

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Free-Surface Flow Simulations with Smoothed Particle Hydrodynamics Method using High-Performance Computing

Corrado Altomare, Giacomo Viccione,
Bonaventura Tagliafierro, Vittorio Bovolin,
José Manuel Domínguez and
Alejandro Jacobo Cabrera Crespo

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.71362>

Abstract

Today, the use of modern high-performance computing (HPC) systems, such as clusters equipped with graphics processing units (GPUs), allows solving problems with resolutions unthinkable only a decade ago. The demand for high computational power is certainly an issue when simulating free-surface flows. However, taking the advantage of GPU's parallel computing techniques, simulations involving up to 10^9 particles can be achieved. In this framework, this chapter shows some numerical results of typical coastal engineering problems obtained by means of the GPU-based computing servers maintained at the Environmental Physics Laboratory (EPhysLab) from Vigo University in Ourense (Spain) and the Tier-1 Galileo cluster of the Italian computing centre CINECA. The DualSPHysics free package based on smoothed particle hydrodynamics (SPH) technique was used for the purpose. SPH is a meshless particle method based on Lagrangian formulation by which the fluid domain is discretized as a collection of computing fluid particles. Speedup and efficiency of calculations are studied in terms of the initial interparticle distance and by coupling DualSPHysics with a NLSW wave propagation model. Water free-surface elevation, orbital velocities and wave forces are compared with results from experimental campaigns and theoretical solutions.

Keywords: SPH, HPC, free-surface flows, Navier-Stokes equations, Lagrangian techniques

1. Introduction

The non-stopping growing of computing power allowed increasing more and more spatial and temporal discretization when simulating engineering problems. The use of modern high-performance computing (HPC) systems, such as clusters equipped with graphics processing units (GPUs) or central processing units (CPUs) structured into a multi-node framework, let academics and professionals solve free-surface flow problems with resolutions unthinkable just a decade ago. Different spatial and temporal scales are often involved when simulating such kinds of phenomena, which may comprise wave generation, propagation, transformation and interaction with coastal or inland defences.

Among the others, smoothed particle hydrodynamics (SPH) method is a promising meshless technique for modelling fluid flows through the use of particles as it is capable to deal with large deformations, complex geometries and inlet wave shapes. Its original frame was developed in 1977 for astrophysical applications [1, 2]. Since then, it has been used in several research areas, e.g. coastal engineering [3–7], flooding forecast [8–11], solid body transport [12–15], soil mechanics [16–20], sediment erosion or entrainment processes [21–24], fast-moving non-Newtonian flows [25–33], flows in porous media [34–36], solute transport [37–39], turbulent flows [40–42] and multiphase flows [43–47], not to mention manifold industrial applications (see, for instance [48–51]). The main feature of SPH is that local quantities are evaluated by weighting information carried by neighbouring particles enclosed within a compact support, i.e. by performing short-range interactions among particles. Since the related neighbourhood definition takes most of the computing time, fast neighbour search algorithms have been developed so far [52–55, 64, 79].

Since a decade or so, SPH has been coded in the massive high-performance computing (HPC) context, making use of the Message Passing Interface (MPI) [56, 57] and the OpenMP library [58, 59], the standards for distributed and shared memory programming, respectively. Several applications involving multicore processors [60, 61] and graphics processing units (GPUs) [62–67] have been proposed so far. Joselli and co-workers [68] showed in 2015 that performing neighbour search on GPUs yields up to 100 times speedup against CPU implementations, therefore proving the benefits on exploiting the high floating-point arithmetic performance of GPUs for general purpose calculations. The same conclusion was drawn earlier in Ref. [69]. The first versions of SPH running on GPUs were presented in Ref. [70] and then in Ref. [69]. Non-Newtonian fluid flow simulations have been carried as well. Bilotta and co-workers, for instance, applied their GPUSPH model to lava flows [71]. In 2013, Wu and co-workers run GPUSPH to model dam-break flood through complex city layouts [72, 73]. Rustico et al. [74] measured the overall efficiency of the GPUSPH parallelization by applying the Karp-Flatt metric [75]. In Ref. [76], massive simulations of free-surface flow phenomena were carried on single and multi-GPU clusters. They used the sorting radix algorithm for inter-GPU particle swapping and subdomain ‘halo’ building to allow SPH particles of different subdomains interacting. In 2015, Cercos-Pita proposed the software AQUA_{gpusph} [77] based on the use of the freely available Open Computing Language (OpenCL) framework instead of using the Compute Unified Device Architecture (CUDA) platform. In Ref. [78], Gonnet proposed

scalable algorithms based on hierarchical cell decompositions and sorted interactions executed on hybrid shared/distributed memory parallel architectures. In Ref. [79], a general rigid body dynamics and an absolute nodal coordinate formulation (ANCF) were implemented to model rigid and flexible objects interacting with a moving fluid. In 2012, Cherfils and co-workers released JOSEPHINE [80], a parallel weakly compressible SPH code written in Fortran 90, intended for free-surface flows. Incompressible SPH (ISPH) algorithms, running on GPUs, have been developed as well [81–83].

This chapter shows some numerical SPH results of typical coastal engineering problems obtained by means of two different supercomputers: the GPU-based machine maintained at the EPhysLab from Vigo University in Ourense (Spain), mounting 14 NVIDIA Kepler-based cards, with a total of 39168 CUDA cores and the Tier-1 Galileo cluster, introduced on January 2015 by the Italian computing centre CINECA, a non-profit consortium, made up of 70 Italian universities, 6 Italian research institutions and the Italian Ministry of Education, University and Research (MIUR). Galileo is equipped with 516 nodes, each mounting 2 8-cores Intel Haswell 2.40 GHz for a total of 8256 cores, up-to-date Intel Phi 7120p (2 per node on 384 nodes) and NVIDIA Tesla K80 accelerators (2 per node on 40 nodes). Comparison with theoretical and experimental results is also included.

2. SPH fundamentals

Recent comprehensive reviews and related applications of the SPH method are given in [84–89]. Governing equations describing the motion of fluids are usually given as a set of partial differential equations (PDEs). These are discretized by replacing the derivative operators with equivalent integral operators (the so-called integral representation or kernel approximation) that are in turn approximated on the particle location (particle approximation). Next, Section 2.1 gives further details about these two steps, with reference to a generic field $f(\vec{x})$ depending on the location point $\vec{x} \in \mathcal{R}^d$, whereas Section 2.2 provides more specific details concerning the treatment of Navier-Stokes equations.

2.1. Approximation of a field $f(\vec{x})$ and its spatial gradients

Following the concept of integral representation, any generic continuous function $f(\vec{x})$ can be obtained using the Dirac delta functional δ , centred at the point \vec{x} (**Figure 1**) as

$$f(\vec{x}) = \int_{\Omega} f(\vec{y}) \delta(\vec{x} - \vec{y}) d\Omega_y \quad (1)$$

where Ω_y represents the domain of definition of f and $\vec{x}, \vec{y} \in \Omega$. Replacing δ with a smoothing function $W(\vec{x} - \vec{y}, h)$, Eq. (1) can be approximated as

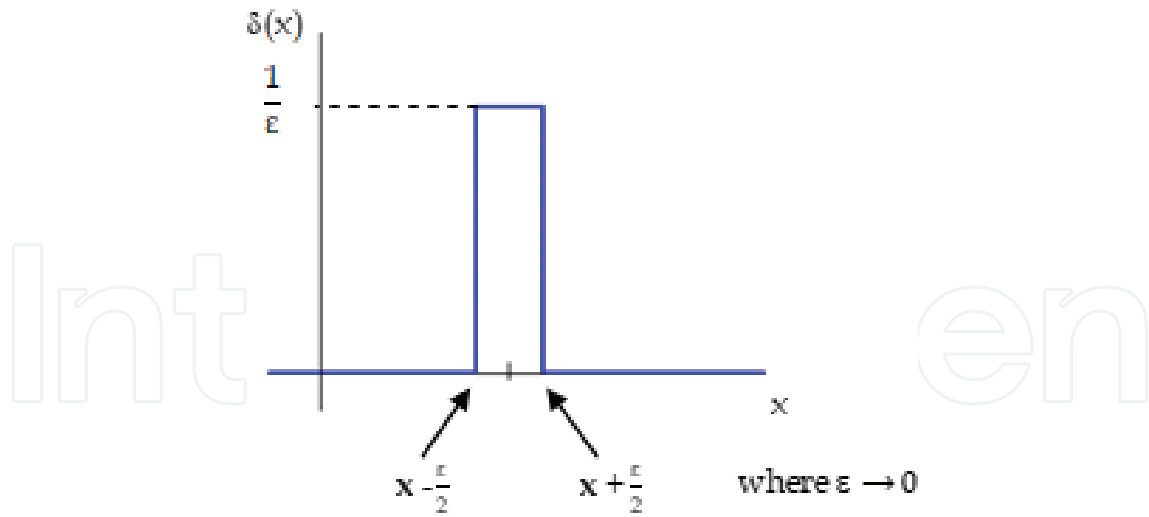


Figure 1. Dirac delta function centred at the point x .

$$f_I(\vec{x}) = \int_{\Omega} f(\vec{y}) W(\vec{x} - \vec{y}, h) d\Omega_y \quad (2)$$

W is the so-called smoothing kernel function or simply kernel and h , acting as spatial scale, is the smoothing length defining the influence area where W is not zero. While Eq. (1) yields an exact formulation for the function $f(\vec{x})$, Eq. (2) is an approximation. The definition of W is a key point in the SPH method since it establishes the accuracy of the approximating function $f(\vec{x})$ as well as the efficiency of the calculation. Note that the kernel approximation operator is marked by the index I .

The kernel function W has to satisfy some properties (see, for instance, [90, 91]). The following condition

$$\int_{\Omega} W(\vec{x} - \vec{y}, h) d\Omega_y = 1 \quad (3)$$

is known as partition of unity (or the zero-order consistency) as the integration of the smoothing function must yield the unity. Since W has to mimic the delta function, Eq. (3) can be rewritten as a limit condition in which the smoothing length tends to zero:

$$\lim_{h \rightarrow 0} W(\vec{x} - \vec{y}, h) \rightarrow \delta(\vec{x}). \quad (4)$$

Still, W has to be defined even, positive and radial symmetric on the compact support:

$$W(\vec{x} - \vec{y}, h) = W(\vec{y} - \vec{x}, h) = W(|\vec{x} - \vec{y}|, h) > 0 \quad |\vec{x} - \vec{y}| < \varphi \cdot h \quad (5a)$$

$$W(|\vec{x} - \vec{y}|, h) = 0 \quad \text{otherwise} \quad (5b)$$

where φ is a positive quantity defining the extent of the compact support. A large number of kernel functions are proposed in literature. Among the others, a computational-efficient and high accurate kernel is proposed by Wendland [92], defined as.

$$W(\vec{x} - \vec{y}, h) = A(n_d) \left(1 - \frac{q}{2}\right)^4 (2q + 1), 0 \leq q \leq 2, \text{ i.e. } \varphi = 2 \quad (6)$$

where $A(n_d)$, depending on the number of dimensions n_d , denotes a scaling factor that ensures the consistency of Eq. (3), whereas q denotes the dimensionless distance $|\vec{x} - \vec{y}|/h$.

The integral representation given by Eq. (2) can be converted into a discretized summation over all particle N within the compact support (**Figure 2**), yielding the particle approximation:

$$f_a(\vec{x}) = \sum_{k=1}^N \frac{m_k}{\rho_k} f(\vec{x}_k) W(\vec{x} - \vec{x}_k, h). \quad (7)$$

where the index k refers to particles within the compact support (see bold ones in **Figure 2**), with mass m_k and density ρ_k being carried. Note that in this case the particle approximation is marked by the 'a' pedix. The subscript will be avoided from now on. Eq. (7) can be rewritten with reference to particle 'i' as

$$f_i = f(\vec{x}_i) = \sum_{k=1}^N \frac{m_k}{\rho_k} f_k W_{ik}. \quad (8)$$

Particle approximation of spatial derivatives of a field function, such as divergence and gradient, is expressed using the gradient of the kernel function rather than the derivatives of the function itself:

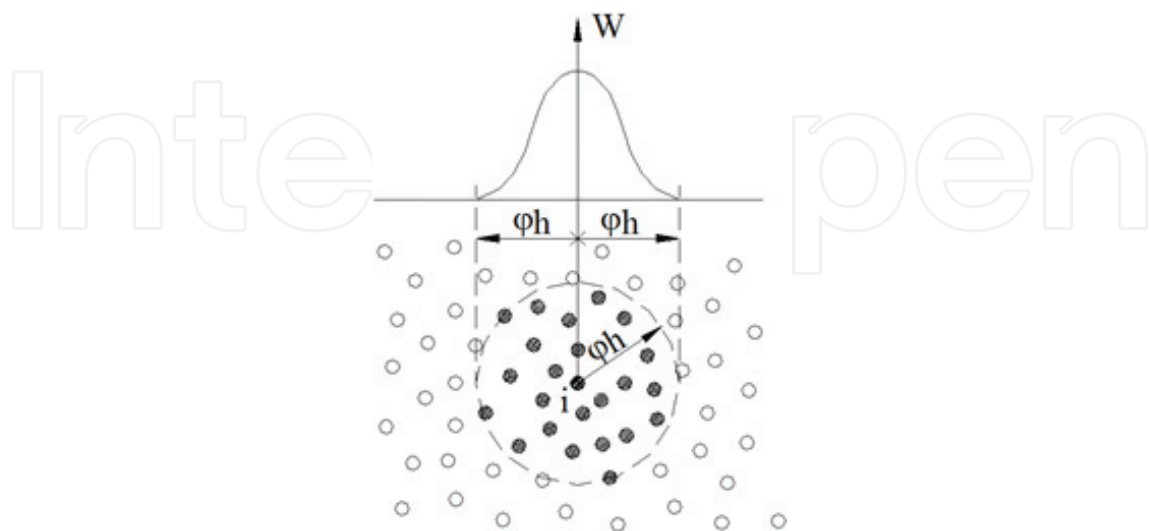


Figure 2. A kernel function defined at the particle 'i' and its support of radius φh . Local neighbourhood corresponds to the bold particles.

$$\vec{\nabla} \cdot \vec{f}_i = \sum_{k=1}^N \frac{m_k}{\rho_k} \vec{f}_k \cdot \vec{\nabla}_i W_{ik} \quad (9)$$

$$\vec{\nabla} f_i = \sum_{k=1}^N \frac{m_k}{\rho_k} f_k \vec{\nabla}_i W_{ik} \quad (10)$$

where the nabla operator $\vec{\nabla}$ is referred to the location of particle 'i'. The symbol ' \cdot ' denotes the dot product. Eqs. (9) and (10) offer the great advantage of estimating their left-hand side in terms of the kernel gradient, i.e. allowing no special hypotheses on the particular field function. A different formulation of the gradient field can be derived by introducing the following identity [87]

$$\vec{\nabla} \cdot \vec{f}(\vec{x}) = \frac{1}{\rho} \left\{ \vec{\nabla} \cdot [\rho \vec{f}(\vec{x})] - \vec{f}(\vec{x}) \cdot \vec{\nabla} \rho \right\} \quad (11)$$

inside the integral in Eq. (2), yielding in this case

$$\vec{\nabla} \cdot \vec{f}_i = \frac{1}{\rho_i} \sum_{k=1}^N m_k (\vec{f}_k - \vec{f}_i) \cdot \vec{\nabla}_i W_{ik} \quad (12)$$

Likewise, the divergence, another particle approximation of the gradient, can be derived, taking into account the following equation:

$$\vec{\nabla} f(\vec{x}) = \rho \left\{ \vec{\nabla} \frac{f(\vec{x})}{\rho} + \frac{f(\vec{x})}{\rho^2} \vec{\nabla} \rho \right\} \quad (13)$$

yielding

$$\vec{\nabla} f_i = \rho_i \sum_{k=1}^N m_j \left(\frac{f_k}{\rho_k^2} + \frac{f_i}{\rho_i^2} \right) \vec{\nabla}_i W_{ik} \quad (14)$$

Eqs. (12)–(14) are conveniently employed in fluid dynamics as they preserve the conservation of linear and angular momentum.

2.2. SPH form of governing equations

The mostly used governing laws ruling fluid motion are the Navier-Stokes equations, which specify that mass and linear momentum are preserved. Conservation laws in Lagrangian form are as follows:

$$\frac{d\rho}{dt} + \rho \vec{\nabla} \cdot \vec{v} = 0 \quad (15a)$$

$$\frac{d\vec{v}}{dt} = -\frac{\vec{\nabla} p}{\rho} + \nu \Delta \vec{v} + \vec{f} \quad (15b)$$

in which ρ and \vec{v} are, respectively, the density and velocity field, p is the isotropic pressure, ν is the laminar kinematic viscosity, $\Delta = \vec{\nabla} \cdot \vec{\nabla}$ is the Laplacian operator and \vec{f} the external force. Different approaches [93–95] are available to derive the density particle approximation of the continuity, Eq. (15a), and momentum, Eq. (15b). For instance, referring to Eq. (12), the density rate at particle ‘i’ can be approximated as follows:

$$\frac{d\rho_i}{dt} = - \sum_{k=1}^N m_k (\vec{v}_k - \vec{v}_i) \cdot \vec{\nabla}_i W_{ik}. \quad (16)$$

The material derivative of the velocity field can be deduced from Eq. (14) for the case of inviscid fluids, that is, $\nu = 0$:

$$\frac{d\vec{v}_i}{dt} = - \sum_{k=1}^N m_k \left(\frac{p_k}{\rho_k^2} + \frac{p_i}{\rho_i^2} \right) \vec{\nabla}_i W_{ik} + \vec{f}. \quad (17)$$

Numerical diffusion in terms of an artificial viscosity, e.g. proposed in Ref. [96], can be added in Eq. (17), allowing shock waves to be properly simulated:

$$\frac{d\vec{v}_i}{dt} = - \sum_{k=1}^N m_k \left(\frac{p_k}{\rho_k^2} + \frac{p_i}{\rho_i^2} + \Pi_{ik} \right) \vec{\nabla}_i W_{ik} + \vec{f} \quad (18)$$

The dissipative term Π_{ik} introduced above is the most general viscosity used in SPH computations, since it provides good results when modelling shock fronts. It is here defined as

$$\Pi_{ik} = \frac{-\alpha \bar{c}_{ik} \vartheta_{ik}}{\bar{\rho}_{ik}} \text{ when } \vec{v}_{ik} \cdot \vec{x}_{ik} < 0$$

$$\Pi_{ik} = 0 \quad \text{otherwise} \quad (19)$$

where

$$\vartheta_{ik} = \frac{-h \vec{v}_{ik} \cdot \vec{x}_{ik}}{\vec{x}_{ik}^2 + \eta^2} \quad (20)$$

The notation $\bar{a}_{ik} = (a_i + a_k)/2$, $b_{ik} = b_i - b_k$ is introduced above. The term c refers to the speed of sound which magnitude has conveniently to be at least 10 times greater than the maximum estimate of the scalar velocity field [94], $\eta = 0.1 h$ is employed to prevent numerical divergences when two particles are approaching and α is a coefficient that needs to be tuned in order to introduce the proper dissipation. A value of $\alpha = 0.01$ is suggested in Ref. [5] for wave propagation and wave-structure interaction studies.

Problem closure is achieved by combining conservation equations in discrete form (16) and (18) with an equation of state, when the weakly compressible scheme is adopted. A relationship between pressure and density is given in Ref. [97]:

$$p_i = \frac{c_0^2 \rho_0}{\gamma} \left[\left(\frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right] \quad (21)$$

where c_0 is the reference speed of sound, large enough to guarantee Mach numbers lower than 0.1–0.01, $\gamma = 7$, $\rho_0 = 1000 \text{ kg/m}^3$, when the liquid is water. c_0 is numerically computed like at least 10 times the expected maximum velocity.

3. The DualSPHysics code

DualSPHysics [98–100] is an open-source code developed by the University of Vigo (Spain) and the University of Manchester (UK) in collaboration with experts from all around the globe that can be freely downloaded from www.dual.sphysics.org. The code, written in two languages, namely, C++ and CUDA, is capable of using the parallel processing power of either CPUs or GPUs making the study of real engineering problems possible. Graphics processing units (GPUs) are massive floating-point stream processors adopted in computer game industry and image processing. Recently, they have been used in scientific computing thanks to the widespread of tools such as CUDA and OpenCL. Using CUDA as the programming framework for SPH leads to possible confusion with the word ‘kernel’. An SPH kernel is the weighting function used in the SPH interpolation process in particle approximation of ruling equations, e.g. Eqs. (16) and (18). A CUDA kernel, however, is defined as a CUDA function that is set up and executed N times in parallel by N different CUDA threads. Herein, to avoid confusion, we use the term function to describe the CUDA kernels. DualSPHysics makes full use of the function hierarchy present within the CUDA framework. A function executed and called by the CPU is declared as a host function, whereas a global function is called by the CPU but executed in parallel by the GPU. A device function, on the other hand, is only called and executed within the GPU by a global or another device function. This hierarchy is used for the computation of the interparticle forces. The simulations in DualSPHysics consist of three main steps: (i) creation of the particle neighbour list (NL), (ii) force computation (FC) for the particle interaction and (iii) the system update (SU) at the end of the time step.

Due to the Lagrangian nature of SPH, the particle interaction results to be the most time-consuming part of the whole algorithm. Each particle, as already stated, only interacts with its neighbour particles. Therefore, the construction of the neighbour list must be optimised. The cell-linked list described in Ref. [54] is implemented in DualSPHysics. This approach is preferred to the traditional approach, named Verlet list [101] that implies higher memory requirements than the cell-linked list. Besides, [54] proposed an innovative searching procedure based on a dynamic updating of the Verlet list and analyzed the efficiency of all the algorithms in terms of computational time and memory requirements.

DualSPHysics has proven its performance, reaching limits like being able to simulate more than 10^9 particles using 128 GPUs with an efficiency close to 100% [67].

3.1. Boundary conditions

Extensive research has been conducted over the last few years to develop accurate and efficient boundary conditions (BCs) in SPH method. Several approaches are proposed in the literature, such as boundary repulsive forces, fluid extensions to the solid boundary and boundary integral representing the term preservation. In DualSPHysics, boundaries (walls, bottom, coastal structures, wave generators, vessels, floating devices, etc.) are described using a discrete set of boundary particles that exert a repulsive force on the fluid particles when they approach. The so-called dynamic boundary condition [102] is used in DualSPHysics, where the boundary particles satisfy the same equations as the fluid particles; however, they do not move according to the forces exerted on them. Instead, they remain fixed (fixed boundary) or move according to some externally imposed movement (gates, flaps, etc.). Using this boundary condition, when a fluid particle approaches a boundary particle and the distance between them decreases beyond the kernel range, the density of the boundary particles increases giving rise to an increase of the pressure. This results in a repulsive force being exerted on the fluid particle due to the pressure term in the momentum equation. This dynamic boundary condition implemented in DualSPHysics does not include a specific value to define wall friction. However, this has been achieved in different validations by specifying a different viscosity value in the momentum equation when the fluid particles interact with the boundary ones.

3.2. Extra functionalities

3.2.1. Long-crested wave generation

The waves are generated in DualSPHysics by means of moving boundaries that aim to mimic the movement of a piston-type and flap-type wavemakers as in physical facilities. Only long-crested wave can be generated at this stage. The implementation of first-order and second-order wave generation theories is fully described in Ref. [3]. For monochromatic waves, this means to include super-harmonics. For random waves, subharmonic components are considered to suppress spurious long waves. Two standard wave spectra are implemented and used to generate random waves: JONSWAP and Pierson-Moskowitz spectra. The generation system allows having different random time series with the same significant wave height (H_{m0}) and the same peak period (T_p), just defining different phase seeds.

3.2.2. Wave reflection compensation

Wave reflection compensation is used in physical facilities to absorb the reflected waves at the wavemaker in order to avoid that they will be reflected back into the domain. In this way, the introduction into the system of extra spurious energy that will bias the results is prevented. The so-called active wave absorption system (AWAS) is implemented in DualSPHysics. The water surface elevation η at the wavemaker position is used and transformed by an appropriate time-domain filter to obtain a control signal that corrects the wave paddle displacement in order to absorb the reflected waves every time step. Hence, the target wavemaker position is corrected to avoid

reflection at the wavemaker. The position in real time of the wavemaker is obtained through the velocity correction of its motion. Further details on AWAS in DualSPHysics are reported in Ref. [3].

3.2.3. Hybridization with SWASH model

The hybridization technique between DualSPHysics model and SWASH model (<http://swash.sourceforge.net/>) is fully described in Ref. [5]. This technique aims to use each model for a specific purpose that best matches with its own capabilities, reducing the total computational cost and increasing the model accuracy. The advantages of using a hybridization technique between SWASH and DualSPHysics can be summarised as follows:

- Fast computations with large domains can be performed with SWASH, avoiding simulating large domains with DualSPHysics that requires huge computation times even using hardware acceleration.
- SWASH is suitable for calculation where statistical analysis is necessary such as computing wave height and good accuracy is obtained for wave propagation.
- SWASH is not suitable for calculation of wave impacts, while DualSPHysics can easily compute wave impacts, pressure load and exerted force onto coastal structures.
- Complex geometries cannot be represented with SWASH, and computation stability problems may appear when applied to rapidly changing bathymetry. Using DualSPHysics, any complex geometry or varying bathymetry can be simulated.

The hybridization between DualSPHysics and SWASH has been obtained through a one-way hybridization at this stage. The basic idea is to run SWASH for the biggest part of the physical domain to impose some boundary conditions on a fictitious wall placed between both media. This fictitious wall acts as a nonconventional wave generator in DualSPHysics: each boundary particle that forms the wall (hereafter called moving boundary or MB) will experience a different movement to mimic the effect of the incoming waves. SWASH provides values of velocity in different levels of depth. These values are used to move the MB particles. The displacement of each particle can be calculated using a lineal interpolation of velocity in the vertical position of the particle. Therefore, the MB is a set of boundary particles whose displacement is imposed by the wave propagated by SWASH and only exists for DualSPHysics. A multilayer approach can be used in SWASH. The SWASH velocity measured at each layer is therefore interpolated and converted into displacement time series for DualSPHysics.

4. Hardware features

4.1. The EPhysLab cluster

The GPU cluster maintained at the Environmental Physics Laboratory (EPhysLab) of Vigo University comprises four computing servers, whose details are as follows:

- Supermicro 7047: $4 \times$ NVIDIA GeForce GTX Titan Black, $2880 \times 4 = 11,520$ CUDA cores, $2 \times$ Intel Xeon E5-2640 at 2 GHz (16 cores), RAM 64 GB, storage 16 TB, estimated performance 6800 GFLOPS

- Supermicro 7047: $4 \times$ NVIDIA GeForce GTX Titan 2688 $\times 4 = 10,752$ CUDA cores, $2 \times$ Intel Xeon E5-2640 at 2 GHz (16 cores), RAM 64 GB, storage 20 TB, estimated performance 6000 GFLOPS
- Supermicro 7046: $4 \times$ NVIDIA GeForce GTX Titan Black, 2880 $\times 4 = 11,520$ CUDA cores, $2 \times$ Intel Xeon E5620 at 2.4 GHz (8 cores), RAM 64 GB, storage 9 TB, estimated performance 6800 GFLOPS
- Supermicro 6016GT-TF-TM2: $1 \times$ NVIDIA Tesla K40 2880 CUDA cores + $1 \times$ NVIDIA Tesla K20 2496 CUDA cores, $2 \times$ Intel X5550 at 2.66 GHz (8 cores), RAM 64 GB, storage 1.7 TB, estimated performance 2855 GFLOPS

4.2. The Galileo supercomputer

Galileo is a Tier-1 supercomputer among the fastest available to Italian industrial and public researchers. Introduced in the Italian computing centre CINECA on January 2015, this IBM NeXtScale model is equipped with up-to-date Intel accelerators (Intel Phi 7120p), NVIDIA accelerators (NVIDIA Tesla K80), as well as a top-level programming environment and a number of application tools. It is characterised by:

- 516 computing nodes with Intel Haswell 2.40 GHz processors, 2×8 core each (8256 cores in total)
- 128 GB of RAM per computing node, 8 GB per core, 66 TB of total RAM
- Internal network: InfiniBand with $4 \times$ QDR switches (≈ 40 Gb/s)
- Two Intel accelerators Phi 7120p per node on 384 nodes (768 in total)
- Two NVIDIA accelerators K80 per node on 40 nodes (80 in total, 20 available for scientific research)
- Eight nodes devoted to login/visualisation
- Theoretical peak performance 1.2 PFlops
- ≈ 480 GFLOPS single-node LINPACK (only CPU) sustained performance
- Disc space: ≈ 2 PB of local scratch
- Operating system: Linux CentOS 7.0

On June 2017, Galileo was ranked in 281st position on the top 500 supercomputer list (<https://www.top500.org/lists/>).

5. Test cases

Aiming to prove the capability of DualSPHysics model to reproduce accurately waves and wave-structure interaction phenomena, three different test cases have been selected and are here reported, namely:

1. Wave generation and propagation of random wave train in 2D.
2. Wave run-up on a cubic block breakwater in 3D.
3. Coupling of DualSPHysics with SWASH model and application to wave forces on coastal structures in shallow water conditions (2D).

5.1. Test case N. 1

The first test case comprises the generation and absorption of random waves in DualSPHysics. The 150 s time series to be generated is calculated starting from a JONSWAP spectrum. The target wave conditions are $H_{m0} = 0.06$ m and $T_p = 1.3$ s. The water depth is 0.36 m. The wavelength is 2.09 m. Second-order wave generation has been used (i.e. bound long waves). The wave conditions correspond to a second-order Stokes wave. The geometrical layout of the case is depicted in **Figure 3**: an 8.4-m long wave tank is modelled. A damping zone (passive absorption) is defined at the end of the tank. The water surface elevation and orbital velocities are measured using a 5-wave gauge array where the central wave gauge is at 2 L from the generator. The numerical results are compared with theoretical solutions.

A sensitivity analysis on the initial interparticle distance, dp , has been carried out. Four different values of dp have been selected in a range of H/dp between 6 (coarsest resolution) and 20 (finest resolution). For each case, the number of fluid particles and the computational runtime are reported in **Table 1**.

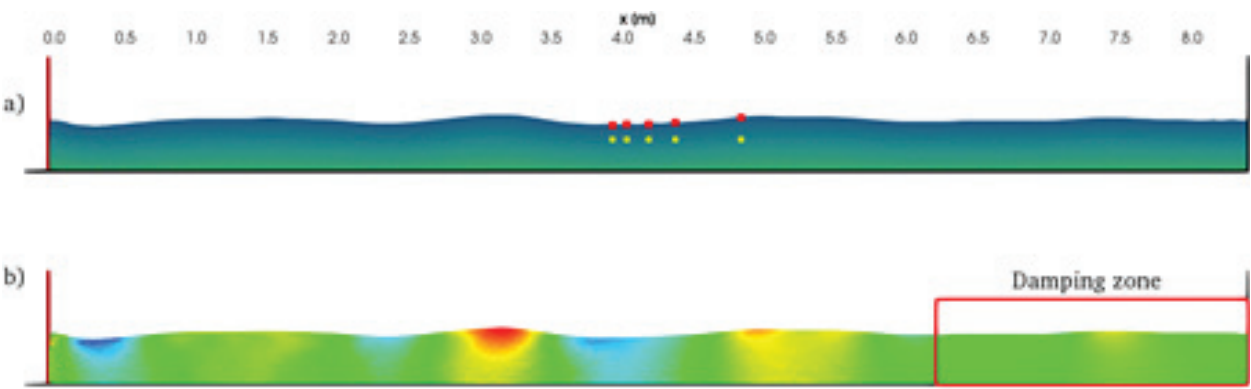


Figure 3. Layout of test case N.1: (a) position of the wave gauges (dots on the free surface) and velocity measurements (inner dots) and (b) horizontal velocity field and indication of the damping zone in the fluid domain.

H/dp	No. of fluid particles	Runtime [h]
6	29,365	0.9
10	82,541	2.9
12	119,209	4.5
20	333,081	16.6

Table 1. Runtimes and the number of fluid particles for each model resolution of test case N. 1.

The case with $H/dp = 10$ has been simulated also on Galileo supercomputer, specifically using one node in order to compare the computing capabilities between Galileo and one GPU from the EPhysLab cluster. The comparison is expressed in terms of the number of calculation steps per second of computational time. For each node in Galileo, 23.9 step/s can be simulated, whereas with a Tesla K20, 156.3 step/s are achieved. These results refer to 2D simulations, and they are expected to be different for 3D modelling.

The numerical results for $H/dp = 10$ have been plotted against the theoretical ones for each sensor position. They are all depicted in **Figures 4–6**. The model accuracy has been estimated in terms of spectral values of wave height and period. The numerical error, together with the calculated values for H_{m0} and $T_{m-1,0}$, is reported in **Table 2** for WG3 ($x = 4.18$ m). For $H/dp = 6$, the wave height is underestimated about 4%; meanwhile, starting from $H/dp = 10$, the errors for both wave height and period are in the order of 1–2%. Similar results are attained for the other four wave gauges. The orbital velocities show same degree of accuracy.

5.2. Test case N. 2

The second test case consists of a 3D case where the wave run-up on an armour breakwater has been simulated. Cubic blocks are displaced forming two layers with regular pattern on the seaward face of the breakwater, which has an angle of 28.3° with the horizontal. The side of each

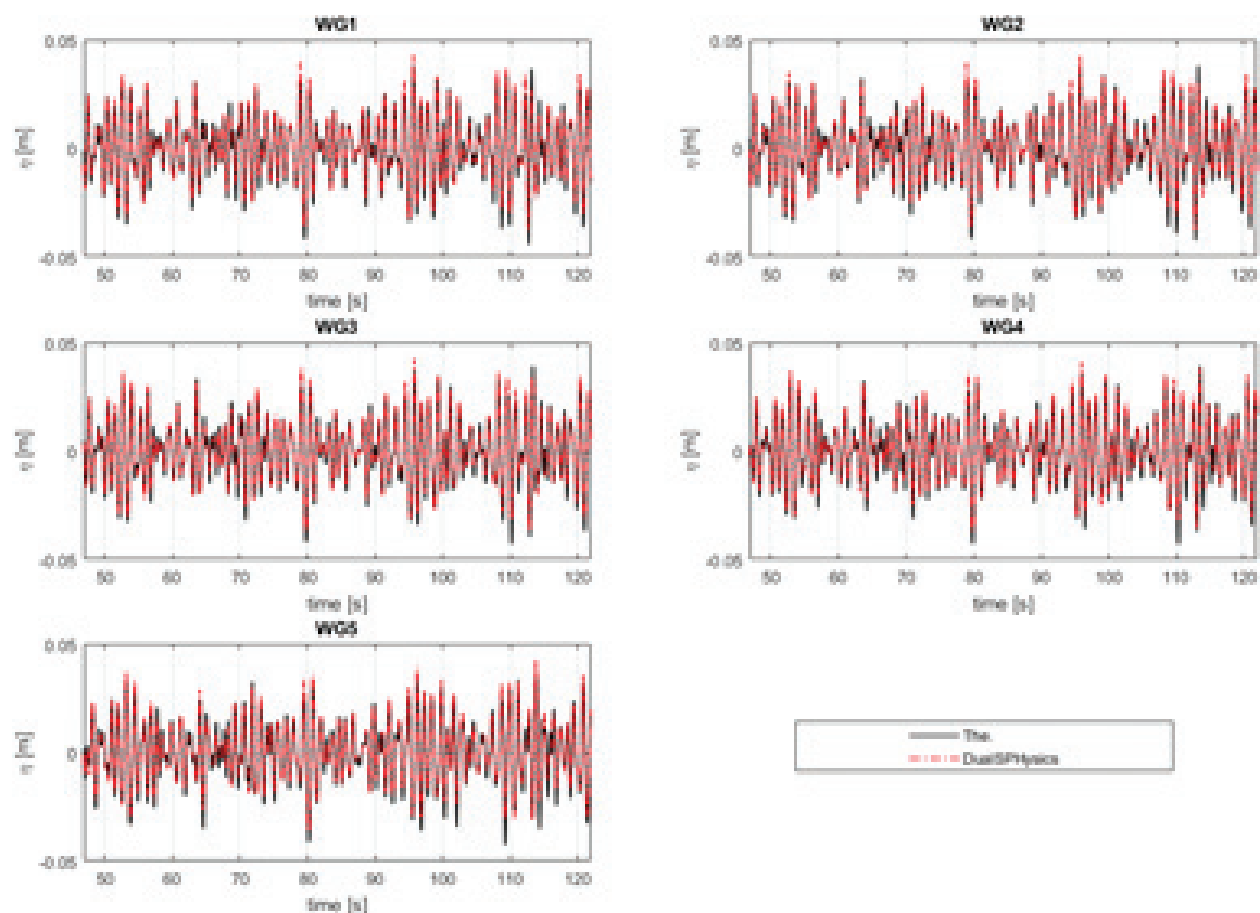


Figure 4. Comparison between the numerical and theoretical free-surface elevation at the 5-wave gauge positions.

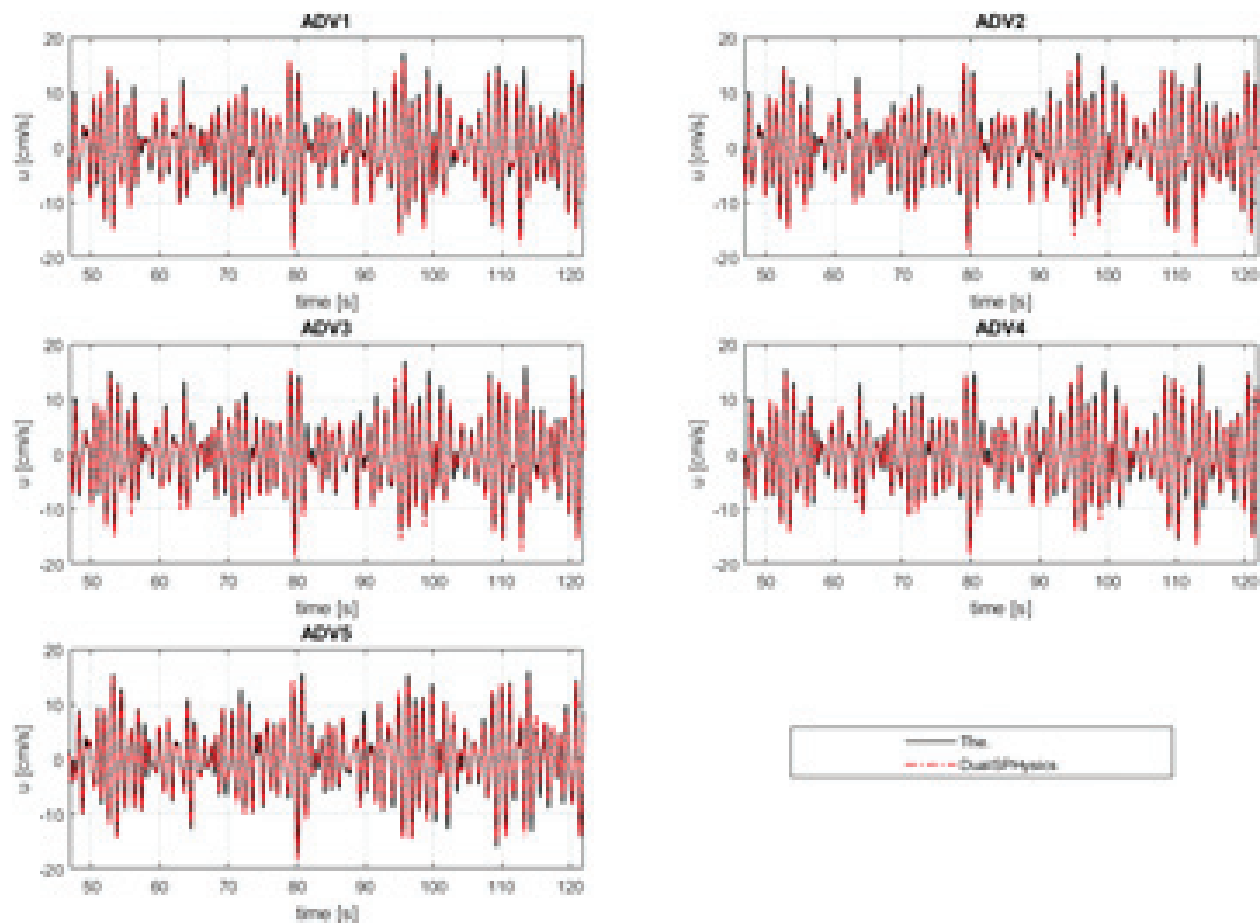


Figure 5. Comparison between the numerical and theoretical horizontal orbital velocity.

block measures 0.058 m. The case resembles an experimental one carried out in the small-scale wave flume CIEMito at the Technical University of Barcelona, Spain. The flume width is 0.38 m. Monochromatic waves have been simulated. The simulated wave height is 0.10 m, with mean period equal to 0.97 s. In total, 15 s of physical time has been simulated. The initial interparticle distance was 0.012 m, about one-eighth of the target wave height, resulting in 1,039,775 fluid particles. The simulation took 8.7 h using the Tesla K20 from the EPhysLab cluster.

Four wave gauges are located to measure the water surface elevation along the flume. The first wave gauge is at 3.10 m from the wavemaker; the last one is at 4.23 m. The distance between the toe of the breakwater and the wavemaker is 5.95 m. Moving boundaries mimicking a piston-type wavemaker are used in DualSPHysics to generate waves. To measure the run-up, the water surface elevation has been measured at 4160 locations across the breakwater. The results, post-processed in Matlab, have given the time series of wave run-up. A three-dimensional view of the numerical model is depicted in **Figure 7**. Using post-processing tools of DualSPHysics, an isosurface of the fluid has been extracted and plotted in ParaView software (www.paraview.org): this is coloured in blue in **Figure 7**. The two layers of cubic blocks are coloured in grey. The four yellow dots on the free surface indicate where the water surface elevation has been measured. The coloured area (from yellow to white) indicates all locations where the run-up has been measured.

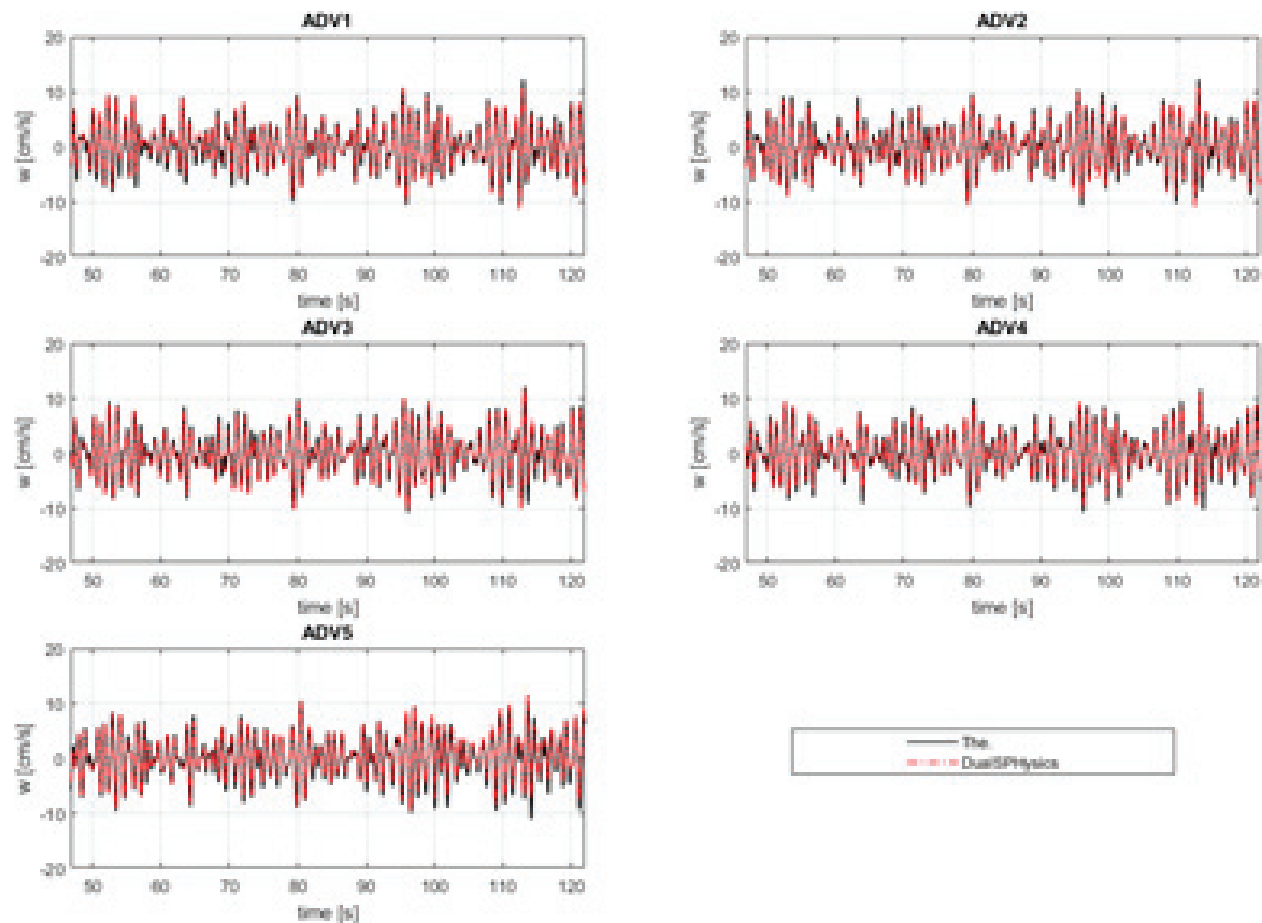


Figure 6. Comparison between the numerical and theoretical vertical orbital velocity.

H/dp	H_{m0-THE} [m]	$T_{m-1,0-THE}$ [s]	H_{m0-SPH} [m]	$T_{m-1,0-SPH}$ [s]	ϵ_H [%]	ϵ_T [%]
6	0.061	1.214	0.058	1.250	-4.52	+3.00
10			0.060	1.233	-1.00	+1.54
12			0.061	1.231	-0.12	+1.43
20			0.062	1.234	+1.70	+1.67

Table 2. Model accuracy at WG3 for different values of H/dp .

Figure 8 shows four different instants of time that make an entire run-up/run-down cycle over the breakwater. The colours indicate the fluid velocity field, i.e. horizontal orbital velocity. Red indicates high positive velocities (directed shorewards), whereas blue indicates negative velocities (directed seawards).

The water surface elevation measured in the numerical tank is depicted in Figure 9 for each wave gauge location. The wave run-up has been calculated for 26 cross sections along the width of the flume: the averaged time series is shown in Figure 10.

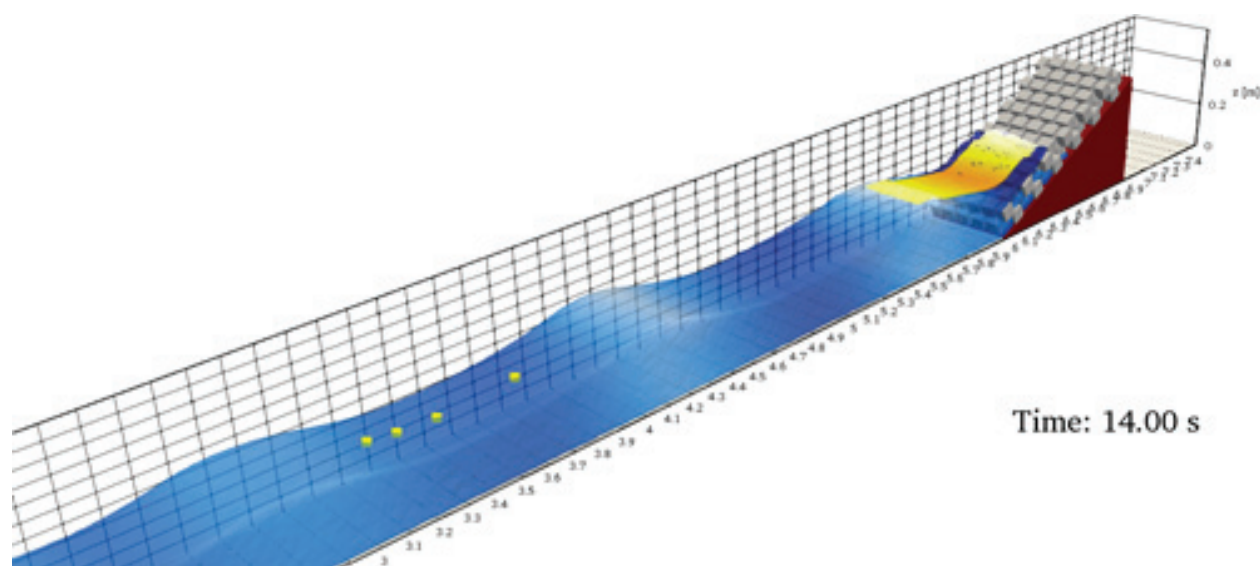


Figure 7. 3D view of the run-up simulation.

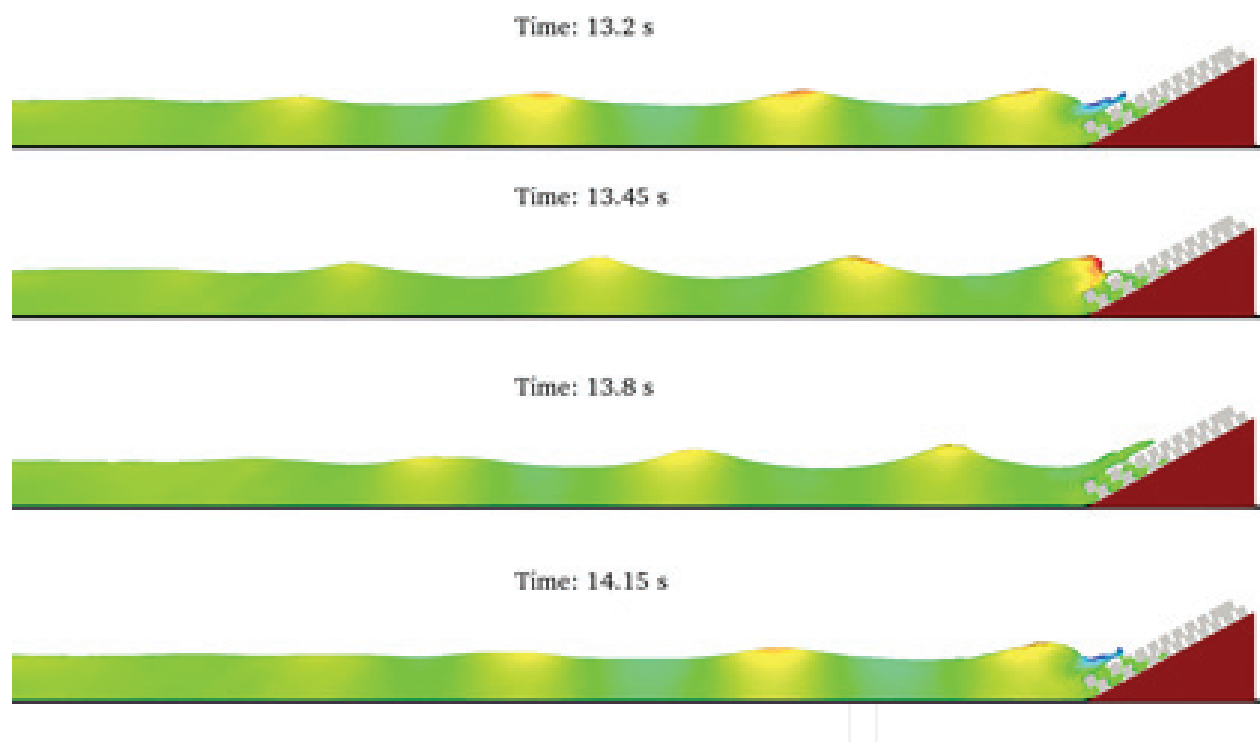


Figure 8. Snapshots of the wave run-up simulation during one run-up cycle.

5.3. Test case N. 3

The third test case comprises the validation of the hybridization technique between DualSPHysics model and SWASH model to study the impact of overtopping flows on multifunctional sea dikes with shallow foreshore. The main aim is to prove that overtopping flow characteristics and wave forces are modelled correctly and that the hybridization can represent a reliable solution that can be used as complementary or alternative to physical

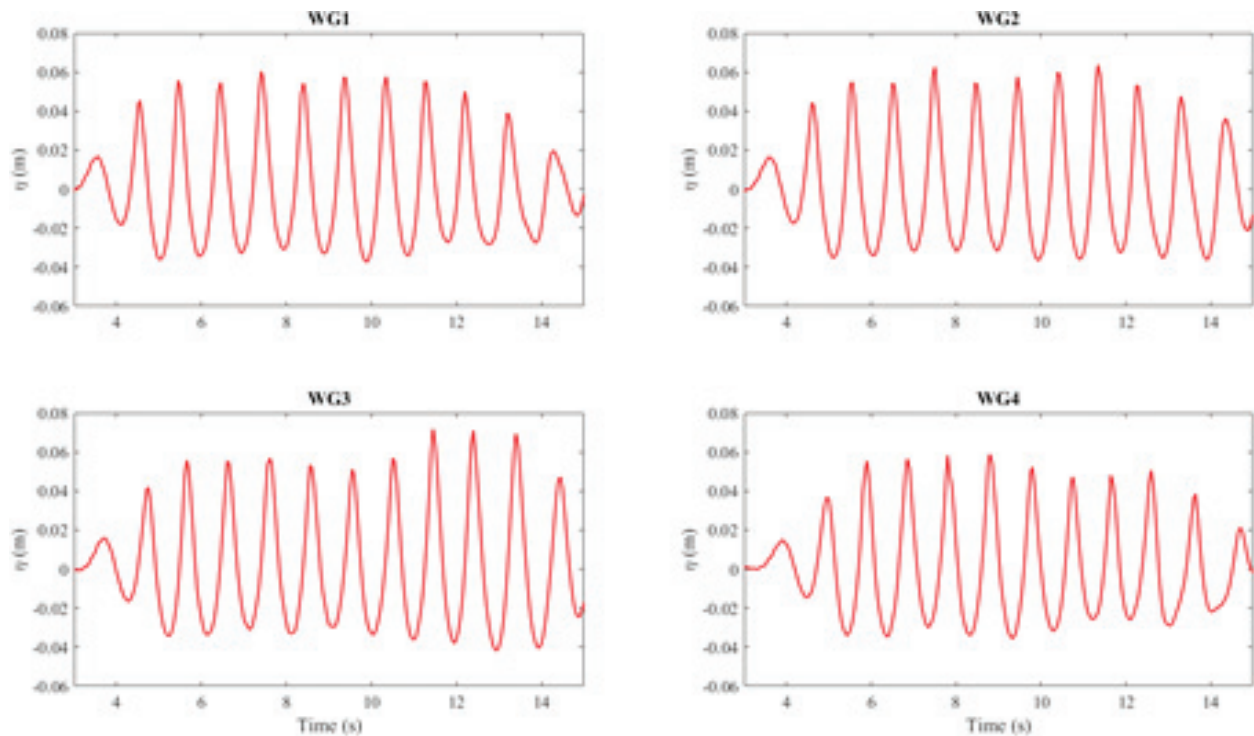


Figure 9. Water surface elevation along the numerical wave tank.

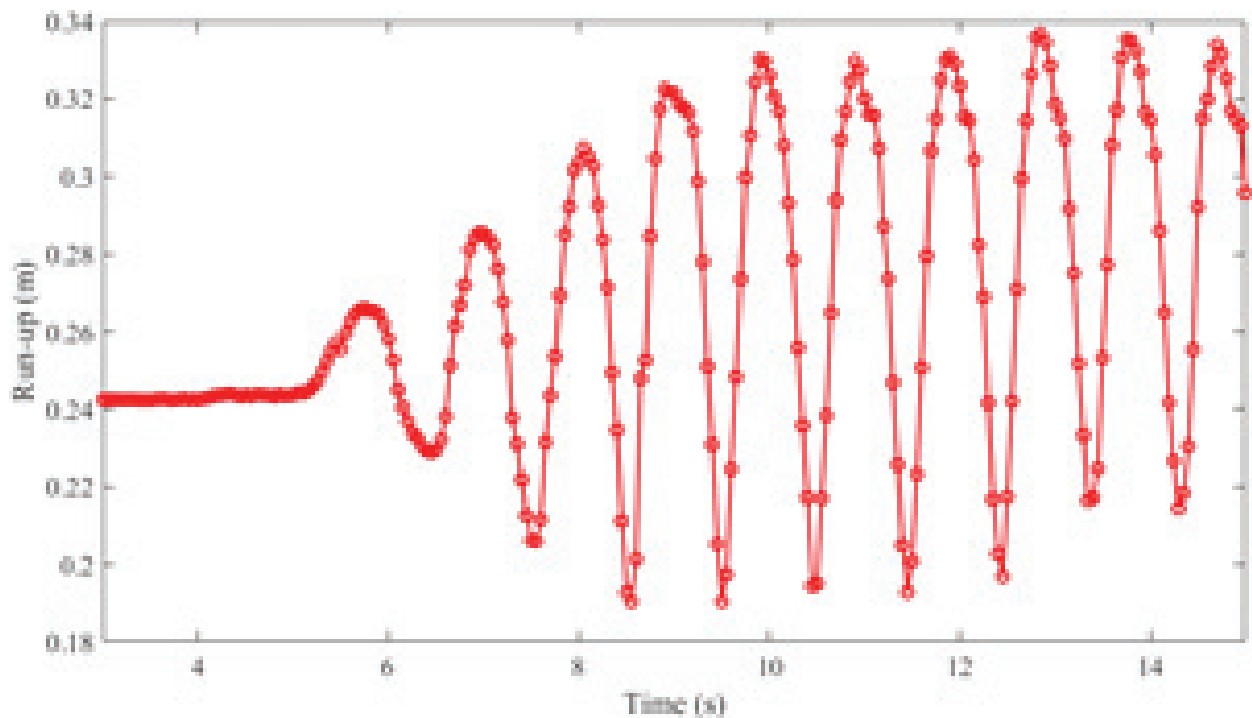


Figure 10. Time series of wave run-up: average along the width of the numerical tank.

modelling. The case of study is a typical case from the Belgian and Dutch coastline, where a building is constructed on the top of the dike. Physical model tests were carried out in a 4.0 m wide, 1.4 m deep and 70.0 m long wave flume at Flanders Hydraulics Research, Antwerp

(Belgium) to measure forces on the vertical wall (i.e. building), the layer thickness and velocities of the overtopping flows [103]. The geometrical layout is depicted in **Figure 11**: the foreshore slope was 1:35 and dike height 0.1 m. The dike slope was 1:3. Here, we refer to only regular wave cases.

SWASH has been previously validated against the physical model results: wave propagation, transformation and breaking have been accurately modelled, and the conditions at the toe of the dike are reproduced as in the physical model test. Then, SWASH has been implemented together with DualSPHysics to model the wave impact. Eight layers have been used in SWASH simulation. A hybridization point along the physical domain has been defined, and it is located at $x = 30.24$ m from the physical wavemaker in its neutral position (**Figure 11**), far enough from the location where the waves start to break (≈ 35.5 m). SWASH provides the boundary conditions for DualSPHysics at that location. DualSPHysics is used to model the part of domain between the coupling point and the dike. The quantities that have been measured and compared with the experimental results are (a) free-surface elevation after the coupling point, (b) overtopping flow thickness in three different locations along the dike crest and (c) wave forces on the vertical wall (measured in the physical model by means of two-load cells of model series Tedeo-Huntleigh 614). Both free-surface elevation and layer thickness were measured in the physical model by means of resistive wave gauges.

An initial interparticle distance, dp , of 0.003 m has been used leading to 494,388 fluid particles in DualSPHysics model. Fifty seconds of physical time has been simulated in the TITAN X graphic card, taking 10.8 h. A case with the whole physical domain modelled in DualSPHysics has been also modelled: in such case, the moving boundary is represented by the physical

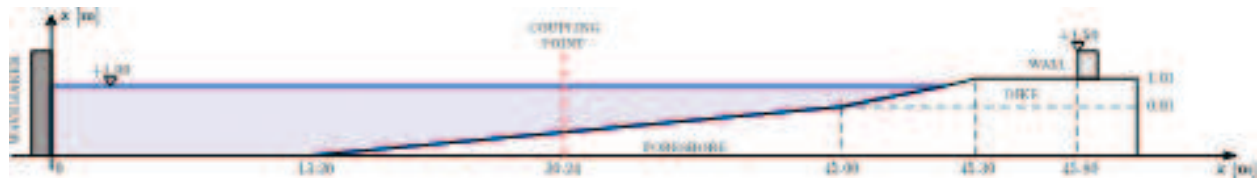


Figure 11. Layout of the flume at FHR and indication of the coupling point location for the SWASH-DualSPHysics model.

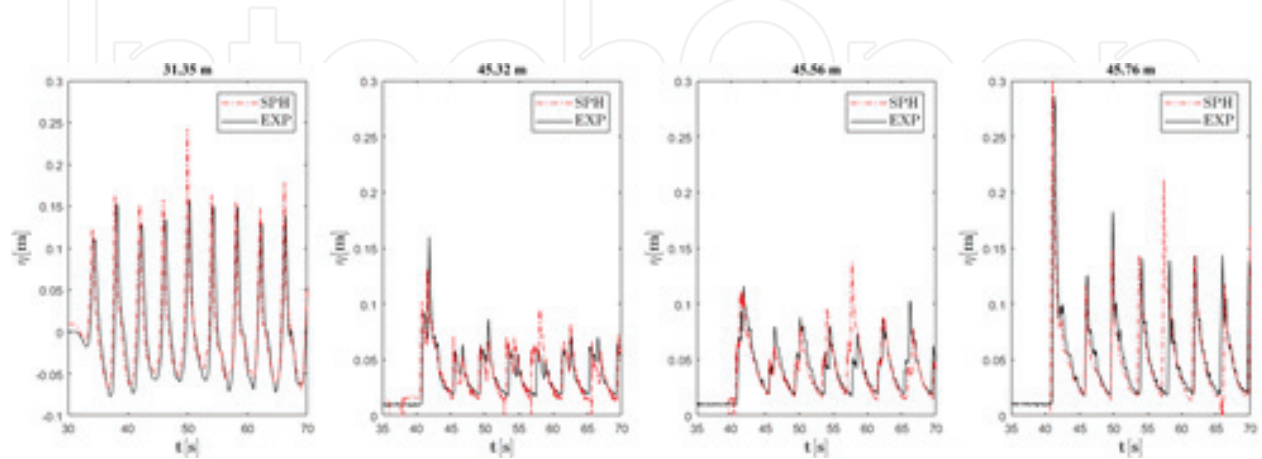


Figure 12. Results of free-surface elevation (left image) and overtopping layer thickness: numerical (red dash-dot line) vs. experimental (black solid line).

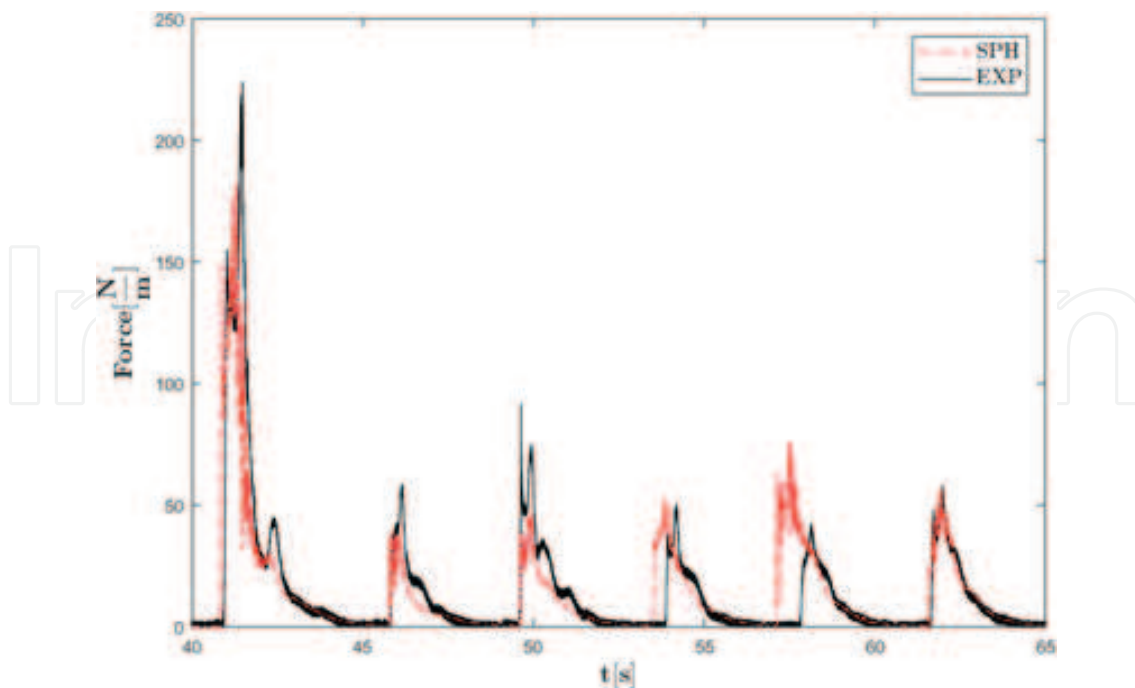


Figure 13. Results of overtopping wave forces on the wall: numerical (red dash-dot line) vs. experimental (black solid line).

wave generator, and its location is then at $x = 0.00$ m. This stand-alone DualSPHysics model took 95.6 h using the same TITAN X to simulate 3,389,266 fluid particles, about 10 times slower than the hybridised model.

The numerical and experimental free-surface elevation and layer thickness are plotted in **Figure 12**, showing that the numerical solution resembles the experimental accurately. The forces on the wall are represented in **Figure 13**. The differences between numerical and experimental results might be explained because of the highly turbulent and stochastic nature of the overtopping wave impact in this case, which makes the experimental test not repeatable (see [3] for further discussion on model inaccuracy for wave impacts).

6. Conclusions

The chapter offers a panoramic on the application of the SPH-based DualSPHysics code on supercomputers maintained at the EPhysLab from Vigo University in Ourense (Spain) and the Italian computing centre CINECA. Three test cases were selected in the general context of the coastal engineering: (1) wave generation and propagation of random wave train in 2D, (2) wave run-up on a cubic block breakwater in 3D and (3) coupling of DualSPHysics with SWASH model and application to wave forces on coastal structures in shallow water conditions (2D). Scalability is discussed by varying the spatial resolution, and efficiency is proved in the case of the hybridization. Comparison with theoretical free-surface elevation and orbital velocities for test case N. 1 and measured overtopping layer thickness and forces on vertical walls for test case N. 3 was satisfactory.

Acknowledgements

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support. Part of the computations was carried within the high-performance computing for Environmental Fluid Mechanics (HPCEFM17) project.

Author details

Corrado Altomare¹, Giacomo Viccione^{2*}, Bonaventura Tagliafierro², Vittorio Bovolin³, José Manuel Domínguez⁴ and Alejandro Jacobo Cabrera Crespo⁴

*Address all correspondence to: gviccion@unisa.it

1 Flanders Hydraulics Research, Antwerp, Belgium

2 Department of Civil Engineering, University of Salerno, Italy

3 University Centre for the Prediction and Prevention of Great Hazards (C.U.G.Ri.), Italy

4 Environmental Physics Laboratory (EPHysLab), Universidade de Vigo, Spain

References

- [1] Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*. 1977;**181**:375-389
- [2] Lucy LB. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*. 1977;**82**:1013-1024
- [3] Altomare C, Domínguez JM, Crespo AJC, González-Cao J, Suzuki T, Gómez-Gesteira M, et al. Long-crested wave generation and absorption for SPH-based DualSPHysics model. *Coastal Engineering*. 2017;**127**:37-54. DOI: 10.1016/j.coastaleng.2017.06.004
- [4] Crespo AJC, Altomare C, Domínguez JM, González-Cao J, Gómez-Gesteira M. Towards simulating floating offshore oscillating water column converters with smoothed particle hydrodynamics. *Coastal Engineering*. 2017;**126**:11-26. DOI: 10.1016/j.coastaleng.2017.05.001
- [5] Altomare C, Crespo AJC, Domínguez JM, Gómez-Gesteira M, Suzuki T, Verwaest T. Applicability of smoothed particle hydrodynamics for estimation of sea wave impact on coastal structures. *Coastal Engineering*. 2015;**96**:1-12. DOI: 10.1016/j.coastaleng.2014.11.001
- [6] Meringolo DD, Aristodemo F, Veltri P. SPH numerical modeling of wave-perforated break-water interaction. *Coastal Engineering*. 2015;**101**:48-68. DOI: 10.1016/j.coastaleng.2015.04.004

- [7] Barreiro A, Crespo AJC, Domínguez JM, Gómez-Gesteira M. Smoothed particle hydrodynamics for coastal engineering problems. *Computers and Structures*. 2013;**120**:96-106. DOI: 10.1016/j.compstruc.2013.02.010
- [8] Prakash M, Rothauge K, Cleary PW. Modelling the impact of dam failure scenarios on flood inundation using SPH. *Applied Mathematical Modelling*. 2014;**38**(23):5515-5534. DOI: 10.1016/j.apm.2014.03.011
- [9] Vacondio R, Rogers BD, Stansby PK, Mignosa P. Shallow water SPH for flooding with dynamic particle coalescing and splitting. *Advances in Water Resources*. 2013;**58**:10-23. DOI: 10.1016/j.advwatres.2013.04.007
- [10] Vacondio R, Mignosa P, Pagani S. 3D SPH numerical simulation of the wave generated by the Vajont rockslide. *Advances in Water Resources*. 2013;**59**:146-156. DOI: 10.1016/j.advwatres.2013.06.009
- [11] Kao H-M, Chang T-J. Numerical modeling of dambreak-induced flood and inundation using smoothed particle hydrodynamics. *Journal of Hydrology*. 2012;**448-449**:232-244
- [12] Canelas RB, Domínguez JM, Crespo AJC, Gómez-Gesteira M, Ferreira RML. Resolved simulation of a granular-fluid flow with a coupled SPH-DCDEM model. *Journal of Hydraulic Engineering*. 2017;**143**(9):06017012. DOI: 10.1061/(ASCE)HY.1943-7900.0001331
- [13] Canelas RB, Crespo AJC, Domínguez JM, Ferreira RML, Gómez-Gesteira M. SPH-DCDEM model for arbitrary geometries in free surface solid-fluid flows. *Computer Physics Communications*. 2016;**202**:131-140. DOI: 10.1016/j.cpc.2016.01.006
- [14] Canelas RB, Domínguez JM, Crespo AJC, Gómez-Gesteira M, Ferreira RML. A smooth particle hydrodynamics discretization for the modelling of free surface flows and rigid body dynamics. *International Journal for Numerical Methods in Fluids*. 2015;**78**:581-593. DOI: 10.1002/fld.4031
- [15] Amicarelli A, Albano R, Mirauda D, Agate G, Sole A, Guandalini R. A smoothed particle hydrodynamics model for 3D solid body transport in free surface flows. *Computers and Fluids*. 2015;**116**:205-228. DOI: 10.1016/j.compfluid.2015.04.018
- [16] Goodin C, Priddy JD. Comparison of SPH simulations and cone index tests for cohesive soils. *Journal of Terramechanics*. 2016;**66**:49-57. DOI: 10.1016/j.jterra.2015.09.002
- [17] Niroumand H, Mehrizi MEM, Saaly M. Application of mesh-free smoothed particle hydrodynamics (SPH) for study of soil behaviour. *Geomechanics and Engineering*. 2016;**11**(1):1-39. DOI: 10.12989/gae.2016.11.1.001
- [18] Nonoyama H, Moriguchi S, Sawada K, Yashima A. Slope stability analysis using smoothed particle hydrodynamics (SPH) method. *Soils and Foundations*. 2015;**55**(2): 458-470. DOI: 10.1016/j.sandf.2015.02.019
- [19] Wu Q, An Y, Liu QQ. A smoothed particle hydrodynamics method for modelling soil-water interaction. *Procedia Engineering*. 2015;**126**:579-583. DOI: 10.1016/j.proeng.2015.11.298

- [20] Grabe J, Stefanova B. Numerical modeling of saturated soils, based on smoothed particle hydrodynamics (SPH): Part 1: Seepage analysis. *Geotechnik*. 2014;**37**(3):191-197. DOI: 10.1002/gete.201300024
- [21] Braun A, Wang X, Petrosino S, Cuomo S. SPH propagation back-analysis of Baishuihe landslide in south-western China. *Geoenvironmental. Disasters*. 2017;**4**:1-10. DOI: 10.1186/s40677-016-0067-4
- [22] Khanpour M, Zarrati AR, Kolahdoozan M, Shakibaeinia A, Amirshahi SM. Mesh-free SPH modeling of sediment scouring and flushing. *Computers & Fluids*. 2016;**129**:67-78
- [23] Ran Q, Tong J, Shao S, Fu X, Xu Y. Incompressible SPH scour model for movable bed dam break flows. *Advances in Water Resources*. 2015;**82**:39-50. DOI: 10.1016/j.compfluid.2016.02.005
- [24] Razavitoosi SL, Ayyoubzadeh SA, Valizadeh A. Two-phase SPH modelling of waves caused by dam break over a movable bed. *International Journal of Sediment Research*. 2014;**29**(3):344-356. DOI: 10.1016/S1001-6279(14)60049-4
- [25] Farhadi A, Emdad H, Rad EG. Incompressible SPH simulation of landslide impulse-generated water waves. *Natural Hazards*. 2016;**82**(3):1779-1802. DOI: 10.1007/s11069-016-2270-8
- [26] Calvo L, Haddad B, Pastor M, Palacios D. Runout and deposit morphology of Bingham fluid as a function of initial volume: Implication for debris flow modelling. *Natural Hazards*. 2015;**75**(1):489-513. DOI: 10.1007/s11069-014-1334-x
- [27] Deng L, Wang W. Smoothed particle hydrodynamics for coarse-grained modeling of rapid granular flow. *Particuology*. 2015;**21**:173-178. DOI: 10.1016/j.partic.2014.08.012
- [28] Xenakis AM, Lind SJ, Stansby PK, Rogers BD. An incompressible SPH scheme with improved pressure predictions for free-surface generalised Newtonian flows. *Journal of Non-Newtonian Fluid Mechanics*. 2015;**218**:1-15. DOI: 10.1016/j.jnnfm.2015.01.006
- [29] Cascini L, Cuomo S, Pastor M, Sorbino G, Piciullo L. SPH run-out modelling of channelised landslides of the flow type. *Geomorphology*. 2014;**214**:502-513. DOI: 10.1016/j.geomorph.2014.02.031
- [30] Cuomo S, Pastor M, Cascini L, Castorino GC. Interplay of rheology and entrainment in debris avalanches: A numerical study. *Canadian Geotechnical Journal*. 2014;**51**(11):1318-1330. DOI: 10.1139/cgj-2013-0387
- [31] Lemiale V, Karantgis L, Broadbrige P. Smoothed particle hydrodynamics applied to the modelling of landslides. *Applied Mechanics and Materials*. 2014;**553**:519-524. DOI: 10.4028/www.scientific.net/AMM.553.519
- [32] Xu X, Ouyang J, Yang B, Liu Z. SPH simulations of three-dimensional non-Newtonian free surface flows. *Computer Methods in Applied Mechanics and Engineering*. 2013;**256**: 101-116. DOI: 10.1016/j.cma.2012.12.017

- [33] Viccione G, Bovolin V. Simulating triggering and evolution of debris-flows with smoothed particle hydrodynamics (SPH). In: International Conference on Debris-Flow Hazards Mitigation: Mechanics, Prediction, and Assessment, Proceedings. 2011. pp. 523-532
- [34] Peng C, Xu G, Wu W, Yu H-S, Wang C. Multiphase SPH modeling of free surface flow in porous media with variable porosity. *Computers and Geotechnics*. 2017;**81**:239-248. DOI: 10.1016/j.compgeo.2016.08.022
- [35] Ren B, Wen H, Dong P, Wang Y. Numerical simulation of wave interaction with porous structures using an improved smoothed particle hydrodynamic method. *Coastal Engineering*. 2014;**88**:88-100. DOI: 10.1016/j.coastaleng.2014.02.006
- [36] Aly AM, Asai M. Three-dimensional incompressible smoothed particle hydrodynamics for simulating fluid flows through porous structures. *Transport in Porous Media*. 2015;**110**(3):483-502. DOI: 10.1007/s11242-015-0568-8
- [37] Mayoral-Villa E, Alvarado-Rodríguez CE, Klapp J, Gómez-Gesteira M, Di G, Sigalotti L. Smoothed particle hydrodynamics: Applications to migration of radionuclides in confined aqueous systems. *Journal of Contaminant Hydrology*. 2016;**187**:65-78
- [38] Boso F, Bellin A, Dumbser M. Numerical simulations of solute transport in highly heterogeneous formations: A comparison of alternative numerical schemes. *Advances in Water Resources*. 2013;**52**:178-189
- [39] Herrera PA, Massabó M, Beckie RD. A meshless method to simulate solute transport in heterogeneous porous media. *Advances in Water Resources*. 2009;**32**(3):413-429
- [40] Hu XY, Adams NA. A SPH model for incompressible turbulence. *Procedia IUTAM*. 2015;**18**:66-75
- [41] Ren B, Wen H, Dong P, Wang Y, Improved SPH. Simulation of wave motions and turbulent flows through porous media. *Coastal Engineering*. 2016;**107**:14-27. DOI: 10.1016/j.jconhyd.2016.01.008
- [42] Violeau D, Issa R. Numerical modelling of complex turbulent free-surface flows with the SPH method: An overview. *International Journal for Numerical Methods in Fluids*. 2007;**53**:277-304. DOI: 10.1002/fld.1292
- [43] Mokos A, Rogers BD, Stansby PK. A multi-phase particle shifting algorithm for SPH simulations of violent hydrodynamics with a large number of particles. *Journal of Hydraulic Research*. 2017;**55**(2):143-162. DOI: 10.1080/00221686.2016.1212944
- [44] Gong K, Shao S, Liu H, Wang B, Tan SK. Two-phase SPH simulation of fluid-structure interactions. *Journal of Fluids and Structures*. 2016;**65**:155-179. DOI: 10.1016/j.jfluidstructs.2016.05.012
- [45] Zhou L, Cai ZW, Zong Z, Chen Z. An SPH pressure correction algorithm for multiphase flows with large density ratio. *International Journal of Computational Methods*. 2016;**81**(12):765-788. DOI: 10.1002/fld.4207

- [46] Chen Z, Zong Z, Liu MB, Zou L, Li HT, Shu C. An SPH model for multiphase flows with complex interfaces and large density differences. *Journal of Computational Physics*. 2015;**283**:169-188. DOI: 10.1016/j.jcp.2014.11.037
- [47] Aristodemo F, Federico I, Veltri P, Panizzo A. Two-phase SPH modelling of advective diffusion processes. *Environmental Fluid Mechanics*. 2010;**10**(4):451-470. DOI: 10.1007/s10652-010-9166-z
- [48] Cleary PW, Hilton JE, Sinnott MD. Modelling of industrial particle and multiphase flows. *Powder Technology*. 2017;**314**:232-252. DOI: 10.1016/j.powtec.2016.10.072
- [49] Shadloo MS, Oger G, Le Touzé D. Smoothed particle hydrodynamics method for fluid flows, towards industrial applications: Motivations, current state, and challenges. *Computers and Fluids*. 2016;**136**:11-34. DOI: 10.1016/j.compfluid.2016.05.029
- [50] Wieth L, Kelemen K, Braun S, Koch R, Bauer H-J, Schuchmann HP. Smoothed particle hydrodynamics (SPH) simulation of a high-pressure homogenization process. *Microfluidics and Nanofluidics*. 2016;**20**(2):1-18. DOI: 10.1007/s10404-016-1705-6
- [51] Harrison SM, Cleary PW, Eyres G, Sinnott M, Lundin L. Challenges in computational modelling of food breakdown and flavour release. *Food and Function*. 2014;**5**(11):2792-2805. DOI: 10.1039/C4FO00786G
- [52] Wang D, Zhou Y, Shao S. Efficient implementation of smoothed particle hydrodynamics (SPH) with plane sweep algorithm. *Communications in Computational Physics*. 2016;**19**:770-800. DOI: 10.4208/cicp.010415.110915a
- [53] Gan BS, Nguyen DK, Han A, Alisjahbana SW. Proposal for fast calculation of particle interactions in SPH simulations. *Computers and Fluids*. 2014;**104**:20-29. DOI: 10.1016/j.compfluid.2014.08.004
- [54] Domínguez JM, Crespo AJC, Gómez-Gesteira M, Marongiu JC. Neighbour lists in smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*. 2011;**67**:2026-2042. DOI: 10.1002/fld.2481
- [55] Viccione G, Bovolin V, Pugliese Carratelli E. Defining and optimizing algorithms for neighbouring particle identification in SPH fluid simulations. *International Journal for Numerical Methods in Fluids*. 2008;**58**:625-638. DOI: 10.1002/fld.1761
- [56] Yeylaghi S, Moa B, Oshkai P, Buckham B, Crawford C. ISPH modelling for hydrodynamic applications using a new MPI-based parallel approach. *Journal of Ocean Engineering and Marine Energy*. 2017;**3**:3-35. DOI: 10.1007/s40722-016-0070-6
- [57] Oger G, Le Touzé D, Guibert D, De Leffe M, Biddiscombe J, Soumagne J, et al. On distributed memory MPI-based parallelization of SPH codes in massive HPC context. *Computer Physics Communications*. 2016;**200**:1-14. DOI: 10.1016/j.cpc.2015.08.021
- [58] Nishiura D, Furuichi M, Sakaguchi H. Computational performance of a smoothed particle hydrodynamics simulation for shared-memory parallel computing. *Computer Physics Communications*. 2015;**194**:18-32. DOI: 10.1016/j.cpc.2015.04.006

- [59] Winkler D, Meister M, Rezavand M, Rauch W. gpuSPHASE—A shared memory caching implementation for 2D SPH using CUDA. *Computer Physics Communications*. 2017; **213**:165-180. DOI: 10.1016/j.cpc.2016.11.011
- [60] Domínguez JM, Barreiro A, Crespo AJC, García-Feal O, Gómez-Gesteira M. Parallel CPU/GPU computing for smoothed particle hydrodynamics models. In: Klapp J, Sigalotti L Di G, Medina A, Gerardo Ruiz-Chavarría AL, editors. *Recent Advances in Fluid Dynamics with Environmental Applications*. 2016. pp. 477–491. DOI: 10.1007/978-3-319-27965-7_34
- [61] Domínguez JM, Crespo AJC, Gómez-Gesteira M. Optimization strategies for CPU and GPU implementations of a smoothed particle hydrodynamics method. *Computer Physics Communications*. 2013; **184**(3):617-627. DOI: 10.1016/j.cpc.2012.10.015
- [62] Alvarado-Rodríguez CE, Klapp J, Mayoral E, Domínguez JM. GPU simulations of fluid and composition dispersion in a porous media with smoothed particle hydrodynamics. In: Gitler I, Klapp J, editors. *High Performance Computer Applications. ISUM 2015. Communications in Computer and Information Science*. Vol. 595. Cham: Springer; 2016. pp. 485-494. DOI: 10.1007/978-3-319-32243-8_34
- [63] Ji Z, Xu F, Takahashi A, Sun Y. Large scale water entry simulation with smoothed particle hydrodynamics on single- and multi-GPU systems. *Computer Physics Communications*. 2016; **209**:1-12. DOI: 10.1016/j.cpc.2016.05.016
- [64] Xia X, Liang Q. A GPU-accelerated smoothed particle hydrodynamics (SPH) model for the shallow water equations. *Environmental Modelling and Software*. 2016; **75**:28-43. DOI: 10.1016/j.envsoft.2015.10.002
- [65] Liang Q, Xia X, Hou J. Efficient urban flood simulation using a GPU-accelerated SPH model. *Environmental Earth Sciences*. 2015; **74**(11):7285-7294. DOI: 10.1007/s12665-015-4753-4
- [66] Mokos A, Rogers BD, Stansby PK, Domínguez JM. Multi-phase SPH modelling of violent hydrodynamics on GPUs. *Computer Physics Communications*. 2015; **196**:304-316. DOI: 10.1016/j.cpc.2015.06.020
- [67] Domínguez JM, Crespo AJC, Valdez-Banderas D, Rogers BD, Gómez-Gesteira M. New multi-GPU implementation for smoothed particle hydrodynamics on heterogeneous clusters. *Computer Physics Communications*. 2013; **184**(8):1848-1860. DOI: 10.1016/j.cpc.2013.03.008
- [68] Joselli M, Junior d SJR, Clua EW, Montenegro A, Lage M, Pagliosa P. Neighborhood grid: A novel data structure for fluids animation with GPU computing. *Journal of Parallel and Distributed Computing*. 2015; **75**:20-28. DOI: 10.1016/j.jpdc.2014.10.009
- [69] Hérault A, Bilotta G, Dalrymple RA. SPH on GPU with CUDA. *Journal of Hydraulic Research*. 2010; **48**(Suppl. 1):74-79. DOI: 10.1080/00221686.2010.9641247
- [70] Harada T, Koshizuka S, Kawaguchi Y. Smoothed particle hydrodynamics on GPUs. In: *Computer Graphics International Conference, Petrópolis, Brazil*. 2007;63–70

- [71] Bilotta G, H  rault A, Cappello A, Ganci G, Del Negro C. GPUSPH: A smoothed particle hydrodynamics model for the thermal and rheological evolution of lava flows. *Geological Society Special Publication*. 2016;**426**(1):387-408. DOI: 10.1144/SP426.24
- [72] Wu J-S, Zhang H, Dalrymple RA. Simulating dam-break flooding with floating objects through intricate city layouts using GPU-based SPH method. *Lecture Notes in Engineering and Computer Science*. 2013;**3**:1755-1760
- [73] Wu J-s, Zhang H, Yang R, Dalrymple RA, H  rault A. Numerical modeling of dam-break flood through intricate city layouts including underground spaces using GPU-based SPH method. *Journal of Hydrodynamics*. 2013;**25**(6):818-828. DOI: 10.1016/S1001-6058(13)60429-1
- [74] Rustico E, Bilotta G, H  rault A, Del Negro C, Gallo G. Advances in multi-GPU smoothed particle hydrodynamics simulations. *IEEE Transactions on Parallel and Distributed Systems*. 2014;**25**(1):43-52. DOI: 10.1109/TPDS.2012.340
- [75] Karp AH, Flatt HP. Measuring parallel processor performance. *Communications of the ACM*. 1990;**33**:539-543. DOI: 10.1145/78607.78614
- [76] Valdez-Balderas D, Dom  nguez JM, Rogers BD, Crespo AJC. Towards accelerating smoothed particle hydrodynamics simulations for free-surface flows on multi-GPU clusters. *Journal of Parallel and Distributed Computing*. 2013;**73**:1483-1493. DOI: 10.1016/j.jpdc.2012.07.010
- [77] Cercos-Pita JL. AQUAgpusph, a new free 3D SPH solver accelerated with OpenCL. *Computer Physics Communications*. 2015;**192**:295-312. DOI: 10.1016/j.cpc.2015.01.026
- [78] Gonnet P. Efficient and scalable algorithms for smoothed particle hydrodynamics on hybrid shared/distributed-memory architectures. *SIAM Journal on Scientific Computing*. 2015;**37**(1):C95-C121. DOI: 10.1137/140964266
- [79] Pazouki A, Serban R, Negrut D. A high performance computing approach to the simulation of fluid-solid interaction problems with rigid and flexible components. *Archive of Mechanical Engineering*. 2014;**61**(2):227-251. DOI: 10.2478/meceng-2014-0014
- [80] Cherfils JM, Pinon G, Rivoalen E. JOSEPHINE: A parallel SPH code for free-surface flows. *Computer Physics Communications*. 2012;**183**(7):1468-1480. DOI: 10.1016/j.cpc.2012.02.007
- [81] Chow AD, Rogers BD, Lind SJ, Stansby PK. Implementing an optimized ISPH solver accelerated on the GPU. 12th International SPHERIC Workshop, Ourense, 2017
- [82] Nie X, Chen L, Xiang T. Real-time incompressible fluid simulation on the GPU. *International Journal of Computer Games Technology*. 2015;**12** pages. DOI: 10.1155/2015/417417
- [83] Qiu LC. OpenCL-based GPU acceleration of ISPH simulation for incompressible flows. *Applied Mechanics and Materials*. 2014;**444-445**:380-384. DOI: 10.4028/www.scientific.net/AMM.444-445.380

- [84] Monaghan JJ. Smoothed particle hydrodynamics and its diverse applications. *Annual Review of Fluid Mechanics*. 2012;**44**:323-346. DOI: 10.1146/annurev-fluid-120710-101220
- [85] Violeau D. *Fluid Mechanics and the SPH Method: Theory and Applications*. Oxford University Press; 2012
- [86] Viccione G, Bovolín V, Pugliese Carratelli E. Simulating Flows with SPH: Recent Developments and Applications. *Intech Hydrodynamics—Optimizing Methods and Tools*. 2011;69-84. DOI: 10.5772/26132
- [87] Liu MB, Liu GR. Smoothed particle hydrodynamics (SPH): An overview and recent developments. *Archives Computation Methods Engineering*. 2010;**17**(1):25-76. DOI: 10.1007/s11831-010-9040-7
- [88] Gómez-Gesteira M, Rogers BD, Dalrymple RA, Crespo AJC. State-of-the-art of classical SPH for free-surface flows. *Journal of Hydraulic Research*. 2010;**48**(sup1):6-27. DOI: 10.1080/00221686.2010.9641242
- [89] Monaghan JJ. Smoothed particle hydrodynamics. *Reports on Progress in Physics*. 2005;**68**(8):1703-1759. DOI: 10.1088/0034-4885/68/8/R01
- [90] Monaghan JJ. Introduction to SPH. *Computer Physics Communication*. 1988;**48**:89-96
- [91] Vila JP. On particle weighted methods and smooth particle hydrodynamics. *Mathematical Models and Methods in Applied Sciences*. 1999;**9**(2):191-209
- [92] Wendland H. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*. 1995, 1995;**4**(1): 389-396. DOI: 10.1007/BF02123482
- [93] Oger G, Doring M, Alessandrini B, Ferrant P. An improved SPH method: Towards higher order convergence. *Journal of Computational Physics*. 2007;**225**(2):1472-1492. DOI: 10.1016/j.jcp.2007.01.039
- [94] Monaghan JJ. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*. 1992;**30**:543-574. DOI: 10.1146/annurev.aa.30.090192.002551
- [95] Viccione G, Bovolín V, Pugliese Carratelli E. Simulating fluid-structure interaction with SPH. *AIP Conference Proceedings*. 2012;**1479**(1):209-212. DOI: 10.1063/1.4756099
- [96] Monaghan JJ, Lattanzio JC. A refined particle method for astrophysical problems. *Astronomy and Astrophysics*. 1985;**149**:135-143
- [97] Dymond JH, Malhotra R. The Tait equation: 100 years on. *International Journal of Thermophysics*. 1988;**9**(6):941-951. DOI: 10.1007/BF01133262
- [98] Crespo AJC, Domínguez JM, Rogers BD, Gómez-Gesteira M, Longshaw S, Canelas R, et al. DualSPHysics: Open-source parallel CFD solver based on smoothed particle hydrodynamics (SPH). *Computer Physics Communications*. 2015;**187**:204-216. DOI: 10.1016/j.cpc.2014.10.004

- [99] Gómez-Gesteira M, Rogers BD, Crespo AJC, Dalrymple RA, Narayanaswamy M, Domínguez JM. SPHysics—Development of a free-surface fluid solver—Part 1: Theory and formulations. *Computers & Geosciences*. 2012;**48**:289-299. DOI: 10.1016/j.cageo.2012.02.029
- [100] Gómez-Gesteira M, Crespo AJC, Rogers BD, Dalrymple RA, Domínguez JM, Barreiro A. SPHysics—Development of a free-surface fluid solver—Part 2: Efficiency and test cases. *Computers & Geosciences*. 2012;**48**:300-307. DOI: 10.1016/j.cageo.2012.02.028
- [101] Verlet L. Computer “experiments” on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Physical Review*. 1967;**159**(1):98-103
- [102] Crespo AJC, Gomez-Gesteira M, Dalrymple RA. Boundary conditions generated by dynamic particles in SPH methods. *CMC: Computers, Materials, & Continua*. 2007;**5**(3): 173-184. DOI: 10.3970/cmc.2007.005.173
- [103] Chen X. Impacts of Overtopping Waves on Buildings on Coastal Dikes. Enschede, The Netherlands: Gildeprint; 2016