Creating Free-Surface Flow Grids with Automatic Grid Refinement

Jeroen Wackers, Ganbo Deng, Emmanuel Guilmineau, Alban Leroyer, Patrick Queutey, and Michel Visonneau

Abstract The objective of this work is to create grids for free-surface water flow simulation entirely with automatic grid refinement. It is shown why it is necessary to refine the mesh iteratively as the solution converges and why refinement and derefinement of hexahedral cells must be treated anisotropically.

The proposed refinement criterion is a combination of the pressure Hessian with refinement at the free surface, in order to capture the flow which drives the surface motion and the position of the surface itself. Smoothing is needed in the computation of the Hessian in order to remove oscillations in the pressure, the pressure Hessian is extrapolated through the free surface to remove its discontinuity there.

Two test cases confirm that effective fine meshes for wave computation can be created with the proposed automatic refinement procedure.

1 Introduction

Free-surface flows with gravity waves, such as the water flow around ships and offshore structures, are created through the interaction between the turbulent viscous flows on both sides of the interface and the motion of the interface. To simulate such flows, the position of the free surface needs to be modelled as well as the velocity and pressure fields. An attractive and very robust model is the surface capturing mixture-fluid approach [8]. Here, the entire fluid is modelled as a numerical mixture of the pure fluids on the two sides of the interface. The proportion of both fluids in the mixture is computed with a convection equation for the volume fraction of one of them, having a discontinuous inflow condition. This discontinuity is convected through the flow, implicitly giving the position of the free surface without any specific treatment of the surface region (as opposed to, for example,

J. Wackers (✉) • G. Deng • E. Guilmineau • A. Leroyer • P. Queutey • M. Visonneau
Laboratoire de recherche en Hydrodynamique, Energétique et Environnement Atmosphérique, CNRS UMR 6598, Ecole Centrale de Nantes, 1 rue de la Noë, B.P. 92101, 44321 Nantes Cedex 03, France
e-mail: jeroen.wackers@ec-nantes.fr

S. Perotto, L. Formaggia (eds.), *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, SEMA SIMAI Springer Series 5,
DOI 10.1007/978-3-319-06053-8_15

the Volume-of-Fluid or Level-Set approaches). However, for accuracy, special care must be taken to keep the numerical interface sharp.

Adaptive local grid refinement is particularly well suited for these simulations. Free-surface water flows have many features which are local in nature and whose exact position is difficult to estimate beforehand, so their precision can be increased with adaptive grid refinement. For example, refinement around the surface strongly increases the resolution of the volume fraction equation, so the modeling of the free surface is improved. But also other aspects of these flows, such as wakes and trailing vortices, are resolved with greater precision when grid refinement is used.

The unstructured Reynolds-averaged Navier-Stokes solver ISIS-CFD which we develop contains an automatic grid refinement method [10–13]. This flow solver, distributed by NUMECA Int. as part of the FINE/Marine software, is aimed at the simulation of realistic flow problems in all branches of marine hydrodynamics. Therefore, the grid refinement method is general and flexible, featuring anisotropic refinement on unstructured hexahedral grids, derefinement of previous refinements to enable unsteady flow computation, and full parallelisation including integrated dynamic load balancing. The anisotropic refinement is based on metric tensors. To our knowledge, it is the first grid adaptation method included with success in a general-purpose hydrodynamic flow solver.

In our earlier work on grid refinement for free-surface flows, the multiphysics character of the flows was not explicitly taken into account for the grid refinement. Instead, the original grid was chosen sufficiently fine to get a reasonable resolution of the flow, then automatic grid refinement was used to improve the accuracy of one particular flow feature. Thus, gravity waves at the water surface were computed with refinement based on the discontinuity in the volume fraction [12] and wake flows with refinement based on the pressure [10, 13].

The objective of this paper is to go beyond these earlier works and create fine meshes for free-surface flow simulation entirely with adaptive grid refinement. This removes the need for original grids with finer cells in all the possible positions of the water surface and other important flow features, that are difficult to generate and very costly if strong waves appear or if a simulated object is free to move. The new refinement approach simplifies the mesh generation for users and can be much more efficient, since all fine cells are placed only there where they are really needed.

For such refinement, the technical part of the algorithm has to be modified to treat both refinement and derefinement of cells in an anisotropic way, allowing division or merging of cells in one direction only. The first part of the paper highlights the aspects of the existing algorithm that are essential for free-surface flows, such as the need to continuously modify the refined mesh as the solution converges, and shows why anisotropic derefinement must be introduced. The most important evolution however is in the refinement criterion, which must be a combination of different physical sensors due to the multiphysics character of two-fluid flows. A criterion is proposed which combines refinement at the free surface with a pressure Hessian criterion, modified to remove the singularity in the pressure gradient at the surface.

The paper is organised as follows. Section 2 describes the ISIS-CFD flow solver and the meshes that we use. Section 3 then discusses the aspects of its grid refinement method which are relevant for free-surface flow. Section 4 looks at anisotropic derefinement, which is currently under development. The combined refinement criterion is introduced in Sect. 5. Finally, Sect. 6 shows a test on two ship flow cases, with a new case which highlights the interest of grid refinement for a flow containing both free surface and strong vortices.

2 Solver and Meshes

ISIS-CFD is an incompressible unsteady Reynolds-averaged Navier-Stokes method. The solver is based on the finite-volume method to build the spatial discretisation of the transport equations. The velocity field is obtained from the momentum conservation equations and the pressure field is extracted from the mass conservation constraint transformed into a pressure equation. These equations are similar to the Rhie and Chow SIMPLE method [9], but have been adapted for flows with discontinuous density fields. As mentioned, free-surface flow is simulated with a mixture flow approach: the water surface is captured with a conservation equation for the volume fraction of water, discretised with specific compressive discretisation schemes. A detailed description of the solver is given by [3, 8].

The unstructured discretisation is face-based: fluxes are computed face by face, the reconstructions of the cell-centred state variables to the face centres are made with interpolations that use the two cells next to a face and their neighbour cells without a-priori assumptions about the cell topologies. And while the linearised systems used to solve the momentum and pressure equations are formulated in the cell-centred unknowns, these systems are constructed by summing the contributions of the faces to each cell. Thus, no cell topology assumptions are made anywhere, which means that cells with any number of arbitrarily shaped faces are accepted.

The solver is mostly used with unstructured hexahedral grids generated by the HEXPRESS grid generator which is also part of FINE/Marine. The grid in Fig. 1 shows the typical features of such meshes: several semi-structured regions, with body-fitted boundary grids near the walls in order to ensure the best possible grid quality in the boundary. The grid consists purely of hexahedral cells, with mesh size variations obtained by placing one large cell next to two or four smaller neighbour cells. Due to its face-based nature, the ISIS-CFD solver treats these grids just the same as any other type of mesh. With these meshes, good solution accuracy is obtained from the semi-structured parts and the body-fitted boundary meshes, combined with the flexibility to mesh complex geometries.

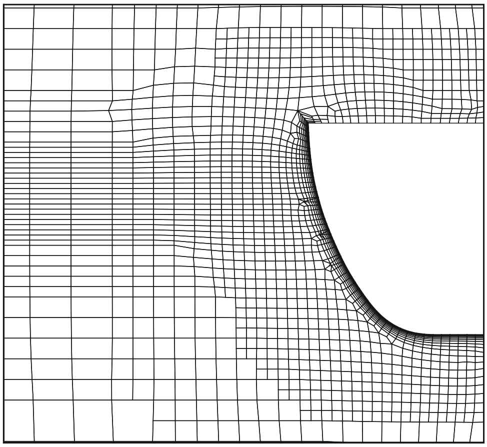

Fig. 1 Cut through an unstructured hexahedral mesh for a ship geometry. The objective of the present work is to generate the fine grid around the water surface with adaptive refinement

3 Grid Refinement Method

The natural method of grid adaptation for unstructured pure-hexahedral meshes is local grid refinement by dividing cells into smaller cells, as this is how the original grids themselves are constructed. Thus, adaptively refined grids are of the same type as all other grids so the flow solver can use them without modifications. This section shows our existing grid refinement method and how refinement criteria in a metric context are used to pilot the refinement, focusing on those aspects which are relevant for free-surface flows. More details of the metric criteria are found in [12].

3.1 Refinement Algorithm

For free-surface flows the simulated position of flow features often depends strongly on the level of grid refinement; for example, waves may become steeper when the flow is resolved on finer and finer grids. This means either that automatic grid refinement should be applied in a large buffer zone around the flow features of interest to take into account their displacement once the grid is refined, or that the refinement procedure must be iterative, continuously changing the grid as the solution converges. Thus, in each refinement iteration, new cells must be refined if the features of interest have moved, while previous refinements of other cells may need to be undone in positions which the features of interest have left. For such an approach, the derefinement of previously refined cells is a necessity.

Our refinement algorithm takes this second approach. ISIS-CFD computes both steady and unsteady flows with a time integration technique. For grid refinement, after a given number of time steps (usually 25 for steady flows and 2–4 for unsteady flows), the grid is adapted to the current solution, after which the time integration continues until the grid is adapted again. For steady flows, this procedure eventually converges (typically after 40–50 refinement cycles). If the refinement criterion (Sect. 3.4) indicates that the grid is well adapted to the flow and the flow solution itself has converged, then the refinement procedure will no longer modify the mesh.

3.2 *Anisotropy*

Anisotropic refinement is the possibility to divide a hexahedral cell in its three directions independently, either in two, four or eight smaller cells. This is essential for our refinement procedure, since isotropic refinement (division in eight cells only) is much too costly in three dimensions if very fine cells are needed to accurately resolve a local flow feature. By applying anisotropic refinement for features that require a fine grid in only one direction (notably, the water surface!), the total number of cells required can be greatly reduced, or much finer flow details can be resolved.

A second reason for directional refinement is, that the original grids (Fig. 1) already contain anisotropic refinement with cells of different aspect ratios lying side by side. Therefore, when refining, we need to control the size of the fine cells in all their directions independently, otherwise refined grids may have smoothly varying sizes in one direction, but repeated changes from fine to coarse and back to fine in another. Isotropic refinement cannot prevent this (see Fig. 2 for an example), so directional refinement is the mandatory choice.

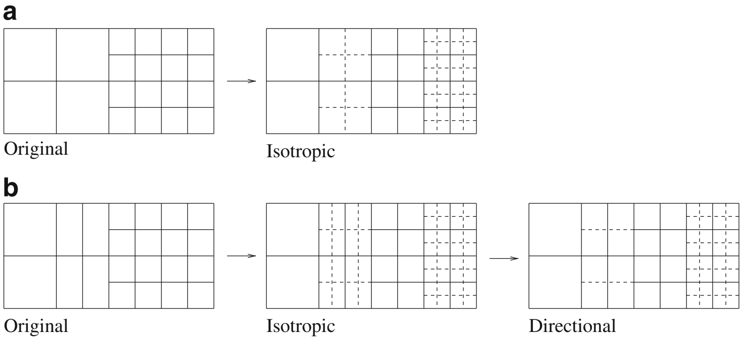

Fig. 2 Isotropic grid refinement is satisfactory on an original grid where all cells have the same aspect ratios (**a**), otherwise directional refinement is needed (**b**). (**a**) Locally isotropic grid; (**b**) Locally anisotropic grid

3.3 Quality of Refined Cells

Since the cells in the refined grid are formed by the subdivision of the cells in the original grid, the shape of the refined cells is entirely determined by the original grid. If these original cells are close to rectangular, then the refined cells will be rectangular as well (this is the case in Fig. 2). However, the quality of bad original cells is deteriorated by grid refinement. Especially if an original cell is arrow-shaped, its subdivision may result in cells that are turned inside out.

Thus, the quality of the refined grids can be assured by making sure that the original grid is as regular as possible. In HEXPRESS meshes, bad quality cells only appear near the surface of objects, typically at inward-facing angles in the geometry. Usually, bad original grids can be prevented by locally imposing fine original cells near these surface features. This is not contrary to the idea of automatic grid refinement; if the original grid is made to capture the geometry well, the rest of the fine grid can be created by automatic refinement.

3.4 Metric-Based Refinement Criteria

The use of metric tensors as refinement criteria allows us to specify different cell sizes in different directions. This technique was first developed for the generation and refinement of unstructured tetrahedral meshes [4]. It is also an extremely useful and flexible framework for the anisotropic refinement of hexahedral meshes.

In the metric context, the refinement criterion is a smoothly varying tensor field whose values at every point in the flow domain indicate what the ideal size for a cell in that position would be. In each cell, the criterion is a 3×3 symmetric positive definite matrix $\mathscr{C}$, which is interpreted as a geometric transformation of the cell in the physical space to a deformed space. The refinement of the cells is decided as follows. Let the criterion tensors $\mathscr{C}$ in each cell be known (their computation from the flow solution is described in Sect. 5). In each hexahedral cell, the cell sizes $\mathbf{d}_j$ $(j = 1, 2, 3)$, which are the vectors between the opposing face centres in the three cell directions, are determined. Next, the modified sizes are computed as:

$$\tilde{\mathbf{d}}_j = \mathscr{C}\mathbf{d}_j. \tag{1}$$

Finally, a cell is refined in the direction j when the modified size exceeds a given, constant threshold value T_r:

$$\|\tilde{\mathbf{d}}_j\| \geq T_r, \tag{2}$$

while a previously refined group of cells is derefined back into one cell if:

$$\|\tilde{\mathbf{d}}_j\| \leq T_r/d \qquad \forall j = 1 \cdots 3, \tag{3}$$

for all cells in the group. d is chosen slightly larger than 2, to prevent cells being alternately derefined and re-refined (because $\tilde{\mathbf{d}}_j$ doubles when cells are derefined). Since the procedure only uses the vectors $\mathbf{d}_j$ to characterise a cell, it can be used for any shapes and sizes of cells.

The objective of the refinement is thus to create a uniform grid in the deformed space. The tensors $\mathscr{C}$ are direct specifications of the desired cell sizes: in a converged refined grid, the cell sizes are inversely proportional to the magnitude of the $\mathscr{C}$. The threshold T_r functions as a global specification of the fineness of the grid; sensible choices for T_r in different situations are discussed in Sect. 6 (see also Sect. 5.3).

4 The Need for Directional Derefinement

The derefinement of previously refined cells, like the refinement, ought to be treated in an anisotropic way. Until now, the refinement of a cell is decided separately for each direction [Eq. (2)] but the derefinement only allows the complete undoing of a refinement step; if a cell was divided in eight, the eight cells have to be derefined back into one which requires the criterion (3) to be satisfied in all three directions j. This section shows why this limitation is much too restrictive and how we will modify our method to anisotropic derefinement. As the transformation of an existing grid refinement method like ours is difficult compared with developing it correctly from the beginning, this section may serve as a warning to others....

For free-surface flows, isotropic derefinement is inefficient because cells refined initially in different directions often need to remain refined in at least one direction. A typical example is the simulation of a travelling wave with grid refinement around the free surface (see Sect. 5.1). When the wave passes, the free surface is inclined so the grid is refined both in horizontal and in vertical direction. Then after the wave has passed and the free surface has returned to its rest state, the vertical refinement must remain in place. Thus, with isotropic derefinement, the horizontal refinement cannot be removed either. This leads to large clusters of unnecessary fine cells and the more the grid is refined, the more this is evident. Since the creation of free-surface grids entirely with automatic refinement requires much refinement of the original cells, this problem has to be solved.

A situation which shows the problem of isotropic derefinement particularly well is the initialisation of the refined grid. Usually the mesh is adapted to the undisturbed position of the free surface before starting the simulation, to have a good initial resolution of the volume fraction. When the original grid contains no specific refinement at the free surface, it has varying cell sizes at the surface position (Fig. 3a): the grid near a solid wall is much finer than elsewhere. Thus, the discontinuity in the volume fraction is thinner near solid walls than far away, which means that its top and bottom are inclined! This, in turn, leads to horizontal refinement of cells. At the end of the initialisation, when the vertical cell sizes around the surface have become equal due to the automatic refinement,

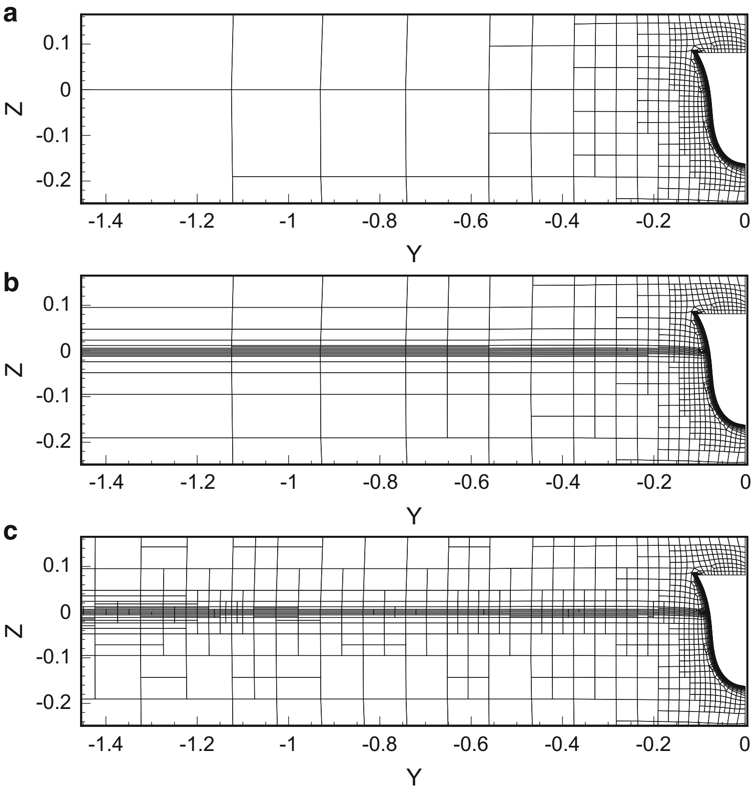

Fig. 3 Initialisation of the free-surface refined grid from the original grid (**a**): as it should be (**b**) and with only isotropic derefinement (**c**). The middle figure was created artificially by initialising without allowing refinement in x- and y-directions

this horizontal refinement is no longer required (Fig. 3b). However, with isotropic derefinement it cannot go away because the vertical refinement must remain in place (Fig. 3c).

Thus, for efficient derefinement it is necessary to remove previous refinements in one direction only (in Fig. 3 the horizontal direction), for example by changing a cell divided in four into one divided in two. The problem that we have is that our data structure is not suited for this, since the history of the refinement is stored in the refined cells as ‘mother’ pointers towards the cells that were divided and ‘sister’ pointers to one of the other refined cells, forming a loop. This structure does not preserve the relative position of the fine cells. The solution that we are working on consists of adaptively modifying the refinement history after the fact in order to suit the required directional derefinement. Clearly, it would be much more elegant to work with a suitable data structure, for example one which directly indicates the position of a small cell within the original cell.

5 Combined Criteria

Suitable refinement criteria for free-surface flows must take into account gravity waves, which propagate through a cyclic exchange of potential (gravity) and kinetic energy, caused by the interaction of the free-surface motion with the pressure and velocity fields below the surface. So in order to correctly simulate free-surface flows, a good resolution is needed of the volume fraction equation which gives the position of the free surface, as well as the pressure and velocity variations below the surface.

A grid refinement criterion for free-surface wave simulation must therefore be based both on the pressure and velocity field and on the volume fraction. For these two, different indicators are used because the volume fraction α is discontinuous at the free surface and constant everywhere else, while the pressure and the velocity are smooth in the whole flow field except at the surface. Spatial derivative based error indicators can identify the regions of importance for the flow field below the surface, but do not work at the water surface itself since any derivative of α goes to infinity when the grid is refined. Also, the grid specified by the criterion must be as uniform as possible near the surface, since transitions from fine to coarse cells lead to large errors in the volume fraction. Basing a criterion only on the value of α itself gives the most stable values for the criterion and thus the best meshes.

This section describes a criterion which is based on the Hessian of the pressure, combined with refinement where α is neither 0 nor 1. These criteria are first introduced individually, then we indicate how they are combined. The section is a continuation of the work in [11].

5.1 Free-Surface Criterion

To resolve accurately the volume fraction field α which is a discontinuity that is convected with the flow, it is sufficient to refine the grid at and around the free surface, in the direction normal to the surface. When the surface is locally aligned with the cell directions, anisotropic refinement can be used to keep the total number of cells as low as possible. It is important that cells locally have the same size, to prevent distortions of the volume fraction.

The free-surface criterion $\mathscr{C}_S$ is non-zero when α is neither 0 nor 1. The normal direction to the surface is computed from a field α_s which corresponds to α, smoothed out by averaging over a cell and its neighbours a given number of times. The gradient of this field gives the normal directions. The criterion is then derived from vectors $\mathbf{v}_\alpha$ in each cell which are unit vectors in this normal direction for those cells where the smoothed α_s field is non-zero:

$$\mathbf{v}_\alpha = \begin{cases} \frac{\nabla \alpha_s}{\|\nabla \alpha_s\|} & \text{if } 0.1 \le \alpha_s \le 0.9, \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{4}$$

Using the smoothed field guarantees that the normals are well defined and also that the mesh is refined in a certain zone around the surface to create a margin of safety.

In tensor form, the free-surface criterion is computed as matrices having only one non-zero eigenvalue, associated with the direction of the vector $\mathbf{v}_\alpha$. The tensors $\mathscr{C}_S$ are computed as follows (with $\otimes$ representing the tensor product):

$$\mathscr{C}_S = \mathbf{v}_\alpha \otimes \mathbf{v}_\alpha. \tag{5}$$

In the directions normal to the vector $\mathbf{v}_\alpha$, the eigenvalues are zero. This implies a modified cell size of zero [Eq. (1)]. As a consequence, the grid is not refined in these directions. Since the $\mathbf{v}_\alpha$ are unit vectors, the only non-zero eigenvalues of $\mathscr{C}_S$ are equal to 1. Thus, Eq. (2) shows that the threshold value T_r directly indicates the desired cell size at the surface. Also, the specified cell size normal to the surface is *exactly* the same in all surface cells, as required. The free-surface criterion has been used on its own, with good results, in our earlier work [10, 12].

5.2 Computing the Pressure Hessian

The Hessian matrix of second spatial derivatives can be linked to the interpolation error of a smooth solution, thus it is a well-known refinement criterion for anisotropic refinement (see for example [6, 7]). Here this criterion is based on the pressure, since we prefer a criterion which is insensitive to boundary layers [12]. The number of layers in the boundary layer grid should be the same everywhere for solution accuracy and the required layer thickness is known, so these layers can be created in the original grid and do not have to come from adaptive refinement.

For the numerical computation of the Hessian (see also [11]) we need discretised second-derivative operators. A particular complication for this discretisation is that our meshes always contain places where the grid size changes abruptly, as small cells lie next to twice larger cells (see Sect. 2). The number of these places increases significantly when automatic refinement is used so a suitable technique for computing the Hessian must be insensitive to cell size variations.

5.2.1 Definition of the Hessian Criterion

The pressure Hessian matrix is:

$$\mathscr{H}(p) = \begin{bmatrix} (p)_{xx} & (p)_{xy} & (p)_{xz} \\ (p)_{xy} & (p)_{yy} & (p)_{yz} \\ (p)_{xz} & (p)_{yz} & (p)_{zz} \end{bmatrix}. \tag{6}$$

Assuming, heuristically, that an indication of the local truncation error is given by $\mathscr{H}$ times the cell sizes to a power b (where b depends on the numerical method) and requiring equidistribution of this error indicator leads to a refinement criterion where the Hessian matrix is modified with a power law:

$$\mathscr{C}_H = (\mathscr{H}(p)^a), \tag{7}$$

where $\mathscr{H}^a$ has the same eigenvectors as $\mathscr{H}$ and eigenvalues that are those of $\mathscr{H}$ (in absolute value) to the power $a = \frac{1}{b}$. In general, we use $a = \frac{1}{2}$ which is appropriate for a second-order accurate discretisation.

5.2.2 Smoothed Gauss (SG) Method

Unfortunately, due to the variations in cell sizes, the numerical pressure is non-smooth. The SIMPLE-based pressure equation in ISIS-CFD contains a Laplace operator in finite-volume form for which normal derivatives on the cell faces are constructed with central interpolation. On non-uniform meshes these interpolations are first-order accurate, so the Laplace operator itself in the worst case has a truncation error of order zero. The pressure still converges because these local errors cancel, but the second derivatives of the pressure have the same order of accuracy as the Laplace operator, i.e. they are inconsistent. This is not due to the numerical evaluation of the second derivatives, but inherent in the pressure solution itself. The consequence for grid refinement is, that refining cells creates large errors in the Hessian on the boundaries between finer and coarser cells. Thus, the grid is not only refined where the solution dictates it, but also in places where it has already been refined. This leads to irregular meshes.

As the error in the Hessian is related to small-scale irregularities in the pressure field, which can be reduced by smoothing, we define a smoothed Gauss (SG) Hessian. Let the Gauss approximation to the gradient of a field q be given as:

$$\overrightarrow{\nabla}_G(q) = \frac{1}{V}\sum_f q_f S_f \mathbf{n}_f, \tag{8}$$

where the face values q_f are computed with central interpolation from the two neighbour cells. V is the volume of the cell, S_f are the areas of the faces. With the same face reconstructions, a Laplacian smoothing $\mathscr{L}$ is defined as:

$$\mathscr{L}(q) = \frac{\sum_f q_f S_f}{\sum_f S_f}. \tag{9}$$

Then the SG Hessian is computed as follows:

1. Compute the gradient of p using $\overrightarrow{\nabla}_G$.
2. Smooth each component of the gradient by applying N times the smoothing $\mathscr{L}$, where $N = 4$ is sufficient in most cases.
3. Compute the gradients of the smoothed gradient components using $\overrightarrow{\nabla}_G$.
4. Symmetrize the resulting Hessian matrix by setting $(\mathscr{H})_{ij} = \frac{1}{2}((\mathscr{H})_{ij} + (\mathscr{H})_{ji})$.
5. Smooth the Hessian by applying N times $\mathscr{L}$ to each component.

The reason for this procedure is, that the second derivatives of the pressure have zeroth-order oscillatory errors which means first-order errors in the derivatives and second-order wiggles in the solution itself. The smoothing operator uses the same type of discretisation as the original Laplace equation, so the smoothing itself introduces second-order wiggles. Therefore, it is useless to smooth the pressure. However, the gradient of the pressure has first-order errors; these are still small compared with the gradient itself but they are large enough to be removed by the smoother. Therefore, step 2 is the core of the procedure. The smoothing of the second derivative cannot further remove errors since they are of the same order as the solution now, but it creates a more regular refinement criterion and thus better meshes.

The smoothing procedure decreases the spurious oscillations in the refinement criterion but also reduces the intensity of physical small-scale features. This limitation of the criterion is the reason that all smoothing should be kept to a minimum.

5.2.3 Hessian at the Free Surface

The Hessian criterion cannot be directly evaluated at the free surface. Due to the presence of gravity, there exists a pressure gradient proportional to the density ρ, which is discontinuous in the normal direction at the free surface. Therefore, the second derivative is a Dirac δ function. For numerical solutions, the second derivative in the zone of varying α has a peak which grows as the grid becomes finer. Numerical differentiation produces large errors in this case.

As a result, no correct values can be computed for the pressure Hessian around the surface so an approximative procedure is needed. In the cells where $0.0001 \leq \alpha \leq 0.9999$, the gradient smoothing (step 2 in the SG algorithm) is not performed. The Hessian smoothing (step 5) is replaced by the following algorithm:

5a. Smooth the Hessian in all cells except those where $0.0001 \leq \alpha \leq 0.9999$ plus two layers of cells around those, to take into account that the perturbed pressure gradient in the cells at the surface influences the Hessian in their neighbours as well.

5b. Copy the computed values of the Hessian from outside the zone in (5a) across the zone, following the vertical direction (this removes the peak at the surface).

The criterion is copied in the upward direction, so the Hessian values computed in the water are used across the surface region.

5c. Smooth the Hessian only in the zone of (5a).

The idea of this approach is, that the SG Hessian at the surface must not be used. Therefore, sensible values must be copied from elsewhere. While this procedure is heuristic, it works well in practice as will be shown in Sect. 6.

5.3 The Combined Criterion

The final criterion is a combination of the two criteria above. Considering the problems of the Hessian criterion at the surface, it is tempting to select the free-surface criterion there and the Hessian everywhere else. However, the free-surface criterion specifies no refinement in the direction parallel to the surface, while this refinement may be needed if only to ensure that the grid at the surface is not less refined than just below it. The criterion in each cell is thus computed from both criteria.

The criteria are combined with a weighted maximum of the two tensors. We want the threshold T_r to indicate directly the desired cell size at the surface (as for the free-surface criterion), so a weighting factor c is applied only to the Hessian criterion:

$$\mathscr{C}_C = \max\left(\mathscr{C}_S, c\,\mathscr{C}_H\right). \tag{10}$$

Computing the approximate maximum of the two tensors is described in [12], a modification of the procedure presented by [4].

There are (as yet) no universal guidelines for choosing c, since appropriate values depend on the type of problem the results of interest. However, for the specification of guidelines for computations which are similar, a non-dimensional $\bar{c}$ is introduced as:

$$c = \bar{c}\left(\frac{L^2}{\frac{1}{2}\rho V_\infty^2}\right)^a. \tag{11}$$

where L, V_∞, and ρ are respectively the reference length, velocity, and the density. The power a is the one in Eq. (7). If two computations are geometrically identical except for a scaling in L and V_∞, then choosing the same $\bar{c}$ will result in identical refinement for both cases.

6 Test Cases

This section presents two test cases which evaluate the capacity of the refinement method to create effective fine meshes for typical ship flow cases and discuss the choice of the weighting factor c. The cases are the Series 60 ship (Sect. 6.1) and the Delft Catamaran in drift condition (Sect. 6.2).

6.1 Series 60 Wave Pattern

The Series 60 hull is studied in straight-ahead motion and calm water. Detailed experiments for this case are available from IIHR [5] at $Fr = 0.316$ and $Re = 5.3 \cdot 10^6$. Apart from this Froude number, we also compute the flow at $Fr = 0.16$ and $Fr = 0.41$ with $Re = 2.7 \cdot 10^6$ and $Re = 6.9 \cdot 10^6$ respectively.

The computations are started from an original grid (242k cells) without any refinement around the free surface. Since directional derefinement is not yet available, the refinement is initialised as in Fig. 3b by allowing only refinement in vertical direction. The computation is then continued with refinement in all directions. The threshold (equivalent to the desired cell size at the surface) is the same for all cases, $T_r = 0.001L$ which is the usual cell size at the surface for ISIS-CFD. For each Froude number, the flow is computed with $\bar{c} = 0.016$, 0.024 and 0.032 which gives the grid sizes in Table 1. Reference results are obtained without grid refinement on a fine mesh of 3.45M cells. Turbulence is modelled with the Menter $k - \omega$ SST model.

The wave pattern at the three Froude numbers is shown in Fig. 4. The wave strength varies strongly with Fr; for the two highest Fr, the bow wave breaks. Figure 5 shows cross-sections of the mesh for the three Froude numbers at the largest weighting factor $\bar{c} = 0.032$. Around the position of the free surface, the meshes have directional refinement. The cell size below the surface decreases gradually from the bottom up, the finest cells are concentrated in the bow wave (left) and the stern wave (right). The refined cells below the waves are predominantly square, although some cells near the surface are smaller in the vertical direction. The size of the cells in the original grid can be seen in the upper right corner of Fig. 5b, so the entire fine grid is actually created by automatic refinement.

Table 1 Number of cells in the refined meshes for the Series 60 test cases

$\bar{c}$	0.016	0.024	0.032
$Fr = 0.16$	698k	1169k	1793k
$Fr = 0.316$	597k	892k	1360k
$Fr = 0.41$	757k	973k	1348k

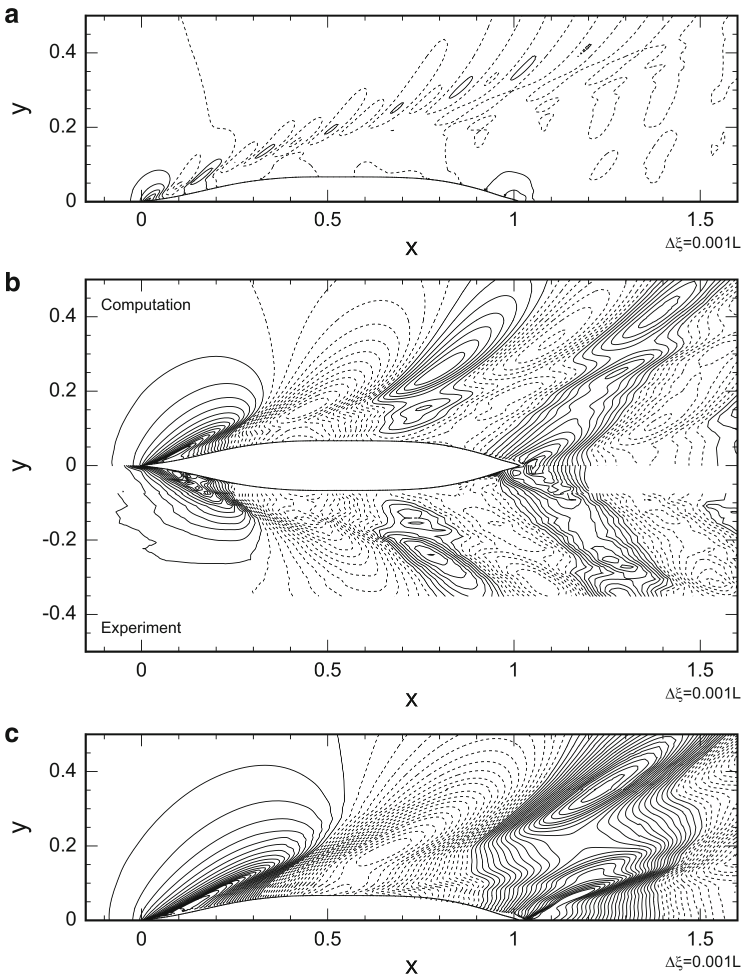

Fig. 4 Wave patterns for the Series 60 at $Fr = 0.16$ (**a**), $Fr = 0.316$ (**b**), and $Fr = 0.41$ (**c**). The isoline distance is the same for all figures and corresponds to $L/1000$. The $Fr = 0.316$ result is compared with experiments from IIHR [5]

The solutions for all cases are compared in Fig. 6, which shows the free surface in three X-cuts when $\bar{c}$ is varied, compared with the non-adapted fine grid. For the highest Froude number (Fig. 6c), all solutions are close to the fine-grid solution, only some discrepancy is seen for the $\bar{c} = 0.024$ solution behind the ship. However, at $Fr = 0.316$ (Fig. 6b) notable differences exist for $\bar{c} = 0.016$. Finally, at $Fr = 0.16$ (Fig. 6a) the computation for $\bar{c} = 0.016$ is the closest to the fine-grid solution, which means that both are questionable! For very small waves, the wave heights may in fact

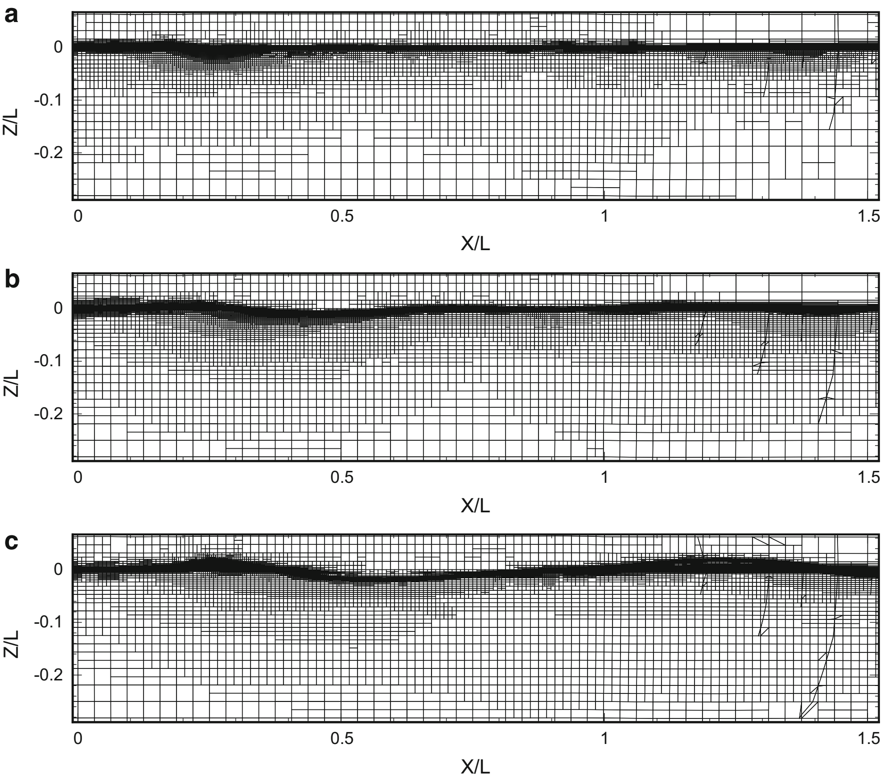

Fig. 5 Series 60, cross-sections of the grid at $Y/L = 0.118$ for $Fr = 0.16$ (**a**), $Fr = 0.316$ (**b**), and $Fr = 0.41$ (**c**)

be overpredicted on too coarse grids [2]. The solutions for $\bar{c} = 0.024$ and $\bar{c} = 0.032$ are reasonably close. Figure 6d shows velocity profiles on a horizontal line below the water at the stern. Only $Fr = 0.316$ is shown here, since the results for other Fr are similar. Results for all $\bar{c}$ are close to the fine-grid solution, the discrepancy with the experiments may be due to the isotropic $k - \omega$ SST turbulence model which is not always well adapted to the simulation of ship wake flows [1, 3].

In conclusion, to accurately model the wave pattern, higher values for $\bar{c}$ are needed at low Fr than at high Fr. However, considering the weak influence of the waves on the rest of the flow field at low Fr, a constant choice for $\bar{c}$ is justified, which gives the added advantage that the grid on the hull below the surface is refined in the same way for all Fr. For slender hulls like the Series 60, setting $T_r = 0.001L$ with a value of $\bar{c}$ around 0.024 gives sufficient accuracy. Higher $\bar{c}$ increase significantly the total number of cells (Table 1).

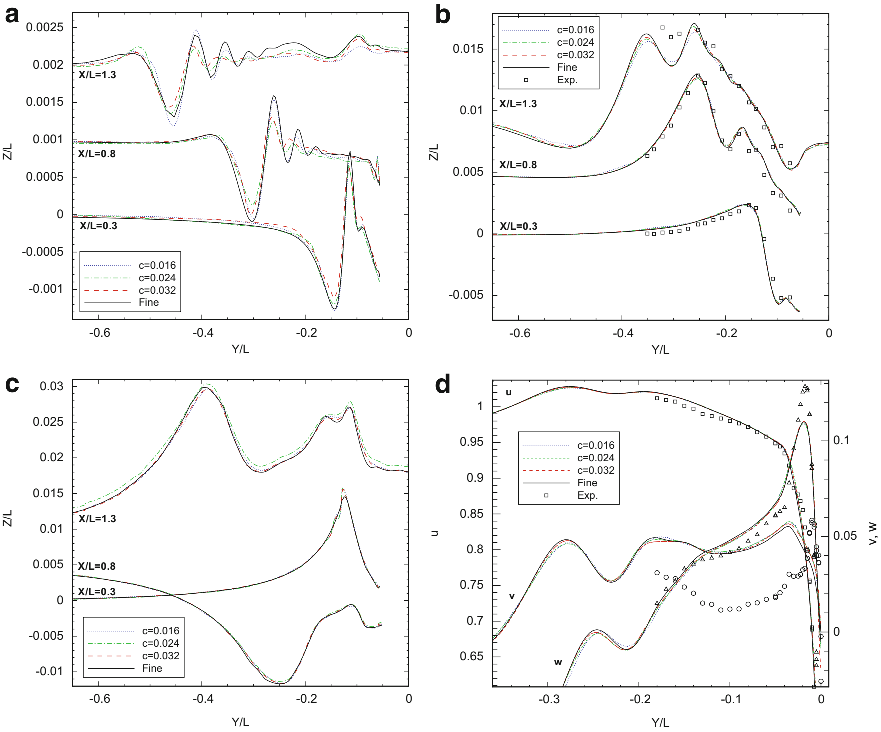

Fig. 6 Series 60, comparisons of the water surface in three X-cuts at $Fr = 0.16$ (**a**), $Fr = 0.316$ (**b**), and $Fr = 0.41$ (**c**). Comparison of the velocities on the line $X/L = 1$, $Z/L = -0.02$ (**d**) for $Fr = 0.316$. Experiments from IIHR [5]

6.2 Delft Catamaran in Sideslip

As a second case, the flow around the Delft-327 Catamaran in sideslip is computed. The particularity of this case is that it has both strong waves and longitudinal vortices created below the hull, so the combined refinement criterion is of great interest since it can capture both these features. The case concerns a motion in steady drift ($\beta = 6^o$) at $Fr = 0.4$. Automatic grid refinement is started from an original grid of 1.0M cells which has some limited refinement around the surface, it is computed with $T_r = L/500$ and with $\bar{c} = 0.064$. This higher value of $\bar{c}$ than for the Series 60 case comes mostly from the higher T_r; the combination creates refinement that is concentrated in the trailing vortices. The converged refined grid has 2.96M cells, the results are compared with a fine-grid solution of 20M cells. The anisotropic EASM turbulence model [1] is used.

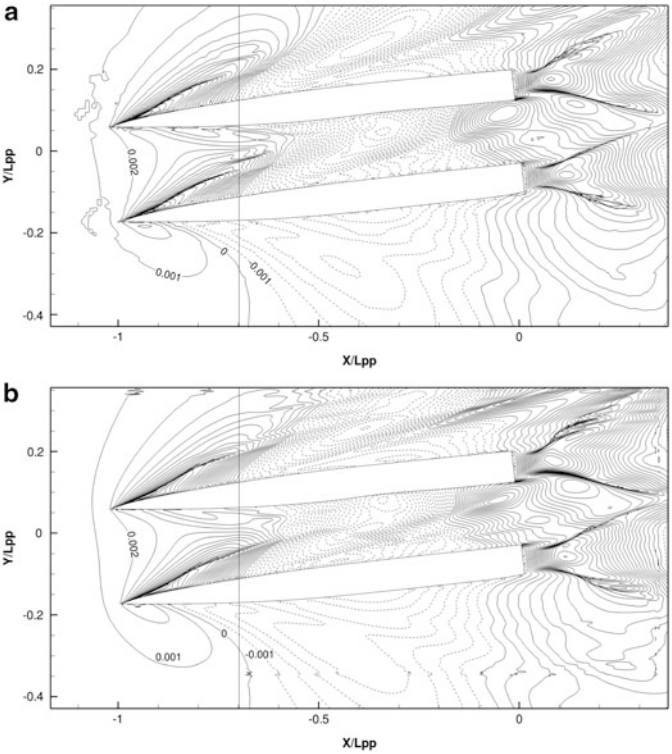

Fig. 7 Delft Catamaran, free surface on adaptively refined (**a**) and fine (**b**) grid. The *vertical line* shows the position of the cut in Fig. 8

A cut through the mesh is shown in Fig. 8a, with refinement around the deformed surface and in the cores of two vortices below the hulls. Figure 7 compares the free surface for the refined-grid and fine-grid solution. The two are similar, although the far-field waves are damped slightly more on the refined grid due to the high threshold. However, the breaking bow waves have more details on the refined grid, because the cells are locally smaller in y-direction than on the fine grid. These small waves (especially between the hulls) are also seen in Fig. 8b, c, which show the axial velocity and the free surface. The vortices are computed on very fine cells in the refined mesh, so they are notably stronger than on the fine mesh.

Thus, also for this case the relevant flow features can be computed well on an adaptively refined mesh. The value for $\bar{c}$, if $T_r = L/1000$ had been used, would have been 0.032. This is still higher than for the Series 60 case, which is justified by the objective to capture the trailing vortices particularly well.

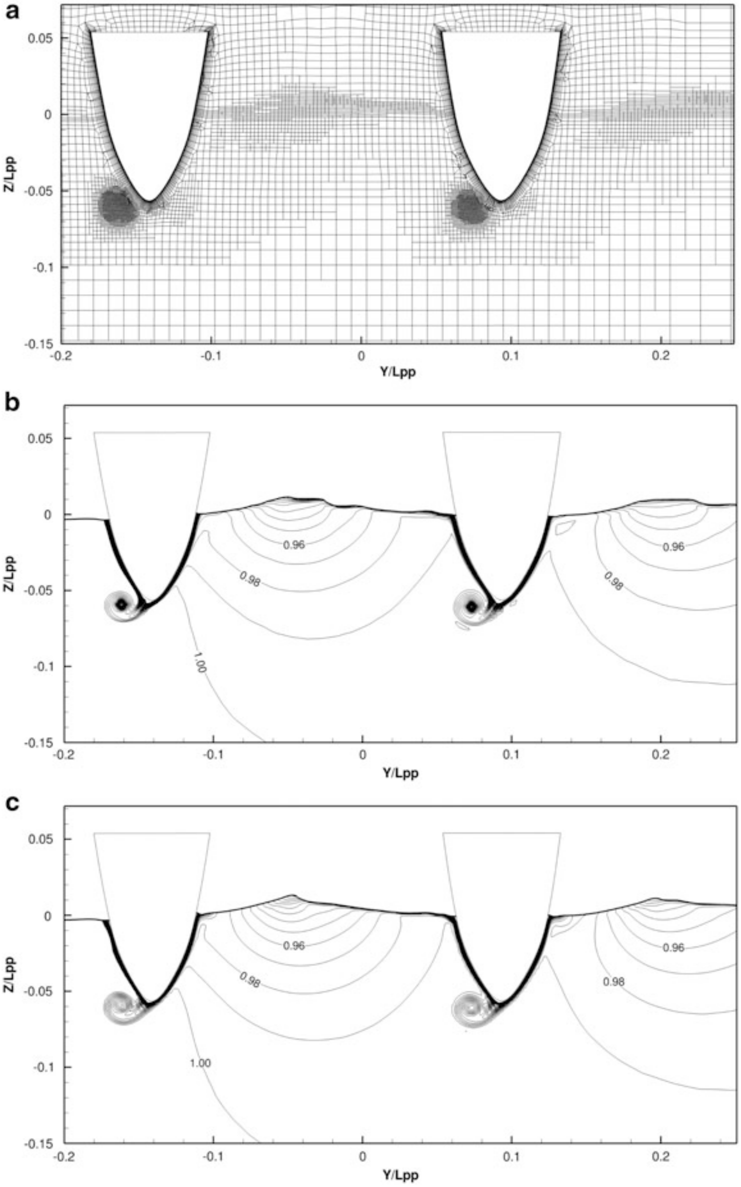

Fig. 8 Delft Catamaran, cut at $x/L = 0.3$. Refined mesh (**a**), axial velocity on refined (**b**) and fine (**c**) mesh

7 Conclusions

The goal of this paper is the simulation of free-surface water flows. For these flows, fine grids are needed around the water surface and the surface position is not known beforehand. Therefore, automatic grid refinement can be very useful, placing fine cells only exactly where they are needed and removing from the user the task of estimating the surface position before the computation in order to generate the mesh.

To create the fine grid needed for flows with waves entirely using grid refinement, the refinement technique must adapt the grid often to the solution as it evolves, since the surface position for example may change position when the grid becomes finer. For refinement by subdivision of cells, this means that both refinement and derefinement are needed even for steady cases, the latter to remove refined cells which are no longer needed as the simulation converges. Refinement and derefinement must be performed in an anisotropic way to limit the total number of cells in three dimensions. The anisotropic derefinement is especially important to allow cells which were refined in several directions and need to remain refined in only one, to get rid of the refinement in the other directions.

Suitable refinement criteria must create refinement both around the surface, to resolve the convection equation for the volume fraction, and in the region below the surface in order to capture the orbital flow fields. To ensure a regular mesh at the surface, a derivative-based criterion suitable for detecting the velocity and pressure fields is combined with a robust free-surface capturing criterion. We choose a criterion which uses directional refinement in the region where the volume fraction is between 0.1 and 0.9 and refinement based on the pressure Hessian. On unstructured hexahedral meshes, the Hessian is found with Gauss derivation twice to compute first the gradient of the pressure and then the second derivatives, followed by smoothing to remove irregularities. The numerical pressure Hessian has a peak at the free surface, due to the discontinuity in the pressure gradient. Therefore, the free-surface region is not smoothed and the computed Hessian outside the surface zone is extrapolated through this zone. To combine the two criteria, their relative weights must be chosen; a non-dimensional scale-independent form for the Hessian criterion is introduced with a weight $\bar{c}$.

Two different ship flow test cases show that the automatic grid refinement is able to create effective fine meshes from original meshes which are neither much refined near the surface, nor around other significant flow features such as vortices. These features were simulated with the same precision as on uniformly fine meshes using 50 % to 85 % less cells. For the weight of the Hessian, the Series 60 case shows that it is acceptable to choose $\bar{c}$ independently of Fr. Sensible values for slender ships are $\bar{c} = 0.02$–0.035 with a refinement threshold set around $T_r = L/1000$, a high must be chosen if wake features are of major importance. Based on these tests, the perspective of generating complete free-surface meshes with automatic refinement seems entirely realistic.

Acknowledgements Computations were performed using HPC resources from GENCI (Grant 2013-21308), which is gratefully acknowledged.

References

1. Deng, G.B., Visonneau, M.: Comparison of explicit algebraic stress models and second-order turbulence closures for steady flows around ships. In: 7th International Conference on Numerical Ship Hydrodynamics, Nantes, France (1999)
2. Deng, G.B., Guilmineau, E., Leroyer, A., Queutey, P., Visonneau, M., Wackers, J.: Verification and validation for unsteady computation. In: Larsson, L., Stern, F., Visonneau, M. (eds.) A Workshop on Numerical Ship Hydrodynamics, Gothenburg, Sweden (2010)
3. Duvigneau, R., Visonneau, M.: On the role played by turbulence closures in hull shape optimization at model and full scale. J. Mar. Sci. Technol. **8**, 11–25 (2003)
4. George, P.L., Borouchaki, H.: Delaunay Triangulation and Meshing – Application to Finite Elements. Hermes, Paris (1998)
5. Longo, J., Stern, F.: Effects of drift angle on model ship flow. Exp. Fluids **32**, 558–569 (2002)
6. Loseille, A., Dervieux, A., Alauzet, F.: Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations. J. Comput. Phys. **229**, 2866–2897 (2010)
7. Majewski, J.: Anisotropic adaptation for flow simulations in complex geometries. In: 36th CFD/ADIGMA Course on hp-Adaptive and hp-Multigrid Methods. Lecture Series 2010–01. Von Karman Institute, Sint-Genesius-Rode, Belgium (2009)
8. Queutey, P., Visonneau, M.: An interface capturing method for free-surface hydrodynamic flows. Comput. Fluids **36**(9), 1481–1510 (2007)
9. Rhie, C.M., Chow, W.L.: A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. AIAA J. **17**, 1525–1532 (1983)
10. Wackers, J., Ait Said, K., Deng, G.B., Queutey, P., Visonneau, M., Mizine, I.: Adaptive grid refinement applied to RANS ship flow computation. In: 28th Symposium on Naval Hydrodynamics, Pasadena, California (2010)
11. Wackers, J., Deng, G.B., Visonneau, M.: Combined tensor-based refinement criteria for anisotropic mesh adaptation in ship wave simulation. In: ADMOS 2011, Paris, France (2011)
12. Wackers, J., Leroyer, A., Deng, G.B., Queutey, P., Visonneau, M.: Adaptive grid refinement for hydrodynamic flows. Comput. Fluids **55**, 85–100 (2012)
13. Wackers, J., Deng, G.B., Queutey, P., Visonneau, M., Guilmineau, E., Leroyer, A.: Sliding grids and adaptive grid refinement applied to ship hydrodynamics. In: 29th Symposium on Naval Hydrodynamics, Gothenburg, Sweden (2012)