Hull shape optimisation using multi-fidelity metamodels and adaptive grid refinement

Jeroen Wackers, Charles-Edouard Jeanson, Patrick Queutey, Michel Visonneau, Riccardo Pellegrini, Andrea Serani, Matteo Diez

Hull shape optimisation using multi-fidelity metamodels and adaptive grid refinement

Jeroen Wackers*, Charles-Edouard Jeanson*, Patrick Queutey*, Michel Visonneau*, Riccardo Pellegrini†, Andrea Serani†, and Matteo Diez†

*LHEEA, Ecole Centrale de Nantes / CNRS-UMR 6598, Nantes/France,

†CNR-INM, National Research Council-Institute of Marine Engineering, Rome/Italy

jeroen.wackers@ec-nantes.fr

1 Introduction

In automatic design optimisation, large numbers of candidate designs need to be evaluated in order to find the best design. If the evaluations are performed with numerical simulation, this may be prohibitively expensive. A way to solve this problem is by using metamodels: a limited number of designs is simulated and a surrogate model for the numerical simulation is created by interpolating between these training points with a suitable technique. The optimisation is then performed over this cheap surrogate model.

The cost of this procedure can be further reduced through multi-fidelity metamodelling. Here, most of the metamodel behaviour is based on relatively cheap, low-fidelity simulations, but these results are corrected with a few expensive high-fidelity simulations. The goal is to obtain a metamodel with the precision of the high-fidelity simulations and the cost of the low-fidelity ones. Different fidelities can be obtained using two separate solvers, such as RANS and potential flow codes. If only RANS is used, the accuracy and cost can be varied by using fine and coarse meshes.

Adaptive grid refinement is a method of creating the mesh during the simulation by locally dividing cells where the flow needs this. By adjusting the global amount of refinement requested, coarse or fine meshes are created in a straightforward manner. Therefore, it is a natural choice to integrate adaptive refinement in an automatic optimisation procedure with multi-fidelity metamodels.

The goal of this paper, a continuation of the work by Pellegrini et al. (2018), is to explore the combination of the multi-fidelity metamodelling techniques developed at CNR-INM with the flow solver ISIS-CFD from ECN – CNRS, which contains adaptive grid refinement. The paper summarises these techniques, focusing on those aspects which are relevant for their combination. Then, a multi-fidelity metamodel-based shape optimisation is presented for the DTMB 5415 destroyer.

2 Adaptive multi-fidelity metamodel

Consider an objective function $f(\mathbf{x})$, i.e. the function to be minimised in the optimisation, where $\mathbf{x} \in \mathbb{R}^N$ is the design variable vector and N the design space dimension. The metamodel prediction $\tilde{f}(\mathbf{x})$ is provided by means of interpolation with stochastic radial basis functions $g(\mathbf{x}, \tau)$, with $\tau \sim \text{unif}[1, 3]$ the stochastic tuning parameter of the metamodel:

$$\widetilde{f}(\mathbf{x}) = \text{EV}\left[g(\mathbf{x}, \tau)\right]_\tau, \quad \text{with} \quad g(\mathbf{x}, \tau) = \sum_{j=1}^{J} w_j \|\mathbf{x} - \mathbf{x}_j\|^\tau, \tag{1}$$

where w_j are unknown coefficients, $\|\cdot\|$ is the Euclidean norm, $\mathbf{x}_j$ are the training points with associated objective function value $f(\mathbf{x}_j)$, and J is the number of training points. The coefficients w_j are determined enforcing the interpolation $g(\mathbf{x}_j, \tau) = f(\mathbf{x}_j)$ by solving $\mathbf{A}\mathbf{w} = \mathbf{f}$, with $\mathbf{w} = \{w_j\}$, $a_{i,j} = \|\mathbf{x} - \mathbf{x}_j\|^\tau$ and $\mathbf{f} = \{f(\mathbf{x}_j)\}$. The uncertainty $U_{\tilde{f}}(\mathbf{x})$ associated with the metamodel prediction is quantified by the 95%-confidence interval of $g(\mathbf{x}, \tau)$, evaluated using a Monte Carlo sampling over τ (Volpi et al., 2015).

The multi-fidelity prediction $\hat{f}(\mathbf{x})$ is defined as a low-fidelity metamodel plus a metamodel of the error between high and low fidelity:

$$\hat{f}(\mathbf{x}) = \tilde{f}_L(\mathbf{x}) + \tilde{\varepsilon}(\mathbf{x}), \quad \text{with} \quad \varepsilon(\mathbf{x}) = f_H(\mathbf{x}) - f_L(\mathbf{x}), \tag{2}$$

where "$\wedge$" indicates the multi-fidelity approximation and f_H and f_L are high- (HF) and low-fidelity (LF) evaluations (Pellegrini et al., 2016). Assuming the uncertainty associated with the low-fidelity and error

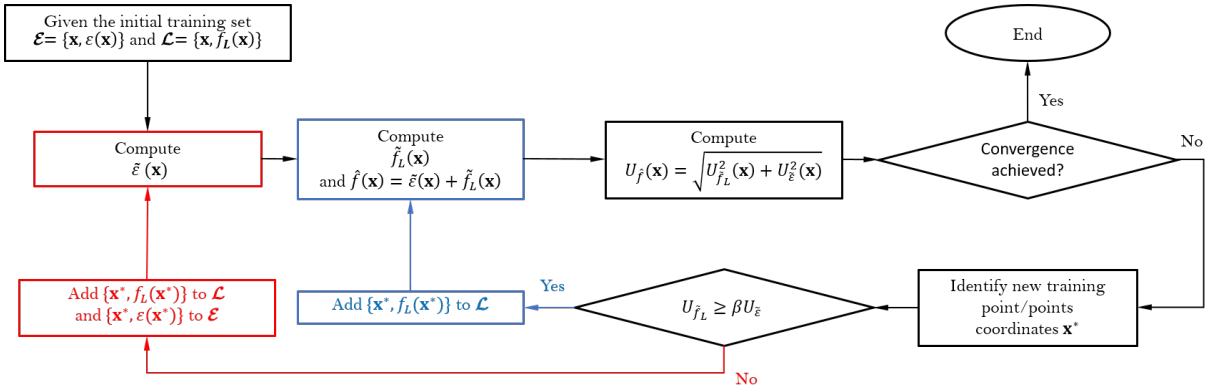

Fig. 1: Adaptive multi-fidelity metamodel updating scheme

metamodels ($U_{\tilde{f}_L}$ and $U_{\tilde{\varepsilon}}$ respectively) as uncorrelated, the uncertainty associated with the multi-fidelity prediction can be defined as:

$$U_{\hat{f}}(\mathbf{x}) = \sqrt{U^2_{\tilde{f}_L}(\mathbf{x}) + U^2_{\tilde{\varepsilon}}(\mathbf{x})}. \tag{3}$$

$\mathcal{L}$ and $\mathcal{E} \subseteq \mathcal{L}$ are the training sets for $\tilde{f}_L$ and $\tilde{\varepsilon}$, respectively. New training points $\mathbf{x}^\star$ for $\mathcal{L}$ and $\mathcal{E}$ are sequentially defined by the adaptive sampling method, as shown in Fig. 1. Once $\mathbf{x}^\star$ is identified, the training sets $\mathcal{L}$ and $\mathcal{E}$ are updated as

$$\begin{cases} \text{If } U_{\tilde{f}_L}(\mathbf{x}^\star) \geq \beta U_{\tilde{\varepsilon}}(\mathbf{x}^\star), & \text{add } \{\mathbf{x}^\star, f_L(\mathbf{x}^\star)\} \text{ to } \mathcal{L}, \\ \text{else}, & \text{add } \{\mathbf{x}^\star, f_L(\mathbf{x}^\star)\} \text{ to } \mathcal{L} \text{ and } \{\mathbf{x}^\star, \varepsilon(\mathbf{x}^\star)\} \text{ to } \mathcal{E}, \end{cases} \tag{4}$$

where $\beta \in [0, 1]$ is the ratio between the LF and HF computational cost. In the first case, only a low-fidelity evaluation is performed, while the second case requires both low- and high-fidelity evaluations for the same $\mathbf{x}^\star$. In this work the aggregate criteria adaptive sampling (ACAS) is applied, it identifies a new training point $\mathbf{x}^\star$ by solving the following single-objective minimization problem:

$$\mathbf{x}^\star = \underset{\mathbf{x}}{\text{argmin}}[\hat{f}(\mathbf{x}) - U_{\hat{f}}(\mathbf{x})]. \tag{5}$$

A deterministic single-objective formulation of the particle swarm optimisation (DPSO) algorithm (Serani et al., 2016) is used for the solution of the minimization problems of Eq. (5). Furthermore, it is used for the multi-fidelity metamodel-based shape optimisation.

3 KLE geometric modeller

The modified geometries ($\mathbf{g}$) are produced by the linear superposition of N orthonormal basis functions ($\boldsymbol{\psi}$) on the original geometry ($\mathbf{g}_0$) as follows

$$\mathbf{g}(\boldsymbol{\xi}, \mathbf{x}) = \mathbf{g}_0(\boldsymbol{\xi}) + \boldsymbol{\delta}(\boldsymbol{\xi}, \mathbf{x}) \qquad \text{with} \qquad \boldsymbol{\delta}(\boldsymbol{\xi}, \mathbf{x}) = \sum_{k=1}^{N} x_k \boldsymbol{\psi}_k(\boldsymbol{\xi}) \tag{6}$$

where $\boldsymbol{\xi}$ are the geometry Cartesian coordinates, whereas $\{x_k\}_{k=1}^N$ and $\{\boldsymbol{\psi}_k\}_{k=1}^N$ are the reduced design variables and the eigenfunctions, respectively, provided by the design-space augmented dimesionality reduction (ADR) procedure described in Serani and Diez (2018). The original design space, formed by $M = 27$ design variables (Serani et al., 2016a), was reduced in dimensionality by the Karhunen-Loève expansion (KLE) (Diez et al. 2015), based on combined geometric and multi-physics/multi-point physical vectors. Design-space ADR by KLE has provided a reduced-order model for the shape modification vector ($\boldsymbol{\delta}$), composed by $N = 14$ reduced design variables ($\mathbf{x}$) retaining at least the 95% of the original design variability. Details about the original design space definition and the ADR method can be found in Serani et al. (2016a) and Serani and Diez (2018), respectively.

4 RANS simulation and meshes for optimisation

Flow simulations are performed with the unstructured finite-volume Navier-Stokes solver ISIS-CFD developed at ECN – CNRS (Queutey and Visonneau, 2007) available in the FINE™/Marine computing suite from NUMECA Int. This section describes the treatment of meshes in this solver, for simulations aimed at shape optimisation.

4.1 Mesh deformation and adaptive refinement

The solver contains an adaptive grid refinement method which adjusts the mesh locally, during the computation, by dividing the cells of an original coarse grid where the solution requires this (Wackers et al., 2017). The decision where to refine comes from a refinement criterion, a tensor field $C(x, y, z)$ computed from the flow. The tensor is based on the water surface position and on second derivatives of pressure and velocity. The mesh is refined until the dimensions $\mathbf{d}_{i,j}$ ($j = 1, 2, 3$) of each hexahedral cell i satisfy:

$$\|C_i \mathbf{d}_{i,j}\| = T_r. \tag{7}$$

As shown in Wackers et al. (2017), if the criterion is not sensitive to grid refinement, then the cell sizes everywhere are proportional to the constant threshold T_r. For multi-fidelity optimisation, this means that high- and low-fidelity results can be obtained by running the same simulations with two different thresholds T_r (Fig. 2), a procedure which is easy to automate in an optimisation loop (Fig. 1).

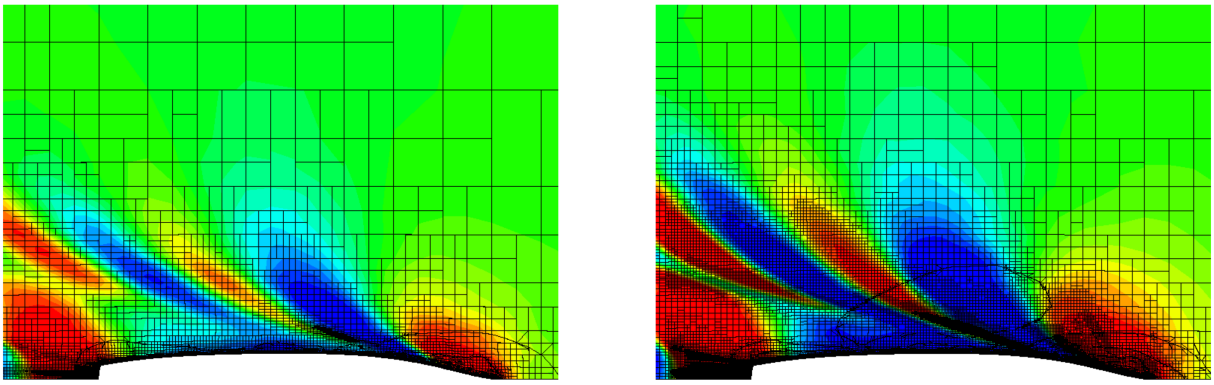

Fig. 2: Fine and coarse meshes are obtained with different thresholds: $T_r = 0.0145$ (left) and $T_r = 0.0072$ (right). Cuts at $z = 0$ coloured with the water volume fraction.

The meshes for the simulation of different geometries in the optimisation process are obtained through mesh deformation. Each simulation starts from the same original mesh (Fig. 3a). The mesh is divided in layers around the hull. For each geometry $\mathbf{g}(\boldsymbol{\xi}, \mathbf{x})$ received from the KLE modeller, the displacement of the hull faces with respect to $\mathbf{g}_0(\boldsymbol{\xi})$ is propagated through these layers (Durand, 2012). The displacements are multiplied with a weighting factor which goes from 1 on the hull to 0 on the outer boundaries, so that the latter are not deformed (Fig. 3b). The original grid is coarse, since deforming these is easier and safer than for fine grids. The final grid, including all the refinement at the free surface, is created using adaptive refinement (Fig. 3c).

4.2 Adaptation for drag computation

A challenge for this work is performing computations with adaptive grid refinement whose sole purpose is the accurate computation of drag. Contrary to simulations meant for studying flow details, locally very fine meshes may not be required. On the other hand, mesh regularity and a low number of cells are essential. Here, some of the ongoing work on improving adaptation for drag computation is shown.

Improving the free-surface meshing To provide adequate surface capturing even on deformed meshes, the fine grid around the free surface is created entirely by adaptive refinement (Fig. 3c). This implies that the original cells have to be divided many times. On undeformed grids (Fig. 4a) this works well: where the surface is at rest, the mesh is refined only in vertical direction and the mesh quality is good. When

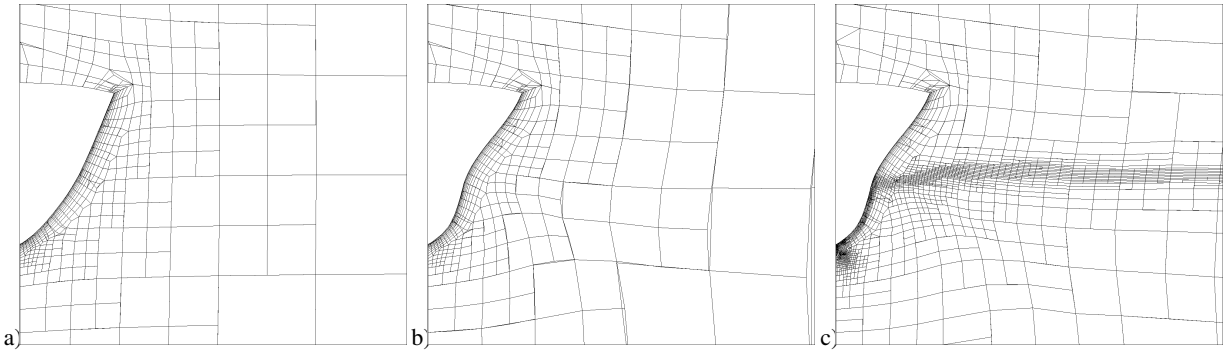

Fig. 3: Mesh creation: original grid (a), after deformation (b), after adaptive refinement (c).

the mesh is deformed (Fig. 4b) the cells are stretched and rotated so they are refined in more than one direction, which leads to unnecessary refinement and mediocre grid quality.

This problem was alleviated by changing the weigting factor. Instead of the Durand (2012) weighting which is based on the distance to the body and this distance squared, a weighting was chosen which is based on the distance to the power 1.6. This induces less deformation of the grid far from the body, which improves the refined mesh (Fig. 4c).

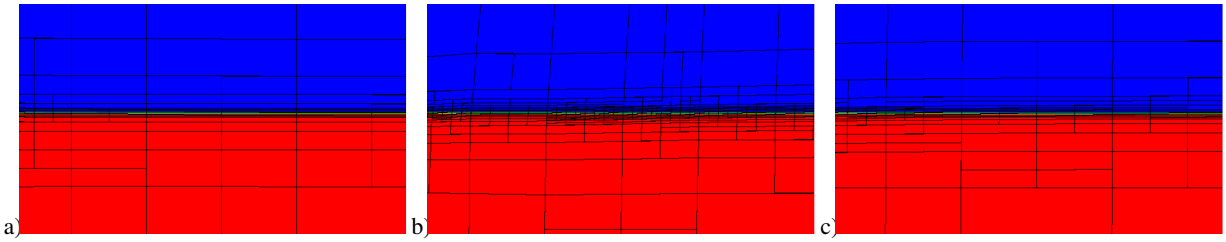

Fig. 4: Undisturbed free-surface mesh detail at $x = 5$: no deformation (a), deformation, old weighting (b) and power 1.6 weighting (c).

Limiting refinement behind the stern Unrestricted application of the velocity / pressure Hessian leads to the accurate resolution of the near and the far wake (Wackers et al., 2017a). This may not be required for drag evaluation. Therefore, tests were run where horizontal refinement is forbidden from a certain distance behind the stern. These results are compared with an estimation of the numerical uncertainty (based on three grids) following Wackers et al. (2017). Fig 5 shows the results for three geometries.

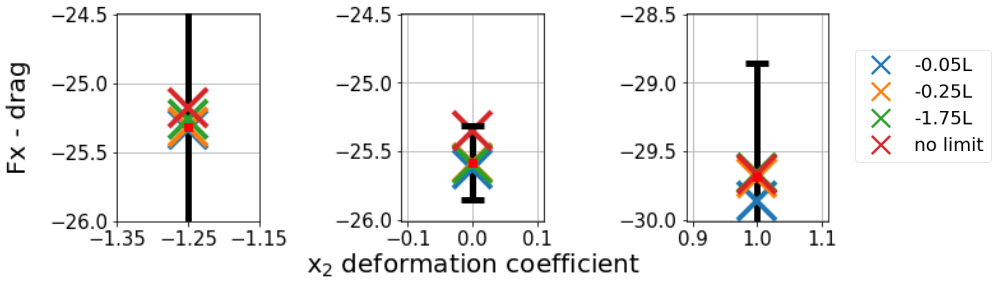

Fig. 5: Drag values when horizontal refinement is forbidden beyond a limit behind the stern, compared with estimated uncertainties on the $-0.25L$ grids, for three geometries.

The uncertainties are bigger for deformed grids than for the undeformed case ($x_2 = 0$). On the other hand, the dependence on the refinement limit position is small, with two exceptions: imposing no limit is a bad idea, since the waves are not damped before the outflow boundary. And the shortest box ($0.05L$) changes the drag when the mesh is deformed. Thus, the resolution of the near wake ($0.25L$) is required for good drag estimation; this also has implications for computations without adaptive refinement.

5 Optimisation of the DTMB 5415 destroyer

5.1 Optimisation settings

The DTMB 5415 is optimised for minimal resistance at even keel, $Fr = 0.30$ and $Re = 1.18 \cdot 10^7$. No constraints apply. The two design variables are the first parameters x_1 and x_2 of the KLE expansion. Initial sensitivity studies showed minimal drag for large negative values of these parameters, so a wide range of $[-1.25, 1.25]$ is used for each parameter. For the initial sample plane, HF and LF simulations were run in the centre of the domain, and with each single parameter at either $+1$ or -1. This results in 5 HF and 5 LF simulations. An optimisation with the ACAS sampling criterion was then run, adding 2 HF and 59 LF points.

5.2 Simulation settings

Simulations are performed on half geometries. The domain runs from $1.5L$ in front of the bow to $3L$ behind the stern, up to $2L$ laterally, and from $-1.5L$ to $0.5L$ vertically. Dirichlet conditions on the velocity are imposed on the inflow and side faces, pressure is imposed on the top, bottom, and outflow side. The hull is treated with a wall law and $y+ = 60$ for the first layer. Turbulence is modelled with $k - \omega$ SST.

The initial mesh has 130k cells; the thresholds for the low- and high-fidelity simulations are $T_r = 0.0145$ and $T_r = 0.0036$ which implies a $4:1$ cell size ratio and results in approximately 260k and 4.3M cells respectively. On a 20-core workstation the computations take about 1.5 and 24 hours each.

5.3 Results

Fig. 6 shows the sampled points and the metamodels at the end of the optimisation. After the initialisation, the adaptive sampler first targets the corners of the design space, then quickly focuses on the region of the optimum. To ensure the correctness of the optimum, a few HF simulations are performed in this region towards the end of the optimisation. The LF simulations appear reliable, since the LF and multifidelity metamodel are similar. However, the error metamodel is not negligible.

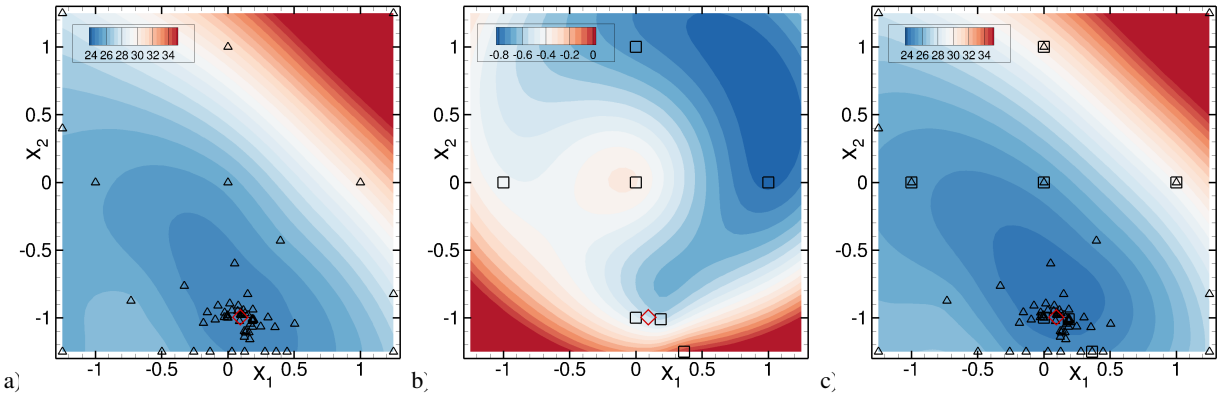

Fig. 6: Low-fidelity $\tilde{f}_L$ (a), error $\tilde{\varepsilon}$ (b) and combined multi-fidelity $\hat{f}$ (c) metamodels. The symbols indicate simulations, Δ: LF, $\square$: HF. The red $\diamond$ is the computed optimum.

The computed optimum is at $x_1 = 0.095$, $x_2 = -0.995$. In this point, the metamodel-predicted drag is 23.98N, the actual drag computed with HF simulation is 24.13N (+0.6%). The original geometry has a HF drag of 25.5412N, which means a gain of 4.9% for the optimisation.

Fig. 7 shows the initial and optimised geometries. The optimum has a more slender aftship which reduces the width of the stern wave, and a slight bulge aft of the bow. This bulge creates a second bow wave out of phase with the first one. These two waves cancel so the total bow wave is reduced. Since this interference depends on the wave lengths, it is expected to be only effective around the target velocity.

6 Conclusions

This work has shown that multi-fidelity metamodelling and adaptive grid refinement function together for the optimisation of realistic ship geometries. Future work will focus on further improving the efficiency of the CFD and on reducing the sensitivity of the metamodel to numerical errors in the simulations.

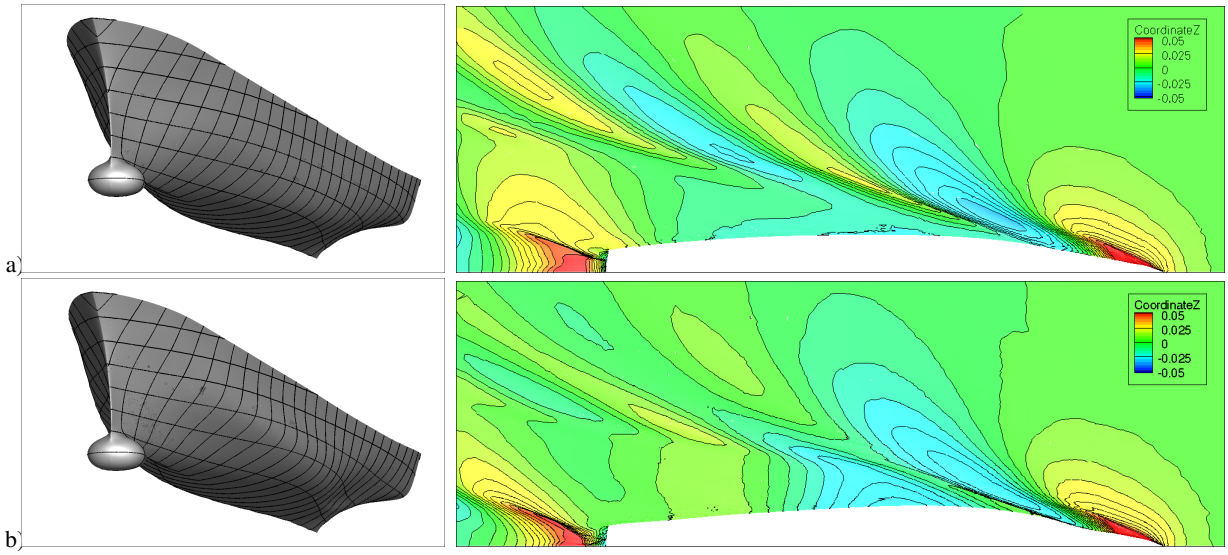

Fig. 7: Initial (a) and optimised (b) geometry and wavepattern.

Acknowledgments

The CNR-INM work is partially supported by the US Department of the Navy Office of Naval Research Global, NICOP grant N62909-15-1-2016 and N62909-18-1-2033, administered by Dr. Woei-Min Lin, Dr. Salahuddin Ahmed, and Dr. Ki-Han Kim, and by the Italian Flagship Project RITMARE, funded by the Italian Ministry of Education. The research is performed within NATO STO Task Group AVT-252 "Stochastic Design Optimization for Naval and Aero Military Vehicles".

References

M. Diez, E.F. Campana and F. Stern (2015). Design-space dimensionality reduction in shape optimization by Karhunen-Loève expansion. *Comput Methods Appl Mech Engrg*, **283**, 1525–1544.

M. Durand (2012). *Interaction fluide-structure souple et légère, application aux voiliers*. Ph.D. thesis, Ecole Centrale de Nantes.

R. Pellegrini, U. Iemma, C. Leotardi, E.F. Campana, and M. Diez (2016). Multi-fidelity adaptive global metamodel of expensive computer simulations. *2016 IEEE Congress on Evolutionary Computation (CEC)*, 4444–4451.

R. Pellegrini, A. Serani, M. Diez, J. Wackers, P. Queutey, and M. Visonneau (2018). Adaptive sampling criteria for multi-fidelity metamodels in CFD-based shape optimization. *ECCM-ECFD*, Glasgow, Scotland.

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

A. Serani, C. Leotardi, U. Iemma, E.F. Campana, G. Fasano, and M. Diez (2016). Parameter selection in synchronous and asynchronous deterministic particle swarm optimization for ship hydrodynamics problems. *Applied Soft Computing*, **49**, 313–334.

A. Serani, G. Fasano, G. Liuzzi , S. Lucidi, U. Iemma, E.F. Campana and M. Diez M. (2016a). Ship hydrodynamic optimization by local hybridization of deterministic derivative-free global algorithms. *Applied Ocean Research*, **59**, 115–118.

A. Serani and M. Diez (2018). Shape Optimization under Stochastic Conditions by Design-space Augmented Dimensionality Reduction. *19th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Atlanta, USA.

S. Volpi, M. Diez, N.J. Gaul, H. Song, U. Iemma, K.K. Choi, E.F. Campana, and F. Stern (2015). Development and validation of a dynamic metamodel based on stochastic radial basis functions and uncertainty quantification. *Structural and Multidisciplinary Optimization*, **51**(2), 347–368.

J. Wackers, G.B. Deng, E. Guilmineau, A. Leroyer, P. Queutey, M. Visonneau, A. Palmieri, and A. Liverani (2017). Can adaptive grid refinement produce grid-independent solutions for incompressible flows? *J Comp Phys*, **344**, 364–380.

J. Wackers, E. Guilmineau, and M. Visonneau (2017a). Unsteady behaviour in RANS simulation of the JBC and KVLCC2. *NuTTS 2017*, Wageningen, The Netherlands.