $\eta^2 = 0.01h^2$; α is a free parameter that can be changed for each different problem according to the different configuration of each case.

For free-surface flows, Monaghan & Kajtar (2009) noted that the parameter α in the artificial viscosity model can be related to the Reynolds number in the following manner $Re = \frac{\sqrt{gD}D}{\upsilon}$ where D is the characteristic water depth and $\upsilon = \frac{1}{8}\alpha h\overline{c_{ab}}$ for the Wendland *kernel*.

2.4.2 Laminar viscosity and Sub-Particle Scale (SPS) turbulence

The momentum conservation equation with laminar viscous stresses is given by:

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho}\boldsymbol{\nabla}P+\mathbf{g}+\upsilon_0\nabla^2\mathbf{v} \tag{2.17}$$

The laminar stress term is simplified (Morris et al., 1997; Lo & Shao, 2002) to:

$$(\nu_0 \nabla^2 \mathbf{v})_a = \sum_b m_b \left(\frac{4\nu_0 \mathbf{r}_{ab} \boldsymbol{\nabla}_a W_{ab}}{(\rho_a + \rho_b)|\mathbf{r}_{ab}|^2} \right) \mathbf{v}_{ab} \qquad (2.18)$$

where ν_0 is the kinetic viscosity term of the fluid in a laminar flow ($0.893 \cdot 10^{-6}$ m^2/s in the case of water at 20 ºC).

Using SPH notation, the equation of momentum conservation using the laminar viscosity can be rewritten as:

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b \left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2} \right) \boldsymbol{\nabla}_a W_{ab} + \mathbf{g} + \sum_b m_b \left(\frac{4\nu_0 \mathbf{r}_{ab} \boldsymbol{\nabla}_a W_{ab}}{(\rho_a + \rho_b)|\mathbf{r}_{ab}|^2} \right) \mathbf{v}_{ab} \qquad (2.19)$$

Rogers & Dalrymple (2004) and Dalrymple & Rogers (2006) also used a large eddy simulation (LES) model (Christensen, 2006). This sub-grid scaling method uses a Smagorinsky (Smagorinsky, 1963) eddy viscosity term.

Gotoh et al. (2001) also used sub-particle scaling with the Moving Semi-implicit (MPS) technique. However, for a compressible fluid, the procedure is slightly different and involves Favre averaging, which is a density-weighted time averaging scheme.

The momentum conservation is:

$$\frac{D\mathbf{v}}{Dt} = -\frac{1}{\rho} \nabla P + \mathbf{g} + \nu_0 \nabla^2 \mathbf{v} + \frac{1}{\rho} \nabla \bar{\tau} \qquad (2.20)$$

where the laminar term can be treated following Eq. (2.18) and $\bar{\tau}$ represents the SPS stress tensor.

The eddy viscosity assumption (Boussinesq’s hypothesis) is often used to model the SPS stress tensor using Favre-averaging (for a compressible fluid):

$$\frac{\tau_{ij}}{\rho} = 2\nu_t S_{ij} - \frac{2}{3}k\delta_{ij} - \frac{2}{3}C_I\Delta^2\delta_{ij}\left|S_{ij}\right|^2 \tag{2.21}$$

where τ_{ij} is the sub-particle stress tensor, $\nu_t = [min(C_s, \Delta l)]^2 \cdot |S|$ the turbulence eddy viscosity, k the SPS turbulence kinetic energy, C_s the Smagorinsky constant (0.12), $C_I = 0.0066$, Δl the particle-particle spacing, $|S| = \left(2S_{ij}S_{ij}\right)^{1/2}$, S_{ij} the element of the SPS strain tensor.

So, following Dalrymple & Rogers (2006), Eq. 2.20 may be written in SPH notation as

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b m_b\left(\frac{P_a}{\rho_a^2} + \frac{P_b}{\rho_b^2} + \frac{\tau_a}{\rho_a^2} + \frac{\tau_a}{\rho_a^2}\right)\nabla_a W_{ab} + \sum_b m_b\left(\frac{4\nu_0\mathbf{r}_{ab}\boldsymbol{\nabla}_a W_{ab}}{(\rho_a + \rho_b)|\mathbf{r}_{ab}|^2}\right)\mathbf{v}_{ab} + \mathbf{g} \tag{2.22}$$

2.5 Continuity equation

The fluid in the standard SPH formalism is treated as compressible, which allows the use of an equation of state to determinate fluid pressure. However, the compressibility is adjusted to slow the speed of sound so that the time step in the model (dependent of the speed of sound) has a reasonable value.

The changes on the fluid density are calculated by:

$$\frac{d\rho_a}{dt} = \sum_b m_b\mathbf{v}_{ab}\boldsymbol{\nabla}_a W_{ab} \tag{2.23}$$

A time differential is used instead of a weighted summation of the mass terms (Monaghan, 1992), since the weighted summation results in a decrease of the density in the interface between fluids, near the free surface, and in the proximity of the boundaries.

2.6 Equation of state and compressibility

As mentioned in the previous section, in the standard SPH formalism, the fluid is treated as weakly compressible, which facilitates the use of an equation of state to determinate the pressure of the fluid since this is faster than solving the Poisson's equation that appears in the incompressible approach. Following Monaghan (1992) and Batchelor (1974), the relationship between pressure and density follows the Tait's equation of state. It can be noted that a small oscillation in density will generate a large variation in pressure.

$$P = B\left[\left(\frac{\rho}{\rho_0}\right)^{\gamma} - 1\right] \tag{2.24}$$

The parameter B is related to the compressibility of the fluid; $\rho_0=1000.0\ Kg/m^3$ is the reference density, taken as the density of the fluid at the surface and γ is the polytrophic constant, usually ranges between 1 and 7.

The speed of sound, c, is defined as the square root of the derivative of Tait's equation with respect to density:

$$c^2(\rho) = \frac{\partial P}{\partial \rho} = \frac{B\gamma}{\rho_0}\left(\frac{\rho}{\rho_0}\right)^{\gamma-1} = \frac{B\gamma}{\rho_0^{\gamma}}\rho^{\gamma-1} \tag{2.25}$$

$$c_0^2 = c^2(\rho_0) = \frac{\partial P}{\partial \rho}|_{\rho=\rho_0} = \frac{B\gamma}{\rho_0} \tag{2.26}$$

where c_0 is the speed of sound at the reference density; $B = c_0^2\rho_0/\gamma$ which provides a maximal limit for the density.

The choice of B plays a key role since it determines the speed of sound. Using a value corresponding to the real value of the speed of sound in water, a very small time step must be chosen for numerical modelling, based on the Courant-Fredrich-Levy condition. Monaghan showed that the speed of sound could be slowed significantly in an artificial way for fluids without affecting the fluid motion, however Monaghan (1994) suggests that the minimum sound speed should be about ten times greater than the maximum expected flow speeds.

2.7 Moving the particles

Particles are moved using the XSPH variant (Monaghan, 1989):

$$\frac{d\mathbf{r}_a}{dt} = \mathbf{v}_a + \varepsilon \sum_b \frac{m_b}{\bar{\rho}_{ab}} \mathbf{v}_{ba} W_{ab} \tag{2.27}$$

where $\bar{\rho}_{ab} = \frac{1}{2}(\rho_a + \rho_b)$ and ε is a constant, whose value ranges between zero and the unity, $\varepsilon = 0.5$ is normally used.

This method is a correction for the velocity of the particle a. This velocity is recalculated taking into account the velocity of the particle and the average velocity of all particles that interact with the particle a. Due to the compact support of the *kernel*, only the closest neighbourhood will be included. This correction allows particles to be more organised, and for high fluid velocities, helps to avoid particle penetration.

2.8 Filters and corrections

SPH may produce uncertainties in the surroundings of the free surface or near the boundaries. Usual locations where the normalization condition is not properly satisfied. A great effort has been made to overcome this issue. Mainly focusing on approximations that correct the *kernel* and/or its first derivative, or applying density filters that can help to stabilise the pressure fields.

2.8.1 Density filter

While the dynamics from SPH simulations are generally realistic, the pressure field of particles may exhibit large oscillations. One of the most straightforward, and computationally least expensive is to perform a filter over the density of the particles and the re-assign a density to each particle with the density values of the neighbouring particles (Colagrossi & Landrini, 2003). The Shepard filter is a fast and simple correction to the density field. The following protocol is applied each certain number of time steps:

$$\rho_a^{new} = \sum_b \rho_b \widetilde{W}_{ab} \frac{m_b}{\rho_b} = \sum_b m_b \widetilde{W}_{ab} \qquad (2.28)$$

where the *kernel* has been corrected with:

$$\widetilde{W}_{ab} = \frac{W_{ab}}{\sum_b W_{ab} \frac{m_b}{\rho_b}} \qquad (2.29)$$

2.8.2 Kernel and kernel gradient correction.

Some corrective terms of the *kernel* to improve the precision of the *kernel* and the estimations of the *kernel* gradient are available on the literature (Belytschko et al., 1998; Bonet & Lok, 1999; Vila, 1999; Chen & Beraun, 2000).

The *kernel* correction proposed here (Bonet & Lok, 1999) consists in a vector variable $\boldsymbol{f}_a$ may be expressed as:

$$\boldsymbol{f}_a = \frac{\sum_b \frac{m_b}{\rho_b} \boldsymbol{f}_b W_{ab}}{\sum_b \frac{m_b}{\rho_b} W_{ab}} \qquad (2.30)$$

In the other hand, the *kernel* gradient correction $\widetilde{\nabla} W_{ab}$ also should be used replacing the *kernel* ∇W_{ab}, therefore:

$$\widetilde{\nabla}W_{ab} = L_a \nabla W_{ab} \tag{2.31}$$

$$L_i = M_i^{-1} \tag{2.32}$$

$$M_a = \sum_{b}^{num} \frac{m_b}{\rho_b} \nabla W_{ab} \otimes (x_a - x_b) \tag{2.33}$$

where *num* is the total number of particles interacting with particle *a*.

2.9 Time step algorithm

The physical magnitudes (velocity, density, position and density) change every time step due to the forces obtained during particle interactions. The time integration to compute the new values of these quantities at the next time in SPH should be at least of second order to obtain accurate results. There are two possibilities implemented: (i) Verlet and (ii) Symplectic.

Consider the position (2.27), momentum (2.13) and density (2.23) equations in the following form:

$$\frac{d\mathbf{r}_a}{dt} = \mathbf{V}_a \tag{2.34}$$

$$\frac{d\mathbf{v}_a}{dt} = \mathbf{F}_a \tag{2.35}$$

$$\frac{d\boldsymbol{\rho}_a}{dt} = D_a \tag{2.36}$$

2.9.1 Verlet algorithm

This time step algorithm (Verlet, 1967) is divided in two parts. Generally the variables are calculated according to:

$$\begin{gathered} \mathbf{r}_a^{n+1} = \mathbf{r}_a^n + \Delta t \mathbf{V}_a^n + 0.5\Delta t^2 \mathbf{F}_a^n; \\ \mathbf{v}_a^{n+1} = \mathbf{v}_a^{n-1} + 2\Delta t \mathbf{F}_a^n \\ \rho_a^{n+1} = \rho_a^{n-1} + 2\Delta t D_a^n \end{gathered} \tag{2.37}$$

Once each N time steps (being N order of 40), the variables are obtained according to:

$$\begin{gathered}\mathbf{r}_a^{n+1} = \mathbf{r}_a^n + \Delta t\mathbf{V}_a^n + 0.5\Delta t^2\mathbf{F}_a^n; \\ \mathbf{v}_a^{n+1} = \mathbf{v}_a^n + \Delta t\mathbf{F}_a^n \\ \rho_a^{n+1} = \rho_a^n + \Delta t D_a^n\end{gathered} \tag{2.38}$$

This helps to avoid the time integration diverging since the equations are no longer coupled.

2.9.2 Symplectic algorithm

In the Symplectic algorithm (Liemkhuler et al., 1999), first, the values of density and acceleration are calculated at half time step as:

$$\begin{gathered}\rho_a^{n+\frac{1}{2}} = \rho_a^n + \frac{\Delta t}{2}\frac{\mathrm{d}\rho_a^n}{\mathrm{d}t} \\ \mathbf{r}_a^{n+\frac{1}{2}} = \mathbf{r}_a^n + \frac{\Delta t}{2}\frac{\mathrm{d}\boldsymbol{r}_a^n}{\mathrm{d}t}\end{gathered} \tag{2.39}$$

In a second time step $(d(\omega_i\rho_i\mathbf{v}_i)^{n+\frac{1}{2}})/dt$ gives the velocity and the position of the particles at the end of each time step:

$$\begin{gathered}(\omega_a\rho_a\mathbf{v}_a)^{n+1} = (\omega_a\rho_a\mathbf{v}_a)^{n+\frac{1}{2}} + \frac{\Delta t}{2}\frac{d(\omega_i\rho_i\mathbf{v}_i)^{n+\frac{1}{2}}}{dt} \\ \mathbf{r}_a^{n+1} = \mathbf{r}_a^{n+\frac{1}{2}} + \frac{\Delta t}{2}\mathbf{v}_a^{n+1}\end{gathered} \tag{2.40}$$

At the end of the time step $d\rho_a^{n+1}/dt$ is calculated using the actualized values of $\mathbf{v}_a^{n+1}$ and $\mathbf{r}_a^{n+1}$ (Monaghan, 2005).

2.9.3 Variable time step

The time step is dependent on the fluid properties. In general, the time step depends on the Courant condition of the force and viscous diffusion terms. A variable time step Δt is calculated according to Monaghan & Kos, (1999):

$$\begin{gathered}\Delta t = C \cdot min(\Delta t_F, \Delta t_{CV}), \\ \Delta t_F = min_a \left(\sqrt{h/f_a}\right), \\ \Delta t_{CV} = min_a \frac{h}{c_s + max_b \left| \frac{h v_{ab} x_{ab}}{r_{ab}^2} \right|}\end{gathered} \tag{2.41}$$

Here Δt_F is based on the force per unit of mass $|f_a|$, and Δt_{CV} controls the Courant condition and the viscosity of the system. C is a constant that can vary between 0.1 and 0.3.

2.10 Boundary conditions

Dynamic boundaries were first introduced by Dalrymple & Knio, (2000) and further studied by Crespo et al. (2007b). Dynamic boundaries consist of a set of boundary particles that satisfy the same continuity equation, Eq. 2.23, as the fluid particles, thus their density and pressure also evolve creating a repulsive mechanism when a fluid particle approaches a boundary particle. However their position is not given by integrating velocity in time; a static boundary will have zero velocity and a moving boundary will have a prescribed motion.

Fig. 2.4 shows the repulsion mechanism. The incoming particle, a, increases the density locally (Fig. 2.4 a) according to Eq. 2.23, which results in an increase in pressure following Eq. 2.24 (Fig. 2.4 b) and in an increase of the pressure term (P/ρ^2) in Eq. 2.12. The normalized pressure term, $NPT_z = (P/\rho^2)_z / (P/\rho^2)_R$, is represented in Fig. 2.4 c, where z refers to the distance from the incoming particle to the wall and $R=2h$. Calculations were run without viscosity.

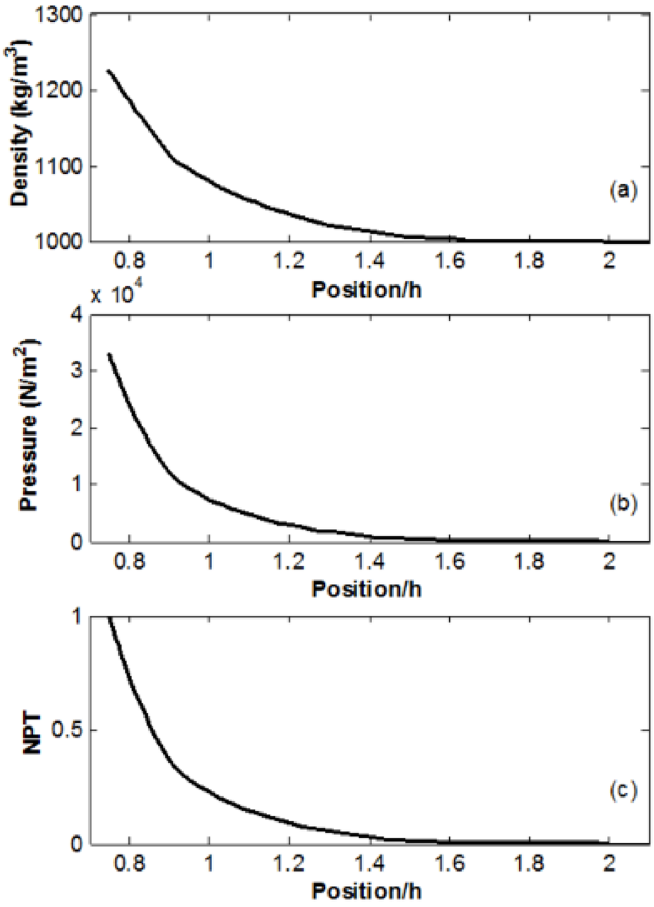


Figure 2.4. Variation of density (a), pressure (b) and normalized pressure term (c) for a moving particle approaching a solid boundary. Calculations were run without viscosity.

Validations with dam-break flows have been published with good results (Crespo et al., 2011) and these BCs have also been shown to be suitable to reproduce complex geometries (Altomare et al., 2014).

A known difficulty of this formulation is the overestimation of the density resulting from an entropy jump across the fluid/solid interface. This results in an increased distance of fluid-solid particles due to the added force from the pressure gradient, effectively disturbing the viscous forces computed at the interface and leading to unphysically larger boundary layers.

3. SPH implementation

The SPH formulation presented in the previous chapter has been implemented in the *DualSPHysics* code (available at www.dual.sphysics.org). The implementation consists in the iteration of three main stages; (i) the creation of a neighbour list, (ii) computing the interactions between particles, where the momentum and continuity equations are solved, and (iii) time integration of the system magnitudes for the next time step of the simulation, for a more detailed description the reader is referred to Gómez-Gesteira et al. (2012b). In Fig. 3.1, a conceptual diagram of the implementation of the code can be observed. The three main steps are repeated till the end of the required simulation time. There is a first data reading from the hard disk with the information of the execution, the initial conditions and the initial values of the physical magnitudes of the particles (position, velocity, density…). The data is kept in the main memory, and only move each several time-steps is stored on the hard disk memory (a much slower memory). When saving simulation information, such as new particle positions, velocities, etc., only the necessary data is stored. This task is not performed very often, so data is stored after many steps (at a specified frequency).

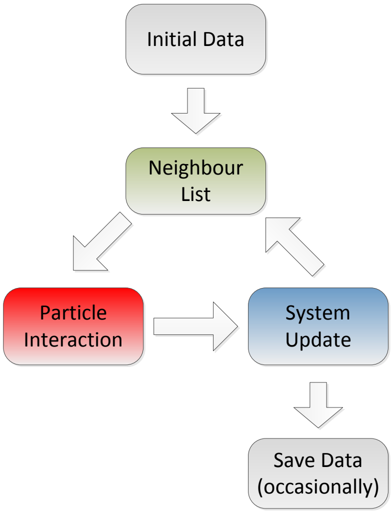


Figure 3.1. Conceptual diagram of the three stages of a SPH code.

3.1 *Neighbour list*

The interactions between particles on SPH are short range, so the use of a neighbour list will reduce significantly the number of computation. For this purpose, instead of comparing the distance between every particle and all the particles in the domain to check if they are inside the interaction range (which would be very expensive and inefficient), a *linked cell list* is used. Since particles only interact with the particles inside the range where the *kernel* function has compact support, the domain is divided in cells of the size of the *kernel*, (usually $2h$, being h the smoothing length). In this way, the search for neighbours is reduced to the same cell and the adjacent cells, since only these cells can contain the real neighbours of each particle. The Fig. 3.2 helps to explain this procedure: the particle of study is marked with a black circle, the domain has been divided in $2h$-sized cells, blue particles belong to the adjacent cells and they are potential neighbours, but only the red particles located within the circle of radius $2h$ are real neighbours. Therefore, the number of interactions to be computed at each time step decreases from Np^2 interactions (being Np the total number of particles in the system) to an order of $Np \cdot log(Np)$. A list of neighbours of each particle is not created, but a list of particles stored according to the cells they belong. In addition, all particle data are reordered according to the order of the cells to obtain a faster memory access during computation since a more efficient usage of the cache memory is achieved. The reader is referred to Dominguez et al., (2011a) and for a more detailed description of the neighbour list.

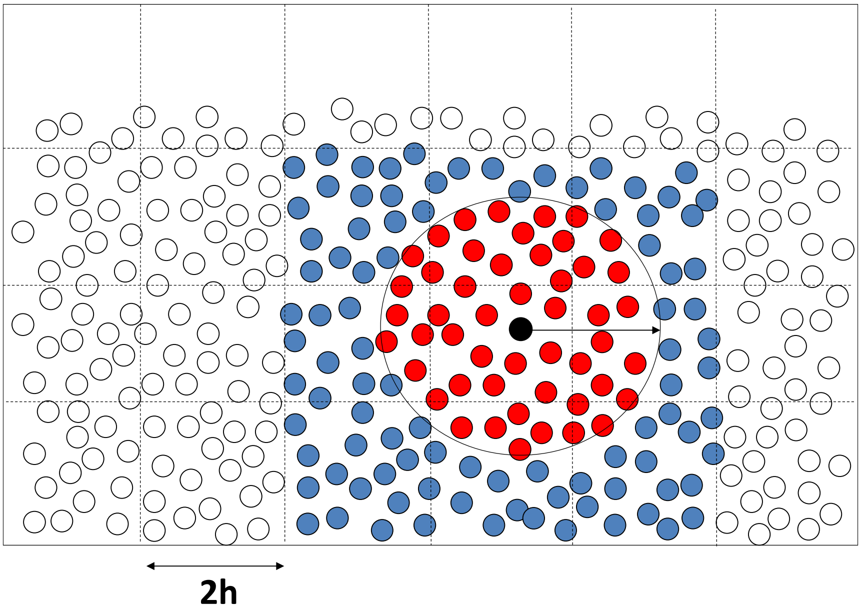


Figure 3.2. Domain division in cells and real neighbours (red) of a given particle (black).

3.2 Particle interaction

The second stage of an SPH code is the force computation. Firstly, each particle looks for neighbours in the adjacent cells and in its own cell. As can be seen in Fig. 3.2, a potential neighbour, blue particles, becomes a real one, red particles, if the relative distance between particles is lesser than the range of interaction. If particles are in the *kernel*'s domain, the equations of momentum conservation and continuity (Eq. 2.22 and 2.23) are solved for each pair of particles. This process can be really time consuming. According to Dominguez et al., (2011), the creation of a neighbour list is the most optimal approach to solve this issue. One of the conclusion of this work is that the particle interaction took over the 90% of the total time using different approaches.

Applying symmetry to the interactions of particles is a way to obtain an improvement in the performance. Due to Newton’s third law, when the interaction force of one particle with a neighbour is calculated, the force of the neighbour particle on the first one is known. This force must have the same magnitude but opposite direction. Thus, the number of adjacent cells to search for neighbours can be reduced if the symmetry in the particle interaction is applied. This process reduces, significantly, the computational time.

3.3 System update

Finally, once the forces between neighbouring particles are known, the new physical quantities for all particles are updated based on the quantities at the previous time step, the forces calculated in the interaction between particles, and the value of the interval or time step (section 2.8).

3.4 Parallel implementation

The execution time of the SPH simulations increase with the number of particles. All simulations need to be executed at a reasonable time, in simulations with a large number of particles the use of high performance computing (HPC) is necessary.

The accessible acceleration technologies proposed here are: (i) use the multiple cores of a CPU or (ii) use new computational architectures such the graphic processor units (GPU), imported from the video game industry.

The CPUs are the main processing units on any computer. Modern CPUs have many cores (4, 8, 16…) and provide the main computational resource in some HPC facilities. For example, some of the TOP500 facilities only use CPUs as main computational device such as Mira (US) ranked 5th, JUQUEEN (Germany) ranked 8th, or Vulcan (US) ranked 9th (http://www.top500.org/list/2014/11/ the TOP500 is list of the most powerful HPC facilities. This ranking is updated every six months). To obtain the best performance of the CPUs the implementations must use all the cores of each single CPU and also many CPUs at the same time. OpenMP and MPI are two APIs that allow the developers to use all the cores of a CPU and multiple CPUs improving this way the performance compared to serial code.

The GPUs were designed to deal with large flows of data (such as data of the SPH particles) and to render pixels at speed in the order of 60 frames per second (fps), reaching in some cases up to 200 fps. From a computational point of view, the GPUs are very efficient due to its multi-thread capability. Thanks to the constant evolution of the video game world and the multi-processor architecture of the GPUs, its computation power has increased much faster than the CPUs. In this way, the GPUs appear in the market as an accessible alternative to accelerate numerical models. Furthermore, the GPUs are easy to maintain in comparison to the large high performance systems composed by hundreds of multi-core machines.

Therefore, most of the sequential tasks and operations that involve a loop over all particles in a serial code can be parallelised. In the case of SPH, a high percentage of the total runtime belongs to the force computation, so the biggest efforts should be focused on parallelising this part of the code.

3.4.1 Multi-core CPU parallelization

Firstly, the work load is distributed among the different cores of a multicore CPU using the parallel programming framework OpenMP. The physical domain is divided into sub-domains, and the computations of different sub-domains are executed by different threads of the CPU. A dynamic load balancing between the sub-domains is also implemented. This multi-core code does not present significant changes in comparison to the single-core version. One of the main advantages of this technique is the reduction in the time dedicated to communication between different cores since they share the same memory space, however the main limitation that exists is the number of cores on a single CPU.

3.4.2 GPU parallelisation

The second approximation is the use of the GPU technology. To obtain an efficient use of all the capabilities of the architecture of these new graphic cards, the coding environment called CUDA is used, which has been developed by Nvidia (http://developer.nvidia.com/cuda). GPUs are massively multithreaded many-core chips. For example, with a GTX 480 card a maximum of 23,040 threads (15 multiprocessors with 1,536 threads per multiprocessor) can be in execution. Thus, in this case, the execution threads of a GPU carry out the sequential tasks over the different particles.

A first approach, on this new parallelization technique, consisted on the execution of *Particle interaction* on the GPU device while the other processes (*Neighbour list* and *System update*) are executed on the CPU (left image of Fig. 3.3). This technique is less efficient since particle data and neighbour list information must be transferred between both processing units each time step, which is very expensive in time. The most efficient option is keep all data in the memory of the GPU where all processes are parallelised. Only output data requires transfer from GPU to CPU. This process is carried out at a specified frequency (one out of one hundred time steps at most) and only represents a low percentage of the total runtime (right image Fig. 3.3).

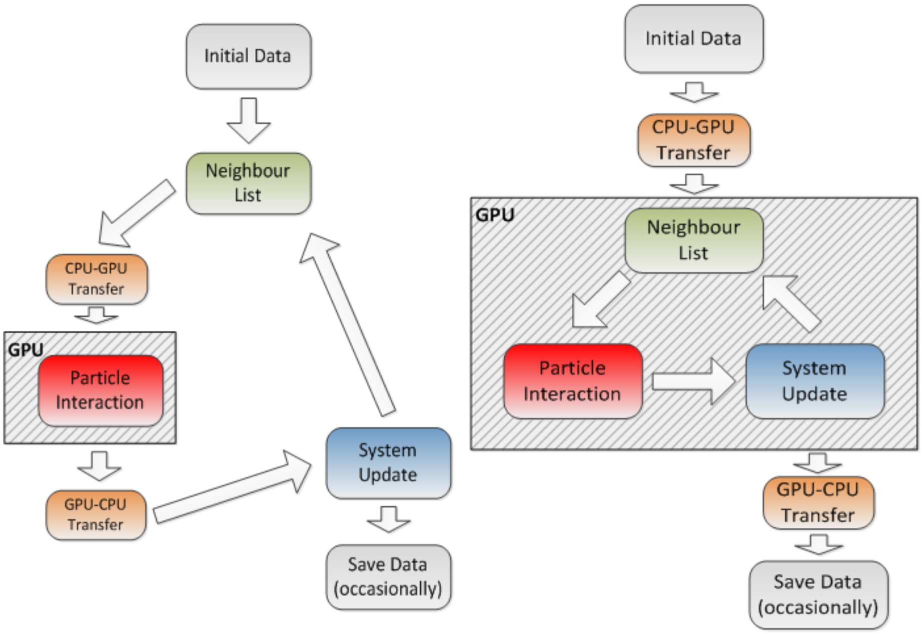


Figure 3.3. Conceptual diagrams of the partial GPU implementation (left) and a full GPU implementation (right).

This full implementation of the SPH code in GPU follows the same linked cell list implemented on CPU. However, the way the forces are computed in GPU differs from the CPU code. In the CPU code, the interaction of all particles of one cell with the neighbouring cells is carried out (interaction in terms of cell). In the GPU code, each particle looks for all its neighbours sweeping all adjacent cells and its own cell. In this way, one GPU execution thread computes, for only one particle, the force resulting from the interaction with all its neighbours. Symmetry of the pair-wise particles cannot be applied in an efficient way since several threads can be modifying the same memory position simultaneously that gives rise to an error.

3.5 Performance

In order to analyse the performance of the parallel versions of the *DualSPHysics* code comparing with the serial version, the experiment of Yeh and Petroff (Washington University) has been used. This experiment has been already tested for the validation of the 3D version of the *SPHysics* code (Gómez-Gesteira & Dalrymple, 2004). In the experiment, a dam break was reproduced within a rectangular tank, with a volume of water initially contained behind a thin gate at one end of the box and a tall structure placed inside the tank. Fig. 3.4 shows different instants of the simulation involving one million particles.

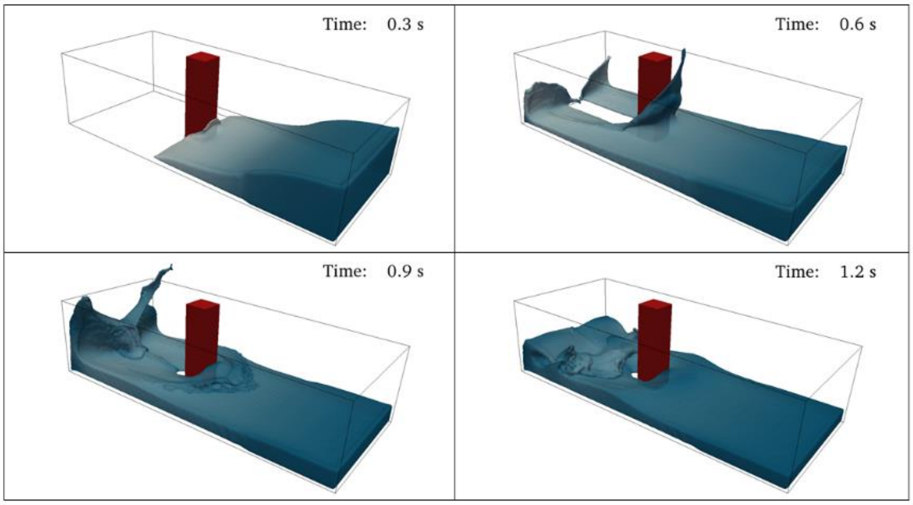


Figure 3.4. Different instants of dam-break flow impacting on a structure with SPH.

Using this validation case and varying the number of particles (*Np*) of the simulation, a comparison between runtimes of the different parallel codes (multi-core and GPU) is analysed. The code is executed on a CPU Intel® Core ™ i7 940 at 2.93GHz with 4 cores and on the GPU GTX 480 at 1.37GHz with 480 cores, which is hosted by an Intel Xeon E5620 at 2.4GHz.

Fig. 3.5 shows the computational runtimes for the CPU serial code (red line), CPU multi-core (blue line) and GPU (green line); the physical time of the simulation was 1.5 seconds changing the number of particles (N_p).

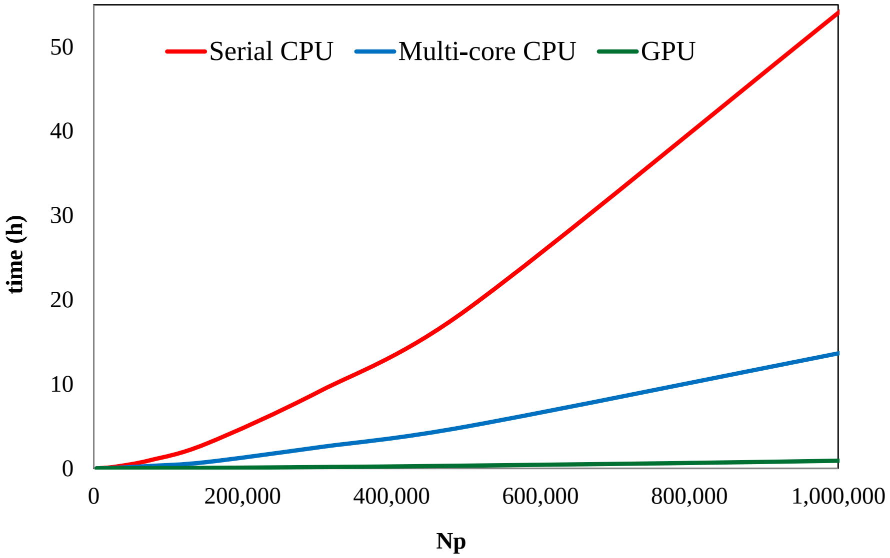


Figure 3.5. Runtime using CPU serial, CPU multi-core and GPU codes for different number of particles.

Note that in the tests performed here the maximum number of particles allocated to a single GTX 480 (with 1.5 GB of memory space) is five million. The same calculations would not be feasible on a CPU in a reasonable time. Thus, results are presented for a maximum number of particles of 1 million for the CPU approaches (Crespo et al., 2011). It can be observed that the computational runtime increases dramatically with the number of particles when simulations are performed on the CPU device while the GPU technology enables acceleration of the code even with only one GPU card (Dominguez et al., 2013).

Table 3.1 summarises the computational time and the speedup achieved by using the multi-core CPU and GPU codes against single-core and four-core CPU approaches. The multi-core implementation accelerates the code up to four times. The simulation of one million particles takes more than 2 days on the single-core CPU and only 55 minutes on a single GPU, thus, a speedup of 60 is achieved.

Table 3.1. Runtimes and speed-ups for both CPU and GPU codes simulating 1 million particles

Code Version	Runtime (Hours)	Speedup vs. Serial CPU	Speedup vs. Multicore
Serial CPU	54.84	--	
Multicore CPU	13.84	**4.0**	--
GPU	0.92	**59.8**	**15.1**

4. Case I: Coastal protection

The capabilities of the *DualSPhysics* numerical tool for coastal protection are shown in this chapter. Validations with analytical solutions and experimental data are presented for wave-structure interactions and a case of application that represents a realistic wave-promenade interaction is studied.

4.1 Design of SPH simulations

The process to generate the geometry of a numerical simulation based on particles is not trivial and can give rise to a significant computational cost. Generating the initial configuration of particles for a SPH simulation implies filling volumes of irregular shapes using particles that must be located equidistant. Since *DualSPHysics* has the capacity to simulate large number of particles at reasonable runtimes, sophisticated pre-processing and post-processing tools are needed to deal with the large data sets.

The pre-processing tool employed in this work (Dominguez et al., 2011b) employs a 3D mesh to locate particles. The idea is to build any object using particles, so these particles are created in the nodes of the 3D mesh; firstly, the mesh nodes around the object are defined and then particles are created only in the nodes that compose the shape the desired geometry. In addition, complex 3D models (.cad, .3ds, .stl, .ply, .dwg, .dxf, .shp, .igs, .vtk, .csv …) created with common used design software such as AutoCAD, Blender or 3D Studio Max can be imported, split into different triangles and each triangle can be converted in particles using the previous procedure. Fig. 4.1 shows an example of the CAD file of a pump model that can be decomposed into triangles and then discretized into particles. A more complex geometry of a seawalk including a balustrade, a lamppost and a bench can be also observed as a cloud of particles in Fig. 4.2.

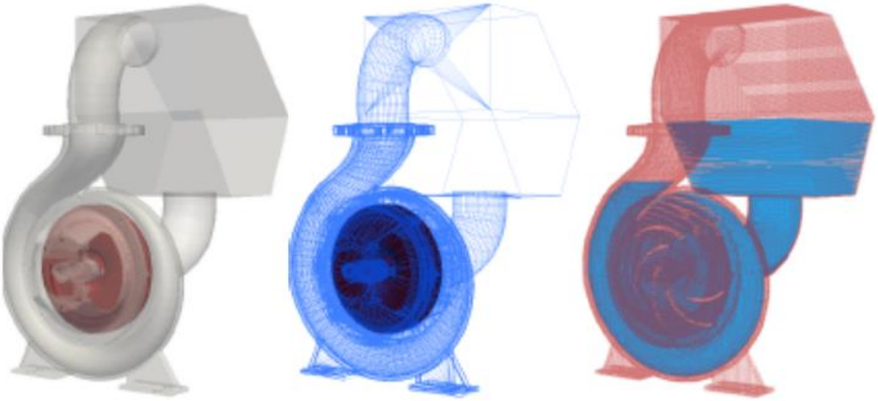


Figure 4.1. Example of a CAD file (left) that can be imported and converted into particles (right).

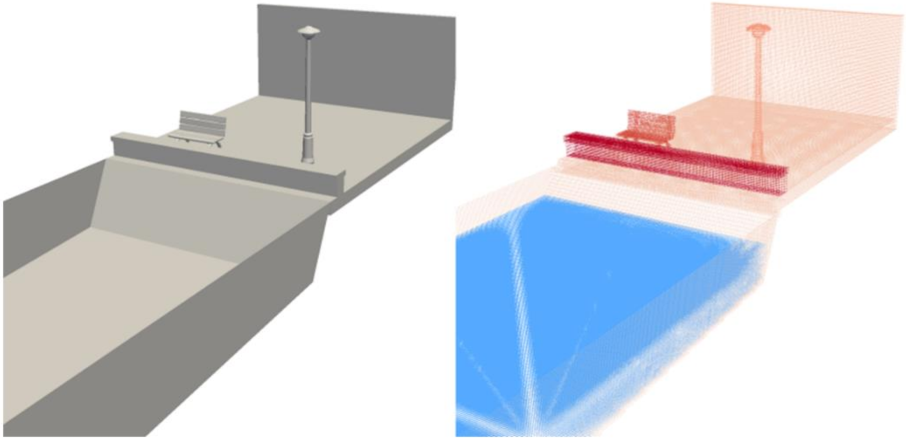


Figure 4.2. Example of complex geometry (left) and the numerical configuration using particles (right).

4.2 Validation

This section shows the ability of *DualSPHysics* code to simulate wave-structure interaction problems and to compute in an accurate way the force exerted onto the structure.

The procedure to compute numerically the forces exerted on a coastal structure is simple and is based on the physical magnitudes known of each SPH particle. The total force exerted by the wave on a structure can be numerically measured as the summation of the forces computed in the proximities of the boundaries (that represent the structure) resulting from the interaction with the fluid neighbouring particles:

$$F_{total} = -m_k \cdot \sum_{i=1}^{k} \left(\frac{d\mathbf{v}_{\boldsymbol{k}}}{dt} \right) \tag{4.1}$$

therefore, F_{total} is the summation of the forces experimented by all the boundary particles k that form the structure, m_k is the mass of a boundary particle and $\frac{d\mathbf{v}_{\boldsymbol{k}}}{dt}$ is the acceleration computed for each boundary particle k with all its fluid neighbouring particles using Eq. 4.1.

Other methods based on the pressure values of the boundary particles were used before showing good comparison with experimental results (Gómez-Gesteira & Dalrymple, 2004; Crespo et al., 2011). However the advantage of the procedure used here is its independence on the boundary condition implemented in the SPH method making it more versatile.

4.2.1 Numerical force compared with analytical solution

Firstly, the value of the force against one wall exerted by still water is numerically analysed with SPH and compared with its analytical solution. Given a 2D water tank with a volume of water 1m high and 1m long, the hydrostatic force against one of the vertical walls is known as: $1/2 \cdot \rho \cdot g \cdot H^2 \cdot B$, where *H=1m* is water elevation, and *B* is the width of the wall (not considered in 2D). Thus, in this case the analytical force will be *4,905N*. Fig. 4.3 shows the time history of the numerical values of the force using different resolution (different initial interparticle distance *dp*) to discretize the volume of water *1m* x *1m* with SPH. Three different resolutions have been simulated; (i) 125 particles with *dp=0.1m,* (ii) 380 particles with *dp=0.05m* and (iii) 2,450 particles with *dp=0.02m*. Some force oscillations that are mitigated in time can be observed at the first steps of the simulation. However, these force oscillations decrease with the value of *dp* and the important issue here is that the three simulations converge finally to the same value, which is in concordance with the analytical one (*4,905N*). Note that the method provides accurate solution even at low resolutions. On the other hand, the reason for the periodic oscillations on the force is also connected to the weakly compressible assumption that allows the use of the Tait's equation of state that relates pressure and density. As mentioned in Section 2.6, little oscillations in density give rise to large variations in pressure according to Eq. 2.24 and these variations are more remarkable when using low resolutions.

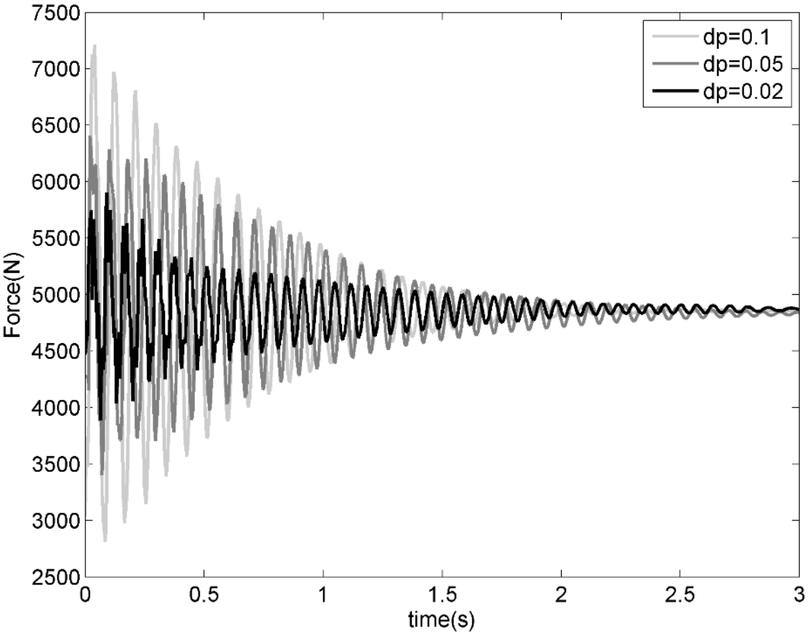


Figure 4.3. Time series of the numerical force exerted onto a vertical wall using different resolutions.

4.2.2 Numerical force compared with experimental data

SPH results are also compared with the experimental data provided by Yeh and Petroff at the University of Washington. A complete description of the experimental setup can be seen in Arnason, (2005). This experiment was a dam break problem confined within a rectangular box 160 cm long, 61 cm wide and 75 cm high. The volume of water initially contained behind a thin gate at one end of the box was 40 cm long x 61 cm x 30 cm high. A tall structure, which was 12 cm x 12 cm x 75 cm in size, was placed 50 cm downstream of the gate and 24 cm from the nearest sidewall of the tank. An instant of the SPH simulation using *DualSPHysics* is depicted in Fig. 4.5. Different numerical parameters related to the number of particles and SPH formulation are summarised in Table 4.2.

Table 4.1. Numerical parameters of the simulation of Yeh and Petroff experiment.

Parameter	Value
Particle size *dp* (m)	0.01
Smoothing length *h* (m)	0.015
Number of particles *np* (m)	101,809
Artificial viscosity α in Eq. 2.15	0.01
Speed of sound c_o in Eq. 2.26	$20 \cdot V_{max}$
CFL number *C* in Eq. 2.41	0.2
Physical time simulated (s)	3

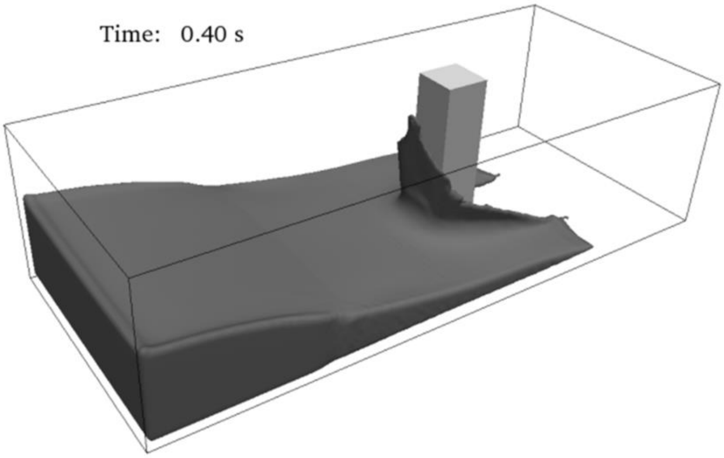


Figure 4.4. Instant (t=0.4s) of the simulation of the fluid-structure interaction using *DualSPHysics*.

Experimental measurements included the time series of the net force exerted on the structure. These experimental data were also used by Gómez-Gesteira & Dalrymple, (2004) to validate their SPH calculations. Fig. 4.6 shows the comparison between the numerical force (dark line) and the experimental data (light line) and it can be concluded that SPH is able to reproduce the dynamic response of the system.

In conclusion, *DualSPHysics* is able to perform simulations where wave propagation and wave-structure interaction are properly reproduced. At this stage, the main objective of this work is to examine the feasibility of the SPH method when applying to a real engineering problem in the field of the coastal protection.

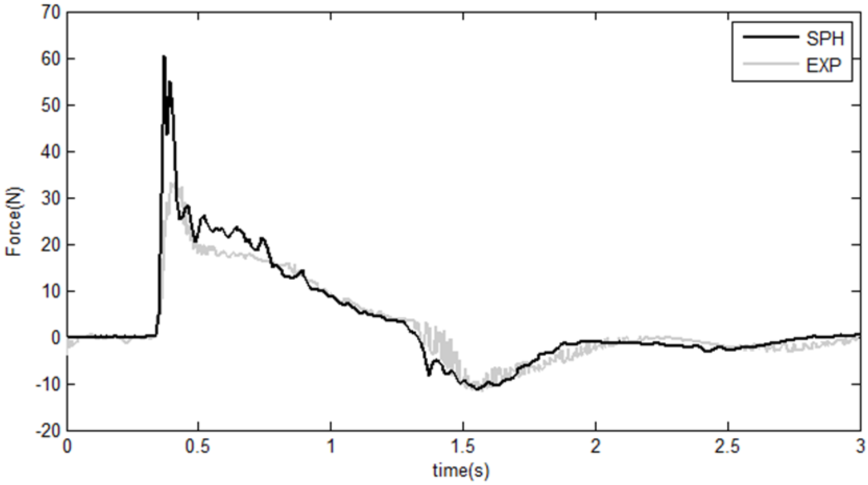


Figure 4.5. Comparison between numerical (dark) and experimental (light) forces exerted by the incoming wave on the structure.

4.2.3 Wave propagation compared with experimental data

Numerical wave heights are compared with experiments to prove the capability of the model to propagate waves. In particular, the CIEMito flume of Universitat Politècnica de Catalunya (UPC) has been used to carry out an experiment where regular waves (0.07m high, and 1s of period) have been generated by a wave piston in a tank 5.05m long, 0.38m wide and with an initial water depth of 0.25m. Three gauges were placed 0.38m (WG1), 1.83m (WG2) and 3.12m (WG3) to record the experimental water elevation (η). Note that the length of tank (5.05m), the period of the generated waves (1s), the duration of the experiment (10s) and the location of the gauge to measure the water elevation have been selected to avoid problems of reflection when waves reach the end of the tank. In order to simulate numerically the experiment, different SPH parameters (Table 4.1) are considered to obtain accurate results to be compared with the experimental data. Fig. 4.4 shows the comparison between the experimental data and the numerical water elevations computed with *DualSPHysics*. The agreement between the numerical and experimental signal depicted in the figures prove the reliability of the model to propagate waves.

Table 4.2. Numerical parameters of the simulation of CIEMito experiment.

Parameter	Value
Particle size *dp* (m)	0.002
Smoothing length *h* (m)	0.00247
Number of particles *np* (m)	318,196
Artificial viscosity α in Eq. 2.15	0.01
Speed of sound c_o in Eq. 2.26	$16 \cdot V_{max}$
CFL number *C* in Eq. 2.41	0.2
Physical time simulated (s)	10

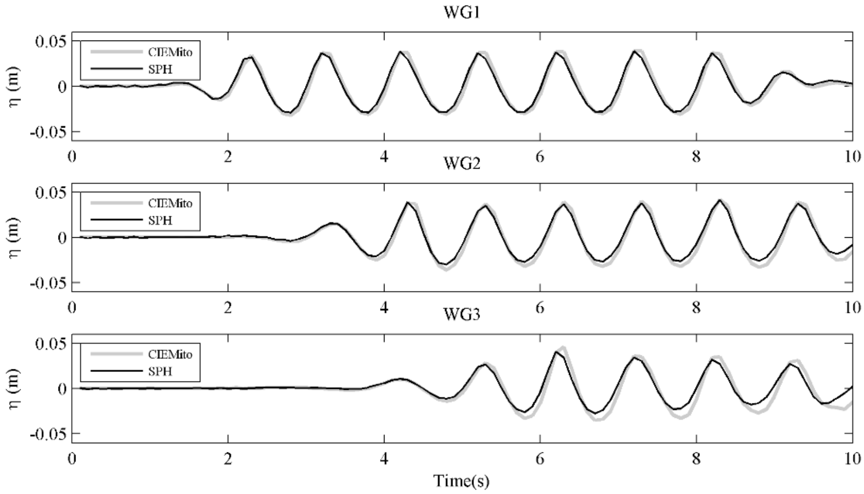


Figure 4.6. Comparison between numerical (dark) and experimental (light) water elevation at three different locations during the wave propagation

4.3 Application case

Once validated the code for both wave propagation and structure interaction, a case of application where the two techniques are combined in the simulation of a real-life situation. The testcase consists on the propagation of an incoming wave and its interaction with a seawalk that also includes some urban furniture (such as a balustrade, a lamppost or a bench). Therefore, the initial geometry has been created using real dimensions (see Fig. 4.7). The use of a model like *DualSPHysics* is of special importance in a case like this. Due to climate change and sea level rise new studies are needed for coastal protection. *DualSPHysics* has the capability to simulate different sea conditions and detect new flood and risk areas.

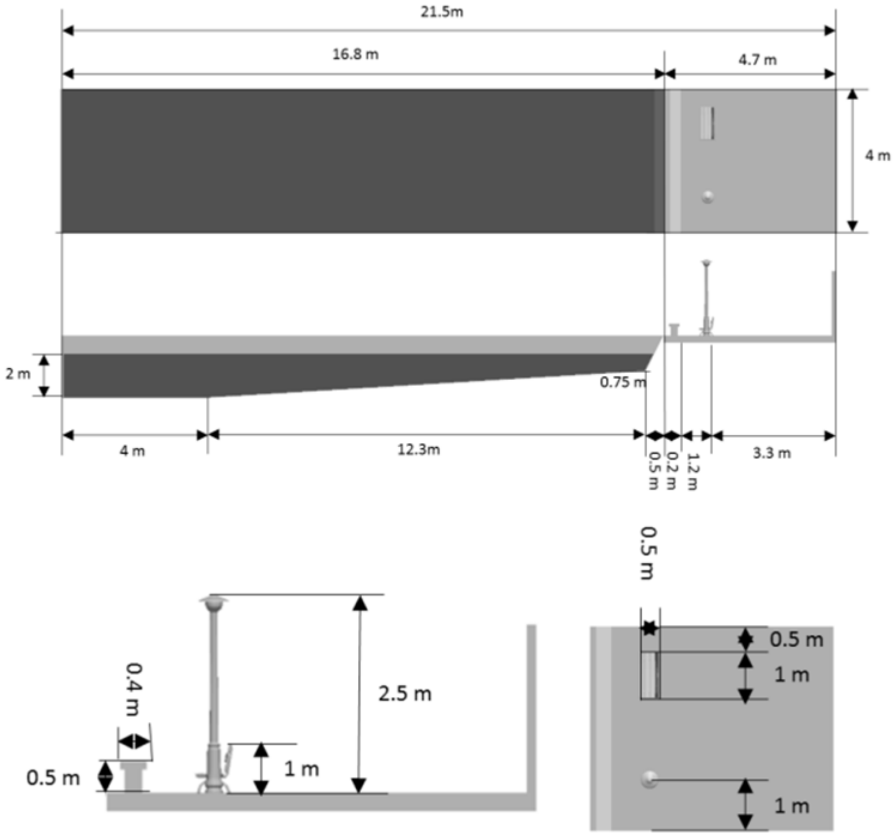


Figure 4.7. Top panel shows the initial configuration of the application, aerial and side view, including the dimensions. Bottom panel present the detailed view of the promenade and the urban furniture placed on the seawalk, aerial and side view, with dimensions.

4.3.1 *Numerical simulation*

The case of application is simulated using the *DualSPHysics* code. The wave is generated and propagated using a numerical wavemaker, and then impacts with the balustrade of the seawalk and interacts with the urban furniture.

In order to create the initial cloud of SPH particles that represents the geometry of the testcase and the volume of water (Fig. 4.2), an interparticle distance of $dp=0.026m$ was used. This implies a smoothing length $h=0.039m$ and a total amount of particles in the system of 3,969,726 where around 500,000 are boundary particles and 3.5M are fluid ones. This resolution has been chosen to get a representative numerical system of the realistic promenade and to obtain a detailed description of the flow using a large number of fluid particles when interacting with the objects. Note that 4M particles are simulated for a physical time of 9 seconds with *DualSPHysics* using the GPU card GTX480 as the execution device. The output frequency in the numerical model is 0.035 seconds so 257 output files have been saved with the state of the particles. The execution times are round 9 hours using the GPU architecture, while using a CPU device the simulation could take days. Table 4.3 shows the different SPH parameters used in these numerical simulations.

Table 4.3. Numerical parameters of the application.

Parameter	Value
Particle size dp (m)	0.026
Smoothing length h (m)	0.039
Number of particles np (m)	3,969,726
Artificial viscosity α in Eq. 2.15	0.25
Speed of sound c_o in Eq. 2.26	$20\cdot V_{max}$
CFL number C in Eq. 2.41	0.2
Physical time simulated (s)	8

Here it will be represented the interaction of different waves with the coastal structure to analyse the different responses in terms of forces exerted on the objects. The displacement of the piston is defined following a sinusoidal movement described by the expression:

$$x_w(t) = A \cdot \sin(2\pi f t + \theta) \tag{4.2}$$

where x_w is the position of a SPH particle that belongs to the wavemaker, A is the amplitude, f the frequency and θ the initial phase.

Thereby, four different simulations are carried out keeping the same initial configuration and the same number of particles but generating four different waves with different frequency and amplitude values as shown in Table 4.4.

Table 4.4. Calibrations of the wavemaker to generate different waves

wave	frequency (Hz)	amplitude (m)	Wave length (m)
1	0.12	1.8	36.211
2	0.12	2.0	36.211
3	0.14	1.8	30.816
4	0.14	2.0	30.816

Fig. 4.8 shows the instant *t=5.18s* of the four different simulations. A visual comparison of the same instant for the four waves suggests that wave #4 is the highest and fastest one. However, wave elevation must be numerically measured to identify the properties of the incoming waves.

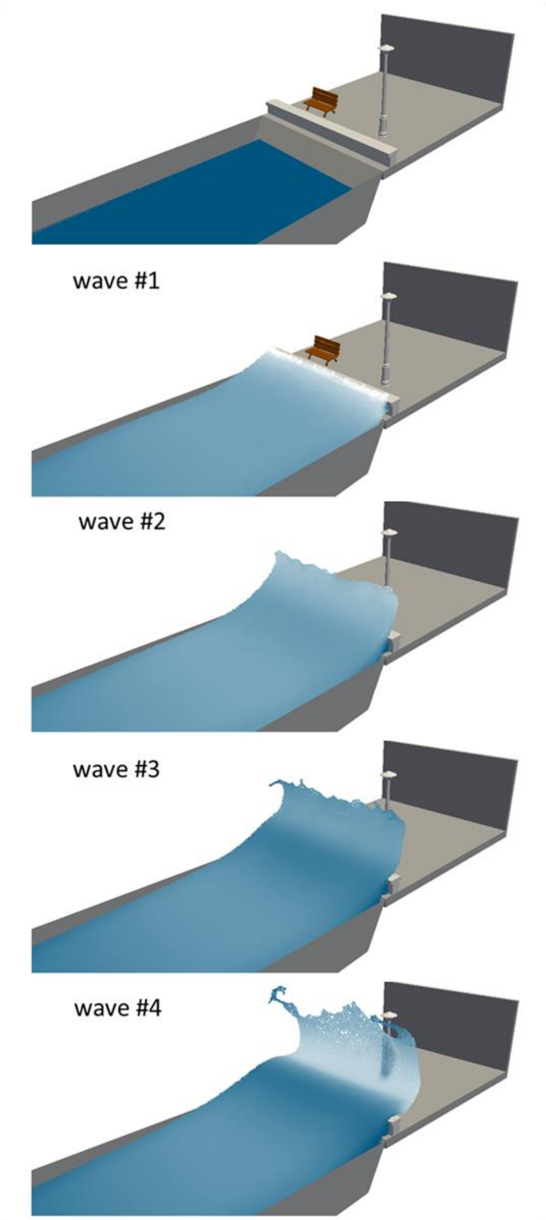


Figure 4.8. Instant (t=5.18s) of the impact of the four waves against the promenade using *DualSPHysics.*

4.3.2 Wave surface elevation

The time series of the numerical values of water surface elevation (η) computed for the different waves are depicted in Fig. 4.9. The values shown in the figure correspond to the maximum water elevation for each time step. These values were measured by a series of numerical probes located in the area of wave generation, first 12 meters of the tank.

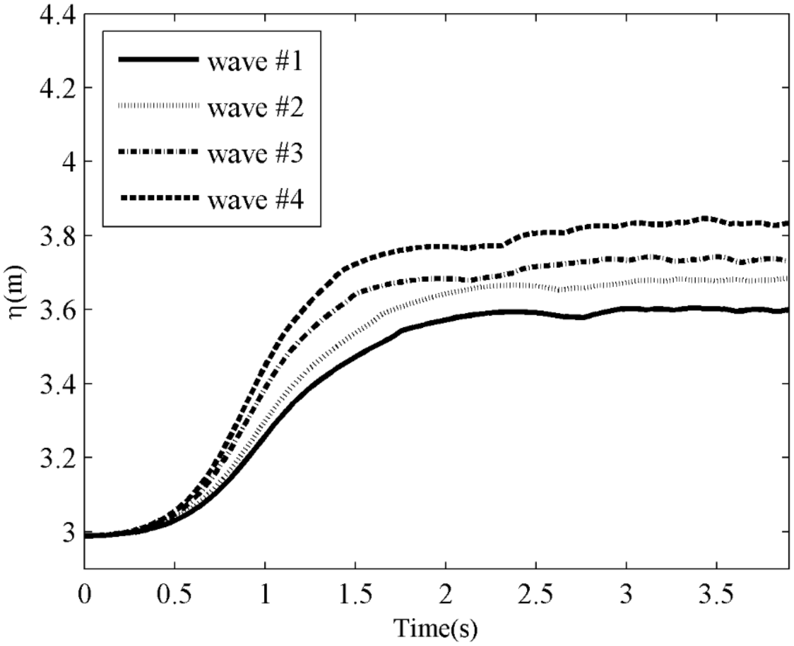


Figure 4.9. Time series of the numerical wave elevation of the four propagated waves.

It can be now concluded that wave #4 is, indeed, the highest in celerity and height of the four waves since its maximum peak value is the highest one and was registered before. Thus, the largest waves are wave #4 (created with the highest amplitude and frequency values), #3, #2 and #1 (created with the lowest amplitude and frequency values), in that order.

4.3.3 Forces exerted on the balustrade

Apart from the wave elevation, some other variables of interest can be provided by *DualSPHysics* code to characterise the interaction between the fluid and the urban furniture of the seawalk. One of them is the numerical force exerted on the balustrade. As mentioned before, the resulting forces of the interaction between the boundary particles of the balustrade and the surrounding fluid particles are accumulated. Thus, the time series of the force exerted by the four simulated waves on the balustrade along the simulation are shown in Fig. 4.10.

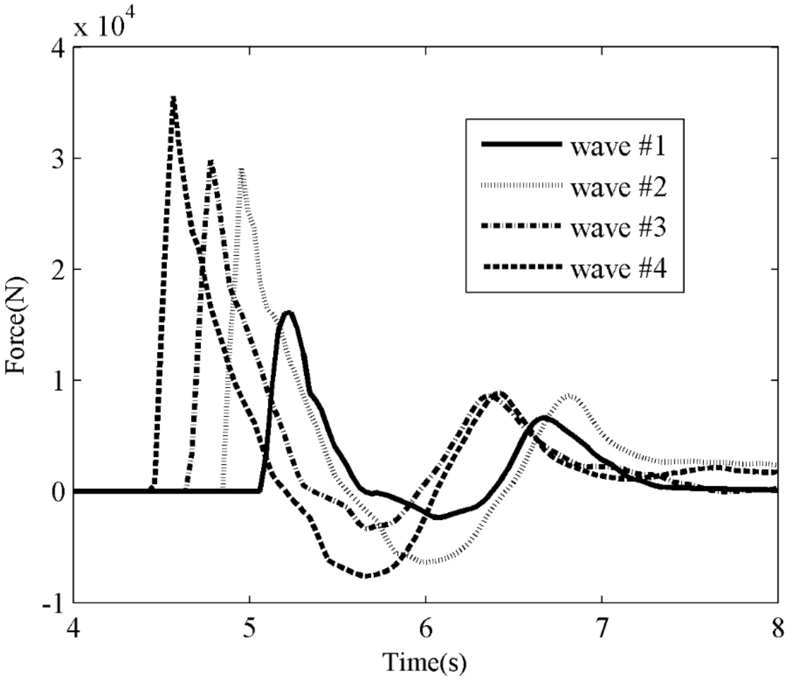


Figure 4.10. Time series of the numerical force exerted in x direction by the four waves on the balustrade.

During the first 4.4 seconds of simulation, no values of force are measured since the waves have not reached the location of the balustrade. The values are no longer zero when the waves arrive at the seawalk, first the fastest (and highest) and last the slowest (and lowest), i.e. wave #4, then #3, #2 and finally wave #1.

Starting from *t=5.2s*, negative values of force are observed in the figure. The forces shown in Fig. 4.10 are the result of adding positive values and negative ones. Next figures show separately the positive values (Fig. 4.11) and the negative ones (Fig. 4.12) of the force exerted on the balustrade. Positive values are considered in the direction of X^+ that corresponds to the direction of wave propagation (from left to right in Fig. 4.7) while forces exerted in the direction of X^- that corresponds to the opposite direction of wave propagation (from right to left in Fig. 4.7) are computed as negative values.

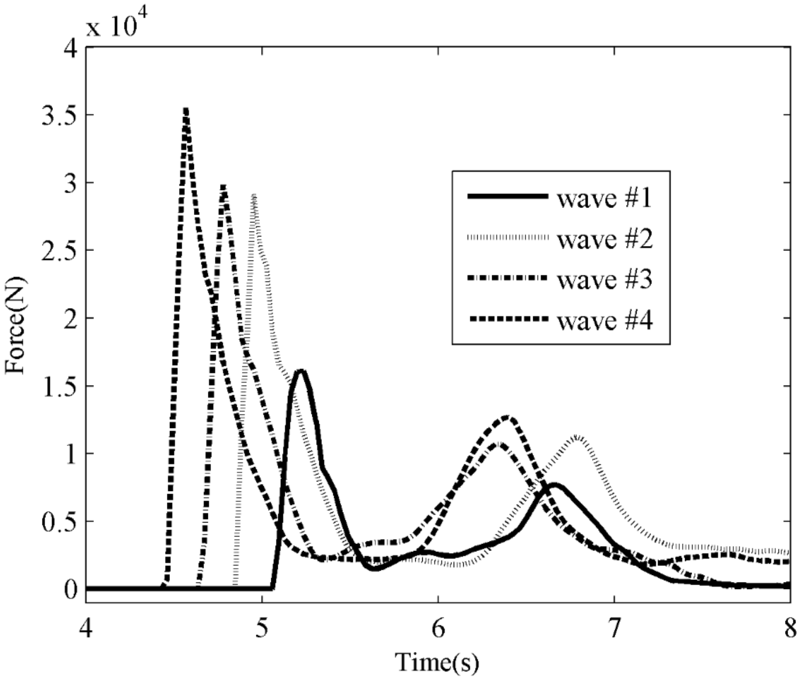


Figure 4.11. Time series of the numerical force exerted by the four waves on the balustrade in the direction of the wave propagation.

The negative values observed in Fig. 4.10 are due to the fact that the wave, after surpassing the balustrade, impacts with the ground generating a splash. The impact projects water in both directions, in the direction of advance of the wave, but also in the opposite direction impacting with the landward face of the balustrade. As it can observed in Fig. 4.12, the highest negative values are registered for wave #4 since the volume of water that impacts the landward face of the balustrade is higher.

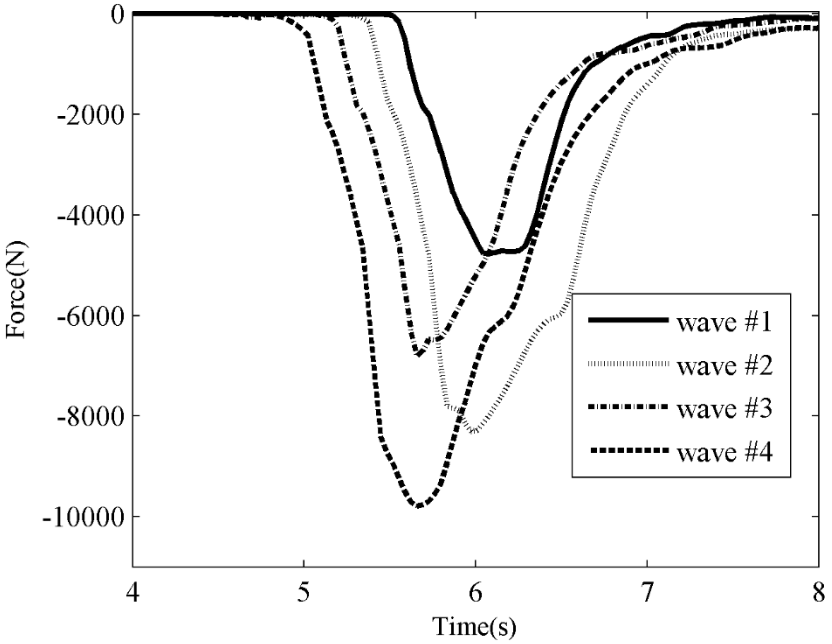


Figure 4.12. Time series of the numerical force exerted by the four waves on the balustrade in the direction opposite to the wave propagation.

4.3.4 Forces exerted on the lamppost

Another object of the urban furniture on which forces exerted by the waves have been computed is the realistic lamppost. To compute numerically the forces, the same procedure as in the case of forces exerted on the balustrade is followed, so the interaction forces registered at the boundary particles that form the lamppost are accumulated and represented over time (see Fig. 4.13).

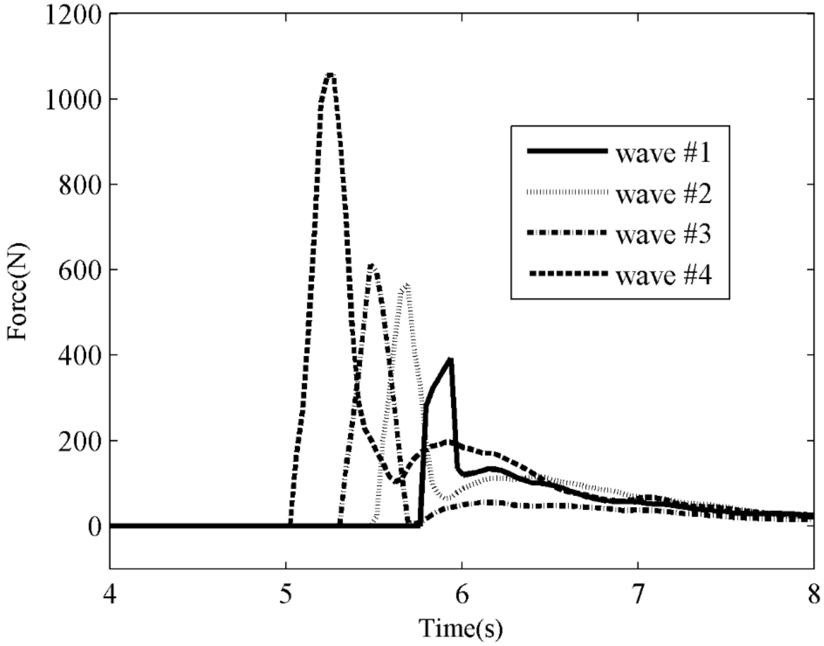


Figure 4.13. Time series of the numerical force exerted by the four waves on the lamppost.

As in the case of the balustrade, the force values shown in Fig. 4.13 are no longer zero when the waves start to interact with the lamppost. The order is also the same as in the previous case, the first wave to arrive at the lamp is wave #4, then the wave #3, #2 and finally the slower wave, the wave #1. The maximum forces are also registered following the same order.

Fig. 4.14 shows the instant *t=5.11s*, just before the impact of the wave #4 with the lamppost. The gaps between waves and objects are observed in the figure because of the mix in visualization; using particles for the water waves and a schematic outline for the promenade, balustrade and lamppost.

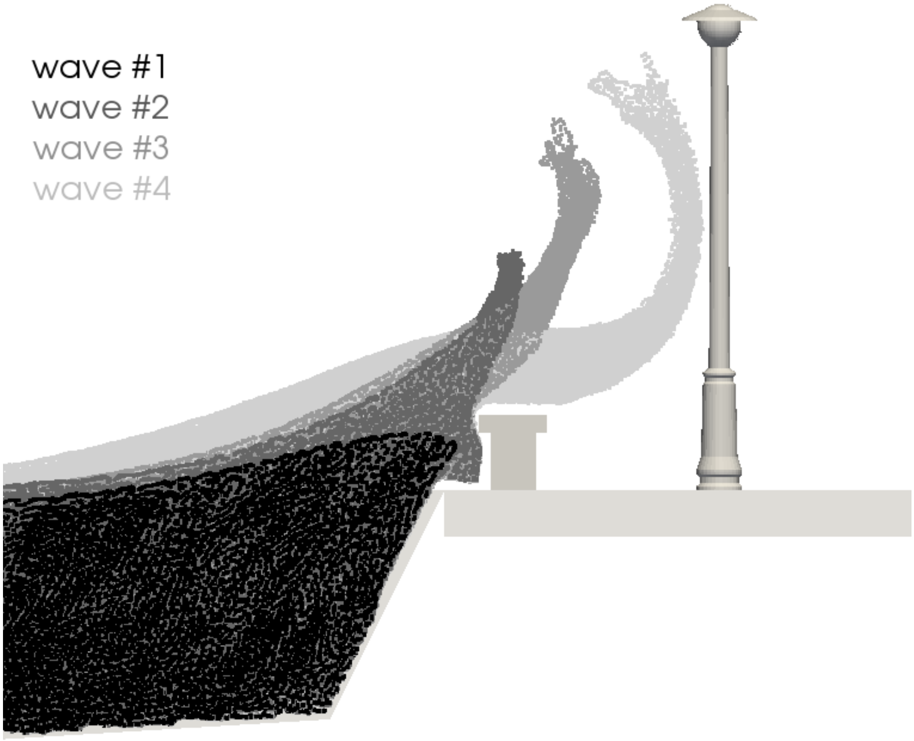


Figure 4.14. Instant (t=5.11s) of the impact of the four waves against the balustrade and lamppost using *DualSPHysics*.

The interaction between the waves and the lamppost can be characterized in terms of the force, but also in terms of the moment. The moment indicates the intensity of the impact and the impact points that are also very relevant in the case of interaction with vertical objects such as the lamppost. Fig. 4.15 shows the moment exerted by the four waves. The maximum values of moment also decrease for the lower waves.

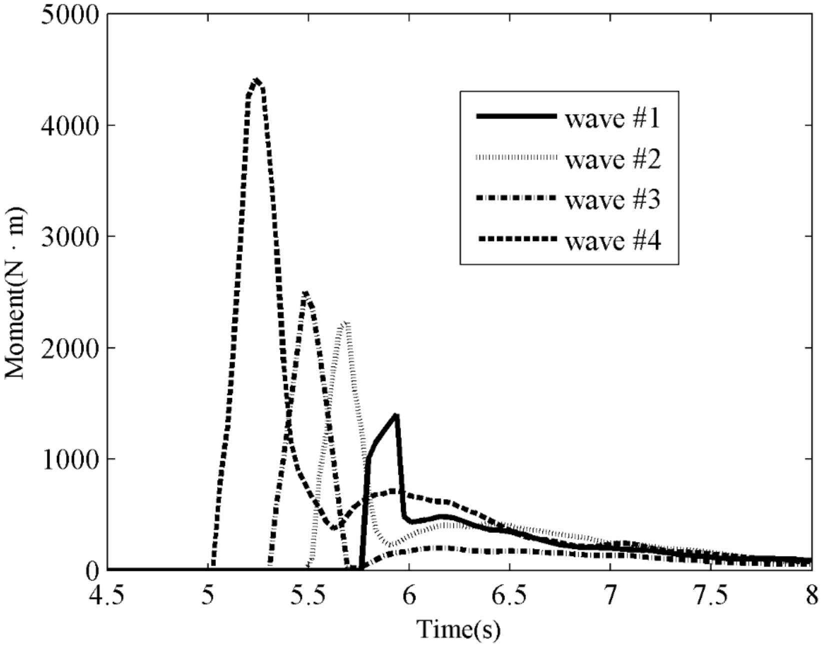


Figure 4.15. Time series of the numerical moment exerted by the four waves on the lamppost.

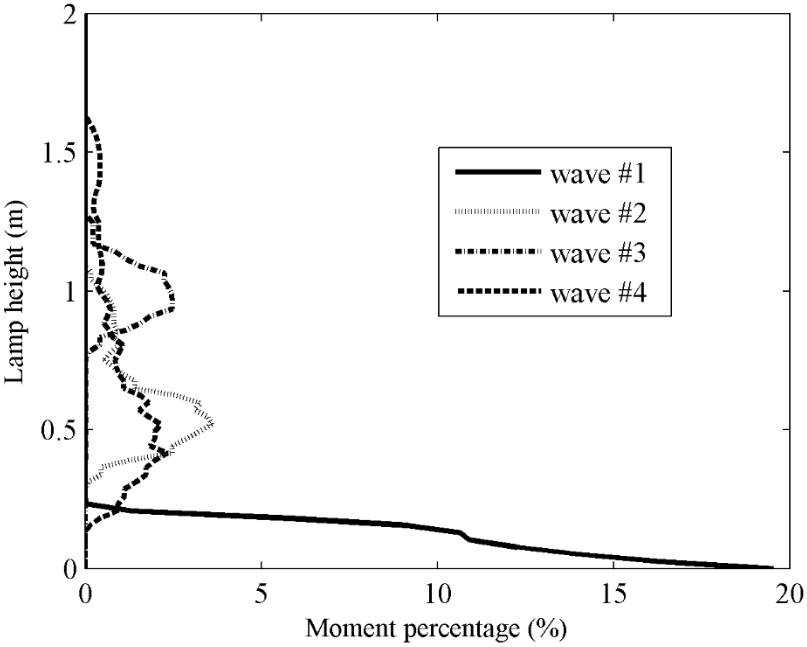


Figure 4.16. Distribution along the vertical positions of the moment exerted by the four waves on the lamppost.

The percentage of moment distributed along the vertical positions of the lamppost is depicted in Fig. 4.16. In this way, it can be seen if the impact has been localized in only one point or if it is more distributed along the vertical.

The impact of the wave #1 is more localized in the bottom of the lamppost, wave #2 hits the lamppost between heights 0.4m and 1.0m and wave #3 between 0.8 m and 1.2 m. However fluid particles of wave #4 (the highest) do not collide in only one range of heights but interact with the lamp at different points of the vertical.

5. *Case II: Floatings & Moorings*

Each year more energy resources are exploited to meet the needs of current society. These new energy sources are mainly located at sea, such as oil & gas, off-shore wind farms, and wave/tide energy devices. Off-shore applications present a new field of study in engineering. These applications have demanded new functionalities that were implemented in *DualSPHysics* are presented in this chapter. On the one hand the buoyancy-driven motion of floating bodies is studied, so that Newton's equations for rigid body dynamics are described and validation is presented. On the other hand, a new implementation to simulate the behaviour of moored lines with *DualSPHysics* is presented in this chapter. This new approach allows reproducing the forces on floating bodies, such as vessels, boats, wave energy devices and other off-shore structures moored to the seabed.

5.1 Fluid Driven objects

Not all the bodies and boundaries in *DualSPHysics* are fixed or have an imposed movement. It is also possible to derive the movement of an object by considering its interaction with fluid particles and using these forces to drive its motion. This can be achieved by summing the force contributions for an entire body. By assuming that the body is rigid, the net force on each boundary particle is computed according to the sum of the contributions of all surrounding fluid particles according to the designated *kernel* function and smoothing length. Each boundary particle *k* therefore experiences a force per unit mass given by

$$\boldsymbol{f}_k = \sum_{a \in WPs} \boldsymbol{f}_{ka} \tag{5.1}$$

where $\boldsymbol{f}_{ka}$ is the force per unit mass exerted by the fluid particle *a* on the boundary particle *k*, which is given by

$$m_k \boldsymbol{f}_{ka} = -m_a \boldsymbol{f}_{ak} \tag{5.2}$$

For the motion of the moving body, the basic equations of rigid body dynamics can then be used

$$M \frac{d\boldsymbol{V}}{dt} = \sum_{k \in BPs} m_k \boldsymbol{f}_k \tag{5.3a}$$

$$I \frac{d\boldsymbol{\Omega}}{dt} = \sum_{k \in BPs} m_k \left(\boldsymbol{r}_k - \boldsymbol{R}_0\right) \times \boldsymbol{f}_k \tag{5.3b}$$

where M is the mass of the object, I the moment of inertia, $\boldsymbol{V}$ the velocity, $\boldsymbol{\Omega}$ the rotational velocity and $\boldsymbol{R}_0$ the centre of mass. Eq. 5.3a and 5.3b are integrated in time in order to predict the values of $\boldsymbol{V}$ and $\boldsymbol{\Omega}$ for the beginning of the next time step. Each boundary particle within the body then has a velocity given by

$$\boldsymbol{u}_k = \boldsymbol{V} + \boldsymbol{\Omega} \times (\boldsymbol{r}_k - \boldsymbol{R}_0) \qquad (5.4)$$

Finally, the boundary particles within the rigid body are moved by integrating Eq. 5.4 in time. The work of Monaghan (2005) and Monaghan et al., (2003) show that this technique conserves both linear and angular momentum.

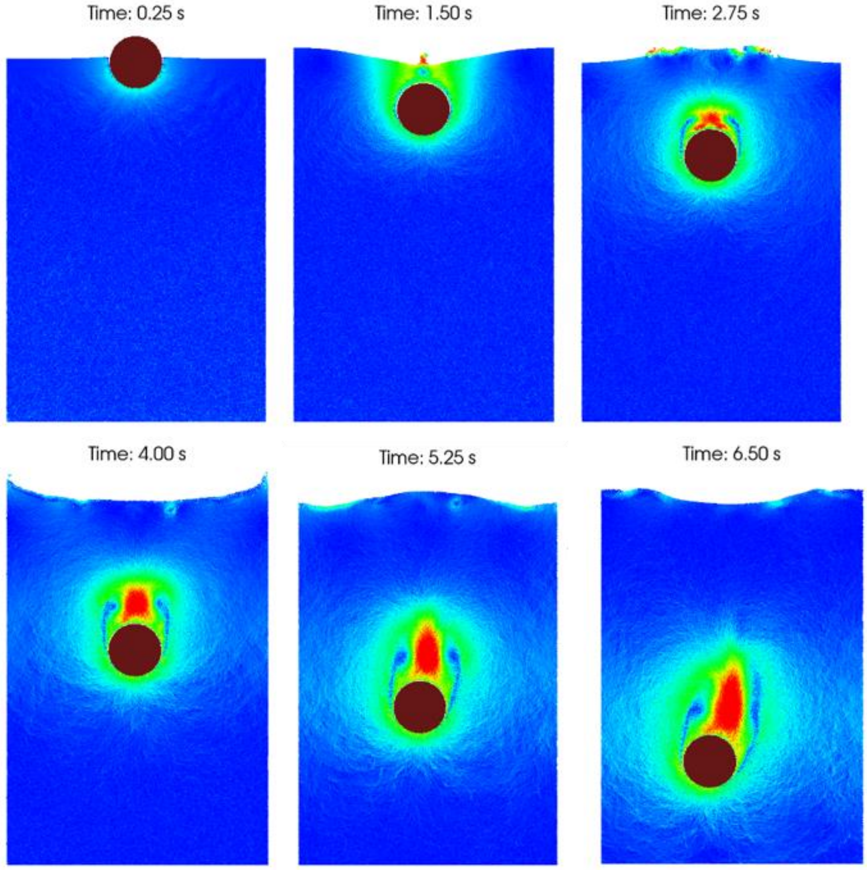


Figure 5.1. Different instants of the sinking cylinder with ρ=1,200 Kg/m^3.

A validation case is now presented to prove the correct buoyancy-driven motion of a floating cylinder. *DualSPHysics* simulates the 2-D system of a sinking cylinder with a r=1 m radius and density ρ=1,200 Kg/m^3 that is half submerged. An initial interparticle distance dp=0.03 m is used leading to 157,717 particles. Figure 5.1 shows the case at different instants and the colour of the particles corresponds to the velocity magnitude.

Fekken (2004) compared these results with those of a numerical model based on a Volume of Fluid (VOF) discretization technique. Figure 5.2 plots the vertical displacement and velocity of the cylinder. The SPH results are close to the VOF solution, particularly in the initial instants.

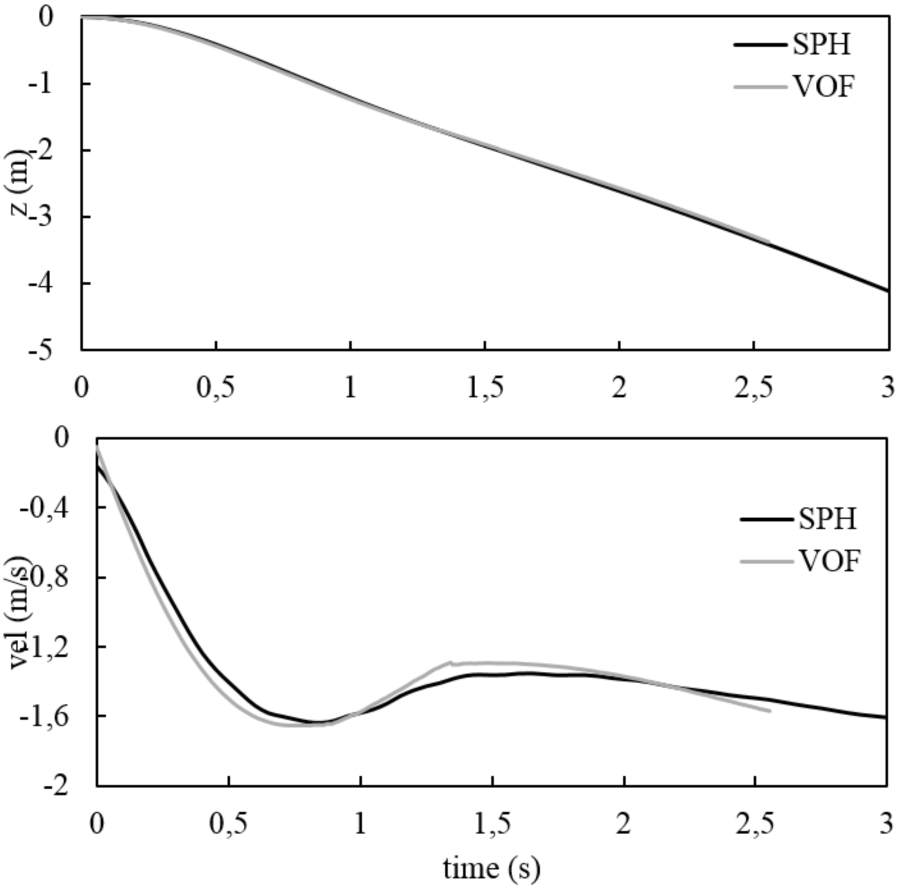


Figure 5.2. Comparison of numerical data of Fekken (2004) and *DualSPHysics* results for displacement of the sinking cylinder (top panel) and vertical velocity (bottom panel).

More validations are performed in Canelas et al., 2015, where the model is tested for buoyancy-driven motion with solid objects larger than the smallest flow scales and with various densities. Simulations are compared with analytical solutions, other numerical methods (Fekken, 2004) and experimental measurements.

5.2 *Moored Lines*

Moorings are defined as the lines that hold in position a vessel or any kind of floating object. More precisely, this implementation is focused on continuous ropes and wires. The shape of this kind of lines, hanging between two points, can be described by the catenary function $y = a\cosh\left(\frac{x}{a}\right)$ where y and x are vertical and horizontal coordinates respectively, and a is a parameter that defines the curvature of the line.

5.2.1 Static approach

Fig. 5.3 presents an element of a mooring line and all the forces acting on it.

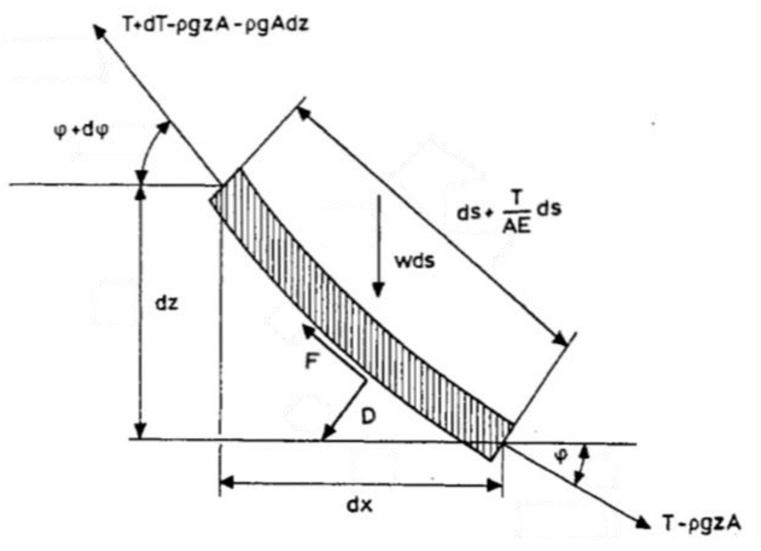


Figure 5.3. Diagram of the different forces acting on an element of a mooring line.

The different quantities shown in Fig. 5.3 need to be defined: T is the line tension, A is the cross-section area of the line, E represents the elasticity modulus, *F* and *D* correspond to the mean hydrodynamic forces both normal and tangential direction respectively, and ω is the submerged weight per unit length.

Following the work of Faltinsen (1993) the maths behind the implementation are.

$$\begin{aligned} dT - \rho g A\, dz &= \left[\omega\, sin\, \phi - F\left(1 + {}^{T}\!/_{(AE)}\right)\right] ds \\ T\, d\phi - \rho g A\, d\phi &= \left[\omega\, cos\, \phi + D\left(1 + {}^{T}\!/_{(AE)}\right)\right] ds \end{aligned} \qquad (5.5)$$

These equations are non-linear, and do not have an explicit solution. To simplify the analysis some approximations were taken: neglecting the effect of the current forces (*D* and *F* both equal to 0), the chain has no elasticity (*E*=0), and the weight per unit of length remain constant in the entire chain (ω constant).

Thus, the solution for the inelastic catenary equations is presented. First, it is important to make some assumptions:

by introducing

$$T' = T - \rho g z A \qquad (5.6)$$

it can be written

$$\mathrm{d}T' = \omega \sin\phi\, \mathrm{d}s \qquad (5.7)$$

$$T'\mathrm{d}\phi = \omega \cos\phi\, \mathrm{d}s \qquad (5.8)$$

operating these two equations

$$\frac{\mathrm{d}T'}{T'} = \frac{\sin\phi}{\cos\phi}\mathrm{d}\phi \qquad (5.9)$$

that is

$$T' = T_0' \frac{\cos \phi_0}{\cos \phi} \tag{5.10}$$

by integrating $T' \mathrm{d}\phi = \omega \cos \phi \, \mathrm{d}s$

$$s - s_0 = \frac{1}{\omega} \int_{\phi_0}^{\phi} \frac{T_0'}{\cos \theta} \frac{\cos \phi_0}{\cos \theta} \mathrm{d}\theta = \frac{T_0' \cos \phi_0}{\omega} [\tan \phi - \tan \phi_0] \tag{5.11}$$

being dx=cos ϕ ds

$$\begin{aligned} x - x_0 &= \frac{1}{\omega} \int_{\phi_0}^{\phi} \frac{T_0' \cos \phi_0}{\cos \theta} \mathrm{d}\theta \\ &= \frac{T_0' \cos \phi_0}{\omega} \left(\log \left(\frac{1}{\cos \phi} + \tan \phi \right) \right. \\ &\left. - \log \left(\frac{1}{\cos \phi_0} + \tan \phi_0 \right) \right) \end{aligned} \tag{5.12}$$

also it is known that dz=sin ϕ ds

$$\begin{aligned} z - z_0 &= \frac{1}{\omega} \int_{\phi_0}^{\phi} \frac{T_0' \cos \phi_0 \sin \theta}{\cos^2 \theta} \mathrm{d}\theta \\ &= \frac{T_0' \cos \phi_0}{\omega} \left[\frac{1}{\cos \phi} - \frac{1}{\cos \phi_0} \right] \end{aligned} \tag{5.13}$$

ϕ_0 is defined as the contact point of the moored line with the seabed, ($\phi_0 = 0$) so Eq. 5.11 can now be rewritten as

$$T_0' = T' \cos \phi \tag{5.14}$$

the horizontal tension in the water surface can be written as

$$T_H = T \cos \phi_w \tag{5.15}$$

being ϕ_w the angle of the moored chain with the water surface. By merging equations 5.6, 5.14 and 5.15

$$T_0' = T_H \tag{5.16}$$

This equation is consistent with the global summation of forces. Other assumptions were, $x_0 = 0$ and $z_0 = -h$. In addition, $s_0 = 0$ → $\varphi_0 = 0$, the angle ϕ_0 can be eliminated from Eqs. 5.11, 5.13 by Eq. 5.12 which can be written as

$$\frac{x\omega}{T_H} = \log\left(\frac{1 + \sin\phi}{\cos\phi}\right) \tag{5.17}$$

that is

$$\begin{aligned} \sinh\left(\frac{x\omega}{T_H}\right) &= \frac{1}{2}\left(\frac{1+\sin\phi}{\cos\phi} - \frac{\cos\phi}{1+\sin\phi}\right) = \tan\phi \\ \cosh\left(\frac{x\omega}{T_H}\right) &= \frac{1}{2}\left(\frac{1+\sin\phi}{\cos\phi} - \frac{\cos\phi}{1+\sin\phi}\right) = \frac{1}{\cos\phi} \end{aligned} \tag{5.18}$$

previous expressions can be now written as

$$s = \frac{T_H}{\omega}\sinh\left(\frac{\omega}{T_H}x\right) \tag{5.19}$$

$$z + h = \frac{T_H}{\omega}\left[\cosh\left(\frac{\omega}{T_H}x\right) - 1\right] \tag{5.20}$$

so the line tension can be found as de combination of Eqs. 5.6, 5.13 and 5.14.

$$T - \rho g z A = \frac{T_H}{\cos\phi} = T_H + \omega(z + h) \tag{5.21}$$

i.e.

$$T = T_H + \omega h + (\omega + \rho g A) z \tag{5.22}$$

in order to find the vertical tension the following expression is used

$$\begin{aligned} \mathrm{d}T_z' &= \mathrm{d}(T' \sin\phi) = \mathrm{d}T' \sin\phi + T' \cos\phi \,\mathrm{d}\phi \\ &= \omega \sin^2\phi \,\mathrm{d}s + \omega \cos^2\phi \,\mathrm{d}s \end{aligned} \tag{5.23}$$

this translates in $T_z' = \omega s$, and at the water surface is

$$T_z = \omega s \tag{5.24}$$

In order to simulate properly the moorings, a function is needed to determine the horizontal tension (T_H) generated by the moored line. This function will depend on the relative position of the link point in the floating body and the anchoring point in the sea bed. In Fig. 5.4 parameters to define a moored line are presented.

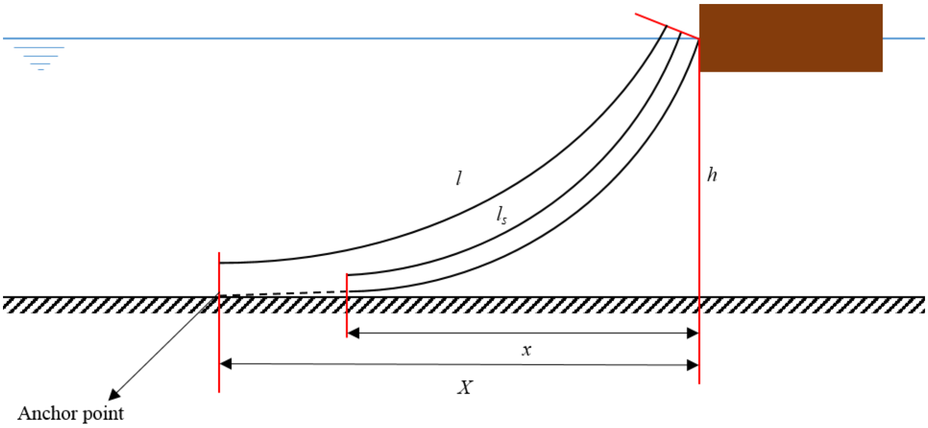


Figure 5.4. Parameters that define a mooring line.

Defining

$$a = \frac{T_H}{\omega} \tag{5.25}$$

Eqs. 5.19 and 5.20 are rewritten as

$$l_s = a \sinh\left(\frac{x}{a}\right) \tag{5.26}$$

$$h = a\left[\cosh\left(\frac{x}{a}\right) - 1\right] \tag{5.27}$$

combining these two equations

$$l_s^2 = h^2 + 2ha \tag{5.28}$$

According to Fig.5.2 horizontal distance can be written as

$$X = l - l_s + x \tag{5.29}$$

where l_s is expressed as shown in Eq. 5.24, and Eq. 5.26 is used to express x.

$$X = l - h\left(1 + 2\frac{a}{h}\right)^{\frac{1}{2}} + a\cosh^{-1}\left(1 + \frac{h}{a}\right) \tag{5.30}$$

the relationship between X and T_H is obtained as $a = {}^{T_H}/_{\omega}$.

Fig. 5.5 presents the horizontal tension as function of the position for two lines using Eq. 5.30. These lines have the following configurations:

- Line 1, ω=850 N/m, h=20 m and l=100 m.
- Line 2, ω=1000 N/m, h=20 m and l=100 m.

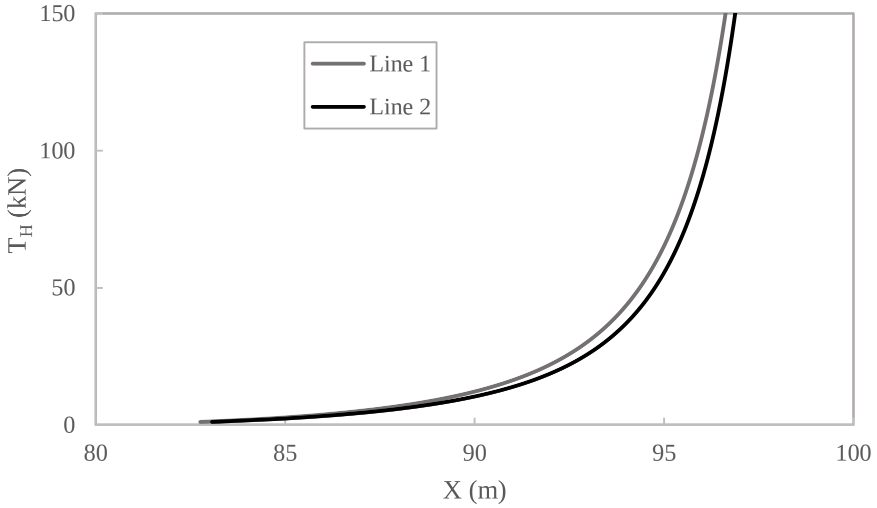


Figure 5.5. Horizontal tension for two different line configuration.

Now a *spread mooring system* is described. Usually vessels, ships, and other floating devices are not hold in position by a single line. The implementation can be generalized for multiple lines. Thus, the total effect of multiple lines will be the summation of every one of them separately acting on the same body. Forces could not be aligned with the Cartesian axis as Fig 5.6 shows. All the forces affecting a floating body must be referred to the Cartesian coordinate system to calculate the movement.

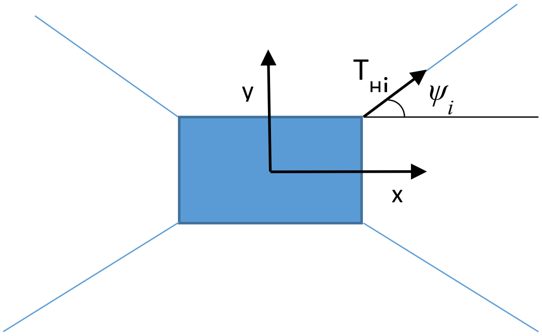


Figure 5.6. Zenith view of a spread mooring system on a vessel

Since both link and anchor point are known, the direction of the moored lines, and T_{Hi} projection are easy to find. Therefore the forces in x and y axis can be written as $F_x = T_{Hi}\, cos\, \Psi_i$ and $F_y = T_{Hi}\, sin\, \Psi_i$, these forces will be applied to the link point in the body. The floating body will be moving as the summation of buoyancy force, wave trust forces, and the forces of all the moored lines. The force of the mooring lines must be properly applied to mimic the pulls and spins that the cables can cause.

5.2.2 Mooring Classification

Normal conditions for moorings are already covered (panels *b* and *c* of Fig. 5.7) but there are two different states of the moored chain not covered yet. These two states are the resting chain, and the fully extended chain (panels *a* and *d* of Fig 5.7)

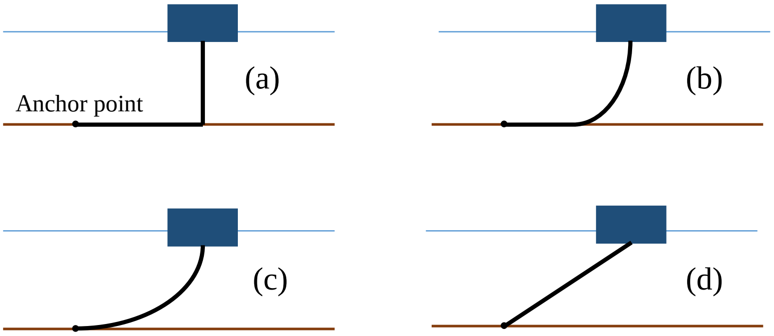


Figure 5.7. Different states of a moored line, (a) totally resting, (b) partially lifted, (c) totally lifted, and (d) totally extended.

In the non-extended case (panel *a* in Fig. 5.7), the horizontal force will be zero and the vertical force will be proportional to the mass of the chain hanging towards the sea bed.

The fourth case consists of the moored line totally extended (panel *d* of Fig. 5.7). The main force exerted by the chain is a reaction force, this force is equal to the main force of the floating object but in an opposite direction. The reaction force will have the chain direction and the modulus of the projection in the direction of the chain. This force will keep the body from going any further from the anchoring point but will not prevent spins and rotations around the anchoring point.

This is simple for only one moored line but for a set of moored lines in a totally extended disposition is not straight forward. In order to solve properly this situation, the summation of the forces generated by all the fully extended moored lines must cancel the forces acting on the body. The projections and angles must be defined in the Cartesian coordinate system as can be seen in Fig. 5.8. A new set of equations must be solved (5.31).

$$\begin{gathered}
\boldsymbol{F}_T = \boldsymbol{F}_1 + \boldsymbol{F}_2 + \boldsymbol{F}_3 + \boldsymbol{F}_4 \cdots \\
F_{Tx} = F_T \cdot \cos\theta_T \cdot \cos\varphi_T \\
F_{Ty} = F_T \cdot \cos\theta_T \cdot \sin\varphi_T \\
F_{Tz} = F_T \cdot \sin\theta_T \\
Force\ for\ an\ i\ mooring \begin{cases} F_{ix} = \dfrac{F_{Tx} \cdot \cos\varphi_i}{\sum_j^n \cos\varphi_j} \\ F_{iy} = \dfrac{F_{Ty} \cdot \sin\varphi_i}{\sum_j^n \sin\varphi_j} \\ F_{iz} = \dfrac{F_{Tz} \cdot \sin\theta_i}{\sum_j^n \sin\theta_j} \end{cases}
\end{gathered} \tag{5.31}$$

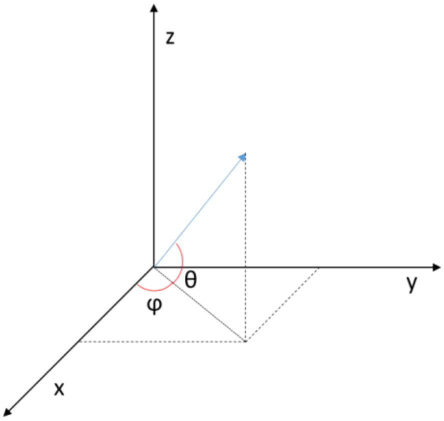


Figure 5.8. Definition of angles and projections.

Once solved, the formulation for the moored lines is complete and covers all the range of cases that can be found both in normal conditions, and most extreme cases.

5.2.3 Model Validation

In this section three different validations are presented in order to verify the implementation carried out in *DualSPHysics*. These validations are taken from Johanning et al. (2006) and Johanning et al. (2007) and provide different scenarios, different scales and multiple line problems.

a) *Validation of one line problems*

Experimental data from Johanning et al. (2007) is used to validate the one line case. This publication provides data for a single moored line in stationary conditions. The experimental set-up is described in Johanning et al. (2007) but main parameters of the moored line are summarized in Table 5.1. The experiment consists of a body with a line attached. The body is moved to sixteen different positions in the water surface, and a charge cell provides the value of the tension for the different positions.

Table 5.1. Moored line configuration.

Parameter	Value
h	2.651 m
ω	1.036 N/m
l	6.98 m
Minimum extension	5.735 m
Maximum extension	6.367 m

In Fig 5.7 the results from Johanning et al, (2007) are compared with the results provided by *DualSPHysics*. The axial loading data comes from other numerical model presented in Johanning el al. (2006).

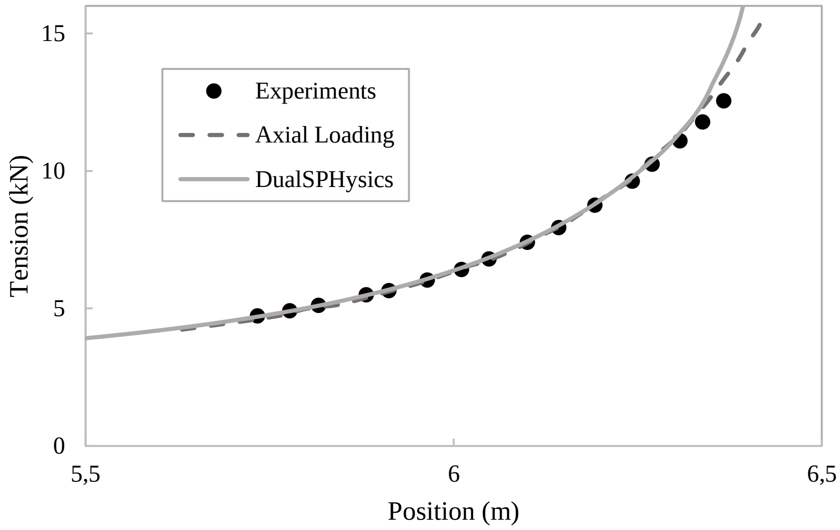


Figure 5.9. Comparison of experimental data, axial loading data and *DualSPHysics* results for horizontal tension

Numerical results are in good agreement with experimental data. A little divergence is observed and the point where the three lines begin to separate match with the point where the chain transitions from part resting in the bottom of the tank (panel *b* in Fig. 5.7) and totally lifted (panel *c* in Fig. 5.7).

Another single mooring line test validation was performed. The configuration of the moored line is presented in Johanning et al. (2006) and the line configuration is summarized in the Table 5.2. This new configuration has a completely different scale and this new scenario is closer to a real-life situation.

Table 5.2. Mooring configuration

Parameter	Value
h	50 m
ω	918.75 N/m
l	150 m
Minimum extension	102 m
Maximum extension	140 m

Fig. 5.10 shows the results obtained with *DualSPHysics* and the experimental data, both data are in a good agreement.

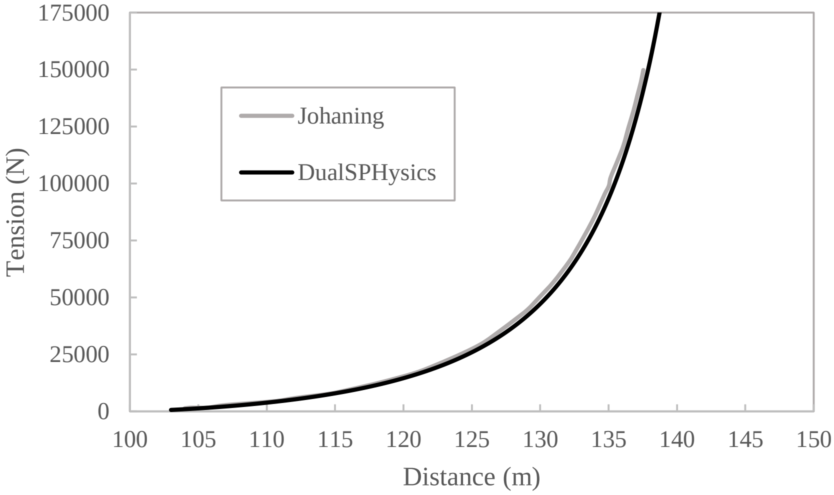


Figure 5.10. Comparison of experimental data and *DualSPHysics* results for the horizontal tension of a moored line.

b) *Validations of problems with two lines*

The last case of validation consists of two lines attached to a body. The configuration of the chains, the set-up, and the reference data is taken from Johanning et al. (2006), and summarized in Table 5.3.

Table 5.3. Chains configuration for two moored lines experiment

Parameter	Value
h	50 m
T_{H0}	50 kN
l	75 m
Minimum extension from resting point	-15 m
Maximum extension from resting point	15 m

Fig 5.11 compares experimental data with numerical results obtained with *DualSPHysics*. Results consist of the horizontal tension

for Line A and B, and the resultant tension exerted on the body when it is moved from equilibrium position towards the anchoring position of the Line B. Zero tension at x=0 represents the resting point (X=55m in the absolute scale). These two chains have the same physical properties. Line A is moored in X=0 m and Line B is moored in X=110 m. Once again, a good agreement with agreement with the data from Johanning et al. (2006) is achieved with *DualSPHysics*.

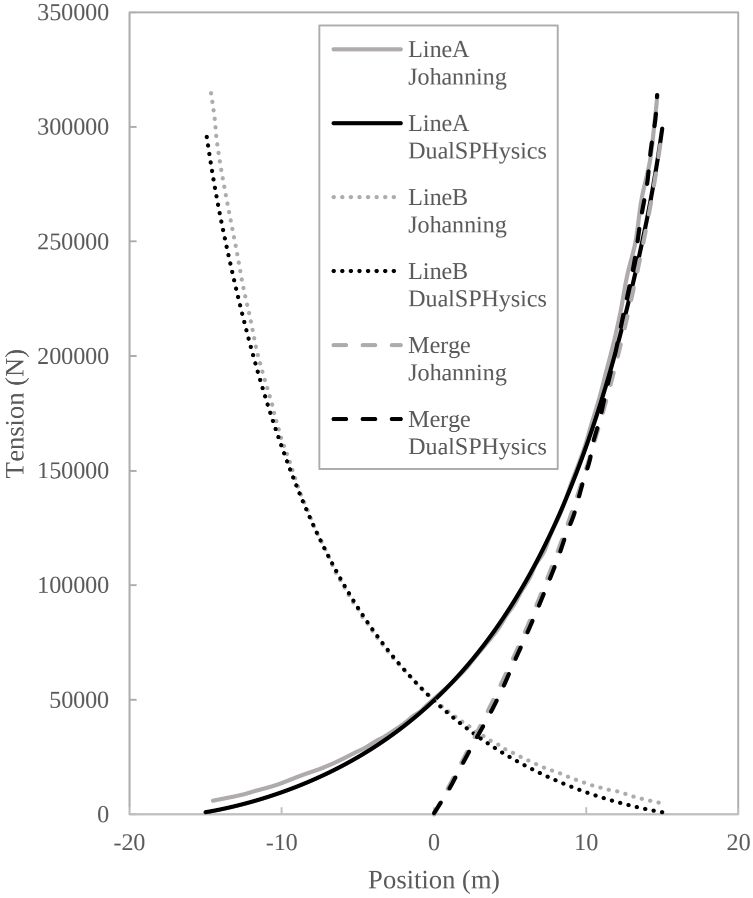


Figure 5.11. Comparison of experimental data with numerical results from *DualSPHysics*; horizontal tension for Line A and B, and the resultant tension exerted on the body.

5.3 Application Cases

Two cases of application of this new implementation are presented now. The first one is a ship moored and interacting with large waves. The ship is moored at the bow, the parameters of the moored line are presented in Table 5.4. As it can be seen in Fig 5.12, the boat moves under the effect of the waves and rotates by the action of the moored line. After a while, the moored line reaches its maximum elongation so the bow is maintain in position and the stern of the boat is aligned with the bow following the direction of the incoming waves. This final new disposition of the boat exhibits less resistance to wave impacts.

Table 5.4. Mooring configuration.

Parameter	Value
Initial water depth	55 m
ω	442 Kg/m
break load	13887kN
l	160 m
Submerged density	7850 Kg m^{-3}

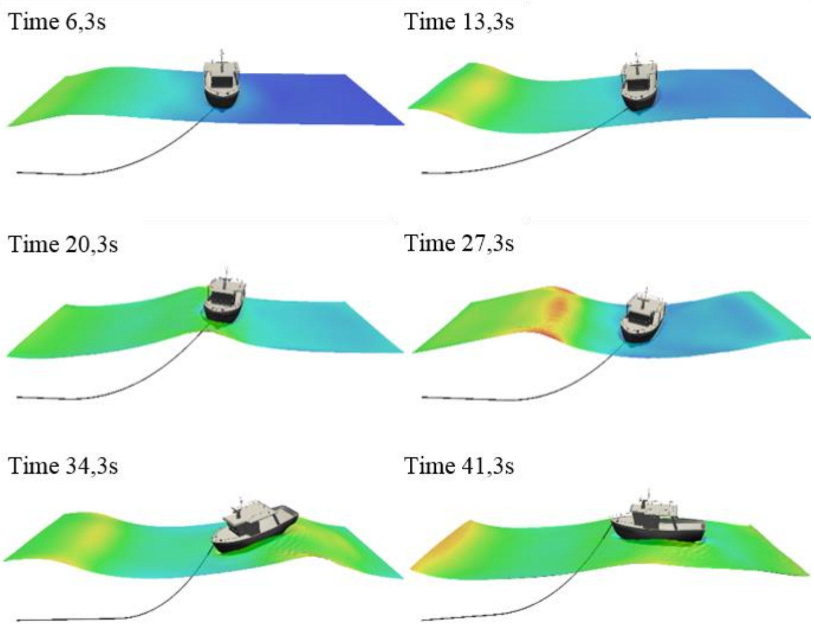


Figure 5.12. Different instants of a moored ship under the action of large waves

The second case represents a base of an off-shore wind turbine. This floating object is secured to the bottom of the sea by three moored lines. This configuration is common in situations where fixed or rigid structures will suffer great stresses due to wave action or currents. The floating object is exposed to extreme wave conditions forcing the moorings to work in the extreme regimes. The main parameters of the three moored lines are the same and presented in Table 5.5. Fig. 5.13 shows how the two front moored lines are totally extended (at 71.10s and 81.10s). This state matches panel *d* of Fig 5.7. This simulation demonstrates how the equations shown in section 5.1.2 solve properly an extreme case and the summation of both front mooring forces cancel properly the wave drag.

Table 5.5. Mooring lines configuration.

Parameter	Value
Initial water depth	20 m
ω	442 Kg/m
break load	1400 kN
l	45 m
Submerged density	7850 Kg m^{-3}

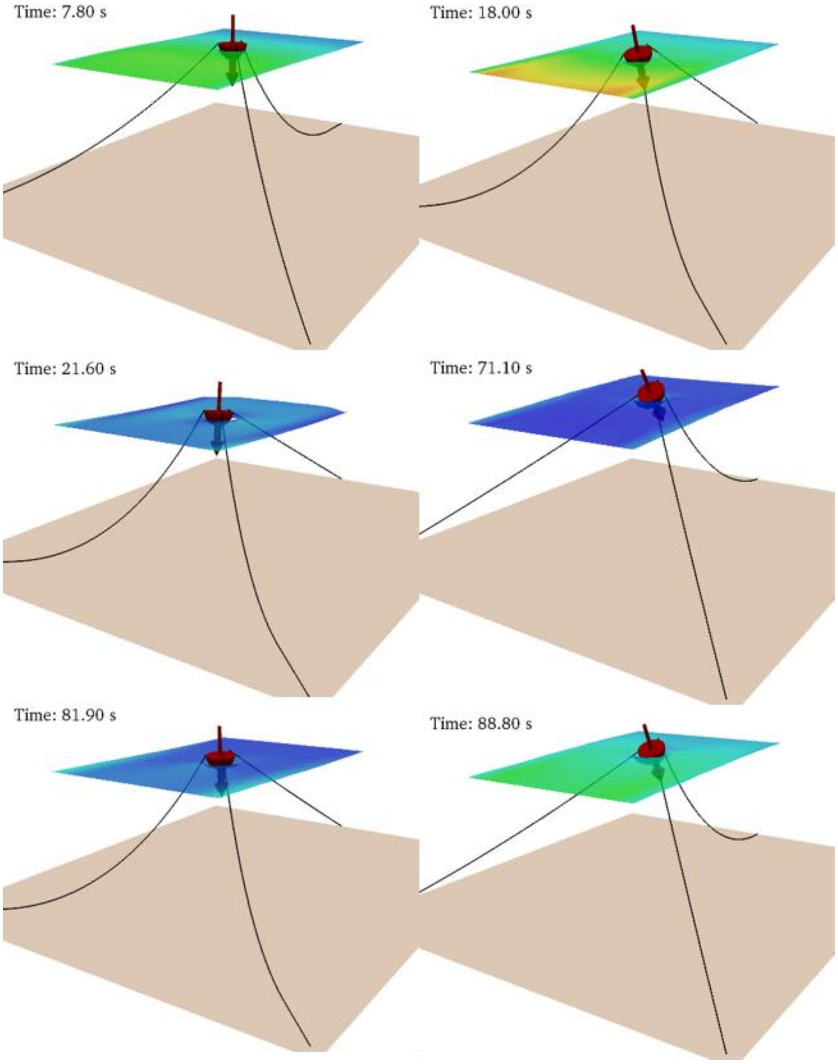


Figure 5.13. Different instants of an off-shore wind turbine base moored with 3 spread lines under the action of waves.

These last working examples proves that the implementation is capable of dealing with these kind of extreme conditions, and also is able to deal with multiple moorings in extreme states.

6. *Case III: Run-off on real terrains*

In this section, *DualSPHysics* is used in a civil engineering case. The code is applied to a real case to simulate extreme run-offs on a terrain whose geometry was obtained using UAV (Unmanned Aerial Vehicle) photogrammetry. The study case simulates how the collected water can flow into a close road and the countermeasures to decrease this kind of pernicious events. Intense rain events are simulated using a water inflow that mimics extreme meteorological conditions in the zone. The amount of water arriving at the road is calculated under different protection scenarios. The effect of a ditch to prevent water flow into the road is analysed. The importance of this case is the combination of two different technologies, UAV photogrammetry and *DualSPHysics*, to study a real civil engineering case. The road affected by run-off is located in 42.2946°N and 7.5888° W, northwest of the Iberian Peninsula. Run-off is a common issue in multiple locations and the methodology described here can be extrapolated to any place with similar problematic.

6.1 UAV photogrammetry

UAV (Unmanned Aerial Vehicle) photogrammetry is employed to obtain accurate topographic information of the zone. UAV systems are been successfully applied to the field of geomatics over the last few years. UAV platforms have become a new tool for remote sensing that can operate under circumstances where the traditional airborne platforms cannot, due mainly to cost, lack of flexibility or danger issues. When UAVs operate in combination with digital cameras, they become photogrammetric measurement platforms that can provide useful topographic data (Remondino et al., 2011). UAV photogrammetry represents a low-cost alternative to the classical manned aerial photogrammetry. In addition, UAVs can work at lower altitude than manned systems and it is possible to achieve very high resolution and accuracy with midrange photographic equipment (Everaerts, 2008).

The aerial inspection unit used for this work was an eight propeller UAV, the Okto XL from Mikrokopter. Data acquisition was first planned in the laboratory using Google Earth geo-referenced images and the Mikrokopter planning software to generate the GPS waypoints for the survey. This way points define the trajectory of the UAV and the place where an image is taken (Fig. 6.1). The waypoint positioning and image acquisition was done autonomously by the UAV. Photoscan photogrammetric software uses computer vision algorithms to the automated extraction of common points on UAV photographs. These points allow obtaining the camera calibration and relative orientation of the images. Bundle adjustment algorithm is used to refine the 3D coordinate calculation. The absolute orientation is completed with six topographic points taken with a total station from the road under study and its neighbourhood. At the end of the process a geo-referenced 3D point cloud with the geometric information of the area is obtained.

Hence, in Fig. 6.1, the place of the camera is shown sing a blue plane and the normal vector of the plane is presented with a black line.

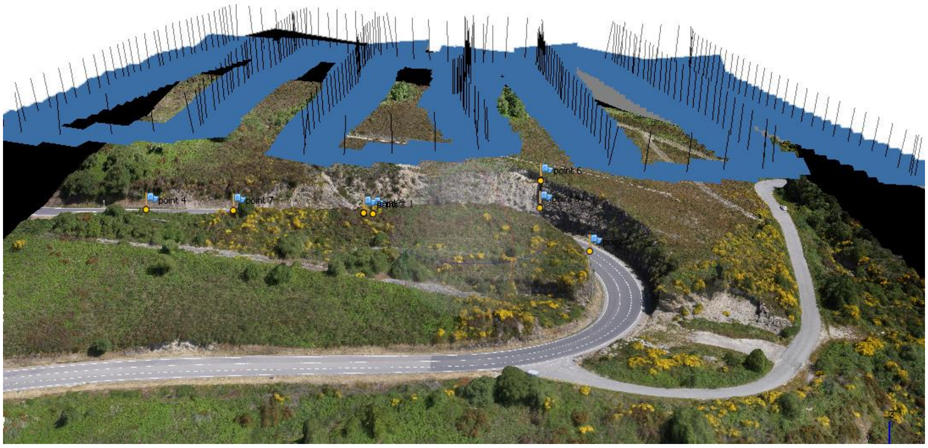


Figure 6.1. UAV images alignment with computed camera positions and a resulting sparse point cloud.

6.2 Data pre-processing

The photogrammetric data acquisition leads to a point cloud with accurate geometric information of the scene. It is necessary to convert that set of points into SPH particles, in particular, the geometry of a scenario will be converted into boundary SPH particles and the flow will be simulated using fluid SPH particles. The conversion of the point cloud into SPH particles is carried out by using surface geometries (described by triangles) as an intermediate step. Thereby, a conversion *POINTS to TRIANGLES* is firstly performed and *TRIANGLES to SPH particles* is finally carried out.

POINTS to TRIANGLES

The point cloud generated by Photoscan is triangulated to obtain a polygon mesh using 2D Delaunay triangulation. A mesh is a Delaunay triangulation if all circumcircles of all triangles of the network are empty. The mesh is colourised using the radiometric information from the images and the orientation of the camera. The colours can be useful to identify the type of terrain, materials, etc. The generation of the mesh is crucial because it closes holes on the geometry and allows the realistic water displacement without abnormal infiltration. Therefore, a STL file can be produced as seen in Fig. 6.2a (colourised) and Fig. 6.2b (no colourised). This file format describes a raw triangulated surface by the unit normal and vertices of the triangles using a 3D Cartesian coordinate system.

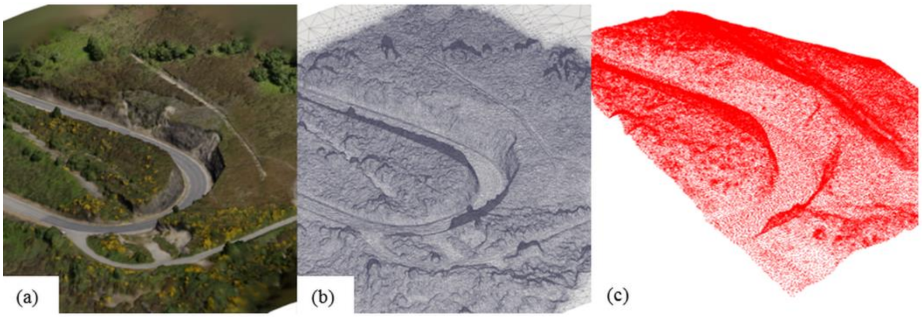


Figure 6.2. Digital Surface Model (DSM) of the test site in STL format (colourised (a) and no colourised (b)) and then converted into SPH particles (c).

TRIANGLES to SPH particles

The pre-processing tool employed in *DualSPHysics* uses a 3D mesh to locate the particles. The idea is to build any object using particles, so these particles are created in the nodes of the 3D mesh, so that the position of the particles is defined by the 3D mesh and depends on its resolution. An initial distance *dp* is defined to create the 3D mesh, so that the nodes of the mesh are equidistant. Firstly, the mesh nodes around the object are checked and then particles are created only in the nodes that compose the shape of the desired geometry.

For the case of triangles, only the nodes at a distance less than 0.6**dp* to one of the sides of the triangle are used to create particles. In this way, the accuracy of the spatial resolution depends on the mesh resolution. In addition, complex 3D models such as STL files can be imported, split into different triangles and each triangle can be converted into particles as described (Fig. 6.2c). More detailed information about the pre-processing can be found in Dominguez et al. (2011).

6.3 Case of study

The geometries shown in Fig. 6.1 and Fig. 6.2 correspond to the scenario of the case of study. More specifically, the road section selected for this study (see Fig. 6.2a) is located around 42.2946°N and 7.5888° W. The rainfall causes important damages and significant geomorphic changes that are visible in the surrounding orography. These kind of incidents motivates the current work and justifies the test site selection. The slopes of the road are mainly formed by shale that is a rock of natural water proofing, consequently, infiltration is minimal and high road run-offs are frequently generated during rainfall events in this stretch. Height oscillates between 10 m and 15 m. Here the focus will be on the sloped terrain above the road side walls where water flows downhill under during rain events.

Fig. 6.3 shows the domain of the numerical simulation using *DualSPHysics* code. The watershed (shaded area in Fig. 6.3a) is the catchment area that concentrates water on the region of interest (Fig. 6.3b). Thus, the watershed outlet corresponds to the upper boundary of the region under study when the inflow is imposed as shown in Fig. 6.3c. Note that the dynamic boundary conditions described above are only used to simulate the ground. Inlet/outlet conditions were considered for the rest of the boundaries. The lower and lateral boundaries are open, in such a way that particles can go through them and leave the domain. Inflow conditions were considered for the upper boundary. These conditions allow fixing the instantaneous number of particles that enter the computational domain in such a way that the water inflow can be controlled. The darker surface in Fig. 6.3b, which has been magnified in Fig. 6.3c corresponds to the SPH numerical area, which is 83 m x 75 m with a steep slope (~17%) in X direction and a gentler slope (~7%) in Y direction. Particles are initially located with an inter-particle spacing of 0.04 m. The presence of a ditch across the terrain can also be observed in the figure. That ditch, which is approximately 0.80 m deep and 0.5 m wide, was built as a preliminary precautionary measure to prevent flooding on the road. Different simulations, with and without the ditch, have been designed.

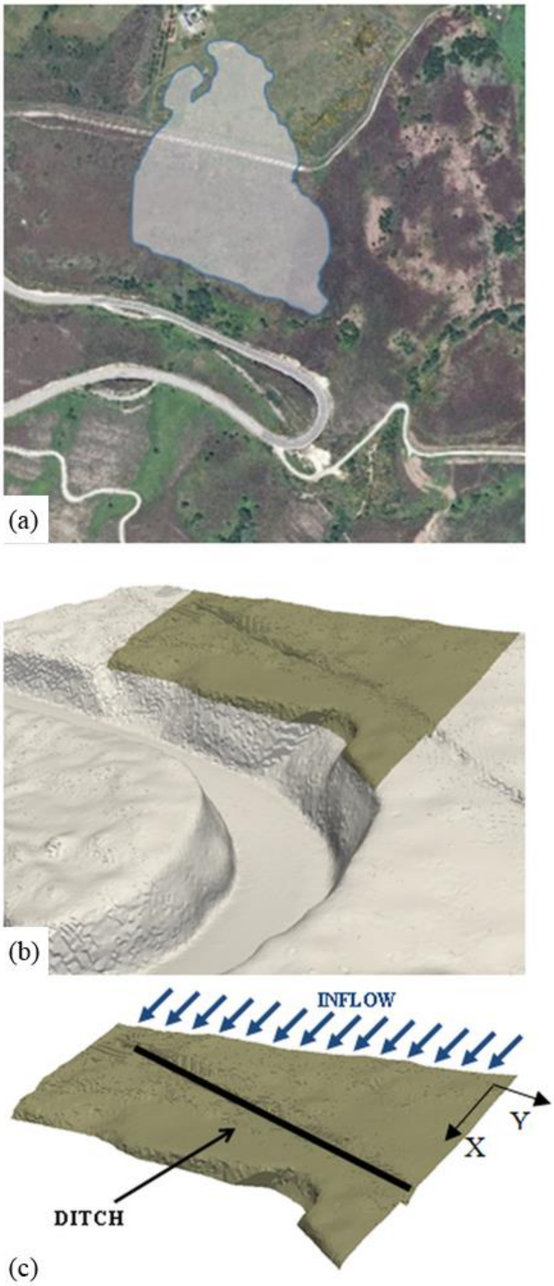


Figure 6.3. Domain of the SPH simulation and location of the ditch. Panel a shows an aerial vision, panels b and c present the domain of the simulation.

In order to solve properly this case configuration some improvements were made to *DualSPHysics*, double precision and inlet condition. Without double precision the numerical uncertainties will make impossible to simulate this case, where a *dp* in the order of centimetres is used to describe a domain with thousands of square meters. Also the inlet condition makes possible to face this case, as mentioned before a centimetre *dp* is used to describe the domain, so the number of boundary particles is large. Normally in a study of this type a large reservoir is created in the top of domain and the water is released by a gate, in this case is not possible. The inlet condition reduces drastically the number of particles that the system has to handle at once and also allows a better control of the amount of water poured into the system.

6.3.1 Numerical setup

Table 6.1 summarises the main SPH parameters used in *DualSPHysics* to perform the simulations of this work.

Table 6.1. Numerical parameters of the simulation of run-off.

Parameter	value
Particle size dp (m)	0.04
Smoothing length h (m)	0.06
Number of particles np (m)	302,181
Artificial viscosity α in Eq. 2.15	0.01
Speed of sound c_o in Eq. 2.26	$20\cdot V_{max}$
CFL number C in Eq. 2.41	0.2
Physical time simulated (s)	900

6.3.2 Inflow conditions

A maximum daily precipitation of P=96 mm day^{-1} was estimated for the area under study according to data provided by the Spanish Meteorological Agency (AEMET) considering a return period of 50 years (http://www.aemet.es). A concentration time of approximately 0.1

hour was calculated by using the length (L=200 m), the slope (S=0.17) and the area (A=0.045 km^2) of the watershed following the expression $t_c = (14.6/60)\, L\, A^{-0.1} S^{-0.2}$, according to (Pilgrim & Cordery, 1993). The discharge (in m^{3}s^{-1}) was calculated following $Q = C_N\, I\, A/3$, where C_N=0.3 is the run-off curve number obtained from (MOPU, 1990), A is the watershed area and I_t is the intensity of the precipitation that can be calculated for the concentration time according to $I_t = I_d (I_1/I_d)^{3.5287-2.5287\, {t_c}^{0.1}}$ following (MOPU, 1990), where $I_d = P/24$ is the intensity per hour and $I_1 = 9\, I_d$, according to tabulated data for the area provided by (MOPU, 1990). Using this rough approximation, the obtained discharge is on the order of 0.5 m^{3}s^{-1}. Therefore, a discharge similar to the previous one is desired to be simulated using *DualSPHysics* code**.** Special inlet conditions were implemented for these simulations. New fluid particles are being inserted at the upper boundary as shown in Fig. 6.3c (blue arrows). A numerical reservoir is created beyond the upper part of the region of interest (Fig. 6.4). The reservoir is initially filled with fluid particles and these ones only enter the domain when overflow. Layers of particles are created above the existing ones in the reservoir at a rate that also determines the overflow. The numerical inflow needs to be 0.5 m^{3}s^{-1}, each m^3 contains 17,802 particles so 8,901 particles are being inserted per second.

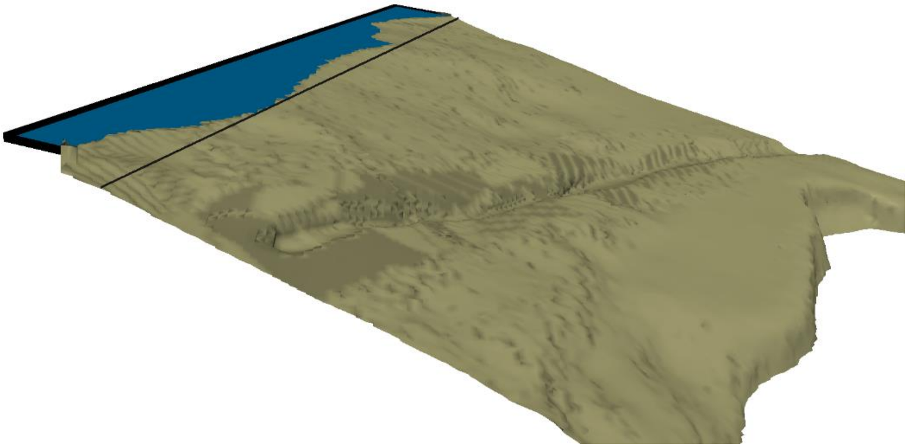


Figure 6.4. Domain of the SPH simulation and reservoir to create the desired inflow rate, the black line represents where the inflow is measured.

Fig. 6.5 shows the time story of numerical water inflow at the upper boundary. The situation represents a short and intense rain event lasting about 15 minutes. Water inflow attains a constant value (around 0.49 m^3s^{-1}) at approximately 240 s, which lasts till 600 s when the inflow is slowly reduced.

The rain falling onto the simulated area is not considered since it is negligible compared to the water collected in the watershed. This can be estimated considering that the precipitation is on the order of 96 mm day^{-1} and the area of the computational domain is close to 6000 m^2, the rain discharge inside the domain is 0.006 m^3s^{-1}, which is negligible when compared with the water inflow 0.5 m^3s^{-1}.

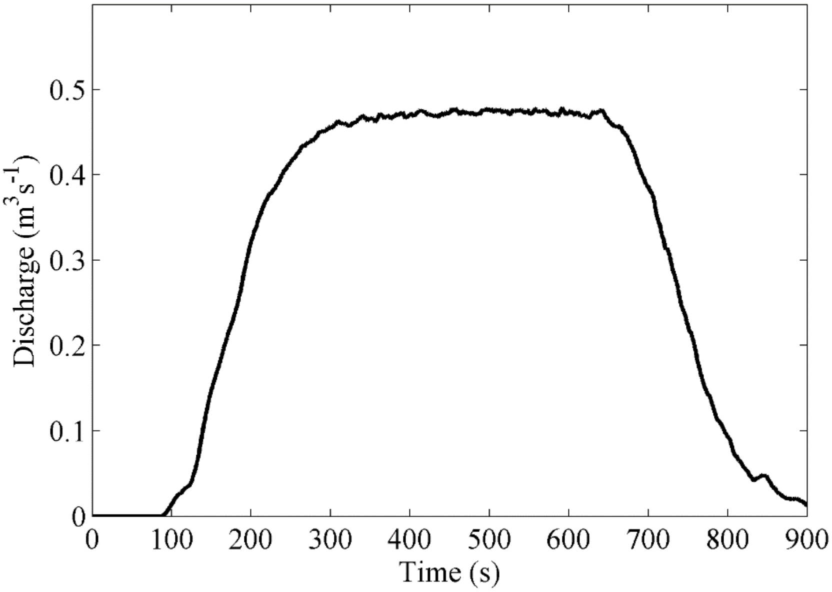


Figure 6.5. Water inflow at the upper boundary

6.3.3 Viscosity calibration for fluid-boundary interaction

As mentioned before, the artificial viscosity corresponding to the interaction between fluid particles and a boundary that represents a terrain with vegetation should be calibrated. First α=0.01 was imposed as in Chapter 4. However, two different values of α (Eq. 2.15.) are considered here to mimic the interaction among fluid particles (α_{FF}) and among fluid and boundary particles (α_{FB}). This last parameter is critical due to the different nature of the terrain. In laboratory experiments, where the bottom is made of Plexiglas and exerts little influence on water propagation, α_{FF}=α_{FB} is usually adopted. Nevertheless, in real cases the roughness of terrain plays a key role on water propagation. The procedure followed to obtain both α values will be described later.

The Manning coefficients n is an empirical coefficient used on engineering representing the ease with which a flow descends a certain slope characterized by the kind of surface and the amount of debris existing in the channel of study. According to Manning equation

$$V = \frac{1}{n} R^{2/3} \sqrt{S} \tag{6.1}$$

the velocity (V) can be calculated in terms of the Manning coefficient (n), the hydraulic radius (R) and the slope of the channel (S).

Taking into account the vegetation of the terrain under scope (area with heavy brush), the Manning coefficient is on the order of 0.075 (see Table 6.2), which results in a velocity of 0.46 ms^{-1} using Eq. 6.1.

Table 6.2. Values of Manning coefficient according to the material of the surface.

Type of surface	Manning coefficient (n)
Glass	0.010±0.002
Concrete	0.012±0.002
Steel	0.014±0.003
Natural channel (low debris)	0.040±0.010
Area (low brush)	0.050±0.020
Area (high brush)	0.075±0.025

Therefore, the model was calibrated to obtain α_{FB} using an artificial channel that mimics the main features of the real terrain to be considered in the study (slope 0.17, width 2 m, length 73 m and water depth 0.04 m). *DualSPHysics* was executed for different ratios between α_{FB} and α_{FF}

(Figure 6.6) in order to obtain a velocity similar to the theoretical one (0.46 ms^{-1}), which was achieved for α_{FB}=16 α_{FF}.

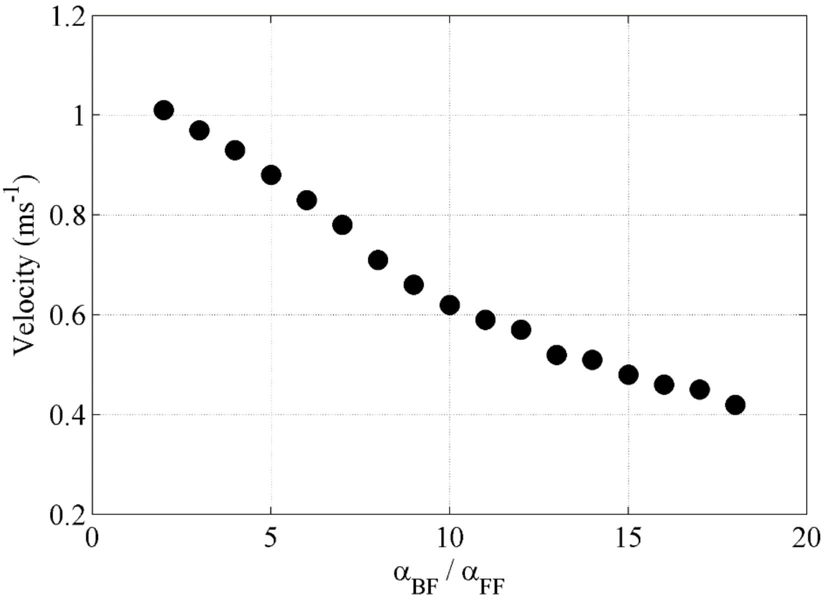


Figure 6.6. Flow velocity dependence on the ratio between α_{BF} and α_{FF}.

6.4 Results

First, run-off was simulated without the ditch (Fig. 6.7). In the first two panels water only affects the upper (Time=100 s) and middle (Time=150 s) parts of the computational domain. In the third panel (Time=200 s) the water has already arrived at the lower part of the domain and falls into the road. From a computational point of view, the run in absence of ditch depicted in Fig. 6.7 was carried out with 5,365,925 boundary particles. The number of fluid particles depended on the instant of the run due to the input/output conditions in the system. Thus, only 302,181 were initially considered, but this number increases until 2,217,033 at the moment of maximum flow. A runtime of 138 hours was necessary to run 900 seconds of physical time. All simulations were executed on a GPU GTX Titan using double precision.

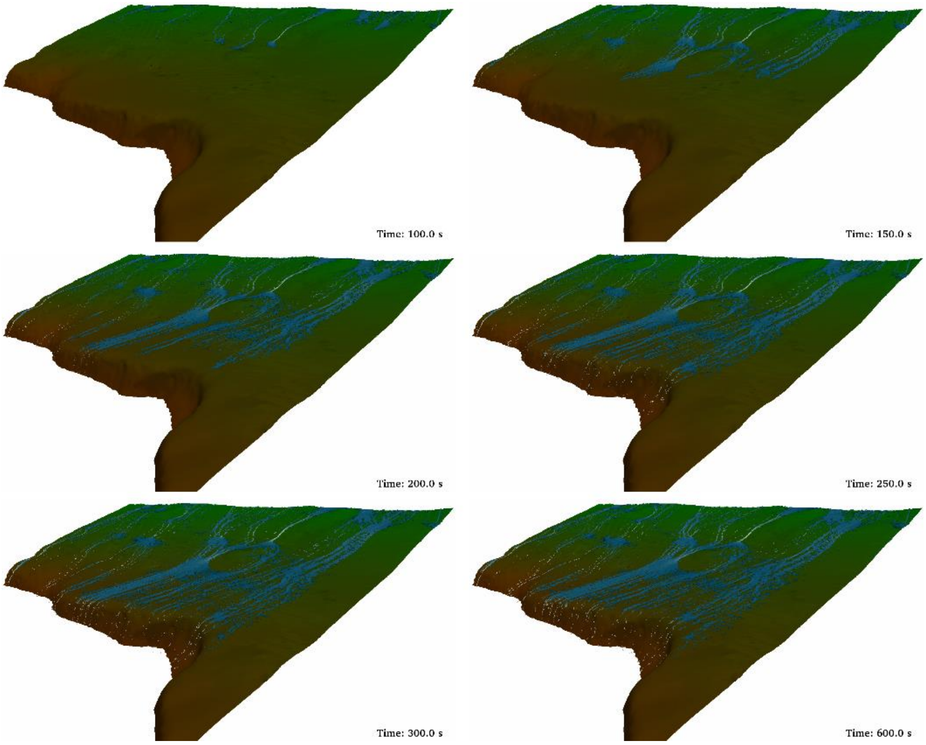


Figure 6.7. Different instants of the *DualSPHysics* simulation without ditch.

The amount of water arriving at the road can be seen in Fig. 6.8. Basically, the plot is similar to the one showing the flow entering the computational domain (Fig. 6.5). The maximum amount of water (around 0.46 m^3s^{-1}) is slightly smaller in Fig. 6.8 because a small percentage of water (around 6%) leaves the computational domain through the open lateral boundaries. In addition, the signal in Fig. 6.8 is delayed in around 50 seconds when compared with the signal in Fig. 6.5. The flow takes around 50 seconds to reach the lower part of the region of interest since entering the computational domain that is, travelling 73 m.

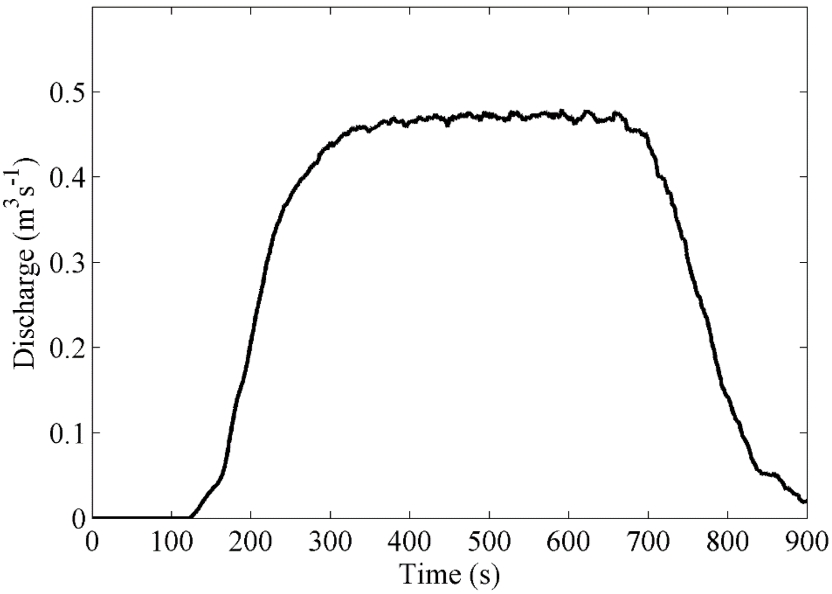


Figure 6.8. Water flow into the road.

Fig. 6.9 shows the same discharge conditions as previously described for Fig. 6.7. The only difference between both simulations is the presence of a ditch that mimics the main features of the real one shown in Figs. 6.2 and 6.3; namely, 80 cm deep and 50 cm wide. The first frames are similar to the one in Fig. 6.7. Nevertheless, the rest of the frames show how water is collected and drained by the ditch. Actually, there is not water discharge to the road under these conditions. The computational conditions are similar to the ones described above. The only difference is the maximum number of fluid particles, which is now 2,431,254, since the presence of the ditch increase the residence time of water inside the numerical domain. Consequently, the runtime also increased and it is now 150 hours.

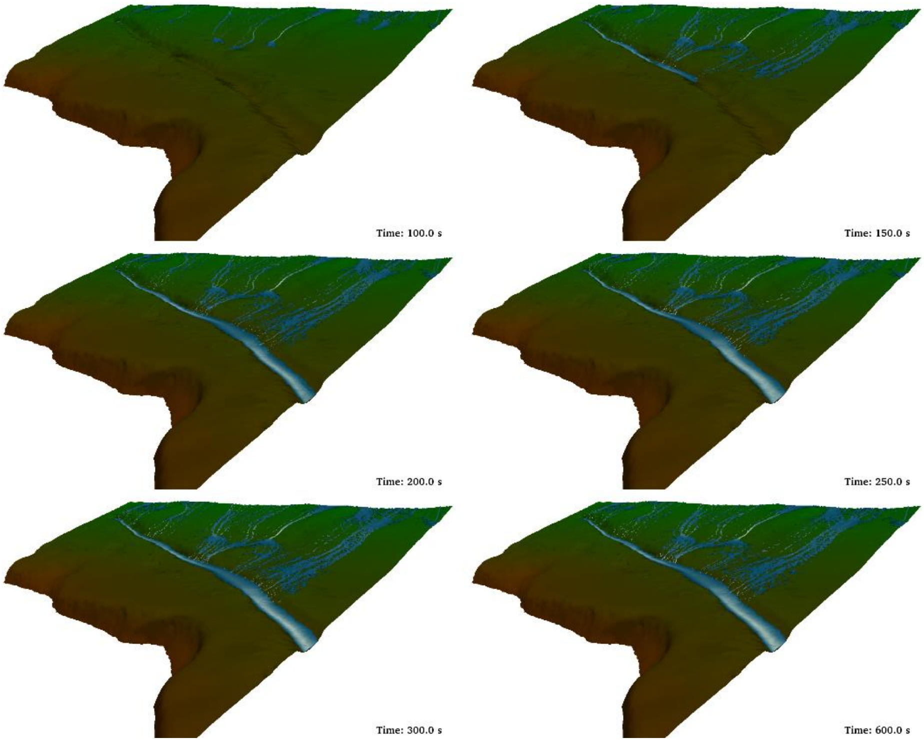


Figure 6.9. Different instants of the *DualSPHysics* simulation with a ditch 0.8m deep. Additional detail about the simulation can be found at http://youtu.be/T1po4onk0v4 .

According to the plans of the builders, this ditch should be enough to drain water even under extreme run-offs. Nevertheless, ditches can become clogged with sediments or debris, especially after strong storm events. Here, the effect of sediment is also analysed, assuming that the maintenance of the ditch was enough to prevent clogs but not to prevent a slow sedimentation inside the ditch that reduced its depth. Fig. 6.10 shows the water discharge into the road and the water drained by the ditch as a function of the depth of the ditch. The same water inflow (Fig. 6.5) was considered in all simulations. The case with depth equal to zero corresponds to the absence of ditch analysed in Fig. 6.7 and 6.8. Ditches with depths less than 0.2 m are shown to be inefficient under strong rain conditions and more than 90% of the discharge arrives at the road. The efficiency increases markedly for ditches deeper than 0.4 m, actually 50% of the discharge is drained by the ditch. Finally, a 0.7 m deep ditch is able to drain the 100% of the discharge under strong rain conditions.

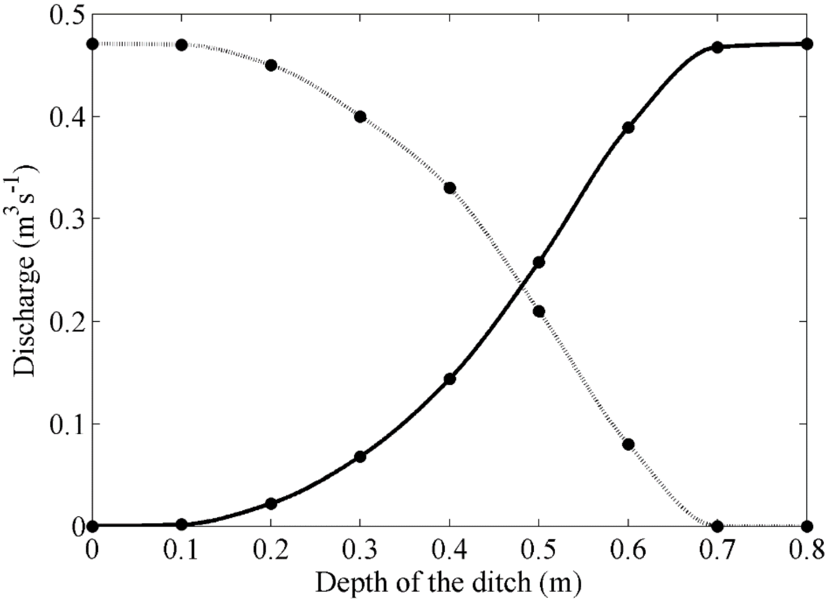


Figure 6.10. Discharge into the road (dashed line) and drainage of the ditch (solid line).

7. Conclusions and future work

The conclusions of this work are provided in this section. Future lines of investigation are also presented.

7.1 Conclusions

A CPU-GPU solver called *DualSPHysics* has been developed and applied to free-surface flow problems. The model was developed from the SPHysics FORTRAN code, inheriting the properties of stability and accuracy of its predecessor but increasing its performance and features. *DualSPHysics* code can be run as either a CPU code or a GPU code depending on the availability of hardware.

Simulations with a large number of particles can be performed on a personal computer equipped with a CUDA-enabled GPU card taking advantage of the performance and memory space provided by the new GPU technology. This means that research can be conducted with available cheap technology for problems that previously required high-performance computing (HPC). The speedups obtained in this work reveal the possibility to study real-life engineering problems at a reasonable computational cost. Hence several realistic problems related to civil and coastal engineering have been studied using *DualSPHysics*. Also new functionalities have been implemented to perform these studies. These functionalities open new fields where *DualSPHysics* can be applied

7.1.1 Conclusions for Case I: Coastal protection

Sea level rise will contribute to increase the flooding risk in such a way that the protection measures should be re-evaluated. Coastal flood hazards are determined by the interaction of storm surges and waves with sea bathymetry and coastal topography. Here both seabed and land cover play a key role to determine the inland extent of flooding. Coastal flood models must not only account for the extent of flooding but also for its destructive power which is intrinsically linked to the force exerted by the incoming wave on coastal structures.

DualSPHysics has been proven to be an accurate model to reproduce both wave propagation in coastal areas and wave impact on structures. The capabilities of *DualSPHysics* to reproduce wave-structure interaction were shown where wave heights and forces exerted onto objects of the coast were numerically computed. The case of study mimics a realistic promenade including the urban furniture and with dimensions and geometries close to the real ones.

7.1.2 Conclusions for Case II: Moorings & Floating bodies

The functionalities necessary to simulate moorings were implemented in *DualSPHysics*. The new implementation is able to solve properly both chain forces and the effect moorings in the floating bodies.

The formulation that governs the floating bodies in *DualSPHysics* was presented. The buoyancy of the floating bodies was validated against numerical data from VOF showing a really good agreement.

A whole new set of equations have been implemented to solve moorings. This code covers all the possible states of a moored line, taking into account parameters such as relative position of anchoring and link points, moored line weight, shape of the catenary function…

The implementation was validated against experimental data and compared with other numerical models. This validation provided really good agreement with both numerical and experimental data at various scales. Also the validation for multiple lines was carried out providing good results.

Two different application examples were presented, the first one is a boat under the action of side waves and only one moored line. The second application consisted of a wind-turbine base with three moorings under the effect of extreme waves.

Hence, this new functionality was validated for single and multiple line problems with good results. This new functionality extends the application range of the *DualSPHysics* code to a whole new set of scenarios.

7.1.3 Conclusions for Case III: Run-off on real terrains

The study represents an approach to model the water run-off on complex geometries, with direct application in the field of civil engineering. The study merges two advanced technologies; namely, UAV photogrammetry and a fluid solver based on SPH technique.

UAV photogrammetry was used to obtain the geometry of the area. The point cloud data provided after photogrammetric restitution can be converted into points through a triangulation process and these points are used to create the initial setup of the SPH meshless method. *DualSPHysics* was applied to compute the water trajectories and its interaction with a complex terrain.

Fast and intense rain events were simulated using a water inflow that mimics extreme meteorological conditions in the zone that were calculated for a return period of 50 years. Thus, an inflow of about 0.5 m3s-1 was imposed through the upper boundary of the computational domain.

The effect of a ditch (0.8 m deep and 0.5 m wide) to prevent water flow into the road was analysed. This precautionary measure was observed to be effective to drain water under extreme conditions. Nevertheless, the efficiency of the ditch was observed to decrease when sedimentation reduces its depth, in such a way that the drainage was only on the order of 50% when the depth of the ditch decreased at about 0.45 m.

In summary, the combination of UAV measurements and *DualSPHysics* simulations has proven to be a suitable tool both to describe water floods on areas adjacent to the roads and to design measures to palliate those floods. The resulting tool can be applied to multiple civil engineering scenarios and give more information than the typical GIS solutions, unable to represent the real behaviour of the fluid.

7.2 Future work

As proven before, the model is able to face a wide range of real engineering problems thanks to its high performance and combined with the new functionalities presented. Nevertheless, there are some aspects where some improvement is necessary to increase the performance and precision. This topics of interests will help *DualSPHysics* to face even more complex cases. Here are summarized some possible lines of research after the work of this thesis

- Body and boundary deformation: *DualSPHysics* covers perfectly the fluid-boundary interaction, but in some extreme events the coastal structures and floating objects can be deformed, even broken, by the action of waves. The challenge in this area is the proper implementation of the material properties and the ability to represent the inner stresses. This implementation can be even more useful for the design of coastal structures or floating devices and can provide the engineers a really complete tool to ease their designing work.

- Complex mooring implementation: more complex moorings can be needed to keep in place off-shore structures. The complex moorings have a floater or buoy mid chain. This configuration helps to mitigate some of the forces in the moored line acting as a spring-like mechanism dissipating forces.

- Dynamic mooring and surge interaction: The moorings implemented in *DualSPHysics* do not interact with the fluid particles and the code neglects the inertia of the chain. A more complete implementation must cover both situations. Studies have proven that forces due to currents hitting the moored line, and the elasticity and vibration of the moored line are not negligible.

- Sediment erosion and infiltration: The run-off study is focused in an extreme event, such a storm, but other issues can be studied for long term events. Longer events can cause sediment moving and infiltration in deeper layers of the terrain. These two issues can deform and create new instabilities that can affect the terrain causing rock fall and landslides. To solve these issues material properties must be added to the boundary particles.

Bibliography

[Altomare et al., 2014] Altomare, C., Crespo, A.J.C., Rogers, B.D., Domínguez, J.M., Gironella, X. & Gómez-Gesteira, M. (2014). Numerical modelling of armour block sea breakwater with Smoothed Particle Hydrodynamics. *Computers and Structures*, 130: 34-45.

[Arnason, 2005] Arnason H. (2005) Interaction between an Incident bore and a free-standing coastal structure, *Ph. D., University of Washington*

[Barreiro et al., 2013] Barreiro, A, Crespo, A.J.C., Domínguez, J.M. & Gómez-Gesteira, M. (2013). Smoothed Particle Hydrodynamics for coastal engineering problems. *Computers and Structures*, 120(15): 96-106

[Barreiro et al., 2014] Barreiro, A., Domínguez, J.M., Crespo, A.J.C., González-Jorge, H., Roca, D., & Gómez-Gesteira, M. (2014). Integration of UAV photogrammetry and SPH modelling of fluids to study runoff on real terrains. *PLoS ONE*, 9(11): e111031.

[Batchelor, 1974] Batchelor, G. K. (1974). Introduction to fluid dynamics. *Cambridge University Press.*

[Benz, 1990] Benz, W. (1990). Smoothed Particle Hydrodynamics: a review. *Numerical Modelling of Nonlinear Stellar Pulsations: Problems and Prospects, Kluwer Acad. Boston Publ*.,269-288.

[Belytschko et al., 1998] Belytschko, T., Krongauz,Y., Dolbow, J. & Gerlach, C. (1998). On the completeness of meshfree particle methods. *Int. J. Numer. Mech. Engrg.,* 43:785–819.

[Bonet & Lok, 1999] Bonet, J. & Lok, T.-S.L. (1999). Variational and momentum preservation aspects of Smoothed Particle Hydrodynamics formulations. *Computat. Methods Appl. Mech. Engineering.*, 180:97-115.

[Canelas et al., 2015] Canelas R.B., Domínguez J.M., Crespo A.J.C., Gómez-Gesteira M., Ferreira R.M.L. (2015). A Smooth Particle Hydrodynamics discretization for the modelling of free surface flows and rigid body dynamics. *International Journal for Numerical Methods in Fluids.*

[Capone et al., 2010] Capone, T., Panizzo, A., & Monaghan, J.J. (2010). SPH modeling of water waves generated by submarine landslides. *Journal of Hydraulic Research Extra Issue (2010*), 48:80–84.

[Chen & Beraun, 2000] Chen, J.K. & Beraun, J.E. (2000). A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems. *Comput. Meth. Appl. Mech. Engrg,*. 190:225–239.

[Christensen, 2006] Christensen, E.D. (2006) Large eddy sumilation of spilling and plunging breakers, *Coastal Engineering*, 53, 463-485.

[Crespo et al., 2007a]Crespo, A.J.C., Gómez-Gesteira, M., Dalrymple, R.A. (2007). 3D SPH Simulation of large waves mitigation with a dike. *Journal of Hydraulic Research,* 45:631-642.

[Crespo et al., 2007b] Crespo, A.J.C., Gómez-Gesteira, M. & Dalrymple, R.A. (2007). Boundary conditions generated by dynamic particles in SPH methods. *CMC: Computers, Materials, & Continua*, 5(3):173–184.

[Crespo et al., 2008] Crespo, A.J.C., Gómez-Gesteira, M. & Dalrymple, R.A. (2008). Modeling Dam Break Behavior over a Wet Bed by a SPH Technique. *Journal of Waterway, Port, Coastal and Ocean Engineering,* 134:313-320.

[Crespo et al., 2011] Crespo, A.J.C., Dominguez, J.M., Barreiro, A., Gómez-Gesteira, M. & Rogers, B.D., (2011). GPUs, a new tool of acceleration in CFD: Efficiency and reliability on Smoothed Particle Hydrodynamics methods. *PLoS ONE*, 6(6), e20685

[Crespo et al., 2015] Crespo, A.J.C., Domínguez, J.M., Rogers, B.D., Gómez-Gesteira, M., Longshaw, S., Canelas, R., Vacondio, R., Barreiro, A., & García-Feal, O. (2015). DualSPHysics: open-source parallel CFD solver on Smoothed Particle Hydrodynamics (SPH). *Computer Physics Communications*, 187: 204-216.

[Colagrossi & Landrini, 2003] Colagrossi, A. & Landrini, M. (2003). Numerical simulation of interfacial flows by Smoothed Particle Hydrodynamics. *Journal of Computational Physics*, 191:448-475.

[Cuomo et al., 2006] Cuomo, G., Panizzo, A., & Dalrymple, R.A. (2006). SPH-LES two phase simulation of wave breaking and wave-structure interaction. In *Proc. 30th International Conference on Coastal Engineering*, 274-286.

[Dalrymple & Knio, 2000] Dalrymple, R.A. & Knio, O. (2000) SPH modelling of water waves. *Coastal Dynamics*, 01, 779-787

[Dalrymple & Rogers, 2006] Dalrymple, R. A. & Rogers, B.D. (2006). Numerical modeling of water waves with the SPH method. *Coastal Engineering*, 53:141-147.

[De Leffe et al., 2010] De Leffe, M., Le Touzé, D., & Alessandrini, B. (2010). SPH modeling of a shallow-water coastal flows. *Journal of Hydraulic Research Extra Issue (2010*), 48:118-125

[Del Guzzo & Panizzo, 2007] Del Guzzo, A. & Panizzo, A. (2007). Application of SPH-NSWE to simulate landslide generated waves. In *SPHERIC, Second International Workshop*, 161-164.

[Domínguez et al., 2011] Dominguez, J.M., Crespo, A.J.C., Gómez-Gesteira, M & Marongiu, J.C. (2011). Neighbour lists in smoothed particle hydrodynamics. International Journal For Numerical Methods in Fluids. doi: 10.1002/fld.2481.

[Domínguez et al., 2011] Domínguez J.M., Crespo A.J.C., Barreiro A., Gómez-Gesteira M. & Mayrhofer, A. (2011) Development of a new pre-processing tool for SPH models with complex geometries. *6th International SPHERIC workshop*, ISBN 978-3-89220-658-3: 117-124.

[Domínguez et al., 2013] Domínguez, J.M, Crespo, A.J.C. & Gómez-Gesteira, M. (2013). Optimization strategies for CPU and GPU implementations of a smoothed particle hydrodynamics method. Computer Physics Communications, 184(3): 617-627.

[Ellero et al., 2007] Ellero, M., Serrano, M., & Espanol, P. (2007). Incomprensible Smoothed Particle Hydrodynamics. *Journal of Computational Physics*, 226:1731-1752.

[Everaerts, 2008] Everaerts J (2008) The Use of Unmanned Aerial Vehicles (UAVS) for Remote Sensing and Mapping. *IAPRS&SIS*, 37(B1): 1187-1192.

[Faltinsen, 1993] Faltinsen, O.M. (1993) Sea Loads on Ships and Offshore Structures, *Cambridge University Press*, Sep 23, 1993.

[Fekken, 2004] Fekken G. (2004). Numerical simulation of free surface flow with moving rigid bodies. *PhD Thesis, University of Groningen*.

[Fourtakas et al., 2014] Fourtakas, G., Rogers, B.D., & Laurence, D.R. (2014). Modelling sediment resuspension in industrial tanks using SPH. *La Houille Blanche,* n°2, 2013, p. 39-45.

[Fourtakas, 2014] Fourtakas, G. (2014) Modelling multi-phase flows in Nuclear Decommissioning using SPH. *PhD Thesis, School of Mechanical, Aeronautical and Civil Engineering. The University of Manchester: Manchester.*

[Geeraerts et al., 2009] Geeraerts, J., Kortenhaus, A., Gonález-Escrivá, J.A., De Rouck, J., Troch, P. (2009). Effects of new variables on the overtopping discharge at steep rubble mound breakwaters — The Zeebrugge case. *Coastal Engineering* 56, 141-153.

[Gingold & Monaghan, 1977] Gingold R.A. & Monaghan, J.J. (1977) Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon Not R Astr Soc* 181: 375–389.

[Gómez-Gesteira & Dalrymple, 2004] Gómez-Gesteira, M. & Dalrymple, R.A. (2004). Using a 3D SPH Method for Wave Impact on a Tall Structure. *J. Waterway, Port, Coastal and Ocean Engineering.*, 130(2):63-69.

[Gómez-Gesteira et al., 2005] Gómez-Gesteira, M., Cerqueiro, D., Crespo, A.J.C., & Dalrymple, R.A. (2005). Green water overtopping analyzed with a SPH model. *Ocean Engineering*, 32:223-238.

[Gómez-Gesteira et al., 2010] Gómez-Gesteira, M., Rogers, B.D., Dalrymple, R.A. & Crespo, A.J.C. (2010) State-of-the-art of classical SPH for free-surface flows, *Journal of Hydraulic Research, 48, Extra Issue*, 6-27.

[Gómez-Gesteira et al., 2012a] Gómez-Gesteira, M., Rogers, B.D., Crespo, A.J.C., Dalrymple, R.A., Narayanaswamy, M. & Domínguez, J.M. (2012). SPHysics - development of a free-surface fluid solver- Part 1: Theory and Formulations. *Computers & Geosciences* 48, 289-299.

[Gómez-Gesteira et al., 2012b] Gómez-Gesteira, M., Crespo, A.J.C., Rogers, B.D., Dalrymple, R.A., Domínguez, J.M. & Barreiro, A. (2012). SPHysics - development of a free-surface fluid solver- Part 2: Efficiency and test cases. Computers & Geosciences 48, 300-307.

[Gotoh et al., 2001] Gotoh, H. Shibihara, T. & Hayashii, M. (2001). Subparticle-scale model for the MPS method-lagrangian flow model for hydraulic engineering. Computational Fluid Dynamics Journal 9, 339–347.

[Higera et al., 2013] Higuera, P., Lara, J.L.,& Losada, I.J. (2013). Simulating Coastal Engineering Processes with OpenFOAM®. *Coastal Engineering* 71, 119-134.

[Hu & Adams, 2007] Hu, X. & Adams, N. (2007). An incompressible multi-phase SPH method. *Journal of Computational Physics*, 227:264-278.

[Hughes & Graham, 2010] Hughes, J.P. & Graham D.I. (2010). Comparison of incompressible and weakly-compressible SPH models for free-surface flows. *Journal of Hydraulic Research Extra Issue (2010*), 48:126-134.

[Janosi et al., 2004] Janosi, I. M., Jan, D., Szabo, K.G., & Tel, T. (2004). Turbulent drag reduction in dam-break flows. *Exp. Fluids*, 37, 219-229.

[Johanning et al., 2006] Johanning, L., Smith, G.H. & Wolfram, J. (2006) Mooring desing approach for wave energy converters. *: Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment,* Vol. 220, No. 4, 2006, p. 159-174.

[Johanning et al., 2007] Johanning, L., Smith G.H. & Wolfram, J. (2007) Measurements of static and dynamic mooring line damping and their importance for floating WEC devices, *Ocean Engineering*, volume 34, pages 1918-1934.

[Kleeffsman et al., 2005] Kleefsman, K.M.T., Fekken, G., Veldman, A.E.P., Iwanowski, B., & Buchner. B. (2005). A volume-of-fluid based simulation method for wave impact problems. *Journal of Computational Physics 206* (1), 363–393.

[Lee et al., 2010] Lee, E., Violeau, D., Issa, R. & Ploix, S. (2010). Application of weakly compressible and truly incompressible SPH to 3-D water collapse in waterworks. *Journal of Hydraulic Research Extra Issue (2010*), 48:50-60.

[Liu, 2003] Liu, G.R. (2003). Mesh Free methods: Moving beyond the infinite element method. *CRC Press*.

[Liu & Liu, 2003] Liu, G.R. & Liu, M.B. (2003). Smoothed Particle Hydrodynamics: a meshfree particle method. *World Scientific*.

[Lo & Shao, 2002] Lo, E. & Shao, S. (2002). Simulation of near-shore solitary wave mechanics by an incompressible SPH method. *Applied Ocean Research*, 24:275-286.

[Lopez et al., 2010] Lopez, D., Marivela, R. & Garrote, L. (2010). Smoothed Particle Hydrodynamics model applied to hydraulic structures: a hydraulic jump test case. *Journal of Hydraulic Research Extra Issue (2010*) Vol. 48:142-158.

[Lucy, 1977] Lucy, L. (1977) A numerical approach to the testing of the fission hypothesis. *Astron J* 82(12): 1013–1024.

[Manenti et al., 2012] Manenti, S., Sibilla, S., Gallati, M., Agate, G. & Guandalini, R. (2012). SPH Simulation of Sediment Flushing Induced by a Rapid Water Flow. *Journal of Hydraulic Engineering.*

[Marongiu et al., 2010] Marongiu, J.C., Leboeuf, F., % Caro, J. (2010). Free surface flows simulations in Pelton turbines using an hybrid SPH-ALE method. *Journal of Hydraulic Research* Vol.48 (1): 40-49.

[Mokos et al., 2013] Mokos, A.D., Rogers, B.D., Stansby, P.K., & Domínguez, J.M. (2013). GPU Acceleration of 3-D Multi-phase SPH Simulations for Violent Hydrodynamics. *8th SPHERIC.*

[Mokos, 2014] Mokos, A. D. (2014). Multi-phase modelling of violent hydrodynamics using Smoothed Particle Hydrodynamics (SPH) on Graphics Processing Units (GPUs). *PhD Thesis, School of Mechanical, Aeronautical and Civil Engineering. The University of Manchester: Manchester.*

[Monaghan, 1982] Monaghan, J.J. (1982). Why particle methods work. *Siam J. Sci. Stat. Comput.,* 3:422-433.

[Monaghan & Lattanzio, 1985] Monaghan, J.J. & Lattanzio, J. C. (1985). A refined particle method for astrophysical problems. *Astron. Astrophys*. 149:135

[Monaghan, 1989] Monaghan, J. (1989). On the problem of penetration in particle methods. Journal of Computational Physics, 82, 1–15.

[Monaghan, 1992] Monaghan, J.J. (1992). Smoothed Particle Hydrodynamics. *Annual Rev. Astron. Appl.*, 30:543-574.

[Monaghan, 1994] Monaghan, J.J. (1994). Simulating free surface flows with SPH. *Journal Computational Physics*, 110:399- 406.

[Monaghan & Kos, 1999] Monaghan, J.J. & Kos, A. (1999). SolitaryWaves on a Cretan Beach. *J. Waterway, Port, Coastal and Ocean Engineering.*, 125:145-154.

[Monaghan et al., 2003] Monaghan, J.J., Kos, A., Issa, A. (2003) Fluid motion generated by impact, Journal of Waterway, Port, *Coastal and Ocean Engineering* 129 (2003) 250-259.

[Monaghan, 2005] Monaghan, J.J. (2005). Smoothed Particle Hydrodynamics. *Reports on Progress in Physics*, 68:1703-1759.

[Monaghan & Kajtar, 2009] Monaghan, J.J. & Kajtar, J.B. (2009). SPH particle boundary forces for arbitrary boundaries. *Computer Physics Comunications*, 180 (10), 1811-1820.

[MOPU, 1990] MOPU (1990) Instrucciones de Carreteras5.2-IC Drenaje Superficial. Ministerio de Obras Públicas y Urbanismo, *Boletín Oficial del estado*, 123, 23-5-1990.

[Morris et al., 1997] Morris, J., Fox, P., & Zhu, Y. (1997). Modeling low Reynolds number incompressible flows using SPH. *Journal of Computational Physics*, 136:214-226.

[Oñate et al., 2011] Oñate, E., Celigueta, M.A., Idelsohn, S.R., Salzar, F., & Suarez, B. (2011). Possibilities of the Particle Finite Element Method for fluid-soil-structure interaction problems. *Computational Mechanichs* 48 (3), 307-318.

[Panizzo & Dalrymple, 2004] Panizzo, A. & Dalrymple, R. A. (2004). SPH modelling of underwater landslide generated waves. In *Proc. 29th International Conference on Coastal Engineering*, 1147-1159.

[Panizzo et al., 2006] Panizzo, A., Cuomo, G., & Dalrymple, R. A. (2006). 3D-SPH simulation of landslide generated waves. In *Proc. 30th International Conference on Coastal Engineering*.

[Pilgrim & Cordery, 1993] Pilgrim, D.H. & Cordery, I. (1993) Flood Runoff. *Handbook of Hydrology, McGrawHill*.

[Remondino et al., 2011] Remondino, F., Barazzetti, L., Nex, F., Scaioni, M. & Sarazzi, D. (2011) UAV photogrammetry for mapping and 3D modeling- current status and future perspectives. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences.*

[Rogers & Dalrymple, 2004] Rogers, B.D. & Dalrymple, R. (2004). SPH modeling of breaking waves. *In Proc. 29th International Conference on Coastal Engineering*, pages 415–427.

[Rogers & Dalrymple, 2007] Rogers, B. & Dalrymple, R.A. (2007). SPH modeling of tsunami waves. In *Advanced Numerical Models for Simulating Tsunami Waves and Runup*.

[Shao & Gotoh, 2004] Shao, S. & Gotoh, H. (2004). Simulating coupled motion of progressive wave and floating curtain wall by SPH-LES model. *Coastal Engineering Journal*, 46(2):171-202.

[Shao et al., 2006] Shao, S., Ji, C.-M., Graham, D., Reeve, D., James, P., & Chadwick, A. (2006). Simulation of wave overtopping by an incompressible SPH model. *Coastal Engineering*, 53:723-735.

[Smagorinsky, 1963] Smagorinsky, J. (1963), General circulation experiments with the primitive equations, i. the basic experiment. *Monthly Weather Review*, 91: pp 99-164, 1963.

[Ulrich et al., 2011] Ulrich, C., Koliha, N., Rung T. (2011). SPH Modelling of water/soil-flows using a variable resolution scheme. *Proc. 6th Int. Spheric Workshop, June 07-10 Hamburg.*

[Vacondio et al., 2010] Vacondio, R., Mignosa, P., Rogers, R.D. & Stansby, P.K. (2010). 2-D numerical modeling of rapidly varying shallow water flows by Smoothed Particle Hydrodynamics technique, *River Flow 2010*, 1622-1628

[Vanneste & Troch, 2012] Vanneste, D., & Troch, P. (2012). Validation of a numerical model for wave interaction with a rubble-mound breakwater. *33rd International Conference on Coastal Engineering, ASCE.*

[Verlet, 1967] Verlet, L. (1967). Computer experiments on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Phys. Rev*., 159:98–103.

[Vila, 1999] Vila, J.P. (1999). On particle weighted methods and SPH. *Math. Models Meth. Appl. Sci.* 9:161–210.

[Wendland, 1995] Wendland, H. (1995). Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.* 4, 389-396.

[Zou & Dalrymple, 2006] Zou, S. & Dalrymple, R.A. (2006). Sediment suspension simulation under oscillatory flow with SPH-SPS method. In *Proc. 30th International Conference on Coastal Engineering*.

[Zou, 2007] Zou, S. (2007). Coastal Sediment Transport Simulation by Smoothed Particle Hydrodynamics. *PhD thesis, Civil Engineering, Johns Hopkins University*.

List of publications

Articles

[Crespo et al., 2011] Crespo, A.J.C., Domínguez, J.M., **Barreiro, A.**, Gómez-Gesteira, M. & Rogers, B.D. (2011). GPUs, a new tool of acceleration in CFD: Efficiency and reliability on Smoothed Particle Hydrodynamics methods. *PLoS ONE* 6 (6), e20685, DOI: 10.1371/journal.pone.0020685.

[Gómez-Gesteira et al., 2012] Gómez-Gesteira, M., Crespo, A.J.C., Rogers, B.D., Dalrymple, R.A., Domínguez, J.M. & **Barreiro, A.** (2012). SPHysics - development of a free-surface fluid solver- Part 2: Efficiency and test cases. Computers & Geosciences 48, 300-307.

[Barreiro et al., 2013] **Barreiro, A.**, Crespo, A.J.C., Domínguez, J.M. & Gómez-Gesteira, M. (2013). Smoothed Particle Hydrodynamics for coastal engineering problems. *Computers and Structures*, 120(15): 96-106. doi:10.1016/j.compstruc.2013.02.010.

[Barreiro et al., 2014] **Barreiro, A.**, Domínguez, J.M., Crespo, A.J.C., González-Jorge, H., Roca, D. & Gómez-Gesteira, M. (2014). Integration of UAV photogrammetry and SPH modelling of fluids to study run-off on real terrains. *PLoS ONE*, 9(11): e111031. doi:10.1371/journal.pone.0111031.

[Crespo et al., 2015] Crespo, A.J.C., Domínguez, J.M., Rogers, B.D., Gómez-Gesteira, M., Longshaw, S., Canelas, R., Vacondio, R., **Barreiro, A.** & García-Feal, O. (2015). DualSPHysics: open-source parallel CFD solver on Smoothed Particle Hydrodynamics (SPH). Computer Physics Communications, 187: 204-216. doi: 10.1016/j.cpc.2014.10.004

Chapters

Barreiro, A., Domínguez, J.M., Crespo, A.J.C., García-Feal, O. & Gómez-Gesteira, M. (2015). Smoothed Particle Hydrodynamics for Free-Surface Flows. *In: Select Topics of Computational and Experimental Fluid Mechanics, Environmental Science and Engineering, Springer International Publishing*, pp. 119-136. ISBN 978-3-319-11486-6. doi:10.1007/978-3-319-11487-3_6.