

## **SIMULACIÓN DE UN OLEAJE MONOCROMÁTICO CON UNA TÉCNICA DE PARTÍCULAS Y COMPARACIÓN CON DATOS EXPERIMENTALES**

**Paolo Gyssels, Lisandro Martinez, Mariana Pagot y Andrés Rodriguez**

Laboratorio de Hidráulica, Universidad Nacional de Córdoba, Av Filloy s/n, 5000 Córdoba,  
Argentina, [pgyssels@efn.uncor.edu](mailto:pgyssels@efn.uncor.edu), <http://www.efn.uncor.edu/investigacion/hidraulica/>

**Palabras claves:** DualSPHysics, modelo físico, modelo numérico de partículas, canal de oleaje.

**Resumen.** Se presenta una aplicación hidrodinámica del modelo de partícula DualSPHysics (Dual Smoothed Particle Hydrodynamics, Hidrodinámica de Partículas Suavizadas – SPH) para estudiar la propagación de olas regulares e irregulares y su interacción con distintas condiciones de contorno en un canal experimental de oleaje bidimensional.

El método SPH es una aproximación lagrangiana que permite simular grandes deformaciones de superficies libres.

Se compararon los resultados con la modelación experimental de la propagación de una ola monocromática y de un oleaje espectral en el canal de oleaje del Laboratorio de Hidráulica de la Universidad Nacional de Córdoba. Se simuló la propagación de este oleaje a lo largo de 13 m de longitud del canal compuesto por un tramo recto y una playa rectilínea. El modelo se validó con los datos experimentales, y se observó una representatividad de los datos modelados numéricamente con una aproximación entre 4 y 5 mm respecto de los datos medidos en el modelo físico.

Así, el modelado con DualSPHysics es factible de utilizar como una herramienta numérica en la predicción de la transformación de diferentes ondas frente a la interacción de flujos complejos.

## 1 INTRODUCCION

En este trabajo se presenta el modelo SPH para estudiar la propagación de olas regulares e irregulares y su interacción con distintas condiciones de contorno.

Se trata de un método totalmente lagrangiano, desarrollado en los años setenta por [Lucy \(1977\)](#), [Gingold y Monaghan \(1977\)](#), conjuntamente con investigadores de la Universidad Johns Hopkins (EE.UU.), la Universidad de Vigo (España), la Universidad de Manchester (Reino Unido) y la Universidad de Roma Sapienza (Italia).

Fue desarrollado en principio para el estudio de fenómenos en el campo de la astrofísica y recientemente se está utilizando para diferentes problemas relacionados con mecánica de fluidos. Cuenta con tres versiones de códigos disponibles: SPHysics Serial, SPHysics Parallel, Dual- SPHysics. El método analiza las propiedades hidráulicas relevantes de cada partícula, y permite representar deformaciones de una superficie libre ([Crespo et al., 2011](#)).

## 2 OBJETIVOS

Los objetivos del trabajo fueron:

1. Reproducir numéricamente la propagación de un oleaje, utilizando el modelo DualSPHysics y comparar los resultados con el mismo oleaje simulado en el canal bidimensional del Laboratorio de Hidráulica (LH) de la Universidad Nacional de Córdoba (UNC).
2. Identificar los fenómenos físicos involucrados en los casos simulados y analizar los resultados;
3. Calibrar el modelo numérico con los resultados alcanzados experimentalmente en el canal de oleaje del LH;
4. Evaluar la replicabilidad de los resultados en futuros proyectos similares.

## 3 METODOLOGÍA

El LH de la UNC cuenta con un canal de oleaje vidriado equipado de un generador bidimensional con absorción dinámica, capaz de simular curvas teóricas de oleaje regular e irregular con períodos y amplitudes predeterminados. Este canal tiene 23.5 m de largo, 0.8 m de altura y 0.5 m de ancho ([Figura 1](#)), pudiendo cumplir dos funciones, tanto como canal de pendiente variable y como canal de oleaje.

En este trabajo se realizó la propagación de una ola monocromática a lo largo del canal de oleaje mediante el movimiento de la pala generadora de oleaje. En particular, se trabajó sobre el proceso de asomeramiento del oleaje en la simulación de una playa de pendiente constante con ambos modelos. Para alcanzar esta condición de ensayo, el canal, ha sido subdividido en dos tramos ([Figura 2](#)):

1. Un tramo plano de 10 m de largo,
2. Una rampa de 3 m de largo,



Figura 1. Canal de oleaje del LH (izquierda) y un sensor resistivo (derecha).

Las alturas de ola se registraron a través de 2 sensores resistivos (Figura 1) colocados sobre el canal según esquema de la Figura 2. Uno de ellos se colocó a lo largo del primer tramo plano, a 5.3 m de la pala generadora, para separar y medir la ola incidente y reflejada por la estructura y por la playa disipativa esquematizada con la rampa a pendiente constante. El segundo sensor se ubicó antes de la rampa, a 10 m desde la pala generadora.

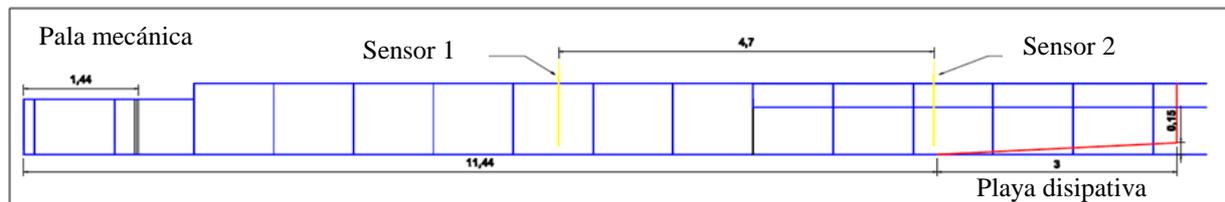


Figura 2. Configuración geométrica del canal para la realización de los ensayos, con ubicaciones de los sensores resistivos.

La misma geometría del ensayo alcanzada en el canal, se simuló con el modelo numérico que utiliza el método SPH, implementando el modelo DualSPHysic utilizando el lenguaje CUDA (Compute Unified Device Architecture). CUDA es una arquitectura de cálculo paralelo de NVIDIA que aprovecha la potencia de la GPU (Graphics Processing Unit) con algunas extensiones al lenguaje C/C++, para proporcionar un incremento extraordinario del rendimiento del sistema. Se logra disminuir el tiempo de simulación respecto de las simulaciones realizadas con las convencionales CPU (Central Proceses Unit).

En el modelo, se aplicó a la pala un movimiento rectilíneo sinusoidal, y se reprodujo exactamente el mismo movimiento del generador. Se definieron dos puntos de mediciones (sensores virtuales) en el modelo numérico en las mismas posiciones que en el canal (Figura 2). El modelo numérico se calibró con los resultados de los ensayos realizados en el canal de oleaje.

### 3.1 El método SPH

El SPH es un método que permite la resolución de problemas de dinámica computacional continua. Las técnicas que utilizan una malla para el estudio de la dinámica de fluidos discretizan el medio continuo, adaptándose a esa malla que se extiende a todo el dominio de integración. Esto implica que se simularán de igual manera los espacios vacíos y los espacios en los que se encuentra el fluido.

En contraposición, los métodos lagrangianos no necesitan de una malla regular que cubra la totalidad del espacio de integración. En este sentido, el volumen ocupado por el fluido es el único que se computa, por tanto, la memoria y el tiempo de cálculo decae.

El SPH integra las ecuaciones de movimiento de la dinámica de fluidos en cada punto de la malla numérica en el formalismo lagrangiano, calculando los valores de las cantidades físicas

relevantes (velocidad, posición, densidad y presión) para cada partícula como una interpolación de los valores de las partículas vecinas.

Para aumentar la resolución del método, el paso de la distribución de fluido (que es un continuo) a partículas (medio discreto) se realiza a través de una función de suavizado, kernel. Esta función tiene un soporte compacto dentro de una región que viene determinada por un radio mucho más pequeño que la escala típica del problema que se está tratando.

El proceso de suavizado modula las contribuciones de las partículas vecinas que intervienen en la interpolación, por tanto, estas contribuciones pueden ser entendidas como la interacción entre pares de partículas.

Las principales ventajas de esta técnica se derivan de su naturaleza lagrangiana. SPH sigue automáticamente flujos complejos y puede fácilmente mantener constante la resolución de la masa. Los resultados mejoran notablemente si se utiliza una longitud suavizada ( $h$ ) que tenga una variación espacial, manteniendo un número constante de píxeles vecinos dentro de un radio aproximadamente igual a  $(2h)$ . Otras ventajas a mencionar son:

- Se manejan fácilmente los problemas multidimensionales así como los que no tienen ninguna simetría.
- El método sólo calcula donde la masa está localizada, por tanto, no se invertirá tiempo en el tratamiento de los espacios vacíos.
- Es simple incluir cualquier proceso físico adicional en el código base.
- Reproduce fenómenos con deformación de la ola.

Las desventajas más importantes son:

- Las condiciones de contorno son normalmente difíciles de implementar. Uno de los problemas que puede aparecer debido a un mal tratamiento de los contornos es la penetración de las partículas fluidas en los contornos, lo cual debe evitarse.
- Los fenómenos de choques no son la mejor aplicación de este método, ya que necesitan de una viscosidad artificial adecuada para resolver los choques.
- SPH trabaja mejor si el ancho del kernel es variable, así la resolución será mejor en las regiones de alta densidad. Las regiones de baja densidad estarían peor resueltas.
- La robustez del método puede llegar a ser una desventaja, ya que si se desarrolla una solución que no tiene sentido físico, ésta puede pasar inadvertida.
- El método de interpolación usado en el SPH es muy simple, por lo que está muy afectado por el desorden de las partículas. El SPH da resultados razonables para los gradientes de primer orden, pero pueden ser peores para derivadas de orden superior.
- Se suele emplear un kernel con simetría esférica. La distribución de los vecinos de una partícula tiene que ser aproximadamente isotrópica para un correcto funcionamiento de la fórmula de interpolación. Esto no se cumple cuando se forman láminas o discos.

### 3.2 El código DualSPHysics y el uso de la GPU

DualSPHysics ha sido desarrollado en C++ y en CUDA, puede llevar a cabo simulaciones en CPU y GPU y permite simular los grandes problemas de ingeniería utilizando modelos SPH. Las unidades de procesamiento gráfico (GPU) son una alternativa económica para optimizar el modelado numérico. Las GPU están diseñadas para manejar grandes cantidades de datos, lo que disminuye el tiempo de procesado, respecto de las convencionales unidades centrales de procesamiento (CPU), optimizando el uso de la memoria.

La GPU, está diseñada para la regeneración de imágenes y el control de los múltiples monitores que se pueden utilizar (Domínguez et al., 2013). Su función es calcular operaciones de punto flotante para ir generando el trazado del escenario y el modelado de objetos en 3D, esto lo logra de forma más eficiente debido a que tiene una estructura de programación en

paralelo, lo que permite que cada parte del escenario se calcule en forma separada. Por lo tanto, la capacidad de cálculo de las GPU también puede ser aplicada para los métodos SPH.

Por otra parte, la CPU hace todos los procesos y cálculos necesarios para desplegar la información que se necesite. A diferencia de la GPU, que cuenta con una estructura de ejecución paralela, la CPU está diseñada para ejecutar sus funciones de forma secuencial, además de que un CPU tiene una estructura que le permite resolver diferentes algoritmos lo que le permite ejecutar una diversidad de aplicaciones. Además, las CPU han evolucionado al concepto de ‘multicore’, lo que le permite realizar operaciones en paralelo utilizando varios núcleos (Figura 3).

Las optimizaciones aplicadas a la ejecución de la CPU con DualSPHysics son:

- La aplicación de la simetría en la interacción entre partículas.
- Dividir el dominio en celdas más pequeñas (Figura 4).
- Multinúcleos programando con OpenMP.

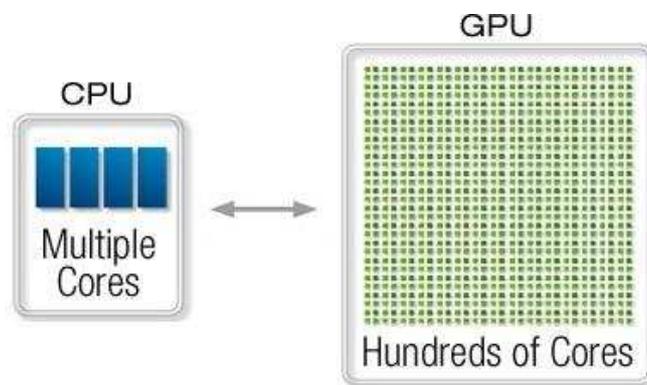


Figura 3: Comparación de la cantidad de núcleos disponibles entre CPU y GPU.

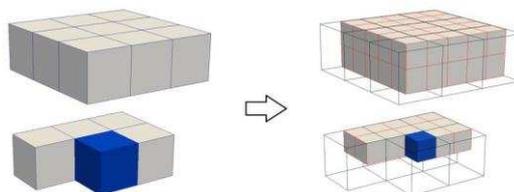


Figura 4: División del dominio en celdas más pequeñas.

### IMPLEMENTACIÓN GPU

La técnica más eficiente para aumentar las velocidades de las corridas consiste en minimizar las comunicaciones para las transferencias de datos entre la CPU y la GPU. Con la implementación de GPU sólo una transferencia CPU-GPU es necesaria en el comienzo de la simulación, mientras que los datos relevantes serán transferidos a la CPU cuando se requiere guardar los datos de salida (normalmente con poca frecuencia).

La implementación GPU presenta algunas diferencias clave en comparación con la versión con CPU. La diferencia principal es la ejecución paralela de todas las tareas que pueden ser paralelizadas, como los mismos bucles de cada partícula a lo largo de la simulación. Un ‘hilo de ejecución’ (thread) en la GPU calcula la fuerza resultante de una partícula al realizar todas las interacciones con sus vecinos. La simetría de la interacción de partículas se emplea en la CPU reduciendo el tiempo de ejecución, pero no se aplica en la aplicación GPU ya que no es

eficaz debido a problemas de coalescencia de memoria.

Sólo en DualSPHysics la misma aplicación se puede ejecutar utilizando la CPU o GPU, lo que facilita el uso del código no sólo en ordenadores con una GPU Nvidia, sino también en máquinas sin GPU CUDA. El código principal tiene una base común para CPU y GPU, con pequeñas diferencias del código fuente. Por lo tanto, la depuración o el mantenimiento es más fácil y las comparaciones de los resultados son más directas.

Un completo y eficaz uso de todas las capacidades de la GPU no es sencillo y por otro lado, la implementación del GPU presenta varias limitaciones, principalmente debido a la naturaleza lagrangiana del método SPH. Por lo tanto, las transferencias GPU-CPU deben ser minimizadas, el código de divergencia minimizado y los accesos a la memoria no coalescente limitarlos ya que otorgan un menor rendimiento. Por lo tanto las siguientes optimizaciones fueron desarrolladas para evitar o minimizar estos problemas:

- La plena aplicación en la GPU.
- Maximizar la ocupación del GPU.
- Simplificar la búsqueda del vecino.
- División del dominio en celdas más pequeñas.

## 4 CASO EXPERIMENTAL EN LABORATORIO

### 4.1 Definición de los parámetros usados en el modelo

Algunas de las variables más importantes utilizadas durante la compilación han sido:

- Se utilizó un tramo plano del canal de 10 metros de longitud y luego el desarrollo de una playa con 5 grados de pendiente hasta completar 13 metros (Figura 2).
- Se utilizó un espaciamiento entre partículas ( $\Delta x$  y  $\Delta z$ ) igual a 0.02 m, con una distancia (h) de Kernel igual a 0.03 m, obtenido con la siguiente ecuación:

$$h = 0.866025 * \sqrt[2]{dx^2 + dy^2 + dz^2} \quad (1)$$

- Se utilizó la formulación de kernel cuadrática:

$$W(r, h) = \alpha_D \left[ \frac{3}{16} q^2 - \frac{3}{4} q + \frac{3}{4} \right] \quad 0 \leq q \leq 2 \quad (2)$$

La función de kernel depende de la longitud de suavizado h y de la distancia adimensional entre las partículas dadas por  $q = r/h$ , donde r es la distancia entre las partículas A y B (Figura 5);  $\alpha_D$  es  $1/(\pi h^2)$  en bidimensional (2D) y  $5/(4\pi h^3)$  en tridimensional (3D).

- La viscosidad artificial se estimó con un valor igual a 0.001.
- El paso de tiempo de cálculo escogido fue 0.0001 s, mientras que el paso de tiempo de muestreo fue de 0.1 s, con un tiempo total igual a 30 s (es decir un total de 300 fotogramas).
- El tiempo total de simulación fue de 2.5 horas.
- Se aplicó un filtro de densidad Shepard = 5.
- Se aplicó la condición de contorno de Tipo III, para los contornos sólidos, que asume la existencia de unas partículas de contorno que verifican las mismas ecuaciones de continuidad y de estado que las partículas fluidas, pero permanecen fijas en su posición a lo largo del tiempo (Dalrymple and Knio, 2000; Cerqueiro et al., 2004).

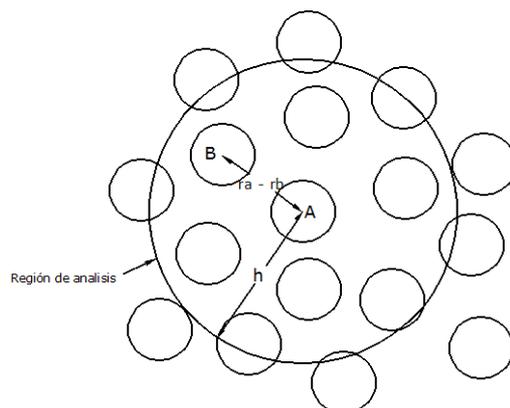


Figura 5: Captura a los 10 s del caso planteado.

## 4.2 Movimiento de la pala

El programa utilizado permite establecer distintos tipos de movimientos, ya sean movimientos rectilíneos uniformes o acelerados, o bien rectilíneos sinusoidal, rotacional sinusoidal o circular sinusoidal entre otros. Para el caso en examen, se aplicó a la pala un movimiento rectilíneo sinusoidal, para reproducir exactamente el mismo movimiento que el generador del oleaje monocromático en el canal.

Para los casos modelados se utilizó en cada movimiento una duración de 5 segundos, una frecuencia de 0.5 Hz en “x” (igual que en el generador de oleaje;  $T = 2$  s), una amplitud de 0.056 m (definido por la ecuación de la dispersión), y una fase igual a  $3/2\pi = 4.71239$  (este valor permite que los movimientos respectivos se encuentren en fase).

## 4.3 Visualización de resultados

Utilizando el programa “ParaView”, se visualizó el comportamiento del fluido en ‘3D’ y la variación de altura de la superficie libre en un punto determinado. A su vez, se determinaron las variaciones y puntos máximos en X e Y. Esta metodología permitió determinar la variación de la superficie libre para el caso presentado, y de esta manera comparar los resultados obtenidos tanto con el modelo numérico como con el generador de oleaje.

## 4.4 Calibración

Se realizó una simulación con la geometría ya descrita y los parámetros por defecto (Dalrymple et al., 2006), cuyos resultados de altura de ola se presenta en la Figura 6.

En esta Figura 6, se observó un rápido decaimiento de la altura de ola. Por esta razón, se decidió modificar la viscosidad, desarrollando de esta forma olas menos amortiguadas. La viscosidad artificial inicial con la cual se trabajó fue de 0.1 (este valor es igual a  $\alpha$ , el cual es un parámetro libre dentro de la formulación de viscosidad artificial en función de cada problema). Se modificaron varios valores de este parámetro (dejando los demás constantes), encontrando los resultados que se representan en la Figura 7.

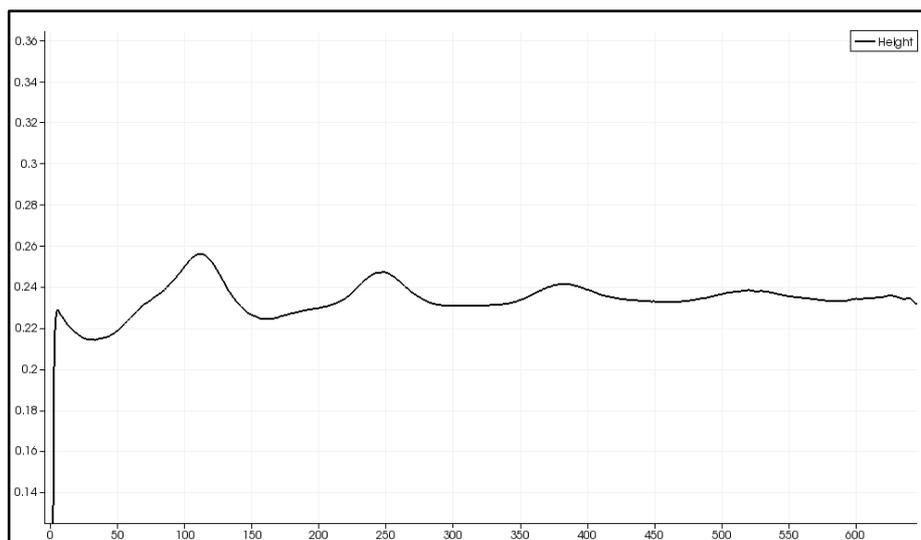


Figura 6: Captura a los 10 s del caso planteado. Eje vertical: Altura de ola (m); Eje Horizontal: Tiempo\*0.1 (s)

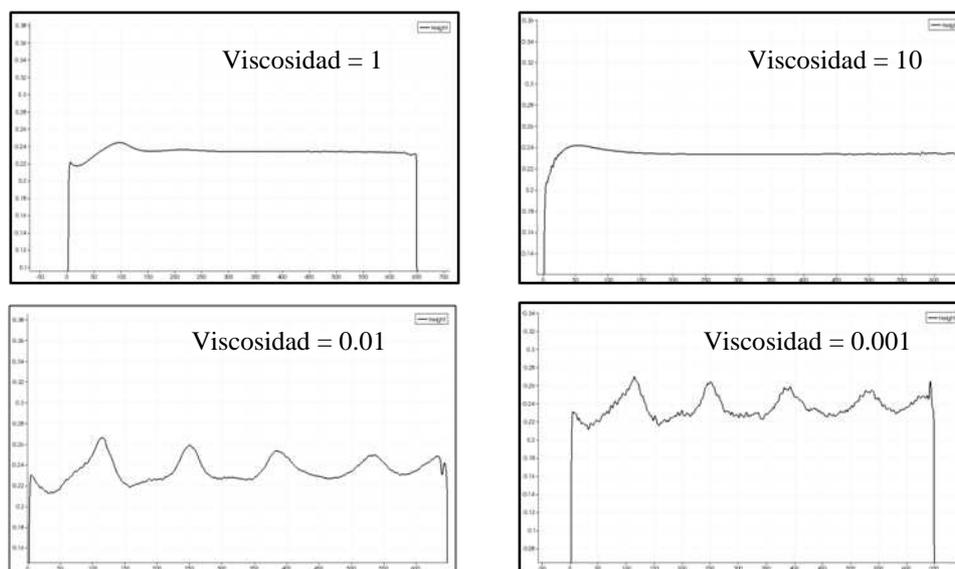


Figura 7: Variación del parámetro 'VISCO' en el caso ensayado (en abscisas 'espacio' y en ordenadas 'altura'). Eje vertical: Altura de ola (m); Eje Horizontal: Tiempo\*0.1 (s)

Observando los resultados obtenidos, tanto la altura de ola como la regularidad del oleaje, se decidió continuar trabajando con un valor de viscosidad igual a 0.001, ya que estos valores eran aceptables. El problema que se planteó resolver a partir de esta instancia, fue deshacer las pequeñas irregularidades que se aprecian en la superficie libre. Se trabajó con varios parámetros para determinar cual producía este "ruido" generado, luego de lo cual se halló que modificando el parámetro 'Shepard density filter' se lograba suavizar la superficie, obteniendo la gráfica de la [Figura 8](#).

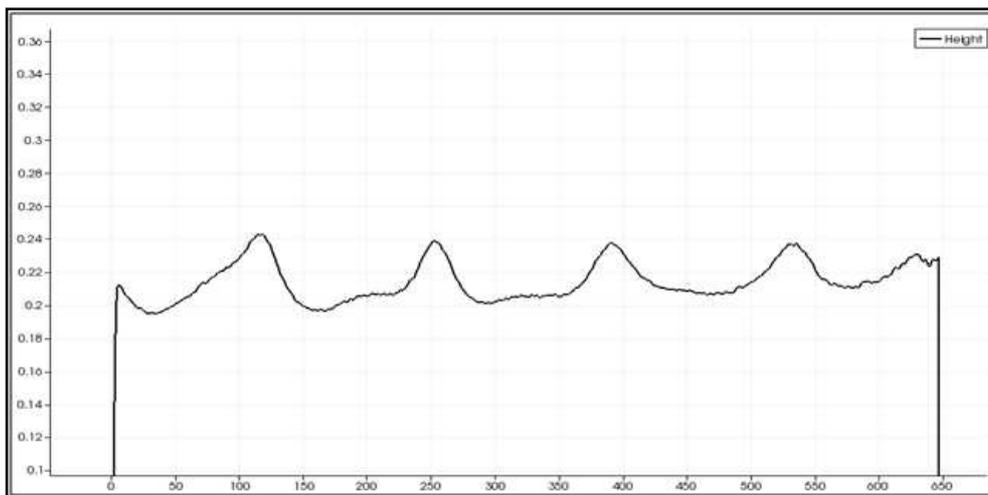


Figura 8: Captura luego de suavizar con el parámetro 'Shepard density filter'. Eje vertical: Altura de ola (m); Eje Horizontal: Tiempo\*0.1 (s)

El valor que presentaba el mejor comportamiento del oleaje fue de 5, ya que para valores mayores los resultados no cambiaron. A partir de aquí, se ajustaron algunos parámetros para reproducir exactamente el caso generado en la pala (como puede ser el calado, el tiempo de simulación, entre otros).

#### 4.5 Resultados obtenidos

A partir de las variables ya definidas, se muestran los resultados obtenidos con un calado de 0.1225 m, distancia entre partículas igual a 0.02 m y 30 s de simulación en la [Figura 9](#) (sensor virtual 1) y en la [Figura 10](#) (sensor virtual 2).

Analizando los gráficos obtenidos, se observó que:

- El oleaje obtenido no es regular, sino que posee un perfil simétrico con doble pico. La altura de ola en promedio es de 3 cm, distinta al valor teórico. La variación de la superficie libre del Sensor 1 presenta doble pico, con una altura de ola 3.5 cm (más similar al valor teórico), y una variación de contorno producto de la reflexión de la ola en el fondo inclinado.
- En cuanto a la variación de la superficie libre del sensor 2, la pérdida de energía del oleaje es mayor, dando una altura promedio de ola de 2.5 cm. A su vez, la influencia de la reflexión es importante, la cual comienza a manifestarse posteriormente al contacto de la ola con el tramo inclinado.

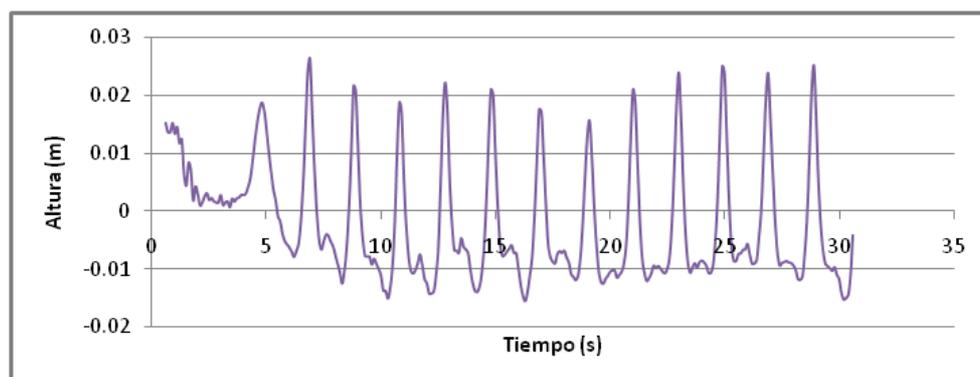


Figura 9: Variación de la superficie libre medida con el Sensor virtual 1.

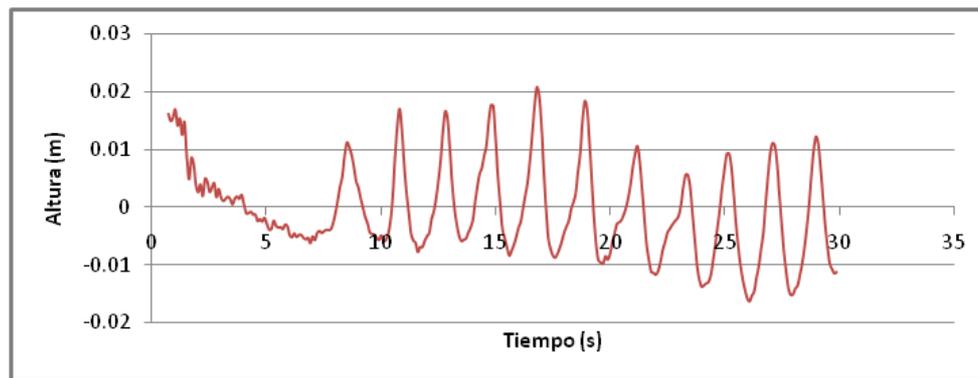


Figura 10: Variación de la superficie libre medida con el Sensor virtual 2.

Analizando la comparación de gráficos, se determinó:

- Con respecto al Sensor 1 (Figura 11), una clara influencia de la reflexión del fondo inclinado sobre la superficie libre, aproximadamente desde los 14 s. Se presentan problemas en la simulación de los valles del tren de ondas, dando resultados irregulares. A partir de los 22 segundos las diferencias disminuyen, dando resultados mejores.
- Con respecto al Sensor 2 (Figura 12), la influencia del plano inclinado es mucho más importante, generando un desfase entre ambas ondas. En algunos instantes (entre los 16 y 18 segundos) se puede observar que ambos gráficos se encuentran en fase, y a partir de donde se puede contemplar que las alturas de olas generadas son similares (diferencias mínimas de 2.5 mm). Cabe recordar que a estas distancias de la pala generadora, la altura de ola teórica tanto en el generador como en el programa, es muy difícil de lograr, debido a la influencia tanto del fondo como del tramo inclinado.

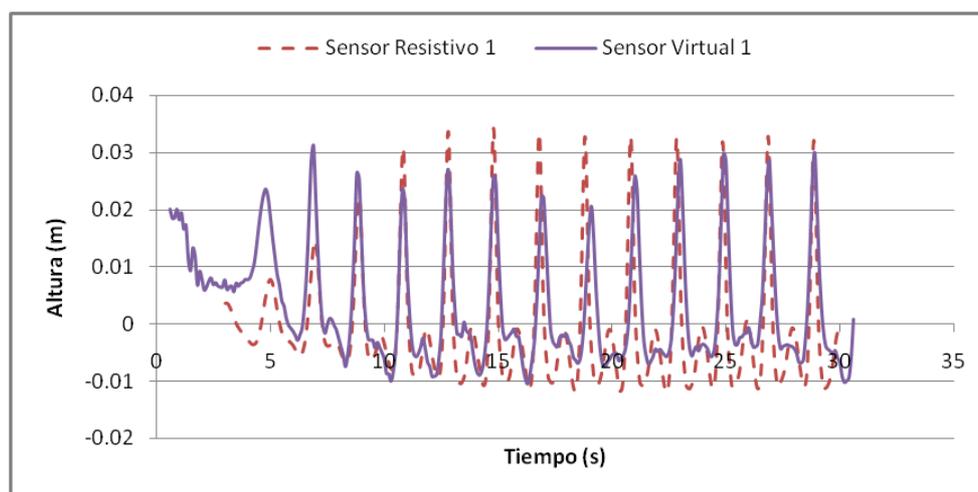


Figura 11: Comparación de la variación de la superficie libre del Sensor 1.

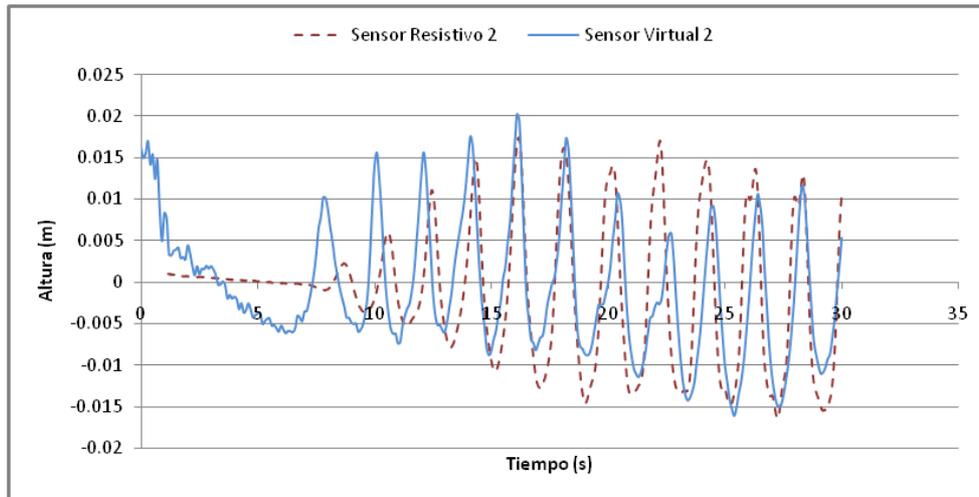


Figura 12: Comparación de la variación de la superficie libre del Sensor 2.

Se optimizó la similitud entre los sensores, a través de un valor menor de la viscosidad artificial del agua. Se optó por utilizar el valor de 0.000001 (con un valor del ‘Shepard density filter’ igual a 5), y se aumentó el tiempo de simulación a 60 segundos, para tener una mejor comparación. Los resultados obtenidos fueron los siguientes:

Analizando los gráficos, se determinó:

- Con respecto al Sensor 1 (Figura 13), se logró mejorar la altura de ola, generando valores similares al valor teórico. Aunque continúan existiendo irregularidades en la zona de valles, los valores de los picos menores coinciden entre ambos gráficos.
- Con respecto al Sensor 2 (Figura 14), si bien continúa existiendo una influencia de la reflexión, ésta es de menor valor. Aunque se pueden observar pequeñas variaciones en el perfil de las crestas, en algunos puntos llegan a generarse ondas idénticas (por ejemplo entre los 45 y 50 segundos).

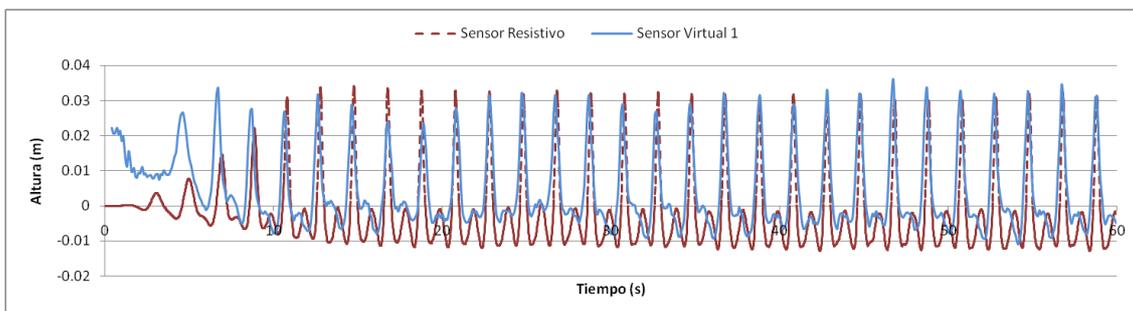


Figura 13: Comparación de la variación de la superficie libre del Sensor 1 utilizando una menor viscosidad.

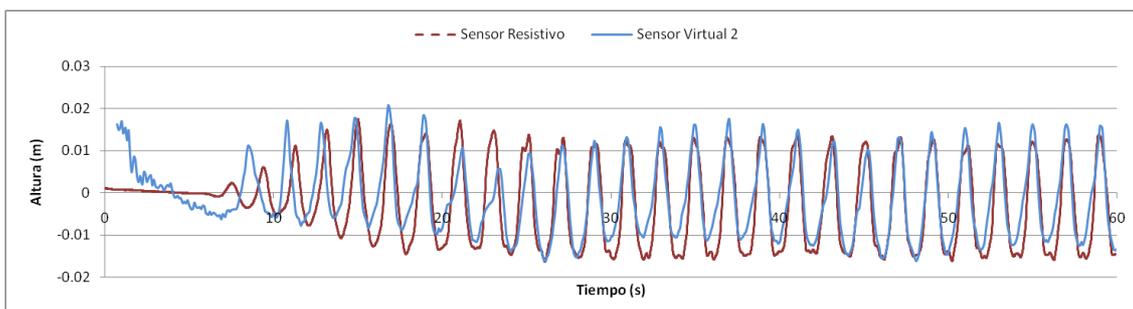


Figura 14: Comparación de la variación de la superficie libre del Sensor 2 utilizando una menor viscosidad.

#### 4.6 Análisis de errores

Se comparó un intervalo de 60 s de resultados en el Sensor 1 (Tabla 1). Se calcularon los valores promedios para cada serie analizada, obteniendo en el primer sensor un valor promedio de 0.03041 m y para el segundo 0.03141 m, valores muy próximos que difieren un 4%. Para el cálculo del error se utilizó la expresión del error medio cuadrático (Ec. (3)) sobre los valores de las crestas de las olas. No se tuvieron en cuenta los desfases iniciales. El error medio cuadrático (RMS “root mean square”) para el Sensor 1 fue estimado en 4 mm.

$$RMS = \sqrt{\frac{\sum_{i=1}^n (e_i - \bar{e})^2}{n}} \quad (3)$$

Por otra parte, comparando un intervalo de 60 s de resultados en el Sensor 2, se obtuvieron los resultados presentados en la Tabla 2. Se calcularon los valores promedios, resultando para el primer sensor un valor de 0.01429 m y para el segundo 0.01300 m, valores a su vez muy próximos que sólo difieren un 9%. El error cuadrático medio para el Sensor 2 fue estimado en 5 mm.

Tiempo (s)	Altura de Superficie Libre (m)		Diferencias (m)
	Sensor Virtual	Sensor Resistivo	
8.9	0.027531	0.022164	2.88047E-05
10.9	0.026718	0.030821	1.68346E-05
12.8	0.031641	0.033611	3.8809E-06
14.8	0.028701	0.034185	3.00743E-05
16.9	0.023799	0.033447	9.30839E-05
19.0	0.023517	0.032729	8.48609E-05
20.9	0.027662	0.03277	2.60917E-05
23.0	0.03145	0.032503	1.10881E-06
24.9	0.031991	0.031826	2.7225E-08
26.9	0.031541	0.03279	1.56E-06
28.9	0.031369	0.032052	4.66489E-07
30.9	0.028961	0.032031	9.4249E-06
32.9	0.026784	0.032298	3.04042E-05
34.9	0.029027	0.031785	7.60656E-06
36.9	0.032108	0.031621	2.37169E-07
38.9	0.031575	0.029898	2.81233E-06
41.0	0.028817	0.031764	8.68481E-06
43.0	0.033077	0.030964	4.46477E-06
44.9	0.031877	0.030964	8.33569E-07
46.9	0.036137	0.030267	3.44569E-05
48.9	0.033701	0.03	1.36974E-05
50.9	0.032777	0.030164	6.82777E-06
52.9	0.031899	0.030821	1.16208E-06
54.9	0.032653	0.031826	6.83929E-07
56.9	0.034414	0.032052	5.57904E-06
58.9	0.031046	0.031334	8.2944E-08

Tabla 1: Comparación de las mediciones entre el sensor virtual 1 y el sensor resistivo 1.

Tiempo (s)	Altura de Superficie Libre (m)		Diferencias (m)
	Sensor Virtual	Sensor Resistivo	
10.3	0.017071	0.006055	0.00012135
12.3	0.016677	0.011122	3.0858E-05
14.2	0.017736	0.015033	7.3062E-06
16.1	0.020816	0.017444	1.137E-05
18.2	0.018442	0.016229	4.8974E-06
20.4	0.010579	0.01408	1.2257E-05
22.5	0.005589	0.01708	0.00013204
24.4	0.003505	0.014749	0.00012643
26.4	0.011109	0.013675	6.5844E-06
28.3	0.012245	0.013027	6.1152E-07
30.3	0.013176	0.011588	2.5217E-06
32.3	0.015568	0.011203	1.9053E-05
34.2	0.016042	0.012074	1.5745E-05
36.2	0.017456	0.012945	2.0349E-05
38.2	0.016298	0.01327	9.1688E-06
40.2	0.014923	0.013148	3.1506E-06
42.4	0.012194	0.013412	1.4835E-06
44.4	0.010128	0.013391	1.0647E-05
46.4	0.013101	0.012054	1.0962E-06
48.3	0.014345	0.013128	1.4811E-06
50.3	0.015428	0.012702	7.4311E-06
52.3	0.016634	0.011284	2.8623E-05
54.2	0.016192	0.011547	2.1576E-05
56.2	0.016257	0.012074	1.7497E-05
58.2	0.015878	0.012702	1.0087E-05

Tabla 2: Comparación de las mediciones entre el sensor virtual 2 y el sensor resistivo 2.

## 5 CONCLUSIONES

Se compararon los resultados experimentales de un oleaje monocromático generado por una pala mecánica en un canal de oleaje bidimensional instalado en el Laboratorio de Hidráulica de la Universidad Nacional de Córdoba, con los resultados del mismo oleaje generado con un modelo numérico (DualSPHysics) que usa la técnica SPH.

Se logró un buen ajuste de los resultados mediante calibración de las variables del modelo (en particular de la viscosidad artificial y del filtro de densidad “Shepard”), resultados que se asemejan al caso de las simulaciones realizadas con el canal de oleaje. La comparación de los resultados del modelo con las realizaciones de simulaciones de oleaje en el canal del Laboratorio de Hidráulica dio como resultados errores cuadráticos medios del orden de 4 mm (en el Sensor 1) y de 5 mm (en el Sensor 2), ambos calculados sobre los valores de las crestas de las olas en series de 60 s de duración.

Se continúa trabajando en las evaluaciones del software con nuevas calibraciones y con casos de geometrías más complejas.

## Agradecimientos

Los autores agradecen al Ministerio de Ciencia y Tecnología de la Provincia de Córdoba, que permitió realizar este trabajo en el marco del “Estudio de la Cota de Inundación Máxima Probable en la Zona Sur Sur-Oeste de la Laguna de Mar Chiquita”, desde el Laboratorio de Hidráulica de la Universidad Nacional de Córdoba.

## 6 REFERENCIAS

- Cerqueiro D., Zou S., Gómez-Gesteira M, and Dalrymple R.A. Boundary conditions generated by static particles in SPH methods. *J. Comput. Phys*, 2004.
- Crespo A.J.C., Dominguez J.M., Gómez-Gesteira M., Barreiro A. and Rogers B.D. User Guide for the DualSPHysics Code v1.2, 2011.
- Dalrymple R.A., Knio O. SPH modelling of water waves. *Proc. Coastal Dynamics*, Lund, 779–787, 2000.
- Dalrymple R.A. and Rogers B.D. Numerical Modeling of Water Waves with the SPH Method, *Coastal Engineering*, 53/2-3, 141-147, 2006.
- Domínguez J.M., Crespo A.J.C. and Gómez-Gesteira M. Optimization strategies for CPU and GPU implementations of a smoothed particle hydrodynamics method. *Computer Physics Communications*, 184(3): 617-627, 2013.
- Gingold A. and Monaghan J. J. Smoothed Particle Hydrodynamics: theory and application to non-spherical stars. *Mon. Not. R. Astr. Soc.* 181, 375-389, 1977.
- Lucy L. A numerical approach to the testing of the fission hypothesis. *Astron. J.* 82(12), 1013-1024, 1977.