

GPU-ACCELERATED SPH MODEL FOR WATER WAVES AND OTHER FREE SURFACE FLOWS

Robert A. Dalrymple¹, Alexis Hérault², Giuseppe Bilotta³, and Rozita Jalali Farahani¹

This paper discusses the meshless numerical method Smoothed Particle Hydrodynamics and its application to water waves and nearshore circulation. In particular we focus on an implementation of the model on the graphics processing unit (GPU) of computers, which permits low-cost supercomputing capabilities for certain types of computational problems. The implementation here runs on Nvidia graphics cards, from off-the-shelf laptops to the top-of-line Tesla cards for workstations with their current 480 massively parallel streaming processors. Here we apply the model to breaking waves and nearshore circulation, demonstrating that SPH can model changes in wave properties due to shoaling, refraction, and diffraction and wave-current interaction; as well as nonlinear phenomena such as harmonic generation, and, by using wave-period averaged quantities, such aspects of nearshore circulation as wave set-up, longshore currents, rip currents, and nearshore circulation gyres.

Keywords: SPH; meshfree methods; wave modeling; surf zone hydrodynamics, rip current

INTRODUCTION

The modeling of water waves using the meshless Lagrangian numerical method Smoothed Particle Hydrodynamics (SPH) has been done since 1994 (Monaghan, 1994). The method (originally developed for astrophysics by Lucy (1977) and Gingold and Monaghan (1977)) involves many irregularly spaced nodes (particles) dispersed throughout the solution domain. Each particle has a fixed mass, m , associated with a volume of fluid of density ρ , which means that mass is automatically conserved as long as particles remain in the appropriate fluid domain. Particle forms of the Navier-Stokes equations are used to determine the fluid acceleration at the location of the node, and then the node is moved according to that acceleration (e.g., Monaghan (1994)). The method, therefore, is Lagrangian and the acceleration is the total acceleration that includes the nonlinear convective acceleration terms. The total derivative removes the nonlinear complications of the convective terms that appear in Eulerian methods. Since the SPH nodes move with the fluid, splashing or other large deformations of the free surface are easily modeled.

In addition to the equations of motion and conservation of mass, there is an additional equation relating fluid density (ρ) to pressure (p). This equation of state results in a weakly compressible fluid model—which has the advantage of a simple calculation for pressure (rather than solving a Poisson-type partial differential equation), but has the downside that the model time step is dictated by the speed of sound in the fluid. Monaghan in 1994 suggested that sound speed can be slowed (to get bigger time steps) without affecting the model results (but no slower than ten times any fluid velocity in the model). For wave problems, a minimum sound speed of ten times the wave phase velocity is also a good criterion, Dalrymple and Rogers (2006a).

The nodes or particles in SPH are smoothed, that is to say, their properties are smoothed out over a volume of a size proportional to a quantity of length h , where h is typically slightly larger than the initial particle spacing in the model. For example, if a fifth-order Wendland kernel is used (Wendland, 2005), the particle has a radius of influence of $2h$. As a consequence, a given particle is only influenced by its neighbors and not by particles at large distances.

For an example of the smoothing, to determine the value of the variable $A(\mathbf{r})$ at location \mathbf{r} , the following summation interpolant is used:

$$A(\mathbf{r}) = \sum_j^N A(\mathbf{r}_j) W(\mathbf{r} - \mathbf{r}_j, h) \frac{m_j}{\rho_j} \quad (1)$$

where $A(\mathbf{r}_j)$ is the value of $A(\mathbf{r})$ for particle j in the neighborhood of the point of interest, m_j/ρ_j is the volume of the SPH particle, and $W(\mathbf{r} - \mathbf{r}_j, h)$ is the kernel function, which decreases monotonically for as its first argument increases. For the Wendland kernel (in 3D) the definition is

$$W(\mathbf{r} - \mathbf{r}_j, h) = \frac{15}{16\pi h^3} \left(1 - \frac{q}{2}\right)^4 (2q + 1) \text{ for } q = \frac{r}{h} \text{ and } q < 2 \quad (2)$$

¹ Department of Civil Engineering, Johns Hopkins University, 3400 No. Charles St., Baltimore, MD 21218

² INGV-Sezione di Catania, Piazza Roma, 2 95134 Catania, Italy

³ Dipartimento di Matematica e Informatica, Università degli Studi di Catania, Italy

The kernel is set to zero for r greater than $2h$, so that particles separated by larger distances do not interact.

The numerical interpolation described in Eqn. 1 can be considered as a Moving Least-Square approach to calculating fluid properties at any point (most SPH schemes use either constant interpolation, as above, or linear interpolation (Colagrossi and Landrini (2003))).

To facilitate the computations, neighbor lists are generated to keep track of the particles within several particle spacings of each particle. This is carried out by conceptually developing a virtual grid comprised of cells of $2h$; this neighbor list grid extends over the full computational domain. Based on the position of a particle in space, the grid location of the particle is immediately known by dividing each of its coordinates by $2h$. Then, after determining the cell locations for each particle, the occupants of each cell are then neighbors, as are the occupants of the adjacent cells. Therefore a block of 9 continuous cells in 2D or 27 cells in 3-D comprises the neighborhood of a particle in the center cell. The numerical interpolation above for a given particle is then carried out using the neighbors only. The neighbor list is updated often (from every time step to every tenth time step, depending on the problem) as particles move in and out of a neighborhood.

Recent reviews of SPH have recently been published for more details (Dalrymple *et al.* (2010), Gómez-Gesteira *et al.* (2010), Monaghan (2005)).

Some recent advances in the methodology include the open source software SPHysics (www.sphysics.org), which was shown to accurately model waves impacting on structures (Gómez-Gesteira and Dalrymple (2004)) and to model waves in the surf zone realistically (Dalrymple and Rogers (2006b)). This model (written in FORTRAN) includes many options for the user in terms of different interpolating kernels, integration schemes, and boundary conditions, and comes with a detailed users manual. The original version of SPHysics was written for a single processor machine. Since that time, numerous advances in modeling technology and hardware have occurred and the computational applications of the SPH method have increased dramatically. First, the use of cluster computing and parallel processing has increased dramatically the processing power of SPHysics, allowing for larger problems to be examined. Whereas it used to be difficult and time-consuming (weeks) to compute solutions with 100,000 particles on a single CPU, now solutions with millions of particles can be computed in hours. ParaSPHysics (coded by B. Rogers) is the parallel version of SPHysics and is open source. Recently, a paradigm shift in parallel processing, involving moving very intensive numerical calculations from the CPU to the Graphics Processing Unit (GPU), has dramatically reduced the price of parallel computing and brought about low-cost supercomputing. Millions of particles can be run on a single work station with the appropriate graphics card within hours.

By programming SPH in CUDA (a C++ extension developed by the Nvidia company for their graphics cards) or now OpenCL for other graphics card manufacturers, the power of hundreds of streaming processors on a single graphics card can be harnessed. For example, the Tesla C1060 card from Nvidia contains 240 streaming processors, and the new Tesla C 2050 contains 480 processors. However, less expensive and more readily accessible computers, such as laptops, are available with 48 processors (MacBookPro, for example), which is comparable to a small cluster computer.

Note that there is a potential drawback to GPU computing—there is a limited amount of memory on the graphics card. For example, the C1060 has 4 Gb of memory to share among the 240 processors (~16 Mb/processor). This can limit the types of problems that can be used on the GPU. Fortunately, SPH is computationally intensive (good for GPU) and data sparse—needing about 1 kB per particle.

GPU-SPHysics is a GPU-accelerated model based on SPHysics, Hérault *et al.* (2010, 2009), which has many options for interpolation kernels, boundary conditions, density smoothing, etc. The code is written in CUDA-extended C++, and numerous convenient objects, such as cubes, cylinders, and spheres, have been defined for use in the model. Dalrymple and Hérault (2009) have used it to examine the breaching of flood walls and the interaction of the flooding waters with houses (see link at www.ce.jhu.edu/dalrymple/GPU). Here we use GPU-SPHysics to model waves in the surf zone on sloping beaches. The cases are waves breaking on a planar beach, waves breaking onto objects in the surf zone such as cylinders and rectangles, and waves breaking over a sand bar with a rip channel. Simulations are done with over three million particles, which can be considered as the count of nodal points within the flow field and hence a measure of resolution.

One of the strong points of GPU-SPHysics is that simulations are displayed graphically with OpenGL while the simulation is running. In addition, screenshots of the simulation can be captured at

prescribed times. This real-time feedback helps with debugging models and provides a convenient way to create movies of the results just after the run is completed.

APPLICATIONS

To illustrate the versatility of the method, a numerical wave tank was used with a wavemaker at one end. The wavemaker is a plane of particles hinged at or below the bottom that moves in a sinusoidal motion about the hinge to create a flap wavemaker. The wave tank bottom is sloping with a short dry beach section at the opposite end of the tank. The first test was simply waves breaking on the beach using 2.8 million SPH particles. These results are shown Figure 1, showing that the waves break near the wavemaker and then propagate (towards the lower right in the figure) to the beach as bores. In this case each broken wave is clearly a turbulent bore, but more importantly there are coherent turbulent structures that are left behind the traveling bore as evidenced by the irregularities in the free surface.

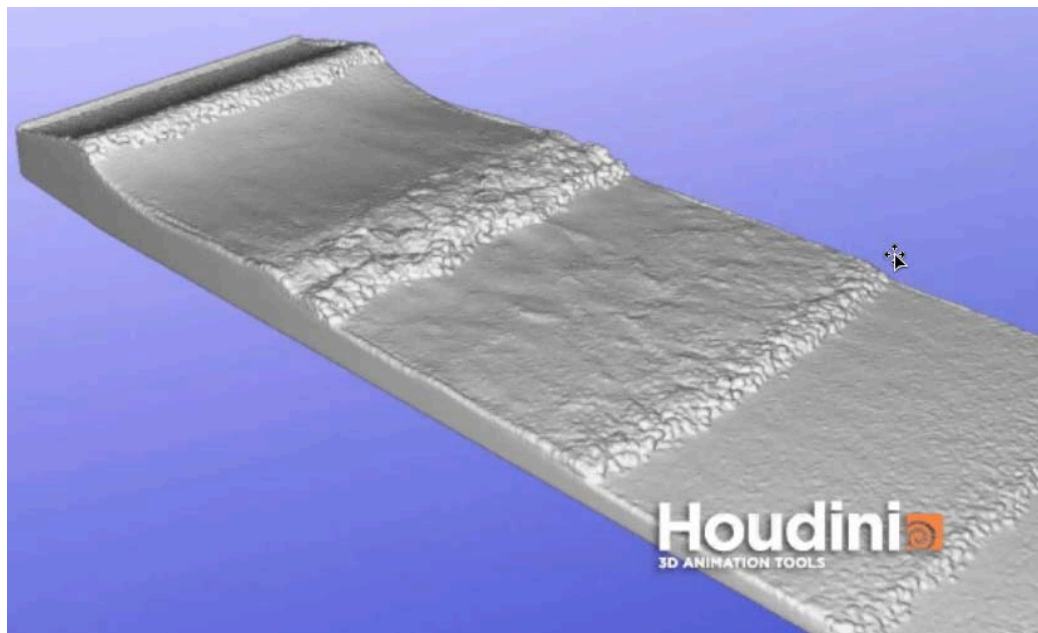


Figure 1 Breaking waves on a sloping beach. The wavemaker is at the left and the beach to the right. Note the turbulent wave bores traversing the surf zone. Visualization by Templeman Automation.

Figure 2 shows three rows of periodically-spaced short piles (10 in all) in the surf zone being overtopped by plunging breakers. Each cylinder computationally is a Cylinder object with no-flow boundary conditions on the sides and ends that has been duplicated and placed in a spatial array. The effects of the piling on run-up and wave height is found by comparing the solutions with and without the presence of the piling. The effects of piling density is easily determined by varying the spacing, size, and numbers of piling. Other shapes can be used in the surf zone, such as parallelepipeds (Cube objects), which can be used for tsunami run-up impacting on shoreline buildings or spheres (Sphere objects).

This test also revealed that GPU-SPHysics also models the mean set-up of water at the beach. This was observed by watching the mean water line move inland with the arrival of the first waves and then staying there throughout the remainder of the test. As wave set-up is a wave-averaged phenomenon, forced by the change in wave momentum flux due to breaking, then it follows that the SPH method can also predict other wave-induced phenomena such as mean longshore, cross-shore, and rip currents. This will be explored after the next example.

Levee Failure and Tsunami Attack on Coastal Community

Dalrymple and Herculat (2009) used GPU-SPHysics to examine various types of levee failure (single rapidly falling panel within a long wall comprised of vertical panels or a single slowly falling panel), which showed that the speed of falling wall affects the impact of water on structures. For example, if the wall section fails immediately, the flow through the wall resembles a three-dimensional

dam break, with most of the flow jetting directly in front of the wall section with considerable momentum flux and hence exerting large forces on structures. However, if the wall fails slowly, then water flows out the widening gaps at both sides of the falling section, directing the flow laterally along the remaining wall. The force is less on structures; nonetheless, after a long period of time, the flooding is the same for both cases. Here we include more (and varied) structures and a sloping upland to examine tsunami impact on a community.

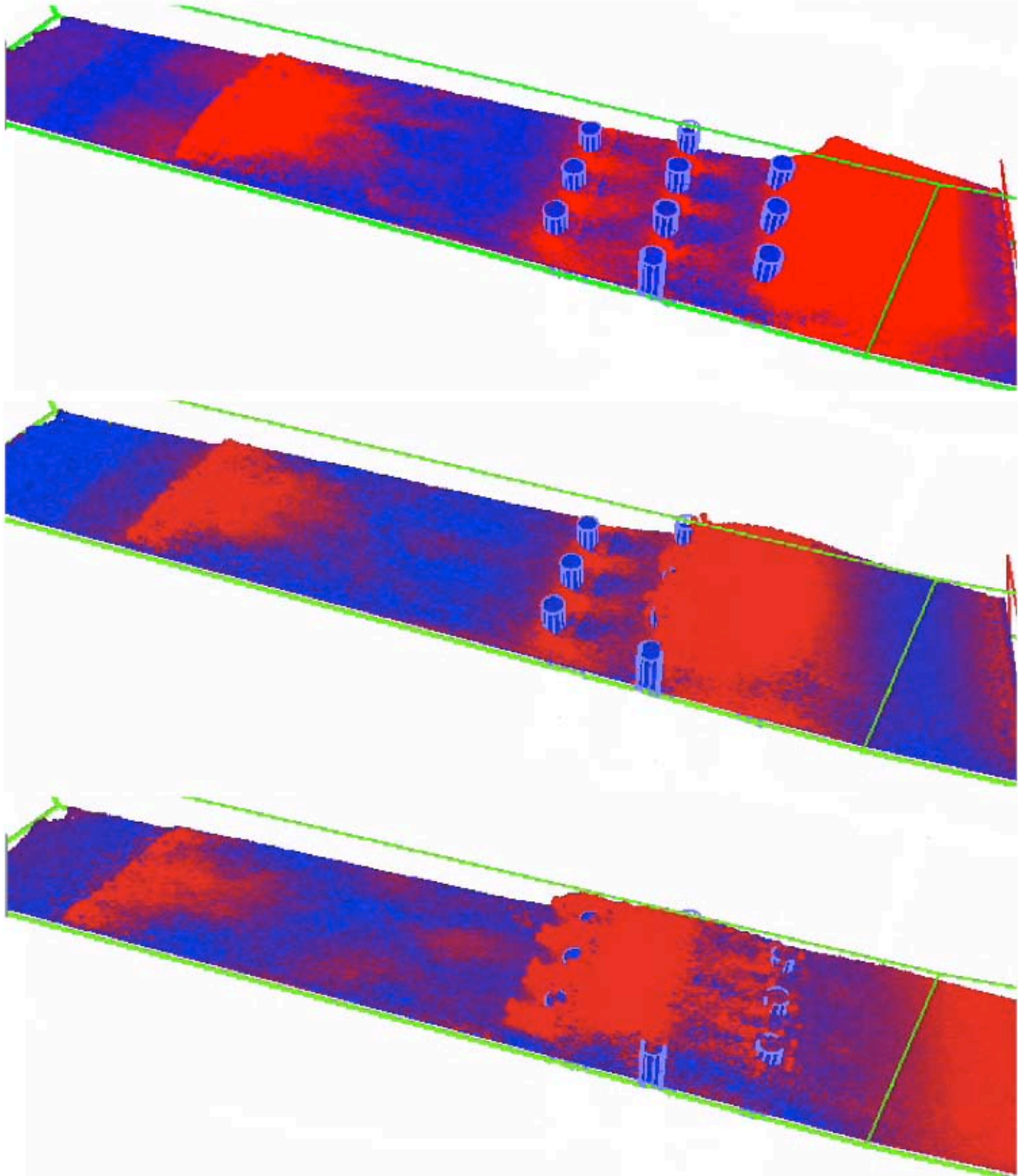


Figure 2 Plunging breakers impinging on a series of short piling in a wave tank modeled by SPH shown at three different times. The waves are created by a paddle wavemaker at the right; beach is to the left. Red denotes higher fluid velocities.

By reusing the cube object, the GPU-SPHysics code can represent a community. Each structure, consisting of a varying size cube with impermeable boundaries, requires 3 lines of code. By duplicating this code and varying the cube dimensions, the community is created. In Figure 3, the tsunami is represented by an instantaneous dam break. The wave front is shown impacting the first row of homes. The run-up on the front of the structures is evident and dramatic.

As the flow progresses, it runs down the shore-normal streets far faster than around the houses.

This simulation was carried out with 245,000 particles and the figure was created in 20 minutes on a MacBook Pro laptop (2009 vintage).

Nearshore Circulation

Using wider domains than a wave tank allows a variety of wave-induced flows to occur. These currents must be obtained by time-averaging the instantaneous wave-dominated flow fields with time to remove the wave motions. (As it is done when using the Boussinesq wave model to examine nearshore circulation, see, *e.g.*, Chen *et al.* (1999)). Here we compute an Eulerian average by using the neighbor search code to determine the fluid velocity on the fixed 3D Cartesian neighbor list grid. The velocity is found by using the SPH interpolation kernel (Wendland) to determine the SPH-interpolated velocity at the grid nodes. By averaging these gridded velocities over a wave period (about 15 samples in time based on our frequency of outputting data during the model run), the 3-D Eulerian wave-averaged velocity was obtained for the fixed grid.

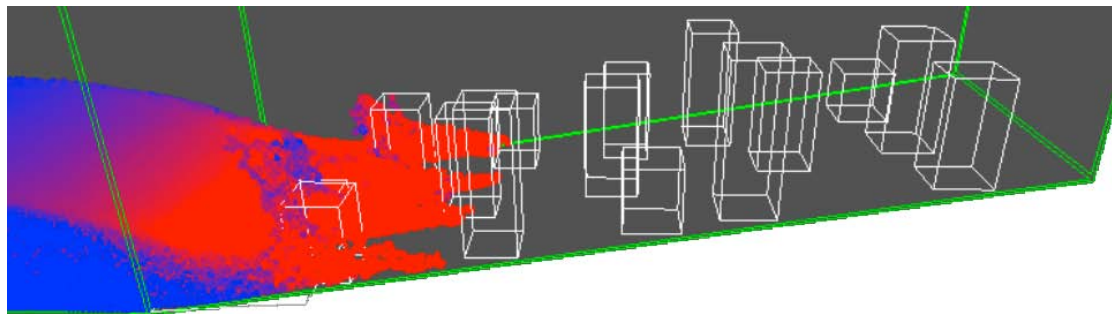


Figure 3 Tsunami attack (modeled by a dam break) on a coastal community on a sloping upland. Note the large run-up on the seaward buildings and the flow between the structures.

Drønen *et al.* (2002) examined the generation of a rip current in a wide wave tank (width = 4 m, length = 30 m) at the Technical University of Denmark. At the shoreline end of the tank, a planar 1:17 beach was installed that intersected with a flat bottomed trough (13 cm deep and extending 1.9 m in the offshore direction) and then another slope of 1:27 was added. The “sand bar” was 13 cm high and extends 3/4 the distance across the tank and 4 m in the offshore direction. Figures 4 and 5 show the tank layout and the numerical representation that was used in the GPU-SPHysics model. In their tests 1A-1C, the incident wave height was 0.19 m, the depth of water over the bar crest was 0.15 m, and the wave period was 1.5 seconds. In our simulations, we shortened the distance from the wavemaker to the bar to 5 m to increase resolution (SPH particle sizes were 0.025 m and there were 3.1 million particles used) and reduce run times.

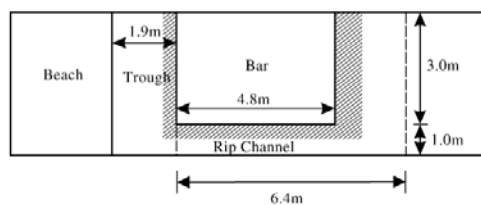


Figure 4 Planform view of the wave tank test of Drønen *et al.* (2002)

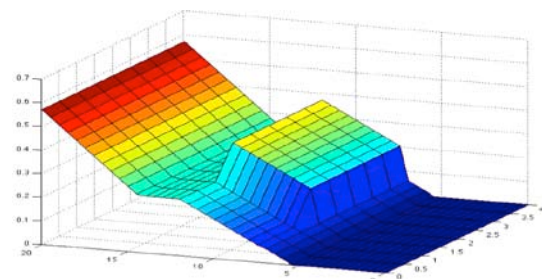


Figure 5 GPU-SPHysics representation of experiment

In Figure 6, two views of the water surface are shown after about 35 seconds of simulation. A water surface has been superimposed on the particles through image processing by Templeman Automation. Examining the lower view, the wavemaker is at the upper right, and the waves propagate towards the beach at the lower left. Over the bar (on the left side of the channel, as if Figure 5 was rotated counter-clockwise 90°), the waves break, while over the deeper channel, they do not. Further inshore on the bar, the waves show signs of the nonlinear phenomena of harmonic generation (Mei,1983, Mei and Ünlüata,1972). As the waves leave the bar region, they propagate over deeper water and the wave

height drops due to inverse shoaling. Nonetheless the reduction in wave momentum flux over the bar has created a local increase in mean water level over the trough. On the right side of the wave tank, over the rip channel, the incident waves interact with an outgoing rip current, created by the mean water level increases. From the results, it is clear that SPH replicates wave shoaling and breaking, harmonic generation, wave-current interaction, and diffraction.

Measurements made by Drønen et al. (2002) include mean water level and mean velocities to examine the wave-induced currents. The mean velocities were obtained from a laser Doppler anemometer located at about 1/3 the depth from the bottom at an array (0.8 m x 0.8 m) of locations. The currents were averaged over 50 wave periods. These data are shown in the bottom panel of Figure 7. GPU-SPHysics was used to create a number of output data files at regular time intervals (0.1 s). These files contain information on each particle: position and velocity, pressure and density, among other parameters. The 3-D neighbor list grid was superimposed upon the data. Then an SPH kernel was used to determine the average velocity at all the grid locations for a particular data set. By averaging over data sets (1.5 seconds worth), a wave-averaged 3D mesh of velocity vectors was obtained. Finally, averaging over the depth, provided the depth and wave-averaged velocity to compare with the lab test. (A quantitative comparison between numerical model and laboratory data can not be done as the original lab data no longer exists.)

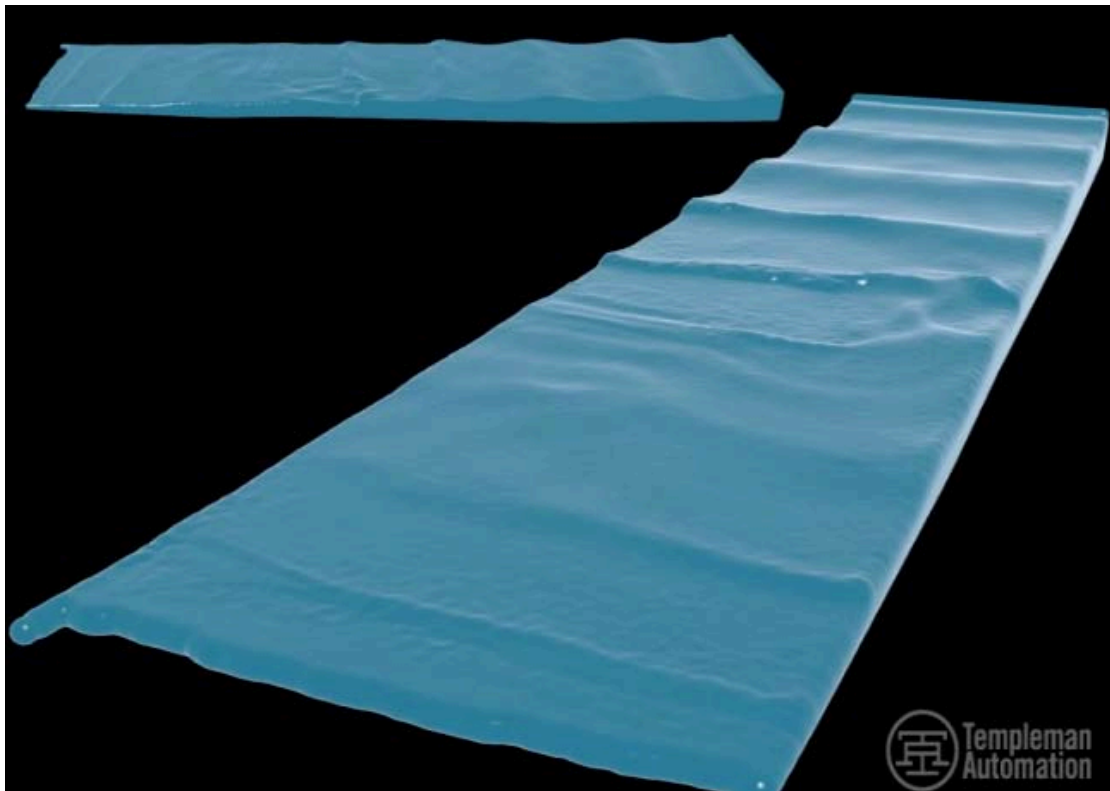


Figure 6 Two views of waves breaking over submerged bar, following Drønen et al. (2002). The rip channel is located on the right in the lower view. Note the breaking on the shoal at the left and the wave-current interaction occurring over the rip current. Visualization provided by Len White of Templeman Automation.

In Figure 8, the numerically computed vorticity is shown along with the vertically-averaged velocity vectors at two different times ($t=20$ s, 40 s) showing that the flow is slightly unsteady. For example, the large counter-clockwise gyre over the sand bar has migrated towards the rip channel in the second frame—this results in an increase in the rip current. Due to the wall boundary at $y=0$ in these figures, the rip current stays close to the wall as it flows offshore. This tends to reduce the jet instability of the rip current seen by Haller and Dalrymple (2001), in a wave basin with a rip channel with a bar on both sides. In that case the rip current migrated back and forth across the width of the rip channel continuously.

CONCLUSIONS

As shown by Monaghan (1994) and Dalrymple and Rogers (2006b), Smoothed Particle Hydrodynamics can model water waves quite well including shoaling and wave breaking. In this paper, we show that the method is also capable of modeling other wave phenomena, both linear and nonlinear, such as refraction, diffraction, harmonic generation in shallow water, and wave-current interaction. Further, the influence of the wave momentum is modeled, as wave set-up and wave-induced currents are engendered in the model. Smoothed Particle Hydrodynamics models are capable of modeling the nearshore circulation correctly and mean flows can be deduced by wave period-averaging the instantaneous SPH results. GPU-SPHysics provides a tremendous speed-up in computing wave problems. Problems that used to take a month on a serial computer now take a day on the Nvidia Tesla card, with its hundreds of massively parallel streaming processors. These graphics cards provide the ability to do low cost supercomputing for problems that are data sparse and computationally intensive like SPH. Further, since the number of processors on the graphics cards are doubling every 18 months or so, the future for GPU computing is extremely bright.

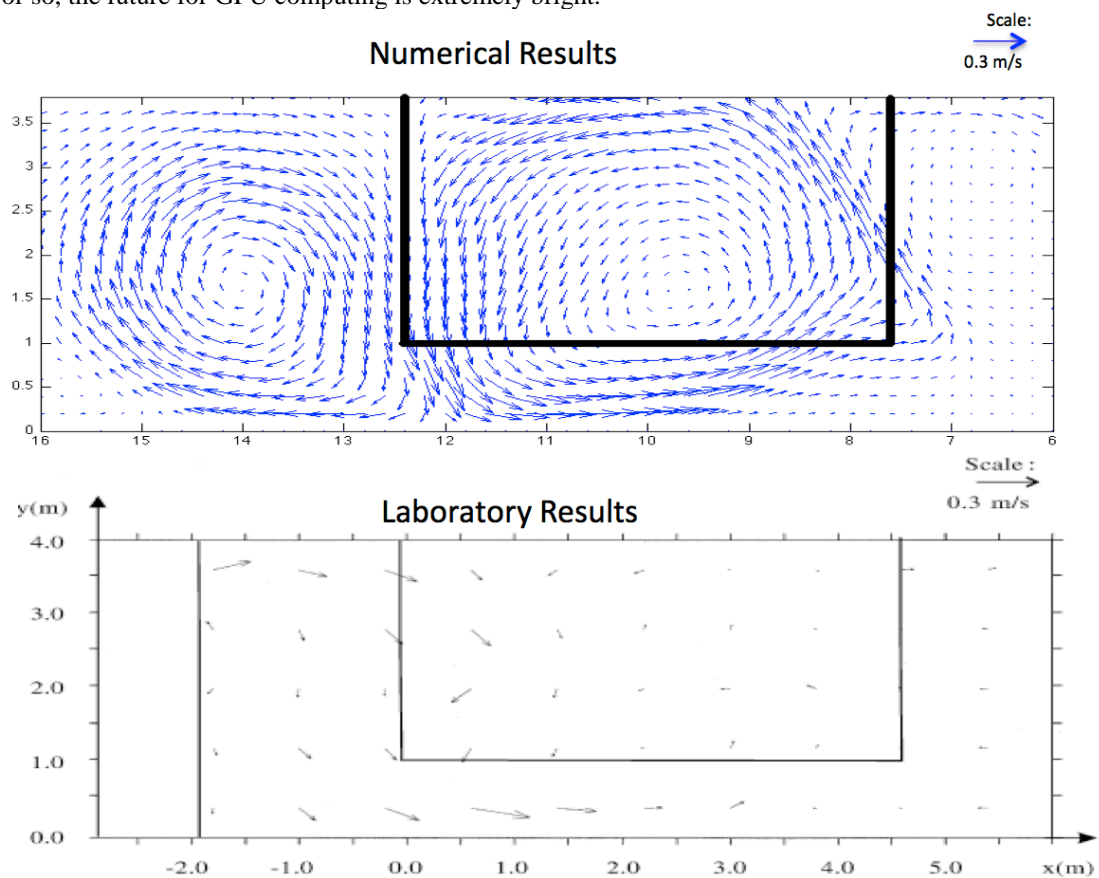


Figure 7 Depth and wave-averaged velocities from GPU-SPHysics and the laboratory measurements

ACKNOWLEDGMENTS

The first and fourth authors thank the Office of Naval Research Coastal Geoscience program for partial support, and A.H. and G.B. acknowledge support of the V3 LAVA project, sponsored by the Italian Dipartimento di Protezione Civile. Thanks to Len White of Templeman Automation for the visualization of two sets of results.

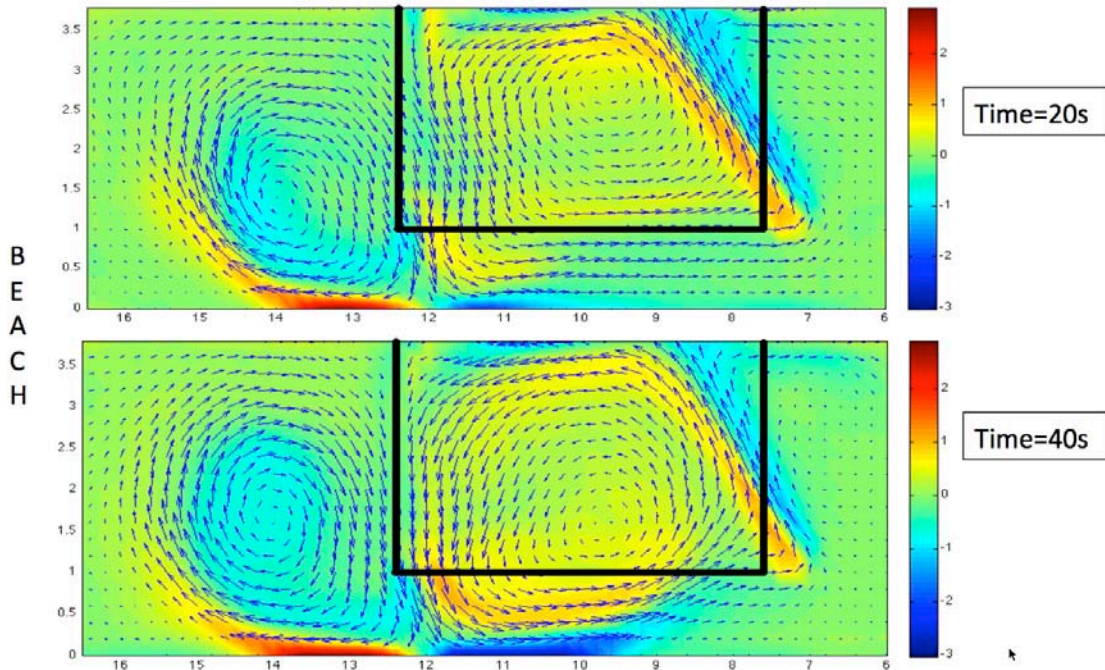


Figure 8 Mean vorticity at two different instances of time. Note how the main gyre over the bar has moved towards the rip channel.

REFERENCES

- Chen, Q., R. A. Dalrymple, J. T. Kirby, A. Kennedy, and M. C. Haller. 1999. Boussinesq modeling of a rip current system, *Journal of Geophysical Research*, 104(C9), 20,617–20,638.
- Colagrossi, A. and M. Landrini. 2003. Numerical simulation of interfacial flows by smoothed particle hydrodynamics, *Journal of Computational Physics*, 191(2), 448–475.
- Dalrymple, R. and A. H erault. 2009. Levee breaching with GPU-SPHysics code, *Proceedings of IV SPHERIC Workshop*, Nantes, (also <http://www.ce.jhu.edu/dalrymple/GPU/SPHERICIVDalrympleHerault.pdf>).
- Dalrymple, R. A., M. G omez-Gesteira, B. D. Rogers, A. Panizzo, S. Zou, A. Crespo, G. Cuomo, and M. Narayanaswamy. 2010. Smoothed Particle Hydrodynamics for Nonlinear Water Waves, in **Advances in Numerical Simulation of Nonlinear Water Waves**, 465–496. World Sci. Press.
- Dalrymple, R. A. and B. D. Rogers. 2006a. A note on wave celerities on a compressible fluid, *Proc. 30th International Conference on Coastal Engineering*. World Scientific Press.
- Dalrymple, R. A. and B. D. Rogers. 2006b. Numerical modeling of water waves with the SPH method, *Coastal Engineering*, 53(2/3), 141–147.
- Dr onen, N., H. Karunarathna, J. Freds e, B. M. Sumer, and R. Deigaard. 2002. An experimental study of rip channel flow, *Coastal Engineering*, 45(3), 223–238(16).
- Gingold, R. and J. Monaghan. 1977. Smoothed Particle Hydrodynamics: theory and application to non-spherical stars, *Monthly Notices of the Royal Astronomical Society*, 181, 375–389.
- G omez-Gesteira, M., B. D. Rogers, R. A. Dalrymple, and A. J. C. Crespo. 2010. State-of-the-art of classical SPH for free-surface flows, *Journal of Hydraulic Research*, 48 (Extra Issue), 6–27.
- G omez-Gesteira, R. and R. A. Dalrymple. 2004. Using a 3d SPH method for wave impact on a tall structure, *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 130(2), 63–69.
- Haller, M. C. and R. A. Dalrymple. 2001. Rip current instabilities, *Journal of Fluid Mechanics*, 433,

- 161–192.
- Héroult, A., G. Bilotta, and R. A. Dalrymple. 2010. SPH on GPU with CUDA, *Journal of Hydraulic Research*, 48(Extra Issue), 74–79.
- Héroult, A., A. Vicari, C. del Negro, and R. A. Dalrymple. 2009. Modeling Water Waves in the Surf Zone with GPU-SPHysics, *Proceeding of the Fourth SPHERIC Workshop*, Nantes.
- Lucy, L. 1977. A numerical approach to testing of the fusion process, *Astronomical Journal*, 88, 1013–1024.
- Mei, C. 1983. **The Applied Dynamics of Ocean Surface Waves**, John Wiley & Sons.
- Mei, C.C. and Ü. Ünlüata. 1972. Harmonic generation in shallow water, in **Waves on Beaches and Resulting Sediment Transport**, 181–202. Academic Press.
- Monaghan, J. 1994. Simulating free surface flows with SPH, *Journal of Computational Physics*, 110, 399–406.
- Monaghan, J. 2005. Smoothed particle hydrodynamics, *Reports on Progress in Physics*, 68, 1703–1759.
- O. R. Sørensen, H. A. Schaeffer, P. A. Madsen, and R. Deigaard. 1995. Wave breaking and induced nearshore circulations. *Proc. 24th Intl. Coastal Engrg. Conf.*, pages 2583-2594, 1995.
- Wendland, H. 2005. Computational aspects of radial basis function approximation, in K. Jetter *et al.* **Topics in Multivariate Approximation and Interpolation**. Elsevier.