Computers & Fluids 69 (2012) 54–66

Contents lists available at SciVerse ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

Study of ship wave breaking patterns using 3D parallel SPH simulations

S. Marrone, B. Bouscasse, A. Colagrossi *, M. Antuono

CNR-INSEAN, The Italian Ship Model Basin, Rome, Italy

ARTICLE INFO

Article history:
Received 5 July 2011
Received in revised form 25 June 2012
Accepted 11 August 2012
Available online 1 September 2012

Keywords:
Smoothed particle hydrodynamics
Ship waves
Breaking waves
Hybrid MPI-OpenMP programming
Free-slip boundary condition
Free-surface flows

ABSTRACT

An analysis of the 3D wave pattern generated by a ship in stationary forward motion has been performed with a specific focus on the bow breaking wave. For this purpose a novel 3D parallel SPH solver has been designed and preliminarily tested using a specifically conceived geometry. Since the numerical effort to simulate 3D flows is considerable, an *ad hoc* hybrid MPI-OpenMP programming model has been developed to achieve simulations of hundred million of particles running on a computer cluster. The outcomes have been compared with experimental measurements and numerical results from RANS calculations.

1. Introduction

The present work deals with the analysis of bow waves generated by a ship in steady-state forward motion with specific focus on the prediction/description of breaking events. From a physical point of view, it is important to understand how the bow wave breaking process develops and how it modifies the flow field. For example, the evolution of the ship wakes is crucial for the ship signature and has relevant implications in military fields.

In the last years, the progress made by the experimental and numerical studies has shed light on the main features of this phenomenon. At present, some of the most used numerical models for free-surface flows are RANS schemes with Volume of Fluid or Level-Set algorithms. These solvers are able to deal with 3D breaking wave phenomena and, specifically, the evolution of bow breaking waves (see, for example, [1,2]). Recently, SPH method has been used to study the dynamics of breaking waves proving to be a reliable numerical method for these phenomena (see e.g. [3,4]). As for the naval hydrodynamic context, several examples of SPH applications can be found in literature (see e.g. [5–9]). However, an in-depth study of SPH capability in dealing with ship-induced 3D wave patterns is still missing. As far as the analysis of bow waves is concerned, Colagrossi et al. [10] proposed a 2D+t SPH model that gave a good qualitative description of the breaking phenomenon. A further inspection of this model has been provided in [11] where the minimum discretization required to catch the breaking inception has been determined.

In the present work, a 3D SPH solver is described and tested in order to model the 3D wave pattern generated by a ship in steady-state forward motion. This purpose is very demanding in the SPH framework since, differently from the RANS schemes, the use of varying spatial discretization is not straightforward. With respect to this, Hernquist and Katz [12] and Nelson and Papaloizou [13] proposed SPH schemes based on a h-variable formulation (here, h indicate the reference length for the SPH interpolation). Unfortunately, these schemes are quite complex and difficult to parallelize and may be not robust enough when strong particle mixing occurs. For these reasons, it has been preferred not to implement h-variable formulations.

The request for a large enough fluid domain to avoid wave reflection at the boundaries and the choice of an SPH scheme with a constant h imply that a standard 3D SPH simulation generally contains about 10^8 particles. For this reason, a hybrid MPI-OpenMP programming model has been developed.

A preliminary assessment of the capabilities of the 3D solver has been performed on a test case specifically conceived. The comparison with the results obtained by using the 2D+t SPH model allowed to find the minimum discretization necessary to describe the plunging jet. Then, the 3D SPH solver has been used to simulate the wave pattern generated by the Alliance ship in forward motion. The SPH results have been compared with experimental measurements and with the simulations obtained by a RANS solver coupled with a Level Set algorithm.

* Corresponding author.
E-mail address: a.colagrossi@insean.it (A. Colagrossi).

0045-7930/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compfluid.2012.08.008

2. Numerical scheme

2.1. The adopted SPH model

As generally assumed by SPH practitioners, the fluid is considered to be barotropic, weakly-compressible, and inviscid. The flow evolution is then governed by

$$\begin{cases} \frac{D\rho}{Dt} = -\rho \nabla \cdot \boldsymbol{u}; \qquad p = c_0^2(\rho - \rho_0) \\ \rho \frac{D\boldsymbol{u}}{Dt} = -\nabla p + \rho \boldsymbol{f} \end{cases} \tag{1}$$

where ρ, p and $\boldsymbol{u}$ are respectively the density, pressure and velocity fields; $\boldsymbol{f}$ is the body force field, ρ_0 is the density at the free surface and c_0 is the reference speed of sound.

In this framework, the δ-SPH scheme proposed by Antuono et al. [14] and validated in [15] and [16] is adopted. In such SPH model a proper artificial diffusive term is used into the continuity equation in order to remove the spurious numerical high-frequency oscillations in the pressure field. As usually done for weakly-compressible SPH models, an artificial viscous term is also included in the momentum equation for stability reasons (see e.g. [17]).

Since the artificial viscosity does not correspond to a physical viscosity, it has been possible to implement a free-slip condition along solid profiles as if the fluid was inviscid. Note that the viscosity play a negligible role on the generation of the ship wave pattern and numerical solvers based on potential flow theory are often used to calculate the wave field. The main limit of those solvers is the breaking wave phenomena that cannot be taken into account in the potential theory framework. Conversely, the SPH model can simulate breaking events.

Differently from the RANS Level-Set solver, the SPH scheme does not implement any turbulence closure. This choice is motivated by the fact that, despite turbulence plays a relevant role during wave breaking, no turbulence model is validated for such kind of flow. For this reason the widespread "artificial viscosity" approach has been preferred. Incidentally, we underline that the implementation of turbulence in SPH schemes is an ongoing field of research (see e.g. [18,19]).

2.2. Boundary conditions

One of the main advantages of the weakly compressible SPH formulation is that the free-surface boundary conditions are implicitly satisfied (for details see [20]). Thanks to this feature, it is not necessary to detect at each time step the particles belonging to the free surface to enforce the dynamic boundary condition (i.e. the pressure value). This is a worthy advantage especially for 3D flows where the free surface configuration/evolution may be very complex. Conversely, it is well known in the SPH literature that the enforcement of the boundary condition on a solid surface can be quite a complicated matter. Different techniques have been presented over the years, those based on ghost-fluid method are among the most promising (see e.g. [17,21,16]). Unfortunately, this kind of techniques are very difficult to manage in 3D, especially when solid surfaces are characterised by complex geometries. To enforce solid boundary conditions along the ship hull and the boundary walls, a robust and simple method is needed. To this purpose, the technique recently presented by De Leffe et al. [22] has been adopted. Such a technique has been developed in the context of Riemann-SPH methods and in the present work is reformulated in the context of the standard SPH scheme. In the following the main points of such a technique are reported.

The SPH gradient of a generic function f is approximated by the convolution integral:

$$\langle \nabla f \rangle(\boldsymbol{r}) = \int_{\Omega} \nabla f(\boldsymbol{r}^*) W(\boldsymbol{r} - \boldsymbol{r}^*) dV^* \tag{2}$$

where Ω is the fluid domain and W a kernel function. Using an integration by parts and the symmetry properties of W, the previous relation can be rewritten as follows (see e.g. [12]):

$$\langle \nabla f \rangle(\boldsymbol{r}) = -\int_{\Omega} f(\boldsymbol{r}^*) \nabla W(\boldsymbol{r} - \boldsymbol{r}^*) dV^* + \int_{\partial\Omega} f(\boldsymbol{r}^*) W(\boldsymbol{r} - \boldsymbol{r}^*) \boldsymbol{n}^* dS^* \tag{3}$$

With respect to (2), the gradient operator has been moved from the function f to W and a second term appears as a surface contribution along the boundary of the fluid domain. Here, $\boldsymbol{n}$ represents the normal unit vector to the boundary $\partial\Omega$ pointing outwards the fluid domain. Starting from this relation, it is possible to derive the discrete differential operators used in the SPH model for a generic particle i (see e.g. [20]):

$$\begin{cases} \langle \mathrm{div} \boldsymbol{u} \rangle_i = \sum_j (\boldsymbol{u}_j - \boldsymbol{u}_i) \cdot \nabla W_{ij} V_j + \sum_k (\boldsymbol{u}_k - \boldsymbol{u}_i) \cdot \boldsymbol{n}_k W_{ij} S_k \\ \langle \nabla p \rangle_i = \sum_j (p_j + p_i) \nabla W_{ij} V_j + \sum_k (p_k + p_i) \boldsymbol{n}_k W_{ij} S_k \end{cases} \tag{4}$$

The summations on j are performed on the fluid particles in the neighbourhood of the particle i while the summations on k are made on the nodes of the discretized boundary $\partial\Omega$ (see sketch in Fig. 1). As mentioned at the beginning of this section, it is not necessary to evaluate the boundary terms on the free surface while they have to be considered along the solid surface to enforce the solid boundary conditions. In the present work a free-slip condition is used in order to avoid the complexities related to the evaluation of boundary layers along the ship hull. This simplification is well-posed since the viscous effects on the wave pattern are generally negligible.

In the first equation of (4) the value $\boldsymbol{u}_k$ is replaced by the velocity of the solid surfaces at the position $\boldsymbol{r}_k$. Conversely in the second equation of (4) the pressure p_k on the solid surfaces is not known a priori. Then, a Taylor expansion around the particle i is used:

$$p_k = p_i + \nabla p_i \cdot (\boldsymbol{r}_k - \boldsymbol{r}_i) + \mathcal{O}(\|\boldsymbol{r}_k - \boldsymbol{r}_i\|^2) \tag{5}$$

that is:

$$p_k = p_i + \frac{\partial p_i}{\partial n_j} (\boldsymbol{r}_k - \boldsymbol{r}_i) \cdot \boldsymbol{n}_k + \frac{\partial p_i}{\partial \tau_k} (\boldsymbol{r}_k - \boldsymbol{r}_i) \cdot \boldsymbol{\tau}_k + \frac{\partial p_i}{\partial b_k} (\boldsymbol{r}_k - \boldsymbol{r}_i) \cdot \boldsymbol{b}_k + \mathcal{O}(\|\boldsymbol{r}_k - \boldsymbol{r}_i\|^2) \tag{6}$$

where $\boldsymbol{\tau}_k$ and $\boldsymbol{b}_k$ are the unit vectors tangent to the solid profile at $\boldsymbol{r}_k$, $(\boldsymbol{n}_k, \boldsymbol{\tau}_k, \boldsymbol{b}_k)$ being an orthogonal triad. The subsequent step is to approximate the partial derivatives of the pressure field along the normal and tangential directions. In the latter case, it could be

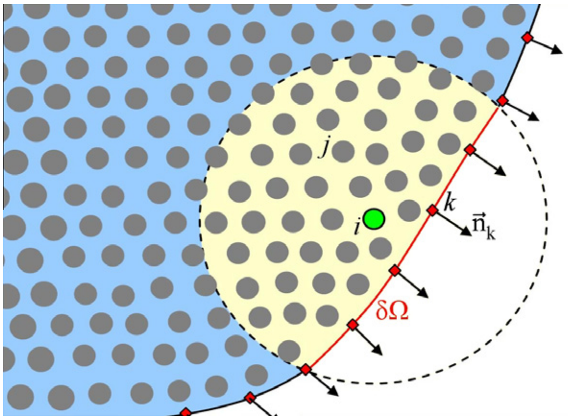

Fig. 1. Schema adopted for the solid boundary treatment.

simply assumed that the body forces are predominant with respect to the inertial forces:

$$\frac{\partial p_i}{\partial \tau_k} = \rho \boldsymbol{g} \cdot \boldsymbol{\tau}_k + \mathcal{O}(h) \qquad \frac{\partial p_i}{\partial b_k} = \rho \boldsymbol{g} \cdot \boldsymbol{b}_k + \mathcal{O}(h) \tag{7}$$

The normal pressure gradient in (6) can be derived by the projection of momentum equation in the $\boldsymbol{n}_k$ direction:

$$\frac{\partial p_i}{\partial n_k} = \rho(\boldsymbol{g} \cdot \boldsymbol{n}_k - \boldsymbol{a}_i \cdot \boldsymbol{n}_k) \tag{8}$$

where $\boldsymbol{a}_i$ is the acceleration of the i-th particle. A simple approximation of $\boldsymbol{a}_i$ is obtained by imposing the impermeability condition along the solid profiles and is briefly described in the following. At time t the particle i close to the solid body is characterised by a normal velocity $\boldsymbol{u}_i \cdot \boldsymbol{n}_k$. Due to the impermeability of the solid surface, a wave signal, travelling at the speed of sound c_0, is reflected by the wall and comes back to the particle i after a time interval equal to $(\boldsymbol{r}_k - \boldsymbol{r}_i) \cdot \boldsymbol{n}_k / c_0$ nullifying the normal velocity $\boldsymbol{u}_i \cdot \boldsymbol{n}_k$. Therefore the acceleration $\boldsymbol{a}_i \cdot \boldsymbol{n}_k$ can be expressed as follows:

$$\boldsymbol{a}_i \cdot \boldsymbol{n}_k = -\frac{c_0(\boldsymbol{u}_i \cdot \boldsymbol{n}_k)}{(\boldsymbol{r}_k - \boldsymbol{r}_i) \cdot \boldsymbol{n}_k} \tag{9}$$

Then, substituting (8) and (7) in (6), we, finally, get:

$$p_k = p_i + \rho c_0(\boldsymbol{u}_i \cdot \boldsymbol{n}_k) + \rho \boldsymbol{g} \cdot (\boldsymbol{r}_k - \boldsymbol{r}_i) \tag{10}$$

This expression is used to evaluate the boundary term of the pressure gradient given in (4). That Eq. (10) is the same derived in [22] in the framework of Finite Volume with Characteristic Flux scheme.

Finally, to study the problem of a ship moving forward at a constant speed, the definition of suitable inflow/outflow boundary conditions is required. This is achieved through a 3D-extension of the algorithm proposed by Federico et al. [23] where suitable sets of inflow and outflow particles are defined to enforce upstream and downstream conditions.

2.3. Parallelization strategy

To perform intensive computations like the simulation of the bow wave breaking, it is necessary to develop a 3D parallel solver able to run efficiently on a computer cluster. The solver has been built targeting a simulation of hundreds of millions particles. The formalism used with the compressible-SPH model is fully explicit, that is a real advantage to parallelize the code. However, because of the Lagrangian nature of the method, the parallelization of the code is not trivial and specific algorithms have to be designed and validated (see e.g. [24–27]).

Contrary to mesh-based methods for which a fixed grid is employed, the SPH method relies on moving particles advected by the calculated fluid flow. Depending on the problem at hand, large particle displacements may occur. The parallel scheme must take this specificity into account.

The Lagrangian nature of the SPH scheme also implies a major disorder of the particle distribution and, consequently, the presence of scattered data in memory. Note that for mesh-based methods, the connections are fixed during the calculation: the neighbourhood of a given cell of interest remains the same throughout the calculation. Thus, a recurrent pattern of interpolation can be used, simplifying the parallelization scheme. Unfortunately, such a simplification is not possible in the SPH formalism, the neighbourhoods of SPH particles constantly changing during the calculation. Therefore, the core of the parallelization algorithm is centred on the neighbour search procedure. The communication of neighbouring particles belonging to different processors must be handled considering the update and trade-off of the processor working loads. The parallelization strategy adopted here combines a domain decomposition, performed on distributed memory architectures through MPI paradigm, and a data decomposition, implemented on shared memory architectures through OpenMP directives. The simulations shown in the present work have been performed at the CASPUR supercomputing centre in Rome. Its architecture is composed by dual quad-core AMD Opteron (tm) Processor 2352 “Barcelona” blades connected through InfiniBand switches.

In order to simplify the MPI programming, some specific hypotheses have been assumed in the present problem. In simulating ship’s waves the fluid domain is characterised by a dominant length corresponding to the main flow direction. Therefore it is possible to divide the fluid domain into parallel slices along the main flow direction, say the x-axis, equally balancing the number of particles per processor (Fig. 2). In this way there is just one direction of communication between adjacent nodes. This allows simplifying the parallelization algorithm: the more the simulated phenomenon has a principal direction of evolution, the more the parallelization will be efficient.

To obtain a fast neighbour search, it is common practice in SPH to create a regular grid overlapping the computational domain. Each cell of the grid is a cube with size equal to the kernel radius. In this way the neighbours of each particle are found in the cells surrounding the particle one (Cell Linked List algorithm see e.g. [28]). At each time step the list of particles and their belonging cells are calculated. In order to further simplify the problem, the computational domain is separated into sub-domains that rely on the grid used for particle neighbour search. The idea is to use this grid to speed up the operations between processors.

Since the list of the particles belonging to different cells is already calculated, it can be used to quickly update the list of particles belonging to different processors and to determine the buffer particles to be communicated to the adjacent processors. Considering a single processor, the buffer particles are virtual particles that are received from the two adjacent processors and are needed to complete the interpolation of particles placed in proximity of the processor boundaries. In other words the buffer particles give contributions to the equation of motion of the local particles but their physical quantities are not updated because they belong to another processor.

According to the parallelization strategy adopted, each processor extends over a finite number of cells along the x-axis, overlapping for one cell the adjacent processors (see Fig. 3), while it covers all the computational domain along the y-axis and z-axis. In this way, a single layer of cells is scanned looking for particles to be sent to the adjacent processor as buffer particles. With this approach, it is also straightforward for each processor to update its belonging particles. As already mentioned, the particles cross naturally the processors boundaries. These particles are sent together with buffer particles and hence are assigned in the cells of the

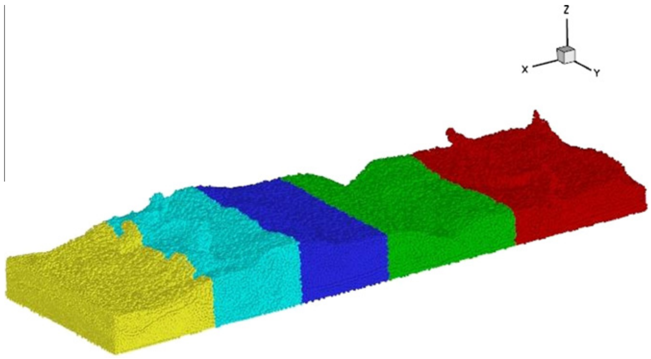

Fig. 2. Example of division of the fluid domain over processors.

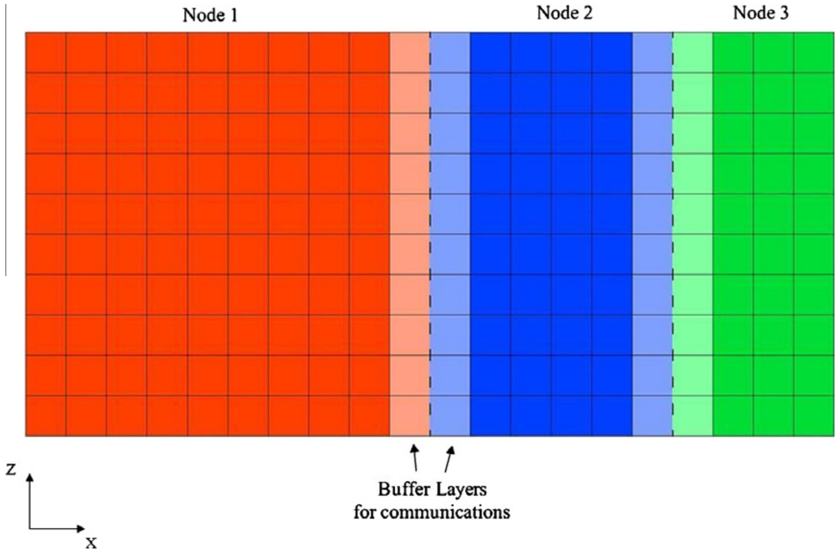

Fig. 3. Sketch of the cell grid and processor sub-domains.

receiving processor. Then, each processor scans the cells of its own sub-domain to update the list of particles.

As concerns the load balancing, it is necessary to pay attention to some constraints resulting from the parallelization strategy adopted. Firstly, the sub-domain size of the processor can change just along the *x*-axis. Secondly, the length of the sub-domains has to be equal to a multiple of the cell size, meaning that each processor has to exchange at least the amount of particles contained in a cell plane parallel to the *yz*-plane. During the simulation, the balancing algorithm is called whenever the most loaded processor has a number of particles that exceeds the average number of particles per processor by a specific threshold. The procedure is designed to balance processors load through a diffusive-like process. Each processor compares its number of particles with the one of the adjacent processor. If it is greater, the cell planes (starting from the one nearest to the receiver) that contains a number of particles as close as possible to the half of the unbalance between the two processors are sent to the adjacent processor. In this way it is not necessary to use collective communications to balance CPU loads.

During the simulation, the balancing algorithm is called whenever the most loaded processor has a number of particles that exceeds the average number of particles per processor by a specific threshold. The cost of the balance algorithm itself is about 2% of a whole Runge–Kutta cycle, resulting in a negligible overhead on the whole computational cost.

In the SPH scheme the particle interaction loops represents the largest part of the computational costs, therefore a highly efficient data decomposition can be performed with OpenMP directives eliminating race conditions. In particular, the pair symmetry of particle interactions is not used leading to a slow down of the serial code but, on the other hand, allowing for an effective fork of the interaction loops. Similar considerations have been drawn also by other research groups regarding GPU implementations of SPH (see e.g. [29]). The resulting parallel efficiency is over 90% with eight cores considering the serial code without taking advantage of pair symmetry.

The advantage of adopting a hybrid implementation relies on the fact that the simple 1D domain decomposition leads to a loss of efficiency when increasing the number of cores. Indeed, the smaller size of each sub-domain causes a larger ratio between communication and calculation costs. A higher efficiency is obtained by introducing a data decomposition of particle interaction loops through OpenMP paradigm which effectiveness does not depend upon particles spatial configuration.

An example of the overall speed-up is displayed in left plot of Fig. 4 for the problem of a uniform flow in a rectangular open channel. The curves are obtained firstly increasing the number of OpenMP threads used (from 1 up to 8) and then increasing the number of MPI processes (from 1 up to 16). This result clearly shows the effectiveness of the parallelization strategy adopted. Indeed, simulations involving at least 1 million of particles can be run on 128 cores (eight threads per 16 nodes) with a fair scalability. Increasing the number of particles the same parallel performance can be achieved on more cores (in particular on Fig. 4 an almost linear scalability is obtained using 256 cores for a simulation with 40 millions of particles). Right plot of Fig. 4 shows the gain of the hybrid implementation against the pure MPI programming model. The CPU time values are obtained using 128 cores for a simulation involving 32 millions of particles and varying the combination of the number of MPI processes and OpenMP threads.

3. Preliminary computations using a simplified geometry

In order to test the capacity of the 3D SPH code to simulate the breaking wave, a simplified test is devised. The geometry is a channel where the right wall (in the stream-wise direction) is a ruled surface. This surface looks similar to a ship bow characterised by a large flare, see Fig. 5. The flow encountering it creates an overturning wave generating intense splash-up cycles. The inflow velocity is fixed and a free outflow condition is given. For the lateral and bottom boundaries, solid walls are used. The height of the water inflow is indicated by H. The velocity of the inflow is indicated by U_{flow} and the Froude number respect to H is defined as $Fn_H = U_{flow}/\sqrt{gH}$.

The ruled surface, say S, can be derived as a linear combination of the parametric representation of two lines $\alpha(t)$ and $\beta(t)$ as follows:

$$S(t,u) = u\alpha(t) + (1-u)\beta(t) \qquad u \in [0,1], \quad t \in [0,5]$$

$$\alpha(t) : \begin{cases} x = 5H \\ y = 0 \\ z = Ht \end{cases} \qquad \beta(t) : \begin{cases} x = 25H \\ y = \frac{\sqrt{2}}{2}Ht + 5H \\ z = \frac{\sqrt{2}}{2}Ht \end{cases} \tag{11}$$

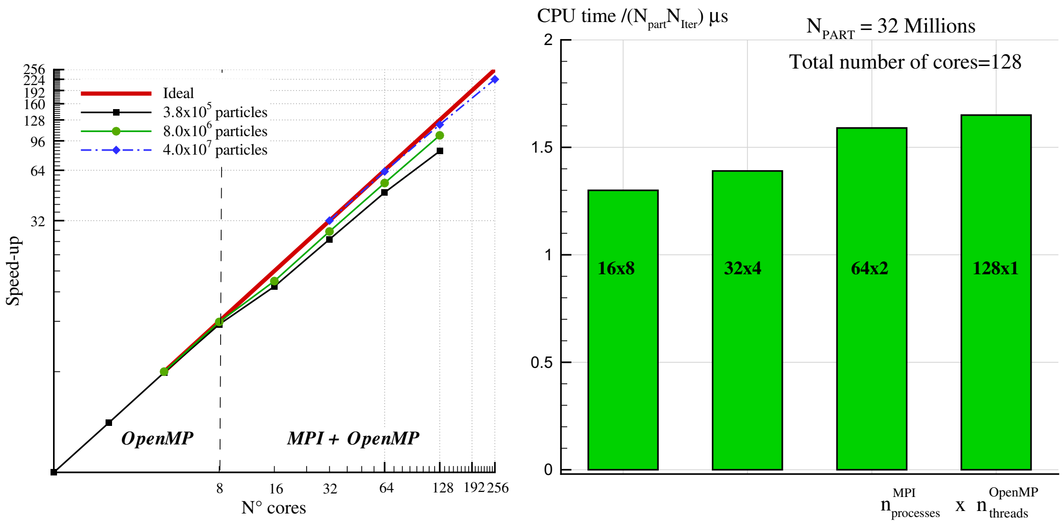

Fig. 4. Performance of the hybrid parallel model for the test case described in Section 3. Left plot: Overall speed-up obtained firstly increasing the number of OpenMP threads used (from 1 up to 8) and then increasing the number of MPI processes (from 1 up to 16). Right plot: normalised CPU time costs for a simulation performed varying the combination between MPI processes and OpenMP threads using 128 cores.

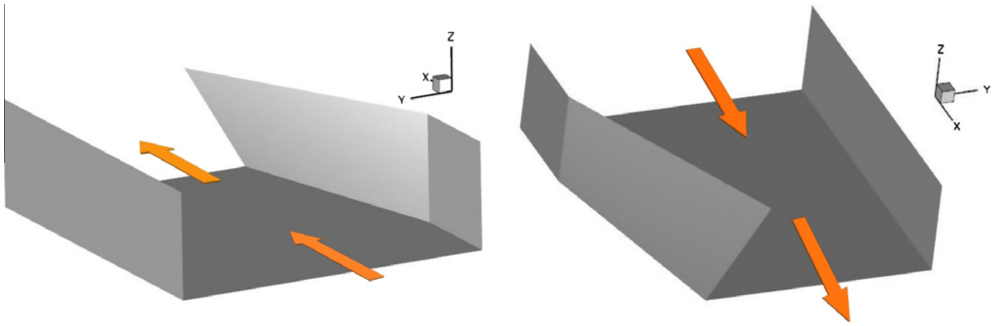

Fig. 5. Geometry of the channel used to generate an overturning wave. View from the inflow (left) and the outflow (right). Arrows indicate the direction of the flow.

Before running the 3D simulation a preliminary analysis has been done using a 2D+t SPH model. This model has been adopted in [30] to gain the physical understanding of the genesis of divergent bow waves. The 2D+t approach allows to approximate the ship motion through a set of equations which are mathematically equivalent to those governing the unsteady two-dimensional free-surface flow generated by a deformable body in the vertical plane transverse to the ship. This deformable body coincides with the ship cross-section in that plane which deforms as the ship moves forward. The main idea is that longitudinal gradients of relevant flow quantities are small compared with vertical and transverse gradients. Consequently there is no transient state and the flow is affected only by the upstream conditions.

The combination of the SPH model to the 2D+t approach has been done in [11,31]. In the 2D+t framework a relevant parameter is the local Froude number $Fr_x = U_{flow}/\sqrt{gx}$ where x is the distance from the bow section in the water-plane area. To be consistent with the 2D+t hypothesis this local Froude number has to be of order of one, therefore with increasing x values the 2D+t evaluations become only qualitative unlike in the 3D simulation.

In this test the Froude number with respect to the depth $Fr_H = U_{flow}/\sqrt{gH}$ has been set equal to 2, this means with this choice the Fr_x remains $\mathcal{O}(1)$ for almost the whole channel length. Therefore this preliminary 2D+t simulation can reproduce the global evolution of the breaking wave giving the main information regarding the relevant scale lengths to be captured in the 3D simulations.

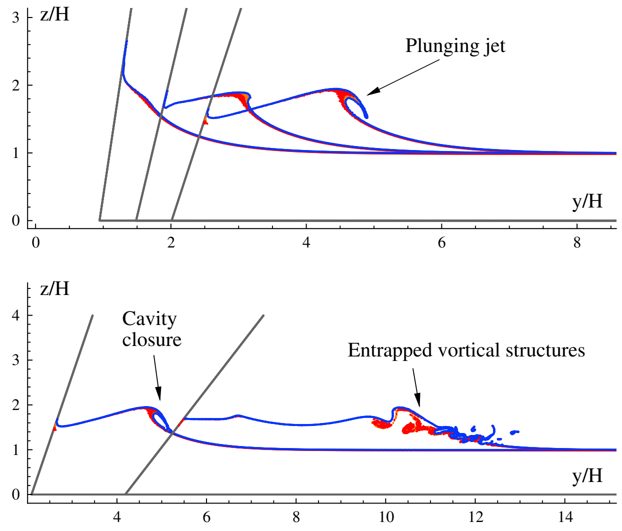

Fig. 6. 2D+t free surface configurations on transversal plane for a flow in an open channel (see Fig. 5 for the channel geometry) with $U_{flow} = 2\sqrt{gH}$. The particles belonging to the free surface are coloured in blue whereas the lagrangian trackers identified as the two top layers of particles at $t = 0$ are coloured in red and orange. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Discretization data for the four simulations of the simplified geometry test case.

	H/dx	Npart $\times 10^6$	H_{pj}/dx
R1	25	4	12.5
R2	30	6.5	15
R3	50	32	25
R4	60	50	30

The 2D+t results are shown in Fig. 6. The particles belonging to the free surface (FS) are plotted in blue whereas the lagrangian trackers identified as the two top layers of particles at $t = 0$ are plotted in orange and red. The top plot of Fig. 6 shows the formation of a steep wave which forms a plunging jet with a height, H_{pj}, of about 0.5*H*. In the following time instants (bottom plot) the jet hits the underlying free surface causing the subsequent splashing-up phenomena and the entrapping of vortical structures, similarly as discussed in [10,32]. The vortical structures flow downstream under the free surface,roughly parallel to the mainstream direction. As observed experimentally in [10] these vortical structures contain air entrained during breakings and resemble air tunnels, which can be seen from above. The presence of these air tunnels also represent an issue for the ship signatures.

The time evolution of the lagrangian trackers enables to highlight this generation of vorticity from the free surface breaking (for details see [16]). To simulate this kind of flow a proper spatial discretization is needed and in particular the plunging jet needs to be properly discretized. In particular the 2D+t analysis suggests that at least 20 particles in the jet height should be used to catch the wave breaking:

$$dx < H_{pj}/20 \tag{12}$$

This relation implies that particles with a size lower than $H/40$ should be used for simulating this flow correctly.

Concerning the 3D simulation, four tests (labelled *R1*, *R2*, *R3* and *R4* see Table 1) with different spatial resolution have been performed in order to verify the convergence of the numerical scheme as well as the validity of the constrain (12).

In Fig. 7, transversal cuts relative to each test are shown just before the closure of the plunging jet. These cuts refer to the time instant $t = 14(g/H)^{1/2}$ when the unsteadiness of the initial transient state can be considered ended. As expected from the 2D+t analysis the spatial resolution of the *R1* test is insufficient to reproduce correctly the plunging jet while very few differences are noticed between the two finest cases *R3*, *R4*.

In Fig. 8, the shapes of the plunging jet obtained by the 2D+t and the *R4* simulations have been compared. Even if differences between the two models are evident the size of the jets are quite close and also the particle velocities are very similar. This confirms the validity of the 2D+t approach as preliminary analysis of such type of flow.

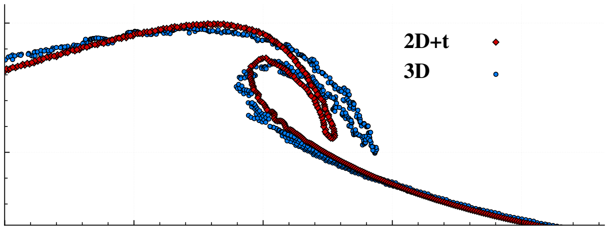

Fig. 8. Comparison of the plunging jet shape obtained with 2D+t and 3D simulations.

Fig. 9 shows the contour plot of the wave elevation for the four 3D cases. From these top views the convergence of the numerical solution can be appreciated. The breaking phenomena are well caught only for the two finest discretization in which the typical scar-lines generated by successive splash-ups are well visible (see e.g. [32]). For *R1*, *R2* the plunging jet is under resolved and the consequent splash-ups are too weak, consequently, the related particles remains on the upper layers without penetrating in the fluid bulk.

Similarly to what has been done in the 2D+t analysis for the *R4* test, lagrangian trackers have been defined using the upper layer of the inflow particles. Top plot of Fig. 10 shows the entrapment of such trackers (particles coloured in orange) in vortical structures generated during the successive splash-ups. To appreciate the 3D nature of such a vortical structure in the bottom plot of Fig. 10 a 3D underwater view towards the free surface is depicted.

Finally Fig. 11 shows a 3D view of the free surface which is coloured according to the modulus of the vorticity, highlighting the high vorticity regions on the splash-up front.

4. Simulation of a ship in steady forward motion

4.1. Description of the computational domain

The main objective of this work is to simulate the free surface flow around a ship in forward motion. In particular, the hull of the Alliance ship has been selected since experimental data are available to validate the present numerical model (see Fig. 12 and Table 2 for body plan and the hull characteristics).

The study is divided in two parts:

- the validation of the code on the global wave pattern;
- a detailed study of the evolution of the bow wave and the breaking inception.

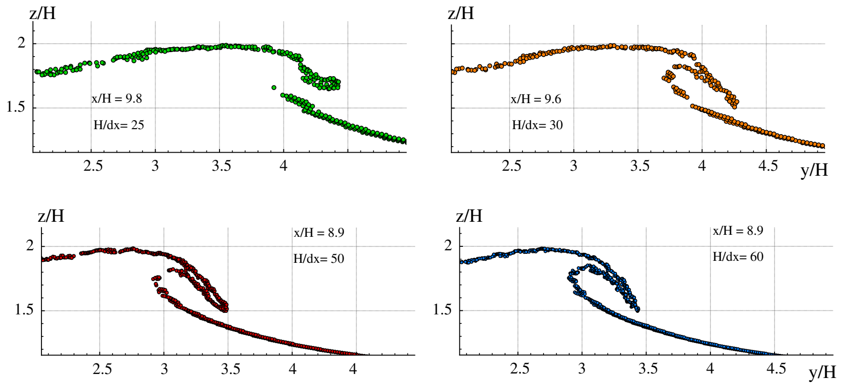

Fig. 7. Transversal cut for the simplified geometry simulations *R1* (top-left), *R2* (top-right), *R3* (top-left) and *R4* (top-right).

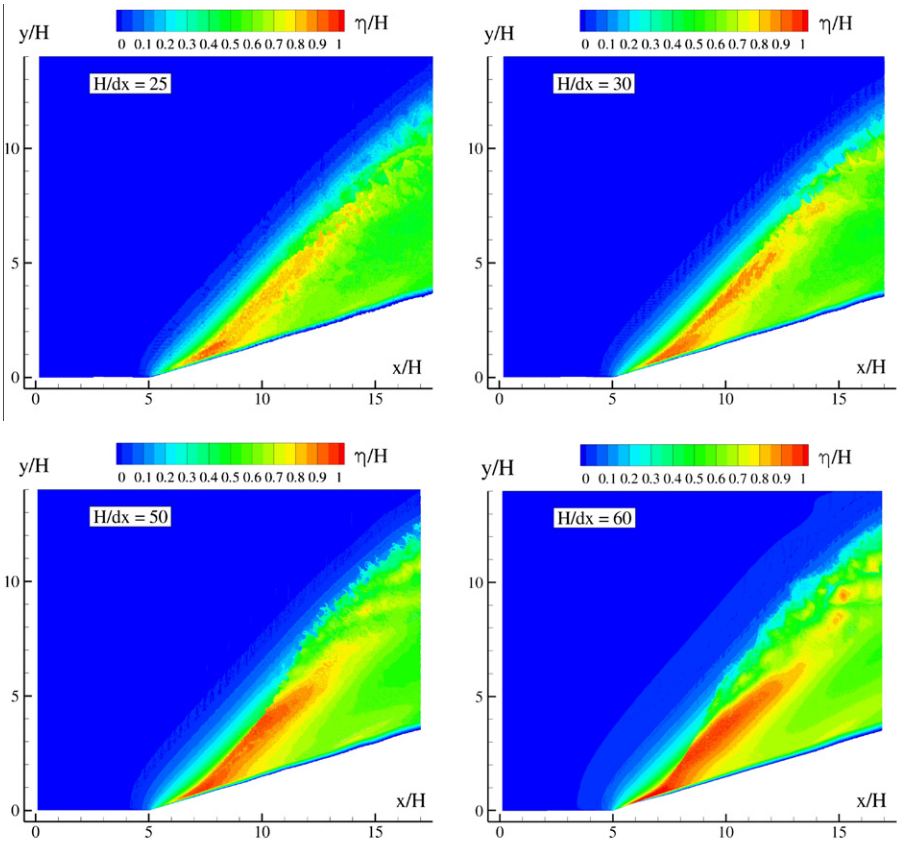

Fig. 9. Wave elevation at the steady state for the simplified geometry simulations *R1* (top-left), *R2* (top-right), *R3* (top-left) and *R4* (top-right).

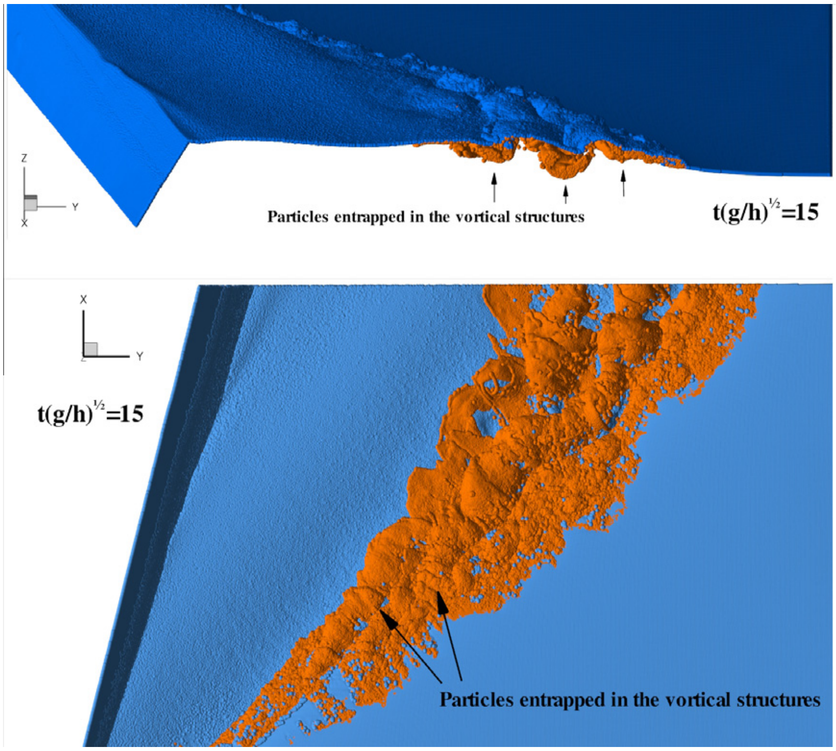
Z
Y
X
Particles entrapped in the vortical structures
t(g/h)^½=15
X
Y
t(g/h)^½=15
Particles entrapped in the vortical structures

Fig. 10. 3D view of the overturning wave generated in the simplified geometry at $t = 15(H/g)^{1/2}$. The particles belonging to the free surface are coloured in blue whereas the lagrangian trackers identified as the first top layer of particles at the inflow are coloured in orange. Top plot: view from the outflow; bottom plot: underwater view towards the free surface. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

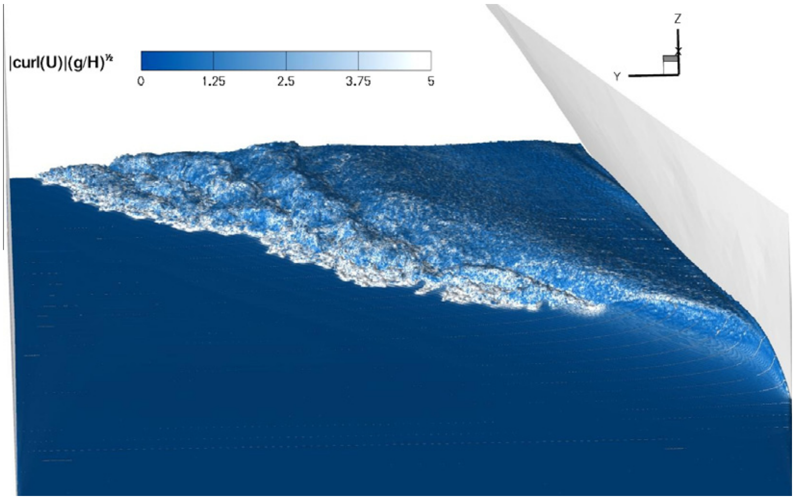

Fig. 11. Contour of the modulus of the vorticity, $t = 15(H/g)^{1/2}$.

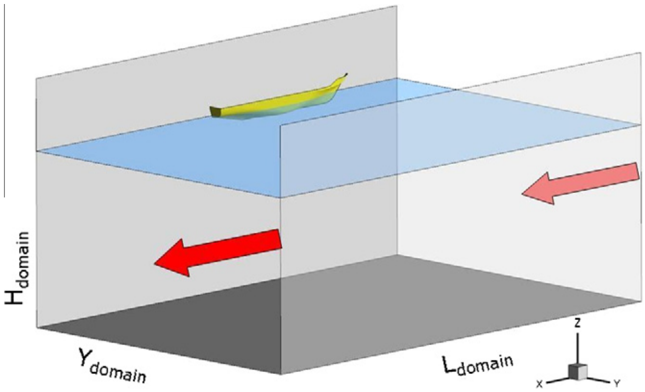

Fig. 12. Sketch of the problem.

Table 2
Alliance ship parameters.

	Full scale	Numerical scale
Length *Lpp* (m)	82	1
g (m/s^2)	9.81	1
Breadth *B* (m)	15.2	0.186
Draft *D* (m)	5.2	0.063
Displacement $\varDelta$	2920 (*t*)	5.3 (m^3)

The simulations are carried on in the frame of reference of the ship, z is the vertical axis (opposite sense respect to gravity), x the longitudinal axis of the hull (stream-wise positive), and y the transversal axis of the model. A forced inflow is fixed at 0.25*Lpp* from the bow, and a free outflow condition is given at the farthest possible distance from the stern. As concerns the lateral and bottom boundaries, solid walls are used (no models to enforce the radiation condition are present in SPH literature). The Froude number of the computation is set to $Fr = 0.328$ (18.1 knots in full scale) in order to compare the numerical outcome with the experimental data [33]. A qualitative sketch of the numerical domain is depicted in Fig. 12 while its dimensions are discussed in the following.

In 3D the computational effort to reach a sufficient discretization is considerable because the fluid domain has to be large enough to simulate a deep water behaviour and avoid wave reflections at the boundaries [35]. As a consequence, the resulting testing condition is necessarily a compromise.

4.2. Pre-processing of ship geometry

In the SPH solver the body surface is defined by a list of points, each being associated with normal vectors. Each point represents a piece of surface. The discretization of the body surface is achieved importing the CAD design into a software for mesh generation. A triangularization of the surface is performed and the area and normal of each triangle is associated with a point placed at the centroid. The numerical scale of the ship model is displayed in Table 2 whereas the ship body plan of the Alliance vessel is depicted in Fig. 13.

4.3. Remarks on breaking bow wave from the 2D+t analysis

For the analysis that follows, it is useful to take advantage of 2D+t SPH studies to simulate the breaking bow wave (see Section 3 and [11]). For high or medium Froude numbers, the evolution of

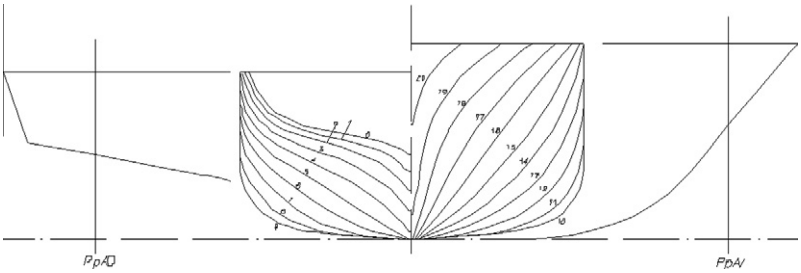

Fig. 13. Body plan of the Alliance vessel.

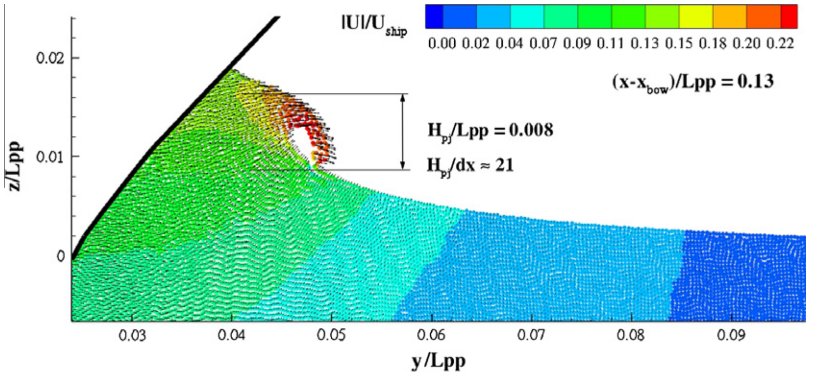

Fig. 14. 2D+t around the Alliance hull (see [31]): snapshots of plunging jet of the bow wave for $Fr = 0.33$. Ship section $x/Lpp = 0.13$.

Table 3
Numerical parameters for the wave pattern computations.

	L_{domain} [$\times Lpp$]	Y_{domain} [$\times Lpp$]	H_{domain} [$\times Lpp$]	Npart [$\times 10^6$]	D/dx [$\times (g/Lpp)^{1/2}$]	t_f [$\times Lpp$]	dx	Multires ratio
Ship1	2.25	0.8	0.56	65	25.2	10	0.0025	1
Ship2	1.25	0.8	0.26	17	25.2	5	0.0025	1
Ship3	1.25	0.8	0.26	22	38	4	0.0017	1.5
Ship4	1.25	0.8	0.26	145	75.9	4	0.0008	1.5

the bow wave is characterised by the generation of a first plunging jet that triggers a cyclic splash-up process. The 2D+t simulations can be useful to get an estimation of the size of the plunging jet.

The Froude number used for the Alliance vessel in the 2D+t scheme is the same used for the 3*D* simulation presented in the next section (see [31]). Note that $Fr = 0.328$ is too small for a correct

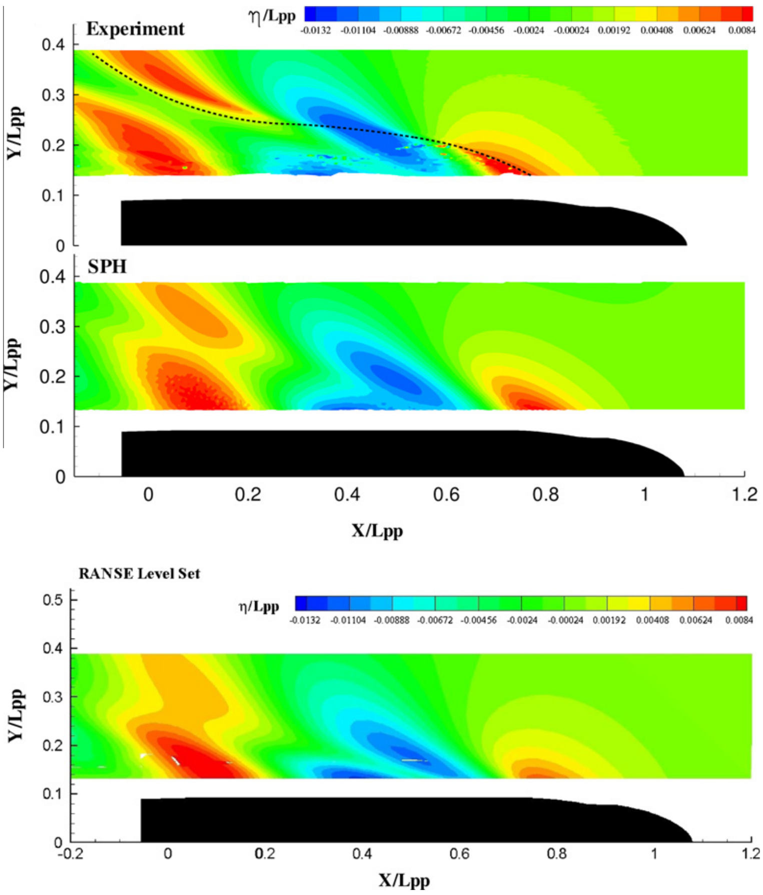

Fig. 15. Contours of the wave elevation for Fr = 0.328; experimental results (top), SPH results (middle), RANS results (bottom). The dashed line highlights the breaking wave front.

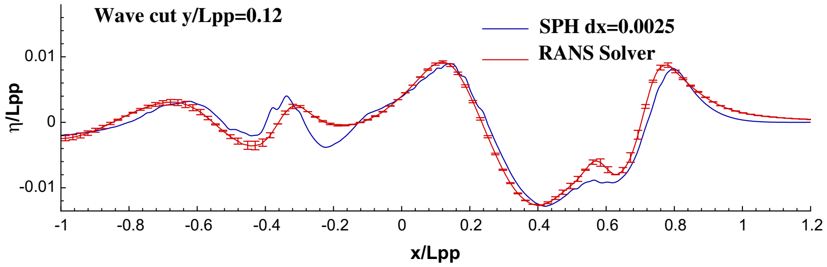

Fig. 16. Longitudinal wave cut a $y/Lpp = 0.12$. The red plot represents the result for the RANS calculation while the blue plot is referred to the SPH simulation *Ship1*. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

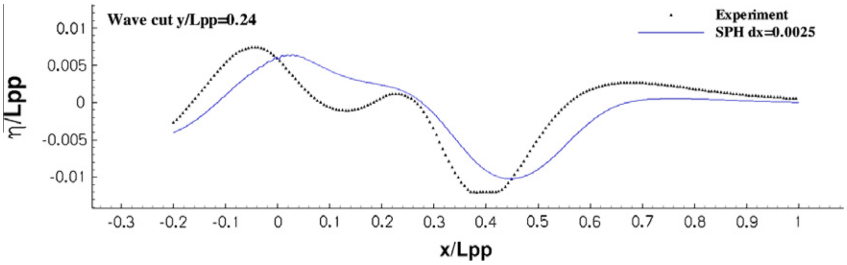

Fig. 17. Longitudinal wave cut a $y/Lpp = 0.24$. The blue plot represents the result for the SPH *Ship1* calculation, the black triangles are extracted from the experimental data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

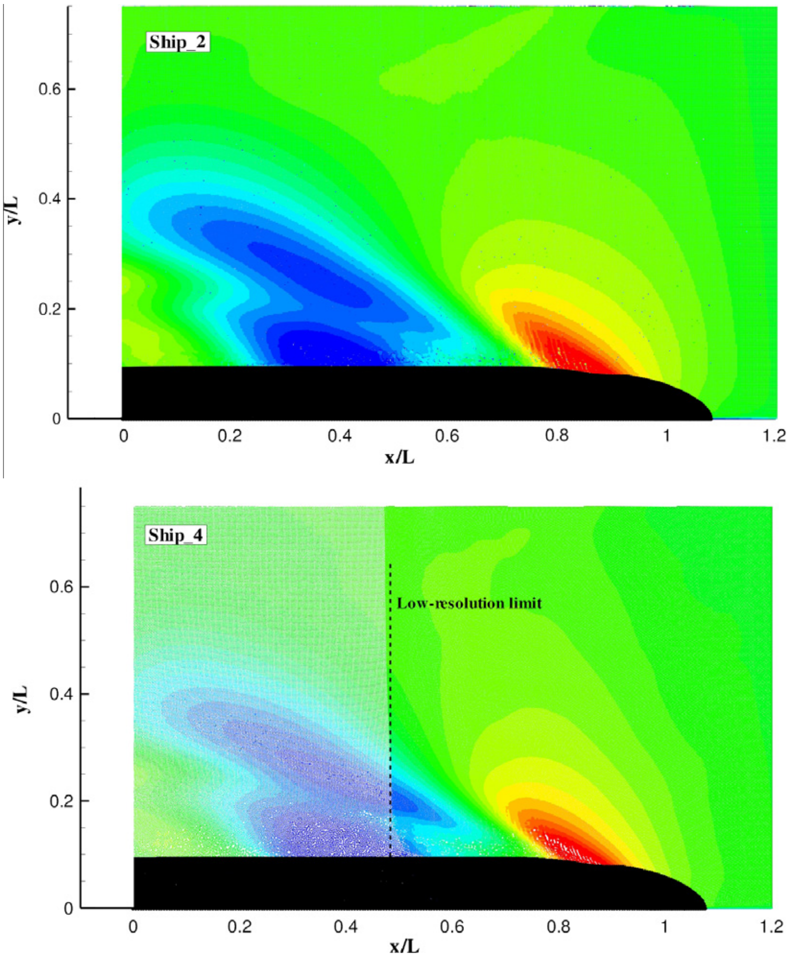

Fig. 18. Wave patterns for the cases *Ship2* (top) and *Ship3* (bottom).

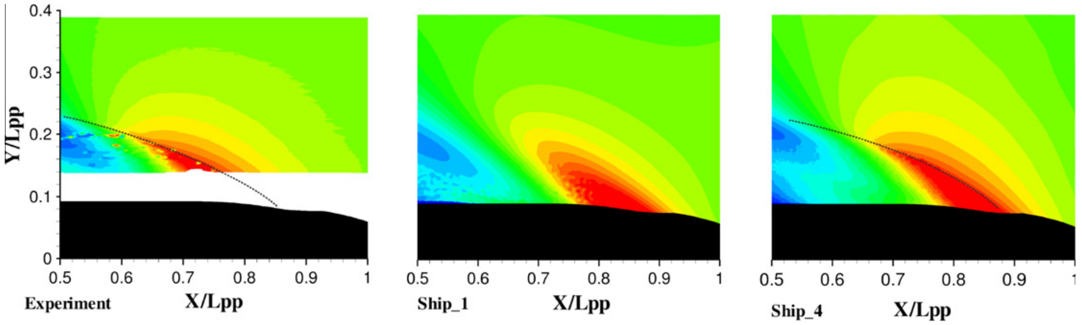

Fig. 19. Contours of the wave elevation for the experiment and SPH simulations *Ship1* and *Ship4*. The dashed line highlights the breaking wave front.

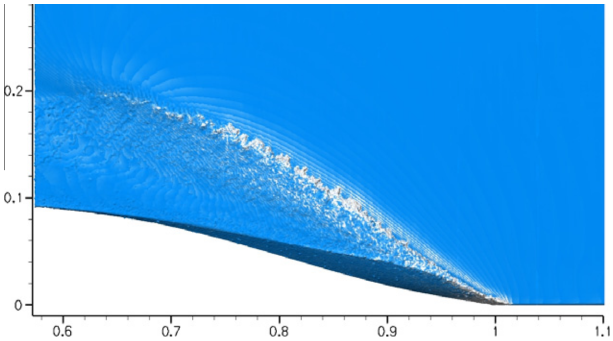

Fig. 20. Top view of the breaking front SPH simulation *Ship4*.

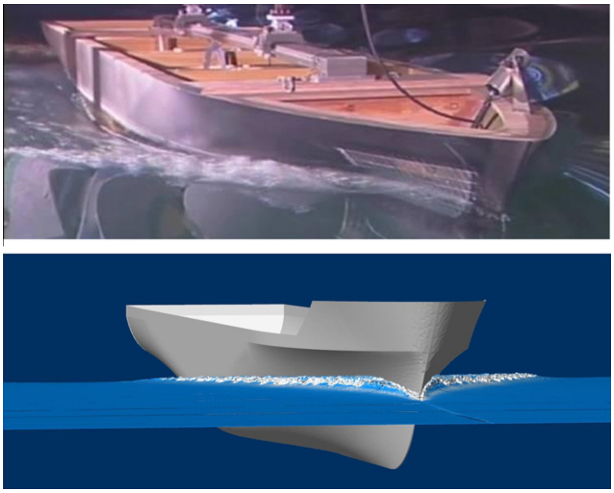

Fig. 21. Breaking bow wave during experiments at DGA-Hydrodynamics (top) and 3D view of the breaking bow wave in the SPH simulation *Ship4* (bottom plot), Fn 0.328.

evaluation of the whole wave pattern through the 2D+t model (see e.g. [34]). Fig. 14 shows a detail of the plunging jet obtained from the 2D+t SPH model described in [31]. The characteristic height of the plunging jet is $H_{pj} = 0.008Lpp$. Since this is very small, it is quite challenging to capture in 3D simulations. As shown in [11], a proper spatial discretization is needed to discretize the plunging jet. In particular the analysis conducted in Section 3 where 3D breaking wave is simulated around a simplified ship bow suggests that the relation (12) has to be verified. This relation in the present case is equivalent to $D = 150dx$, where D is the ship draft.

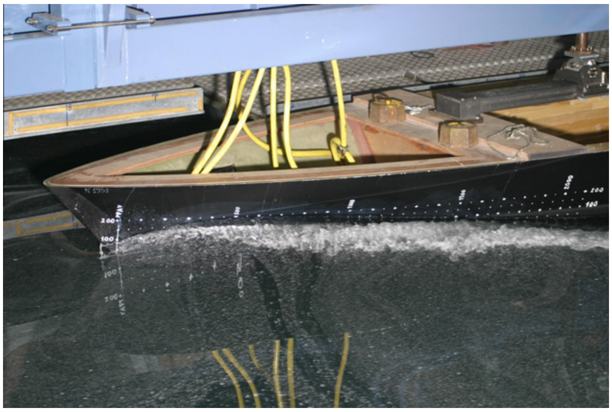

Fig. 22. Details of the breaking bow wave during experiments at DGA-Hydrodynamics, Fn 0.328.

5. Discussion of the results

In this section, the 3D SPH code is validated by comparing the global wave pattern and the breaking bow wave pattern generated by the Alliance vessel with the experimental data from DGA-Hydrodynamics, see [33]. The measurements have been done thanks to a technique involving an ultrasonic probe coupled with optic registration. The error associated with the data has been quantified in $10^{-3}Lpp$. The cartesian coordinate system has its origin fixed at the intersection between the still water level and the first perpendicular of the ship. The x-axis points forward and the z-axis points up.

In Table 3 details of the SPH computations are given. The *Ship1* testcase is conceived to compare the global wave pattern. *Ship2* is performed with the same resolution but adopting a shorter domain in length and depth in order to focus on the features of the breaking bow wave. *Ship3* and *Ship4* are obtained using the same domain adopted in *Ship2* but introducing a multi-resolution technique this time (see Section 5.2). All the adopted domain dimensions have been chosen in order to avoid influences of the bottom and side walls on the wave pattern (see e.g. [35]).

5.1. Analysis of the global wave pattern

The analysis of the global wave pattern is performed on the SPH results of *Ship1* in Table 3.

A numerical reference solution is achieved using a RANS – Level Set code (INSEAN Xnavis code [2]). For the RANS solution two chimera blocks have been used: one with high resolution close to the ship, and one much coarser far from the hull. Differently from the SPH simulation, a no-slip boundary condition has been

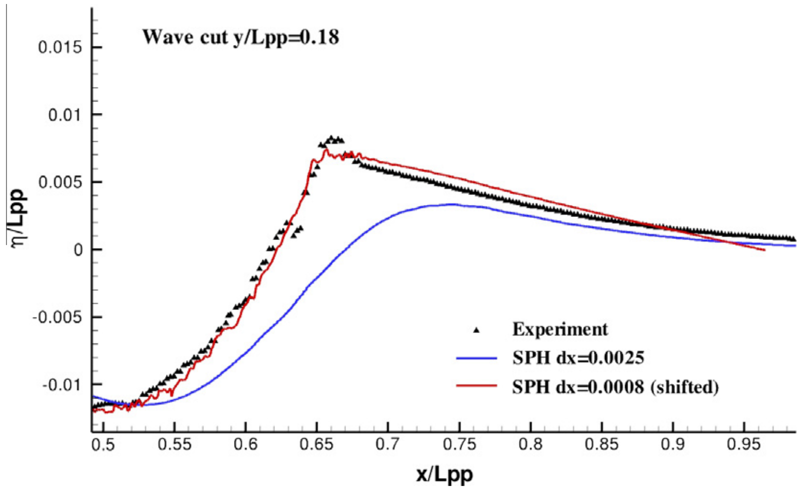

Fig. 23. Longitudinal wave cut a $y/Lpp = 0.18$. The blue plot represents the result for the SPH calculation with coarse resolution (*Ship1*), red plot reports SPH results with fine resolution (*Ship4*), black triangles are extracted from the experimental data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

enforced in the RANS solution and therefore the inner mesh has been stretched to capture the velocity gradient in the the boundary layer region. Just outside the latter region, the grid size is similar to the SPH particle size in *Ship1* (0.0025*Lpp*), while far from the hull the resolution of the second chimera block becomes coarser (0.015*Lpp*).

Top plot of Fig. 15 reports experimental measurements of the wave elevation, the middle plot of Fig. 15 shows the SPH wave contour whereas the bottom panel shows the results obtained by the RANS code.

The SPH wave contour plot is similar to the RANS calculation one. Both the numerical schemes well reproduce the bow wave measured in the experiment even if they seem to underestimate the rear divergent waves. Since experimental data close to the hull are not available, the SPH wave field can be compared only to the RANS simulation. Longitudinal wave cuts extracted at $y/Lpp = 0.12$ are displayed in Fig. 16 for the two numerical solvers. The errors bars relatively to RANS computations (see [36] for more details) are also displayed in this plot. The agreement between the SPH and RANS is satisfactory especially in the fluid region alongside the ship and the measured relative integral error in L_2 norm is 28%.

In Fig. 17 a wave cut farther from the hull ($y/Lpp = 0.24$) is displayed. Here, the SPH solution shows a poor agreement with the available experimental data. The experiments indicate that the breaking bow wave affects the wave pattern close to the bow region as well as the rear divergent wave. This is highlighted in the top of Fig. 15 where a dashed line is drawn to track the region of influence of the breaking bow wave on the rear wave pattern. This region is characterised by a high curvature of the free-surface that, however, is not properly modelled. Indeed, the discretization adopted by both the numerical solvers is not sufficient to capture the breaking wave phenomena and their influence on the wave pattern.

5.2. Focus on the bow wave

To capture the physics of the breaking wave, the resolution has to be increased. Unfortunately, the computational resources available for the present numerical simulations do not allow an increase of the spatial discretization maintaining the fluid domain of *Ship1*. Then, a run with a smaller domain is attempted in *Ship2* (see Table 3). The field is reduced in two directions, that is, the water height and the outflow position.

The bow region is kept almost undisturbed despite the rear wave field being deformed by the outflow, as observed by comparing the results of *Ship2* (top plot of Fig. 18) with the one obtained in *Ship1* (middle plot of Fig. 15). In fact, the wave fields are similar up to $x = 0.5L$. Therefore, the main idea is to capture the breaking bow wave performing a simulation with the same domain extension as in *Ship2* with a spatial resolution as fine as possible.

To further increase the spatial discretization, the multi resolution technique with constant h described in [3] has been used. On a layer of thickness $0.25D$ close to the free surface, the ratio h/dx is set equal to 1.33 (266 neighbours). The thickness is given by the maximum depth reached by vortical structures, as discussed in [31]. The ratio h/dx is reduced to 0.90 (80 neighbours) for the rest of the domain where small gradients of flow field occur. In this way the size of particles in the low-resolution zone is larger by a 1.5 factor (multi-resolution ratio in Table 3). Another simulation (*Ship3*) is performed in order to verify the robustness and the correctness of the solution with this technique. The size of the domain is the same as in *Ship2* as well as the discretization of the low-resolution region. Bottom plot of Fig. 18 shows the wave pattern given by *Ship3* matching the pattern from *Ship2*, demonstrating the validity of the proposed multi-resolution technique.

Finally, *Ship4* is designed with a spatial resolution large enough to capture the breaking of the bow wave. In Fig. 19 the wave fields relative to the experiment, the SPH *Ship1* and *Ship4* testcases are plotted for the region close to the bow. Comparing the shape of the bow wave, it is possible to observe that the shape of the bow wave obtained in *Ship4* is closer to the experimental one respect to *Ship1*. This is underlined also by the dashed line tracking the breaking wave in the experiment that matches well with the wave field in *Ship4*. The extension and the shape of the breaking zone is well reproduced, the numerical breaking front could be better appreciated in Fig. 20, where the top view of the bow wave is depicted, the colour depending on the magnitude of the vorticity.

Further, to better analyse this result, a longitudinal wave cut at $y/Lpp = 0.18$ has been plotted in Fig. 23. The comparison of the shape and height of the bow wave with the experiment displays a clear improvement with respect to the results with a coarser resolution (i.e. *Ship1*). Note that the curve related to *Ship4* has been shifted by a quantity equal to $\Delta x/Lpp = -0.02$ to better appreciate the agreement with the experimental shape. This shift is probably due to the limited extension of the domain that may influence the bow wave.

In the lower plot of Fig. 21 a 3D view of the bow wave simulated in *Ship4* is shown. A qualitative comparison with the top picture of Fig. 21 taken at DGA-Hydrodynamics shows that the main features of the phenomenon are retained. In any case, the resolution is still too coarse to fully capture all the wave breaking features. The SPH seems to reproduce only a typical spilling breaking evolution while Fig. 22 shows that energetic splash-up events occur. This is in accordance with 2D+t simulations carried out in [31] where two splash-up cycles were observed.

6. Conclusions

A 3D SPH solver has been applied to simulate the wave pattern generated by a slender ship in stationary forward motion focusing on the bow breaking wave phenomenon. A simple and efficient new parallel hybrid programming model has been developed for this purpose. The outcomes have been compared with experimental results, as well as RANS level-set computations. A feasibility analysis based on results of a 2D+t study has been conducted to set-up a suitable simulation taking into account the computational costs. Preliminary simulations have been performed to evaluate the minimal resolution needed to catch the bow breaking wave, to verify the capabilities of the solver to simulate wave breaking phenomenon. After the determination of the smallest numerical domain that allows a correct modelling, a detailed validation has been conducted on the global wave pattern. The results show a fair agreement with both the RANS calculations and the experimental measurements. Specific simulations have been performed to better catch the breaking of the bow wave. In any case the numerical resolution is still not fine enough to reproduce all the wave breaking features around the bow of a ship. Future works will be devoted to the development of proper algorithms to manage multi-resolution techniques.

Acknowledgements

This work has been done in the framework of Research Project EUROPA ERG1 RTP N110.067 “Development and Validation of Tools for the Prediction of Hydrodynamics Signatures”, DALIDA, financially supported by the Italian and the French Navies throughout the European Defence Agency. The numerical computations presented here have been performed on the parallel machines of CASPUR Supercomputing Center (Rome) under the Standard HPC Grant 2012 within the project “SPH simulations of bow waves dynamics”, their support is gratefully acknowledged. This work has been also partially supported by European Community’s Seventh Framework Programme (FP7/2007-2013) under Grant Agreement No. 225967 “NextMuSE”. Finally, we would like to thank Dr. Riccardo Broglia for his useful comments and suggestions.

References

[1] Carrica PM, Wilson RV, Stern F. Free surface flows around ships: progress toward simulation of high-speed flows and motions. In: Larreteguy A, editors. Mecanica Computacional XXIV; 2005.

[2] Di Mascio A, Broglia R, Muscari R. On the application of the single-phase level set method to naval hydrodynamic flows. Comput Fluids 2007;36:868–86.

[3] Landrini M, Colagrossi A, Greco M, Tulin MP. Gridless simulations of splashing processes and near-shore bore propagation. J Fluid Mech 2007;591:183–213.

[4] Dalrymple RA, Rogers BD. Numerical modeling of water waves with the SPH method. Coast Eng 2006;53:141–7.

[5] Patel MH, Vignjevic R, Campbell JC. An SPH technique for evaluating the behavior of ships in extreme ocean waves. Int J Maritime Eng 2009;151:39–47.

[6] Marsh A, Khaddaj-Mallat C, Le Touzé D, Oger G. Numerical predictions of ship flooding scenarios using SPH. In: Proc. 5th international SPHERIC workshop, Manchester, 22–25 June, 2010.

[7] May DA, Monaghan JJ. Can a single bubble sink a ship? Am J Phys 2003;71(9):842–9.

[8] Doring M, Oger G, ALessandrini B, Ferrant P. SPH simulations of floating bodies in waves. In: Proc. international workshop on water waves and floating bodies (IWWWFB), Cortona, Italy, 2004.

[9] Cartwright B, Xia J, Cannon S, McGuckin D, Groenenboom P. Motion prediction of ships and yachts by smoothed particle hydrodynamics. In: Proc. 2nd high performance yacht design conference, Auckland, 14–16 February, 2006.

[10] Colagrossi A, Landrini M, Tulin MP. Numerical studies of wave breaking compared to experimental observations. In: Proc. 4th numerical towing tank symposium (NuTTS), Hamburg, 2001.

[11] Marrone S, Colagrossi A, Antuono M, Lugni C, Tulin MP. A 2D+t SPH model to study the breaking wave pattern generated by fast ships. J Fluids Struct 2011;27.

[12] Hernquist L, Katz N. TREESPH: a unification of SPH with the hierarchical tree method. Astrophys J Supp Ser 1989;70.

[13] Nelson RP, Papaloizou JCB. Variable smoothing lengths and energy conservation in SPH. Mon Not R Astronom Soc 1994;270:1–20.

[14] Antuono M, Colagrossi A, Marrone S, Molteni D. Free-surface flows solved by means of SPH schemes with numerical diffusive terms. Comput Phys Commun 2009;181(3):532–49.

[15] Antuono M, Colagrossi A, Marrone S, Lugni C. Propagation of gravity waves through an SPH scheme with numerical diffusive terms. Comput Phys Commun 2011;182(4):866–77.

[16] Marrone S, Antuono M, Colagrossi A, Colicchio G, Le Touzé D, Graziani G. Delta-SPH model for simulating violent impact flows. Comput Methods Appl Mech Eng 2011;200(13–16):1526–42.

[17] Monaghan JJ. Smoothed particle hydrodynamics. Rep Prog Phys 2005;68:1703–59.

[18] Shao S, Gotoh H. Simulating coupled motion of progressive wave and floating curtain wall by SPH-LES model. Coast Eng J 2004;46:171–202.

[19] Violeau D, Issa R. Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview. Int J Numer Methods Fluids 2007;53:277–304.

[20] Colagrossi A, Antuono M, Le Touzé D. Theoretical considerations on the free surface role in the SPH model. Phys Rev E 2009;79/5:056701:1–056701:13.

[21] Di Monaco A, Manenti S, Gallati M, Sibilla S, Agate G, Guandalini R. SPH modeling of solid boundaries through a semi-analytic approach. Eng Appl Comput Fluid Mech 2011;5(1):1–15.

[22] De Leffe M, Le Touzé D, Alessandrini B. Normal flux method at the boundary for SPH. In: Proc. 4th international SPHERIC workshop, Nantes; 2009.

[23] Federico I, Marrone S, Colagrossi A, Aristodemo F, Veltri P. Simulating free-surface channel flows through SPH. Proc. 5th international SPHERIC workshop, Manchester; 2010.

[24] Sbalzarini IF, Walther JH, Bergdorf M, Hieber SE, Kotsalis EM, Koumoutsakos P. PPM a highly efficient parallel particle-mesh library for the simulation of continuum systems. J Comput Phys 2006;215(2):566–88.

[25] Sbalzarini IF, Walther JH, Polasek B, Chatelin P, Bergdorf M, Hieber SE, et al. A software framework for portable parallelization of particle-mesh simulations. Lecture notes in computer science 2006;4128:730–9.

[26] Ferrari A, Dumbser M, Toro EF, Armanini A. A new 3D parallel SPH scheme for free surface flows. Comput Fluids 2009;38(6):1203–17.

[27] Ihmsen M, Akinci N, Becker M, Teschner M. A parallel SPH implementation on multi-core CPUs. Comput Graph Forum 2011;30(1):99–112.

[28] Allen MP, Tildesley DJ. Computer simulation of liquids. Oxford University Press; 1999.

[29] Oger G, Jacquin E, Doring M, Guilcher PM, Dolbeau R, Cabelguen PL, et al. Hybrid CPU-GPU acceleration of the 3D parallel code SPH-Flow. In: Proc. 5th international SPHERIC workshop, Manchester, 22–25 June, 2010.

[30] Tulin MP, Wu M. Divergent bow waves. In: Proceedings of the 21st ONR symposium on naval hydrodynamics, Trondheim, Norway; 1996. p. 99–117.

[31] Marrone S, Antuono M, Colagrossi A. INSEAN technical report, Task N5.IN2. INSEAN, Technical report Dalida project; 2011.

[32] Olivieri A, Pistani F, Wilson R, Campana EF, Stern F. Scars and vortices induced by ship bow and shoulder wave breaking. J Fluids Eng 2007;129:1445–59.

[33] Perelman O. Minutes of towing tank test, Task E6.BE2, DGA-Hydrodynamics. Technical report Dalida project; 2010.

[34] Fontaine E, Faltinsen OM, Cointe R. New insight into the generation of ship bow waves. J Fluid Mech 2000;421:15–38.

[35] Faltinsen OM. Hydrodynamics of high-speed marine vehicles. Cambridge University Press; 2006.

[36] Roache PJ. Quantification of uncertainty in computational fluid dynamics. Annu Rev Fluid Mech 1997;29:123–60.