INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **49**:785–816

Computation of free-surface flows with local mesh adaptation

A. Hay*,† and M. Visonneau‡

Laboratoire de Mécanique des Fluides (CNRS UMR 6598), Equipe de Modélisation Numérique,
Ecole Centrale de Nantes, B.P. 92101, 1 rue de la Noë, 44321 Nantes, France

SUMMARY

This paper describes a modern free-surface capturing strategy implemented in an unstructured finite-volume viscous flow solver that can handle moving grids composed of arbitrary-shaped control volumes. An adaptive mesh strategy is fully integrated in the code making it a single tool for dynamically maintaining a prescribed density of grid points around the steady or unsteady interface between air and water. The whole adaptive procedure is described in detail. The efficiency of the overall approach is examined on two- and three-dimensional hydrodynamic applications. The adaptive strategy achieves interesting gains in terms of computational and human efforts compared to single-mesh computations. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: free-surface flow; adaptive method; unstructured mesh; three-dimensional hull

1. INTRODUCTION

The use of computational fluid dynamics (CFD) tools for predicting the powering performance of ships remains challenging because of the numerous physical difficulties which characterize the flow around a real ship. Among the hard points which have to be solved to get a reliable simulation of ship flow, one can list, without being exhaustive, the accurate simulation of ship stern flows which is highly dependent on turbulence modelling, the modelization of the hull/propeller coupling and the simulation of the free-surface deformation. On the one hand, one must notice that the accuracy of CFD tools for predicting crucial global quantities like resistance and self-propulsion factor is still limited. On the other hand, CFD is complementary to towing tank tests because it provides a large amount of detailed informations on the flow which helps the designer to improve the performance of a new ship.

It has been observed during the last Gothenburg 2000 workshop [1] that the free-surface capturing methodology was more and more popular among the CFD developers dealing with

---

*Correspondence to: Alexander Hay, Laboratoire de Mécanique des Fluides (CNRS UMR 6598), Equipe de Modélisation Numérique, Ecole Centrale de Nantes, B.P. 92101, 1 rue de la Noë, 44321 Nantes, France.

†E-mail: hay.alexander@free.fr

‡E-mail: michel.visonneau@ec-nantes.fr

Received 15 December 2004
Revised 2 June 2005
Accepted 4 June 2005

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 phenomenon, which may occur during the initialization period, is perfectly tolerated. When discretization schemes with compressive property are used to discretize 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 the 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.

The present study deals with such a strategy to emphasize its ability to precisely capture the free-surface discontinuity for a reduced computational and human effort. To the authors's knowledge, such an approach has never been developed nor studied for hydrodynamic applications. Section 2 first presents the computational techniques used to enable the treatment 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 arbitrary-shaped control volumes. This feature enables flexible grid adaptation as the mesh can easily be refined but also coarsened by agglomeration of control volumes with no need for particular treatment of adapted cells [4, 5]. Also presented is the free-surface capturing method implemented with its specific discretization schemes that are required to get an accurate description of a density discontinuity. Section 3 describes the local adaptive mesh procedure which is designed in the framework of unstructured grids. The adaptive process is made dynamic and low CPU time consuming by the use of an adequate data structure [4]. The adaptive procedure is entirely included in the flow solver making it a complete automatic single tool for computing hydrodynamic flows. It is first applied, in Section 4, to the computation of the free-surface elevation behind a two-dimensional fully submerged hydrofoil moving horizontally at constant speed and angle of attack. Then, the Rayleigh–Taylor instability problem is examined. The main parameters of the adaptive procedure are studied for determining their influences on the overall accuracy of the proposed methodology. Finally, the three-dimensional simulation of the free-surface flow around a Wigley hull is presented. The advantages of the global approach are demonstrated in terms of accuracy, optimal use of discretization points and user-friendliness.

2. COMPUTATIONAL APPROACH

2.1. Presentation of the flow solver

The ISIS flow solver, developed by the EMN (Equipe de Modélisation Numérique i.e. CFD Department of the Fluid Mechanics Laboratory), uses the incompressible URANS. The solver is based on the finite-volume method to build the spatial discretization of the transport equations. The face-based method is generalized to unstructured meshes for which non-overlapping 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.

In the case of turbulent flows, additional transport equations for modelled variables are solved. They are discretized and solved using the same principles as for the momentum equations. Several near-wall low-Reynolds number turbulence models, ranging from the one-equation Spalart–Allmaras model [6], two-equation k–ω closures [7], to a full-stress transport R_{ij}–ω model [8], are implemented in the flow solver to take into account the turbulence phenomena.

Incompressible and non-miscible flow phases are modelized through the use of conservation equations for each volume fraction (or concentration) c_i of each phase i.

2.2. *Conservation equations*

The flow solver can deal with multi-phase flows and moving grids. Considering incompressible flow of viscous fluid under isothermal conditions, mass, momentum and volume fraction conservation equations can be written as follows (using the generalized form of Gauss' theorem):

$$\frac{\partial}{\partial t}\int_V \rho\,\mathrm{d}V + \int_S \rho(\mathbf{U}-\mathbf{U}_d)\cdot\mathbf{n}\,\mathrm{d}S = 0 \tag{1a}$$

$$\frac{\partial}{\partial t}\int_V \rho\mathbf{U}\,\mathrm{d}V + \int_S \rho\mathbf{U}(\mathbf{U}-\mathbf{U}_d)\cdot\mathbf{n}\,\mathrm{d}S = \int_V (\rho\mathbf{g}-\nabla p)\,\mathrm{d}V + \int_S (\mathbb{T}+\mathbb{T}_t)\cdot\mathbf{n}\,\mathrm{d}S \tag{1b}$$

$$\frac{\partial}{\partial t}\int_V c_i\,\mathrm{d}V + \int_S c_i(\mathbf{U}-\mathbf{U}_d)\cdot\mathbf{n}\,\mathrm{d}S = 0 \tag{1c}$$

where V is the domain of interest (i.e. a control volume) bounded by the closed surface S moving at the velocity $\mathbf{U}_d$ with a unit normal vector $\mathbf{n}$ directed outward. $\mathbf{U}$ and p represent, respectively, the velocity and the pressure. $\mathbb{T}$ and $\mathbb{T}_t$ refer to the viscous and Reynolds stress tensors, whereas $\mathbf{g}$ is the gravity vector. While $\mathbb{T}_t$ is determined according to the turbulence model used, $\mathbb{T}$ follows the classical relation of newtonian fluid for incompressible flows.

The effective flow physical properties (viscosity μ and density ρ) are obtained from the phase physical properties (μ_i and ρ_i) by the following constitutive relations:

$$\rho = \sum_i c_i\rho_i, \quad \mu = \sum_i c_i\mu_i, \quad \sum_i c_i = 1 \tag{2}$$

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

$$\frac{\partial}{\partial t}\int_V \mathrm{d}V - \int_S \mathbf{U}_d\cdot\mathbf{n}\,\mathrm{d}S = 0 \tag{3}$$

2.3. *Discretization techniques*

2.3.1. Spatial discretization. All the flow variables are stored at the geometric centres of the arbitrary-shaped cells. Surface and volume integrals are evaluated according to second-order accurate approximations by using the values of integrand that prevail at the centre of the face f, or cell C, and neighbouring cells. The various fluxes appearing in the discretized equations are built using centred and/or upwind schemes. For example, the convective fluxes are obtained by the hybrid differencing scheme (HDS) which is a combination of the upwind differencing scheme (UDS) and the centred differencing scheme (CDS). Unlike the practical approach [9, 10] where the CDS/UDS blending is fixed to a global blending factor for all the faces of the mesh, the HDS results from a local blending factor based on the signed Peclet number associated to the face.

Besides, a pressure equation is obtained in the spirit of the Rhie and Chow [11] procedure. Momentum and pressure equations are solved in a segregated way like in the well-known SIMPLE coupling algorithm.

2.3.2. Discretization schemes for the concentration transport equation. Except for the convection terms and volumetric mass fluxes, interfacial quantities q_f are rebuilt linearly from the cell-centred quantities and their available cell-centred gradients. Special attention has to be paid to face reconstructions of the volume fraction c_i. The challenge posed by the discretization of the transport equation for the concentration is the accurate modeling of a contact discontinuity, i.e. the free-surface. In order to ensure face bounded reconstructions and to avoid unrealistic oscillations, the search for an acceptable compromise between accuracy and boundedness of the concentration ($0 \leqslant c_i \leqslant 1$) is a key point [12, 13]. Moreover, the method must also preserve the sharpness of the interface through the transport equation (1c).

These requirements are fulfilled by the inter-gamma differencing scheme (IGDS) [14] which introduces downwind differencing since compressive characteristics are required for an accurate interface capturing. Thanks to a normalized variable diagram (NVD) analysis [15], this scheme enforces local monotonicity and the convection boundedness criterium (CBC) [16]. 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 follows: $Co = \Delta t \mathscr{F}/V$ where $\mathscr{F}$ is the total (positive) velocity flux through the considered face, V is the volume of the upwind cell and Δt is the global time step of the temporal discretization. Therefore, the CFL condition requires the discretized time step not to be too large.

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].

2.3.3. Temporal discretization. The temporal discretization is based on a three-steps upwind discretization which is second-order accurate. The convection and diffusion terms are treated implicitly. When computing flows with different immiscible phases, the global time step of the discretization has to be defined in agreement with the Courant number limitation of the discretization scheme for the concentration equation as explained previously. This issue will be addressed when applications are considered in Section 4.

3. ADAPTIVE TECHNIQUES

As indicated before, a free-surface capturing methodology is based on the transport of a contact discontinuity. Although the compressive discretization schemes described above guarantee an interface captured over three to five cells, it is crucial to keep as small as possible the characteristic length of the cells on either side of the interface to avoid too much a 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 follow the temporal evolution of the interface and to maintain a fine computational grid around it. Moreover, an automatic grid adaptation process frees the user from a tedious task: the generation of a mesh suited everywhere to the unknown interface position since one can start from a uniform grid which will be automatically refined around the interface.

For being 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 simulations on complex geometries. Secondly, when unsteady flow are considered, the free-surface may be submitted to rapid variations in time so that mesh adaptation may be required very often during numerical simulations 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 [4]. Moreover, in order to preserve the CPU time cost of the local grid adaptation process to its minimum, the adaptive procedure should be based on a data structure enabling dynamic grid alterations. Lastly, a mapping routine has to be part of the whole adaptation step to interpolate the solution computed on the last considered grid to the new adapted one for the computation to be continuous.

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

3.1. Data structure

For the mesh adaptation process to be flexible, quick and easy to implement, a suitable data structure is required. The present local mesh adaptation procedure is based on the notion of relationship between the successive generations of elements of the grids. Connectivities of *relationship* are considered for both *the control volume* type of element and *the face* type of element. They are presented here only for the control volumes but similar notions applied to the faces of the grids. The relationships introduced between the cells lead to a natural vocabulary of *family*, *father*, *son* and *brother* as illustrated in Figure 1 (either for two-dimensional control volumes or three-dimensional faces). The initial mesh is composed 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. And, the father and its sons form a family. 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. Besides, negative generations can exist due to the unrefinement by agglomeration that coarsen the initial grids (see Section 3.2). It should be also pointed out that all the elements of the successive generations are kept in memory (they are only erased by unrefinement) but the additional memory cost is reduced since only local adaptation is performed.

Such a data structure permits to address 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 its father (and also the destruction of its sons). Thus, the refinement/unrefinement process becomes very dynamic and quick to achieve. Secondly, it permits to easily and exactly recover the initial mesh if refinement is no longer necessary in an area of the computational grid as it is the case for unsteady problems.

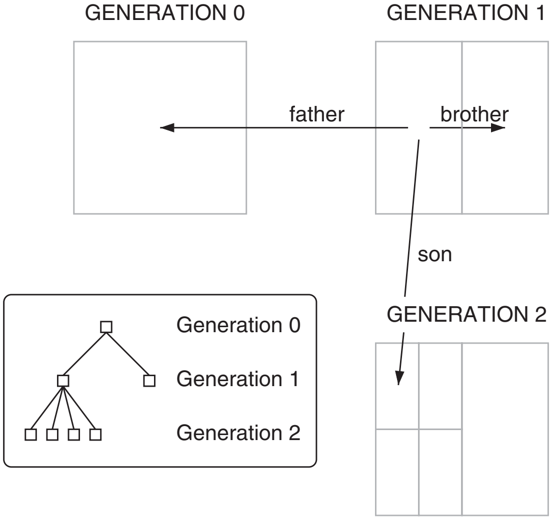


Figure 1. Relationship between the different generations of elements.

3.2. Grid alteration

During the refinement process, each control volume to be refined is split into several smaller ones of the same topology. Thus, the topology of elements is kept unchanged and only their sizes are adapted. Doing so, the initial desired local mesh quality is preserved everywhere during all the simulation. For two-dimensional grids, as illustrated in Figure 2, the refinement process can occur with a possible directional sensitivity for flows with simple features. For three-dimensional meshes, only isotropic refinement is possible and leads to the cell divisions presented in Figure 3. However, one may underline that altered grids do not require any particular treatment in the flow solver. As a matter of fact, a non-refined neighbour of a refined cell presents a so-called *hanging-node* which is accounted for naturally by our face-based finite-volume method: a face with a hanging-node is simply seen as several smaller faces.

Besides, an original kind of grid alteration can also be performed based on agglomeration algorithms [4]. It permits to proceed to coarsen the elements of generation 0 (i.e. elements that have not been previously refined). The agglomeration algorithms permit to unrefine the selected parts of the initial grid by grouping neighbouring cells. The different groups are then merged into a bigger cell. The agglomeration can result from a fusion of a face or of a node as illustrated in Figure 4. As shown, the resulting agglomerated cell is of no usual shape but once again the face-based methodology is designed to deal with such peculiarities.

Finally, special treatments are necessary when refining a curved boundary surface since two requirements have to be fulfilled when a new node is added on the surface of the body:

- the node must be placed on the exact geometry which should be described either analytically or by a CAD description readable by the flow solver,
- for convex boundaries and high aspect ratio cells, the inclusion of a new node on the surface of the body should not generate neighbouring cells with negative volumes.

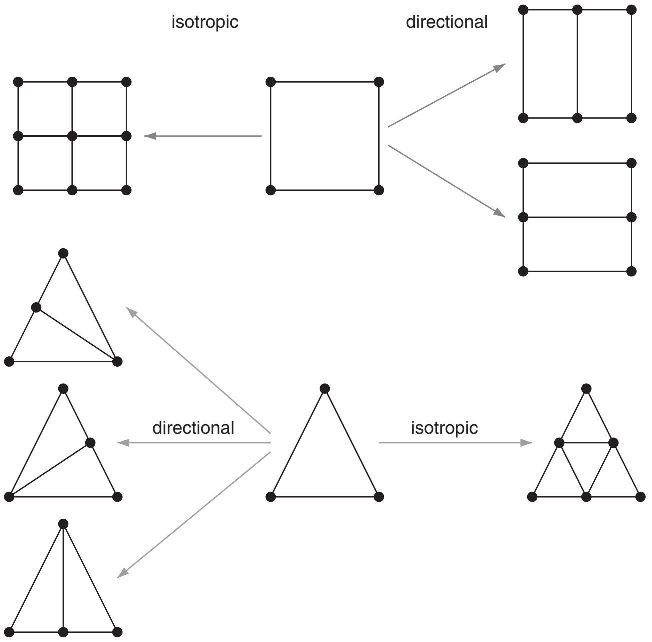


Figure 2. Refinement of two-dimensional volumes.

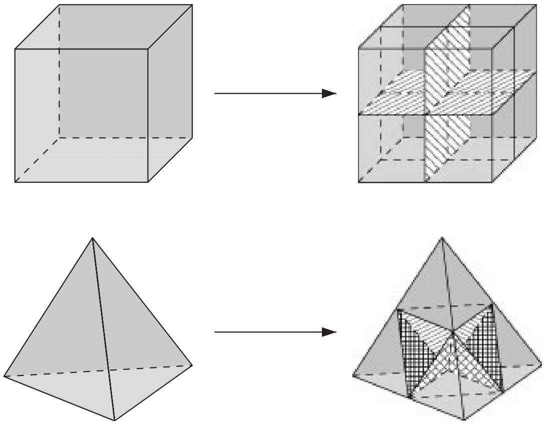


Figure 3. Isotropic refinement of three-dimensional volumes.

To avoid the apparition of ill-conditionned near-wall cells, the grid is moved in accordance with the discrete boundaries deformation by generalizing the mesh deformation tools already included in the flow solver and previously used for shape optimization studies. For two- and three-dimensional cases, a spring analogy is employed to control the deformation of the grid [17]. The association of lineal and torsional springs provides a powerful mesh deformation

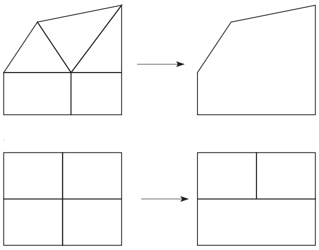


Figure 4. Examples of agglomerated cells.

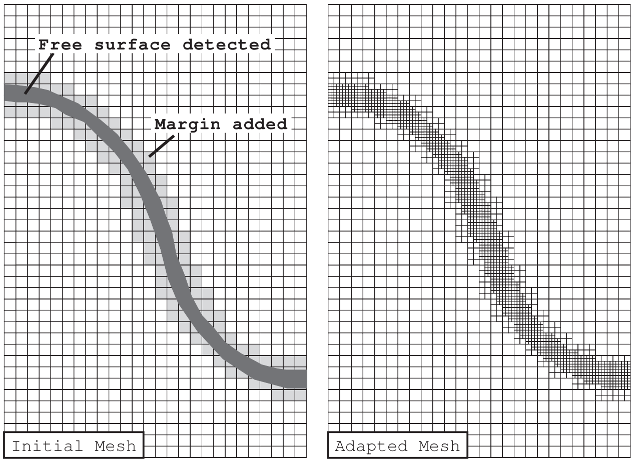


Figure 5. Safety margin around the interface.

tool, which maintains the grid quality near the wall even for high deformations. However, this technique is not described here since the applications considered in this study do not require it.

3.3. Description of the procedure

The goal of the proposed adaptive procedure is to maintain a desired prescribed cell size around the free-surface of the considered problem. Therefore, an indicator is required to localize in space the free-surface and is easy to derive using information from the volume fraction c_i. The considered indicator is proportional to the norm of the gradient of the concentration which is already computed during the calculation of the flow. As the indicator is derived using only quantities already computed, it is referred to as an explicit indicator. However, a safety margin is also added around the detected zones as indicated in Figure 5. The question of what should be the typical size of the safety margin immediately arises. But, as no previous study of such a procedure for capturing free-surface flows is available, the answer is difficult to formulate *a priori*. However, it is legitimate to consider that an accurate

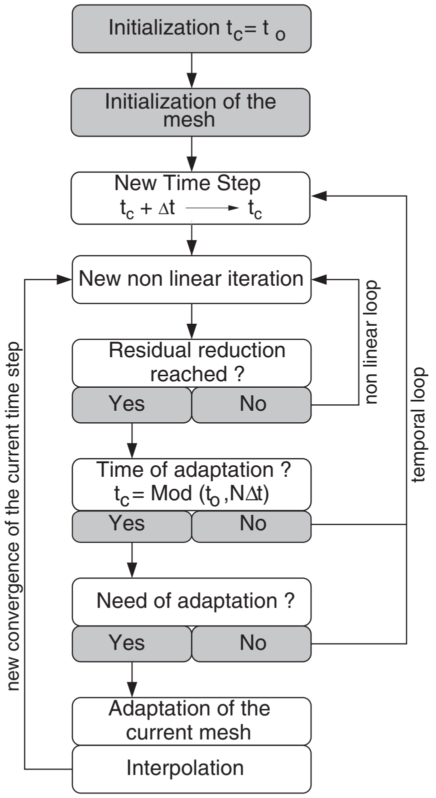


Figure 6. Adaptive unsteady procedure.

computation of the free-surface may demand that the flow around the free-surface to be correctly computed which further require an adequately large margin. Therefore, the size of the safety margin may drastically influence the accuracy of the captured free-surface. This issue will be addressed in Section 4.1.

The adaptive procedure for unsteady computation is summarized in Figure 6. An adaptation step is asked every N time steps during the temporal loop. This parameter has to be chosen sufficiently small so that the adapted meshes can correctly follow the temporal evolution of the interface. Actually, N is always fixed to a small value so that the need for adaptation is checked very often. But, if no significant adaptation is required, the computation is resumed. Otherwise, the current mesh is modified according to the explicit indicator. Doing so, the parameter N has no longer any influence on the adaptive algorithm. Following the mesh refinement/unrefinement, a mapping procedure is called to interpolate the solutions computed on the previous grid onto the new adapted mesh. The mapping operators are described in the next section. However, the current interpolated flow variables no longer satisfy the discrete unsteady operator. Thus, it is mandatory to achieve a new convergence of the flow residuals at the current time step as shown in Figure 6. The CPU time overhead 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 which is the maximum number of generations allowed to be created. Thus, the local characteristic length h of any part of the initial mesh can, at most, be reduced down to $h/2^{\mathsf{NGen}}$ in each direction. The higher the NGen is, the more accurate the free-surface capturing is expected to be.

3.4. Mapping procedure

This section is devoted to the mapping procedure developed to interpolate any flow variable ϕ, stored at the centres of the control volumes, from grid to grid. In the framework of the finite-volume methodology, the value ϕ_i at the centre $\mathbf{x}_i^c$ of a cell i is the mean value of ϕ over a control volume:

$$\phi_i = \frac{1}{V_i} \int_{V_i} \phi \, \mathrm{d}V \tag{4}$$

where V_i is the volume of the cell i. Besides, a second-order accurate finite-volume discretization is considered so that any flow variable is assumed to vary linearly over the control volume:

$$\phi(\mathbf{x}) = \phi_i + (\nabla \phi)_i \cdot (\mathbf{x} - \mathbf{x}_i^c) \tag{5}$$

As explained previously, a refined cell is called a father and is composed of N_{son} smaller cells. When the refinement is performed, the mapping procedure has to interpolate ϕ at all the centres $\mathbf{x}_{\text{son}}^c$. The interpolation is achieved locally according to Equation (5) and the known value of ϕ and its derivatives at the centre of the father cell:

$$\phi(\mathbf{x}_{\text{son}}^c) = \phi_{\text{son}} = \phi_{\text{father}} + (\nabla \phi)_{\text{father}} \cdot (\mathbf{x}_{\text{son}}^c - \mathbf{x}_{\text{father}}^c) \tag{6}$$

And, when a family of sons is unrefined, the mapping procedure has to interpolate ϕ at the centre $\mathbf{x}_{\text{father}}^c$ of the father cell. The interpolation is achieved locally according to Equation (4) and the known values of ϕ at the centres of the son cells:

$$\phi_{\text{father}} = \frac{1}{V_{\text{father}}} \int_{V_{\text{father}}} \phi \, \mathrm{d}V = \frac{1}{V_{\text{father}}} \sum_{\text{son}} \left(\int_{V_{\text{son}}} \phi \, \mathrm{d}V \right) = \frac{1}{V_{\text{father}}} \sum_{\text{son}} (V_{\text{son}} \phi_{\text{son}}) \tag{7}$$

The proposed mapping procedure is applied to any flow variable ϕ at any time step. As explained previously, the current interpolated flow variables no longer satisfy the discrete unsteady operator. Therefore, a new convergence of the flow residuals at the current time step is achieved. As a result, the mapped quantities at the current time step are only usefull for initializing the new iterative solution procedure. However, the discretization of the terms involving a temporal derivative requires the values of ϕ at the two previous time steps (see Section 2.3). These values are computed on the new adapted grid using the proposed mapping procedure. Thus, a small error is introduced which would be very difficult to evaluate. However, this methodology has been shown to ensure the continuity in time of the computed solutions and no significant error has been observed so far.

4. APPLICATIONS

4.1. Free-surface related to a submerged hydrofoil

4.1.1. Description of the problem. The free-surface deformation behind a two-dimensional immersed hydrofoil is first considered in this section. Duncan [18] has performed an extensive experimental study of breaking and non-breaking waves produced by a fully submerged two-dimensional hydrofoil moving horizontally at constant speed and angle of attack. The configuration of the test case is shown in Figure 7. The hydrofoil has a NACA 0012 shape with a chord of $c = 20.3\,\text{cm}$ and the depth of submergence is fixed to $h = 23.6\,\text{cm}$. The angle of attack is set to 5° and the foil speed is $U_{\text{NACA}} = 0.8\,\text{m}\,\text{s}^{-1}$. The two considered fluids are water and air with corresponding densities of $\rho_{\text{water}} = 998.1\,\text{kg}\,\text{m}^{-3}$ and $\rho_{\text{air}} = 1.2\,\text{kg}\,\text{m}^{-3}$. Based on the chord of the hydrofoil, the Reynolds number (in water) is $Re = 1.62 \times 10^5$ and the Froude number is $Fr = 0.567$ since the gravity is fixed to $g = 9.81\,\text{m}\,\text{s}^{-2}$. This experimental set-up corresponds to a steady free-surface deformation without any wave breaking phenomenon. A comparison between numerical results from a surface tracking method and a surface capturing method has already been performed in Reference [19]. And, using the surface tracking method, the shape of the submerged hydrofoil has also been optimized [20] to minimize the free-surface elevation generated. Following these previous studies, the one-equation model of Spalart–Allmaras, which includes a low Reynolds number formulation, is used to modelize the turbulence. However, the present study is focused on capturing an accurate free-surface elevation for the lowest computational and human costs using the adaptive mesh refinement/unrefinement strategy.

Similar to the experimental set-up, for all the computations reported, the speed of the hydrofoil is progressively increased from zero to U_{NACA} between $t = 0\,\text{s}$ and $t = T_f = 5\,\text{s}$ according to the speed law defined by (8) and plotted in Figure 8. Unsteady simulations are performed until steady states are reached. For all the transient waves produced as the foil starts from rest to be evacuated, 15 s of simulations are always performed so that the submerged hydrofoil runs a 10 m distance during a computation which corresponds to almost 50 chords.

$$\begin{aligned} &\forall t < T_f: U(t) = \frac{U_{\text{NACA}}}{2}\left[\sin\left(\pi\left(\frac{t}{T_f} - \frac{1}{2}\right)\right) + 1\right] \\ &\forall t \geqslant T_f: U(t) = U_{\text{NACA}} \end{aligned} \tag{8}$$

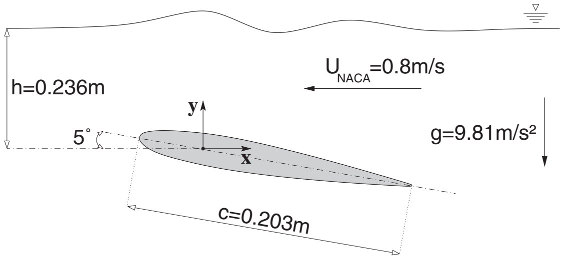


Figure 7. Configuration of the immersed hydrofoil.

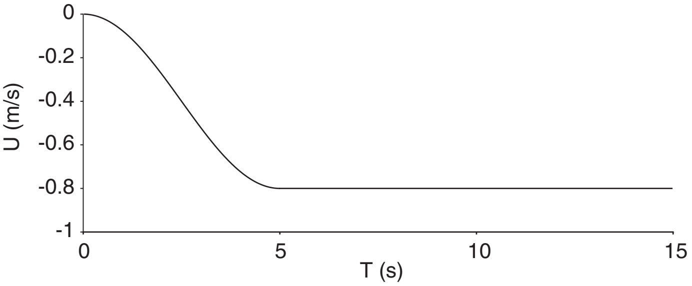


Figure 8. Imposed velocity of the hydrofoil.

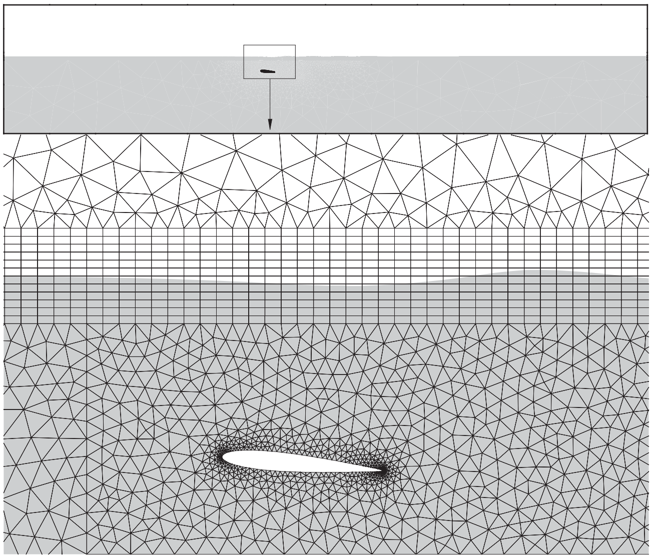


Figure 9. Grid 0 in the vicinity of the hydrofoil.

The moving body is handled easily since the whole computational grid is submitted to the same imposed displacement as the hydrofoil and moving grids are accounted for naturally solving equations presented in Section 2.2.

The considered computational domain is illustrated in Figure 9 along with a typical mesh. Hybrid grids are employed with a structured topology in a large band centred onto the interface and an unstructured topology made of triangular elements elsewhere. The structured region ensures, in the area of the free-surface, the presence of quadrangular elements which are known to be better suited regarding the capturing methodology [19]. Triangle elements are preferred in the far field and around the foil to reduce the number of points and to ease the mesh generation process.

At the left, right and bottom limits of the computational domain, the fluid velocity and the modified turbulent kinetic viscosity of the Spalart–Allmaras turbulence model ($\tilde{\nu}$) are imposed to zero through a Dirichlet boundary condition. The volume fraction c_i and the pressure are submitted to a Neumann boundary condition which is an imposed zero gradient flux. And, at the upper section, the fluid velocity and $\tilde{v}$ are allowed to be evacuated using a Neumann boundary condition. Furthermore, c_i is imposed to its exact value in the pure air phase and the pressure is fixed to zero which thus corresponds to the reference of pressure level. At the wall of the hydrofoil, classical no-slip boundary conditions are employed [19, 20].

For all the computations reported, the time step used, in the second-order accurate discrete scheme of the temporal terms involved, follows a similar time evolution than the imposed velocity of the hydrofoil. The corresponding law of the time step 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. Doing so, the discretization scheme (IGDS) for the concentration equation, presented in Section 2.3, is in the best configuration regarding its capturing capacities since the CFL condition is fully respected. Furthermore, the time step is automatically adapted to the current cell sizes of the grid.

It should also be noticed that, during each temporal loop, the non-linear/coupling loop (see Figure 6) ends when the residual of each single equation is reduced by three orders of magnitude.

First, a grid refinement study has been performed based on single-grid computations (no adaptation) using Grid 0–3 for which corresponding numbers of points are presented in Table I. Between two consecutive grids, a systematic isotropic refinement ratio of 2 is observed in the structured part of the domains. In the unstructured area, a similar rule applies but can not be ensured everywhere for obvious reasons linked to grid generation. As a consequence of the space refinement ratio considered, the typical time step (calculated at each step of the temporal loop according to a desired Courant number in the vicinity of the free-surface) for each single-grid computation is successively divided by two. As an illustration, Table I gives the time step used during the final constant speed part of the simulation for all the computations.

The free-surface elevations computed on these unadapted grids are compared to the experimental results of Duncan [18] in Figure 10. The quality of these results is similar to the one observed in Reference [19]. However, the relative difference between the free-surface elevations computed on the two finest grids (Grids 2 and 3) is small so that either of the two can be regarded as the mesh-independent solution. Thus, it will be considered as a reference solution to be compared to adaptive solutions in what follows.

Table I. Informations on the performed computations.

Computations	NCells	Δt (ms)	T_{CPU} (min)
Grid 0	5381	6	12
Grid 1	13 149	3	58
Grid 2	35 182	1.5	364
Grid 3	113 984	0.75	3001
NGen = 1	6900	3	35
NGen = 2	12 800	1.5	135

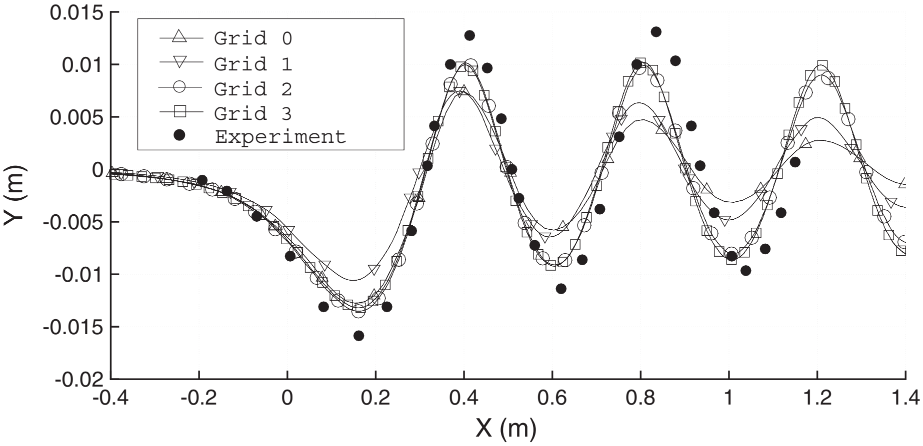


Figure 10. Free-surface elevations computed on unadapted grids.

4.1.2. Adaptive computations. For all the adaptive computations performed, the starting grid is always the coarsest one (Grid 0). Allowing a maximum of two generations of cells ($\mathsf{NGen} = 2$) leads to adapted grids with the same local cell size in the vicinity of the interface as on Grid 2. Similar features are observed between Grid 1 and the adapted grid based on Grid 0 and a maximum of refinement level of one. As an illustration, Figure 11 give a view of the considered meshes along with the typical free-surface computed on them.

On these figures, the interface corresponds to values of the volume fraction c_i between 0.01 and 0.99. It can be observed that the interface is diffused on typically three to four cells as expected and that the developed procedure successfully adapts meshes to the free-surface deformations computed. As explained, the typical local sizes in the vicinity of the discontinuity for (c) and (d) are equal. Thus, the smearings of the interface are similar.

In these figures, the safety margin applied has been set to $M = 0.03\,\mathrm{m}$ on either side of the discontinuity. However, as indicated in Section 3.3, the influence of the width of the safety margin may be not negligible at all and its correct definition is difficult to figure out *a priori*. Thus, four different adaptive computations have been examined using four different safety margins that range from $M = 0$ to $0.06\,\mathrm{m}$. The maximum number of generations allowed is set to $\mathsf{NGen} = 2$. The resulting adapted grids are presented in Figure 12. It can be observed that the biggest margin ($M = 0.06\,\mathrm{m}$) is very large compared to the amplitude of the waves generated. On the other hand, with no margin at all ($M = 0\,\mathrm{m}$) the interface is located close to the limits between two different levels of refinement. The free-surface elevations from single-grid computations, using the starting mesh Grid 0 and reference mesh Grid 2, are compared to the ones calculated from the four adaptive computations in Figure 13. Since values of the volume fraction c_i between 0 and 1 indicate the presence of a mixture, the value of 0.5 is selected as a definition of the interface. This remark applies throughout the remainder of this study. First and foremost, this figure reveals that the interface is captured as accurately as on Grid 2 using the present adaptive procedure even though the starting mesh Grid 0 leads to a far less accurate solution and a far more smeared interface (see also Figure 11). Thus, the developed methodology is shown to be relevant. However, the influence of the safety margin is

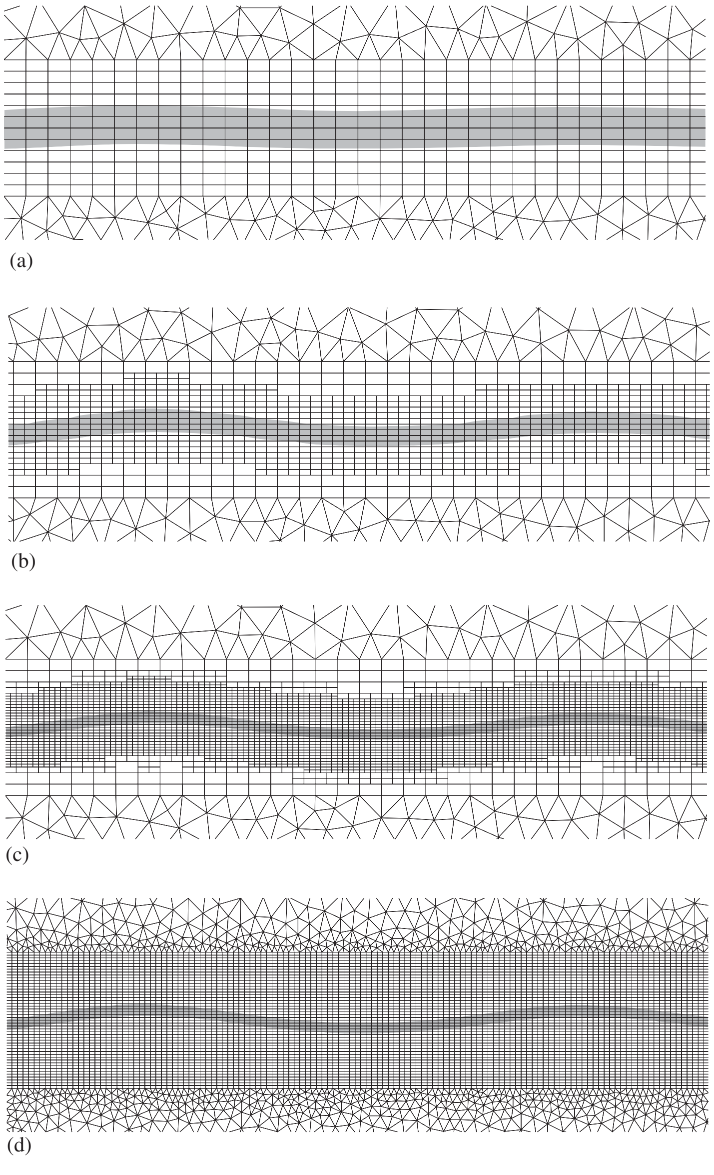


Figure 11. Comparison of grids in the vicinity of the free-surface: (a) Grid 0—$\mathsf{NGen}=0$; (b) Grid0—$\mathsf{NGen}=1$; (c) Grid 0—$\mathsf{NGen}=2$; and (d) Grid 2—$\mathsf{NGen}=0$.

almost negligible since all the adaptive solutions are of similar accuracy. Therefore, a small margin is preferable to bigger ones since it reduces the number of computational points. Therefore, the safety margin of $M=0.015\,\mathrm{m}$ is considered as optimum and informations for this adaptive computation with $\mathsf{NGen}=2$ are reported in Table I (also for $\mathsf{NGen}=1$). The computed adaptive solution has been shown to be as accurate as the solution calculated on Grid 2 but the overall computational cost is approximately divided by a factor three.

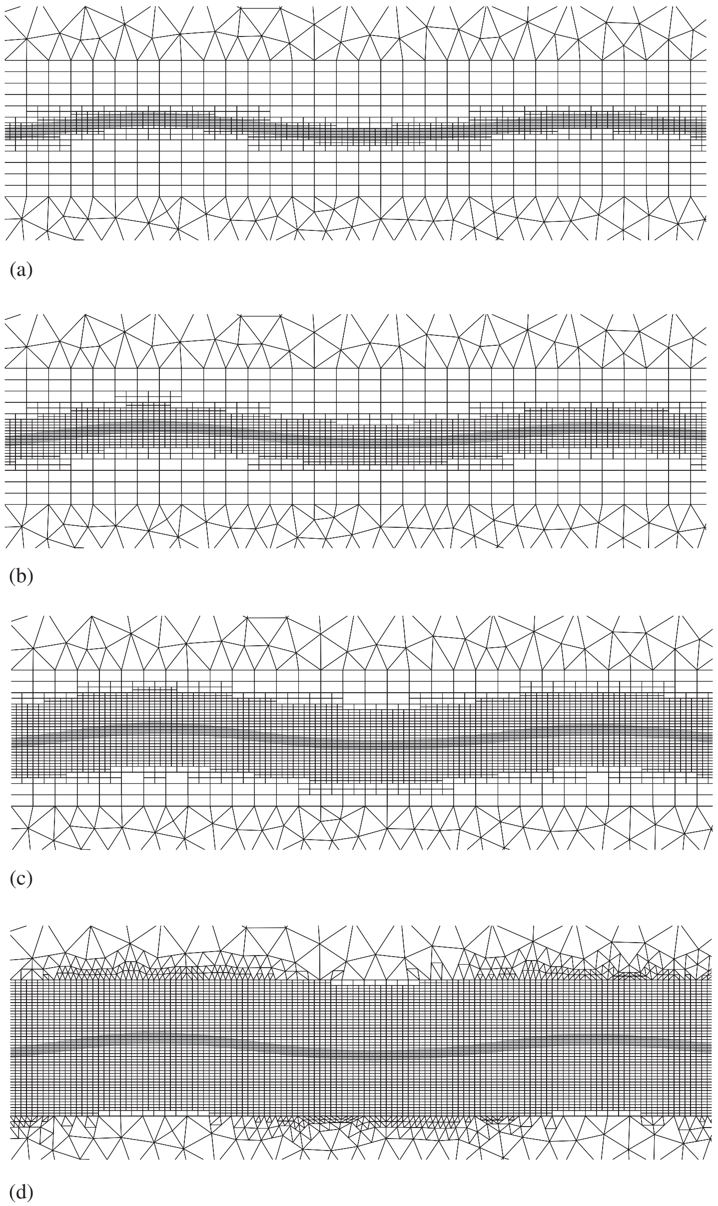


Figure 12. Adapted grids in the vicinity of the free-surface for different margins: (a) $M = 0\,\text{m}$; (b) $M = 0.015\,\text{m}$; (c) $M = 0.03\,\text{m}$; and (d) $M = 0.06\,\text{m}$.

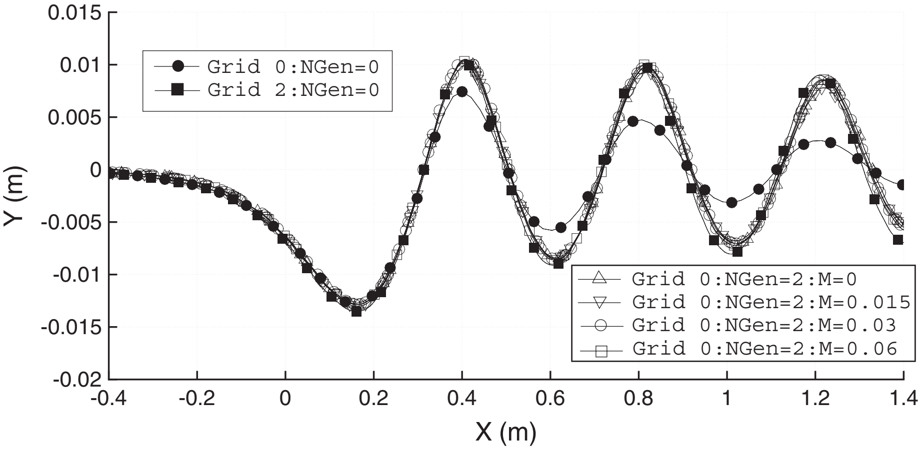


Figure 13. Comparison of free-surface elevations for different margins.

4.2. *Rayleigh–Taylor instability*

4.2.1. Description of the problem. This section is devoted to the treatment of the Rayleigh–Taylor instability problem using the proposed computational approach. When a heavy fluid is superimposed over a light fluid in a gravitational field, the fluid interface is unstable. Any perturbation of this interface tends to grow with time, producing the phenomenon known as Rayleigh–Taylor instability. It has been studied theorically and numerically in References [21–23]. The presently considered configuration is the one proposed by Daly [22] which is illustrated in Figure 14 and corresponds to a viscous case. At time $t = 0\,\mathrm{s}$, the interface is flat and coincides with $y = 0$. From this initial configuration, a perturbation of the velocity field is supplied. Following Daly's advice [22], the imposed perturbation is made appropriate for an incompressible flow calculation. As a consequence, a single wavelength perturbation is introduced through the following velocity field:

$$\begin{aligned} U(x,y) &= \frac{\pi A \delta y}{\lambda} \sin\left(\frac{2\pi x}{\lambda}\right)(2H(y)-1)\exp\left(\frac{-2\pi|y|}{\lambda}\right) \\ V(x,y) &= \frac{\pi A \delta y}{\lambda} \cos\left(\frac{2\pi x}{\lambda}\right)\exp\left(\frac{-2\pi|y|}{\lambda}\right) \end{aligned}$$

where A is the amplitude of the perturbation, $L = \lambda/2$ is the half wavelength of the perturbation corresponding to the width of the domain (see Figure 14), δy is the initial mesh spacing in the y direction and H is the Heavyside function defined as follows:

$$H(x) = \begin{cases} 1 & y > 0 \\ 0 & y < 0 \end{cases}$$

The parameters that fully defined the case studied and the imposed perturbation are presented in Table II. The two fluids considered are not water and air but fictitious fluids. However only the rate of densities of fluids is relevant and is presently equal to 2.

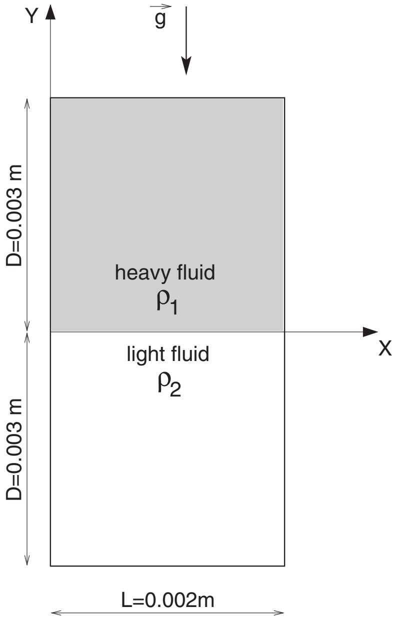


Figure 14. Initial configuration of the Rayleigh–Taylor instability.

Table II. Parameters of the Rayleigh–Taylor instability.

ρ_1/ρ_2	Re	Fr	A' $(\mathrm{m\,s^{-1}})$	λ
2	100	1	10^{-3}	$2L$

The Chandrasekhar's linear theory [21] permits to theorically determine the evolution of the free-surface at the early stage of the instability (time should be small enough for the non-linear effects to be negligible). For viscous fluids and given λ and Re, Chandrasekhar has noticed that the temporal evolution of the free-surface follows a positive exponential law which growth rate n is given by:

$$n = \sqrt{\frac{2g\pi}{\lambda}\left(\frac{\rho_2 - \rho_1}{\rho_2 + \rho_1}\right)} \tag{9}$$

The computational domain is depicted in Figure 14. On all the limits, slip wall boundary conditions are applied as in References [22, 23]. And, for each computation reported, the discretization time step used in the temporal loop (see Figure 6) is automatically defined by the desired Courant number that controls the compressivity of the discretization scheme IGDS presented in Section 2.3. That is, at the beginning of each temporal loop, the correct time step is calculated, from the current variables and based on the grid cell sizes, for ensuring a maximum Courant number of approximately 0.3 everywhere on the domain. Thus, whatever is the density of grid points of the current mesh, the time step is always defined in a way that respects the accuracy of the capturing scheme. And the non-linear/coupling convergence is reached (see Figure 6), at each step of the temporal loop, when the residual of each single equation considered is reduced by three orders of magnitude.

Before considering adaptive computations, a mesh refinement study has been performed using single-mesh calculations. The grids used are uniform, structured and their number of cells ranges from 520 to 133 120 as reported in Table III. The refinement ratio between two consecutive grids is 2. The calculated exponential growth rates using these uniform grids converge to a practically mesh independent value (see Table III) which is slightly different from the linear theory of Chandrasekhar but in good agreement with it.

4.2.2. Adaptive computations. For all the adaptive computations reported, the starting grid is always the coarsest uniform structured one with 520 cells. As demonstrated previously, the safety margin has no influence so that its width is fixed to a small value that is applied for all the adaptive calculations. Four adaptive computations have been performed with NGen ranging from 1 to 4.

The resulting grids after 1 s of simulation are shown in Figures 15(a)–(d). In the vicinity of the interface, the grid density of adapted grids is similar to one of the uniform grids presented previously since they have a refinement ratio of 2. As an example, the adapted grid with $\mathsf{NGen} = 3$ can be related to the uniform structured grid 321×105.

Figure 16 further give the temporal evolution of the computational grid for $\mathsf{NGen} = 3$. The location of the interface can be easily seen in these figures. The ability of the adaptive procedure to follow the temporal evolution of the free-surface is clearly demonstrated here. Besides, the capability of the whole numerical approach to compute wave breaking is also illustrated since the interface rolls up, yielding the half-mushroom pattern typical of this problem.

In the last figure, at time $t = 2$ s, it can be observed that there is still a thin layer of mixture of fluid close to the bottom boundary of the computational domain instead of pure heavy fluid phase ($c_i = 1$). Actually, the presence of the mixture (containing mostly the heavy fluid phase $0.95 < c_i < 1$) is due to the numerical treatment of the boundary condition. At this section, the volume fraction is submitted to a zero gradient flux condition (Neumann boundary condition) which is implemented explicitly in the flow solver. A delay in the volume fraction transport in

Table III. Informations on the performed computations.

		NCells	CPU time (s)	n
Linear theory		—	—	4.95
Single-mesh computations	41×14	520	159	6.97
	81×27	2080	816	7.48
	161×53	8320	3188	4.70
	321×105	33 280	25 878	5.27
	641×209	133 120	82 260	5.14
Adaptive computations	$\mathsf{NGen} = 1$	700–900	242	7.47
	$\mathsf{NGen} = 2$	1000–1400	417	4.70
	$\mathsf{NGen} = 3$	1600–2200	826	5.26
	$\mathsf{NGen} = 4$	2700–3700	2625	5.14

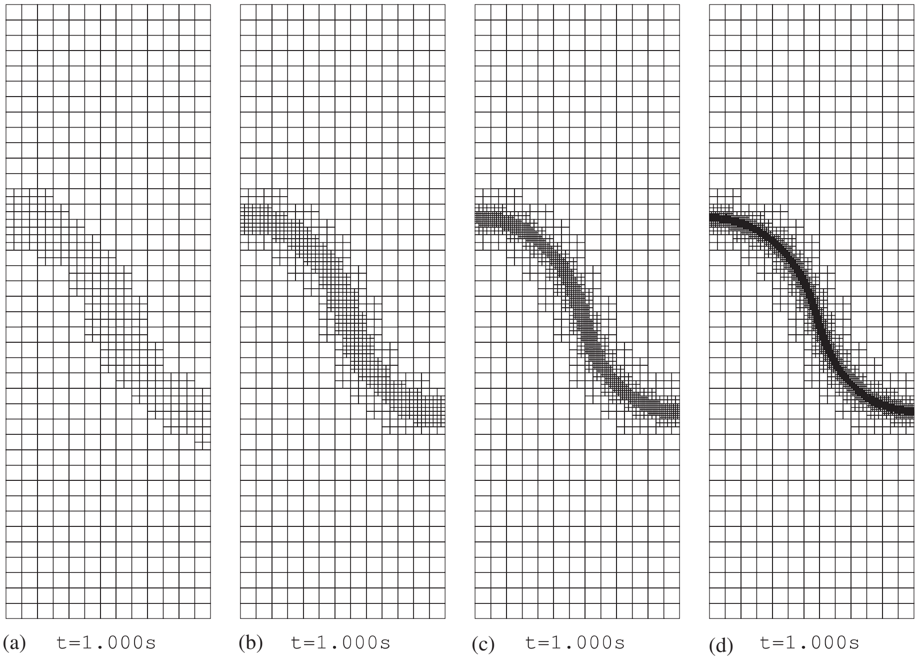


Figure 15. Comparison of grids for different values of NGen: (a) NGen = 1; (b) NGen = 2; (c) NGen = 3; and (d) NGen = 4.

the vicinity of boundaries is yielded by this explicit treatment of the Neumann condition but has little influence on the whole multi-phase flow since c_i is very close to its value in the pure heavy fluid phase. However, since the explicit indicator, which drives the adaptive procedure, is designed to identify the presence of mixture, refinement of the grids is performed in this part of the computational domain. The implicit treatment of the boundary condition should tackle this issue and is in development stage at present.

Figure 17 provides a comparison of the location of the free-surface computed for different values of NGen after 1 s of simulation. It can be observed that the computed positions of the interface converge when NGen is increased.

The computed growth rates of the interface n are now examined and compared in Table III. As for the mesh-independent study, the growing rates computed from adaptive computations converge to a value close to the one predicted by the analytical theory of Chandrasekhar. Moreover, the computed value from each single adaptive computation is almost equal to the value computed using the uniform grid that has the same grid density in the vicinity of the interface. For example, the growth rate from the adaptive computation with NGen = 3 is equal to the one computed using the uniform grid 321×105 so that these two calculations have a similar accuracy. Thus, as reported in Table III, for a similar accuracy, the adaptive procedure permits to significantly reduce CPU times. These reductions range from a factor 3 to a factor 30 comparing the CPU times for, respectively, the single-mesh computation on the finest uniform grid (641×209) and the adaptive computation with NGen = 4.

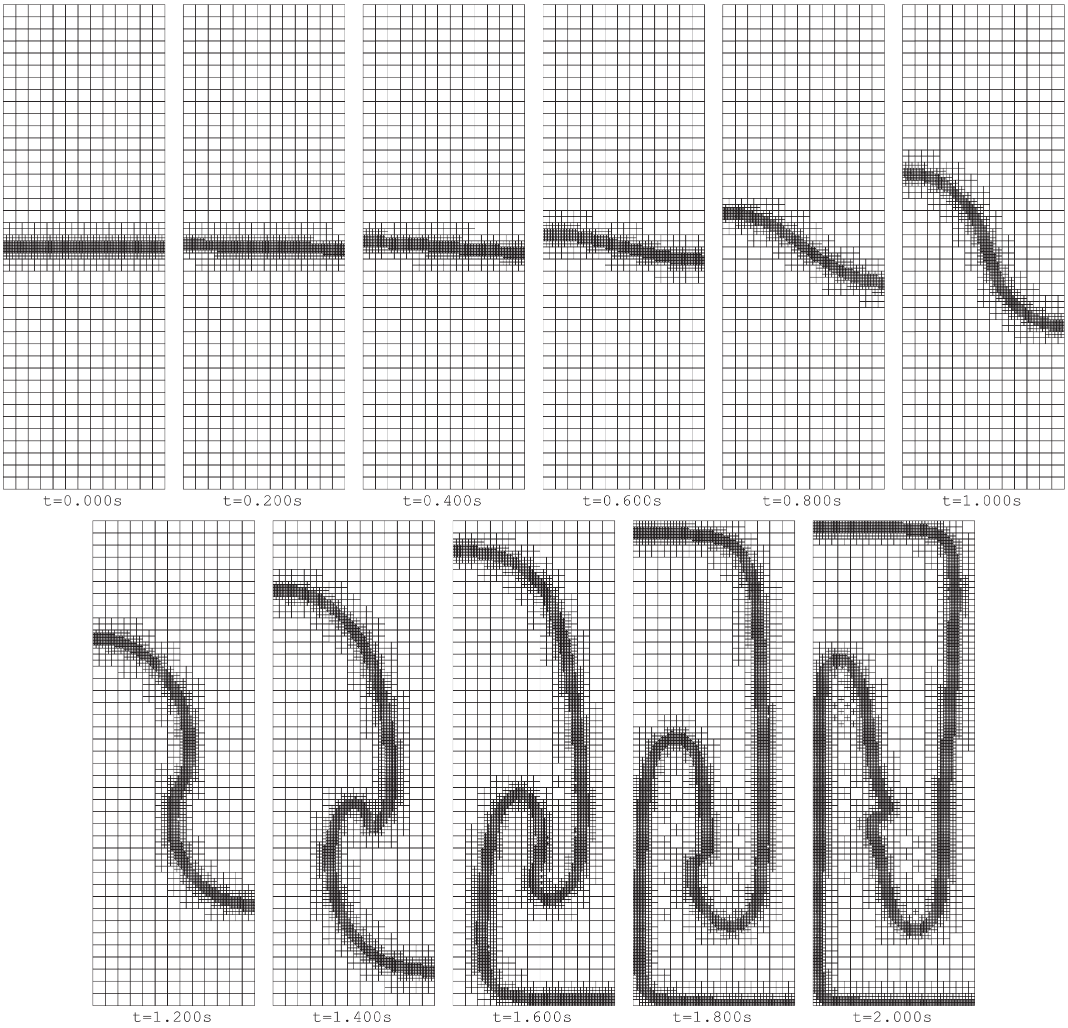


Figure 16. Temporal evolution of the computational grid ($\mathsf{NGen}=3$).

4.3. Free-surface around a Wigley hull

4.3.1. Description of the problem. The treatment of a fully three-dimensional hydrodynamic problem using the adaptive procedure is considered in the present section. The flow around the so-called Wigley ship has been studied experimentally in Reference [24] at model scale. The Wigley hull has a shape which corresponds to a parabolic model defined as follows:

$$Y(X,Z) = \pm\frac{B}{2}\left[1-\left(\frac{2X}{L_{\text{ref}}}\right)^2\right]\left[1-\left(\frac{Z}{D}\right)^2\right] \tag{10}$$

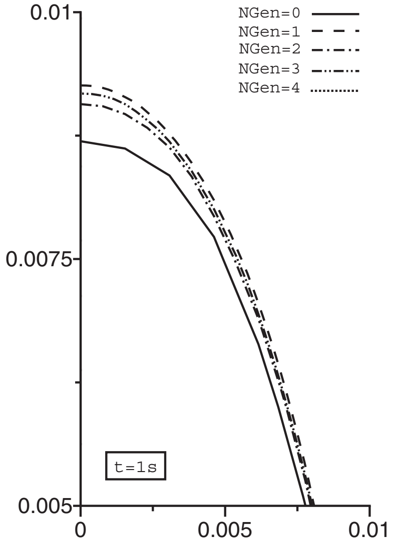


Figure 17. Comparison of free-surface computed for different values of NGen.

Table IV. Parameters of the flow around the Wigley hull.

Parameter	Dimensional value	Adimensional value
L_{ref}	4 m	1
B	0.4 m	0.1
D	0.25 m	0.0625
U_{ref}	1.81 m s^{-1}	1
g	9.81 m s^{-2}	11.973

where L_{ref} is the total length of the hull, B is the maximum thickness of the hull and D is the draft of the hull. This hull has been preferred to the more realistic one because its analytical definition facilitates the inclusion of new grid points on the surface of the body as explained in Section 3.2. Naturally, the two considered fluids are again water and air which have the previously given densities. Previous studies have shown that the viscosity of fluids can be neglected for this problem for a Froude number of $Fr = 0.289$. Furthermore, this experimental set-up corresponds to a steady free-surface deformation without any wave-breaking phenomenon [24]. The problem is considered in its adimensional form and the corresponding values of physical parameters are presented in Table IV. The adimensional domain of computation is depicted in Figure 18. Only half of the geometry is modelized since the plane $[(X,Z): Y/L_{\mathrm{ref}} = 0]$ including the boundaries of the hull is a symmetry plane. At rest, the interface is located in the plane $[(X,Y): Z/L_{\mathrm{ref}} = 0]$ and the bottom of the hull, which reduces to a segment, is included in the plane $[(X,Y): Z/L_{\mathrm{ref}} = D/L_{\mathrm{ref}}]$.

At the inflow and outflow sections (planes $[(Y,Z): X/L_{\mathrm{ref}} = -2.5]$ and $[(Y,Z): X/L_{\mathrm{ref}} = 3.5]$, respectively), at the bottom section $([(X,Y): Z/L_{\mathrm{ref}} = -2])$ and at the front section

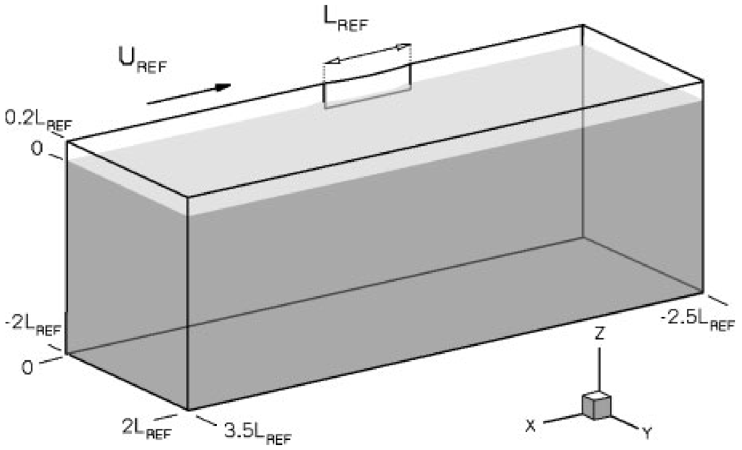


Figure 18. Global view of the computational domain.

($[(X, Z) : Y/L_{\text{ref}} = 2]$), the fluid velocity and the volume fraction are imposed and the pressure is submitted to a zero-gradient flux condition. At the top section ($[(X, Y) : Z/L_{\text{ref}} = 0.2]$), the pressure is imposed (which further defined the reference of pressure level) and the fluid velocity and the volume fraction are both derived by a zero-gradient flux condition. At the last limit of the computational domain, boundary conditions of symmetry are applied to all the variables excepted at the wall of the Wigley ship where slip wall conditions apply.

As for the hydrofoil test case, the velocity of the Wigley hull is progressively increased from zero to U_{ref} according to a speed law defined similar to (8). It avoids strong transient effects with unphysical waves as it could be observed if an impulsive start was prescribed. However, in order to get well converged results (a steady state is expected) and especially to satisfy the wave propagation in the far field, it is mandatory to compute up to an adimensional time of $t = 10$.

Once again, the discretization time step used in the temporal loop is automatically defined by the speed law followed by the ship, the desired Courant number and the mesh cell sizes. And, at each step of the temporal loop, the non-linear/coupling convergence is achieved by reducing the residual of each single equation considered by three orders of magnitude.

To evaluate the influence of the number of discretization points on the solution, a grid-independent study has been carried out on three grids composed of hexahedric volumes. Figure 19 gives an inside view of the typical grid close to the Wigley ship. The free-surface is initially located at the middle of the finest cells zone on the hull. The characteristics of the three grids used for performing single-grid computations are described in Table V with CPU time spent. The finest grid has about 1.8M cells. The calculation on this grid has been performed using a parallel approach on a multi-processor computer (see Section 5). The outcomes of the single-mesh computations will be presented and analysed along with the ones from adaptive computations for comparison. The solution on the finest grid will be considered as a reference solution.

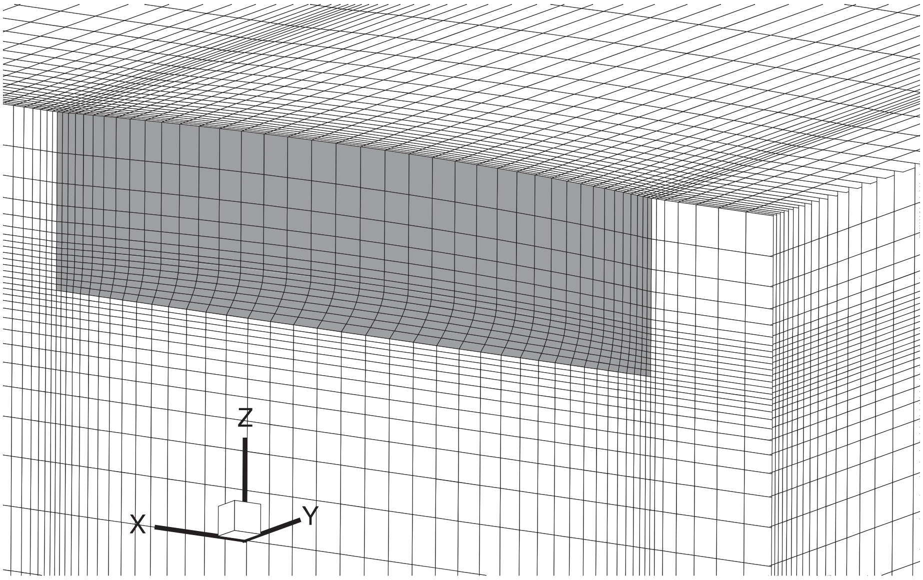


Figure 19. Inside view of a typical grid near the Wigley hull.

Table V. Characteristics of the computations of the free-surface around the Wigley hull.

Computation	NCells	T_{CPU} (h)
Coarse—NGen = 0	64 000	5
Medium—NGen = 0	306 000	77
Fine—NGen = 0	1 800 000	960
Coarse—NGen = 2	307 000	70

4.3.2. Adaptive computations. The coarsest grid presented previously, which has only 64 000 cells (i.e. 40 points in each direction), is the starting mesh for adaptive computations. It will be systematically adapted according to the explicit indicator based on the gradient of the concentration c_i and presented in Section 3.3. However, to avoid the inclusion of points in regions where the free-surface is hardly deformed, this criteria is activated only inside an area around the hull as illustrated in Figure 20. It shows the typical grid point concentration near the interface after two levels of adaptation. Figure 21 shows a transversal cut comparing the finest structured grid and the adapted grid with NGen = 2 generations allowed. It should be emphasized that the added nodes on the hull are correctly located according to the analytical definition of the hull (Equation (10)). However, the quality of the adapted grids is preserved without any special treatment since the slope of the hull is reduced and cells close to the slip wall boundary are not too stretched. Therefore, the grid remeshing technique, presented in Section 3.2, is useless for the present application. Besides, one can notice that the typical grid cell size of the adapted mesh is still larger than the corresponding size of the finest grid

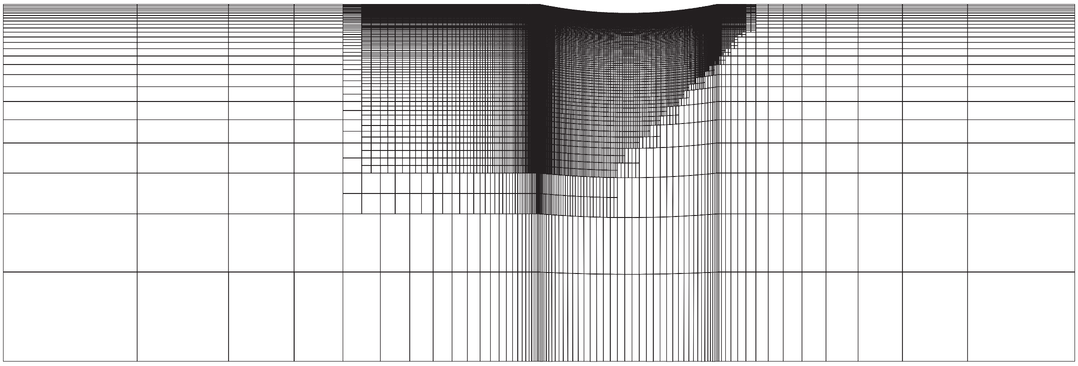


Figure 20. Global view of the region where the adaptation criteria is activated ($[(X, Y) : Z/L_{\text{ref}} = 0]$).

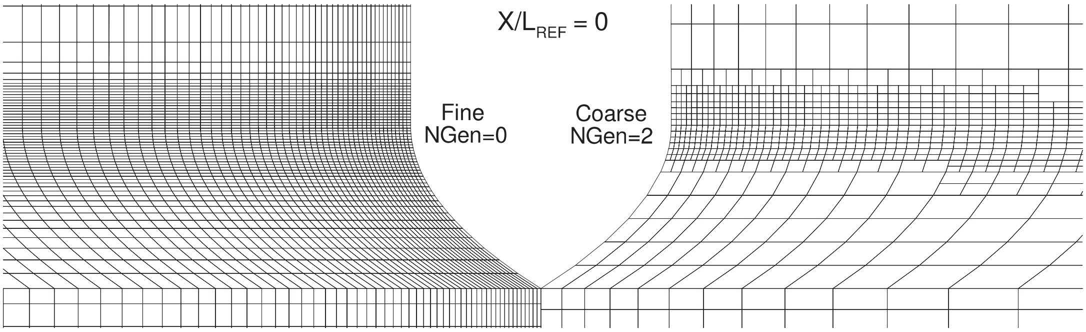


Figure 21. View of the structured and adapted grids ($[(Y, Z) : X/L_{\text{ref}} = 0]$).

involved in single-mesh computations. But, an adaptive computation with three generations of cell division is not affordable since it will require beyond the memory limit of the single-processor computers available. Such a computation can only be performed with reasonable CPU time cost by use of a suitable parallel strategy that is not yet coupled with the adaptive procedure. This issue will be discussed later in Section 5. Figure 22 shows the skin meshes of the hull on the coarse grid and the fine grid that can be further compared to the adapted meshes computed with, respectively, $\mathsf{NGen} = 1$ and 2. The coarse mesh (Figure 22(a)) has been strongly refined in the vicinity of the free-surface by the adaptive process (see Figure 22(c)).

The results obtained from the adaptive computation with $\mathsf{NGen} = 2$ are now compared to the ones obtained on (i) the fine structured grid made of 1.8M cells, (ii) the medium-structured grid having the same number of cells than the final adapted grid. The computational costs are listed in Table V. Obviously, the number of cells of the adaptive grids is not fixed during the whole calculation since the free-surface is moving. During the early stage of the computation, the interface is hardly deformed and thus the number of grid points is lower than the one given in Table V. This is why the adaptive computation has a lower CPU time that the single-grid computation performed on the medium grid. However, it should be noted that computing the error indicator, performing the adaptation of the grids and mapping the solutions have

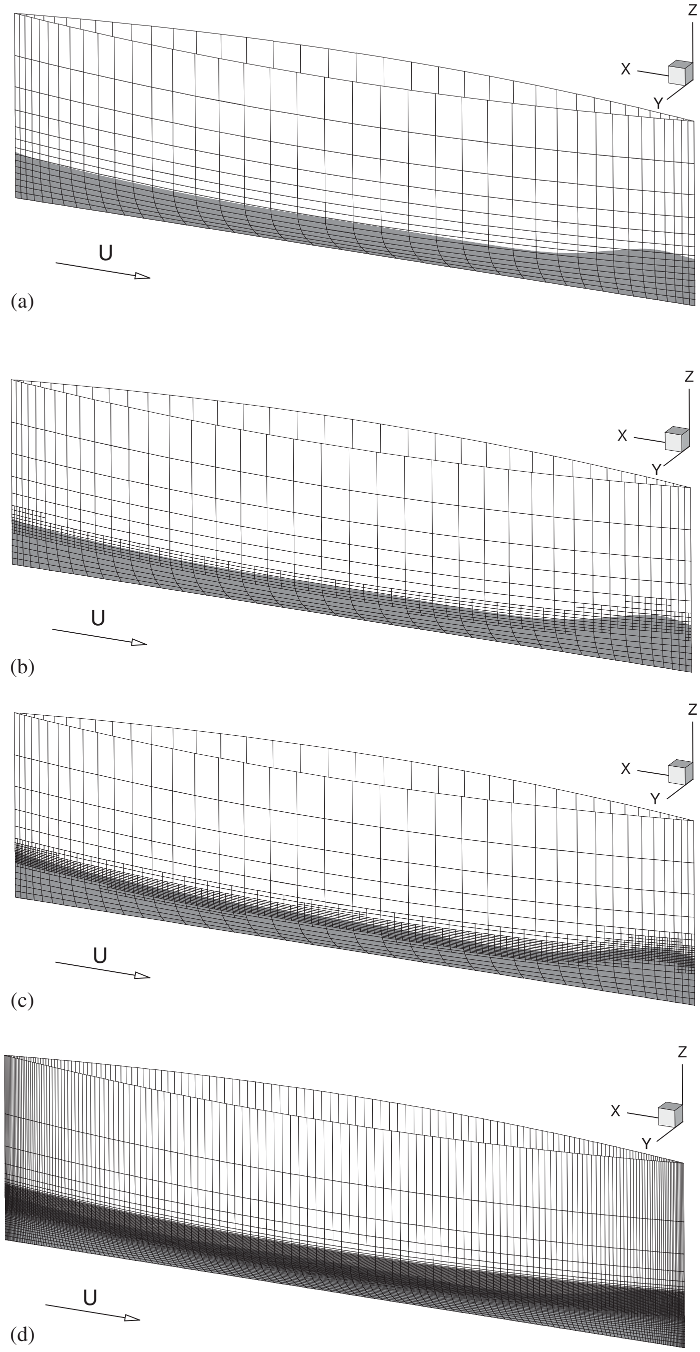


Figure 22. Adapted and unadapted meshes along the waterline: (a) Coarse—$\mathsf{NGen} = 0$; (b) Coarse—$\mathsf{NGen} = 1$; (c) Coarse—$\mathsf{NGen} = 2$; and (d) Fine—$\mathsf{NGen} = 0$.

a negligible CPU cost regarding the whole adaptive calculation. Actually, the main cost of the adaptive procedure comes from the requirement of performing a new convergence of the solutions at the times of adaptation (see Section 3.3).

First, the elevations on the waterline calculated by the different computations are considered and plotted in Figure 23 along with the experiments from Reference [24]. It can be noticed that the computed free-surface elevations are in good agreement with the experimental results. However, the accuracy of the adapted solution is similar to the fine grid prediction even though the solution computed on the initial coarse grid is far less accurate. Actually, the free-surface elevation computed on the medium grid is also less accurate than the adapted one compared to the fine grid solution.

A global view of computed free-surface elevations is now examined. The respective free-surface elevations are shown in Figure 24. Although the solution obtained on the adapted grid is not as accurate as the one provided on the fine structured grid, the interface is well captured since all the waves are visible although slightly attenuated. The influence of the right positioning of grid points is illustrated by the solution obtained on the medium-structured grid made of the same number of points as the adapted grid (roughly 300 000 cells), but distributed according to the criteria used to build the structured meshes. The free-surface elevation obtained on that medium-structured grid is obviously far too smeared. These results show how the adaptive procedure permits to reach a more accurate solution for a similar computational cost (see Table V).

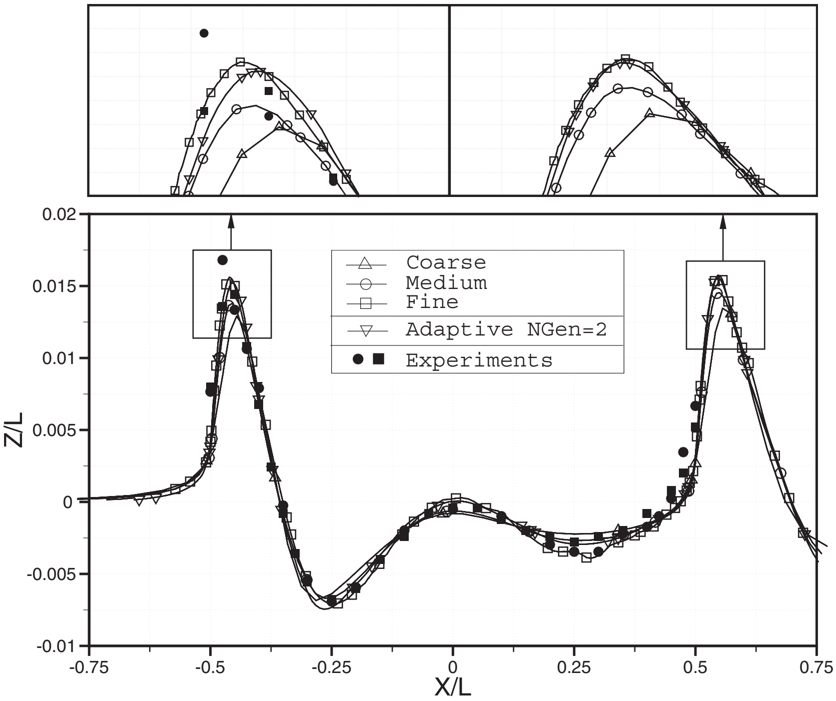
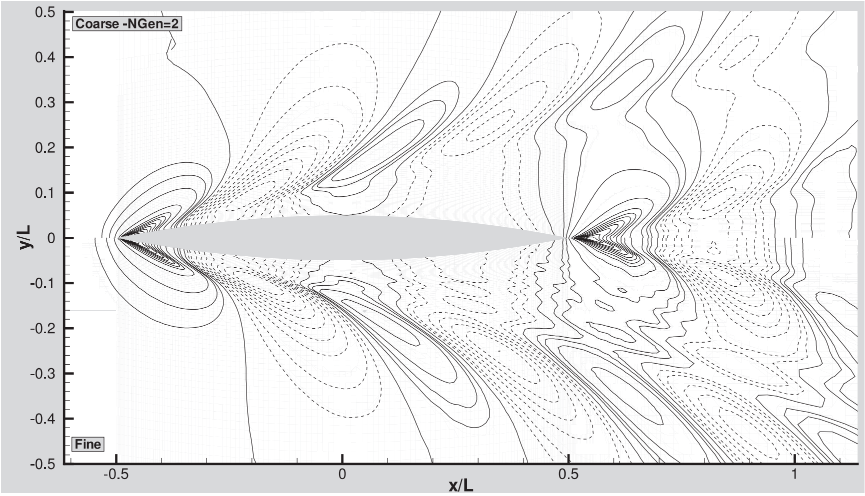
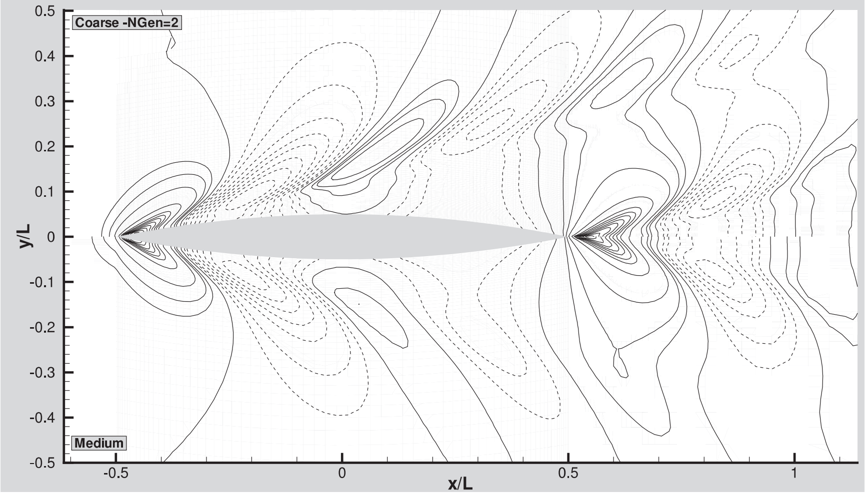


Figure 23. Free-surface elevation along the waterline for the different grids.



(a)



(b)

Figure 24. Comparisons of the free-surface elevations for different computations: (a) Adaptive $\mathsf{NGen} = 2$ (top) vs fine structured grid (bottom); and (b) Adaptive $\mathsf{NGen} = 2$ (top) vs medium-structured grid having the same number of points (bottom).

In order to assess more clearly this feature, Figures 25–27 show transversal wave profile cuts for $Y/L_{\mathrm{ref}} = 0.15$, 0.30 and 0.45, respectively. Considering the fine grid solution as a reference, it can be observed that the adaptive wavecuts are always more accurate than the ones computed using the medium grid. But, the adaptive solution, with the chosen number of allowed generations, is not always as accurate as the fine grid solution. As explained previously, one more level of refinement is not possible without a suitable parallelization

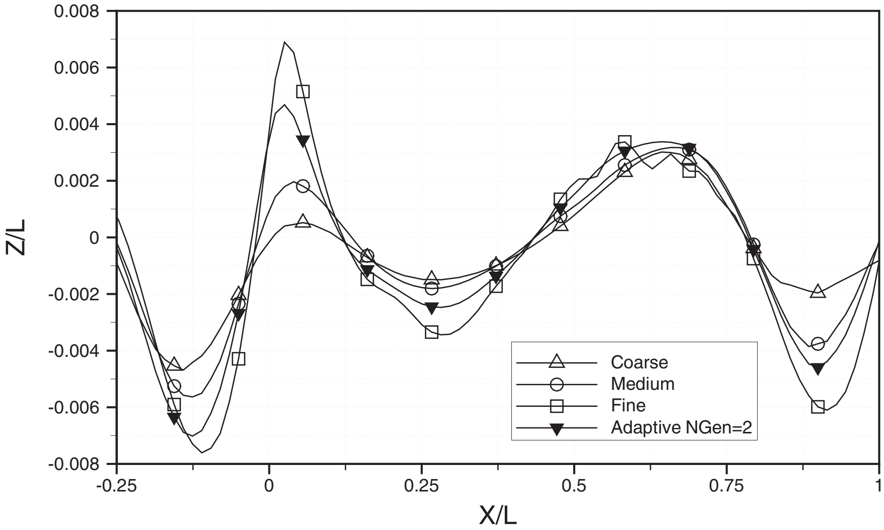


Figure 25. Transversal wave profile cut for $Y/L_{\mathrm{ref}} = 0.15$.

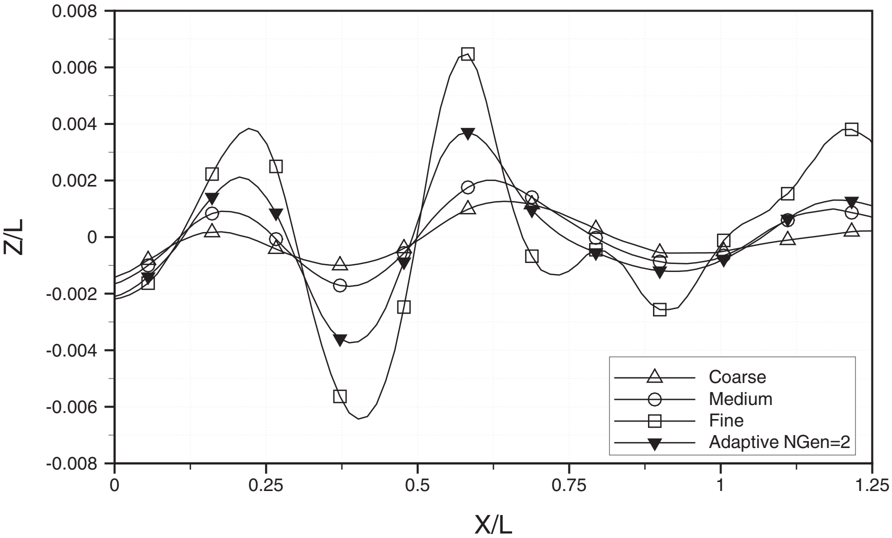


Figure 26. Transversal wave profile cut for $Y/L_{\mathrm{ref}} = 0.30$.

technique with dynamic load-balancing that may be difficult to derive and implement (see Section 5). Besides, looking at the starting grid of the adaptation process, one may fear a certain lack of grid points away from the interface where no adaptation is performed. Indeed, the adaptation procedure is concentrated on the free-surface. It thus permits to have

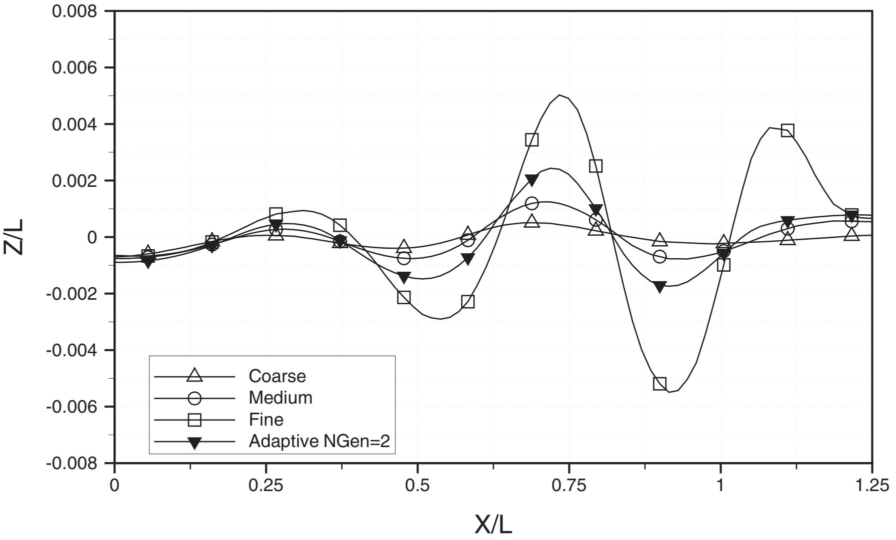


Figure 27. Transversal wave profile cut for $Y/L_{\mathrm{ref}} = 0.45$.

a description of the discontinuity as accurate as desired using a more and more important number of generations allowed (within the computer memory limit). But, this description of the interface is for a given pressure field evaluation that may suffer from the lack of grid points away from the interface. A careful investigation of this particular point is currently underway. However, for a similar overall cost, the adaptive procedure permits to compute more accurate solutions which is a very promising result.

5. CONCLUDING REMARKS AND FUTURE DEVELOPMENTS

This paper has described a modern free-surface capturing strategy implemented in an unstructured finite-volume viscous flow solver that can handle moving grids composed of arbitrary-shaped control volumes. A mesh adaptation strategy has been fully integrated in the code making it a single tool for performing adaptive computations. The data structure used, the grid alterations performed, the explicit indicator and the adaptation procedure have been detailed precisely.

The whole adaptive strategy, that only requires an initial grid from the user, has been applied to three different free-surface problems. First, the two-dimensional flow over a submerged hydrofoil has been considered. Then, the Rayleigh–Taylor instability problem has been examined. It has been demonstrated that interesting gains in terms of CPU time have been obtained using the adaptive strategy. These results have shown that an approach linking the use of unstructured adaptive grids with several levels of local refinement to a free-surface capturing strategy is interesting in terms of robustness and accuracy.

Then, the three-dimensional flow around the Wigley hull has been examined. The use of the adaptive procedure has yielded very encouraging results. It has been shown that more accurate solutions can be computed for a fixed overall cost. But, the accuracy of the fine grid solution cannot be completely reproduced by the adaptive computations with the chosen number of generations. Actually, three-dimensional computations require a substantial number of points even with the use of a suitable mesh adaptation strategy. Thus, future developments will aim at developing an efficient parallelization with a dynamic load-balancing. Actually, the adaptive procedure can be parallelized the same way as the flow solver. But, ensuring a dynamic load-balancing may be a more tedious task and this feature is mandatory for the parallelization of the adaptive procedure to be efficient. The ISIS code performs parallel computations using the message passing interface (MPI) [25]. When single-grid calculations are considered, the computational domain can be decomposed into a convenient number of equal-sized partitions. Each partition is treated by an attached physical processor. Thus, the load distribution across the processors of the parallel machine is naturally ensured which further leads to an efficient parallel processing. But, considering adaptive computations for which the number of grid points is dynamically modified, the load-balancing may no longer be respected. Indeed, since the computational domain is locally adapted, one partition of the decomposed domain may reach an important number of points when the other partitions are kept with no modification. Typical implementations of the MPI do not support this dynamic feature which is responsible for the loss of efficiency of the whole parallel process. However, the adaptive procedure can be parallelized using adaptive MPI (AMPI), an adaptive implementation of MPI, which is better suited for such applications. The basic idea behind AMPI is processor virtualization which is a powerful technique that enables the runtime system to carry out intelligent dynamic optimization like resource management [26]. The load is distributed over a large number of virtual processors, independent of the number of physical processors, and the runtime system is responsible for assigning virtual processors to physical processors. It has been shown in Reference [27] that treating a programme as a collection of communicating objects, measuring the execution time consumed by those objects at runtime, and achieving proper load-balancing by automatically moving those objects from processor to processor allows the implementation of efficient load balancers. Using such a technique is thus well-suited for *h*-adaptive computations that will no longer be limited in terms of maximum number of generations allowed.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the scientific committee of CINES (project dmn2050) and IDRIS (project 1308) for the attribution of CPU time.

REFERENCES

1. Larsson L, Stern F, Bertram V. Summary, conclusions and recommendations of the Gothenburg 2000 workshop. In *A Workshop on Numerical Ship Hydrodynamics*, Larsson L, Stern F, Bertram V (eds), Chalmers University of Technology, Göteborg, 2000.
2. Queutey P, Visonneau M, Ferrant P. Numerical investigation of wave interaction with a fixed vertical circular cylinder. *Proceeding of the 30th International Offshore and Polar Engineering Conference*, Honolulu, Hawai, 2003.
3. Queutey P, Visonneau M. 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, 2004.

4. Hay A. A study of numerical error estimation and local adaptive unstructured mesh refinement strategies for the Reynolds Averaged Navier–Stokes equations. *Ph.D. Thesis*, University of Nantes, 2004 (available online (in French) at `ftp://ftp.ec-nantes.fr/pub/DMN/Thesis/these_hay.ps.gz`).
5. Hay A, Visonneau M. Adaptive error control strategy: application to a turbulent flow. *Proceeding of the AIAA 16th Computational Fluid Dynamics Conference*, *AIAA Paper 2003-3848*, Orlando, Florida, 2003.
6. Spalart P, Allmaras S. A one-equation turbulence model for aerodynamic flows. *Proceeding of the AIAA 30th Aerospace Sciences Meeting*, *AIAA Paper 92-0439*, Reno, Nevada, 1992.
7. Menter F. Zonal two-equation k–ω turbulence models for aerodynamic flows. *Proceeding of the AIAA 24th Fluid Dynamics Conference*, *AIAA Paper 93-2906*, Orlando, Florida, 1993.
8. Deng G, Visonneau M. Comparison of explicit algebraic stress models and second-order turbulence closures for steady flow around ships. *7th Symposium on Numerical Ship Hydrodynamics*, Nantes, France, 1999.
9. Demirdzić I, Muzaferija S. Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology. *Computer Methods in Applied Mechanics and Engineering* 1995; **125**:235–255.
10. Ferziger J, Perić M. Further discussion of numerical error in CFD. *International Journal for Numerical Methods in Fluids* 1996; **23**:1263–1274.
11. Rhie CM, Chow WL. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. *AIAA Journal* 1983; **17**:1525–1532.
12. Jasak H, Weller HG, Gosman AD. High resolution NVD differencing scheme for arbitrarily unstructured meshes. *International Journal for Numerical Methods in Fluids* 1999; **31**:431–449.
13. Pržulj V, Basara B. Bounded convection schemes for unstructured grids. *Proceeding of the AIAA 15th Computational Fluid Dynamics Conference*, *AIAA Paper 2001-2593*, Anaheim, CA, 2001.
14. Jasak H, Weller HG. Interface tracking capabilities of the inter-gamma differencing scheme. *Internal Report*, Mechanical Engineering Department, Imperial College of Science, London, 1995.
15. Leonard BP. Simple high-accuracy resolution program for convective modelling of discontinuities. *International Journal for Numerical Methods in Fluids* 1988; **8**:1291–1318.
16. Gaskell PH, Lau AKC. Curvature compensated convective transport: SMART, a new boundedness preserving transport algorithm. *International Journal for Numerical Methods in Fluids* 1988; **8**:617–641.
17. Farhat C, Degand C, Koobus B, Lesoinne M. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computational Methods in Applied Mechanics and Engineering* 1998; **163**:231–245.
18. Duncan J. The breaking and non-breaking wave resistance of a two-dimensional hydrofoil. *Journal of Fluid Mechanics* 1983; **126**:507–520.
19. Deng G, Guilmineau E, Queutey P, Visonneau M. Interface capturing and interface tracking of incompressible and immiscible viscous flows. *Proceeding of the 8th International Conference on Numerical Ship Hydrodynamics*, Nantes, France, 2001.
20. Duvigneau R, Visonneau M. Hydrodynamic design using a derivative-free method. *Structural and Multidisciplinary Optimization* 2004; **28**:195–205.
21. Chandrasekhar S. *Hydrodynamics and Hydromagnetics Stability*. Oxford University Press: London, 1961.
22. Daly BJ. Numerical study of the two fluid Rayleigh–Taylor instability. *The Physics of Fluids* 1967; **10**:297–307.
23. Pan D, Chang CH. The capturing of free-surfaces in incompressible multi-fluid flows. *International Journal for Numerical Methods in Fluids* 2000; **33**:203–222.
24. Cooperative experiments on Wigley parabolic models in Japan. *Proceeding of the 9th R.C. International Towing Tank Conference* (*ITTC*), 1983.
25. MPI-2: extensions to the message-passing interface. *Technical Report*, University of Tennessee, Knoxville, TN, 1997 (can be downloaded at `http://www-unix.mcs.anl.gov/mpi/`).
26. Huang C, Lawlor O, Kalé LV. Adaptive MPI. *Proceeding of the 16th International Workshop on Languages and Compilers for Parallel Computing*, College Station, TX, 2003.
27. Brunner R. Versatile automatic load-balancing with migrable objects. *Ph.D. Thesis*, University of Illinois at Urbana-Champaign, 2000.