Evaluation of Propeller Inflow Improving Device with Adaptive Grid Refinement Computation

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

Eighth International Conference on Computational Fluid Dynamics (ICCFD8), Chengdu, China, July 14-18, 2014

ICCFD8-2014-0180

Evaluation of Propeller Inflow Improving Device with Adaptive Grid Refinement Computation

G. Deng[1], E. Guilmineau[1], A. Leroyer[1], P. Queutey[1], M. Visonneau[1], J. Wackers[1] & F. Salvatore[2]
[1]LHEEA Lab. CNRS UMR 6598,Ecole Centrale de Nantes, France
[2]Marine Propulsion and Cavitation Lab, CNR-INSEAN, Italy
Corresponding author: Ganbo.Deng@ec-nantes.fr

Abstract: A ship design with boundary layer alignment device (BLAD) aiming at improving the inflow of a propeller is evaluated using a RANSE computation with adaptive grid refinement. This paper is focused on model scale simulation for which experimental results can be used for validation. Propeller performance is evaluated with a RANSE/BEM coupling procedure. Numerical simulation reveals that adaptive grid refinement is an efficient way to obtain a reliable prediction for such configuration. Moreover, grid refinement criterion is found to be of crucial importance. The newly proposed smoothed pressure Hessian is capable of evaluating correctly the impact of the BLAD on the performance of the propeller.

Keywords: Propeller-hull interaction, Adaptive grid refinement, Propeller inflow improving device.

1 Introduction

Due to greenhouse effect, regulation concerning gas emission is becoming more and more restrictive for ship building industry. There is an increasing interest to improve the propeller-hull interaction efficiency to fulfill the new regulation. In addition to hull form and propeller optimization, propeller inflow improving device is another interesting approach. Additional appendages such as vortex generator (VG), boundary layer alignment device (BLAD), pre-swirl stator (PSS), etc are added. Those installations aim to modify the inflow to the propeller such that propeller efficiency is improved, or to produce thrust on its own. To be efficient, devices such as VG and BLAD must be installed at an upstream position relatively far away from the propeller where boundary layer thickness begins to grow quickly. Although additional resistance due to such devices can be predicted without too much difficulty, it is a much more challenging task to capture its influence far downstream to the propeller and to simulate correctly the reaction of the propeller due to those modifications. As the length scale of those devices is about one order or more smaller that the length scale of the ship stern, it is mandatory that numerical approaches employed for such simulation are capable of capturing flow structures with different length scales. Before numerical simulation tools can be used with confidence in a design procedure for propeller inflow improving devices, assessment on the accuracy of numerical prediction for such applications needs to be made. This is one of the objectives of the European FP7 STREAMLINE project. Several propeller inflow improving devices including VG, BLAD and PSS have been designed [1]. The objective of this design is not to improve propeller efficiency, but to provide a representative configuration that can be used for the assessment of numerical simulation tools. Model scale measurements have been made which can be used for numerical validation [2]. We have been involved in the evaluation of the BLAD during this project and simulations using adaptive grid refinement have been made. It has been found that grid refinement criterion is of crucial importance. This validation work will be reported in the present paper.

2 Flow solver and grid refinement

Computation in the present study is performed with our in-house flow solver ISIS-CFD which is the flow solver of FINE^{TM}/Marine compoutational suite distributed by NUMECA Int. It is based on a finite-volume discretisation for the the Unsteady Reynolds-Averaged Navier Stokes (URANS) equations on unstructured grids. This section describes the governing flow equations, the discretisation, and the type of meshes used, concentrating on those aspects that are most important for grid refinement and the construction of refinement criteria. Full details of the discretisation can be found in [3, 4].

2.1 Governing equations

The ISIS-CFD flow solver which we develop resolves the incompressible URANS equations in a mixture-fluid formulation to model water–air two-fluid flow. Here, the entire fluid is modelled as a numerical mixture of the pure fluids on the two sides of the interface. The system uses the conservation laws for momentum, total mass, and mass of each fluid. When the densities of the individual fluids are constant, the latter two reduce to $\nabla \cdot \mathbf{U} = 0$ and to a volume-of-fluid equation. In integral form, the eqations are written as follows:

$$\frac{\partial}{\partial t}\int_V \rho U_i dV + \int_S \rho U_i \mathbf{U}\cdot\mathbf{n} dS = \int_S (\tau_{ij} I_j - p I_i)\cdot\mathbf{n} dS + \int_V \rho g_i dV, \tag{1}$$

$$\int_S \mathbf{U}\cdot\mathbf{n} dS = 0, \tag{2}$$

$$\frac{\partial}{\partial t}\int_V c_i dV + \int_S c_i \mathbf{U}\cdot\mathbf{n} dS = 0, \tag{3}$$

where V is a volume, bounded by the closed surface S with a unit normal vector $\mathbf{n}$ directed outward. $\mathbf{U}$ and p represent, respectively, the velocity and pressure fields. τ_{ij} and g_i are the components of the viscous stress tensor and the gravity vector, whereas I_j is a vector whose components are zero, except for the component j which is equal to unity. c_i is the volume fraction for fluid i and is used to distinguish the presence ($c_i = 1$) or the absence ($c_i = 0$) of fluid i. In the case of turbulent flows, additional transport equations for modelled variables are solved in a form similar to the momentum equations and they are discretized and solved using the same principles.

The effective flow physical properties (viscosity μ and density ρ) are obtained from the physical properties of the constituent fluids (μ_i and ρ_i) with the following constitutive relations. The last identity follows from the definition of the volume fraction.

$$\rho = \sum_i c_i \rho_i, \qquad \mu = \sum_i c_i \mu_i, \qquad 1 = \sum_i c_i. \tag{4}$$

Thus, for two fluids, equation (3) only has to be solved for fluid 1.

In this framework, free-surface water flows are modelled by specifying a discontinuous inflow condition for c_i ($c_i = 1$ below the surface and $c_i = 0$ above it). As equation (3) is a pure convection equation, the resulting solution for c_i is discontinuous in the whole domain. This discontinuity represents the free surface. Thus, a sharp water surface is obtained without a specific model at the surface. This method is robust and straightforward, which means that it can be used for a wide range of flows.

2.2 Face-based discretisation

The flow equations of the previous subsection are discretised in a finite-volume framework. Pressure-velocity coupling is obtained through implicit time integration with a Rhie & Chow SIMPLE-type method. Thus, each time step is solved through a series of nonlinear iterations with segregated updates for the variables: the velocity updates come from the momentum equations (1) and the pressure is given by the mass conservation law (2), transformed into a pressure equation. The volume fraction c_i for fluid 1 comes from a discretisation of the linear convection equation (3) which is solved in each nonlinear iteration, decoupled from the pressure and velocity updates.

The discretisation is face-based. While all unknown state variables are cell-centered, the systems of equations used in the implicit time stepping procedure are constructed face by face. Fluxes are computed in a loop over the faces and the contribution of each face is then added to the two cells next to the face. This technique poses no specific requirements on the topology of the cells. Therefore, the grids can be completely unstructured, cells with an arbitrary number of arbitrarily-shaped faces are accepted.

2.2.1 Central discretisations

The core of the discretisation are the reconstructions of the state variables and their derivatives from the cell centres to the faces. For the diffusive fluxes and the pressure equation, these are basically central approximations of the normal derivatives. In case of misalignment, i.e. when the face normal vector is not aligned with the line between the neighbouring cell centres, extra correction terms are added using the cell-centered gradients computed with a Gauss method.

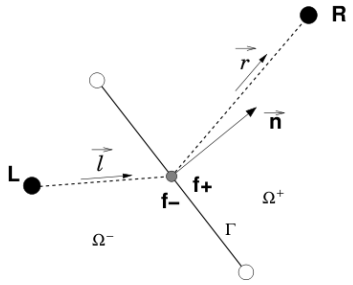

Figure 1: Cell face notations.

As an example, the reconstruction of the pressure and its normal derivative is shown here for the case of constant density, i.e. below the surface. More details on the reconstruction, including the weighting of the reconstruction with the density when it is non-constant, are given in [3, 4]. From Taylor series expansion on both sides of the face, the reconstruction of the pressure on the face is established in the following compact form involving left and right side cell-centered data only (see figure 1 for notations). This pressure reconstruction is used in the momentum equations.

$$p_f = \frac{h^+}{h} p_L + \frac{h^-}{h} p_R + \boxed{\left(\frac{h^- \mathbf{E}^+ - h^+ \mathbf{E}^-}{h} \right) . \left(\frac{h^+}{h} \overrightarrow{\nabla} p_L + \frac{h^-}{h} \overrightarrow{\nabla} p_R \right)}. \tag{5}$$

The framed term is kept explicit in each solution step, while the non-framed term is implicited in the solver. Geometrical vectors $\mathbf{E}^{\pm}$ are introduced so that the framed term contribution goes to zero when the grid becomes orthogonal ($\mathbf{Lf}.\mathbf{n} = \mathbf{fR}.\mathbf{n} = 0$):

$$\begin{aligned} \mathbf{E}^- &\triangleq (\mathbf{Lf}.\mathbf{n})\,\mathbf{n} - \mathbf{Lf}, \\ \mathbf{E}^+ &\triangleq (\mathbf{fR}.\mathbf{n})\,\mathbf{n} - \mathbf{fR}. \end{aligned} \tag{6}$$

Distances used are the projected distances to the face $h^{\pm}$ and the projected distance h between the L and R cell centers:

$$h^- = \mathbf{Lf}.\mathbf{n}\,, \quad h^+ = \mathbf{fR}.\mathbf{n}\,, \quad h = h^- + h^+ = \mathbf{LR}.\mathbf{n}. \tag{7}$$

The reconstruction of the pressure normal derivative for the pressure equation is obtained with a reconstruction following the same rules as for the quantity on the face:

$$\left(\overrightarrow{\nabla} p.\mathbf{n} \right)_f = \frac{p_R - p_L}{h} + \boxed{\left(\frac{\overrightarrow{\nabla} p_L.\mathbf{E}^- + \overrightarrow{\nabla} p_R.\mathbf{E}^+}{h} \right)}. \tag{8}$$

Here again, the framed (explicit) term contribution goes to zero when the grid becomes orthogonal. The non-framed term is the implicit part that goes into the matrix for the pressure equation.

2.2.2 Convective fluxes

The convective fluxes are computed using limited schemes in Leonard's Normalised Variable Diagram (NVD) [5]. These schemes use interpolation between three points for the reconstruction at a cell face, depending on the flow direction, see figure 2. The points C and D are the cell centres of the two neighbour cells, in the upwind and downwind direction. The point U, for structured grids, is the cell centre upwind of point C; for unstructured grids there usually is no suitable cell at this point, so the value in U is extrapolated from C using the gradient computed in C. Then, a nonlinear reconstruction from these three points is used to find the face value such that the solution remains monotone. Many different schemes exist, the standard scheme in ISIS-CFD is the AVLSMART scheme [6].

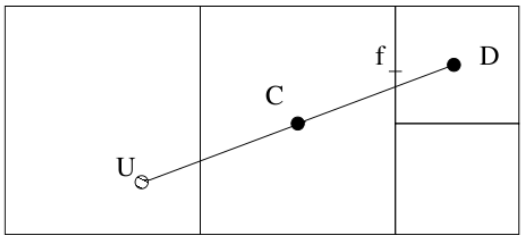

Figure 2: Points for NVD face reconstruction. The points C and D are the neighbouring cells of the face f, the point U is the mirror image of D w.r.t. C.

For the volume fraction equation, the solutions are known a-priori to be numerical approximations to a discontinuity at the water surface, with constant values elsewhere. For these special solutions, normal accuracy considerations do not hold. Therefore, downwind-biased NVD schemes are used which add artificial antidiffusion (negative diffusion) to the equation while preserving stability, see for example [7]. Antidiffusion continuously compresses the interface in c_i so it remains as sharp as possible. As the discontinuous c_i does not have a well-defined gradient, the state U in our BRICS scheme [4] is not extrapolated from C but interpolated from the cell centre nearest to the point U and its neighbours.

Misalignment corrections, for the case where the face centre does not lie on the midpoint of the line CD (the point of interpolation) are not used. For the volume fraction equation, these cannot even be envisaged because no useful gradient information exists. However, misalignments have a strong effect on the accuracy of the solution for c_i, so if possible the mesh should be made such that they are prevented as much as possible near the free surface.

2.3 Meshes

For this study, as usual for ISIS-CFD, unstructured hexahedral meshes generated with the HEXPRESS^{TM} grid generator from NUMECA Int. are used (see figure 3 for an example). In these meshes, variations in cell size are handled by having small cells laying next to larger cells. This situation is called 'hanging nodes' in the literature and solvers often needs specific discretisations to handle these topologies. In ISIS-CFD, due to the face-based algorithm, these cells are treated in exactly the same way as all the others: the larger cells are simply seen as cells with more than 6 faces. Thus, no specific hanging-node treatment is included.

Unstructured hexahedral grids are ideal for automatic grid refinement. Isotropic or anisotropic grid refinement can be applied to any of the hexahedral cells, the result will still be an unstructured hexahedral mesh. Therefore, locally refined meshes can be used directly in a flow solver that supports unstructured hexahedral meshes, without requiring changes to the flow solver.

Due to the small cell – large cell transitions, strong cell misalignments exist in parts of the grid. When the grid is refined, these situations persist: no matter the size of the grid, there will always be cells that have neighbours twice smaller than themselves. Thus, misalignment problems may limit the accuracy of the solutions. However, when the grids become finer, the structured zones between the cell size transitions

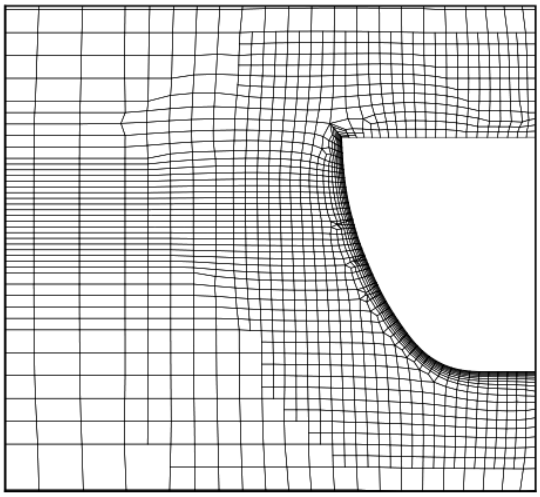

Figure 3: Cut through an unstructured hexahedral mesh.

should become larger and larger. Thus, the percentage of cells that have misalignments is reduced, which increases the overall accuracy. For automatic mesh refinement, it is therefore essential to ensure grids with smoothly varying cell sizes in order to limit the number of cells that have misalignments.

3 Grid refinement procedure

The grid refinement procedure developed for ISIS-CFD [8, 9] is integrated completely in the flow solver. The method is entirely parallelised, including automatic redistribution of the grid over the processors. During a flow computation, the refinement procedure is called repeatedly. In such a call, first a refinement criterion is calculated, which is a real field variable based on the flow field that indicates where cells should be refined. Then, in a separate step of the procedure the grid is refined based on this criterion. These steps are kept separated so the criterion can be changed easily without modifying the rest of the refinement method. For steady flow, the refinement procedure eventually converges: once the grid is correctly refined according to the criterion, further calls to the procedure no longer cause any changes.

3.1 Anisotropic refinement

Grid refinement for hexahedral cells can be either isotropic, where a cell is always refined in all its directions at once, or anisotropic where division in only one or two directions is possible as well. For realistic applications, anisotropic refinement is essential. Isotropic grid refinement is very costly in three dimensions, since every refinement of a cell means a division in eight. Thus, creating very fine cells to accurately resolve a local flow phenomenon becomes almost impossible. However, by applying anisotropic refinement for flow 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 our refinement is based on unstructured hexahedral original grids as shown in figure 3. In these grids, cells of completely different aspect ratios lie 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 [9]. Isotropic refinement is not enough to prevent this, so directional refinement is the mandatory choice.

3.2 Tensor refinement criteria

For directional refinement, a way is needed to specify different cell sizes in different directions. The use of metric tensors as refinement criteria is such a way. This technique was first developed for the generation and refinement of unstructured tetrahedral meshes [10, 11, 12]. It is also an extremely useful and flexible framework for the refinement of unstructured 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. As such, it can be thought of as the continuous equivalent of a mesh [10]. This ideal mesh depends on the flow field. There exists an 'exact' criterion which is computed from the exact solution; as the grid is refined the actual computed criterion converges to this exact criterion (this is different from the classical error indicating criteria where the criterion is halved when a cell is refined.) Adaptive grid refinement is performed to get the actual cell sizes in the refined grid as close to these ideal sizes as possible, so the refined mesh can be considered as a 'discretisation' of the criterion.

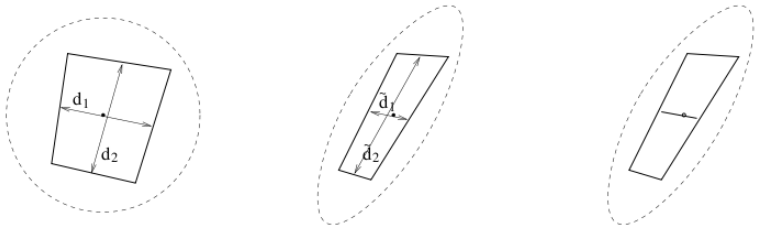

Figure 4: Tensor refinement criterion. Cell Ω_i and unit circle (reference) in the physical space (a), deformed cell $\tilde{\Omega}_i$ and deformed circle after application of the transformation C_i (b), and refinement to create a uniform grid in the deformed space (c).

The refinement criterion in each cell is a 3×3 symmetric positive definite matrix $\mathcal{C}_i$, which is interpreted as a geometric transformation of the cell in the physical space to a deformed space (figure 4). The refinement of the cells is decided as follows. Let the criterion tensors $\mathcal{C}_i$ in each cell be known (their computation from the flow solution is described in section 3.3). In each hexahedral cell, the cell size vectors $\mathbf{d}_{j,i}$ $(j = 1, \ldots, 3)$, which are the vectors between the opposing face centres in the three cell directions, are determined. Next, the modified sizes are computed as:

$$\tilde{\mathbf{d}}_{j,i} = \mathcal{C}_i \mathbf{d}_{j,i}. \tag{9}$$

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

$$\|\tilde{\mathbf{d}}_{j,i}\| \geq T_r. \tag{10}$$

The tensors $\mathcal{C}_i$ are direct specifications of the desired cell sizes: in a converged refined grid, the cell sizes are inversely proportional to the magnitude of the $\mathcal{C}_i$.

3.3 Pressure Hessian criterion

One of the most suitable criterions to capture vortical structure existing in a wake of a body is the criterion based on the Hessian matrix of second derivatives of the pressure. This section describes how to compute those second derivatives in a finite-volume code.

3.3.1 Definition of the Hessian criterion

The pressure Hessian matrix is:

$$\mathcal{H}(p) = \begin{bmatrix} (p)_{xx} & (p)_{xy} & (p)_{xz} \\ (p)_{xy} & (p)_{yy} & (p)_{yz} \\ (p)_{xz} & (p)_{yz} & (p)_{zz} \end{bmatrix}. \tag{11}$$

To compute the refinement criterion, the Hessian matrix is modified with a power law. Thus, the criterion in cell i is:

$$\mathcal{C}_{H,i} = (\mathcal{H}(p)^a)_i, \tag{12}$$

where $\mathcal{H}^a$ has the same eigenvectors as $\mathcal{H}$ and eigenvalues that are those of $\mathcal{H}$ (in absolute value) to the power a. In general, we use $a = \frac{1}{2}$.

3.3.2 Computing the pressure Hessian

Hessian-based criteria are often used to control anisotropic grid refinement [10, 12, 13]. Here this criterion is based on the pressure, because of our refinement strategy in boundary layers [9]. We consider that the number of layers in the boundary layer grid should be the same everywhere, to ensure the best grid quality. And since the approximate thickness of the boundary layer is known, these grid layers can be inserted on the original grid. Therefore, it is unnecessary to employ a criterion which has very high values in the boundary layer region. The pressure varies little over the thickness of a boundary layer, so its second derivatives there are limited.

To compute the Hessian matrix of a numerical solution, second-derivative operators must be discretised. 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 section 2.3). These places do not disappear when the mesh is refined, on the contrary their number increases significantly when automatic refinement is used. However, many discretisations of the second spatial derivatives depend on the mesh becoming smoother and smoother as it is refined, in order to obtain second-order accuracy. In particular, the well-known computation of the Hessian by using the Gauss theorem for finding the gradients of the quantity, then applying the Gauss theorem again to the gradients in order to compute the second derivatives, has an accuracy of order zero in places where the grid size changes abruptly. A suitable technique for computing the Hessian, on the other hand, must be insensitive to these cell size changes. This section describes two possible ways of computing the Hessian matrix.

3.3.3 Third-order least-squares approximation

A first solution for the computation of the Hessian is to use a least-squares approximation [9]. In each cell, we construct a least-squares fit of a third-order polynomial to the solution in the cell, its neighbour cells and its neighbours' neighbours. The approximated Hessian is then computed from the second derivatives of this polynomial. Let $P_j(\mathbf{x})$, $j = 1 \cdots 20$ be the set of basic three-dimensional polynomial functions in $\mathbf{x}$ of up to third order (i.e. $P_1(\mathbf{x}) = 1$, $P_2(\mathbf{x}) = x$, $P_3(\mathbf{x}) = y$, $\cdots$, $P_5(\mathbf{x}) = x^2$, $\cdots$, $P_{11}(\mathbf{x}) = x^3$, $\cdots$,$P_{20}(\mathbf{x}) = xyz$). Furthermore, $\mathbf{I}$ is the vector of cell indices of a cell i, its neighbours and its neighbours' neighbours. Then we shall search coefficients β such that the polynomial:

$$\phi_i(\mathbf{x}) = \sum_{j=1}^{20} \beta_j P_j(\mathbf{x} - \mathbf{x}_i), \tag{13}$$

is the closest fit to the values of p in the cell centres of $\mathbf{I}$, within the space defined by the set $P_j(\mathbf{x} - \mathbf{x}_i)$. Defining the matrix A and vector $\mathbf{b}$ as:

$$A_{jk} = P_j(\mathbf{x}_{\mathbf{I}_k} - \mathbf{x}_i), \qquad b_k = p_{\mathbf{I}_k}, \qquad j = 1 \cdots 20,\ k = 1 \cdots \text{size}(\mathbf{I}), \tag{14}$$

the coefficients β are found as:

$$\beta = (A^T A)^{-1} A^T \mathbf{b}. \tag{15}$$

According to the definition of the least-squares procedure, there is no better third-order polynomial fit to the points $\mathbf{I}$, so the error in the fit is at least fourth-order. Therefore, if p is a sufficiently smooth function, the approximated Hessian $\mathcal{H}_{LS3}(p)$ is a second-order accurate approximation to $\mathcal{H}(p)$ (two orders are lost by the double differentiation). Tests with manufactured solutions in [9] confirm this on our refined grids.

3.3.4 Smoothed Gauss method

Unfortunately, the numerically evaluated pressure p_h is *not* a smooth function. Our SIMPLE-based pressure equation contains a Laplace-type operator in finite-volume form, for which the fluxes over the faces are based on the normal derivatives of the pressure computed with equation (8). On arbitrary meshes, these are first-order accurate. Therefore, the truncation error of the Laplace equation which contains the derivatives of the fluxes is formally of order zero. These local truncation errors cancel globally, because they depend on the relative sizes of a cell and its neighbours so they have opposite signs in small and large cells lying next to each other. Therefore the solution itself for the pressure is at least first-order accurate ($p_h = p + \mathcal{O}(h)$ where h is a measure of the grid size). However, the second derivatives of the pressure appear directly in the pressure equation so they have the same order of accuracy as the truncation error, i.e. $\mathcal{H}(p_h) = \mathcal{H}(p) + \mathcal{O}(1)$. It has been numerically confirmed for an 1D case that the LS3 Hessian gives errors of order zero where small cells lie next to larger cells.

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 spurious refinement leads to irregular meshes.

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

$$\overrightarrow{\nabla}_G(q_i) = \frac{1}{V_i} \sum_f q_f S_f \mathbf{n}_f, \tag{16}$$

where the face values q_f are computed with the expression (5) and V_i is the volume of the cell, S_f are the areas of the faces. Also, define a Laplacian smoothing $\mathcal{L}$ as:

$$\mathcal{L}(q_i) = \frac{\sum_f q_f S_f}{\sum_f S_f}. \tag{17}$$

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 $\mathcal{L}$, 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 $(p)_{jk} = \frac{1}{2}((p)_{jk} + (p)_{kj})$,
5. Smooth the Hessian by applying N times $\mathcal{L}$ to each component.

Since the error in the pressure p_h has an oscillatory component of $\mathcal{O}(h^2)$, differenciating this solution creates an oscillatory error of $\mathcal{O}(h)$ in the first and $\mathcal{O}(1)$ in the second derivatives. The smoother $\mathcal{L}$ uses the same type of interpolation to the faces as the Laplace operator in the pressure equation, so it produces $\mathcal{O}(h^2)$ oscillations itself. Therefore, $\mathcal{L}$ cannot increase the smoothness of p_h, which is the reason why the pressure is not smoothed. On the other hand, the smoothing of the gradients (2) is essential. The $\mathcal{O}(h)$ wiggles in $\overrightarrow{\nabla} p_h$ are small compared to the gradient of p so smoothing is very effective for removing these wiggles. The new $\mathcal{O}(h^2)$ oscillations introduced by $\mathcal{L}$ are inoffensive since the gradients are only differenciated once more. In the Hessian, the remaining oscillations of the original solution are of the same order as the solution itself so smoothing cannot improve the accuracy. The step (5) is only applied to create better mesh quality through a smoother criterion.

The resulting Hessian is not second-order accurate but its smoothness makes it interesting as a refinement criterion, since smooth criteria provide good mesh quality. However, smoothing 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.

4 Applications

The grid refinement procedure described above is applied to a propeller-hull interaction configuration. The test case chosen is the STREAMLINE tanker investigated during the European FP7 STREAMLINE project. Model tests have been conducted both with and without the BLAD for ship resistance and self-propulsion for different advancing speeds. Computation will be performed for design speed V=1.773m/s only. As our main interest is propeller performance prediction, a double model computation is performed without taking into account the effect of free-surface. Propeller-hull interaction is simulated with a RANSE/BEM coupling procedure developed in a joint research work between ECN/CNRS and CNR-INSEAN during the STREAMLINE project. The RANSE/BEM coupling approach has been validated in [14] with several test cases. It can provide accurate prediction for propeller-hull interaction problem with much lower cost than a simulation with full RANSE approach where the propeller is also simulated with RANSE solver. The BEM code PRO-INS developed by INSEAN is based on a boundary integral formulation for marine propellers in arbitrary onset non-cavitating and cavitating flow conditions. The RANSE code provides the velocity field in front of the propeller to the BEM code as inflow condition. Propeller loading by BEM is recast as volume force distribution at the propeller plane. This body force representing the propeller action is added as source term in the momentum equations of RANSE simulation.

4.1 Summary of model test results

The BLAD is designed by HSVA [1]. Model test measurements are conducted by CTO [2]. Table 1 summarizes model test results. Rt is the hull resistance measured without the propeller. Other quantities are measured during self-propulsion test. n is the propeller revolution rate at the self propulsion point, T propeller thrust, Q propeller torque, and Fd friction correction force applied to the hull. There are two measurements performed without the BLAD. The first one shown in the second line was performed within the same period as the one including the BLAD, while the second one shown in the third line was conducted at the beginning of the STREAMLINE project. As the experimental condition at different periods might be different, we focus on the comparison of the two measurements performed during the same period shown in the first and the second lines only. Based on those two measurement results, we can see that when the BLAD is added, hull resistance increases by about 0.50N, while under self-propulsion condition, propeller thrust increases by about 1.41N. As those quantities are very small compared with hull resistance (only about 0.8% and 2.3% respectively), it is a very challenging task in a numerical simulation to predict the effect of the BLAD on hull resistance and propeller propulsion with accuracy.

Table 1: Model test results

BLAD	Rt(N)	n(rps)	T(N)	Q(Nm)	Fd(N)
Yes	61.47	9.02	60.63	2.310	13.57
No	60.97	8.98	59.22	2.276	13.63
No(*)	61.10	8.92	58.68	2.253	13.86

(*) Repeated test conducted in different period

4.2 Double model computation with grid refinement

As in the measurement, numerical simulations have been performed for both configurations for resistance and propulsion results. With RANSE-BEM coupling approach, both computations are performed with exactly the same mesh. As free-surface effect is not taken into account in the computation, hull resistance can not be compared with measurement result. However, hull resistance difference between both configurations should be almost the same as in measurement. It is also impossible to perform a self-propulsion computation by adjusting propeller revolution rate. Instead, propeller revolution rate measured in model test is imposed in the computation. Unlike the resistance, propeller thrust and torque obtained in this way by the computation can be compared directly to the measurement result, since the effect of free-surface to propeller propulsion is

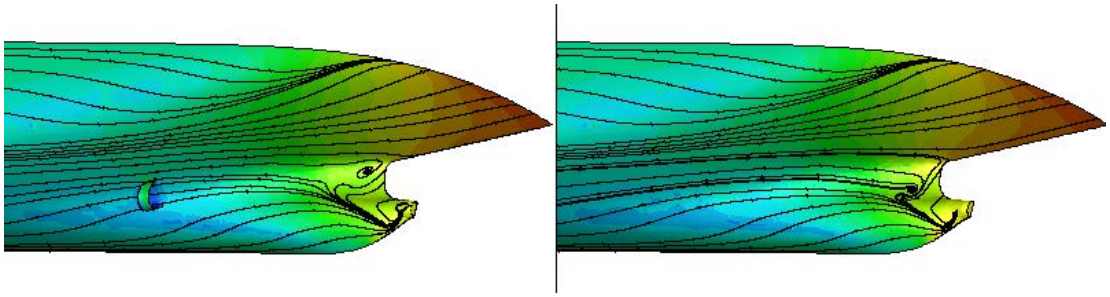

Figure 5: Predicted wall limiting streamlines with and without the BLAD

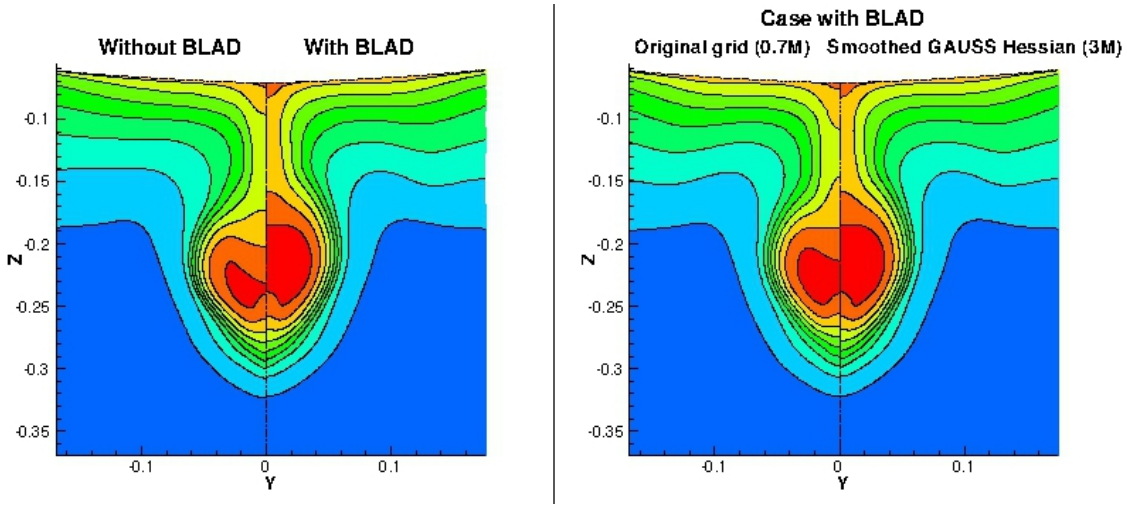

Figure 6: Predicted Nominal wake. Left: effect of the BLAD; Right: effect of grid refinement

expected to be small. Computations were performed with half domain without taking into account propeller tangential force. The initial mesh contains 700K cells and 576K cells respectively for the cases with and without the BLAD. To assess the effect of grid refinement, for each case, we perform grid adaptation for the resistance computation by restricting the refinement in a region between the BLAD and the propeller. The threshold of grid refinement criterion is adjusted such that the refined grid contains 1M, 2M, 3M and 4M cells respectively. Such refined grid is kept unchanged for the computation with RANSE/BEM coupling approach to evaluate the propeller performance. The two different pressure Hessian evaluation methods presented in sub section 3.3 have been tested. The Smoothed Gauss method gives better prediction than the third-order least-squares approach. In this paper, only the results obtained with the Smoothed Gauss method is presented.

The predicted wall limiting streamlines for the case with and without the BLAD are shown in Figure 5. Results are slightly different near the stern. Figure 6 compares the nominal wake at the propeller plane. Picture on the left side compares the case with and without the BLAD, while that on the right side compares the prediction for the case with the BLAD with and without adaptive grid refinement. It can be seen that the predicted nominal wake for the case with the BLAD without using adaptive grid refinement is very similar to the case without the BLAD, which indicates that the effect of the BLAD can not be captured correctly without using adaptive grid refinement.

Predicted hull resistance as well as propulsion results for the case without the BLAD are given in table 2. The effect of grid refinement is mainly observed on propeller thrust. It can be considered that converged solution is obtained with a refined grid containing 2M cells. Compared with model test result, propeller thrust differs by less than 2% for all grids, which can be considered as good prediction for engineering application. Error observed in propeller torque is about the same. Similar results for the case with the BLAD is given in table 3. Again, a very good prediction is obtained on propeller thrust with an error less

Table 2: Predicted results without the BLAD

N cells	Rt(N)	Rsp(N)	T(N)	Q(Nm)
576K	46.01	53.69	60.00	2.322
1M	46.06	53.68	60.19	2.324
2M	46.05	53.66	60.27	2.325
3M	46.04	53.65	60.27	2.325
4M	46.04	53.64	60.22	2.325

Table 3: Predicted results with the BLAD

N cells	Rt(N)	Rsp(N)	T(N)	Q(Nm)
700K	46.47	54.11	60.74	2.348
1M	46.57	54.15	60.86	2.349
2M	46.59	54.11	60.83	2.348
3M	46.58	54.11	60.83	2.348
4M	46.61	54.10	60.81	2.347

than 0.33%. Effect of grid refinement can be observed both on the hull resistance and on the propeller thrust. Like in the previous case, converged solution can be obtained with a refined grid containing 2M cells.

The difference between the two cases on different quantities are presented in table 4. In numerical simulation, the difference is evaluated with the result obtained with similar grid density. We believe that this is a more reliable evaluation, since numerical error might cancel together with such evaluation. For all quantities, converged solution is obtained starting from 2M cells. It is expected that the difference in ship resistance dRt, self-propulsion resistance dRsp, and propeller thrust dT should be almost the same. It is the case for numerical prediction. They differ only by about 0.15% (based on hull resistance value). The difference can be attributed to numerical uncertainty. Concerning measurement data, they differ by about 1.5%. We believe that this gives an indication of measurement uncertainty.

Numerical prediction indicates that hull resistance is increased by 0.54N when the BLAD is added. Compared with the measurement value 0.50N, it differs only by 0.066% based on hull resistance (about 61N). Hence, it is a very good prediction. Self-propulsion resistance Rsp is a direct numerical result in a RANSE/BEM coupling computation for self-propulsion. This quantity can not be measured directly in the experiments. In table 4, we consider the sum of propeller thrust T and friction correction force Fd as self-propulsion resistance for measurement data. The difference in self-propulsion resistance should be about same as the difference in hull resistance. It is really the case in numerical simulation result. But for measurement data, it is more than two times higher. More interesting result is observed on propeller thrust. It is also expected that the difference in propeller thrust under self-propulsion condition should be about the same as the difference in hull resistance. With the initial mesh, the difference in propeller thrust 0.74N is

Table 4: Effect of the BLAD on different quantities

N cells	dRt(N)	dRsp(N)	dT(N)	dQ(Nm)
Initial	0.46	0.51	0.74	0.026
1M	0.50	0.47	0.66	0.025
2M	0.54	0.46	0.56	0.023
3M	0.54	0.45	0.56	0.023
4M	0.57	0.46	0.54	0.023
Exp	0.50	1.35	1.41	0.034

much higher than the difference in hull resistance 0.46N. When the mesh is refined to 2M cells, the difference in propeller thrust changes to 0.56N and the difference in hull resistance changes to 0.54N. It must be noticed that those two quantities are obtained with two different codes, one with the RANSE code, and the other by the BEM code. The fact that they converge to almost the same value as the mesh is refined demonstrates the good convergence behavior of numerical simulation. They become almost the same as expected. The difference in propeller thrust is due to the difference in propeller revolution rate and the difference in the wake with and without the BLAD. It is possible in a numerical simulation to evaluate those effects separately. By applying propeller revolution rate n=9.02rps for the case without the BLAD, propeller thrust increases by 0.84N. Hence, the wake with the BLAD decreases the propeller thrust by 0.84-0.56=0.28N at n=9.02rps. Measurement results indicate that propeller thrust increases by 1.41N when the BLAD is added. This value is almost 3 times the difference observed in hull resistance. We believe that this might be a consequence of measurement uncertainty. We can use propeller open water performance result to make an estimation. Under open water condition, if the propeller advancing speed is chosen (V=1.3209m/s) such that propeller thrust T=59.22N at n=8.98rps (the measurement self-propulsion point without the BLAD), then, at n=9.02rps, propeller thrust increases by 0.78N. In the wake with the BLAD, propeller thrust should decrease by 0.28N according to numerical simulation. Hence, in the case with the BLAD, propeller thrust should increase by 0.78-0.28=0.50N, which is exactly the same as the difference observed in hull resistance. Based on this estimation, we believe that the numerical prediction with 0.54N increase in propeller thrust for the case with the BLAD is a very good prediction. And the expected true value should be 0.50N rather than 1.41N as obtained in the measurement. Similar estimation can be applied to propeller torque to demonstrate that difference observed in model test for propeller torque is likely over-estimated. Based on propeller open water measurement result, propeller torque in open water condition increases by 0.0267Nm when propeller revolution rate changes from 8.98rps to 9.02rps. CFD prediction indicates that propeller torque decreases by 0.004Nm in the wake with the BLAD at n=9.02rps. Hence, the estimated difference in propeller torque is 0.0267-0.004=0.0263Nm. This value is much more closer than the CFD prediction 0.023Nm compared with the measurement result 0.034Nm. Finally, we can notice that the convergence behavior observed with the finest refined grid containing 4M cells is not always very good. This can be explained by the fact that in our grid refinement exercise, a constraint on minimum cell size is applied. With this constraint, 3M cells may be sufficient to refine the mesh in the region where it is really needed. When grid cells increase from 3M to 4M, the additional 1M cells may be added in the region where numerical error is no longer large. This explain why the convergence behavior is not always very good when the grid is refined.

5 Conclusions

The effect of a propeller inflow improving device is evaluated with a RANSE-BEM coupling approach. It has been found that adaptive grid refinement plays an important role in obtaining a reliable numerical prediction, especially for propeller performance, even for the case without such device. While experimental data can provide accurate enough data for hull resistance in a resistance test and propeller revolution rate in a self-propulsion test, other quantities such as propeller thrust and torque measured with an uncertainty of about 1.5% are not accurate enough to assess correctly the effect of such device. On the other hand, with grid refinement using pressure Hessian evaluated by a smooth Gauss method as refinement criterion, numerical prediction with RANSE-BEM coupling approach can provide much more reliable prediction. The present study has been conducted with double model computation using the propeller revolution rate obtained by measurement. It will be interesting in a future study to take into account the effect of free-surface and perform a self-propulsion computation to see if the same result can be obtained.

Acknowledgment

This work was granted access to the HPC resources of IDRIS under the allocation 2014-21308 made by GENCI (Grand Equipement National de Calcul Intensif). Part of the work was supported by the STREAMLINE project, a collaborative R&D project, partly funded by the 7th Framework Program of the European Commission (FP7). The authors are indebted to HSVA for the design of the BLAD, and to CTO for model test results.

References

[1] R. Bensow, D. Hafermann, S. Gatchell, M. Manzke, and Queutey P. Deliverable D21.3 - Design and evaluation of inflow- improving devices. STREAMLINE project report, 2012.

[2] F. Difelice, M. Falchi, G. Aloisio, A. Pecoraro, F. Salvatore, and T. Bugalsky. Deliverable D21.11 - Results of model tests on optimised configurations. STREAMLINE project report, 2013.

[3] P. Queutey and M. Visonneau. An interface capturing method for free-surface hydrodynamic flows. *Comput Fluids*, 36(9):1481–1510, November 2007.

[4] J. Wackers, G. B. Deng, and M. Visonneau. Combined tensor-based refinement criteria for anisotropic mesh adaptation in ship wave simulation. In *Proceedings of ADMOS 2011*, Paris, France, 2011.

[5] B. P. Leonard. The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection. *Comp Meth Appl Mech Engrg*, 88:17–74, 1991.

[6] V. Pržulj and B. Basara. Bounded convection schemes for unstructured grids. In *15th AIAA Computational fluid dynamics conference*, AIAA paper 2001-2593, Anaheim, CA, 11-14 June 2001.

[7] O. Ubbink. *Numerical predictions of two fluid systems with sharp interfaces*. PhD thesis, Imperial College of Science, Technology & Medicine. University of London., January 1997.

[8] J. Wackers, K. Ait Said, G. B. Deng, P. Queutey, M. Visonneau, and I. Mizine. Adaptive grid refinement applied to RANS ship flow computation. In *Proceedings of the 28th Symposium on Naval Hydrodynamics*, Pasadena, California, 2010.

[9] J. Wackers, A. Leroyer, G. B. Deng, P. Queutey, and M. Visonneau. Adaptive grid refinement for hydrodynamic flows. *Comput Fluids*, 55:85–100, 2012.

[10] F. Alauzet and A. Loseille. High-order sonic boom modeling based on adaptive methods. *J Comput Phys*, 229(3):561–593, 2010.

[11] P. L. George and H. Borouchaki. *Delaunay Triangulation and Meshing - Application to Finite Elements*. Hermes, 1998.

[12] A. Loseille, A. Dervieux, and F. Alauzet. Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations. *J Comput Phys*, 229:2866–2897, 2010.

[13] J. Majewski. 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, 2009.

[14] G. Deng, P. Queutey, M. Visonneau, and F. Salvatore. Ship Propulsion Prediction with a Coupled RANSE-BEM Approach. V International Conference on Computational Methods in Marine Engineering, 2013.