H-adaptive Navier-Stokes simulations of free-surface flows around moving bodies

Alexander Hay, Alban Leroyer, Michel Visonneau

H-adaptive Navier–Stokes simulations of free-surface flows around moving bodies

Alexander Hay, Alban Leroyer, and Michel Visonneau

Laboratoire de Mécanique des Fluides (CNRS UMR 6598), Equipe de Modélisation Numérique, Ecole Centrale de Nantes, BP 92101, rue de la Noë 44321, Nantes, France

Abstract This article deals with Navier–Stokes simulations of multiphase flows around moving bodies coupled with an adaptive mesh refinement strategy. The numerical framework is considered first: the Navier–Stokes solver, the methodologies for handling multiphase flows and moving bodies, the remeshing techniques, and the adaptive procedure are explained and detailed. Then an application involving hydrodynamic impacts is presented in detail and studied to highlight the relevance of the whole global approach. Of particular interest is the accurate computation of pressure peaks arising during impacts.

Key words Navier–Stokes · Moving bodies · Multiphase flows · H-adaptive methods

1 Introduction

Over the past few years, numerical methodologies have stirred much interest in the engineering community. The use of computational fluid dynamics (CFD) tools is no longer limited to simple physical problems, but can enlarge its field of interest as solvers integrate new physical features to deal with more and more realistic applications. This new trend is illustrated in this article, which presents the coupling between:

1. a finite-volume Navier–Stokes solver for arbitrary unstructured grids;
2. a free-surface capturing approach;
3. a resolution procedure for simulating the motions of freely moving bodies;
4. an automatic local adaptive mesh refinement and coarsening methodology.

Address correspondence to: Michel Visonneau (michel.visonneau@ec-nantes.fr)

Thus, the simulations of complex free-surface flows around real ships can be undertaken with control of the numerical approximation using mesh adaptation. Determining the dynamic response of moving bodies is a large area of concern in ship hydrodynamics: ship maneuvring, behavior in the seaway, slamming phenomena, etc. For all these applications, CFD is complementary to towing-tank tests because it provides a large amount of detailed information on the flow, which helps the designer to improve the performance of new ships.

It was observed during the last Gothenburg 2000 workshop[1] that the free-surface capturing methodology was more and more popular among CFD developers dealing with viscous naval hydrodynamics. This increasing interest is due to the fact that this approach is more robust than those based on a free-surface fitting methodology, since no regridding is necessary and the numerical wave-breaking, which may occur during the initialization period, is perfectly tolerated. When specific compressive discretization schemes are used to solve the concentration transport equation, one can ensure that the density discontinuity between air and water is captured on three to five control volumes.[2,3] However, if the discontinuity occurs in a region where there are not enough grid points, the free-surface elevation is dramatically attenuated, making the free-surface capturing strategy far less accurate (and far more expensive) than classical algorithms based on free-surface fitting. Therefore, the almost perfect numerical strategy should integrate a coupling between a free-surface capturing approach and an automatic local adaptive mesh refinement and coarsening methodology in order to maintain dynamically a prescribed density of grid points around the steady or unsteady interface between air and water.

In this context, the goal of this study was two-fold: first, the emphasis was on demonstrating the robustness and flexibility of the whole numerical approach in order to simulate the motions of moving bodies in multiphase

flows accurately. The focus was also on its ability to capture precisely the free-surface discontinuity for a reduced computational and human effort. While each of the four issues enumerated in the very beginning of this introduction has already been the subject of studies in the literature, such a complete and fully coupled approach has never been developed or studied for hydrodynamic applications. Many surveys of the earlier literature dealing with multiphase flows can be found.[1–7] Similarly, the subject of freely moving bodies in fluid flows has been addressed,[8–10] while an extensive study of h-adaptive methods has also been reported.[11]

This article is organized as follows. Section 2 provides a detailed presentation of the numerical methods. Section 2.1 presents the computational techniques used in order to enable the considerations of hydrodynamic applications. It describes the incompressible flow code which solves the unsteady Reynolds-averaged Navier–Stokes equations (URANS) on unstructured grids with a finite-volume formulation that naturally handles arbitrarily shaped control volumes. Also presented is the free-surface capturing method implemented with the specific discretization schemes which are required to get an accurate description of a density discontinuity.

Section 2.2 describes the numerical techniques that allow the treatment of an arbitrary number of bodies, rigid or flexible, in free or imposed 3D motions. Details are given concerning the resolution of Newton's laws, regridding strategies for the mesh to be adapted to the new position of bodies in time, and fluid–structure coupling.

Section 2.3 describes the local adaptive mesh procedure that is designed in the framework of unstructured grids. The adaptive process is made dynamic and able to consume less CPU time by the use of an adequate data structure. The adaptive procedure is entirely included in the flow solver, making it a complete automatic single tool for computing hydrodynamic flows.

Following these numerical methods, Sect. 3 introduces the physical problem considered in this study. It concerns the two-dimensional simulation of drop tests, which are of relevance in investigating slamming problems and planning boats operating at high speed in a seaway. Sections 4 and 5 consider the detailed results for symmetric and asymmetric cases, respectively. Of particular interest is the computation of pressure peaks that arise during impacts. The results of this general methodology are further compared with the predictions provided by an asymptotic method developed by Scolan et al.[12] and Cointe et al.[13] and with experimental data obtained by Peterson et al.[27] A discussion about the advantages of the global approach, that are demonstrated in terms of the accuracy of the predictions, the optimal use of discretization points, and user-friendliness, concludes this article.

2 Computational approach

2.1 Flow solver

The ISIS solver, developed by EMN (Equipe de Modélisation Numérique, i.e., CFD Department of the Fluid Mechanics Laboratory), uses the incompressible unsteady Reynolds-averaged Navier–Stokes equations (URANS). The solver is based on the finite-volume method to build the spatial discretization of transport equations. The face-based method is generalized to unstructured meshes for which nonoverlapping control volumes are bounded by an arbitrary number of constitutive faces. The velocity field is obtained from the momentum conservation equations, and the pressure field is extracted from the mass conservation constraint, or continuity equation, transformed into a pressure equation. Picard's procedure is used for the linearization of the momentum equations, and this is combined with an iterative process. Each iteration in this process is referred to as a *nonlinear iteration* in the remainder of this study.

In the case of turbulent flows, additional transport equations for the modeled variables are solved in a form similar to the momentum equations, and they can be discretized and solved using the same principles. Several near-wall low-Reynolds-number turbulence models, ranging from the one-equation Spalart–Allmaras model,[14] and two-equation $k-\omega$ closures,[15] to a full stress transport $R_{ij}-\omega$ model,[16] are implemented in the flow solver to take the turbulence phenomena into account.[17]

Incompressible and nonmiscible flow phases are computed using conservation equations for each volume fraction (or concentration) c_i of each phase i. Considering the incompressible flow of a viscous fluid under isothermal conditions, the mass, momentum, and volume fraction conservation equations can be written as follows (using the generalized form of the Gauss theorem):

$$\oint_S \vec{U}\cdot\vec{n}\,\mathrm{d}S = 0 \tag{1a}$$

$$\frac{\delta}{\delta t}\int_V \rho\vec{U}\mathrm{d}V + \oint_S \rho\vec{U}\left(\vec{U}-\vec{U_d}\right)\cdot\vec{n}\,\mathrm{d}S = \int_V\left(\rho\vec{g}-\vec{\nabla}P\right)\mathrm{d}V + \oint_S\left(\mathrm{T}+\mathrm{T}_t\right)\cdot\vec{n}\,\mathrm{d}S \tag{1b}$$

$$\frac{\delta}{\delta t}\int_V c_i\mathrm{d}V + \oint_S c_i\left(\vec{U}-\vec{U_d}\right)\cdot\vec{n}\,\mathrm{d}S = 0 \tag{1c}$$

where V is the domain of interest, or the control volume, bounded by the closed surface S moving at the velocity $\vec{U}_d$ with a unit normal vector $\vec{n}$ directed outward, $\vec{U}$ and P represent the velocity and pressure,

respectively, $\mathbb{T}$ and $\mathbb{T}_t$ refer to the viscous and Reynolds stress tensors, respectively, and $\vec{g}$ is the gravity vector. The time derivative following the moving grid is written $\delta/\delta t$. While $\mathbb{T}_t$ is determined according to the turbulence model used, $\mathbb{T}$ follows the classical relation of Newtonian fluids for incompressible flows. The effective physical flow properties (viscosity μ and density ρ) are obtained from each phase property (μ_i and ρ_i) with the following constitutive relations:

$$\rho = \sum_i c_i \rho_i; \qquad \mu = \sum_i c_i \mu_i; \qquad \sum_i c_i = 1 \tag{2}$$

When the grid is moving, the so-called *space conservation law* must also be satisfied:

$$\frac{\delta}{\delta t}\int_V \mathrm{d}V - \oint_S \vec{U}_d \cdot \vec{n}\,\mathrm{d}S = 0 \tag{3}$$

Except for the convection terms and volumetric mass fluxes, interfacial quantities q_f are rebuilt linearly from the quantities themselves and their available cell-centered gradients. Special attention has to be paid to face reconstructions of the volume fraction c_i. The challenge posed by the discretization of a transport equation for the concentration is the accurate modeling of a contact discontinuity, i.e., the free-surface. In order to assume face-bounded reconstructions and to avoid unrealistic oscillations, the search for an acceptable compromise between the accuracy and boundedness of $c_i \in [0, 1]$ is a key point.[18,19] Moreover, the method must also preserve the sharpness of the interface through the transport equation (Eq. 1c).

These requirements are fullfilled by the inter-gamma differencing scheme (IGDS).[20] This is based on the GD scheme,[18] but it introduces downwind differencing since compressive characteristics are required to capture the interface accurately. Through a normalized variable diagram (NVD) analysis,[21] this scheme enforces local monotonicity and the convection boundedness criterium (CBC).[22] The main disadvantage of the IGDS scheme is a Courant number limitation: Co < 0.3 in multidimensional cases, known as the Courant–Friedrich–Levy (CFL) condition. The Courant number of any face is defined as

$$\mathrm{Co} = \Delta t \mathcal{F}/V \tag{4}$$

where $\mathcal{F}$ is the total (positive) velocity flux through the face considered, V is the volume of the upwind cell, and Δt is the global time-step of the temporal discretization. Therefore, the discretized time-step needs to be sufficiently small by the CFL condition. It has been shown that the role played by the compressive property of the IGDS is fundamental to get a reliable simulation of the free surface.[3] Interested readers can find more details on the IGDS in the literature.[3,23]

2.2 Moving bodies

An arbitrary number of rigid or flexible bodies can be considered, along with the flow around them, by the ISIS solver. Furthermore, their motions can be either imposed or solved.[9]

Considering the applications involved in this study, this section focuses mainly on the particular point of freely moving rigid bodies that raises three different issues. First and foremost, Newton's laws have to be solved. Then the computational grid must be moved in accordance with the displacement of the bodies, and last special attention has to be paid to flow-motion coupling for the global procedure to be stable. These topics are addressed in the following subsections.

2.2.1 Resolution of Newton's laws

The aim here is to calculate the temporal evolution of the kinematic characteristics of a rigid body submitting to forces acting on it (hydrodynamic forces, gravity, etc).

The first step consists in the definition of an inertial frame of reference $\mathcal{R}_0 = (O_0; \beta_0)$, linked to the physical space, and assimilated to a Galilean referential (β_0 is the basis associated with $\mathcal{R}_0$). The origin O_0 of this referential is fixed to the mass center of the body G at the initial time. In the case of a flexible body, the desired shape is then imposed on $\mathcal{R}_0$. A rigid motion (that changes $\mathcal{R}_0$ to a new referential $\mathcal{R}_1$) is then carried out to place the body in space (Fig. 1). Thus, the characteristics of the body motion are given by the transformation $\mathcal{R}_0 \rightarrow \mathcal{R}_1$ (position and orientation).

In the general framework of a three-dimensional resolution, a classical description of the body's orientation by three successive rotations (like Euler angles), written (ψ, θ, ϕ), is unsuitable because of singular configurations for which the triplet (ψ, θ, ϕ) is not unique. The use of a quaternion eliminates these problems. This technique has already been used by McDonald and Whitfield.[24] Quaternion space $\mathbb{H}$ is a four-dimensional algebra spanned by a real axis **e** and three orthogonal imaginary axis denoted by **i**, **j**, **k**. It can

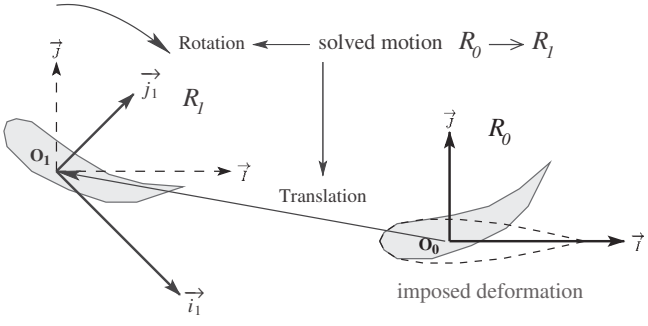

Fig. 1. Parameterization of the motion

be roughly considered as a spatial extension of the plane representation of complex numbers. Each rotation around the axis directed by the unit vector $\vec{u}$ (its corresponding quaternion is denoted by $\mathbf{u}$) with angle θ is associated with only one quaternion $\mathbf{q} = \cos\left(\frac{\theta}{2}\right)\mathbf{e} + \mathbf{u}\sin\left(\frac{\theta}{2}\right)$ on the $\mathbb{H}$ basis ($\mathbf{e}$, $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$). This quaternion (and its time-derivative) can be related to the instantaneous rotation vector $\vec{\Omega}$. This relation makes it possible to calculate $\mathbf{q}$ by knowing $\vec{\Omega}$ (for further information, see Leroyer[9]). Thus, the solved system reduces here to

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{A1}{M_S} & \frac{-F1}{M_S} & \frac{-E1}{M_S} \\ 0 & 0 & 0 & \frac{-F1}{M_S} & \frac{B1}{M_S} & \frac{-D1}{M_S} \\ 0 & 0 & 0 & \frac{-E1}{M_S} & \frac{-D1}{M_S} & \frac{C1}{M_S} \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \alpha \\ \beta \\ \gamma \end{bmatrix}$$

$$= \frac{1}{M_S} \begin{bmatrix} \vec{R} \cdot \vec{x_0} \\ \vec{R} \cdot \vec{y_0} \\ \vec{R} \cdot \vec{z_0} \\ \overrightarrow{\mathcal{M}_{O_1}} \cdot \vec{x_0} - \vec{S_I} \cdot \vec{x_0} \\ \overrightarrow{\mathcal{M}_{O_1}} \cdot \vec{y_0} - \vec{S_I} \cdot \vec{y_0} \\ \overrightarrow{\mathcal{M}_{O_1}} \cdot \vec{z_0} - \vec{S_I} \cdot \vec{z_0} \end{bmatrix} \tag{5}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \tag{6}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\alpha & -\beta & -\gamma \\ \alpha & 0 & -\gamma & \beta \\ \beta & \gamma & 0 & -\alpha \\ \gamma & -\beta & \alpha & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{7}$$

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \tag{8}$$

with the following notation:

— M_S, mass of the body;
— ($A1$, $B1$, $C1$, $D1$, $E1$, $F1$), elements of the inertia matrix of the body (see Eq. 5);
— (x, y, z), coordinates of the vector $\overrightarrow{O_0O_1} = \vec{X}$ on the basis β_0;
— $(\dot{x}, \dot{y}, \dot{z})$, coordinates of the vector $\overrightarrow{V\left(O_1/\mathcal{R}_0\right)}$ on the basis β_0;
— (α, β, γ) coordinates of the vector $\overrightarrow{\Omega_0^1}$ on the basis β_0;
— (q_0, q_1, q_2, q_3) coordinates of $\mathbf{q}$ on the basis β_0;
— $\{\vec{R}, \overrightarrow{\mathcal{M}_{O_1}}\}$, screw of external forces acting on the body;
— $\vec{S_I}$ inertial source term.

The expression of $\vec{S_I}$ is

$$\begin{aligned} \vec{S_I} &= \int_{\text{body}} \rho\left(\overrightarrow{O_1M} \cdot \overrightarrow{\Omega_0^1}\right)\left(\overrightarrow{O_1M} \times \overrightarrow{\Omega_0^1}\right) \mathrm{d}v \\ \vec{S_I} \cdot \vec{x_0} &= \alpha\gamma F1 - \alpha\beta E1 + \left(\gamma^2 - \beta^2\right)D1 + \beta\gamma\left(C1 - B1\right) \\ \vec{S_I} \cdot \vec{y_0} &= \alpha\beta D1 - \beta\gamma F1 + \left(\alpha^2 - \gamma^2\right)E1 + \alpha\gamma\left(A1 - C1\right) \\ \vec{S_I} \cdot \vec{z_0} &= \beta\gamma E1 - \alpha\gamma D1 + \left(\beta^2 - \alpha^2\right)F1 + \alpha\beta\left(B1 - A1\right) \end{aligned} \tag{9}$$

For two-dimensional simulations, this approach with quaternions is unnecessary since the resolution of the rotation reduces to $C1\ \ddot{\gamma} = \mathcal{M}_{O_1}$, which can be integrated twice without any difficulty.

2.2.2 Regridding strategies

To implement the motion of bodies in a flow solver, the mesh must be adapted in time to the new position of the bodies. In order to keep an appropriate grid, three complementary methods have been integrated:

— the spring analogy regridding procedure;
— rigid transformation of the mesh;
— the analytical weighted regridding approach.

These different regridding strategies can be mixed, which provides a useful flexibility when one wants to combine deformation and rigid body motion modes.

In the first method, the mesh is represented by a mechanical structure composed by fictitious lineal and torsional springs.[25] The new mesh is obtained by resolving a pseudostructural system after imposing the new positions of nodes belonging to bodies and other boundaries. This technique is very convenient because of its capacity to deal with any deformation. However, the computational cost is quite high, since it requires the resolution of three additional elliptic equations.

With the rigid transformation, all the nodes are moved according to the solid motion of the body from the resolution of Newton's laws or from an imposed law. This is not strictly speaking a regridding approach, since there is no deformation. The grid simply follows the body in physical space. This approach can only be used for simulations with one body in an infinite fluid domain, but tolerates movement of arbitrary amplitudes.

The last one approach, named analytical weighed regridding, is a type of extrapolation of rigid transformation, but in this case, the displacement of each node is weighted by a factor k_w varying between one and zero according to its distance from the body (1 for nodes belonging to the body and 0 for nodes on other boundaries, (Fig. 2)). The values of k_w are calculated at the beginning of the simulation by solving a Laplacian operator, and in the case of multibody simulations, a weighting factor is computed for each body.

As far as rigid bodies are concerned, the weighting is directly applied to the parameters of the rigid transformation $\mathcal{R}_0 \rightarrow \mathcal{R}_1$. Thus, this regridding strategy is far less CPU time-consuming than the spring analogy method, and can favorably replace it in the case of peculiar motions such as rigid ones. An example is given in Fig. 3 for a square body with a 60° rotation. Regardless of the fineness of the mesh, the technique does not raise any problems for near-body cells with a high aspect ratio because the weighting factor in the vicinity of the body is almost 1. The orthogonality is therefore well conserved. An extension of this technique has also been developed for beam-like bodies (for further information, see Leroyer[9]).

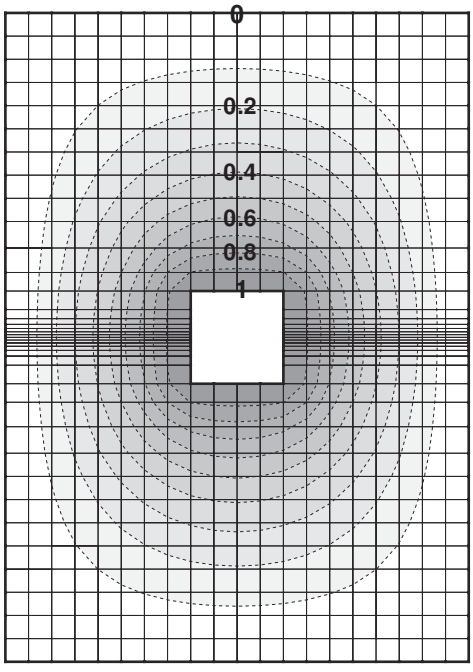

Fig. 2. Isolines of the weighting factor k_w

In the flow solver, mesh mobility is taken into account by calculating the grid displacement velocity flux on each face. For deformation techniques (spring analogy and weighted regridding), this is obtained by computing the exact volumes swept by cell faces, which ensures that the space conservation law is automatically satisfied.[26] If a rigid transformation also has to be used, its contribution to the displacement velocity fluxes is computed directly by using the parameters of the rigid transformation $\mathcal{R}_0 \rightarrow \mathcal{R}_1$.[9]

2.2.3 Flow-motion coupling

In the case of prescribed motions, the bodies are first displaced at every time step, then the mesh is rebuilt, and lastly the flow is solved. Therefore, there is no real coupling when motion is imposed owing to the lack of fluid feedback on the body's position. The problem is more complex when motion is solved with Newton's laws: then the body's kinematics are linked to the flow at the same time-step by forces acting on them. Reciprocally, the body's motion modifies the flow through the displacement velocity fluxes, as illustrated in Fig. 4.

If the hydrodynamic forces and moments provided by the solution of the RANS equations are only calculated at the end of every time-step to compute the new positions of bodies, instabilities appear if the body's density is close to the density of the surrounding fluid. A stable procedure is obtained as follows: at the end of each single nonlinear iteration (see Sect. 2.1), the estimated flow is used to calculate the forces and moments acting

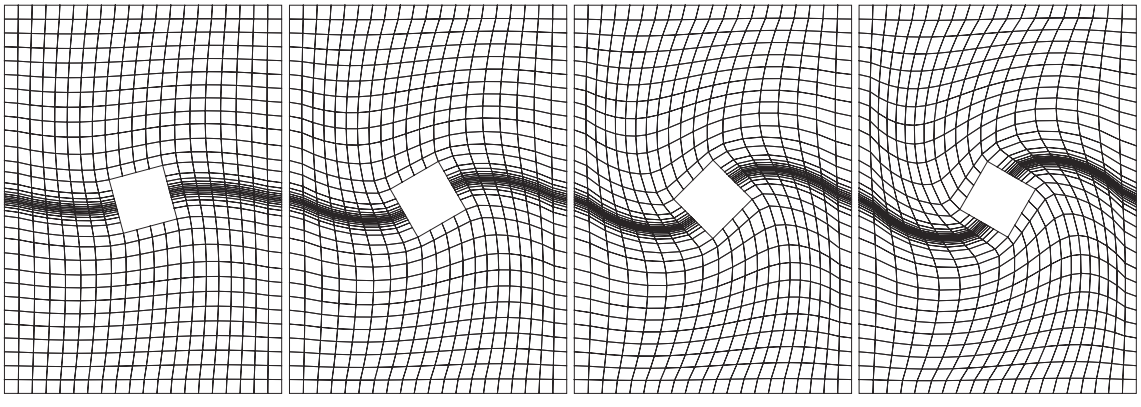

Fig. 3. Example of the weighted regridding approach. **a** 15°. **b** 30°. **c** 45°. **d** 60°

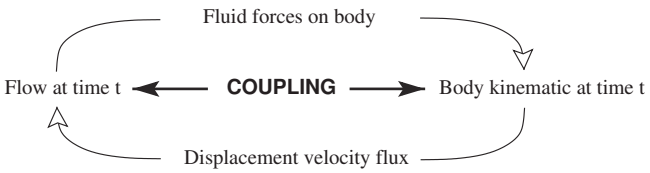

Fig. 4. Diagram of the fluid–structure interaction

on the bodies and then to compute the new positions of the bodies (see Fig. 8). Thus, both the flow equations and Newton's laws are solved during each iteration of Picard's procedure (called nonlinear iteration). Consequently, the resolution of the flow and body's kinematics are closely linked. This ensures a stronger flow/motion coupling that is responsible for the stability of this procedure. However, such an approach means that it is necessary to calculate the forces and to solve Newton's laws more often, but the associated CPU time is almost negligible compared to the computation time for the flow. Similarly, the mesh has to be adjusted to the new computed positions of the bodies at each iteration of Picard's procedure, that is to say very often. As explained before, the analytical regridding technique is straighforward and takes very little CPU time, making it very appealing compared with the spring analogy procedure in such a context.

2.3 Local mesh adaptation

As indicated above, the free-surface capturing methodology is based on the solution of a transport equation of a contact discontinuity that indicates the location of the interface. Although the compressive discretization scheme IGDS guarantees that the interface is captured over three to five cells, it is essential to keep the characteristic length of the cells on either side of the interface as small as possible in order to avoid too much numerical smearing of the discontinuity. Clearly, an automatic local adaptive mesh procedure (h-refinement), based on successive refinement and unrefinement steps, is well suited to following the temporal evolution of the interface and to maintaining a fine computational grid around it. Moreover, an automatic grid adaptation frees the user from a tedious task: the generation of a mesh suited to all parts of the unknown interface, since one can start from a uniform grid which will be automatically refined near the interface thanks to an explicit interface indicator.

To be efficient, the local adaptive procedure has to fulfill some requirements that can be figured out a priori. First and foremost, the methodology must be able to handle unstructured grids, since such topologies are now well known to facilitate (and sometimes to make possible) the treatment of real hydrodynamic

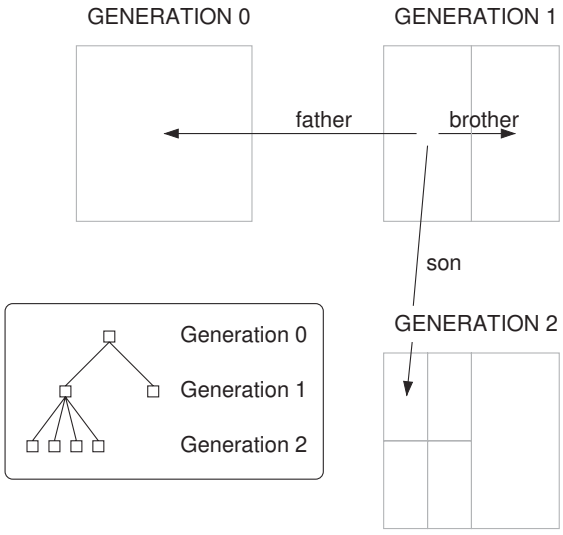

Fig. 5. Relationship between the different generations of elements

problems on complex geometries. Second, when unsteady flows are considered, the free surface may be submitted to rapid variations in time, so that mesh adaptation may be required very often during the numerical simulation for the current grid to fit the discontinuity. Such a constraint clearly points out the need for local mesh adaptation rather than adaptive mesh generation where each adaptation step corresponds to an automatic global mesh generation.[11] Moreover, in order to keep the CPU time for the local grid adaptation process to a minimum, the adaptive procedure should be based on a data structure which allows dynamic grid alterations. Lastly, a mapping routine has to be part of the whole adaptation step in order to interpolate the solution computed on the last grid considered to the newly adapted one in order for the computation to be continuous.

The remainder of this section is devoted to the presentation of the adaptive techniques developed and their main characteristics.

2.3.1 Data structure and grid alteration

For the mesh adaptation process to be flexible, quick, and easy to implement, a suitable data structure is required. Thus, the present local mesh adaptation procedure is based on the notion of a relationship between the successive generations of elements of grids. Connectivities of *relationship* are considered for both the *control volume* type of element and the *face* type of element. These are presented here only for the control volumes, but similar notions apply for the faces of the control volumes. The relationships introduced between the cells of the meshes lead to a natural vocabulary of *family*, *father*, *son*, and *brother*, as illustrated in Fig. 5

(either for two-dimensional control volumes or three-dimensional faces). The initial mesh consists of the elements of generation 0, and a first refinement step leads to the creation of elements of generation 1. A refined control volume becomes a father, and is split into several sons that share a brotherly relationship. The father and sons form a family. A further refinement step will induce elements of generation 2, and so on. It should be emphasized that this structure does not reduce the generality of the grid alteration since it does not take into account how the elements are refined. In addition, negative generations can exist due to the unrefinement by agglomeration that coarsens the initial grids.[11] As an example, the cell (father) resulting from the agglomeration of cells (sons) of the initial grids (generation 0) belongs to the generation –1. It should also be pointed out that all the elements of the successive generations are kept in the memory (they are only destroyed by unrefinement), but the cost of additional memory is reduced since only a local adaptation is performed.

Such a data structure addresses several problems at once. First, the unrefinement of a family (thus of a previously refined cell) becomes straightforward since it corresponds to the recovery of the father (and also the destruction of the sons). Thus, the refinement/unrefinement process becomes very dynamic and is quick to achieve. Secondly, it permits the easy and exact recovery of the initial mesh if refinement is no longer necessary in an area of the computational grid, as is the case for an unsteady problem.

During the refinement process, each control volume to be refined is split into several smaller ones of the same topology. Thus, the topology of an element on any part of the mesh will always be the same, and only its size will be adapted. In this way, the initial desired local mesh quality is preserved everywhere throughout the simulation. For two-dimensional grids, as illustrated in Fig. 6, the refinement process can occur with a possible directional sensitivity for flows with simple features.

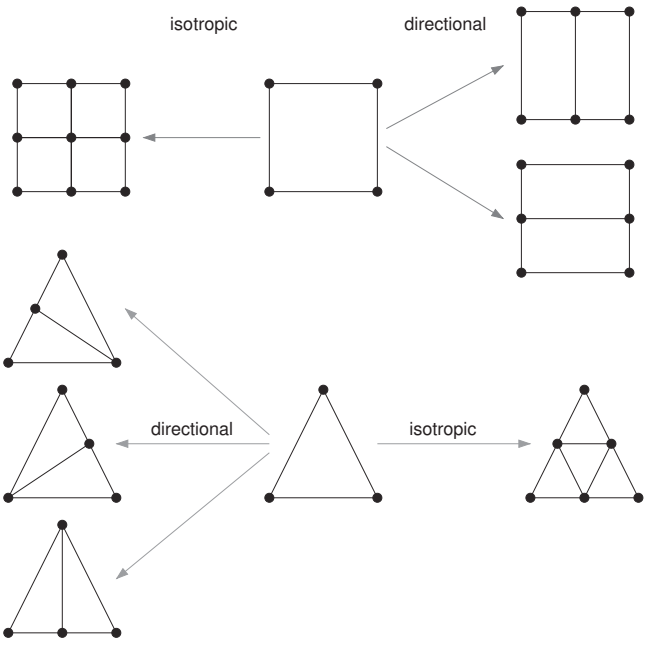

Fig. 6. Refinement of two-dimensional volumes

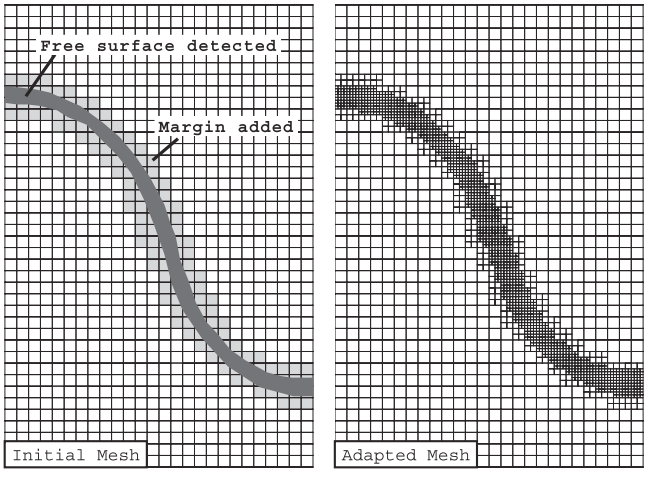

Fig. 7. Safety margin around the interface

2.3.2 Description of the procedure

The goal of the adaptive procedure presented is to maintain a desired prescribed cell size around the free surface. In the framework of capturing methodologies, an explicit indicator is required in order to locate the free surface in space, and this is easily derived using information from the volume fraction. The indicator considered is proportional to the concentration c_i, which has already been computed during the calculation of the flow. This permits the identification of cells for which the volume fraction is between 0.01 and 0.99, and which are thus filled with a mixture of the two fluids. However, a safety margin is also added around the detected zones, as indicated in Fig. 7, to guarantee that the free surface is inside the finest mesh level. Its thickness is improved by the number of layers of neighboring cells.

The adaptive procedure for an unsteady computation is summarized in Fig. 8. An adaptation step is required every N time-steps during the temporal loop. This parameter has to be sufficiently small so that the adapted meshes can follow the temporal evolution of the interface correctly. Thus, it is related to both the Courant number and the safety margin selected. Actually, N is always fixed to a small value so that the need for adaptation is checked very often. However, if no significant adaptation is required, the computation is resumed. Otherwise, the current mesh is modified ac-

cording to the explicit indicator, and a mapping procedure is introduced in order to interpolate the solutions computed on the previous grid onto the newly adapted mesh. The mapping operator is designed in such a way that it preserves the conservative property of any function to be interpolated, as required by the finite-volume methodology.[11] However, the current interpolated solution will no longer satisfy the discrete unsteady operator. Thus, it is necessary to achieve a new convergence of the solution at the current time with the previously computed positions of the bodies, as shown in Fig. 8. The additional CPU time due to these *added* steps in the temporal loop is part of the overall computational cost of any adaptive calculation.

The typical cell size in the vicinity of the free surface is controlled by the parameter NGen. This is the maximum number of generations which can be created. Thus, the local characteristic length h of any part of the initial mesh can, at most, be reduced to $h/2^{NGen}$ in each direction. The higher the value of NGen, the more accurate the free-surface capturing is expected to be.

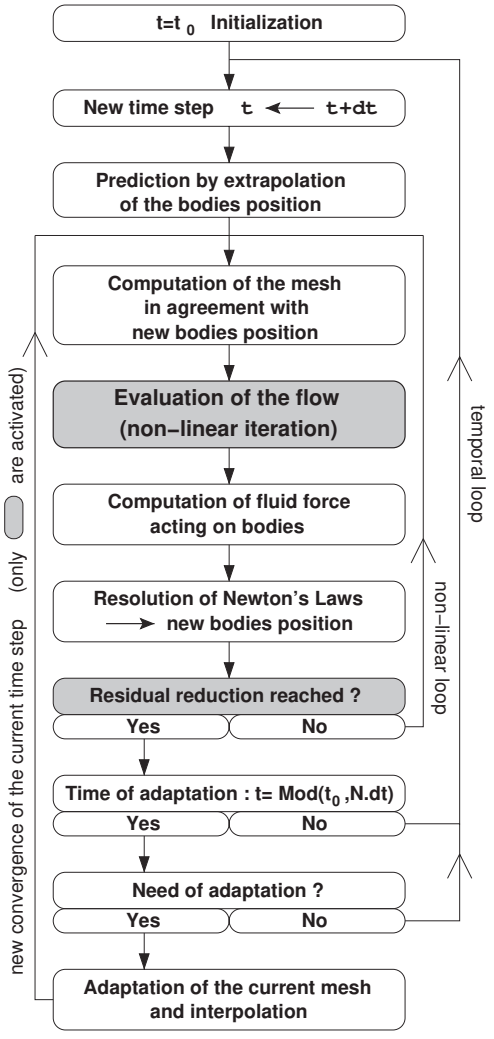

Fig. 8. Adaptive unsteady procedure

3 Physical problem

3.1 Experimental configuration

The emphasis of this study is on drop tests for which experimental data and theoretical results exist. The unsteady flows around a prismatic hull falling through the air and impacting the free surface of water are considered. Symmetric and asymmetric drop tests have been investigated experimentally by Peterson et al.[27] and theoretically by Xu et al.[28] The main interest in such a physical problem comes from the need to understand the dynamic response of high-speed ships during asymmetric water impacts in order to reduce the hull damage resulting from boat slamming.[29]

The physical model used is a high-aspect-ratio prismatic wedge with a 20° deadrise, as illustrated in Fig. 9. This is dropped from different heights H, with different weights M_S, and with initial zero (symmetric cases) or nonzero (asymmetric cases) heel angles α. For the asymmetric tests, the position of the center of gravity and the inertia are also parameters, and instruments with accelerometers recorded the roll and the vertical acceleration time-history. The different configurations considered are summarized in Table 1.

3.2 Numerical framework

The basic grid for all the computations performed in this study is illustrated in Fig. 10 (for the symmetric

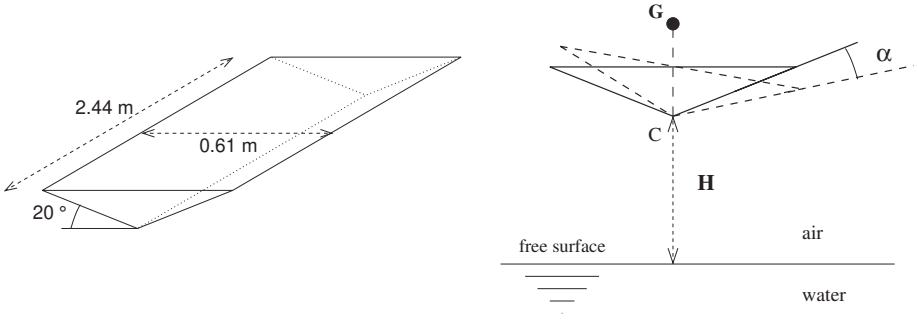

Fig. 9. Experimental configuration of drop tests. **a** Geometry of the prismatic hull. **b** Experimental parameters

Table 1. Parameters of the configurations studied

Name	Sym-light	Sym-medium	Asym-light	Asym-medium
H (m)	0.61	0.61	0.61	0.61
α	0°	0°	5°	5°
M_S (kg)	122	291	124	293
CG (m)	—	—	0.216	0.165
I_G (kg m^2)	—	—	8.85	10.95

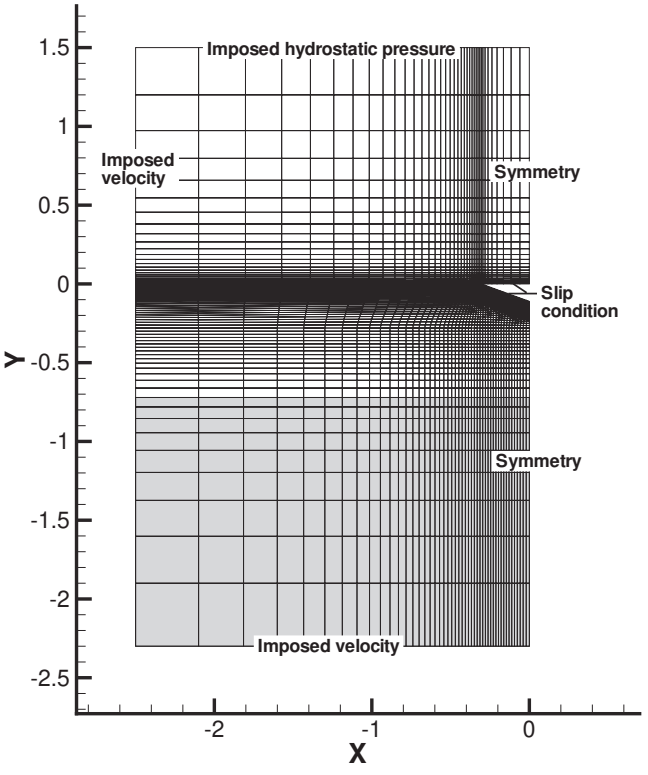

Fig. 10. Initial configuration

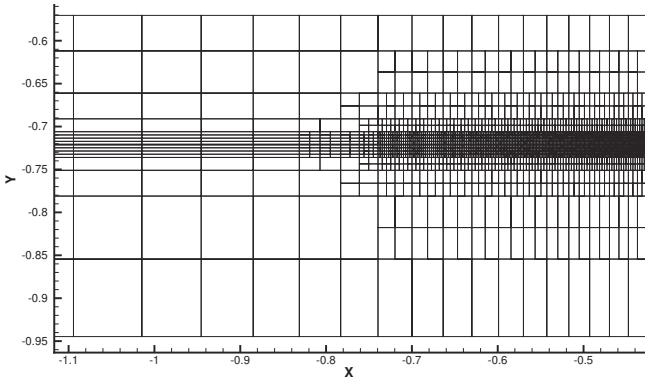

Fig. 11. Isotropic → directional transition

problems). It is a coarse grid, with only 3400 control volumes. For the asymmetric cases, the computational mesh is simply reproduced for positive values of the X-axis (and the heel angle is modified), and thus it has twice as many cells.

The simulations considered involve the motion of a rigid body in an infinite fluid domain. Thus, rigid transformations of the meshes can be used to address translations. This is combined with the analytical weighted regridding approach when rotations have to be considered (asymmetric cases). As can be seen in Fig. 10, the basic grid is initially very coarse at the free-surface location, but as the grid falls vertically, the mesh is finer and finer in the area of the interface (whatever the set value of NGen) until the wedge goes through the free surface (the clustering of point reaches a maximum at $y = 0$). This change in mesh size in the vicinity of the free surface during the computation does not pose any numerical problems. Indeed, the compressive property of the IGDS scheme (see Sect. 2.1) ensures that the finer the mesh, the smaller the free-surface thickness.

Figure 10 also shows the typical boundary conditions. Far from the body, the disturbances of the fluid are assumed to be minimal. Thus, the fluid velocity is presumed to be zero except on the upper boundary, where it is possible to fix the pressure to its known hydrostatic value (which is not constant, since the mesh falls down along with the prismatic hull). The slip wall condition also holds on the boundaries corresponding to the body, since the viscous friction there can easily be neglected in such a problem. In the figure, the shaded area is filled with water. The initial free-surface position is $H + T = 0.721011$ m, since the upper part of the hull is located at $Y = 0$, where T is the thickness of the hull ($T = \frac{0.61}{2}\tan 20°$, see Fig. 9a) and H is the dropping height (see Table 1).

Starting from the basic grid previously presented, adaptive computations are performed for all the configurations using the different numbers of maximum generation size allowed (NGen). As shown in Fig. 11, both isotropic and directional refinement are considered. Indeed, accurate free-surface capture requires a fine spatial discretization in its normal direction only. Away from the body, the free surface is barely deformed, and thus is aligned with the mesh lines. Therefore, directional refinement is very suitable, bearing in

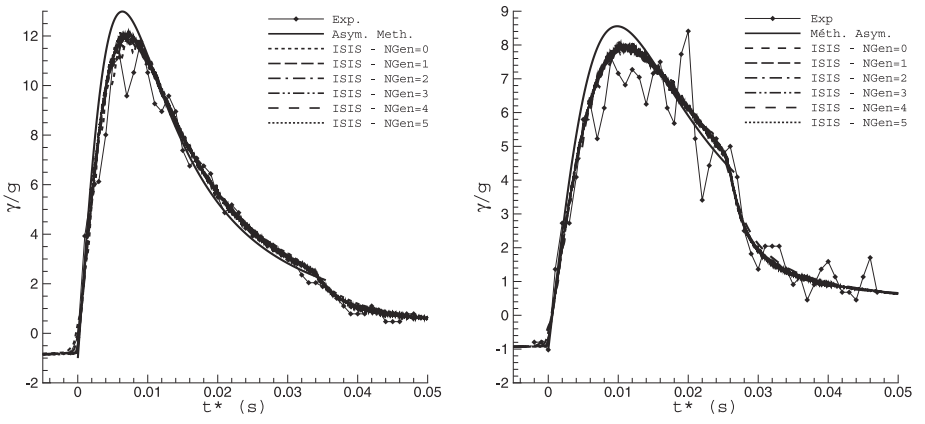

Fig. 12. Adimensional vertical acceleration. **a** Sym-light. **b** Sym-medium

mind that its computational cost is less than that of isotropic refinement. However, isotropic refinement is required closer to the body where the interface is strongly deformed. The limit is located at $|X| < 0.75\,\mathrm{m}$ where both isotropic and directional refinement are performed alternately in order to ensure a smooth transition.

For all the computations performed, the time-step Δt linked to the second-order accurate time discretization scheme follows an adaptive law. Thus, the adaptation is performed in both space and time. In the framework of free-surface capturing methodology, it has been observed that the CFL condition applied in time-steps (see Sect. 2.1) is very restrictive, and thus leads to small values of Δt.[2,3] Therefore, no significant improvement is achieved over temporal discretization if one considers even smaller time-steps. In the present study, the corresponding time-step law is derived in such a way that the Courant number is always slightly under the critical value of 0.3 in the vicinity of the free surface using Eq. 4. Then the IGDS scheme for the transport equation of concentration is in the best configuration in terms of its capturing capacities since the CFL condition is fully respected at any time in the simulation and whatever the mesh size. Furthermore, the user is no longer asked to give an adequate time-step law, which is particularly interesting when considering adaptive computations for obvious reasons. This particular point is more thoroughly addressed in Sect. 4.4.

It should also be noticed that during each temporal loop, the nonlinear loop (see Fig. 8) ends when the residual of each single equation is reduced by two orders of magnitude, which ensures sufficiently converged solutions when using small time-steps such as those considered here.

4 Symmetric cases and numerical study

4.1 Dynamics of the impacts

The emphasis is first on the dynamic response of the two symmetric drop tests. Figure 12 shows the time history of the vertical upward acceleration of the prismatic hull, in units of the gravitational constant ($g = 9.81\,\mathrm{ms}^{-2}$), during symmetric impact under light (sym-light) and medium (sym-medium) weight conditions. The time t^* considered is zero when the impact occurs, i.e., when the lowest point of the hull (keel) touches the water surface (it takes only 0.356 s for the wedge to reach the water). The results from all the simulations performed using a growing NGen number from 0 (no adaptation) to 5 were compared with the corresponding experimental data from the CSS and with the predictions from an asymptotic method developed by Scolan et al.[12] and Cointe et al.[13] and based on Wagner's theory. The body forces applied on the hull can then easily be deduced.

All the simulations performed are in fairly good agreement with the experimental results, showing that the motions of moving bodies in multifluid flows are correctly predicted by the procedure previously described. However, looking at these results, the h-adaptive strategy is relatively useless since the dynamic responses computed show no significant improvement with an increase in NGen. In fact, the outcomes of the single-mesh computations (NGen = 0) are already satisfactory, which means that the global dynamic forces exerted on the body are easy to compute and cannot be used to assess the accuracy of a discretization method applied to impact flows.

4.2 Free surface

This section deals with the accuracy of the free surface computed by the numerical simulations. First, it should be noted that the specific discretization scheme used to

Fig. 13. Influence of NGen on the free-surface definition. **a** NGen = 1. **b** NGen = 3. **c** NGen = 5

Table 2. Exact error in the position of the computed interface ($c_i = 0.5$) at $t = 0.3$ s

NGen	z_{ci} (m)	$Err = \lvert z_{ci} - z_p \rvert$ (m)	Relative error Err/d_v (%)
0	–0.7171	0.003911	0.9
1	–0.72015	0.000861	0.2
2	–0.72075	0.000261	0.06
3	–0.72085	0.000161	0.037
4	–0.72090	0.000111	0.025
5	–0.72098	0.000031	0.007

solve the volume fraction equation (see Sect. 2.1) always successfully captures the free surface over typically three to five cells. As the prismatic body falls, the free surface gets closer and closer to it and thus the size of the control volumes becomes smaller and smaller (for a fixed number of generations in the adaptive procedure), as can be seen in Fig. 10 for NGen = 0. As a consequence, the free surface is more and more finely captured thanks to the good behavior of the IGDS scheme.

Obviously, this result also holds when the local cell size is reduced by using a higher maximum number of generations in the adaptive strategy. This result is illustrated in Fig. 13, showing the influence of NGen on the free-surface definition. Figure 13a–c shows snapshots of the captured free surface at the same time but for different values of NGen (1, 3, and 5, respectively). It is clear from these figures that the higher NGen, the finer the description of the free surface. Thus, the interest of such an adaptive process becomes clear. In order to quantify it more precisely, Table 2 gives the positions of the computed interface z_{ci} (for the present purpose, the free surface corresponds to the isoline $c_i = 0.5$) slightly before the impact ($t = 0.3$ s) for all values of NGen considered along with their corresponding exact errors (the relative error is computed using the total distance travelled down by the body during that time, $d_v = 0.438$ m). Since the mesh follows the prismatic hull as it falls through the air, the free surface is theoretically always located at its initial position of $z_p = 0.721011$ m as long as the body is sufficiently far away from it. The first free-surface deformations can be observed only shortly before the impact, and they are due to the movement of air around the approaching body. However, during the mesh fall the solver has to transport the free surface accordingly. Even though the IGDS scheme ensures a reduced numerical diffusion of the volume fraction c_i for a given mesh size, a deviation of the position of the interface to its theoretical value always exists. As can be seen in Table 2, this error can almost be cancelled using a suitable *h*-adaptive procedure since the computed positions of the interface converge to the exact value as NGen is increased.

4.3 Pressure peak

The emphasis is now on the pressure peak that arises during any drop test. When the hull hits the free surface, its violent change in speed is related to the great increase in pressure on its surface. The pressure peak corresponds to the maximum pressure applied on the prismatic body. In the early stage of the impact, the peak is localized on the keel of the body, quickly moves up onto the surface of the body, and then decreases. Thus, the phenomenon is local in both space and time, and consequently is difficult to compute accurately[9,11] or to measure experimentally.

Figure 14 shows the temporal evolution of the vertical position of the pressure peak on the prismatic hull. As no experimental data are available, the results of the simulations are compared with the asymptotic method of Scolan et al.[12] Similarly, the evolution of the intensity of the pressure peak on the prismatic hull is shown in Fig. 15. As the local cell size is reduced by increasing NGen, the convergence of the computed intensity of the pressure peak is clearly observed, and emphasizes the interest of the *h*-adaptive strategy used. The agreement between the finest adaptive computation and the

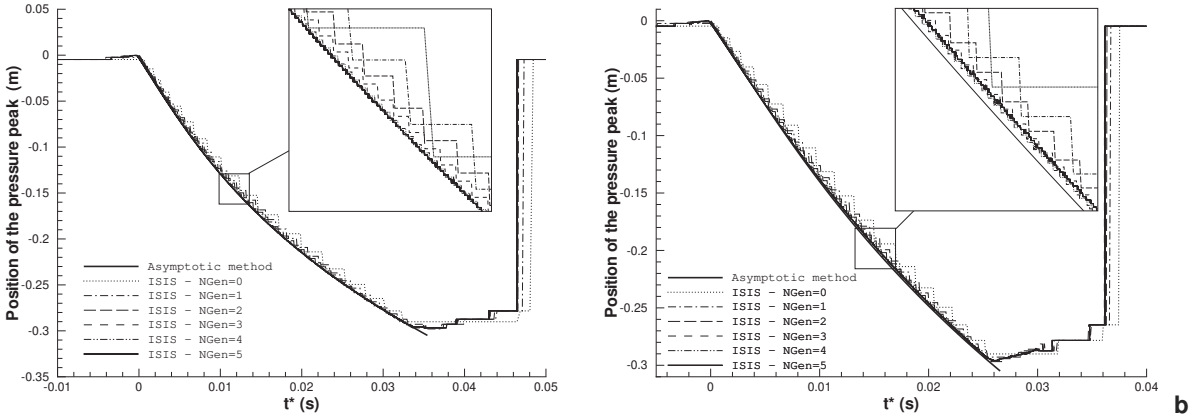

Fig. 14. Evolution of the vertical position of the pressure peak on the prismatic hull. **a** Sym-light. **b** Sym-medium

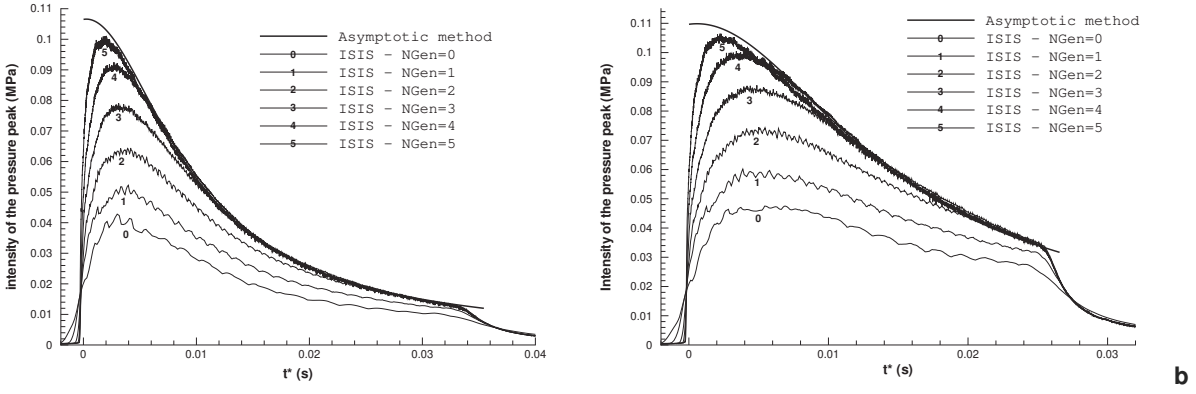

Fig. 15. Evolution of the intensity of the pressure peak on the prismatic hull. **a** Sym-light. **b** Sym-medium

asymptotic method is extremely good. This illustrates the advantages provided by the joint use of free-surface capturing techniques and automatic grid refinement.

In Fig. 15, oscillations of the computed pressures can be seen. These oscillations have nothing to do with physics, but are due to numerical considerations. Firstly, the discrete pressure is linear on each single control volume (second order finite-volume method, see Sect. 2.1) but is discontinuous at volume interfaces. Thus, the pressure peak is also discontinuous as it goes from cell to cell. Secondly, the pressure field shows very strong variations in space in the vicinity of the body, and the second-order discretization scheme with its limiters has difficulties with such rapid variations. However, these oscillations are small compared with the values of the pressure.

4.4 Numerical considerations

The physical results have already been examined. For completeness, we now consider the computational costs of the simulations.

As explained before, during any simulation both space and time adaptations have been performed. The benefits of the *h*-refinement strategy in terms of accuracy have been identified in the previous section. The advantages of time-adaptation are now considered.

In classical simulations of unsteady flows, the discretized time step Δt is prescribed either by a constant or by an imposed prescribed law that is derived a priori. As explained previously, in the context of free-surface capturing methodology, accurate predictions of the free surface can be achieved provided that the so-called CFL condition is satisfied. Thus, using a constant time-step is seldom interesting in terms of CPU time saving since it will lead to excessively small time-steps during much of the computation in order to ensure the CFL condition at all times. It should be noted that 10 inner iterations are typically needed to reach convergence for a single full time-step with grid adaptation and body movement.

For comparison, we now consider a classical imposed law of the time-step that is derived in order to fully respect the CFL condition while trying to save CPU

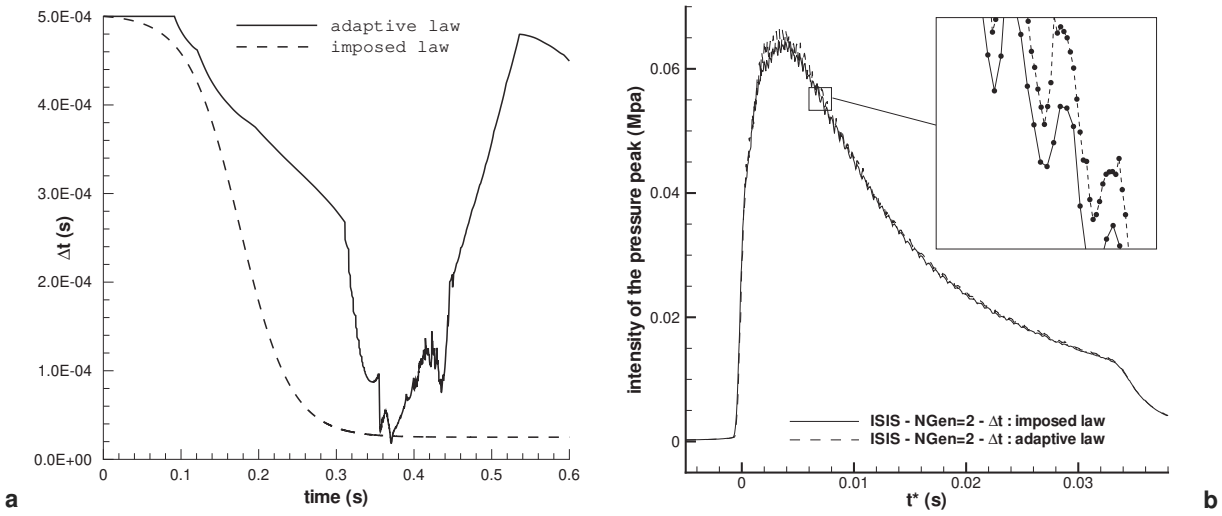

Fig. 16. Comparison between the imposed and adaptive laws of the time-step. **a** Evolution of the time-step. **b** Computed pressure peak

Table 3. Information on the computations performed ($t \in [0, 0.41\,\text{s}]$)

NGen	0	1	2	3	4	5
T_{CPU} en (min)	25	40	80	300	1 000	9 000
Max(N_{CELL})	3400	4000	5200	7000	11 000	20 000
Min(N_{CELL})	3400	4200	5700	8500	15 000	31 000

time. It should be emphasized that the derivation of this prescribed law requires some additional effort from the user, and also some knowledge about the flow solution that can be difficult to figure out a priori. The law considered is based on the hyperbolic tangent function, and is illustrated in Fig. 16a along with the time-step evolution produced by the adaptive law. This figure clearly shows that the time-step resulting from the imposed law always satisfies the CFL requirement since the values produced are lower than those from the adaptive law. However, it is also uselessly too small most of the time, leading to greater CPU time cost. Indeed, the computation with the imposed law requires 21 590 time-steps, and that with the adaptive law only requires 3580 time steps (for 0.6 s of real time of simulation and with NGen = 2). Thus, the number of time-steps is as much as six times higher in the first case. Similarly, the CPU time cost with the imposed law is five times higher than with the adaptive law. Nevertheless, it should be noted that the adaptive time-step evolution is somewhat noisy, so that the question of the influence of this phenomenon on the accuracy of the solution arises even though no numerical problems have been encountered. This particular point is investigated in Fig. 16b, which compares the intensity of the computed pressure peak using either the adaptive law or the imposed law. No significant differences can be observed.

Finally, the computational costs for all the simulations for the sym-light case are listed in Table 3. Both the CPU time (on a Pentium IV 3 GHz) and the number of cells (closely linked to the memory requirement) are considered. Similar results holds for the sym-medium case. Obviously, the number of cells for adaptive computations is not fixed during the whole calculation since the free surface is moving. During the early stage of the computation, the interface is barely deformed, and thus the number of grid points is lower than during and after the impact. It can be seen that the CPU time requirements for these simulations are reasonable bearing in mind the accuracy of the local pressure peak evolution. However, considering the ratio of the *accuracy* over the CPU time cost, the NGen = 4 simulation is a suitable compromise.

5 Asymmetric cases

This section deals with asymmetric cases of the drop tests under consideration. In the sequel, the emphasis is on computed physical results, since the salient features

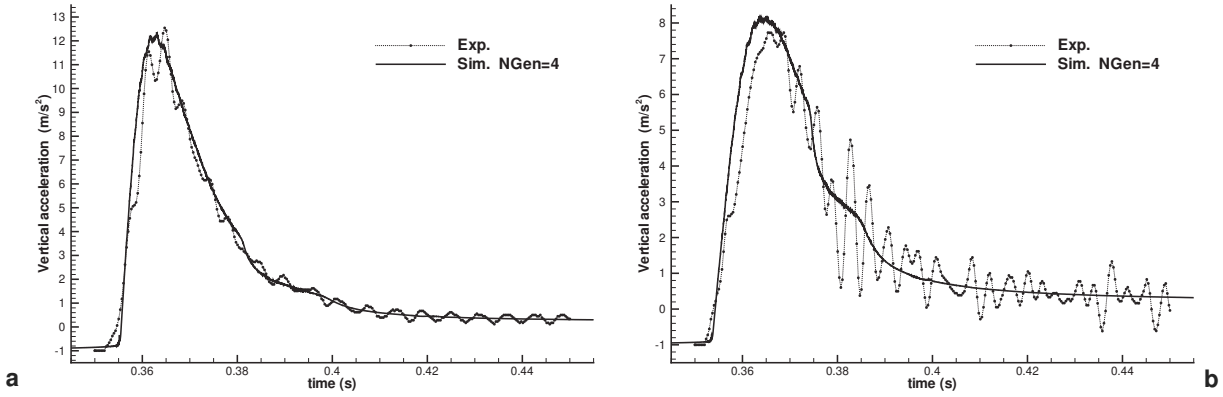

Fig. 17. Vertical acceleration. **a** Asym-light. **b** Asym-medium

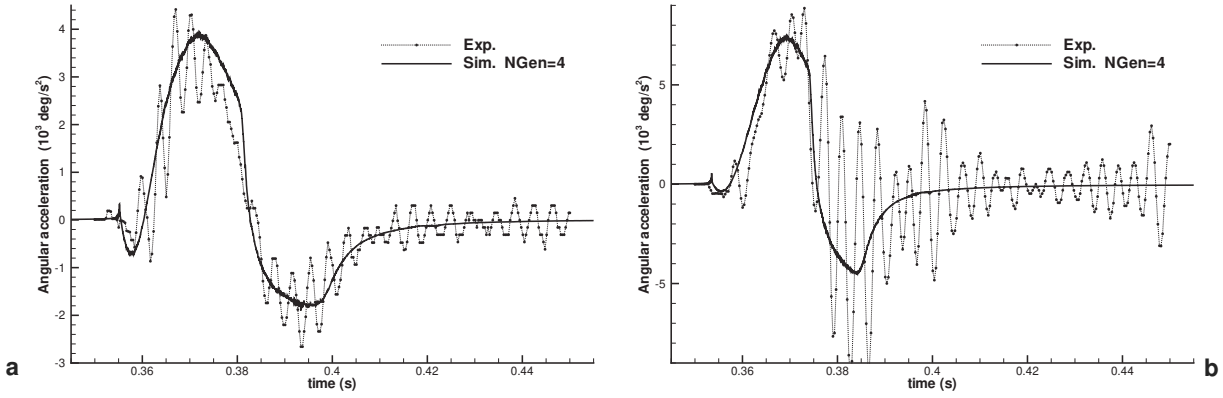

Fig. 18. Angular acceleration. **a** Asym-light. **b** Asym-medium

of the global numerical approach have been presented and analysed. Following the previously drawn conclusions, the numerical simulation is performed using NGen = 4.

5.1 Dynamic of the impacts

As previously described, the emphasis is first on the dynamic response of the drop tests under light (asym-light) and medium (asym-medium) weight conditions. Figure 17 shows the time-history of the nondimensional vertical acceleration of the prismatic hull, and Fig. 18 shows the corresponding angular accelerations for both the numerical simulation and the experiments. The experimental results clearly contain mechanical vibrations associated with structural resonances which become more pronounced as the weight increases. By its very nature, the present methodology cannot deal with such vibrations since the body is modeled as a rigid one. However, the numerical results fit quite well with the mean values of the experimental data.

5.2 Free surface

As the numerical simulation performed has a general flavor, it is particularly well suited for studying complex evolutions of the free surface. Indeed, this type of computation is not limited to the predictions of quantities defined on the surfaces of bodies. This trend is illustrated in Fig. 19, which gives a sequence of snapshots of the free-surface position around the prismatic hull. Unfortunately, no experimental data are available for comparison. However, these figures show that the free surface is finely captured thanks to the adaptive strategy. In addition, the capacity of the whole numerical approach to compute wave breaking is also illustrated here since the interface rolls up and eventually crashes onto the top surface of the hull. Closer views of the meshes around the computed free surface are given in Fig. 20 for NGen = 4. As expected, the discontinuity is strictly located inside the finest mesh area.

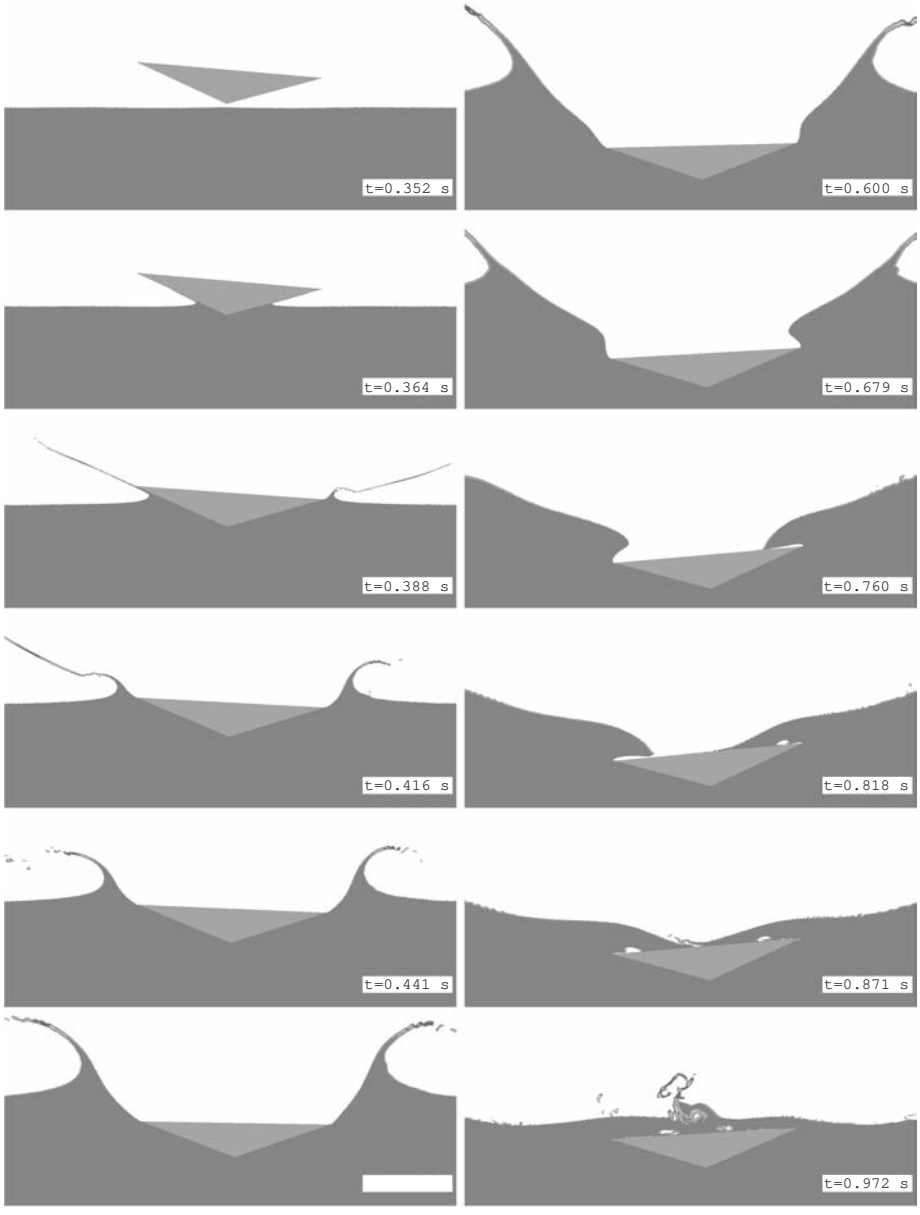

Fig. 19. Temporal evolution of the free surface (asym-light).

5.3 Pressure peak

The emphasis is once again on the pressure peak that arises during any drop test. As the problem is asymmetric, two different pressure peaks appear: one on each side of the hull. The evolution of the positions of these peaks as they quickly rise are shown in Fig. 21. The greater the mass of the hull, the shorter the time for the peaks to reach the upper part of the geometry.

Figure 22 shows the evolution of the intensities of the pressure peaks on the prismatic hull. As expected, the right side is submitted to a stronger pressure peak since the 5° heel angle brings it more directly in contact with the free surface during the impact. For the two cases considered, the maximum rises of pressure on the right side are more than twice as large as the corresponding ones on the left side, and also larger than the pressure peaks computed for the symmetric cases (see Fig. 15).

6 Conclusion

This article has described a modern free-surface capturing strategy implemented in an unstructured finite-volume viscous flow solver. Furthermore, in a general framework, the methods that allow the consideration of moving bodies have been presented in detail. Special attention has been paid to the regridding strategies for the mesh to be adjusted to the new positions of bodies in order to reduce the CPU time. The issues of solving

Newton's laws and performing fluid–structure coupling have also been addressed. A mesh adaptation strategy has been fully integrated in the code, making it a single tool for performing accurate computations of free-surface flows. The data structure used, the grid alterations performed, the explicit indicator, and the adaptation procedure have been detailed precisely.

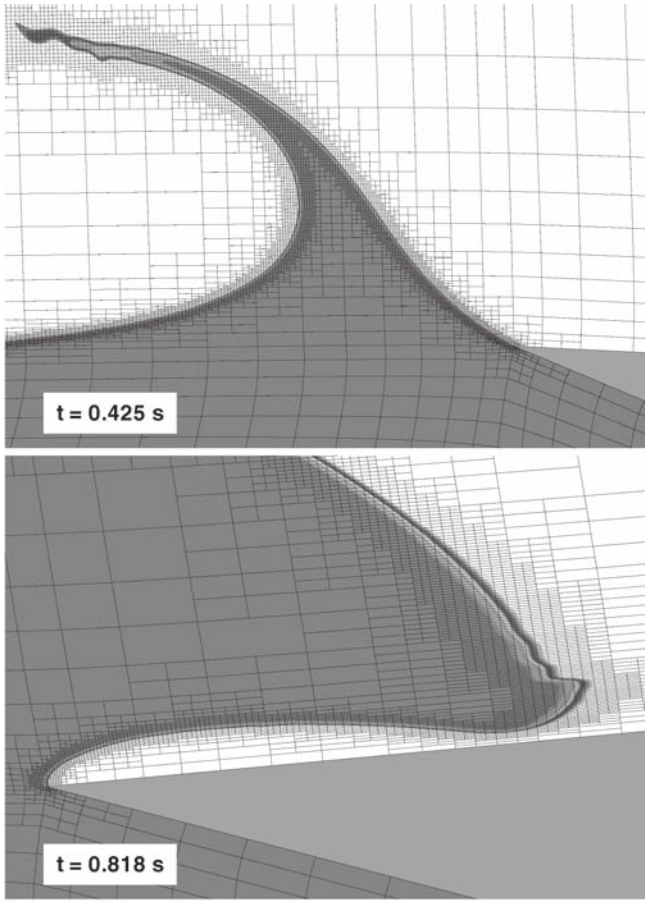

Fig. 20. Inside views of the free surface and the corresponding meshes.

The whole numerical approach has been applied to the computation of unsteady flows around a prismatic hull falling through the air and impacting the free surface of water. Symmetric and asymmetric drop tests have been considered, and extensive comparisons have been presented between numerical results and both experimental data and theoretical results from an asymptotic method. The excellent agreement observed has shown that the global approach is relevant for the accurate computation of free-surface flows with moving bodies. Of particular interest was the demonstration that a local and intense phenomenon, namely the pressure peak arising during the impact, can be accurately computed in the general framework of the presented simulations, thanks to the *h*-adaptive strategy, for a reasonable CPU time and a minimum user effort.

Clearly, the next step will involve three-dimensional applications while keeping the generality of the approach. The extension of the present methodology to 3D problems has already been carried out.[2,23,30] However, the remaining question of building a parallelized version of the adaptive procedure is not straightforward. Indeed, three-dimensional computations require a large number of points even with the use of a suitable mesh adaptation strategy. Thus, a domain-decomposition method is a way to maintain reasonable CPU time using the performance of multiprocessor computers. Future works will aim at developing an efficient parallelization of the adaptive procedure with a dynamic load-balancing. In fact, the adaptive procedure can be parallelized in the same way as the other numerical techniques presented (whose parallelization is based on the message passing interface (MPI)[31]), But ensuring

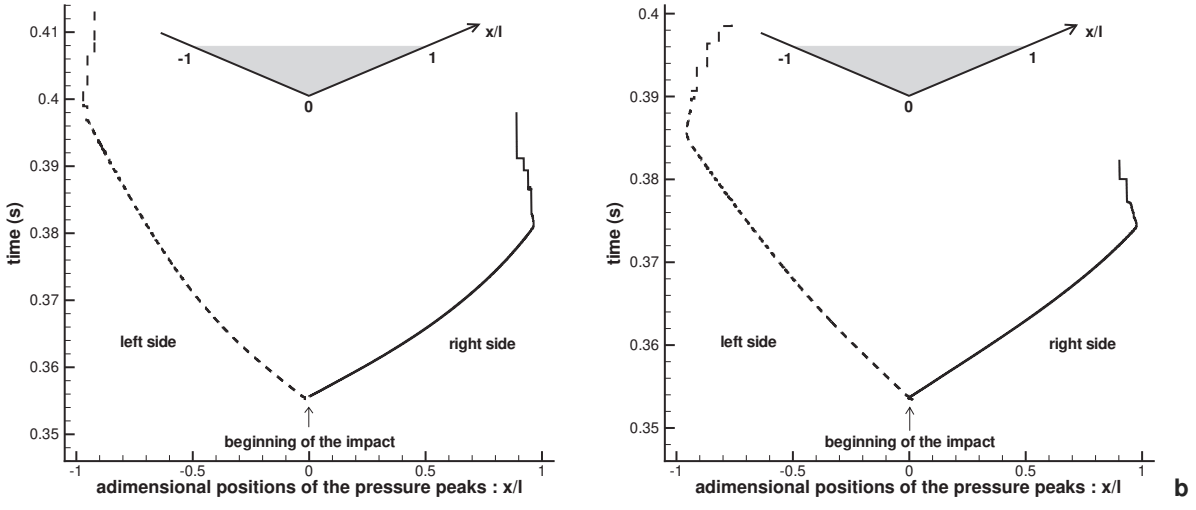

Fig. 21. Evolution of the positions of the pressure peaks on the prismatic hull. **a** Asym-light. **b** Asym-medium

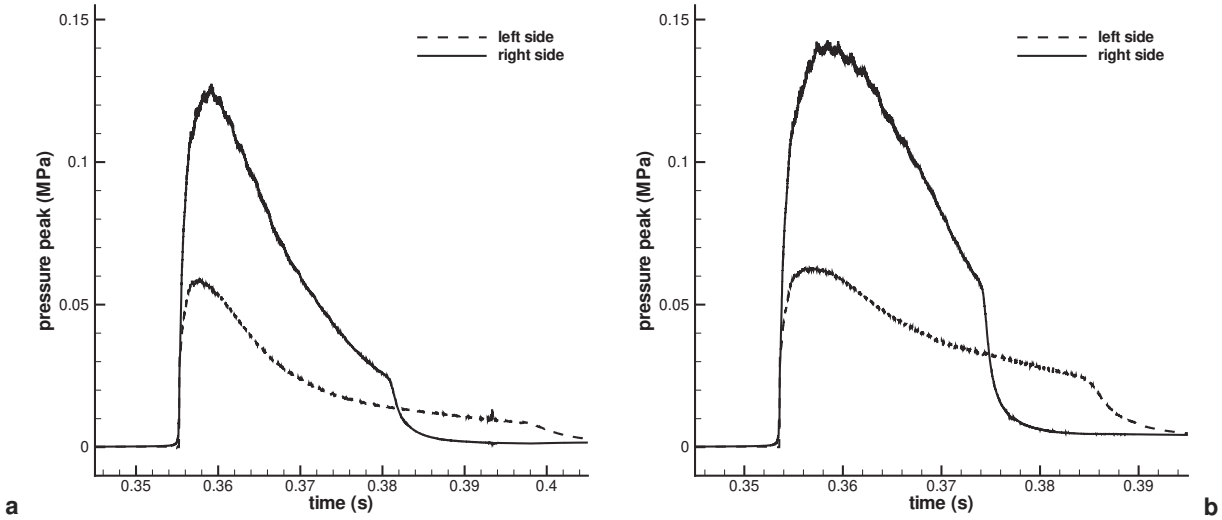

Fig. 22. Evolution of the intensities of the pressure peaks on the prismatic hull. **a** Asym-light. **b** Asym-medium

a dynamic load-balancing may be a more tedious task and this feature is mandatory for the efficiency of the parallelization of the adaptive procedure.

Acknowledgments. The authors gratefully thank Yves-Marie Scolan for making the results from the asymptotic method available.

References

1. Larsson L, Stern F, Bertram V (2000) Summary, conclusions and recommandations of the Gothenburg 2000 workshop. In: A workshop on numerical ship hydrodynamics. Chalmers University of Technology, Göteborg, Sweden
2. Queutey P, Visonneau M, Ferrant P (2004) Numerical investigation of wave interaction with a fixed vertical cylinder. Int J Offshore and Polar Engineering (IJOPE) 14:202–209
3. Queutey P, Visonneau M (2004) Three-dimensional CFD simulations using a free-surface capturing strategy. Proceeding of the 3rd International Conference on Computer and IT Applications in the Maritime Industries, Siguënza, Spain
4. Ubbink O (1997) Numerical prediction of two-fluid systems with sharp interfaces. PhD Thesis, Imperial College of Science Technology and Medicine, University of London
5. Muzaferija S, Perić M (1998) Computation of free surface flows using interface-tracking and interface-capturing methods. In: Mahrenholz O, Mankiewicz M (eds) Nonlinear water wave interaction. Computational Mechanics, Southampton, pp 59–100
6. Deng G, Guilmineau E, Queutey P (2001) Capture et suivi d'interfaces d'écoulements de fluides visqueux incompressibles non-miscibles. 8ème Journées de l'Hydrodynamique, Nantes, France, p 17–30
7. Muscari R, Mascio AD (2004) Numerical modeling of breaking waves generated by a ship's hull. J Mar Sci Technol 9: 158–171
8. Azcueta R (2001) Computation of turbulent free-surface flows around ships and floating bodies. PhD Thesis, University of Hambourg
9. Leroyer A (2004) Fluid/motion interaction for solid and flexible bodies by resolution of the Navier–Stokes equations. Contribution to the numerical modelisation of cavitating flows. PhD Thesis, University of Nantes/Ecole Centrale de Nantes, France. Can be downloaded (in French) at ftp://ftp.ec-nantes.fr/pub/DMN/Thesis/these_leroyer.ps.gz
10. Yamasaki J, Miyata H, Kanai A (2005) Finite-difference simulation of green water impact on fixed and moving bodies. J Mar Sci Technol 10:1–11
11. Hay A (2004) A study of numerical error estimation and local adaptive unstructured mesh refinement strategies for the Reynolds averaged Navier–Stokes equations. PhD Thesis, University of Nantes/Ecole Centrale de Nantes, France. Can be downloaded (in French) at ftp://ftp.ec-nantes.fr/pub/DMN/Thesis/these_hay.ps.gz
12. Scolan YM, Coche E, Coudray T (1999) Etude analytique et numérique de l'impact hydrodynamique sur des carènes dissymétriques. 7ème Journées de l'Hydrodynamique, Marseille
13. Cointe R, Fontaine E, Molin B (2004) On energy arguments applied to the hydrodynamic impact force. J Eng Math 48:305–319
14. Spalart P, Allmaras S (1992) A one-equation turbulence model for aerodynamic flows. AIAA 30th Aerospace Sciences Meeting, Reno, NV, AIAA Paper 92-0439
15. Menter F (1993) Zonal two-equation $k-\omega$ turbulence models for aerodynamic flows. AIAA 24th Fluid Dynamic Conference, Orlando, FL, AIAA Paper 93-2906
16. Deng G, Visonneau M (1999) Comparison of explicit algebraic stress models and second-order turbulence closures for steady flow around ships. 7th Symposium on Numerical Ship Hydrodynamics, Nantes, France, p 4.4.1–4.4.15
17. Duvigneau R, Visonneau M, Deng G (2003) On the role played by turbulence closures in hull shape optimization at model and full scale. J Mar Sci Technolo 8:11–26
18. Jasak H, Weller H, Gosman A (1999) High-resolution NVD differencing scheme for arbitrarily unstructured meshes. Int J Numer Methods Fluids 31:431–449
19. Pržulj V, Basara B (2001) Bounded convection schemes for unstructured grids. Proceedings of the AIAA 15th Computational Fluid Dynamics Conference, Anaheim, CA, AIAA paper 2001–2593
20. Jasak H, Weller H (1995) interface tracking capabilities of the inter-gamma differencing scheme. Unpublished results
21. Leonard B (1988) Simple high-accuracy resolution program for convective modelling of discontinuities. Int J Numer Methods Fluids 8:1291–1318

22. Gaskell P, Lau A (1988) Curvature-compensated convective transport: smart, a new boundedness preserving transport algorithm. Int J Numer Methods Fluids 8:617–641
23. Hay A, Queutey P, Visonneau M (2004) Computation of three-dimensional free-surface flows with an automatic adaptive mesh refinement and coarsening strategy. 25th ONR Symposium on Naval Hydrodynamics, St. John's, Newfoundland and Labrador, Canada
24. McDonald H, Whitfield D (1997) Self-propelled maneuvering underwater vehicles. 21th Symposium on Naval Hydrodynamics, Washington, p 478–489
25. Farhat C, Degand C, Koobus B (1998) Torsional springs for two-dimensionnal dynamic unstructured fluid meshes. Comput Methods Appl Mech Eng 163:231–245
26. Ferziger J, Perić M (2002) Computational methods for fluid dynamics, 3rd edn. Springer, Berlin, New York
27. Peterson R, Wyman D, Franck C (1997) Drop tests to support water-impact and planing boat dynamics theory. Technical Report TR-97, CSS Technical Report, Coastal Systems Station, Panama City
28. Xu L, Troesch A, Peterson R (1999) Asymmetric hydrodynamic impact and dynamic response of vessels. J Offshore Mech Arctic Eng 121:83–89
29. Okada S, Sumi Y (2000) On the water impact and elastic response of a flat plate at small impact angles. J Mar Sci Technol 5:31–40
30. Leroyer A, Visonneau M (2003) Moving bodies in viscous flow simulation. 6th Numerical Towing Tank Symposium, Rome
31. (1997) MPI-2: extensions to the message-passing interface. Technical Report, University of Tennesse, Knoxville, TN. See http://www.mpi-forum.org/docs/mpi-20-html/mpi2-report.html