Numerical optimization methods for ship hydrodynamic design

Emilio Fortunato Campana

Italian National Research Council

124 PUBLICATIONS **909** CITATIONS

SEE PROFILE

Daniele Peri

Italian National Research Council

68 PUBLICATIONS **431** CITATIONS

SEE PROFILE

Yusuke Tahara

National Maritime Research Institute

30 PUBLICATIONS **270** CITATIONS

SEE PROFILE

Manivannan Kandasamy

Flow Inc.

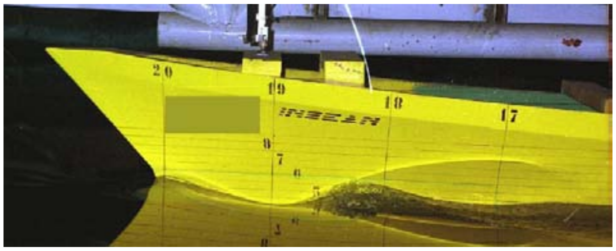
36 PUBLICATIONS **175** CITATIONS

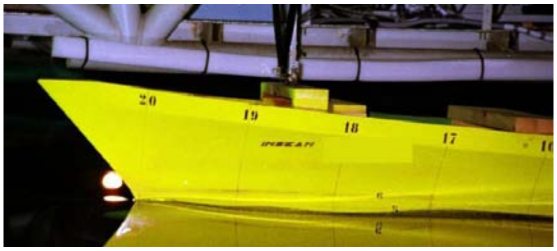
SEE PROFILE

Numerical Optimization Methods for Ship Hydrodynamic Design

Emilio F. Campana[1] **(V)**, **Daniele Peri**[1] **(V)**, **Yusuke Tahara**[2] **(M)**, **Manivannan Kandasamy**[3] **(V)**, and **Frederick Stern**[3] **(M)**

[1]INSEAN — Italian Ship Model Basin, Rome, Italy
[2]NMRI — National Maritime Research Institute, Tokyo, Japan
[3]IIHR — Hydroscience & Engineering, University of Iowa, USA





The use of computational methods in design engineering is growing rapidly at all stages of the design process, with the final goal of a substantial reduction of the cost and time for the development of a design. Simulations and optimization algorithms can be combined together into what is known as Simulation-Based Design (SBD) techniques. Using these tools the designers may find the minimum of some user defined objective functions with constraints, under the general mathematical framework of a Non-Linear Programming problem. There are problems of course: computational complexity, noise, robustness and accuracy of the numerical simulations, flexibility in the use of these tools; all these issues will have to be solved before the SBD methodology can become more widespread. In the paper, some derivative-based algorithms and methods are initially described, including efficient ways to compute the gradient of the objective function. Derivative-free methods — such as genetic algorithms and swarm methods—are then described and compared on both algebraic tests and on hydrodynamic design problems. Both local and global hydrodynamic ship design optimization problems are addressed, defined in either a single- or a multi-objective formulation framework. Methods for reducing the computational expense are presented. Metamodels (or surrogated models) are a rigorous framework for optimizing expensive computer simulations through the use of inexpensive approximations of expensive analysis codes. The Variable Fidelity idea tries instead to alleviate the computational expense of relying exclusively on high-fidelity models by taking advantage of well-established engineering approximation concepts. Examples of real ship hydrodynamic design optimization cases are given, reporting results mostly collected through a series of projects funded by the Office of Naval Research . Whenever possible, an experimental check of the success of the optimization process is always advisable. Several examples of this testing activity are reported in the paper—one is illustrated by the two pictures at the top of this page, which show the wave pattern close to the sonar dome of an Italian Navy Anti-Submarine Warfare corvette: left, the original design; right, the optimized one.

INTRODUCTION TO SIMULATION-BASED DESIGN METHODS

Optimization means much more than *improvement*. Nevertheless, many researchers and design engineers still employ the terminology *optimization* when what they mean in practice is that after starting from a non-satisfactory configuration, they have tried two or three other ones and chosen at the end the best one. This is undoubtedly related to optimization, but in a minimal sense. In the present paper optimization means the use of some minimization algorithm to find the best possible solution (*in some defined sense*) constrained by appropriate conditions. More specifically, we will focus on optimization methods driven by numerical simulations applied to the optimal hydrodynamic design of ships.

The growing reliance on rigorous computational methods in engineering at all stages of the design process seems to be an irreversible process, the final goal being to reduce the time and cost of both design development and standard testing activities (which currently are very time consuming and expensive to run). By using computational methods, candidate designs are numerically evaluated to establish their respective merits. To date, though, this way of using numerical simulations has had a more dramatic impact on the design *process*, rather than on the design *per se*, which still relies greatly on the experience of the designer and on heuristic procedures *i.e.*, on the *art* of engineering. Additionally, the complexity of modern engineering systems makes the use of heuristic methods alone increasingly challenging: "Radically new designs present a difficult problem because designers cannot rely on historical databases. Moreover, some design areas experience a loss of immensely valuable design knowledge with the retirement of designers. There is also a realization that meeting a minimal set of requirements may not suffice to ensure success of new designs." (Alexandrov [1]).

The above drawbacks make the use of Simulation-Based Design (SBD) techniques—which combine (i) simulations, (ii) optimization algorithms and (iii) grid and geometry deformation methods—much more feasible. Design performance feeds an algorithm capable of finding the minimum of some user-defined objective functions and constraints, under the general mathematical framework of a Non-Linear Programming problem. The design performance is evaluated by adopting some numerical code chosen by the user. What enables the pursuit of SBD is the development of better numerical models of the governing disciplines, faster optimization algorithms, and the ever increasing computational capacity. In addition to simulation codes and optimization algorithms, a third element is fundamental in the development of a SBD framework for shape optimization—a geometry-modeling method that provides the necessary link between the design variables (and their variations) and the deformation of the body shape. When an analysis tool is based on the solution of a Partial Differential Equation (PDE) on a volume grid surrounding a complex geometry, developing a geometry-modeling method can be a difficult task and require attention to detail. Flexibility of the modeling method may greatly affect the freedom of an optimizer to explore the design space.

The aim of this paper is to highlight some important factors of the many relevant issues in the context of SBD applied to ship hydrodynamic design, rather than give a detailed review of SBD methods. Historically, in the naval hydrodynamic context the focus was initially on minimizing total resistance, which was evaluated by using wave resistance via the thin-ship theory plus a term that attempted to account for the frictional resistance. The papers by Webster & Wehausen [102] and Lin, *et al.* [52] seem to be among the earliest papers dealing with the problem of producing three-dimensional, ship forms of low resistance. The ship geometry was approximated via finite Fourier series and the minimization approach was based on the method of Lagrange multipliers [30]. Baba [4] proposed a method based on the knowledge of the wave spectrum of the initial ship design and on the successive superposition of a thin ship which would reduce the wave-making resistance. Using a thin-ship approximation and the method of Lagrange multipliers, Hsiung [38] and Hsiung & Shenyan [39], initially solved the problem for wave resistance and subsequently for total resistance, using the ITTC 1957 correlation line formula to add the viscous part, which resulted in a "hammer-head shark"form similar to those obtained by Lin, *et al.* [52]. Some experiments were also carried out on the optimized shapes.

Salvesen, *et al.* [83], Papanikolaou & Androulakakis [68] and Scragg, *et al.* [85] optimized SWATH hull forms using the Lagrange multipliers method. Salvesen, *et al.* [83] estimated the viscous contribution to the resistance by using a simple boundary-layer approach. Additionally, Papanikolaou & Androulakakis [68] and Scragg, *et al.* [85] carried out experiments on the optimized hulls to establish the success of the optimization procedure; the hull form of Scragg, *et al.* [85] was actually built full scale.

In the late 1980s, Pironneau [79] and Jameson

NOMENCLATURE

B	= Ship's beam
C_f	= Friction resistance coefficient
C_t	= Total resistance coefficient
C_t^*	= Adjoint total resistance coefficient
C_w	= Wave resistance coefficient
c_1, c_2	= Cognitive and social coefficients
f_i	= i-th Objective function
G	= Green function
GM	= Ship's metacentric height
g	= Acceleration of gravity
g_w	= $\in [0, 1]$, Particle's inertia decrease rate
N_{body}	= Number of panels on the ship hull surface
N_{fs}	= Number of panels on the free-surface
N_{tot}	= Total number of panels
p	= Dynamic pressure
p^i	= Particle's best minimum
p^b	= Swarm's best minimum
$\Re^N$	= real N-dimensional space
r_1, r_2	= Random coefficients
r_{PQ}	= Distance between P and Q
S	= Feasible solution set
$\mathcal{S}$	= Boundary of the computational domain
S_i	= Surface of the i-th panel
U	= Ship's speed
u_i	= Velocity components, in the x-, y-, z-directions
$\mathbf{v}$	= Velocity of a point on the body relative to $Oxyz$ coordinate system
F_r	= Froude Number $= U/\sqrt{gL}$
R_e	= Reynolds Number $= UL/\nu$
v_k^i	= Velocity of the i-th particle of the swarm at the k-th step
$\mathbf{X}$	= Design variable vector
x, y, z	= Coordinate system, with x aft, y to starboard, and z upward, coordinate of a field point
x_k^i	= i-th particle of the swarm at the k-th step
w_k	= Particle's inertia at the k-th step
$\beta(x)$	= Correction factor for Variable Fidelity
Δ	= Ship's displacement
δx_i	= Finite perturbation on the i-th design parameter
γ	= Variogram of the objective function
λ^i	= Weights in the Kriging metamodel
λ^l	= Lagrangian multipliers of the penalty function
ξ	= Vector of the N design parameters
ρ	= Fluid density
$\sigma(Q)$	= Source strength at a generic point Q
Φ	= Total Velocity potential
φ	= Perturbation potential
$\phi_L(x)$	= Low Fidelity model of the objective function
$\phi_H(x)$	= High Fidelity model of the objective function
χ	= Constriction factor

ACRONYMS

AM	= Adjoint method
BCGA	= Binary Coded Genetic Algorithm
BC-MOGA	= Binary Coded Multi Objective Genetic Algorithm
CAD	= Computer Aided Design
CFD	= Computational Fluid Dynamics
CG	= Conjugate Gradient
DACE	= Design and Analysis tool for Computer Experiments
DDFPSO	= Deterministic Derivative-Free Particle Swarm Optimization
DOE	= Design Of Experiment
FD	= Finite Differences
FEM	= Finite Element Method
FFD	= Free Form Deformation
GA	= Genetic Algorithm
GO	= Global Optimization
HF	= High Fidelity
IGES	= Initial Graphics Exchange Specification
LF	= Low Fidelity
LU	= Lower Upper matrix factorization
MDO	= Multidisciplinary Design Optimization
MODPSO	= Multi-Objective Deterministic Particle Swarm Optimization
MOGA	= Multi Objective Genetic Algorithm
MLIM	= Multidimensional Linear Interpolation Method
MPI	= Message Passing Interface
NURBS	= Non-Uniform Rational B-Spline
OA	= Orthogonal Array
PDE	= Partial Differential Equation
PSO	= Particle Swarm Optimization
RCGA	= Real Coded Genetic Algorithm
RC-MOGA	= Real Coded Multi Objective Genetic Algorithm
RAO	= Response Amplitude Operator
RANS	= Reynolds-Averaged Navier-Stokes
SAM	= Sensitivity Analysis Method
SBD	= Simulation Based Design
SEM	= Sensitivity Equation Method
SQP	= Sequential Quadratic Programming
UDS	= Uniformly Distributed Sequences
UNDX	= Unimodal Normally Distributed Crossover
URANS	= Unsteady Reynolds-Averaged Navier-Stokes
VFM	= Variable Fidelity Method

[42], developed an adjoint optimization method, a powerful tool for shape optimization using the control theory of systems constrained by partial differential equations. Early attempts to adopt this efficient, gradient based optimization technique in ship hydrodynamic optimizations, using potential flow solvers, are reported in Huan & Huang [40], Valorani, *et al.* [99, 100] and Ragab [80]. Löhner [55] and Martinelli [61] presented numerical results in ship hydrodynamic design using an adjoint formulation for an Euler analysis. The adjoint approach was extended to incompressible RANS flow by Martinelli & Cowles [60].

In developing SBD techniques there are good reasons to adopt optimization algorithms that are not based on gradient information, *e.g.*, lack of robustness of the gradient information, multi-objective formulations, unavailability of the source code used for the analysis. Using an analysis tool based on potential flow, Day & Doctors [19] introduced a genetic algorithm to solve a global optimization problem. More complex analysis tools, more sophisticated optimization algorithms (e.g. multiobjective problems), and more complex and realistic problems, have then been introduced by numerous authors, *e.g.* [73, 78, 96, 65, 24, 106, 50, 9][4], demonstrating the increasing consideration given to shape optimization techniques in the design of efficient ships.

This paper will describe some algorithms and methods for the numerical optimization of a ship's performance, for either local or global optimization problems, with both single- or multi-objective functions. Through a series of ONR projects, the authors of this paper have been working on and accumulating experience in the application of optimization techniques to find solutions to complex ship hydrodynamic design problems [95, 96, 13].

There are many optimization algorithms available and many methods are appropriate only for certain types of problems. Typically, optimization problems are classified according to the mathematical characteristics of the objective function, the constraints and the design variables [81]. Ship hydrodynamic design problems are often nonlinear in the objective function and in the constraints.

We start with a brief introduction to some algorithms for gradient-based optimization methods for which rigorous convergence proofs exist. These methods require the calculation of the gradient and/or Hessian (or approximate Hessian) of the objective function. These gradients are obtained by approximating their elements by finite differences. More efficient ways of computing the sensitivity derivatives of the objective function are reported (*i.e.*, the Sensitivity Equations Method and the Adjoint Method) and the relative performances of these approaches for the solution of the design optimization of a tanker are compared. Evolutionary algorithms are then illustrated by either genetic or swarm methods. These approaches do not require gradient information, which represents an advantage due to the numerical noise typically present when the objective function is evaluated numerically.

Statistical metamodels (surrogate models) are presented in the next section. Metamodels are usually developed to replace complex, time-consuming simulation programs or expensive physical experiments, and to facilitate fast and accurate analysis. Various metamodeling methods have been reported in the literature, each having its strong and weak points.

The third section of this paper is devoted to the illustration of global optimization strategies to deal with multi-objective design problems, with the focus on reducing the number of objective function evaluations needed in the approximation of the Pareto front. As with most technological problems, an optimal ship design is a multi-objective problem, with the improvement of a specific aspect of the global design usually causing the worsening of other design aspects.

In section four of the paper, computational models of varying fidelity are presented. Variable fidelity procedures may be obtained by changing the physics, the grid density or the computational accuracy. If the ability of the low-fidelity model to guide the optimization process is monitored and its quality is improved when necessary, this approach may result in a substantial reduction of CPU time without losing optimization accuracy. A number of examples and applications are reported, either for single or multi-objective design problems.

Next, Multidisciplinary Design Optimization (MDO) and Robust Design are briefly introduced in section five. MDO problems arise when the performance of a large-scale, complex system *like a ship* can be affected through the optimal design of several smaller functional units or subsystems. Simultaneous simulations of several interacting field problems (fluid mechanics, heat transfer, elasticity, electromagnetism, etc.), may be involved in the design. MDO can be used to properly formulate the optimization of a system involving these diverse disciplines. Engineers increasingly rely on computer simulation to develop new products and to understand emerging technologies. In practice, this process is

[4] This is only partial list as it would be futile to try to cover all the recent applications of SBD to ship design in general.

permeated with uncertainty: manufactured products deviate from designed products; actual products must perform under a variety of operating conditions. In recent years, engineers have become increasingly concerned with managing these uncertainties, and Robust Design provides tools for dealing with the uncertainties. Statistical decision theory, specifically the Bayes principle, provides a conceptual framework for quantifying uncertainty. The difficulty with exploiting this framework is computational, involving the numerical integration of expensive simulation outputs with respect to uncertain quantities.

ALGORITHMS FOR SIMULATION-BASED DESIGN

In this section we will briefly introduce some basic mathematical frameworks in which the optimization problems we are typically dealing with may be defined and solved. For a general overview and detailed introduction to continuous, global or multiobjective optimization see for instance [30], [98], and [92], respectively.

The basic unconstrained optimization problems can be defined as follows[5]:

$$\underset{\mathbf{X}\in\Re^N}{\text{minimize}} \quad f(\mathbf{X}),$$

where $f : \Re^N \to \Re$ is the objective function, continuous or not, and N is the dimension of the *design variable vector* $\mathbf{X}$. However, typical engineering problems in design involve both geometrical (e.g. maximum dimensions) and functional (e.g. some design's performance) constraints, transforming the original, unconstrained problem into a constrained one, that is:

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{X}) \\ \text{subject to} \quad & g(x) = 0 \\ & h(x) \leq 0, \end{aligned} \tag{1}$$

where $g(x)$ and $h(x)$ are the equality and inequality constraints, respectively. This problem, Eq. (1), can be rewritten as:

$$\begin{aligned} \text{minimize} \quad & f(\mathbf{X}) \\ \text{subject to} \quad & \mathbf{X} \in \mathcal{S}, \end{aligned}$$

where $\mathcal{S} \subseteq \Re^{\mathcal{N}}$ is the feasible solutions set, resulting in a subset of the space $\Re^N$ after deleting the portion of $\Re^N$ prohibited by the constraints. Examples on the role of the constraints on the definition of the problem and on the selection of the proper algorithm to find the solution are reported in Appendix A. In the following, among the plethora of available algorithms, some gradient-based and gradient-free methods are briefly introduced.

Gradient-Based, Local Optimization Methods

The interest in gradient-based, local optimization methods, for which rigorous convergence proofs exist (under some assumptions regarding the objective function), is due to fast convergence rates offered by first- and second-order methods, based on the knowledge of the gradient and/or Hessian (or approximate Hessian) of the objective function. Using local information about the objective function, this class of methods is able to detect a local minimum, that is, the closest minimum with respect to the starting point. Since the objective function is usually computed by using a numerical tool whose source codes are not available (or hardly modifiable!), these vector and matrix quantities are generally obtained by approximating their elements by *Finite Differences* (FD). In the FD method, the design parameters are perturbed (around the current design x^k), one at a time by δx_i, according to some centered difference scheme; the objective function is then computed by feeding the perturbed values of the design parameters to the flow solver, which is treated as a black box. By following this procedure, an approximation of the gradient of the objective function is obtained at each step and the number of objective function evaluations needed is then proportional to the number, N, of design parameters (typically, $2N + 1$), which may be not small. Furthermore, the effect of different amplitudes of the finite perturbations, δx_i, over the accuracy of the cost function gradient has to be investigated ([100]).

A widely investigated alternative is the use of local models (first or second order) of the objective function. These models are typically linear or polynomial, derived on the basis of a limited number of objective function evaluations. If the training points are properly selected, they are also able to smooth out the numerical noise that is inevitably produced by the numerical solvers when evaluating a design.

This is the basis of the Sequential Quadratic Programming (SQP) algorithms (see for example [30]), whose main characteristic is the generation of a sequence of second-order models, utilized to approach the solution of the optimization problem. That the interpolating set be well posed is one of the crucial points of the method. In fact, the number of coefficients of a second-order polynomial model is

[5] The case of the minimization of all the objective functions is depicted here, the maximization of the generic f being equivalent to the minimization of $-f$

$[(N+1)(1+N/2)]$, so that the computational cost is higher by a factor of about $[1/2+(N+1)/4]$ compared with a simple FD approach. On the other hand, if we try to use some of the previously computed points (in order to reduce the number of new objective-function computations), problems rapidly arise as to how well posed the system is. That is, the faster convergence is paid for by a larger effort in the generation of the local model, and these difficulties rapidly increase with the dimension of the problem.

Optimal Shape Design of the Sonar Dome of a Surface Combatant

An example of the application of a local linear model for the computation of the first derivative of the objective function is reported in [75], and the results are illustrated in the two pictures shown on the first page of this paper. The problem to be solved was that of a naval combatant in an anti-submarine mission, with the sonar fitted in the bulbous bow. During the early missions of the ship, a hydrodynamic noise problem was discovered: due to the presence of the large sonar dome, a high, steep bow wave was observed at the patrol speed (10–12 knots), and, depending also on the sea conditions, wave breaking was cyclically produced causing hydrodynamic noise; therefore reducing the capability of the ship to fulfil her mission. The final goal was, of course, to avoid breaking of the bow wave, and a conjugate gradient algorithm was adopted for the optimization of he bow and the sonar dome. The local gradient was computed by using a local linear model of the objective function: $N+1$ points were distributed in the design space, in a small area around the current design, allowing simultaneous variation of all the design variables (while FD was analyzing only variations of a single design variable at a time). Then a least-squares problem was solved obtaining the best local linear approximation of the objective function. The derivatives were the angular coefficients of the resulting hyper-plane. In Fig. 1 the original dome is compared with the optimal (the wave profile and the iso-contours of the pressure coefficients are also shown). The numerical analysis of the wave pattern clearly shows an improvement at the speed of 12 knots (Fig. 2). A reduction of about 78% was predicted. The experimental evidence of the final improvements (the two pictures on the front page) supports the observation that a local minimization is able to produce large effects on the behavior of a system.

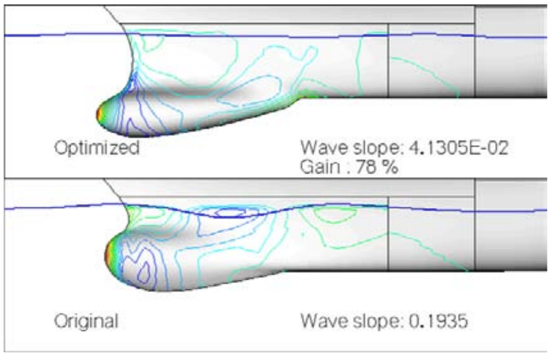


Fig. 1 Original (bottom) and optimized (top) sonar dome. The objective function to be minimized was the maximum slope of the bow wave.

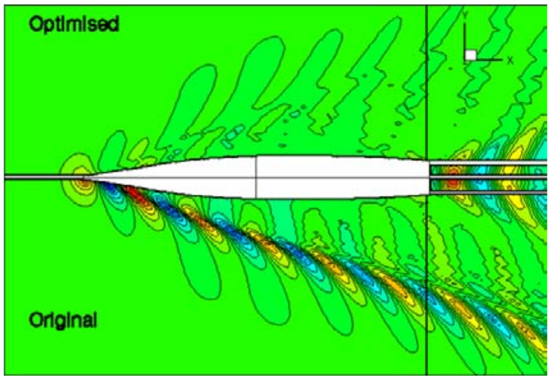


Fig. 2 Wave pattern generated by the combatant at the speed of 12 knots: original (bottom) and optimized (top) sonar dome.

Sensitivity Analysis Methods

As already mentioned, the computation of the gradient is always an expensive task and represents a major limitation in the solution of realistic optimization problems, particularly when the number of design variables is not small. There exist much more efficient ways of computing the sensitivity derivatives of the objective function: however, it has to be underlined here that all these methods require a large manipulation of the code adopted for the analysis, which therefore has to be available. Furthermore, if a different objective function or different constraints are required, the source code has to be modified again.

A relatively recent approach computing the sensitivity derivatives is the use of automatic differentiation tools (see e.g. [32]): by using this software (see [28, 29] for applications), if the CFD solver (adopted for the analysis) is properly coded, derivatives are provided at the end of the run together with the objective function value. This software requires some precise structures of the code in order to operate correctly, and the source code must be rewritten to some extent: the code preparation is not simple at all.

An alternative approach consists in computing the derivatives of the objective function directly from the continuous form of the PDE to be solved by the CFD code. This methodology is normally referred to as *Sensitivity Analysis Method* (SAM). SAM exploits the existence of a set of partial differential equations describing the flow dynamics to predict the sensitivity of the flow field to perturbations of the design parameters. To achieve this goal SAM only requires one flow solution evaluated at the unperturbed design plus a number of additional linear system solutions. This allows a drastic reduction in the number of flow solutions: from $2N$ to just 1, plus the solution of the additional linear systems. These approaches become more and more profitable as the number of design parameters increases.

In the SAM framework, two alternative methods exist: the *Sensitivity Equations Method* (SEM) (a direct formulation of the sensitivity equations) and the *Adjoint Method* (AM). The former [33] involves differentiating the state equations and using the chain rule to find sensitivity derivatives of the cost function: this method requires only one flow solution and the solution of N sets of linear sensitivity equations. The latter [79, 42] resorts to a formulation inherited from the control theory. It requires only one flow solution and the solution of one set of linear adjoint equations.

Sensitivity and Adjoint Methods: Formulations and a Computational Example

A general introduction to these formulations can be found in Lion [54], Pironneau [79] and Jameson & Reuther [42]. We will briefly describe here only some of the basic elements based on the authors' work [99, 100], while other results on numerical ship hydrodynamics can be found in [100, 80, 61, 65]. In [100], SEM and AM methods are compared with the FD approach for the shape optimization of a tanker aimed at the reduction of total resistance at a design speed. A free-surface linear panel solver [5] is adopted to compute the ship's wave resistance, whereas the friction resistance is obtained via the ITTC formula. The mathematical formulation of the problem is based on the hypothesis that the fluid is inviscid and incompressible, the flow irrotational and the pressure constant on the free surface. The fluid velocity can then be written as the gradient of a velocity potential, Φ, that can be decomposed into the uniform stream, Ux, contribution and a perturbation potential, φ, to yield

$$\Phi = Ux + U\varphi.$$

Under this assumptions, the perturbation potential φ at each point P of the semi-infinite 3D space, bounded by the ship-hull surface and the calm-water free surface, can be found by the integral relation:

$$\varphi(P) = \int_S \sigma(Q)G(P-Q)dS = \int_S \frac{\sigma(Q)}{r_{PQ}}dS, \quad (2)$$

where $\sigma(Q)$ is the source strength at an arbitrary point Q taken on the boundary of the integration domain; S is the sum of the ship-hull surface and free-surface areas. The Green function G specific to the flow problem of interest is the Rankine singularity defined as

$$G(P-Q) = \frac{1}{r_{PQ}},$$

where r_{PQ} is the Euclidean norm (i.e., the distance) between point P and Q. As usual, the boundary conditions are adopted to find the source strengths: a Neumann-type boundary condition at the ship-hull surface:

$$n_x + \varphi_x = 0 \quad (3)$$

and a linearized boundary condition on the calm-water surface:

$$\varphi_{xx} + \frac{\varphi_z}{Fr^2} = 0. \quad (4)$$

Equations (3) and (4) are in non-dimensional form; the reference values selected to make Eqs. (3) and (4) non-dimensional are the ship speed U and the ship length L. The Froude number is defined as $Fr = U/\sqrt{gL}$, where g is the acceleration of gravity. The subscripts xx and z denote partial derivatives evaluated along the x- and z-axis of the Cartesian frame of reference, which moves at constant speed U, along with the ship; the x-axis lies parallel to the direction of the ship travel.

Discretizing the potential perturbation φ (in order to be able to solve the problem numerically) defined by Eq. (2) yields

$$\varphi(P_i) = \sum_{j=1}^{N_{tot}} \int_{S_j} \sigma_j G_{ij} dS_j = \sum_{j=1}^{N_{tot}} \int_{S_j} \sigma_j \frac{1}{r_{ij}} dS_j,$$

where $i = 1, N_{tot}$. The boundary S is approximated[6] by N_{tot} panels, whose finite surface area is S_j; P_i is the point at the panel centroid; N_{tot} comprises N_{body} panels approximating the ship-hull surface plus N_{fs} panels approximating the free surface.

[6] Only the region of the free surface closest to the the ship is discretized

The total resistance coefficient C_t is defined as:

$$C_t(\sigma_j, \xi) = C_w(\sigma_j, \xi) + C_f, \qquad (5)$$

where the wave resistance coefficient C_w is defined as

$$C_w(\sigma_j, \xi) = \frac{1}{S} \sum_{i=1}^{N_{body}} \left[\left(\frac{z_i}{Fr^2} - p_i \right) (n_x)^i \Delta S_i \right],$$

and the friction drag coefficient C_f is estimated with the ITTC formula [41]:

$$C_f = \frac{0.075}{(\log_{10}(Re) - 2)^2}.$$

The variable ξ is the vector of the N design parameters used to parameterize the hull shape.

The SEM involves differentiating the cost function C_t, Eq. (5), with respect to the design parameters ξ_k to obtain N relations

$$\frac{dC_t}{d\xi_k} = \sum_{j=1}^{N_{tot}} \left[\left. \frac{\partial C_t}{\partial \sigma_j} \right|_{\xi} \hat{\sigma}_j^k \right] + \left. \frac{\partial C_t}{\partial \xi_k} \right|_{\sigma} \qquad k = 1, N,$$

where $\hat{\sigma}_j^k$ is the sensitivity of the source strength at the j-th panel with respect to perturbations of the k-th design parameter defined as:

$$\hat{\sigma}_j^k = \frac{\partial \sigma_j}{\partial \xi_k} \quad j = 1, N_{tot} \quad k = 1, N.$$

The AM formulation can be derived as follows. An adjoint cost function, C_t^*, is introduced by adding to the original cost function, C_t, a number of penalty function terms which constrain the optimum state to satisfy the set of PDE which describe the flow evolution:

$$C_t^* = C_t + \sum_{l=1}^{N_{tot}} \lambda^l w_l(\sigma, \xi),$$

where the λ^l terms are the Lagrangian multipliers of the penalty functions and w_l represent the discrete forms of Eqs. (3) and (4). By definition, one finds that

$$\frac{dC_t^*}{d\xi_k} = \frac{dC_t}{d\xi_k} + \sum_{l=1}^{N_{tot}} \lambda^l \frac{dw_l}{d\xi_k}.$$

If the flow equations are satisfied, the following implications hold:

$$w_l(\hat{\sigma}_j, \xi) = 0 \Rightarrow \frac{dw_l}{d\xi_k}(\hat{\sigma}_j, \xi) = 0 \Rightarrow \frac{dC_t^*}{d\xi_k} = \frac{dC_t}{d\xi_k}.$$

Details about the explicit form of terms like $\partial\sigma_j/\partial\xi_k$ and the implications of such an approach are reported in [100] as well as numerical evidence of the improvements obtained by SEM and AM with respect to standard FD.

In Figs. 3–6 a summary of the results is reported (for all the details see [100]). The optimization exercise has been carried out for a tanker hull form of 19,300 tons (Fig. 3), advancing in calm water at a speed of 16 knots (at even keel). The problem is solved at a single design speed, but numerical and experimental tests have been performed for several speeds, ranging from 12 to 18 knots. Due to the inviscid nature of the potential flow solver used in the computation, the optimized region has been restricted to the forward part of the ship (the first 15% of the hull, see Fig. 4). Only the y-coordinate of the hull was allowed to vary, while the centerline profile was kept fixed. Different gradient methods where tested (steepest descent, conjugate gradient, sequential quadratic programming). The main findings from the comparison of FD, AM, and SEM were the following:

- The comparative analysis of the CPU time requirements of the three methods have shown that SEM and AM deliver almost identical computational speeds, which was about 1.6 times faster than the FD method (they converge in the same number of cycles; see Fig. 5). The main reasons allowing SEM and AM to perform better than FD are the reduction of the number of flow solutions needed to compute the cost function gradient and the possibility of using the same LU factored matrix both for the flow solver and the SEM or AM equations;
- The SEM is, in principle, more efficient than the standard FD method, mainly because only one flow solution is required to compute the sensitivity derivatives instead of the $N + 1$ (or $2N + 1$ for second-order accuracy), flow solutions needed by the FD method. However, a high efficiency can be actually achieved only if the cost of computing the coefficients of the sensitivity equations is computationally small with respect to the cost of one flow solution. This balance is directly related to the specific set of PDE's involved in the optimization procedure.
- All three methods tested—FD, SEM and AM—found the same optimal shape, Fig. 4, thus suggesting that the good correlation between the cost function gradients produced by the three methods holds everywhere within the admissible domain of the optimization process, even when the design point approaches the domain boundaries set by the design constraints.

We underline here that both the SAM and FD approaches are methods to compute, with different computational complexity, the gradient of the objective function. The gradient information has to be included in some other method, such as Steepest Descent, Conjugate-Gradient (CG), and SQP (see *e.g.* [30]). In the conjugate-gradient method the search direction p at a step k is given by:

$$p_k = -g_k + \beta_k p_{k-1},$$

where g_k is the gradient of f at step k and β_k is a scalar quantity to be determined on the basis of the value of the gradient of the objective function and of the line search direction at the previous and current step. Several methods have been tested to compute β_k (e.g. Fletcher-Reeves, Polak-Ribire, Hestenses-Stifel [30]). The simple Steepest-Descent approximation is obtained for $\beta = 0$. In [76], the problem is tackled numerically by adopting the three above mentioned gradient-based techniques (*i.e.* Steepest Descent, CG and SQP), wherein the gradient components are estimated by FD, and different sets of geometrical constraints are imposed. A typical selection of geometrical constraints includes local bounds on the width of the transverse section of the hull, on the total displacement, plus several equality constraints (*e.g.* the fixed length, beam and draft of a ship hull). Two optimal hull geometries, obtained with different optimization algorithms (CG and SQP), have also been tested experimentally to assess the success of the optimization procedure. Their performance has been compared against that of the original design through an experimental program. Results of this comparison, reported in [76], confirmed the validity of the approach. A reduction of the total resistance (≈ 3%) at the design speed was observed on both the tested models as a consequence of the reduced height of the free surface wave pattern (see Fig. 6).

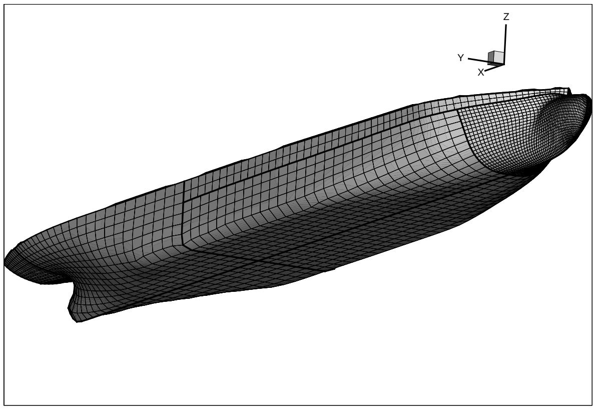


Fig. 3 SEM vs. FD methods. A perspective view of the discretized hull used as test case.

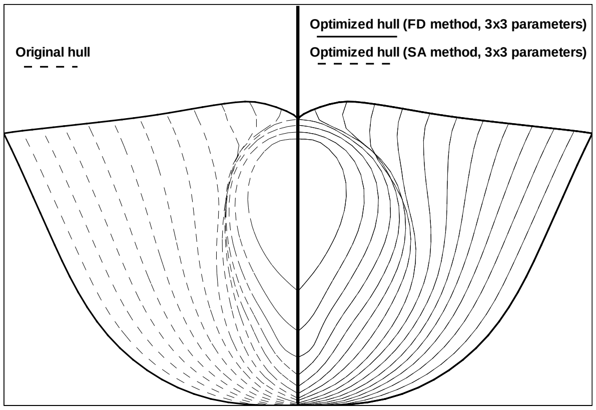

Original hull
Optimized hull (FD method, 3x3 parameters)
Optimized hull (SA method, 3x3 parameters)

Fig. 4 SEM vs. FD methods. Initial and optimal bulb shapes, hull surface parameterized by using 9 design parameters, optimization processes carried out by using the FD and the Analysis Sensitivity (SEM and AM) methods. SEM and AM give the same solution but with different computational time.

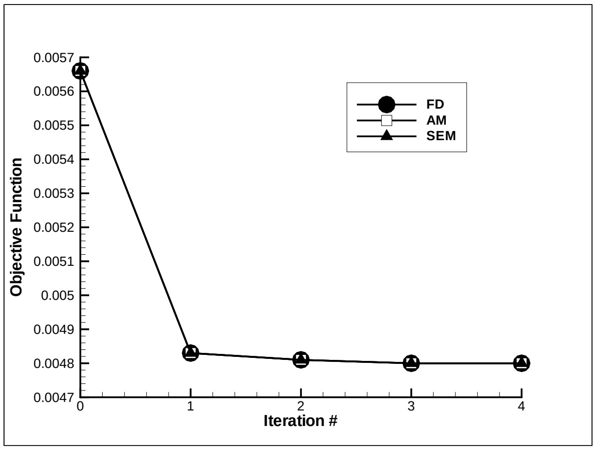


Fig. 5 Convergence histories of the optimization process as obtained by using the FD method, SEM and AM (fine mesh resolution).

Derivative-Free, Global Optimization Methods

The development of an SBD framework which combines costly analysis tools and global optimization algorithms may appear to be a paradox, but good reasons exist to pursue this line of research. Numerical evaluations of the objective function almost always display noise and non-smoothness (and in general, unavailability of derivatives). If the descent direction is not correctly computed, local optimizers might be trapped by spurious, local minima. Quoting Kolda, Lewis & Torczon [49]: “It is widely appreciated in the simulation-based optimization community that the results of complex cal-

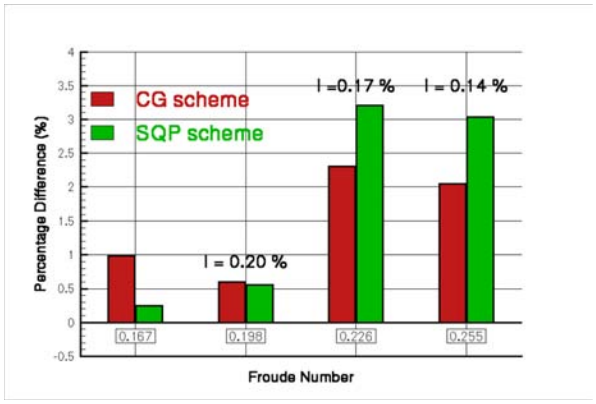


Fig. 6 Experimental results on the two optimized models tested (CG and SQP). The experimental uncertainty, also reported in the picture, is significantly smaller than the measured reduction of total resistance.

culations . . . may fail to have the level of precision necessary for a reliable finite-difference approximation to the gradient, ruling out an off-the-shelf finite-difference quasi-Newton code." There are two more good reasons for going to global optimization methods: (i) the feasible design spaces of ship design optimization problems are very often *non-convex* due to the presence of nonlinear geometrical and functional constraints that have to be enforced to prevent unrealistic results and provide a meaningful design; (ii) when the margin for design improvements is narrowing, the probability that further improvements could come from local optimization methods is small. Of course, one needs to keep in mind that in non-convex problems such as those in which we are interested, there exists no proof of convergence to a global optimum. The goal is to develop methods which have great effectiveness and efficiency in terms of number of objective function evaluations. To this aim, we draw the attention to the *Particle Swarm Optimization* (PSO) [10] method and to the *Real Coded Genetic Algorithm* (RCGA) approach. The PSO method will be described in the next section (together with some applications reported in [13]), while the description of the RCGA approach will be postponed to the multi-objective problems section.

A Particle Swarm Optimization Algorithm: DDFPSO

The Particle Swarm Optimization (PSO) algorithm is a recent addition to the list of global search methods. Since it was originally introduced, [47], the PSO algorithm has been studied by many authors (cf. [46, 71, 87, 88]). In the following, the standard PSO method is described, and a new version is introduced and tested.

The swarm strategy simulates the social behavior of a set of individuals (particles) which share information among themselves while exploring the design variables' space. In the basic PSO method each particle has its own memory to remember the best places that it has visited, whereas the swarm has a global memory to remember the best place ever visited. Moreover, each particle has an adaptable velocity to move itself across the design space. According to these principles, each particle investigates the search space analyzing its own travel experience and that of the other members of the swarm.

The original PSO algorithm is composed of the following four steps:

Step 0. (Initialization): Distribute a set of particles x_0^i inside the design space with random distribution and random initial velocities. Set $k = 1$.

Step 1. (Compute velocity): Calculate a velocity vector v^i for each particle, using the particle's memory and the knowledge gained by the swarm according to:

$$\begin{aligned} v_k^i = \chi \left[w_k v_{k-1}^i + c_1 r_1 \left(p^i - x_{k-1}^i \right) \right. \\ \left. + c_2 r_2 \left(p_{k-1}^b - x_{k-1}^i \right) \right] \end{aligned} \tag{6}$$

where χ is a constriction factor, w is called *inertia weight*, c_1 and c_2 are positive constants, r_1 and r_2 are random numbers equally distributed between 0 and 1, p^i is the best position found by particle, i, and p_{k-1}^b is the best position found by the swarm up to iteration $k - 1$.

Step 2. (Update position): Update the position of each particle, x^i, using the velocity vector and previous position

$$x_k^i = x_{k-1}^i + v_k^i \tag{7}$$

Step 3. (Check convergence) Set $k = k + 1$. Go to Step 1 and repeat until convergence occurs.

Literature reports that fine tuning of the parameters in Eq. (6) is crucial for the optimization process, and that the final solution and the calculation time are strictly linked to the parameters setting. The inertia weight w regulates the trade-off between the global (wide-ranging) and local (nearby) exploration abilities of the swarm. A large inertia weight facilitates global exploration (searching new areas), while a small one facilitates local exploration. Experimental results indicate that it is better to initially set the inertia to a large value, in order to promote global exploration of the design-variables' space and gradually decrease it to get a more refined solution [101]. For these reasons, an initial value for w is set and the decrease rate is calculated by

$$w_k = w_{k-1}\, g_w$$

where w_k is the new value for the inertia weight, w_{k-1} is the previous one and g_w is a constant chosen between 0 and 1. In [101] it is suggested that one use $0.35 < w < 1.4$, and $g_w = 0.975$.

While the inertia w is employed to control the impact of the previous history of velocities on the current one, χ offers to the user the chance to select the search resolution. Quantitatively, if box constraints (e.g. $x_i^l \leq x_i \leq x_i^u$, where x_i^l (x_i^u) is the lower (upper) limit of the design variable x_i) are given, χ takes a value equal to a fraction of the characteristic dimension of the box.

The constants c_1 and c_2 are called the *cognitive* and *social* parameters, respectively. The cognitive parameter indicates how much confidence the particle has in itself, while the social parameter indicates how much confidence it has in the swarm. In the basic PSO algorithm [47], the authors propose $c_1 = c_2 = 2$, so that the mean of stochastic multipliers of Eq. (6) is 1. In [101] different values for the two coefficients are used. In particular, $c_1 = 1.5$ and $c_2 = 2.5$ work well in their examples.

In [10], Campana, *et al.* carried out a more rigorous analysis. In this paper a generalized PSO iteration is described by means of a dynamic linear system whose properties are analyzed. The influence of the particles' starting points and the use of deterministic or stochastic parameters are investigated and some partial convergence results are given. In particular, the PSO parameters are selected by imposing the constraint that the particles' trajectories are confined in a suitable compact set.

Recently, Campana, *et al.* [11] developed an enhanced version of the basic PSO, named *Deterministic Derivative-Free Particle Swarm Optimization* (DDFPSO). The main modifications are:

(i) *Initialization of the swarm:* The use of GO algorithms based on expensive analysis tools imposes a substantial reduction of the particles' number. Instead of using a random distribution, a deterministic one is proposed. In particular, at Step 0 the initial swarm is built with one particle at the center of each face of the n-dimensional hyper-cube which represents the design space. As a consequence, the total number of particles is $2n$. Moreover, the initial velocity vector, defined in Eq. (6), is set equal to 0.

(ii) *Boundary search phase:* According to Eqs. (6) and (7) each particle is attracted by the others. As a consequence, in an example with two design parameters, the particles can not escape from the dashed region indicated in Fig. 7 unless the inertia term is sufficiently strong, i.e., the corners of the design space cannot be reached by any particle of the swarm. In order to force the search along the boundaries of the design space, a threshold value is introduced that limits the d-component of the particle's speed orthogonal to the face where the particle has been initially placed:

$$|v_k^d| \leq \frac{|x_{max}^d - x_{min}^d|}{g^d}, \quad g^d < 1.0$$

This threshold value is initially set as a fraction of the box's dimension in the d-th direction. This limit is progressively relaxed during the iterations. By applying this *normal* speed limiter it is possible to force the exploration of the corners of the design space (as reported in the example shown in Fig. 7). The dashed (inner) region of the feasible space is obtained by connecting the initial positions of the particles. Without the normal speed limiter (Fig. 7a), particles are strongly attracted to each other and tend to be confined inside the dashed region, hence failing to locate the global optimum. The use of the normal speed limiter (Fig. 7b) allows the particles to explore the design space near the boundary and to find the global optimum.

(iii) *Suppression of random coefficients:* In the new version PSO is modified according to a deterministic flavor. In Eq. (6), we decide to fix the parameters r_1 and r_2 equal to 1, thus eliminating the random factor introduced by these two coefficients. In this way we transform a pure stochastic method into a deterministic one. The motivation stems from the use of PSO in combination with CPU time-intensive numerical simulations used to obtain both objective function information and constraints' information. A stochastic approach would require repeated runs which might require simply too much computing time for real-life industrial applications.

(iv) *Particles with violated constraints:* The original PSO algorithm is defined only for unconstrained optimization problems. Because freedom in the design of a ship is always limited by a large number of constraints, we transformed the method by setting the weighting term of particles with violated constraints equal to 0. As a consequence, Eq. (6) is replaced by

$$\begin{aligned} v_k^i = \chi \big[& c_1 \left(p^i - x_{k-1}^i\right) \\ & + c_2 \left(p_{k-1}^b - x_{k-1}^i\right)\big] . \end{aligned}$$

In most cases the new velocity vector will immediately point back to the feasible region in one step. This feature of the DDFPSO algorithm is particularly useful in optimization problems with non-convex feasible design spaces.

(v) *Convergence criterion for the global search phase:* In the original algorithm there was no stopping criterion. For real applications, however, the maximum number of iterations is typically fixed, driven by elapsed-time constraints. An heuristic convergence criterion for the PSO phase is therefore introduced to switch from the global to the local search phase. For the convergence of the global phase we focus on the identification of all the particles which fall into the same basin of attraction. At each iteration, k, all the particles of the swarm are analyzed; if a descent trend Eq. (8) is shown for a given number of consecutive steps, the ith particle is marked as *attracted* by a basin.

$$\begin{cases} \left[f(x_k^i) - f(x_{k-1}^i)\right] \\ \quad \times \left[f(x_{k-1}^i) - f(x_{k-2}^i)\right] \geq 0 \\ f(x_k^i) - f(x_{k-1}^i) < 0 \end{cases} \tag{8}$$

Then the distance among the *attracted* particles is computed. These particles are assumed to be in the same basin if the maximum distance among them is a small fraction of the maximum initial distance. An average radius of the distance among the *attracted* particles is computed and the center of the basin is estimated. This radius is finally used to check the remaining, *non-attracted*, particles of the swarm. Given their distances from the center of the basin and the remaining steps before the maximum iteration is reached, if their current speed is already too slow to bring them back to the basin where the current optimum is located, they are abandoned. When all the swarm particles are in the basin or abandoned, the global search phase is terminated.

(vi) *Local search phase:* It has been observed that `PSO` is generally fast in the identification of the attraction basins but it is quite slow to converge. The strategy adopted here is to use a two-phase, global-local, approach. Thus, we introduce the following additional step to perform a local refinement of the solution:

Step 4. (Local search): Starting from the point with the lowest objective function value, we perform a local minimization with the derivative-free line-search method `DF` proposed in [56]. However, any local search method may be applied in this phase.

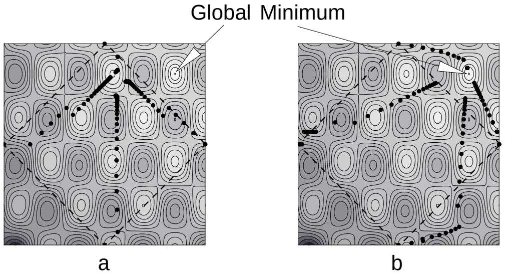


Fig. 7 The path of the swarm particles for the $n = 2$ Griewank test function, with and without the speed limiter introduced in DDFPSO.

An Example: Hull-Form Optimization for Seakeeping with DDFPSO

The problem selected for showing the capabilities of the Swarm global approach is the optimization of the peak of the Response Amplitude Operator (RAO) for the heave motion when a containership (S175) is advancing at a constant speed of 16 knots in head seas. In particular, the minimum of the response is searched for non-dimensional frequencies higher than 0.4. More details can be found in [11].

The seakeeping performances of the ship are numerically evaluated with a potential flow solver based on strip theory. The hull-shape parametrization is performed via a Béziér patch that is superimposed on the original hull shape [76]. The six patch-control points ($N = 6$) are used as design variables to modify the hull shape: $x = (x_1, x_2, \ldots, x_6)^T$.

Some geometrical constraints are imposed during the optimization process. In particular we define a range of variation for the displacement, Δ, and for the beam, B, as:

$$2398 \text{ t} \leq \Delta(x) \leq 2460 \text{ t} \tag{9}$$

$$25 \text{ m} \leq B(x) \leq 26 \text{ m}. \tag{10}$$

Furthermore, in order to avoid unrealizable geometries, box constraints are imposed on the design variables, x_i:

$$-20.0 \leq x_i \leq 20.0 \quad i = 1, \ldots, 6.$$

Finally, a constraint on the metacentric height GM is also imposed. The ensemble of these constraints Eqs. (9) and (10) form a nonlinear constraint and therefore create a nonconvex, feasible, design space. The `DDFPSO` algorithm is able to deal with this problem. In this example `DDFPSO` is compared with existing numerical optimization codes, in particular `DIRECT` [45] and `FILLDIR`, for which the reader is referred to [11].

Table 1 Comparative results for the RAO's heave motion peak optimization ($\mathtt{maxnf} = 100\,N$).

	$\mathtt{maxnf} = 100\,N$		
Algorithm	f_{min}	improv.(%)	nf
FILLDIR	0.8796	33.36	320
DDFPSO	0.8648	34.48	600
DIRECT	0.8636	34.57	601

Table 2 As in Table 1 but with $\mathtt{maxnf} = 1000\,N$.

	$\mathtt{maxnf} = 1000\,N$		
Algorithm	f_{min}	improv.(%)	nf
FILLDIR	0.8572	35.06	982
DDFPSO	0.8645	34.50	714
DIRECT	0.8627	34.64	2345

The maximum number of objective function evaluations (maxnf) is adopted as stopping criterion. Two runs have been performed for each algorithm, one run with maxnf equal to $100\,N$ and the other run with maxnf equal to $1000\,N$. Values of the objective function, improvements with respect to the initial objective function value, and total number of function evaluations required by each method are reported in Tables 1 and 2. All the algorithms show good results, very close to each other.

In Fig. 8 we provide the RAOs associated with the various final hull forms obtained in the $100\,N$ case. All the RAO's are significantly better than the starting one, and the heave motion peak of the RAO has been damped. In Fig. 9 the original and optimized geometries are compared and the values of the design parameters are shown too.

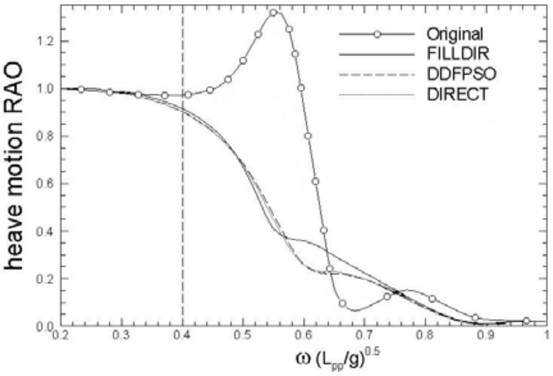


Fig. 8 Comparison of the heave motion RAOs of the original and optimized hulls ($\mathtt{maxnf} = 100\,N$); the objective function minimum has been searched for non-dimensional frequencies higher than 0.4.

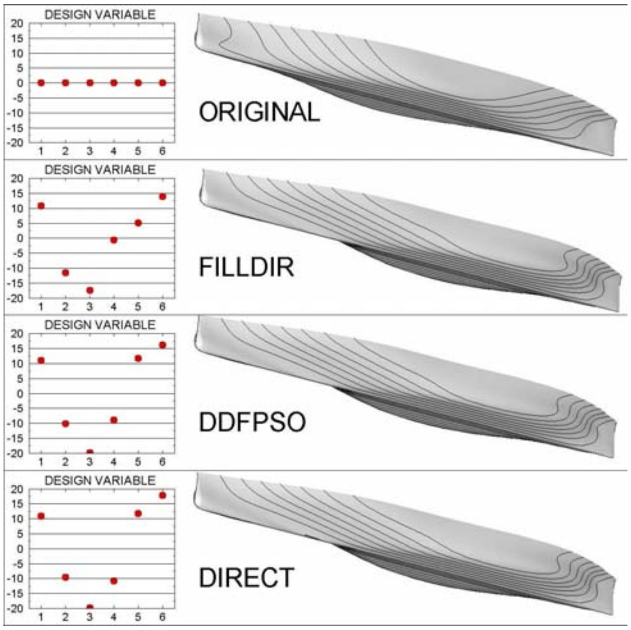


Fig. 9 S175 hull shape and optimum design variables comparison ($\mathtt{maxnf} = 100\,N$).

SELECTION OF DESIGN PARAMETERS

The choice of the shape parametrization technique has a large impact on the practical implementation and often also on the success of the optimization process. There exist a large number of approaches, from simple morphing techniques, to CAD systems, and to Free Form Deformation (FFD) techniques. As a general statement we might say that, in principle, the number (and the type) of parameters implicitly defines the *diversity* of the admissible shapes: since the larger the variety of potential designs, the larger the improvements we can hope to find (starting from the original design), the importance of a proper choice of the design parameters is evident.

First of all, we need to reproduce an initial design, because most of the time one has to (or wants to) start from a known shape: all the details of the original hull form have to be described correctly. This represents a problem with dedicated parametric CAD systems: quoting Samareh [84]: "To parameterize an existing model is still a challenging task for current CAD systems ... and the models created are not always good enough for automatic grid generation tools. Designers may believe their models are complete and accurate, but unseen imperfections (e.g., gaps, unwanted wiggles, free edges, slivers, and transition cracks) often cause problems in gridding for CSM[7] and CFD." Another problem

[7] Computational Structural Mechanics

is that there is no standard interface for data access across CAD systems. Furthermore, the number of resulting parameters may not be at all small, and they still cannot provide analytical sensitivity analysis. For a more detailed discussion about these topic refer to [84] and the references cited therein. The same problem arises if we try to use directly the set of NURBS present inside some of the standard formats for the description of a 3D surface, like the IGES format. Also in this case, the number of control points describing the hull surface may be large, sometime causing the generation of *wavy*, unrealistic, shapes.

Another issue that has to be addressed is that the computational grid adopted in the analysis must be regenerated or deformed each time there is the need to evaluate a new perturbed design, and this operation has to be performed in background, without any guarantee about the quality of the new mesh. When this has to be done in conjunction with RANS solvers, the regridding issues may become extremely relevant to the performance and the final result of the optimization. Several techniques available for the field grid movements are summarized in [84].

Apart from some extreme hull shape deformations, the experience of the authors with the deformation propagation applied directly onto a high-quality (block structured) mesh is that it is possible to preserve the good characteristics for the deformed mesh, provided that attention is paid to moving grid points in the boundary layer region and close to high curvature regions. Deformation (more than regeneration) of the computational grid represents a good approach to the problem: (1) one doesn't need to regenerate the whole volume grid each time the hull shape is perturbed, (2) the initial hull shape is preserved and (3) one can deform some part of the hull with a prescribed degree of continuity. As a counterbalance there is the difficulty in translating the optimal shape into a CAD system: this operation requires the solution of a problem of reverse engineering, in which the grid has to be fitted by a number of NURBS to be imported into the CAD system. The use of multi-block quadrilateral structured grid for the hull surface is of great help in this phase, and negligible discrepancies have been observed at the end of the shape translation.

In Peri, *et al.* [76], Campana, *et al.* [13] and Tahara, *et al.* [95] some alternatives have been explored for hull shape parametrization (details are reported in Appendix B). The first alternative is the superposition of a polynomial surface, controlled by a limited number of parameters, on the original hull shape (or a portion of it). This approach is documented in [76], where a Béziér surface is adopted. Proper fairing conditions are applied in order to guarantee fairness of the resulting surfaces, so to avoid discontinuities in the optimal hull surface. The value of the Béziér surface is computed at each grid point and added along a selected direction. More than a single surface (patch) can be adopted in order to allow for three directional movements (Fig. 10).

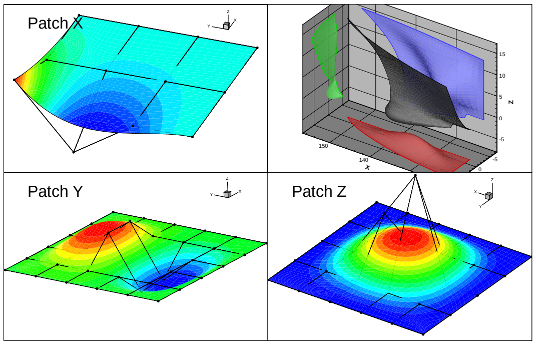


Fig. 10 Parametrization of the bow region of DTMB Model 5415 by means of *Béziér* patches. Each patch controls the movement in one coordinate direction.

A second and more complex approach is based on the free-form deformation of a portion of the *space* containing the hull (or a part of it). The FFD approach [86] has been adopted in [13] and other applications. The idea is to define a parallelepiped including the part of the ship hull we intend to deform (Fig. 11). This parallelepiped is subdivided into a number of regular intervals, and the nodes of this grid are the potential control points. By moving one or more of these control points, we are deforming the parallelepiped, and therefore the shape inside the volume too. The number and direction of the subdivisions guarantee continuity and variety for the deformed shape. Some of the control points can also be linked together, allowing complex relative movements: the flexibility of this approach represents its major advantage. Two more approaches, namely (i) morphing and (ii) CAD based approach, are briefly summarized Appendix B.

METAMODELS

In the past few years, there has been a growing interest in the use of metamodel methods for numerical design optimization. These methods fit simple functions, such as low-order polynomials, to a set of known (and computationally or experimentally expensive) data. All numerical optimization

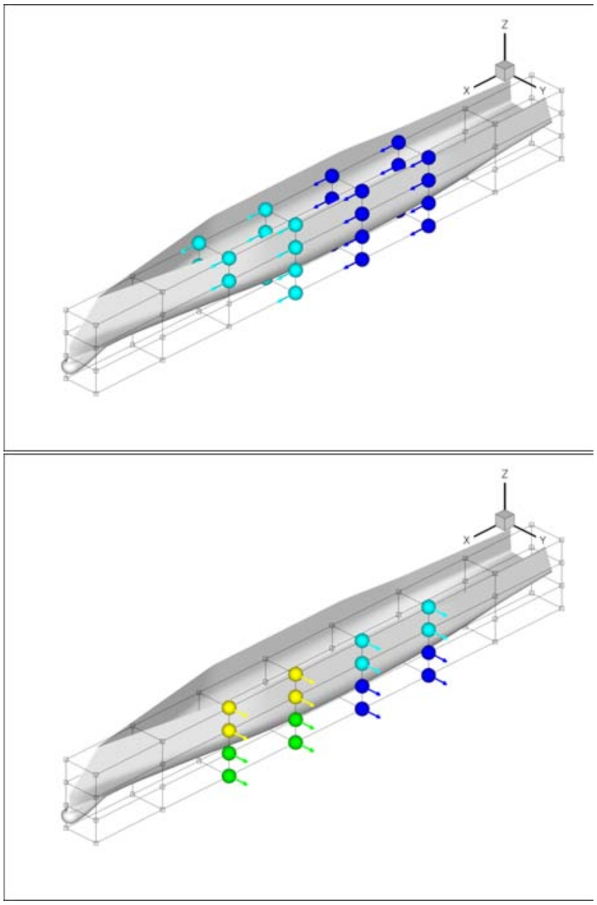


Fig. 11 Parametrization of one hull (the ONR Topside Series tumble-home hull form [8]) by means of the FFD approach. Each vertex of the grid is a potential design variable. To reduce the total number of design variables, different vertices can be linked together along different coordinate directions. This case shows a 6 design variable example. *top* (*bottom*): two (four) groups of control points allow the deformation along the longitudinal (transverse) direction.

algorithms are based on combining information, i.e. supplying the vector of design variables, $\mathbf{X}$, and receiving the vector of responses, $F(\mathbf{X})$, *via* some performance analysis methods (e.g. CFD simulations). When these simulations are computationally expensive—if they are inexpensive, any brute-force search technique will do—their use in exploring large design spaces can be prohibitive. One possible approach to deal with this problem is to construct a metamodel, that is an approximated model (computationally inexpensive) based on a limited number of trial solutions. The term *metamodel* literally means *model of a model*. Metamodels are intended to describe relationships between design performances and adopted variables, with the following advantages: (i) they yield insight into the relationship between responses, F, and design variables, $\mathbf{X}$; (ii) they provide fast analysis tools for design-space exploration since cheap-to-run approximations are used; and (iii) they facilitate the integration of discipline-dependent analysis codes into the overall design strategy. Furthermore, they can be easily parallelized with almost theoretical speed-up.

The three fundamental steps in using metamodelling are: 1) to choose a distribution of trial points in the design space for generating data; 2) to choose a model to represent the data and fit the model to the sampled points, and 3) to validate the model. We will briefly describe here how to select the trial points [Design Of Experiments (DOE)] and how to use different types of metamodels.

There are various types of metamodels. In [44], Jones reviews the most important developments: response surface models [63]; regression polynomials; Kriging [59]; neural networks [34]; etc. (Also see [43] & [89] for a comparative evaluation of various types of metamodels.) In this paper the focus is on Kriging, which will be introduced in the following subsection.

A more recent approach is to build *ensembles of metamodels* with the available data ([53], [31], [72]). The idea is to extract as much information as possible from the data required for developing the metamodel. These ensembles of metamodels can be developed without significant expense compared to the cost of acquiring data, and they have proven effective in improving the predictions. In particular, [72] shows how the use of two metamodels provides information for the selection of trial solutions, producing in turn a more efficient computational base for the metamodels.

Design of Experiments (DOE)

The basic element needed by a metamodel to produce an estimate of the function is a set of known values at given positions. These known values can be split into two different sets: the *training set* and the *validation set*. The training set is used to tune the model, usually minimizing the error of the model on these points, while the validation set is used for validation, that is, for checking the prediction on points *unknown* to the metamodel.

The selection of training points is crucial for the quality of the model. Uniformly Distributed Sequences (UDS, [51]) are the most preferable for the selection of the training set. Two different strategies have been investigated for this purpose: LP$_\tau$ nets [92] and Orthogonal Arrays (OA) [35]. Once the level of description for each design variable is fixed, OA provides an uniform orthogonal distribution of points, minimizing the number of training points while still preserving the density of the sampling. The LP$_\tau$ net belongs to the class of UDS too; this distribution guarantees a high degree of uniformity. Unfortunately, it is available only for a limited number of design variables less than or equal to 20.

Metamodelling Techniques: Kriging

The Kriging method was originally developed for use in the computer science and engineering fields as a Design and Analysis tool for Computer Experiments (DACE) modeling by Sacks, *et al.* [82]. Kriging methods have been used to model the response of many engineering systems. Martin and Simpson conducted a study on using Kriging models to approximate deterministic computer models and discussed the applicability of various Kriging variants [59], whereas an interesting recent application of Kriging in a surrogate management framework is reported in [58].

Kriging is an interpolation method that provides an estimate of the system response by a weighted sum of a limited set of the known values,

$$\hat{f}(x) = \sum_{i=1}^{N} \lambda_i f(x_i),$$

under the condition that

$$\sum_{i=1}^{N} \lambda_i = 1.$$

By the definition of interpolation and by construction, Kriging preserves the response value on the training points. The optimal weights are obtained by applying some spatial correlation considerations among the training points coming from the DOE. Kriging is based on spatial correlation between locations. The underlying idea is that any sample is spatially correlated with the others, so that the weights, λ_i, can be derived once the spatial correlation law is known. In order to do that, we define the *variogram* for the objective function as

$$\gamma(x+h, x) = E\left|f(x+h) - f(x)\right|^2,$$

where $E|f|$ is the expected value of f. Once the variogram is known, we can express the weighting coefficients by writing an equation for each training point plus one for a Lagrangian multiplier in the form,

$$\sum_{i=1}^{N} \lambda_i \gamma(x_i, x_j) + \mu = \gamma(x_j, x_0), \quad j = 1, \ldots, N,$$

where μ is the Lagrange multiplier. The last equation is the cardinality condition:

$$\sum_{i=1}^{N} \lambda_i = 1.$$

In this way, the weights, λ_i, are calculated so that the response in x_0 is unbiased and optimal, that is, with a minimum squared error of estimation. The only missing element is the variogram which can be estimated by fitting the known values on the training points with some simple model (e.g., linear). The error, σ, associated with the estimate is expressed by the equation,

$$\sigma_k^2(x_0) = \sum_{i=1}^{N} \lambda_i \gamma(x_i, x_0) + \mu.$$

A number of different options are available. Kriging can be derived by using a limited number of training points (e.g., the closest points to the computational point), and by using a large number of variograms. As a consequence, a large number of different types of Kriging can be found in the literature. In global optimization problems (cf. [74]), one can use the full training set in order to produce a global approximation. The variogram is assumed to be a linear function of the distance between two points. The coefficients are computed by fitting the experimental variogram, computed as

$$\hat{\gamma}(h) = \frac{\sum_{i,j \in N(h)} |f_i - f_j|^2}{N(h)},$$

where $N(h)$ is the set of points such that $|x_i - x_j|^2 = h$ with a certain tolerance on h.

The advantage of Kriging over neural networks is the time needed for training. Neural networks typically require an optimization of the parameters, which can take a very long time if the number of design variables and the training set are large. In contrast Kriging only requires the LU decomposition of a single matrix (see e.g. [36]), whose dimensions are equal to the number of training points plus one, independent of the space dimension, the other operations are essentially inexpensive. The weak point of Kriging is that noisy data may sometimes lead to confusing predictions, whereas neural networks are able to naturally smooth out the roughness of the objective function. A desingularization of the distance calculation in the form,

$$h_{ij} = |x_i - x_j| + \epsilon,$$

with ϵ a small positive constant, may be helpful in reducing oscillations in the prediction. Also, Kriging is not able to deal with multiple values on a single location due to singularity of the deriving matrix. Preprocessing of the data is needed and an average operator has to be applied.

Ensembles of Metamodels

In order to improve the predictive qualities of the metamodel, the DOE should be designed properly and adapted to the peculiarities of the response. Since the response is unknown *a priori*, in order to find clues about the locations in which it is advisable or necessary to add more training points, one may try to extract some indications by the combined use of more metamodels [31, 72].

The basic idea for the ensembles of Metamodels is simple: even if trained with the same set of data, metamodels produce, in principle, different predictions. One can then find the positions in which the largest discrepancies between the two metamodels occur; these regions are the ones in which it is useful to insert a new training point, in order to improve the predictive performances of the metamodel.

Kriging has been defined previously, and it appears as a relatively complex modelling strategy. On the opposite side, the most intuitive way to produce an approximation of the response is a local linear model: the Multidimensional Linear Interpolation Method (MLIM), described in [72], is a $\Re^N$ extension of a linear, 2D interpolation. It is based on a Delaunay triangularization (cf. [25]) of the training set. Starting from that, we identify the hyper-tetrahedron that includes the computational point. Since the coordinates of the computational point can be obtained by a linear interpolation of the coordinates of the vertices of this hyper-tetrahedron, one can use the same interpolation coefficients as weights for the linear combination of responses available at the vertices, providing the approximated response at the computational point.

Numerical experiments produced in [72] give evidence of the effectiveness of this strategy.

MULTI-OBJECTIVE OPTIMIZATION

As with most technological problems, optimal ship design is a multi-objective problem, since improvement of a specific aspect of the complete design usually causes the worsening for some others. More generally, the evaluation of different objectives makes it possible to consider the trade-off between different qualities of the ship. Under multiple objectives, the concept of a global minimum point is no longer available for the selection of the best configuration: in fact, there usually exist different minimum points, one for each objective. A different definition of *optimal solution* must therefore be adopted. The concept of multi-objective optimality (i.e., the optimum *trade-off*) was proposed by the economist Wilfredo Pareto in the early 1900's. In brief, he defined as a *dominated solution* all the configurations whose objective function values may all be improved (or at least not deteriorated) by another configuration. Consequently, the set of configurations can be divided into two different sets: *dominated* and *non-dominated* solutions. The latter is the optimal set, i.e., the set of solutions that cannot be further improved without losing performances on at least one objective. For a general introduction to multi-objective problems and solution methods see [17, 62].

Aggregated Approaches

A solution of a multi-objective optimization problem can still be found with single objective methods (cf. [70]). In an *aggregated* (also referred as *scalar*) approach, the multi-objective problem is reduced to a single objective problem by using a linear weighted combination of all the objective functions. The weighted function is often referred to as merit function and implicitly defines a preference order among the objective functions: any change in the weight distribution requires the solution of a new optimization problem. Another scalar approach implies the definition of the optimal solution in the sense of the desired values for the objective functions (*goal programming*). The objective function is then represented by the Euclidean norm, computed in the objective function's space, between the current solution y_i^C and the ideal solution y_i^I:

$$\hat{f} = \sum_{i=1}^{M} \left(y_i^C - y_i^I\right)^2 .$$

The problem with the aggregated approach is that it provides a single final solution, without any information about the sensitivity of the system with respect to the different objectives. As a consequence, we have no idea about the performance gains (or losses) we would have on the system if we had put a greater (or lower) emphasis on another objective, and we cannot understand if the weight distribution for the linear combination was the most effective or not. Therefore, we prefer to always deal with *true* multi-objective algorithms that provide the designer with an entire set of optimal trade-off solutions.

High-Performance Multi-Objective Genetic Algorithm MOGA

GA and MOGA

MOGA is an extended version of the Genetic Algorithms (GA) (for a general introduction and recent survey see [17]), whose basic algorithm proceeds as follows: (i) generation of an initial population of individuals in a random manner; (ii) decoding (if necessary) and evaluation of some predefined quality criterion, referred to as the *fitness*; (iii) selection of individuals based on a probability proportional to their relative fitness; and (iv) crossover (and mutation if necessary). Steps (ii) through (iv) are repeated until the maximum generation. For a general single-objective optimization, objective function, F, is directly related to objective-fitness function, f_0, e.g., by using the *sigmoid* function: $f_0 = 1/[1+\exp(-F)]$ for maximization, and $f_0 = 1/[1+\exp(F)]$ for minimization. The functional constraints are accounted for by using a penalty function approach, which artificially lowers the fitness if the constraints are violated and is expressed as (with reference to Eq. (1)):

$$f = f_0 - r\left[\sum_{j=1}^{p}\left|g_j(\vec{\beta})\right| + \sum_{j=1}^{q}\left|\min\left\{0, h_j(\vec{\beta})\right\}\right|\right],$$

where r is a penalty parameter.

The extension of GA for multi-objective optimization is straightforward: the non-dominated set of the entire feasible search space is the globally Pareto-optimal set [21]. We state that a design variable vector $\mathbf{X}$ is *particularly* less (in some defined sense) than $\mathbf{Y}$ (symbolically $\mathbf{X} < P\mathbf{Y}$) when the following holds: $\mathbf{X} < P\mathbf{Y} \Leftrightarrow (\forall i)(x_i \leq y_i) \wedge (\exists i)(x_i < y_i)$. Under this circumstance, we say that design $\mathbf{X}$ dominates design $\mathbf{Y}$. If a design is not dominated by any other, we say that it is non-dominated or non-inferior. The basic definition may be used to find non-inferior points in MOGA in association with the Pareto-ranking technique. The non-dominated individuals of the entire population define front 1 (R_P=1, see Fig. 12); in the subset of remaining individuals, the non-dominated ones define front 2, and so on; the worst individuals define front W, where W is the number of fronts. At each generation, higher fitness f_0 is given to individuals of higher Pareto ranking, i.e.,

$$f_0 = \frac{1}{R_P}. \tag{11}$$

In practice, the population size for CFD-based optimization is limited to a number of available processors for computation. The Pareto front must be searched by a limited number of individuals in the feasible space as illustrated in Fig. 12. Therefore, a scheme that yields diversity of individuals in the feasible space will be preferred over that which yields the convergence of individuals toward a particular region. A final Pareto front can be defined by using the history of all individuals generated in all generations.

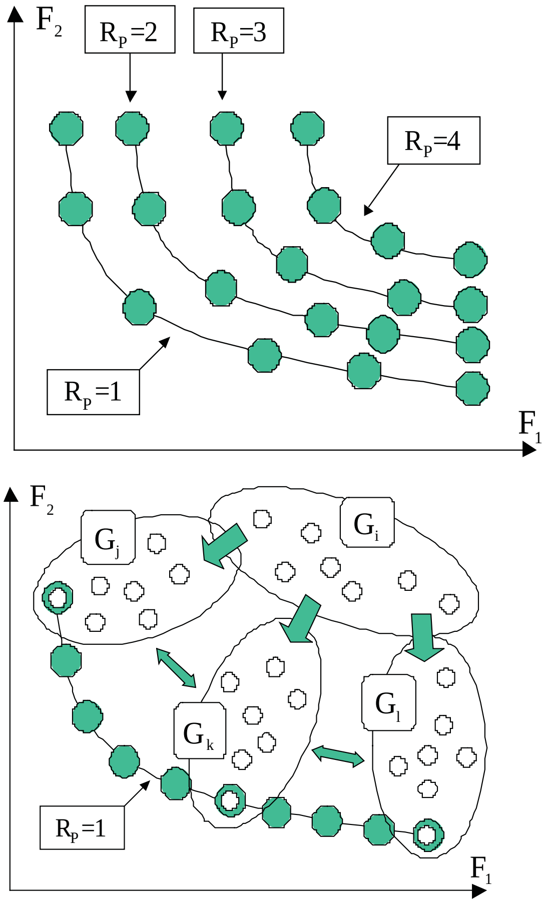


Fig. 12 Pareto ranking and sharing operations. Fitness is given based on Pareto ranking and additive fitness is based on the uniformity of individual distributions on the Pareto optimal set.

BCGA and RCGA

Genetic algorithms are categorized as Binary-Coded GA (BCGA) and Real-Coded GA (RCGA), which are characterized by 3 actions—selection, crossover, and mutation. They are defined as

Selection:

$$x_i(t+1) = \frac{f_i}{\bar{f}(t)} x_i(t), \quad i = 1, \ldots, n$$

where n is the population size, f_i is a fitness of the individual B_i that is mutating from one generation to another, and $\bar{f}(t)$ is the average fitness of a population. In a similar manner, changes in the frequency through crossover and mutation are given by

Crossover:

$$x_i(t+1) = \sum_{i=1}^{n}\sum_{j=1}^{n} C(k|i,j)x_i(t)x_j(t)$$

Mutation:

$$x_i(t+1) = \sum_{j=1}^{n} M_{ij}x_j(t)$$

where C is a crossover tensor, and M_{ij} is a mutation matrix which stands for the probability of mutation from B_j to B_i over one generation. C and M include crossover and mutation ratios, both of which are system parameters.

In application to optimization problems that have a continuous search space, some difficulties with the original BCGA may appear (see e.g. [21]). In this aspect, RCGA implementation is more suitable, since real parameters are used without any string coding and the problems defined in real parameters are directly solved. A concern in RCGA is implementation of crossover and mutation operations, mainly because the string length is no longer finite. Ono and Kobayashi [67] proposed a unimodal normally distributed crossover (UNDX) operator, where three or more parent solutions are used to create two or more offspring. Offspring are created from ellipsoidal probability distribution with one or more axis formed along the line joining two of the parent solutions. The extent of the orthogonal direction is decided by the perpendicular distance of the third parent from the axis (see Fig. 13). When creating a new solution this operator assigns more probability near the center of the space between first two parents than near the parents themselves.

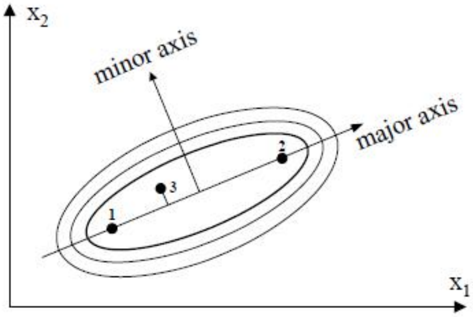


Fig. 13 Illustration of Real-coded genetic algorithm (RCGA) with unimodal normal distribution crossover (UNDX).

For the present RCGA [96, 95], the mutation operation is not explicitly implemented but is simulated by using the "gene pool" approach. That is, a gene pool whose size is larger than n is initially generated in a random manner, and genes in the pool are used to replace those for the individuals with low fitness. Through this approach, new genetic information is always involved in creating new generations, which practically imitates the BCGA mutation. Hence, the mathematical formulation of the present RCGA is written as,

Selection:

$$x_i(t+1) = \frac{f_i}{\bar{f}(t)}x_i(t), \quad i = 1, \ldots, n$$

Crossover:

$$x_i(t+1) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n_0} C(k|i,j,k)x_i(t)x_j(t)x_k(t)$$

where, $n < n_0$, and n_0 is the size of gene pool and set to be three times larger than n in the present work. The crossover ratio in the present RCGA is inherently 1.0. As described in the earlier section, BCGA and RCGA, described above, are used in the form of a multi-objective optimization scheme, i.e., BC-MOGA and RC-MOGA, respectively. An important drawback of GA is the large computational load, larger than that of any gradient-based optimization algorithm. To deal with this problem, a parallel computing technique has been adopted. The details are described in the following section.

Parallel Coding Method

When GA was originally proposed, it was already recognized that there is a parallel nature of the algorithm along with the inherent efficiency of parallel processing. Nevertheless, relatively little work has been done in mapping GA to existing and advanced parallel computing environments. A method for parallel coding adopted here is the master-slave model. The layout of this model is straightforward, i.e., a single master process performs GA operators, while the slaves simply perform function evaluation. In the present study, a process is assigned to a processor so as to maximize CPU performance. Fig. 14 illustrates the present approach. Processor 0 is assigned to master the overall process; processors assigned to groups G-0 through G-m (where $m + 1$ is number of populations), simultaneously execute the CFD method in parallel computational mode. In this scheme, total number of processors is $n(m + 1) + 1$, where n is the number of processors used for each CFD execution. This approach fully utilizes the advantage of parallel coding for CFD as well as optimization algorithms. The present parallel coding is based on Message Passing Interface (MPI) architecture, which is considered a suitable protocol for the present, distributed-memory-model parallel environment. The master and slave processes execute the same code, and each role is defined in a different subroutine. Fig. 14 also shows an example for the main routine. If the process ID (`myid`) is zero, subroutine `master()` is executed, and for other cases, subroutine `slaves()` is executed. In the master routine, `MPI_SEND()` and `MPI_RECEIVE()`, routines are used to communicate with the slave nodes by sending signals to the slaves to execute CFD computations, and to receive a signal back from the slaves when the computation is done. Each calculation can be run in a parallel computational mode by using an assigned MPI group communicator and n processors. When the slave routines are called, the slave nodes assigned to each group execute the calculation and send a signal to the master node when the calculation is complete. The present coding method results in a considerably simplified message transmission as well as in a clear description of separate roles for master and slave nodes.

Algebraic test of `BC-MOGA` *and* `RC-MOGA`

In the following discussion, the performance of the proposed `BC-MOGA` and `RC-MOGA` are evaluated by solving some algebraic test problems originally proposed in [20]. Three test functions are used: Eqs. (12)–(14), showing convex, concave and noncontiguous convex Pareto fronts, respectively:

$$\begin{cases} F_1 = x_1, \quad G = 1 + \dfrac{9}{N+1}\displaystyle\sum_{i=2}^{N} x_i, \quad N = 30, \\ F_2 = G\left(1 - \sqrt{\dfrac{F_1}{G}}\right); \end{cases} \tag{12}$$

$$\begin{cases} F_1 = x_1, \quad G = 1 + \dfrac{9}{N+1}\displaystyle\sum_{i=2}^{N} x_i, \quad N = 30, \\ F_2 = G\left[1 - \left(\dfrac{F_1}{G}\right)^2\right]; \end{cases} \tag{13}$$

$$\begin{cases} F_1 = x_1, \quad G = 1 + \dfrac{9}{N+1}\displaystyle\sum_{i=2}^{N} x_i, \quad N = 30, \\ F_2 = G\left(1 - \sqrt{\dfrac{F_1}{G}} - \dfrac{F_1}{G}\sin(10\pi F_1)\right). \end{cases} \tag{14}$$

Because we are focused on multi-objective optimization problems with expensive objective functions, the tests have a maximum number of function evaluations fixed at $100\,N$. Bounds on the range of the design variables are also applied, i.e., $0 \leq x_i \leq 1$. The number of populations is 50, i.e., in this case 50 processors are used for function evaluation. Hence, the total number of processors used is $n(m+1)+1 = 51$ (i.e., $n = 1$, $m+1 = 50$). The maximum number of generations is 60, and the system parameters are as follows: for `BC-MOGA`, the crossover rate = 0.75, the mutation rate = 0.3, and the single-point crossover mode is used. For `RC-MOGA`, the crossover rate is inherently 1.0.

Figures 15 through 17 show a comparison of solutions for the three test cases. In the figures, left and right are solutions from `BC-MOGA` and `RC-MOGA`, respectively. It is seen that the two `MOGA` results indicate different convergence characteristics toward the Pareto front. In other words, `RC-MOGA` results show slower convergence but more diversity, which results in more widely distributed individuals on the Pareto front in the feasible space. Obviously, the performance indicated in `RC-MOGA` is more promising regarding the aforementioned goal of the present

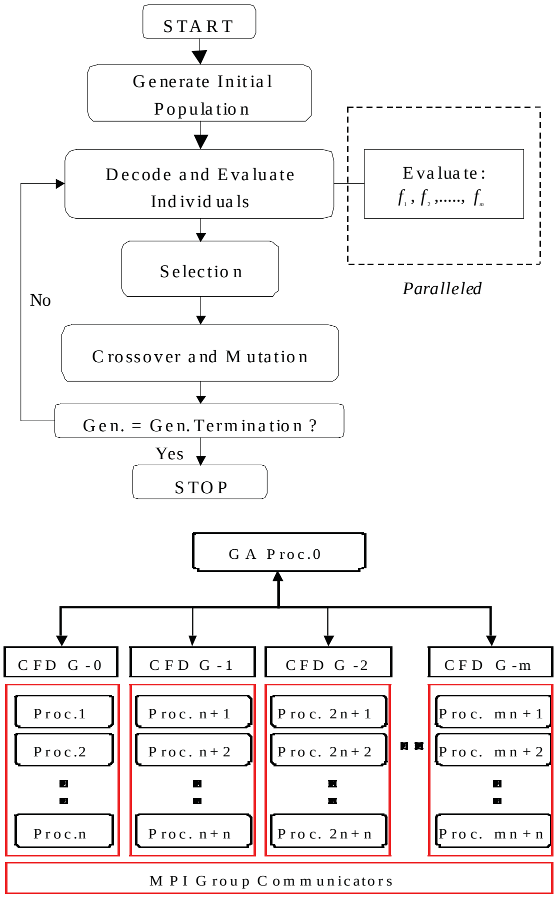


Fig. 14 High-performance parallel-computing architecture and coding for the multi-process algorithm.

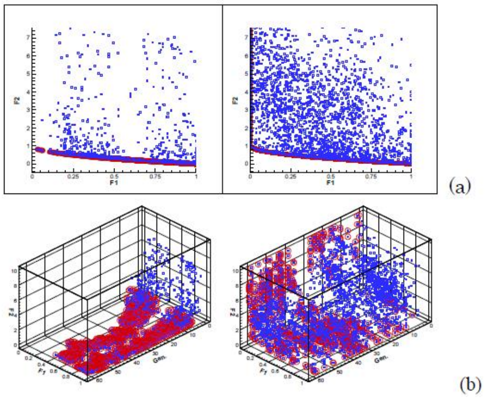


Fig. 15 Comparison of solutions for a multi-objective optimization test case. For T_1 (Eq. 12). Left and right are solutions from `BC-MOGA` and `RC-MOGA`, respectively. (a) F_1 vs. F_2; and (b) 3-Dimensional distribution (F_1, F_2, Generation).

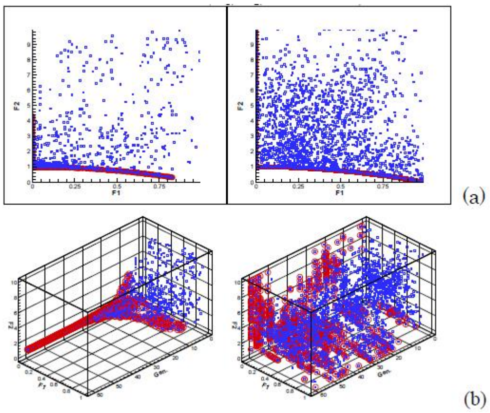


Fig. 16 Comparison of solutions for a multi-objective optimization test case. For T_2 (Eq. 13). Left and right are solutions from `BC-MOGA` and `RC-MOGA`, respectively. (a) F_1 vs. F_2; and (b) 3-Dimensional distribution (F_1, F_2, Generation).

multi-objective optimization.

A Multi-Objective Particle-Swarm Optimization Method

The PSO algorithm presented in a previous section has been extended to deal with multi-objective problems (`MODPSO`). The swarm particles, which move themselves in the design space, are driven by a combination between the personal best position for each particle and the overall best position among all the particles, with a velocity given by Eq. (6). In order to apply this algorithm to multi-objective problems, the concept of "best position" is replaced by the concept of *closest Pareto point*.

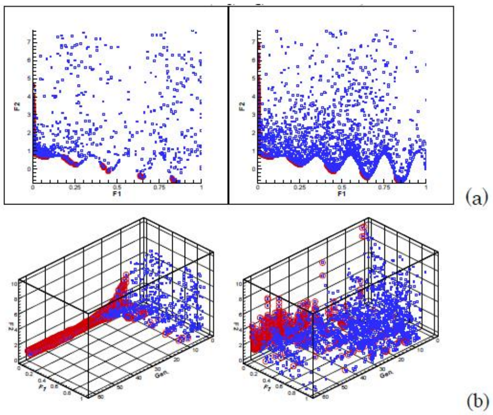


Fig. 17 Comparison of solutions for a multi-objective optimization test case. For T_3 (Eq. 14). Left and right are solutions from `BC-MOGA` and `RC-MOGA`, respectively. (a) F_1 vs. F_2; and (b) 3-Dimensional distribution (F_1, F_2, Generation).

Each Pareto optimal solution is defined as a possible new p_b, *i.e.*, a guide, and the swarm is subdivided into l smaller swarms, all capable of independent evolution, each swarm following its own guide. By defining different guides for the sub-swarms it is possible to build a wider Pareto front. The strategy adopted to assign the particles to the guides (i.e. to form the sub-swarms) is based on the distance in the design space between the particles and the Pareto solutions (details are given in [78]).

Step I. (*Distance evaluation*) the ith particle evaluates its distance, in the design variables space, from the Pareto optimal points;

Step II. (*Guide selection*) the ith particle selects its closest Pareto optimal point as a guide, $p_{i,b}$. Set $i = i + 1$ and go to Step I until $i = N_{sw}$.

As a consequence, the global best position is replaced with the closest Pareto point coming from the Pareto front obtained by considering all the evaluations by all the particles (Fig. 18), and the personal best position is now the closest Pareto point among those of the Pareto front. The equation for the computation of the velocity is the same, but the meaning of the two attractors has changed.

Based on [78, 95, 96], results obtained with these global optimization strategies (`MOGA` and `MODPSO`) for the solution of complex multi-objective design problems will be reported in the following section.

VARIABLE FIDELITY MODELING

Design engineers typically have a suite of different tools to evaluate the performances of a ship, ranging from some simple—*i.e.*, low-fidelity (`LF`) models

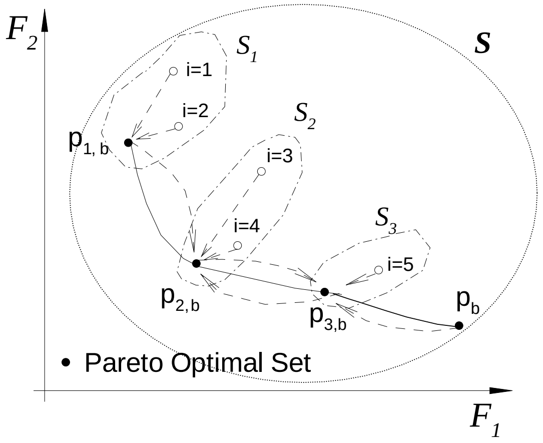


Fig. 18 Shortest Distance Criterion procedure for constructing sub-swarms in multi-objective problems: **S** is the original swarm; S_l are the sub-swarms; $p_{i,b}$ is the best particle of the sub-swarm S_l.

based on some simplified theory, often identical to what was state-of-art in engineering practice decades ago—to more complex simulation codes. Complex—*i.e.*, high-fidelity, (HF))—physics-based models provide obviously high quality in the predictions, at the expense of large computational costs.

Variable fidelity modeling techniques (VFM) reduce the number of expensive HF analyses by taking advantage of cheap LF models. The idea of using computational models of varying fidelity has a long history in engineering design: perform most of the computations with the LF model and correct these predictions by using indicators coming from a HF model.

VFM procedures may be obtained by changing the physics which are modelled, or by using different grid densities or computational accuracies [16, 3]. Occasional (heuristic) recourse to HF models does not ensure the convergence to HF solutions. The ability of the LF model to guide the optimization process has to be monitored and its quality improved when required, while consistency constraints have to be enforced to ensure global convergence to the original HF solution.

VFM for Gradient-Based Algorithms: an Example

Although a LF model may not capture a particular feature of the physical phenomenon to the same degree of accuracy as its higher fidelity counterpart, a LF model may still have satisfactory global predictive properties for the purposes of finding a good direction for improvement of the design. However, the risk of a purely heuristic approach is that sometimes the LF model gives a poor or even an incorrect prediction of the HF model's actual behavior. An example is given with the following preliminary design study of the ballast bulb of a sailing yacht, which is reported in detail in [74]. The bulb is parameterized with two variables only. The design space has been explored analyzing a number of trial designs with both a potential flow solver (with a simplified model for the vorticity shed from the keel) and a RANS code. Computational conditions are heel = 10°, yaw = 4°, $Fr = 0.38$ (10 knots) and $Re = 7.59 \times 10^7$.

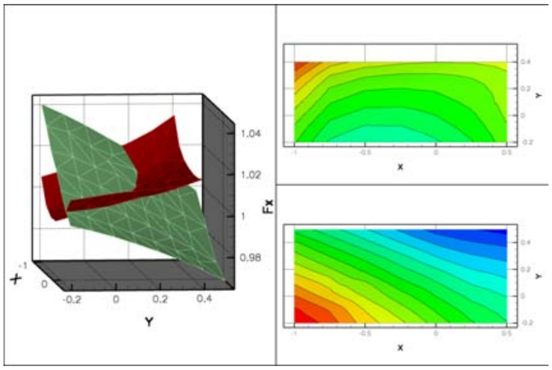


Fig. 19 An example of Variable Fidelity. Left: a perspective view of the potential and RANS solutions as a function of the two design variables. Right: a 2D view of the contour lines of the same quantities (top: RANS solution, bottom: potential results).

In Figs. 19 and 20 the objective function (the total resistance of the complete sailing yacht), evaluated by using the two codes, is reported as a function of the two design variables. The potential flow model predicts a minimum for total resistance at (0.5,0.5), in contrast with the RANS prediction (minimum at (-0.4,-0.2)). Evidently, in this case, the potential flow code for the analysis of the function to be minimized would have driven the optimization algorithm toward a quite *wrong* design. Any heuristic VFM approach, with occasional recourse to the HF prediction, would have encountered this problem.

Nevertheless, the assumption that the potential solver is not useful in the solution of this problem would be wrong. If a scaling function capable of correcting the predictions of the potential solver had been available, then the potential solver might have still been useful. The VFM proposed in [3] is designed with this aim. Defining $\phi_H(x)$ and $\phi_L(x)$ as the HF and LF models, respectively, a possible correction function is given by:

$$\beta(x) = \frac{\phi_H(x)}{\phi_L(x)}$$

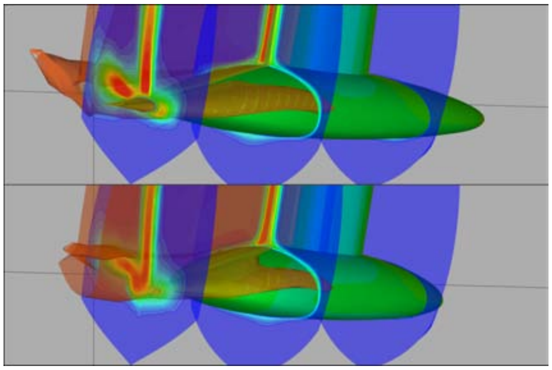


Fig. 20 Two sample solutions provided by the RANS solver around two different configurations. Axial velocity is plotted in three transversal sections. An axial vorticity iso-surface is also highlighted.

so that:

$$\beta(x) \cdot \phi_L(x) \Rightarrow \phi_H(x).$$

By expanding $\beta(x)$ in a Taylor series up to the first order around the current design x_0, one can obtain the local approximation of the correction function:

$$\beta_{loc}(x) = \beta(x_0) + \nabla\beta(x_0)^T(x - x_0),$$

and hence, the approximate value of the true, computational expensive `HF` is (in the region surrounding the current design x_0):

$$\tilde{\phi}_H(x) = \beta_{loc}(x)\,\phi_L(x).$$

This scaling function can be defined as *first order multiplicative*. Recently this approach has also been extended to second order. (There also exist first order *additive* corrections.) It can easily be verified that the approximate model $\tilde{\phi}_H(x)$ satisfies a first order consistency condition [3]. The basic idea is to solve the problem by adopting the approximated model $\tilde{\phi}_H(x)$, while the `HF` model is used just as a check. This is possible due to the fact that β_{loc} is capable of realigning the local gradient of `LF` to the direction of the `HF` gradient. In Figs. 21 and 22, which refer to the total resistance of the sailing yacht discussed above, the effect of the correction on the potential flow-response surface is clearly evident.

However, with $\tilde{\phi}_H(x)$ based on quantities local to x_0, one has to keep in mind that the approximation cannot be applied in the whole design space since we are progressively moving away from x_0. Hence, the validity of the approximation has to be monitored during the process. *Trust-region* methods (see [18] for a complete reference) offer the possibility of defining a systematic way to check and adjust the region (of radius r) in which we can still trust the correction factor developed at design point x_0.

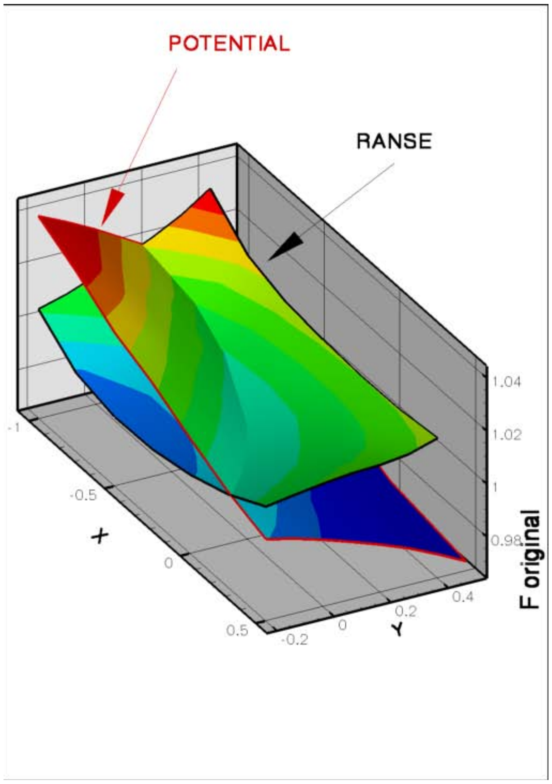


Fig. 21 Total resistance of the sailing yacht as a function of the shape of the ballast bulb: a perspective view of the potential and RANSE response surfaces

The novelty of the method lies in the combination of the local model with a *trust-region* approach, monitoring the radius of the of the *trust region* at each step by analyzing the quantity,

$$R = \frac{\phi_H(x_k) - \phi_H(x_{k+1})}{\phi_H(x_k) - \tilde{\phi}_H(x_{k+1})},$$

and recomputing the scaling factor $\beta(x_0)$ and its gradient $\nabla\beta(x_0)$ only when needed (*i.e.*, when R is far from 1), avoiding any heuristic approach. A complex ship design optimization problem solved with this method is described in [73], where an overall CPU time saving of about 70% was obtained.

VFM for Global (Derivative-Free) Optimization Algorithms

The transposition of the previous approach to global (derivative-free) algorithms is not straightforward, because the idea of using a local correction model $\beta_{loc}(x)$ cannot be extended to a method which is *intentionally not-local*! To overcome this difficulty, Gano, *et al.* [27] and Peri & Campana [74] recently

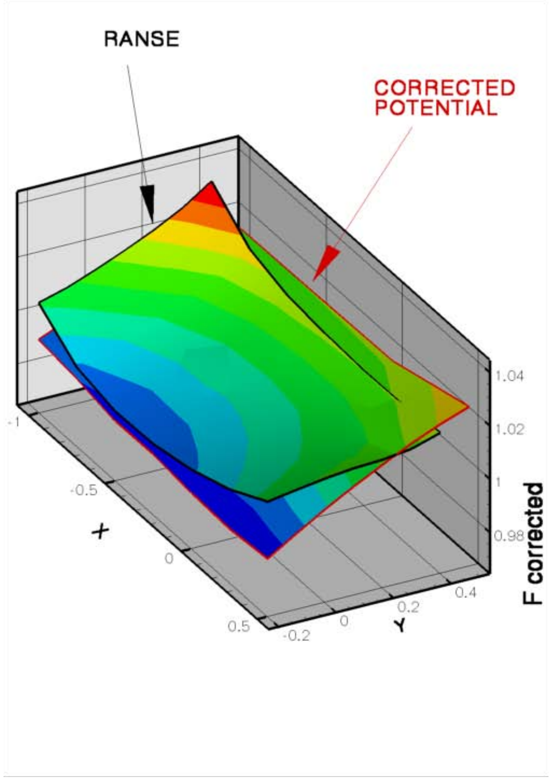


Fig. 22 The *corrected* potential and RANS response surfaces show substantial agreement.

introduced a Kriging-based scaling function to better approximate the high fidelity response on a more *global* level. The validity of the approximation is extended to the whole design space, allowing the global optimization problem to be solved using derivative-free methods. However, for the global version of VFM, first-order consistency is lost, and the issue of the equivalence of the original and the proposed formulation is still an open question.

In order to reinforce the credibility of the solution—avoiding a false value of the corrected function to drive the optimization process—Peri & Campana [74] check the solution further with the HF model as soon as a potential minimum is detected, *if* the distance between the actual minimum and a previously computed point is larger than a prescribed value. This strategy is useful in fixing a criteria for the control of the algorithm by the high-fidelity solver. Also in this case, the trust-region radius is monitored in order to reduce unnecessary computations of the objective function by means of the HF solver, with a further reduction of the computational time.

EXAMPLES AND APPLICATIONS

In this section different applications will be presented, in order to demonstrate the applicability of numerical optimization techniques to the design of a ship hull. Realistic geometrical and functional constraints have been adopted in the problem's formulation. It may be worthwhile to emphasize here that for the complete definition of the shape design problem to be solved, some fundamental items must be addressed: (i) the selection of an initial design (or set of designs) and of the spacial extent of the region(s) to be modified; (ii) the choice of the objective function to be minimized, (iii) the number of and the spatial distribution of the design variables, and (iv) the definition of the problem's constraints.

Single-Objective Application: DTMB Model 5415 Optimization

The single-objective optimization of DTMB Model 5415 has been performed by the authors by using two different SBD frameworks (SBD-A and SBD-B). The SBD frameworks were developed using different parametrization approaches (CAD-based and CAD-free), optimization algorithms (derivative-based and derivative-free) and CFD solvers. Two different RANS codes were adopted as the CFD solvers: MGShip [23] and Version 3.02 of CFDShip-Iowa [105]. Both codes use a surface fitting approach to compute the wave pattern. As a consequence, two different final geometries are obtained. However, it is important to emphasize that the two final geometries present common geometrical modifications. Details on the problem, solutions and model experiments on the final optimized shapes are reported in [13].

The initial design is DTMB Model 5415, which was conceived as a preliminary design for a US Navy surface combatant[8]. There is a large experimental database for Model 5415, due to an international collaborative study on experimental/numerical uncertainty assessments between IIHR, INSEAN and the Carderock Division, Naval Surface Warfare Center (see [94]).

Problem Formulation

The objective function to be minimized is the total resistance, R_T, of the model advancing in calm water at $Fr = 0.28$. This condition corresponds to $Re = 1.67 \times 10^7$ when using a reference length of 5.72 m, which is the length of the ship's model adopted in the ensuing experimental validation. The

[8] Complete details of this transom-stern hull form can be found on the world wide web at http://www.dt.navy.mil/hyd/sur-shi-mod/

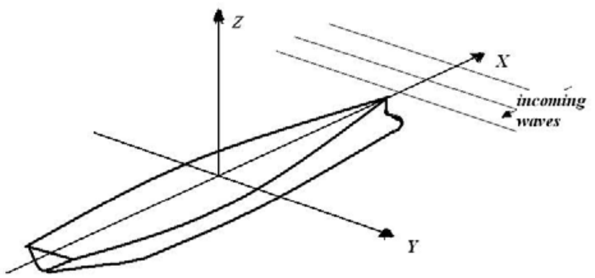


Fig. 23 Sketch of the problem. The resistance optimization is carried out in calm water, whereas for the seakeeping constraint the ship is studied as advancing with constant forward speed in head seas.

modifiable region is only the foremost part of the ship, *i.e.*, the bow and the sonar dome, about 20% of the overall ship length. This is a typical redesign problem of some part of an existing complex system, a necessity which may arise in real industrial applications when the performance of the new design is below the expectations. The test is a difficult one, because the optimizer has reduced freedom and hence expected improvements are small. The problem is solved for the bare hull.

For SBD-A, the displacements of the NURBS control points in the CAD-Based approach are the design variables of the optimization problem. For SBD-B the variables are the control points of *Béziér* patches used in the CAD-free approach (see Appendix B). In the latter case, 11 variables have been used for the parametrization of the deformation: with reference to Fig. 23, two variables serve for y-modifications of the region above the dome, four for the y-modification of the dome, three for the x-modification of the dome and two for the z-modification of the keel line below the dome.

To introduce the elements of a real, complex design problem, functional and geometrical constraints have been enforced. The functional constraints are relative to seakeeping and propulsion (see Table 3). For seakeeping, the monitored quantities are the peaks of the heave and pitch RAO for head seas (defined as the square of the amplitude of the regular wave transfer function at each frequency). The seakeeping qualities of a design were evaluated using the SMP seakeeping code [64]. Another functional constraint was imposed on the problem, relative to the vorticity shed in the fluid by the dome. This quantity is somewhat connected with the propulsion efficiency, cavitation inception and hydrodynamic signature. The sonar dome vortices may indeed travel along the side and keel of the ship and finally interact with the incoming flow seen by the propellers. The stronger the vortices are, the less uniform the propeller inflow is. A control region was placed immediately aft of the sonar dome where the mean value of the axial vorticity should not be greater than the average relative to the original hull.

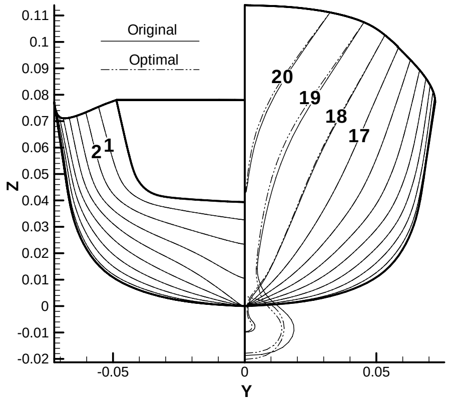


Fig. 24 Body plans of one optimized hull and of the original one (results computed with the CFDShip-Iowa RANS solver). Modifications were allowed only from the foremost section (stem) to 20% of the LBP aft of the FP.

Geometrical constraints are also imposed on the design variables, on the sonar dome volume, on the bow entry angle, on the displacement and on the principal dimensions of the ship. A complete definition of the problem, objective functions and constraints, is given in Table 3.

Although the computational meshes used during the process were relatively coarse (on the order of 250K grid points), they proved to be successful in guiding the optimization algorithms. The final shape was re-computed with a much finer grid (on the order of 1.75M grid points) to check the estimated improvements. The computational domain extends 1.0 LPP forward, 1.0 LPP aft and 1.25 LPP side-to-side. The grids are block-structured with hexahedrical elements, and the transom region has been properly modeled with a dedicated block.

A Variable-Fidelity (Variable-Grid) approach has been applied for SBD-B, and only 36 calls to the high-fidelity solver (i.e. the finest grid) were required, while 314 calls to the low fidelity solver (the coarsest grid) were used.

The Optimized Designs: Numerical Results and Experimental Data

The optimization processes ended with final geometries that clearly display some common geometrical trends. A careful analysis of the shape (Figs. 24 and 25) shows:

- a reduction of the maximum width of the dome

Table 3 Definition of the nonlinear constrained optimization problem. λ^*, is the wavelength of the incoming waves nondimensionalized by the model length L_{pp}. "o", optimized; "p", parent.

Constraint Type	Definition	Note
Functional		
On seakeeping	$S_C = 0.5\frac{\xi_3^o}{\xi_3^p} + 0.5\frac{\xi_5^o}{\xi_5^p}$ $H_C = 0.5\frac{\xi_3^o}{\xi_3^p} \leq 1.02$ $P_C = 0.5\frac{\xi_5^o}{\xi_5^p} \leq 1.02$	ξ_3 peak heave RAO ξ_5 peak pitch RAO o, optimized; p, parent All quantities computed for $\lambda^* \geq 0.4$.
On sonar dome vortices	$\frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\omega_x^o)_i^2}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\omega_x^p)_i^2}} \leq 1 \ \forall i \in R_C$	R_C: circular region placed at $x = -0.30$, centered at $y = 0.02$, $z = -0.07$, with radius $r = 0.018$.
Geometrical		
Bow entry angle	Maximum amplitude variation of 5°	2.5° per side
Sonar dome dimension	A sonar array of radius R_s and height H_s should fit inside the dome	$H_s = 3$ m, $R_s = 2.5$ m, ship scale
Sonar dome position	Maximum forward position fixed	
Main dimensions	L_{PP} and depth fixed	
Displacement	Maximum variation ±2%	

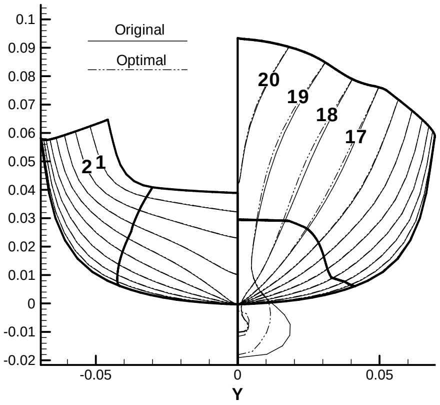


Fig. 25 As in the previous figure, results computed with the MGShip RANS solver.

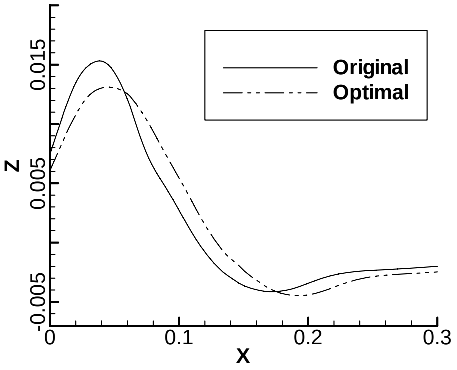


Fig. 26 Wave profiles along the hull in the bow region, as computed with CFDSHIP (left) for the original shape and for one of the optimized shapes.

(and of the dome volume)—decreased by at least 20%;

- a trend to increase of the length of the dome in the forward (x) direction. This is probably limited by the grid topology that was adopted, which did not allow large variations on the bulb length.
- a reduction of the entry angle and an increase of the flare in the region immediately above the sonar dome, differences extend from the stem to 20% of the LBP aft of the FP.

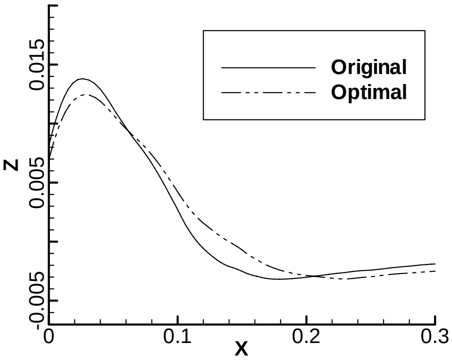


Fig. 27 As in the previous figure, results computed with the MGShip RANS solver.

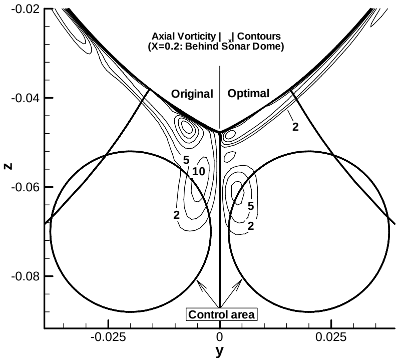


Fig. 28 Comparison of axial vorticity contours between the original and optimal hull forms in a transverse section behind the sonar dome. The control region is reported as a black circle.

The numerical results for the objective function show that the SBD's were able to identify improved designs with lower total resistance with respect to the original Model 5415. R_T reductions for the two optimized shapes are found by both SBDs (-5.32% and -3.01%, respectively).

As shown in Figs. 26 and 27, the computed wave patterns also reflect the improved resistance. The optimized models display remarkably reduced bow wave amplitudes. Furthermore, the steepness of the first wave crest and the first trough are also appreciably reduced. Improvements are also found in the pressure distribution (refer to [13]). As a consequence of the change in the bow volume, the optimized model shows slightly improved seakeeping, more for the heave RAO than for the pitch RAO. The axial vorticity contours reported in Fig. 28 clearly show that the constraint has been satisfied. The optimal model reduces the core of the main vortex, which appears to be confined near the hull surface.

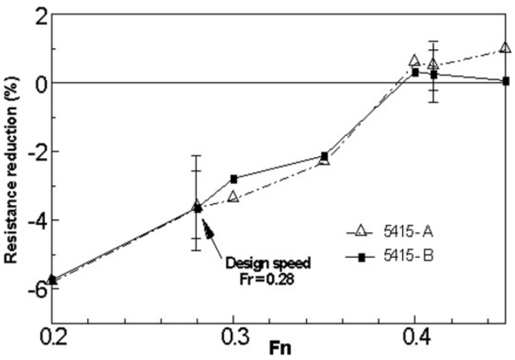


Fig. 29 Experimental validation of the two optimized shapes of DTMB Model 5415 (5415-A and -B). Resistance reduction (%) as a function of the Froude number, error bars are plotted for $Fr = 0.28$ and $Fr = 0.41$

To assess the success of the optimization process a dedicated experimental campaign has been carried out on both the optimized models (SBD-A and SBD-B). The success of the optimization processes is confirmed by the experimental measurements. Reductions of the total resistance with respect to the original hull are reported in Fig. 29 as a function of the Froude number (values below 0% represent improved performance). At the design speed ($Fr = 0.28$) the measured reduction of the total resistance is about 3.80% for both the optimized models, while the experimental uncertainties are clearly smaller than this value. It may be of interest to look at off-design conditions too: in the entire tested speed range, a maximum reduction of about 6% is obtained at $Fr = 0.20$, while at the highest speed ($Fr = 0.41$) a very small increase is measured which, however, is largely inside the error bar of the experimental uncertainty.

Multi-Objective Application: High-Speed Catamaran

The problem reported in this section concerns the multi-objective design optimization of a high-speed catamaran. Details of the problem and of the solution may be found in [12, 95]. The definition of the optimization problem is summarized in Table 4. The ship to be hydrodynamically optimized is a Bath Iron Works (BIW) design for a fast displacement catamaran traveling in a range of speeds between $Fr = 0.4$ and $Fr = 0.7$. Minimum and maximum speeds are connected with the higher and lower displacement (at the beginning and at the end of the mission). A constraint is imposed on the maximum draught.

The ship optimization is performed in two

Table 4 Problem description of the multiobjective optimization problem (p = parent hull form).

OBJECTIVE FUNCTIONS	GEOMETRICAL CONSTRAINTS	FUNCTIONAL CONSTRAINTS
B (*bridge*) = (128, 0, 15) m D (*flight deck*) = (21, 0, 5) m *Sea state*: 5, HEAD SEAS $\ddot{z}_B$ = *Vertical acceleration at the bridge (RMS value)* $\dot{z}_D$ = *Vertical velocity at the flight deck (RMS value)* VCB = 4.193 m Target speed: $Fr = 0.541$ **minimize**: $F_1 = R_T$ $F_2 = 0.5\dfrac{\ddot{z}_B}{0.2\ \mathrm{g}} + 0.5\dfrac{\dot{z}_D}{1.0}$	Max. overall length = 170 m Max. overall beam = 40 m Draft ≤ 6.5 m Total displacement = 10785 t $0.3 \leq L_{CB}/L \leq 0.7$ $\mathrm{KMT}_{original} \leq \mathrm{KMT}_{optimal}$ $\mathrm{KML}_{original} \leq \mathrm{KML}_{optimal}$ Hull waterplane area ≥ 300 m^2	$\left.\dfrac{R_T}{R_T^p}\right\vert_{Fr=0.460} \leq 1$ $\left.\dfrac{R_T}{R_T^p}\right\vert_{Fr=0.622} \leq 1$ $\ddot{z}_B \leq 0.2$ g $\dot{z}_D \leq 1.0$ m/s

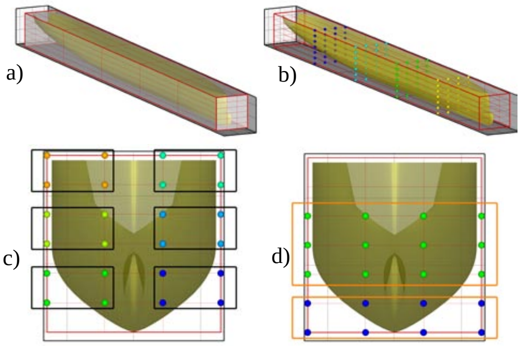


Fig. 30 Parametrization of the fast displacement catamaran by means of the FFD approach. Some of the vertices are grouped together, for a total of 50 design variables for each demihull.

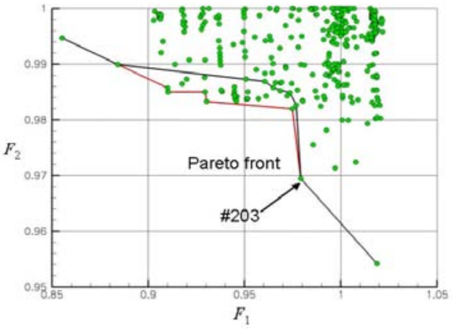


Fig. 31 Pareto front approximation obtained by means of the low-fidelity analysis with 50 design variables for the fast displacement catamaran.

phases. The problem is initially solved by using an LF potential flow solver, adopting 50 design variables (a relatively *high-resolution* design-space approach), in order to perform a careful exploration of the possible design solutions. After that, the overall geometrical trends have been identified and are used to define Phase 2 of the problem, solved—with a small number of design variables—by using a HF RANS code, with higher accuracy (and higher computational costs). The RANS solver is CFDShip-Iowa Version 4 [14, 15]. The strip-theory code normally used for seakeeping is not suitable for multihulls; therefore, a 3D frequency domain code is used, the INSEAN boundary-element method FreDOM [57].

In Phase 1, the parametrization of the hull is performed by using the FFD approach [95]; the hull is embedded in a control volume (see Fig. 30), and partitioned into a prescribed number of subdivisions. Intersections of the resulting grid can move in any direction, deforming the control volume. Potentially, each vertex of the FFD grid is a design variable, but some of the corners are linked together, so that they move the same amount in one direction. A complete scheme of the parametrization is reported in [95].

The solution of the first (LF-driven) problem is performed by adopting a hybrid optimization algorithm, reported in [95]. A metamodel is used for the approximation of the objective function, and the optimal solution is computed alternatively by a PSO algorithm and a derivate-free, direct-search algorithm (see [77] for more details). When an optimal solution is found (by using the metamodel), the optimum is verified by the CFD solver (the HF analysis)

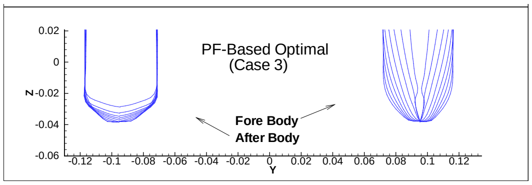


Fig. 32 Design fast displacement catamaran solution #203.

and then added to the training set of the metamodel to improve its predictions. Final approximation of the true Pareto front is reported in Fig. 31. The design solution labeled as #203 represents the lowest value of the second objective function that still improves the first objective function too. The nondimensional improvements are about −3% on both of the two objective functions. The body plan of solution #203 is shown in Fig. 32. This design was finally selected as the preferable shape for this problem. Solution #203 was obtained by performing 160 objective function evaluations for the initial training phase of the metamodel. Then 140 more objective-function evaluations were performed for each possible optimal design detected during the iterations. The time for the identification of the optimal solution is negligible, since a metamodel (whose computation is incomparably faster than a CFD solver) has been applied.

Comparison between the original and the Phase 1 optimal solution, in terms of wave elevation, is reported in Fig. 33, where clear improvements may be observed.

The geometry of the Phase 1 optimal solution has been assumed as a base for creating a number of hull shapes to be adopted for the Phase 2 optimization, solved with the `RC-MOGA` approach using a RANS solver for the analysis. The system parameters of the `RC-MOGA` are as follows: crossover rate = 0.75, population size = 16, and maximum generation = 50. The tentative designs are obtained as a weighted sum of three baseline designs which were produced manually using the trends and the indications raised from the previous phase. Hence, only two parameters are left free in the Phase 2 optimization. Their shapes are shown in Fig. 34. The resulting Pareto front is given in Fig. 35.

An interesting and balanced solution among those of the Pareto set has been selected for towing-tank verification, for which the hull lines are shown in Fig. 34. Two 4 meter models (the original and the optimal) have been built and tested in the INSEAN model basin (Fig. 36). Some results are reported in Figs. 37 and 38. An improvement on the resistance of the optimal hull is well documented for the whole speed range, except for the very low speeds (outside the optimizing range). At the optimization speed, a large improvement was found (above 12%).

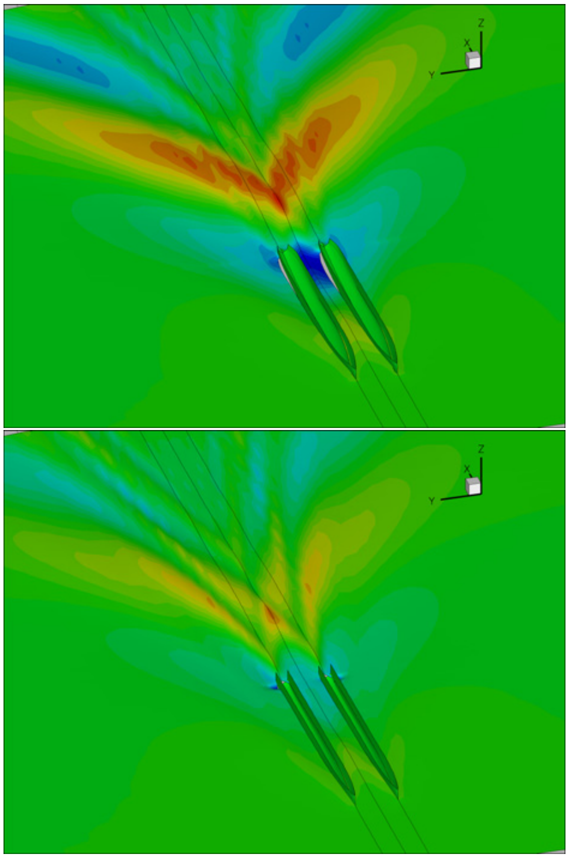


Fig. 33 Free surface elevation of the original (top) and optimal (bottom) fast displacement catamaran design. Red indicates high waves, blue indicates deep troughs.

Regarding vertical motions, the vertical acceleration at the bridge increased by around 2%, while the vertical velocity at the flight deck is reduced by about 16%, producing a final reduction of about 9% for the second objective function.

Single-Objective Application: Small Water-Plane Area Twin Hull (SWATH) Ship

The results—presented in this section and reported in full detail in [93]—are from a collaborative effort by Bath Iron Works; Flight Safety Technologies, Inc.; the Carderock Division, Naval Surface Warfare Center; and the University of Iowa, in response to the US Navy's need for design and analysis capability for a High-Speed Sea-Lift (HSSL) ship. The shape optimization was carried out through parallel design optimization using a number of flow solvers with different fidelity, based on

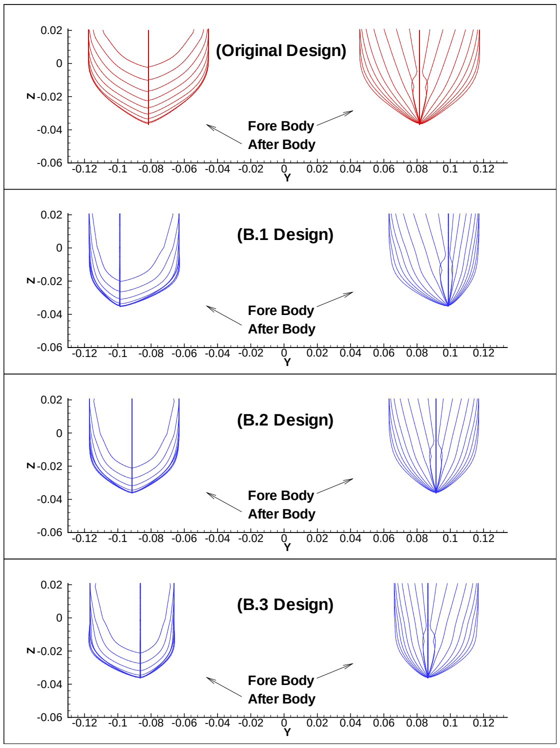


Fig. 34 Original fast displacement catamaran hull shape (top left) and three baseline hull shapes adopted for the solution of the optimization problem using the RANS solver.

the optimization opportunities identified from an analysis of the initial design provided by BIW. The BIW-SWATH was optimized in parallel using different potential-flow solvers as well as the URANS code, CFDShip-Iowa, adopting different optimization strategies. The results were cross verified using the different codes to check for consistency.

The capabilities of the code, CFDShip-Iowa, were extended during this study by implementing numerous applications [14, 15]. For ship motion predictions at an arbitrary heading, regular and irregular, and unidirectional and multidirectional waves were implemented using a Bretschneider spectrum for the wave frequency and $\cos^2$ spreading of the spectrum, as recommended by ITTC (1978). To allow for the computation of large-amplitude motions, a dynamic overset grid capability was used. This was accomplished using the interpolation tool SUGGAR [66].

The CFDShip-Iowa optimized SWATH was finally selected for model testing. The experimental data from the model testing of the baseline SWATH catamaran clearly showed that the conventional-design approach for multihulls resulted in substantial interference drag. The comparison of the numerical results from potential and viscous flow solvers showed that viscous phenomena dominated the SWATH optimization and hence necessitated a URANS solver for the optimization.

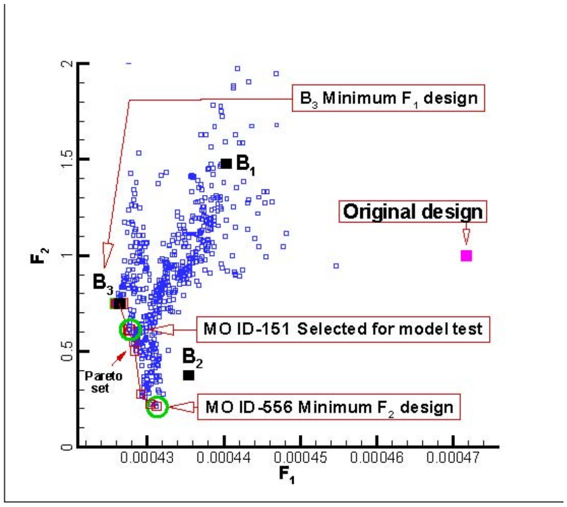


Fig. 35 Exploration performed by the MOGA in the space of the objective functions. The Pareto set is reported together with the performances of the original fast displacement catamaran hull plus the three hulls selected as the baseline for the morphing procedure. The selected design for experimental verification is also reported.

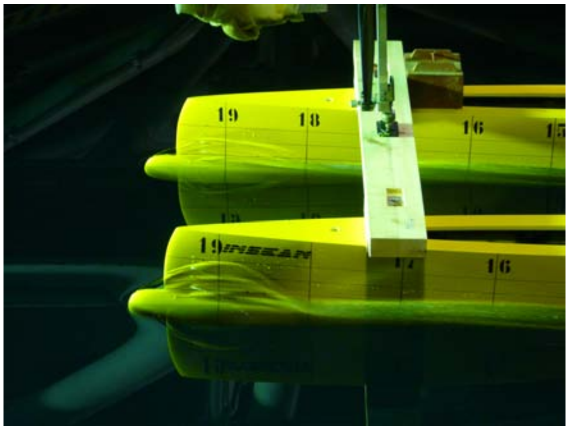


Fig. 36 The optimized fast displacement catamaran during model tests.

The objective function was total resistance at full scale with additional terms added for:

- amplitude of the near-field waves,
- panels with large aft-facing normals,
- total beam in excess of constraint,
- displacement less than baseline.

Design optimization based on potential flow showed the capacity to reduce the drag of multi-hull

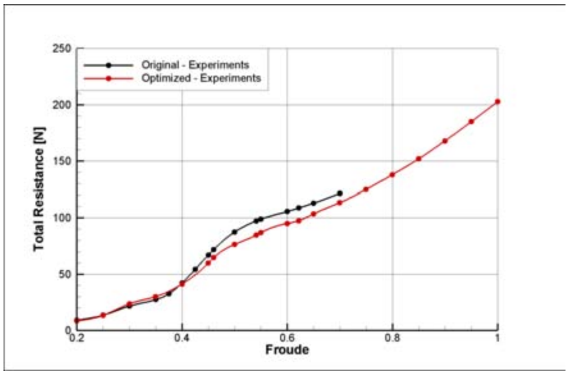


Fig. 37 Total resistance of the original and the optimal fast displacement catamaran hull shapes.

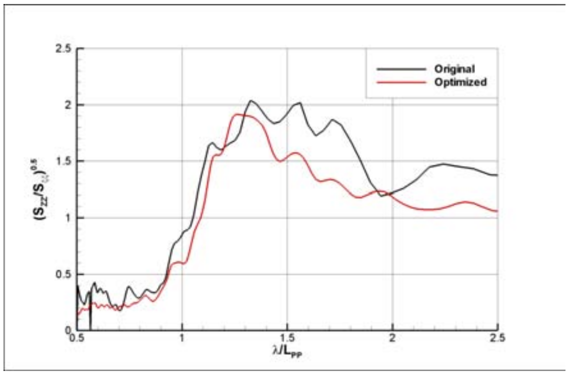


Fig. 38 Heave motion RAO at $Fr = 0.541$ for the original and the optimized fast displacement catamaran shapes.

vessels significantly by reshaping the longitudinal volume and demihull meanline distributions. However, sufficient constraints are required to prevent the optimized designs from taking on characteristics that increase drag due to phenomena not modeled in potential flow. For example, a significant increase in the bow-bulb sectional area, which manifests itself as an increase in the sectional beam-to-draft ratio if the bulb profile is constrained, can cause secondary flows that increase the viscous drag and offset a reduction in the vessel wave-making drag. This highlights the importance of a non-heuristic variable, fidelity approach, as discussed previously in this paper.

CFDShip-Iowa was interfaced with the Genetic Algorithm reported in this paper [96]. The optimization used a single objective function (the total resistance). Prior to the optimization runs, two kinds of sensitivity studies were performed: one-generation initial-population runs; and three-generation propagation runs. The purpose of these runs was to investigate the following:

- Initial feasible population size: the larger the initial population, the lower the number of generations required to get the optimal solution, the cost of doing this is the number of initial processors. Trial runs were conducted with initial population sizes of 6, 9 and 18, with 30, 45 and 90 processors, respectively. An initial population of 9 was deemed feasible as the 18 population case crashed due to one member being sent to a bad processor node. This case also took much longer in the batch queue.
- Number of solver iterations necessary to get the trends: The rank of each member in the population (lowest resistance to highest) determines how the current population will propagate. It is not necessary to get the fully converged solution at each generation. The rank becomes evident after 150 iterations for the captive model (w/o sinkage and trim). Full convergence requires 300–400 iterations for the captive model. For the free model (predicted sinkage and trim) the rank becomes evident after about 250 iterations. Full convergence requires 500–600 iterations.

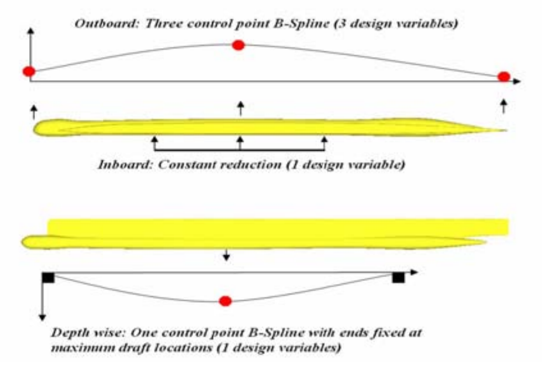

Outboard: Three control point B-Spline (3 design variables)
Inboard: Constant reduction (1 design variable)
Depth wise: One control point B-Spline with ends fixed at maximum draft locations (1 design variables)

Fig. 39 Geometry modification technique for the SWATH designed by BIW

Following the preliminary sensitivity studies and initial propagation runs, the final optimization run was carried out with a limit of 15 generations. The final optimization run used 5 design variables: 1 inboard, 3 outboard and 1 depth wise, based on the sensitivity studies (see Fig. 39).

CFDShip-Iowa predicted a total resistance reduction of 5.2% for model scale, which stems mainly from the reduction of C_r. At full scale, the contribution of C_r to the total resistance increases and hence the expected total resistance reduction in the full-scale ship is about 8.65%. A comparison of the wave elevation and the surface stream lines is shown in Fig. 40 The optimized SWATH shows reduced wave amplitudes in the interference region between the hulls, which is responsible for most of the drag

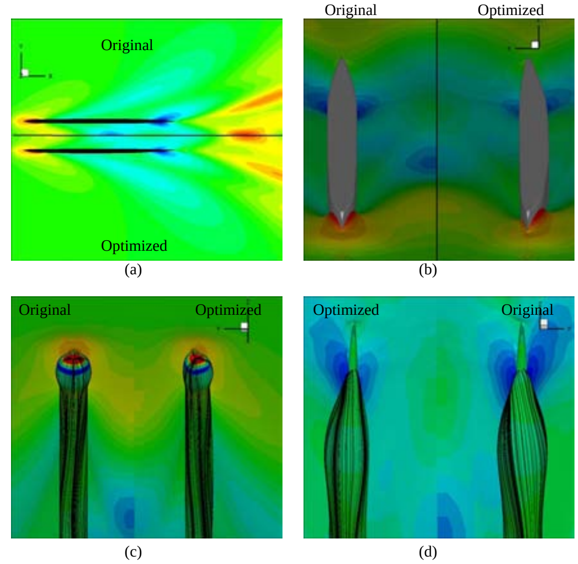


Fig. 40 Flow-field comparison between original and optimized geometries: (a) Free-surface elevation, top view; (b) Free-surface elevation, top bow view; (c) Free-surface elevation with stream lines and hull-surface pressure, bottom bow view; (d) Free-surface elevation with stream lines and hull-surface pressure, bottom stern view.

reduction. The local stream lines near the aft end of the optimized SWATH do not exhibit the cross flow seen in the original SWATH design. This turned out to be an added benefit as it relates to the vortical flow past the aft end of the hulls. Fig. 41 shows the iso-surface of Q depicting regions of vortical flow for the original and the optimized configurations. As seen, the optimized hull does not exhibit the vortex flow emanating from the aft end, which would be a benefit for propeller performance.

ROBUST AND MULTIDISCIPLINARY DESIGN OPTIMIZATION FOR SHIPS

In this section we introduce two relatively less diffused branches of SBD frameworks, namely methods for Robust Design and for Multidisciplinary Design Optimization.

As the engineering environment in general becomes extremely competitive, designers have become increasingly concerned with managing uncertainties—manufactured products deviate from designed products; actual products must perform under a variety of operating conditions. Robust Design methods, developed to improve product quality and reliability in industrial engineering, are to prevent such uncertainties. The tools of statistical decision theory, specifically Bayes principle, provide a sound framework in which to formulate the problems of Robust Design. The difficulty with exploiting this framework is computational, involving the numerical integration of expensive simulation outputs with respect to uncertain quantities. To what extent this difficulty can be overcome remains to be seen, but we are encouraged by the advances in computational tools as well as in computer hardware.

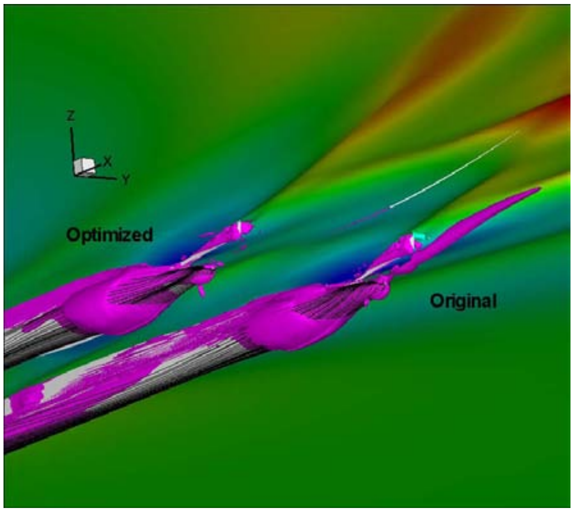


Fig. 41 Comparison of aft-end vortical flow between original and optimized hull forms using the iso-surface $Q = 5$.

Multidisciplinary Design Optimization (MDO) evolved as a new discipline during the late 1990s [90] and refers to an area of research that deals with the development of systematic approaches to the design of complex engineering systems governed by interacting physical phenomena. As summarized in [1], the traditional design approach has been about meeting requirements (i.e. the constraints) rather than finding optimal solutions. In the Mulltidisciplinary Design approach, the emphasis is on series of disciplinary simulations and individual disciplinary optimizations, with results reconciled among disciplines. On the other hand, MDO deals with the explicit manipulation of the design variables in the disciplinary computational models, in the context of the optimization algorithms (see [2] and the special issue [1] for a general introduction to definitions and for recent applications of MDO).

Robust Design using Numerical Simulations

Statistical decision theory, via the Bayes principle, provides a conceptual framework for quantifying uncertainty, but it involves the numerical integration of expensive simulation outputs with respect to uncertain quantities. For this reason, the application of statistical decision theory to Robust Design has rarely been attempted and lies at the frontier of current engineering practice.

Using ideas from statistical decision theory, the problem of Robust Design can be formulated as an optimization problem. We may consider an objective function of the form $f : A \times B \to \Re$ where $a \in A$ represents the design (controlled by the designer), $b \in B$ represents the uncertainty (*not* controlled by the designer), and $f(a;b)$ quantifies the design performance's loss when condition b occurs. The optimization problem could then be to find $a^* \in A$ such that, for every $b \in B$,

$$f(a^*;b) \leq f(a;b) \quad \forall a \in A.$$

The problem of finding $a^* \in A$ that simultaneously minimizes $f(a;b)$ for each $b \in B$ can be seen as an attempt to find a decision rule that simultaneously minimizes risk for every possible state of nature. This is a central problem of statistical decision theory, unfortunately it is *unsolvable.* There are methods to accomplish partial objectives [26]. The Minimax approach tries to minimize the risk of the worst-case scenario, often leading to designs that are too conservative.

Alternatively, the Bayes principle can be stated as:

$$\min_{a \in A} \varphi(a), \quad \varphi(a) = \int_B f(a;b)\, p(b)\, db,$$

where p is a probability density function on B. In other words, the Bayes principle seeks to minimize the average loss in a sense that can be tailored to the application, by a suitable choice of the probability distribution p. This formulation of the quality control problem was first proposed by Welch *et al.* [104]. Optimization under uncertainty therefore replaces our objective function f with the more complex function:

$$\varphi(a) = \int_B f(a;b)\ p(b)\, db.$$

If each evaluation of f is expensive (e.g. one RANS solution), the numerical integration of f is *very* expensive. The Design and Analysis of Computer Experiments (`DACE`, [103]) has been among the first approaches suggesting a simple procedure to minimize the use of expensive simulations through the use of surrogates. To integrate the kernel of the probability integral $K(b) = f(a;b)\ p(b)$, one can choose $b_1, b_2, \ldots, b_n$ and evaluate K at each b_i and approximate the integral by some quadrature of the form:

$$\int_B K(b)\, db \simeq \sum_{i=1}^{n} c_i\, K(b_i).$$

The `DACE` idea is to replace the exact K with an approximation, $\hat{K}$, and rely on the approximation:

$$\int_B K(b)\ db \simeq \int_B \hat{K}(b)\, db.$$

The approximation $\hat{K}$ can be obtained by interpolating values of K at some given b_i or, alternatively, by using some low-fidelity—computationally inexpensive—model $\hat{f}$ of f so that:

$$\hat{K}(b) = \hat{f}(a;b)\, p(b).$$

Even if applications of Robust-Design optimization methods to ship-design problems are infrequent, these techniques have already been applied in other related design areas such as the aerospace, aeronautical and automotive industries (see [69], [7]). Some preliminary results are reported in [22], where a bulk-carrier design problem—proposed in Parsons and Scott [70]—is addressed.

Multidisciplinary Design Optimization

The design of a ship encompasses interacting physical phenomena such as hydrodynamics, structural mechanics, and control, to name a few. These interactions make the ship a synergistic whole that is greater than the sum of its parts. Unfortunately, these interactions are often not easy to untangle, and the detailed design work has to be split into specialty areas centered on a physical phenomenon.

Examples of MDO applications to industrial fields other than the naval field can be found mainly in the aerospace and automotive industries (cf. [2, 1, 91]). To introduce some generalities about MDO formulations, simple two discipline problem is reported in Appendix C. MDO examples applied to ship hydrodynamic designs are still very limited, and are mostly connected with the early stages of the design (e.g. [48]). A recent example of MDO, from [12], presents simulations of the detailed design of the fin keel for a sailing yacht, simultaneously accounting for hydrodynamics and elasticity.

In Fig. 42, a picture of a classical configuration for this class of yacht is shown. The heavy bulb (sometimes up to 90% of the weight of the yacht!) is attached to the hull by a long thin fin keel (Fig. 43), responsible for the side reaction of the hull. The fin keel is asked to give an high side force without paying for it with large drag. Due to its extreme slenderness, the fin's deformation under the hydrodynamic loads is not negligible: (sailing teams often

Fig. 42 Classical configuration of an America's Cup Class Yacht.

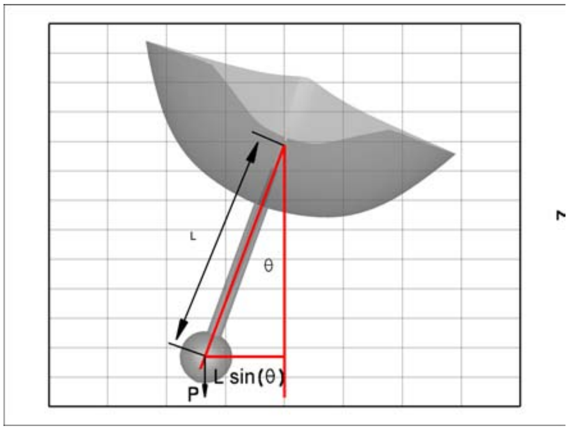


Fig. 43 The righting moment given by the ballast bulb.

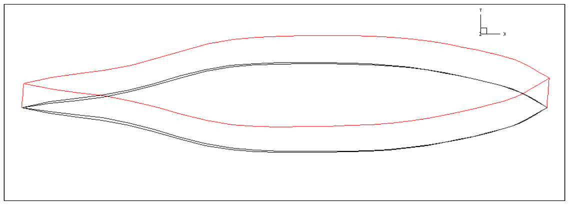


Fig. 44 Maximum deformation of elastic keel (thin red line), with respect to the rigid one (thick black line), at the junction with the keel bulb.

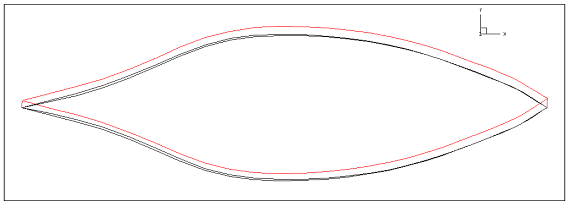


Fig. 45 Same as the previous figure but for the final optimized geometry. The thick black (thin red) line is the optimized geometry in undeflected (deflected) condition.

report a relatively large deformation of the fin keel as seen from aboard when the yacht is heeled).

A pure fluid-dynamic approach would simply consider the hull, the fin keel and the bulb as *rigid*, connected bodies. In an MDO framework these are considered instead as elastic, and the shape of the fin keel is hence modified by the *hydrodynamic* loads. As a result, the final performance of the yacht is also influenced by the structural behavior of the fin. In [12], the optimization problem is solved comparing different MDO formulations in a global optimization framework.

In [12] the shape of the fin keel is parameterized using 4 design variables. The modified geometry is obtained by superposition of a Béziér patch on the original fin keel. The original fin-keel design was taken from the available data for the America's Cup sailing yacht *Il Moro di Venezia* [6], as were the material properties (e.g. the elastic modulus of the material). The original fin keel was cylindrical—the section of the fin does not change throughout the span. The objective was to maximize the ratio of the side force to the drag. Two different codes were used for the solution of the structural and hydrodynamic problems. For the hydrodynamic problem, a free-surface potential-flow solver with lifting surfaces (the INSEAN code `WARP`, [5]) was applied, while a FEM solver was used for the structural problem. In the conditions analyzed, the elastic deformations produced a reduction in the efficiency of the fin of about 10% (the drag increased and the side force decreased).

A comparison of the initial and the final shapes shown in Figs. 44 and 45 reveals that the optimal shape is (as expected) thicker than the original one, which in turn leads to a reduced elastic deformation under the effect of the hydrodynamic loads, as can be observed by comparing the lateral bending. Even if the thickness is higher, the fin efficiency (the objective function) is increased by about 41%.

CONCLUSIONS

In recent years, computational tools for hydrodynamic optimization (also referred to as Simulation-Based Design methods) of ships have been developing at a fast pace. The key elements of these tools, besides the accuracy and robustness of the numerical flow solvers, are algorithms for constrained, continuous optimization, mesh deforming techniques, and surrogate and variable fidelity models. The paper briefly describes some gradient-based (for local optimization problems) and derivative-free (for global optimization) methods, analyzing some of the available techniques for the selection of the design parameters, for the construction of the metamodels and for variable-fidelity approaches. Robust design and multidisciplinary design optimization methods are finally briefly introduced. Some hydrodynamic ship design problems (either single- and multi-objective problems) are solved, demonstrating the maturity and applicability of these techniques to real-life design problems. The improvement of the accuracy and the robustness of the state-of-the-art numerical solvers and a deeper integration between optimization code developers and ship designers are probably the key points in the diffusion of these methodologies.

ACKNOWLEDGMENTS

This work has been partially sponsored by the U.S. Office of Naval Research under grants administered by Dr. Patrick Purtell (N000140210489, N000140210304 and N00140210256) and Dr. Ki-Han Kim (00140210256, 000140510617 and 00014-0810957). The authors thank Matteo Diez and Giovanni Fasano for their contributions on Robust Optimization and MDO, Russ Hoffman, Steve Smith, Charles Cary, Philip Taylor (Bath Iron Works) and Colen Kennell (NSWC/CD) for their help in the definition of realistic conditions for some of the design problems. The authors would like express their sincere appreciation to Dr. Arthur Reed (NSWC/CD) for his detailed and helpful suggestions both during the projects and in the preparation of this paper. Finally, we would also like to thank Suzanne Reed for her conscientious work in editing the manuscript.

REFERENCES

[1] Alexandrov, N.M. (2005) Editorial—Multidisciplinary Design Optimization. *Optimization and Engineering*, (Special issue on Multidisciplinary Design Optimization), **6**(1):5–7.

[2] Alexandrov, N.M. & M.Y. Hussaini (1997) Multidisciplinary Design Optimization—state of the art. *Proc. ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization*, SIAM Proc. Series.

[3] Alexandrov, N.M., R.M. Lewis, C.R. Gumbert, L.L. Green & P.A. Newman (2001) Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models. *J. Aircraft*, **38**(6):1093–1101.

[4] Baba, E. (1974) Ship Form Improvement by Use of Wave Pattern Analysis. *Japan Shipbuilding and Marine Engineering*, **8**(1):35–43.

[5] Bassanini P., U. Bulgarelli, E.F. Campana & F. Lalli (1994) The Wave Resistance Problem in a Boundary Integral Formulation. *Surveys on Mathematics for Industry*, **4**(3):151–194.

[6] Battistin, D., D. Peri & E.F. Campana (2005) Geometry and Resistance of the IACC Systematic Series *Il Moro di Venezia*. *Proc. 17th Chesapeake Sailing Yacht Symposium.*

[7] Beyer, H.G. & B. Sendhoff (2007) Robust optimization—A comprehensive survey. *Comput. Methods Appl. Mech. Eng.*, **196**(33-34):3190–3218.

[8] Bishop, R.C., W.F. Belknap, C. Turner, B. Simon & J.H. Kim (2005) Parametric Investigation on the Influence of GM, Roll Damping, and Above-Water Form on the Roll Response of Model 5613. Carderock Division, Naval Surface Warfare Center Report NSWCCD-50-TR-2005/027, 185 p.

[9] Campana, E.F., G. Fasano, D. Peri & A. Pinto (2007) Nonlinear Programming Approaches in the Multidisciplinary Design Optimization of a Sailing Yacht Keel Fin. *9th Int'l Conf. Numerical Ship Hydrodynamics*, Ann Arbor, Michigan.

[10] Campana, E.F., G. Fasano & A. Pinto (2006) Dynamic system analysis and initial particles position in Particle Swarm Optimization. *IEEE Symp. on Swarm Intelligence*, Indianapolis, Indiana.

[11] Campana, E.F., D. Peri, S. Lucidi, A. Pinto, G. Liuzzi & V. Piccialli (2009) New Global Optimization Methods for Ship Design Problems. *Optimization and Engineering*, doi: 10.1007/s11081-009-9085-3.

[12] Campana, E.F., D. Peri, A. Pinto, Y. Tahara, M. Kandasamy, F. Stern, C. Cary, , H. Hoffman, J. Gorski & C. Kennell (2006) Simulation Based Design of Fast Multihull Ships. *26th Symp. Naval Hydro.*, Rome, Italy.

[13] Campana, E.F., D. Peri, Y. Tahara & F. Stern (2006) Shape Optimization in Ship Hydrodynamics using Computational Fluid Dynamics. *Comput. Methods Appl. Mech. Eng.*, **196**(1-3):634–651.

[14] Carrica, P.M., R. Wilson, R. Noack, T. Xing, M. Kandasamy, J. Shao, N. Sakamoto & F. Stern (2006) A Dynamic Overset, Single-Phase Level Set Approach for Viscous Ship Flows and Large Amplitude Motions and Maneuvering. *26th ONR Symp. Naval hydro.*, Rome, Italy..

[15] Carrica, P.M., R. Wilson & F. Stern (2007) An Unsteady Single-Phase Level Set Method for

Viscous Free Surface Flows. *Int'l J. Numerical Methods Fluids*, **53**(2):229–256.

[16] Chang, K.J., R.T. Haftka, G.L. Giles & P.-J. Kao (1993) Sensitivity-based Scaling for Approximating Structural Response. *J. Aircraft*, **30**(2):283–288.

[17] Coello Coello, C.A., G.B. Lamont & D.A. Van Veldhuizen (2007) *Evolutionary Algorithms for solving Multi-Objective Problems*, Springer, New York.

[18] Conn, A.R., N.I.M. Gould & P.L. Toint (2001) *Trust region methods*, MPS-SIAM Series on Optimization, Philadelphia.

[19] Day, A.H. & L.J. Doctors (2000) The survival of the fittest—Evolutionary tools for Hydrodynamic Design of Ship Hull Form. *Trans. Royal Inst. Naval Architects*, **142**C:182–197.

[20] Deb, K. (1999) Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, **7**(3):205–230.

[21] Deb, K. (2001) *Multi-Objective Optimization using Evolutionary Algorithms*, J. Wiley & Sons, New York.

[22] Diez, M. & D. Peri (2009) Global optimization algorithms for robust optimization in naval design. *8th Int'l Conf. Computer Applications and Information Technology in the Maritime Industries* COMPIT'09, Budapest, Hungary.

[23] Di Mascio, A., R. Broglia & B. Favini (2000) A second-order Godunov-type scheme for Naval Hydrodynamics. In: *Godunov Methods: Theory and Application*, Kluwer Academic Publishers, Singapore.

[24] Duvigneau R. & M. Visonneau (2004) Hybrid genetic algorithms and artificial neural networks for complex design optimization in CFD. *Int'l J. Num. Methods Fluids*, **44**(11):1257–1278.

[25] Farrashkhalvat M. & J.P. Miles (2003) *Basic Structured Grid Generation*. Butterworth-Heinemann (Elsevier Science), Oxford.

[26] Ferguson, T.S. (1967) *Mathematical Statistics: A Decision Theoretic Approach*. Academic Press, New York.

[27] Gano, S.E., J.E. Renard & B. Sanders (2004) Variable Fidelity Optimization Using a Kriging Based Scaling Function. *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conf.*, Albany, New York.

[28] Giering, R. & T. Kamnski (1998) Recipes for Adjoint Code Construction. *ACM Trans. Math. Software*, **24**(4):437–474.

[29] Giering, R., T. Kamnski & T. Slavig (2005) Generating efficient derivative code with TAF: Adjoint and tangent linear Euler flow around an airfoil. *Future Generation Computer Systems*, **21**(8):1345–1355.

[30] Gill, P.E., M.A. Murray & M.H. Wright (1981) *Practical optimization*, Academic Press, New York.

[31] Goel, T., R.T. Haftka, W-. Shyy & N.V. Queipo (2007) Ensemble of surrogates. *Structural Multidisciplinary Optimization*, **33**(3):199–216.

[32] Griewank, A. (2000) Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation. *Frontiers in Applied Mathematics*, **19**, SIAM.

[33] Haug, E.J., K.K. Choi & V. Komkov (1986) Design Sensitivity Analysis of Structural Systems. *Mathematics in Science and Engineering*, **177**, Academic Press, Orlando.

[34] Haykin, S. (1994) *Neural Networks: A Comprehensive Foundation*. Macmillan, New York.

[35] Hedayat, A.S., N.J.A. Sloane & J. Stufken (1999) *Orthogonal Arrays: Theory and Applications*, Springer-Verlag, New York.

[36] Hoffman J.D. (1992) *Numerical Methods For Engineers and Scientists*. McGraw-Hill, Inc., New York.

[37] Holland, J. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan.

[38] Hsiung, C.C. (1981) Optimal Ship Forms for Minimum Wave Resistance. *J. Ship Res.*, **25**(2):95–116.

[39] Hsiung, C.C. & D. Shenyan (1984) Optimal Ship Forms for Minimum Total Resistance. *J. Ship Res.*, **28**(3):163–172.

[40] Huan, J. & T.T. Huang (1998) Sensitivity Analysis Methods for Shape Optimization in Non-Linear Free Surface Flow. *Osaka Coll. Advanced CFD Applications to Ship Flow and Hull Form Design*, Osaka, Japan.

[41] ITTC, (1957) *Proc. 8th ITTC—International Towing Tank Conference*, Madrid, Spain.

[42] Jameson, A. & J. Reuther (1988) Aerodynamic Design Via Control Theory. *J. Scientific Computing*, **3**(3):233–260.

[43] Jin, R., W. Chen & T. Simpson (2001) Comparative Studies of Metamodeling Techniques under Multiple Modeling Criteria. *Structural and Multidisciplinary Optimization*, **23**(1):1–13.

[44] Jones, D.R. (2002) A taxonomy of global optimization methods based on response surfaces. *J. Global Optimization*, **21**(4):345–383.

[45] Jones, D.R., C.D. Perttunen & B.E. Stuckman (1993) Lipschitzian optimization without the Lipschitz constant. *J. Optim. Theory and Appl.*, **79**(1):157–181.

[46] Kennedy, J. & W.M. Spears (1998) Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multimodal Problem Generator. *IEEE Int'l Conf. Evolutionary Computation*, Anchorage, Alaska.

[47] Kennedy, R. & R.C. Eberhart (1995) Particle Swarm Optimization. *Proc. IEEE Int'l. Conf. Neural Networks*, IEEE Service Center, Piscataway, NJ, **IV**:1942–1948.

[48] Kodiyalam, S. & C. Yuan (2000) Evaluation of Methods for Multidisciplinary Design Optimization (MDO), Phase I. NASA/CR-2000-210313.

[49] Kolda, T.H., R.M. Lewis & V. Torczon (2003) Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Review*, **45**(3):385–482.

[50] Kuhn, J., K. Chevalier, E. Schlageter, C. Scragg & D. Wyatt (2007) The Use of Linear Programming and Basis Functions for Hull-Form Optimization. *9th Int'l Conf. Numerical Ship Hydrodynamics*, Ann Arbor, Michigan.

[51] Kuipers, L. & H. Niederreiter (2006) *Uniform Distribution of Sequences*, Dover Publications, New York.

[52] Lin, W.C., W.C. Webster & J.V. Wehausen (1963) Ships of Minimum Total Resistance. *Int'l Seminar on Theoretical Wave Resistance*, U. Michigan, Ann Arbor, Michigan.

[53] Lin, Y., F. Mistree, J.A. Allen, K.L. Tsui & V.C.P. Chen (2004) Sequential Metamodeling in Engineering Design. *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conf.*, Albany, New York.

[54] Lions, J.L. (1971) *Optimal Control of Systems Governed by Partial Differential Equations*, Springer-Verlag, New York.

[55] Löner, R., O. Soto & C. Yang (2003) An Adjoint-Based Methodology for CFD Optimization Problems. *41st Aerospace Sciences Meeting*, Reno, Nevada.

[56] Lucidi, S. & M. Sciandrone (2002) On the Global Convergence of Derivative Free Methods for Unconstrained Optimization. *SIAM J. Optimization*, **13**(1):97–116.

[57] Lugni, C., A. Colagrossi, M. Landrini & O.M. Faltinsen (2004) Experimental and Numerical Study of Semi-displacement Mono-hull and Catamaran in calm water and incident waves. *25th Symp. Naval Hydro.*, St. Johns, Newfoundland and Labrador.

[58] Marsden, A.L., W. Wang, J.E. Dennis, Jr. & P. Mion (2004) Optimal Aeroacoustic Shape Design Using the Surrogate Management Framework. *Optimization and Engineering*, **5**(2):235–262.

[59] Martin, J.D. & T.W. Simpson (2003) A Study on the Use of Kriging Models to Approximate Deterministic Computer Models. *ASME Design Engineering Technical Conf. and Computers and Information in Engineering Conf.*, No. DET2003/DAC-48762, Chicago, Illinois.

[60] Martinelli, L. & G.W. Cowles (2000) A control-theory based method for shape design in incompressible viscous flow using RANS. *AIAA Fluids 2000*, AIAA paper 00-2544, Denver, Colorado.

[61] Martinelli, L. & A. Jameson (2007) An Adjoint Method for Design Optimization of Ship Hulls. *9th Int'l Conf. Numerical Ship Hydrodynamics*, Ann Arbor, Michigan.

[62] Miettinen, K.M. (1999) *Nonlinear Multiobjective Optimization*, Kluwer Academic Publisher.

[63] Myers, R.H. & D.C. Montgomery (1997) *Response Surface Methodology*, J. Wiley & Sons, New York.

[64] Meyers, W.G., T.R. Applebee & A.E. Baitis (1981) User's Manual for the Standard Ship Motion Program, SMP81. Carderock Division, Naval Surface Warfare Center Report DTNSRDC/SPD-0936-01.

[65] Newman, J.C., III, R. Pankajakshan, D.L. Whitfield & L.K. Taylor (2002) Computational Design Optimization Using RANS. *24th Symp. Naval Hydro.*, Fukuoka, Japan.

[66] Noack, R. (2005) SUGGAR: a General Capability for Moving Body Overset Grid Assembly. *17th AIAA Computational Fluid Dynamics Conf.*, Toronto, Ontario, Canada.

[67] Ono, I. & S. Kobayashi (1997) A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover. *7th Int'l Conf. Genetic Algorithms*, pp. 246-253.

[68] Papanikolaou, A. & M. Androulakakis (1991) Hydrodynamic Optimization of High-Speed SWATH. *1st Int'l Conf. Fast Sea Transportation* (FAST), Trondheim, Norway.

[69] Park, G.J., T.H. Lee, K.L. Lee & K.H. Hwang (2006) Robust Design: An Overview. *AIAA J.*, **44**(1):181–191.

[70] Parsons, M.G. & R.L. Scott (2004) Formulation of multicriterion design optimization problems for solution with scalar numerical optimization methods. *J. Ship Res.*, **48**(1):61–76.

[71] Parsopoulos, K.E. & M.N. Vrahatis (2002) Recent approaches to global optimization problems through Particle Swarm Optimization. *Natural Computing*, **1**(2-3):235–306.

[72] Peri, D. (2009) Self-Learning Metamodels for Optimization. *Ship Tech. Res.*, **56**(3):94–108.

[73] Peri, D. & E.F. Campana (2005) High-Fidelity Models and Multiobjective Global Optimization Algorithms in Simulation-Based Design. *J. Ship Res.*, **49**(3):159–175.

[74] Peri, D. & E.F. Campana (2008) Variable Fidelity and Surrogate Modelling in Simulation-Based Design. *27th Symp. Naval Hydro.*, Seoul, S. Korea.

[75] Peri, D., E.F. Campana & U.P. Bulgarelli (2002) Optimal shape design of a surface combatant with reduced wave pattern. *NATO AVT 90 Spring 2002 Conf.*, Paris, France, 22-26 April .

[76] Peri, D., M. Rossetti & E.F. Campana (2001) Design optimization of ship hulls via CFD techniques. *J. Ship Res.*, **45**(2):140–149.

[77] Pinto, A., D. Peri & E.F. Campana (2004) Global Optimization Algorithms in Naval Hydrodynamics. *Ship Tech. Res.*, **51**(3):123–133.

[78] Pinto, A., D. Peri & E.F. Campana (2007) Multiobjective Optimization of a Containership using Deterministic Particle Swarm Optimization. *J. Ship Res.*, **51**(3):217–228.

[79] Pironneau, O. (1974) On Optimum Design in Fluid Mechanics. *J. Fluid Mech.*, **64**(1):97–110.

[80] Ragab S.A. (2001) An Adjoint Formulation for Shape Optimization in Free-Surface Potential Flow. *J. Ship Res.*, **45**(4):269–278.

[81] Rao, S.S. (1996) *Engineering Optimization: Theory and Practice*, Wiley-Interscience.

[82] Sacks, J., W.J. Welch, T.J. Mitchell & H.P. Wynn (1989) Design and Analysis of Computer Experiments. *Statistical Science*, **4**(4):409–423.

[83] Salvesen, N., C.H. von Kerczek, C.A. Scragg, C.P. Cressy & M.J. Meinhold (1985) Hydronumeric design of SWATH-ships. *Trans. SNAME*, **93**:325–346.

[84] Samareh, J.A. (2005) Geometry and Grid/Mesh Generation Issues for CFD and CSM Shape Optimization. *Optimization and Engineering*, **6**(1):21–32.

[85] Scragg, C.A., A.M. Reed, D.C. Wyatt & T.J. Ratcliffe (1998) Hull Form Development of the *Sea Shadow* and Applications of the Technology to Monohulls. *Trans. SNAME*, **106**:443–483.

[86] Sederberg, T.W. & S.R. Parry (1986) Free-Form Deformation of Solid Geometric Models. *Proc. SIGGRAPH '86, Computer Graphics*, **20**(4):151–159.

[87] Shi, Y. & R.C. Eberhart (1998) Parameter Selection in Particle Swarm Optimization. *7th Annual Conf. Evolutionary Programming*, San Diego, California.

[88] Shi, Y. & R.C. Eberhart (1998) A Modified Particle Swarm Optimizer. *Proc. IEEE Int'l Conf. Evolutionary Computation*, Anchorage, Alaska.

[89] Simpson, T.W., J. Peplinski,h, P.N. Koc & J.K. Allen (2001) Metamodels for Computer-Based Engineering Design: Survey and Recommendations. *Engineering with Computers*, **17**(2):129–150.

[90] Sobieszczanski-Sobieski, J. (1995) Multidisciplinary Design Optimization: an Emerging, New Engineering Discipline. *Advances in Structural Optimization*, J. Herskovits (Ed.), Kluwer Academic Publishers, pp. 483–496.

[91] Sobieszczanski-Sobieski, J. & R.T. Haftka (1997) Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments. *Structural Optimization*, **14**(1):1–23.

[92] Statnikov, R.B. & J.B. Matusov (1995) *Multicriteria Optimization and Engineering*, Chapman & Hall, New York.

[93] Stern, F., P. Carrica, M. Kandasamy, S.-K. Ooi, J. Gorski, *et al.* (2008) Computational Hydrodynamic Tools for High-Speed Sealift: Phase II Final Report. IIHR Technical Report No. 465, Iowa City, Iowa .

[94] Stern, F., J. Longo, R. Penna, A. Olivieri, T.J. Ratcliffe & H. Coleman (2000) International collaboration on benchmark CFD validation data for surface combatant DTMB model 5415. *23rd Symp. Naval Hydro.*, Val de Reuil, France.

[95] Tahara, Y., D. Peri, E.F. Campana, M. Kandasamy & F. Stern (2008) Single and Multiobjective Design Optimization of a Fast Multihull Ship: numerical and experimental results. *27th Symp. Naval Hydro.*, Seoul, S. Korea.

[96] Tahara, Y., D. Peri, E.F. Campana & F. Stern (2008) CFD-Based Multi-Objective Optimization of a Surface Combatant by Using Global Optimization Methods. *J. Mar. Sci. Tech.*, **13**(2):95–116.

[97] Tahara, Y., S. Tohyama & T. Katsui (2005) CFD-based multiobjective optimization method for ship design. *Int'l J. Numer. Meth. Fluids* **52**(5):499–527.

[98] Torn, A.A. & A. Zilinskas (1989) *Global Optimization*. Springer-Verlag, Berlin, Germany.

[99] Valorani, M., D. Peri & E.F. Campana (2000) Efficient Strategies to Design Optimal Ship Hulls. *AIAA 8th Multidisciplinary Analysis and Optimization Conf.*, AIAA 2000-4731.

[100] Valorani, M., D. Peri & E.F. Campana (2003) Sensitivity Analysis Techniques to Design Optimal Ship Hulls. *Optimization and Engineering*, **4**(4):337–364.

[101] Venter, G. & J. Sobieszczanski-Sobieski (2003) Particle Swarm Optimization. *AIAA J.*, **41**(8):1583–1589.

[102] Webster, W.C. & J.V. Wehausen (1962) Schiffe geringsten Wallenwiderstandes mit vorgegebenem Hinterschiff. *Schiffstechnik*, **9**:62–68.

[103] Welch, W.J. & J. Sacks (1991) A system for quality improvement via computer experiments. *Communications in Statistics: Theory and Methods*, **20**(2):477–495.

[104] Welch, W.J., T.K. Yu, S.M. Kang & J. Sacks (1990) Computer experiments for quality control by parameter design. *J. Quality Technology*, **22**(1):15–22.

[105] Wilson, R.V., E. Paterson & F. Stern (2000) Verification and validation for RANS simulation of a naval combatant. *Proc. Workshop on Numerical Ship Hydro.*, Gothenburg, Sweden.

[106] Zalek, S.F., M.G. Parsons & R.F. Beck (2006) Evolutionary Multicriterion Optimization for Propulsion and Seakeeping. *26th Symp. Naval Hydro.*, Rome, Italy.

APPENDIX A—CONSTRAINTS IN OPTIMIZATION

The fundamental role of constraints in the solution of an optimization problem can be easily obtained from a simple example. Let's assume our optimization problem to be the following:

$$\begin{aligned} \text{minimize} \quad & F(x_1, x_2) \\ \text{subject to} \quad & x_2 < -x_1^2 + 10 \\ & x_2 > x_1^2 - 10 \end{aligned} \tag{15}$$

The feasible set (shown in Fig. 46) is represented by the portion of space $\Re^2$ included between the two curves $x_2 = x_1^2 - 10$ and $x_2 = -x_1^2 + 10$. The set is closed and convex, *i.e.* each point of a segment connecting any pair of points of the feasible set still belongs to the feasible set. This property of the problem allows an easy exploration of the feasible set. On the contrary, if we add a new constraint to the previous problem, the problem is now:

$$\begin{aligned} \text{minimize} \quad & F(x_1, x_2) \\ \text{subject to} \quad & x_2 < -x_1^2 + 10 \\ & x_2 > x_1^2 - 10 \\ & x_1^2 + x_2^2 > 4 \end{aligned} \tag{16}$$

(the resulting feasible set is shown in Fig. 47). In this second example, which will also be helpful in the discussion about local and global optimization methods, the feasible set is non-convex and disjoint (non-connected), presenting two distinct islands. If we try to proceed along a continuous path inside the feasible set, starting from an initial point inside one of the two islands, we cannot reach the points of the second island. Since local optimization methods are used to proceed along pseudo-continuous paths, this means that local optimization methods are not able to fully explore this class of feasible sets. This is one reason to switch from local to global optimization methods, which do not suffer from this inconvenience, and will be discussed later. By a further shift of the constraints, we can reduce the extension of the feasible set, so that the identification of a feasible solution may become very difficult. This situation, which may happen in real problems, is not easily detected since the correlation between design parameters and feasible set is often not as straightforward as in these algebraic examples.

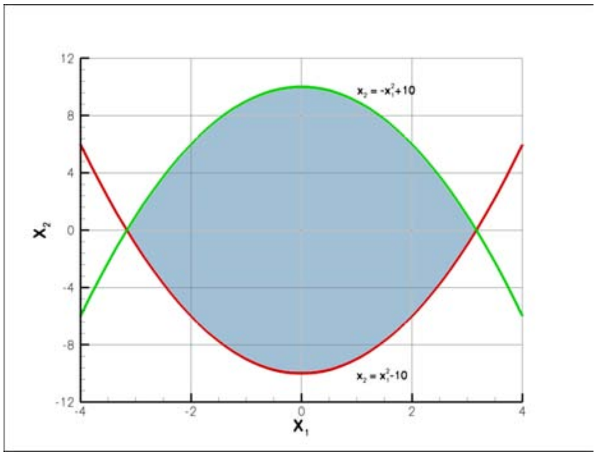


Fig. 46 Feasible set resulting from the example in Eq. (15). The x_1- and x_2-axes are not the same scale. The feasible set is represented by the shaded region.

The application of geometrical constraints often translates itself into bounds:

$$b^m{}_i \le x_i \le b_i^M, \quad i = 1, \ldots, N,$$

defining a parallelepiped Π in the N-dimensional design variable space. In a similar way functional constraints may also be imposed, assuming that

$$c_j^m \le g_j(\mathbf{X}) \le c_j^M, \quad j = 1, \ldots, l, \tag{17}$$

where the $g_j(\mathbf{X})$ describes some performance as a function of the design variables—for instance, the ship's total resistance at a given speed, or the requirements concerning the stability of the ship—which the designer may decide to confine within a suitable range. The constraints in Eq. 17, ap-

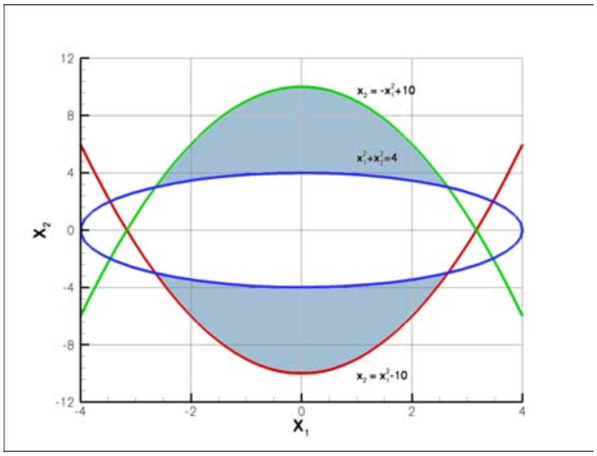


Fig. 47 Disjoint feasible set resulting from the example in Eq. (16). The x_1- and x_2-axes are not the same scale. The feasible set is represented by the shaded region.

plied to Π, single out the feasible set, $\mathcal{S}$, in the N-dimensional design space, *i.e.*, $\mathcal{S} \subset \Pi$ is the set of design solutions that satisfy all the constraints. The shape of the feasible set, $\mathcal{S}$, may result in a non-convex, non-connected region; this represents a crucial difficulty in the solution of the optimization problem. In fact, some optimization algorithms produce a convergent sequence of trial solutions along a nearly-continuous path, and are unable to jump from one side to another of a non-connected feasible set. As a consequence, this class of algorithms may be highly penalized by the selection of the initial solution, because the starting point may be in the wrong portion of the feasible set. Furthermore, if the feasible set is non-convex, they may be stuck in a corner, unable to proceed toward the optimal point. In general, constraints need to be accounted for, for proper treatment in an optimization problem, the final goal being the translation of a constrained problem into an unconstrained one.

APPENDIX B—PARAMETRIZATION

In deforming the geometry of a hull form, one has to follow some general guidelines:

- if the modified part is a subset of the entire hull, the modified geometry has to join the original design without discontinuities in the first and second derivative;
- the number of design variables should be kept as small as possible to minimize the number of evaluations of the gradient of the objective function, but ...
- ... the hull modification algorithm should be as flexible as possible in order to allow the analysis of a wide range of possible solutions.

Details of the Béziér-patch approach

A simple way to deform the ship geometry is to superimpose a Béziér patch on the hull that gradually reduces to zero perturbation when approaching the unmodified portion of the hull. The patch is controlled by a Béziér frame of $m \times n$ nodes, which are related to the patch via:

$$Y_B(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} I_{n,i}(u)\, J_{m,j}(v)\, p_{i,j}, \qquad (18)$$

where Y_B are the coordinates of the Béziér surface, u and v are nondimensional parameters along the x- and z-directions, $p_{i,j}$ are the y-coordinates of the Béziér frame nodes and

$$I_{n,i} = \binom{n}{i} (1-u)^{n-i}\, u^i$$
$$J_{m,j} = \binom{m}{j} (1-v)^{m-j}\, v^j.$$

Since the Béziér surface is defined on a 2D-unit square, we need to map the hull surface onto an unit square. To do that, we can use the coordinate lines provided by the regular structured grid commonly used in CFD codes to digitalize the hull surface. A structured regular $(N \times M)$ grid defines two indexes for each grid point, (say from 1 to N and from 1 to M). We can normalize these values by the maximum value so that each grid point is now associated with a pair of real numbers included in $[0:1]$. The Béziér surface is defined by a net built on this unit square. Since the Béziér surface is defined as a continuous function in the selected portion of the 2D space, we can compute the height of this surface on every point of the unit square, in particular those coordinates which are associated with the grid points. These values are now simply superimposed on the y-offsets of the hull to obtain the modified geometry via

$$Y_{H_{mod}} = Y_{H^o}(u,v) + Y_B(u,v). \qquad (19)$$

More patches can be used allowing for a complete modification of the geometry along the three directions.

Details on the Free-Form Deformation (FFD) approach

The FFD, introduced by Sederberg and Parry [86] in computer graphics, is a very flexible approach to deform a 3D object, whose geometry is given by points. This approach can be essentially reduced to a 4D-Béziér patch to be applied to the hull surface. In fact, in the previous superposition approach, we are using a perturbation function mapping an unit square defined by the nodal indexes. If we now define a box surrounding the hull surface, we can define a Béziér polynomial inside this 3D domain, producing a scalar function of the 3D space. This function, defined as

$$F(u,v,w) = \sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{k=0}^{l} I_{n,i}(u) \times J_{m,j}(v)\, K_{l,k}(w)\, p_{i,j,k}, \tag{20}$$

with the same structure as Eq. 18, could represent the displacement along a prescribed direction, and is applied to every hull point in order to produce the deformed geometry. The definition of coefficients is similar to those for Equation 18, and comes again from the theory of the Béziér polynomials. The Béziér function is defined inside the box placed around the object to be deformed, and we can compute the deformation on every point inside this box. The deformation is not defined outside.

Details on the Morphing approach

In the morphing approach, the deformation is produced starting from a number of prescribed hull shapes. As a consequence, we need P designs for the same hull, and we also need a computational grid with the same subdivision. We are going to adopt a mapping between the grid points of the different designs based on the grid topology. If we have a suite of grids, all subdivided into $(N \times M)$ intervals, we can pick a generic (i,j) point on the grid, thus defining

$$X(i,j) = w_1 X_1(i,j) + w_2 X_2(i,j) + \cdots + w_p X_p(i,j),$$

plus the condition

$$\sum_{i=1}^{P} w_i = 1.$$

Here $X_k(i,j)$ represents the vector of coordinates of the generic point with grid coordinates, (i,j), belonging to the kth grid. This approach produces a linear blending among the available geometries, which obviously limits the possible shapes implicitly defined by the designs provided. However, we can simply investigate some specific solutions by means of a really limited number of parameters ($1-P$, due to the condition of the sum of weights equal to 1). The transition from one shape to another can also be different than the linear one, and the reference grid can be also of different topology from design to design, allowing a larger variety of hull shapes.

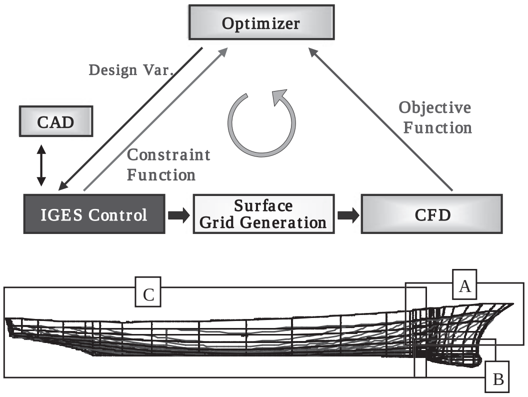


Fig. 48 Implementation of CAD-based hull-form modification in the optimization environment (IGES, Initial Graphics Exchange Specification).

CAD-based approach

To modify the ship geometry, a CAD-based hull-form modification method was adopted. Two approaches are possible, i.e., CAD direct control and CAD emulation approaches. Both approaches were successfully demonstrated by the present authors [97] and others. Here, the CAD emulation approach is reported. As shown in Fig. 48, a module is implemented in order to emulate manual CAD manipulation of the mathematical surface. This approach offers an advantage over the CAD direct-control approach for complete independence from the CAD system, *i.e.*, designers are able to use any CAD system and give and receive the initial, optimized hull-form geometry in Initial Graphics Exchange Specification (IGES) data format.

A NURBS surface is given mathematically by:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j}},$$

where u and v are the parameters; $N_{i,p}$ and $N_{j,q}$ are normalized B-spline basis functions of degree p and q in the u and v directions, respectively; $P_{i,j}$ are the vectors of the control points; and $w_{i,j}$ are the weights.

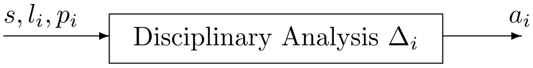
Finally, the surface is defined by $[(n+1)(m+1)]$ control points and weights, and knot vectors of $n+p+2$ and $m+q+2$ elements in the u and v directions, respectively. A modified surface is defined corresponding to new location vectors, P^n, so that:

$$P^n_{i,j} = P^0_{i,j} + \delta P_{i,j},$$

where $P^0_{i,j}$ and $\delta P_{i,j}$ are the original and displacement location vectors. $\delta P_{i,j}$ can be the design variables of the optimization problem.

APPENDIX C—A TWO-DISCIPLINE MDO PROBLEM

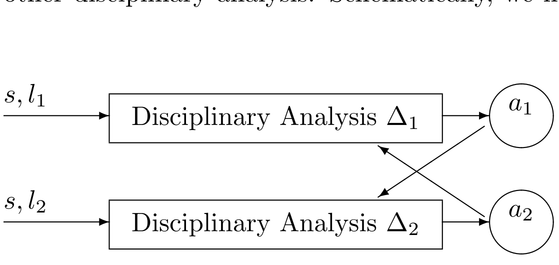
We introduce here a simple, two-discipline MDO problem (Δ_1 and Δ_2): the hydroelastic coupling between hydrodynamics (Δ_1) and structural analysis (Δ_2) for the keel fin of a sailing yacht in steady forward motion with a drift and a yaw angle as in [12]. The two disciplinary systems are the basic elements of the MDO problem. We assume that each discipline is based on a disciplinary analysis (from simple algebraic formulas to complex PDEs) that may be schematically depicted as an input-output relation:



The input of each discipline are a set of design variables, (s, l_i), and parameters, p_i, and the analysis produces a set of outputs, a_i. The *system-level* design variables, s, are those shared by both disciplines. The *disciplinary* design variables, l_1 and l_2, are local to Δ_1 and Δ_2, respectively. Parameters, p_i, are derived from the analysis outputs, a_j, $j \neq i$, of the other discipline. They are not directly manipulated by the designer in Δ_i. In our hydroelastic example, the input, p_1, from structures to hydrodynamics would include the fin keel shape, while the input, p_2, from hydrodynamics to structures would include the hydrodynamic loads.

The disciplinary analyses have the functional form $a_i = A_i(s, l_i, p_i)$. A_1 and A_2 are assumed to be independently solvable. In our hydroelastic example, given the shape of the fin (the values p_1) one can solve A_1 and obtain the flow field and the pressure around the fin (a_1) and analogously for A_2.

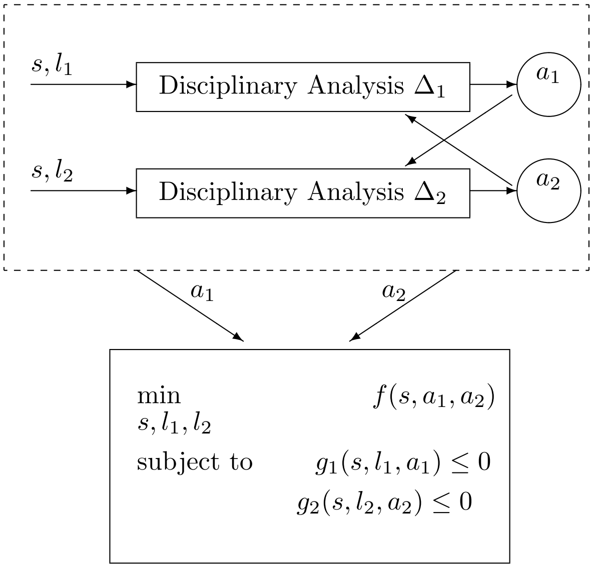
Now, in the context of the MDO problem, the coupled Multidisciplinary Analysis System (MDA) reflects the physical requirement that a solution simultaneously satisfy the two disciplinary analyses. The input parameters, p_i, for each discipline are now required to correspond to some (or all) of the outputs, a_j, from the other disciplinary analysis. Schematically, we have:


s, l_1
Disciplinary Analysis Δ_1
a_1
s, l_2
Disciplinary Analysis Δ_2
a_2

The multidisciplinary analysis system is therefore given by the simultaneous system of equations:

$$\begin{cases} a_1 = A_1(s, l_1, a_2) \\ a_2 = A_2(s, l_2, a_1), \end{cases} \tag{21}$$

where the solution of the discipline Δ_1 gives the input for the discipline analysis Δ_2 and vice versa, therefore implicitly defining a_1 and a_2 as functions of (s, l_1, l_2). Solving the fully coupled system Eq. (21) leads to a full multidisciplinary analysis (MDA). The solution is in this case a *consistent* solution that satisfies both disciplines. Again, if Δ_1 represents hydrodynamic analysis of the flow around the fin keel and Δ_2 represents structural analysis of the fin, a_1 and a_2 may represent the flow field near the keel and the deformed shape of the fin keel due to structural response and hydrodynamic loads, respectively. The calculation of the flow field a_1 requires the shape of the fin keel, which is contained in a_2, while the calculation of the fin deformation a_2 requires the hydrodynamic loads, contained in a_1. The formulation of a *two-discipline MDO problem* involves the previous definitions but in the framework of an optimization problem. Up to now we were just looking for a multidisciplinary *equilibrium* between the two disciplines. The most natural optimization problem formulation is to impose an optimizer over the MDA Eq. (21). In fact, this approach has been commonplace in engineering for many years. The formulation is depicted as follows:


s, l_1
Disciplinary Analysis Δ_1
a_1
s, l_2
Disciplinary Analysis Δ_2
a_2
a_1
a_2
min $f(s, a_1, a_2)$
s, l_1, l_2
subject to $g_1(s, l_1, a_1) \leq 0$
$g_2(s, l_2, a_2) \leq 0$

where g_1 and g_2 are the disciplinary design constraints. Many different MDO formulations exist that can be built starting from the above formulation. Recent efforts in analysis and development of problem formulations for MDO can be found in the recent, special issue that the journal, *Optimization and Engineering*, dedicated to MDO (see [1] for the complete reference).

Discussion

Dane Hendrix, Member

I congratulate the authors on compiling a remarkable collection of hydrodynamic optimizations. This collection is remarkable not just for the variety of cases addressed but more for the variety of tools used to attack those problems. I am especially interested in learning more about meta-models and intend to follow up on some of the references you provide.

There are a number of questions I could ask about the work presented, but I will restrict myself to just a few more general questions.

(1) Is there a reason that you don't use a series of gradient method optimizations to generate a Pareto front? It seems that for smooth objective functions this would be a more efficient method than using a global optimization method.

(2) Most of the examples you give are point designs. That is, the objective is for a single speed or wave frequency. Do you have any experience that would indicate that the optimization process is better behaved when you use an objective function that addresses a range of conditions? My experience is that this reduces the problems of local minima keeping a descent method from finding a global minima (Percival, 2001).

(3) I find it helpful to think of optimization as consisting of three parts:

- the geometry modification
- the objective function evaluation
- the optimization algorithm

(3a) You have presented one very good example of using different optimization algorithms for the same problem in the hull-form optimization for seakeeping section. Here you use three different optimization algorithms to analyze the same problem. From Tables 1 and 2 it is not clear that they have all converged and the intermediate results presented in Figure 9 suggest that they may not even be converging toward the same minima. In Table 2, FILLDIR appears to have produced a better result than DDFPSO or DIRECT while in Table 1, the opposite is true. Does this say something about the initial efficiency of each of these methods? Would it be instructive to push each of these algorithms to convergence?

(3b) In one case where you investigated two different objective functions for the application to a sailing yacht keel, you found that for the conditions investigated that the initial gradient from your low fidelity model (potential flow) was almost exactly opposite to that given by your high fidelity model (RANS). Is this because of significant separation in the original geometry? Or was the low fidelity potential model ignoring lift? Or was there some other characteristic of the flow that led to this result?

(3c) Have you looked at the effect of different forms of geometry modification on the optimization result? In the HSSL SWATH example you discuss using initial sensitivity studies to down select design variables. Could you offer some more information on how this was done?

(3d) Do you have any opinion on whether increasing the number of design variables for different types of hull parameterizations converge to the same result?

(4) What are the sources of the differences in the 5415 optimizations presented (solver, hull discretization, how RAOs are evaluated, other)?

Again congratulations on an excellent paper and I look forward to your responses.

Percival, S., Hendrix, D., and Noblesse, F. 2001 "Hydrodynamic optimization of ship hull forms," App. Ocean Research, 23, pp. 337-355.

John Kuhn, Member

The authors have made a significant contribution by presenting a broad assortment of techniques that cover many important issues in hydrodynamic optimization. In addition, they have demonstrated that large performance gains are possible if pertinent aspects of a design are able to be changed with sufficient freedom by the optimization process.

Many elements of the paper are worthy of discussion, but in the interest of brevity I will restrict my attention to only one item: seakeeping optimization. The example in the paper based on heave motion of the S175 containership in head seas reminds me of an issue that we confronted recently. Specifically, we have found that it can be important to include a range of headings in work of this sort, and I was wondering if any analysis was done to assess the way in which the heave optimum for head seas performs at other headings, or in other degrees of freedom (i.e., roll or pitch).

In one of our recent studies we minimized roll motion in beam seas. A significant improvement was achieved in beam seas, but the roll became worse at another heading. Although roll may be more sensitive than heave in this regard, the results that we found illustrate a basic issue, so I will summarize some of our findings to motivate discussion. Briefly, we minimized the root mean square (RMS) roll motion of a large displacement ship in sea state 5 beam seas with 20 knots forward speed. The problem contained an assortment of practical design constraints, and was solved with the SHAPE optimization code (Kuhn et al., 2007). All of the seakeeping calculations for the objective function were done with the Large Amplitude Motions Program (LAMP; Lin et al., 1999). The RMS roll was reduced from 6º (for the baseline) to 4.2º (for the optimization), which is a 30% reduction. However, performance at other headings that were not included in the optimization varied in both favorable and unfavorable directions, as shown in Figure 1.

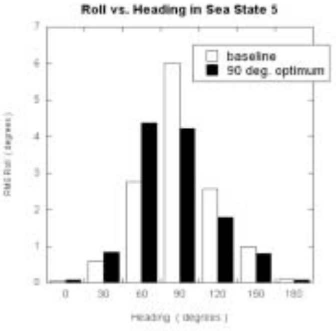


Figure 1 Results for optimum based on roll minimization at 90º heading (0º is following seas, 180º is head seas

Although the roll motion declined significantly for the single heading that was included in the optimization (i.e., 90º), it became considerably more severe at the 60º heading (which was not included in the optimization). In fact, it looks like the most adverse heading for the optimum has shifted to somewhere between 60º and 90º, but we did not examine any intermediate headings in an attempt to find the new peak.

We decided to include multiple operating conditions in the optimization as a consequence of this behavior. One could, in principle, approach this as a multiobjective problem with individual objective functions for every operational condition of interest. However, this could entail a significant matrix of headings, speeds, and sea states. The matrix should also include multiple degrees of freedom (i.e., heave, pitch, and roll). The overall matrix could easily become very large, making the use of Pareto optimality based on *all* such objectives an onerous chore. There could easily be 30 or 40 objectives (or even more).

As an alternative, we have found it effective to use constraints for those conditions that are not handled as an objective. Thus far, this has only been investigated for local optimization in the vicinity of the baseline design, but it has produced some interesting results. For example, it is possible to constrain the roll motion for headings other than beam seas, while minimizing roll in beam seas. This yields the result shown in Figure 2. Note that this figure also contains the result given in the previous figure to facilitate easy comparison.

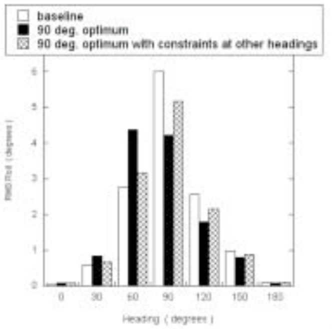


Figure 2 Effect of constraints on roll for headings not included in objective function

As shown in the figure, the roll constraints mitigate growth in roll at the 60º heading, and a 15º reduction in roll is still possible at the 90º heading. The constraints have arbitrarily chosen bounds for this example, but they could be based on standardized criteria for motions, if desired. This is just one example of what can be done. To be safe, however, it would probably be prudent to include constraints on headings that are more closely clustered around the peak than has been done for this example.

Because seakeeping is a vibration problem, sensitivity to operational conditions is not surprising. They determine the excitation of the motion. An optimizer could easily “tune” a design in ways that improve performance for the conditions that are not included in the optimization. This may be most acute for cases with forward speed, because the frequency of excitation shifts with changes in heading. Some degrees of freedom may be more sensitive to this issue than others, and more research is needed to fully understand it.

Dealing with problems like this entails a large amount of computational (and logistical) effort. Your paper already provides valuable guidance in this regard, but any additional comments or observations that you have about practical ways to handle a full permutation of headings, speeds, and degrees of freedom for seakeeping optimization would be most welcome. And, of course, it is also desirable to simultaneously include other design metrics in the optimization process, such as drag or construction producibility. Perhaps the answer involves a combination of Pareto optimality for a manageable number of objective functions based on the most critical metrics, combined with constraints to control behavior for other metrics that are not contained in any other objective functions.

Once again, your paper contains many valuable findings in a complex area. Thank you for making this contribution to our field.

References

Kuhn, J. C., K. L. Chevalier, E. C. Schlageter, S. A. Scragg, and D. C. Wyatt. 2007 The use of linear programming and basis function for hull-form optimization, 9th International Conference on Numerical Ship Hydrodynamics, Ann Arbor, Michigan.

Lin, W. M., S. Zhang, K. Weems, and D. K. P. Yue. 1999 A mixed source formulation for nonlinear ship-motion and wave-load simulations. 7th International Conference on Numerical Ship Hydrodynamics, Nantes, France.

Authors’ Closure

The authors thank Dr. Hendrix for the stimulating discussion.

Question 1

Is there a reason that you don't use a series of gradient method optimizations to generate a Pareto front? It seems that for smooth objective functions this would be a more efficient method than using a global optimization method.

Gradient-based optimization methods are extremely accurate and fast for local, single-objective optimization problems. As Dr. Hendrix suggests in his question, if one needs to solve a multiobjective problem and wants to use a gradient-based algorithm, the different objective functions have to be aggregated into a single merit function *f**:

$$f^*(x) = \sum_{i=1}^{N_f} w_i f_i(x)$$

where N_f is the number of objective functions and w_i are the corresponding weights of the different objectives f_i. With this approach, one can find a single point on the Pareto front (it should be mentioned anyhow that it is not guaranteed that the local minimum identified by the gradient method belongs to the Pareto set). To identify more solutions – belonging to the Pareto set one hopes - one therefore needs to perform repeated gradient-based searches (i) starting from different initial designs and (ii) with different combination weights w_i for the merit function. This is certainly possible and it can be easily done. There are of course also good reasons to develop true global optimization methods for multiobjective problems. To quote an excellent reference on multiobjective methods (Statnikov and Matusov, 1995): “Numerous attempts to construct a generalized criterion in the form of a combination of particular criteria proved to be fruitless. By cramming a multicriteria problem into the Procrustean bed of a single-criterion one, we replace the initial problem with a different one that has little in common with the original problem.” We have no direct experience with the procedure suggested in the question. However, in our experience, multistart gradient-based methods for single-objective problems are much less accurate than global optimization methods. An example is given in the following picture, which refers to the solution of a single optimization problem, namely the design optimization problem for the Ship S175. The objective function *F* was the RAOs peak value for the heave motion in head seas (at 16 knots) and six design variables were used for the solution. Geometrical constraints were imposed on the minimum and maximum beam and on the ship displacement, and a simple strip-theory code was used as analysis tool for computing *F*. In Figure 1 the solutions obtained with a number of different algorithms - local and global - are presented. The RAO's peak is reduced by using all the optimization algorithms: among *local* solutions, the one obtained with a multistart gradient method is arguably the best one. It is, however, clear that all the global methods tested are superior to the multistart.

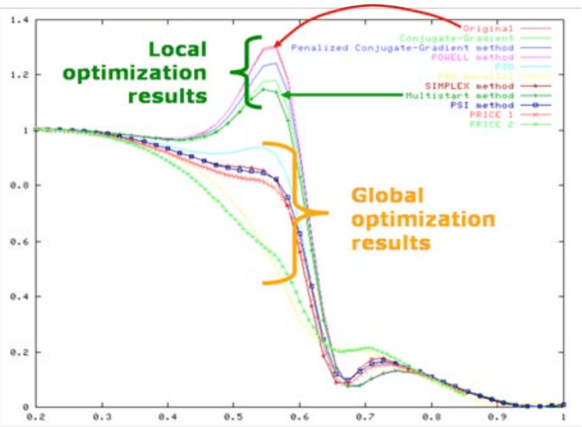


Figure 1 Heave RAO in head seas at 16 knots: local optimization procedures are able to improve the original design, whereas global optimization techniques are successful in finding a much better design!

Question 2

Most of the examples you give are point designs. That is, the objective is for a single speed or wave frequency. Do you have any experience that would indicate that the optimization process is better behaved when you use an objective function that addresses a range of conditions? My experience is that this reduces the problems of local minima keeping a descent method from finding a global minima.

As far as we understand, in principle there is no reason for which an optimization process should be better behaved when using a multipoint objective function, addressing a range of conditions. Conversely, we would agree on that a multipoint *solution* is more *robust*, that is the performances of a single point design tend to drop in off-design conditions. Indeed, techniques for robust optimization, of the type described in the paper, address exactly this problem.

Question 3a

You have presented one very good example of using different optimization algorithms for the same problem in the hull-form optimization for seakeeping section. Here you use three different optimization algorithms to analyze the same problem. From Tables 1 and 2 is it not clear that they have all converged and the intermediate results presented in Figure 9 suggest that they may not even be converging toward the same minima. In Table 2, FILLDIR appears to have produced a better result than DDFPSO or DIRECT while in Table 1, the opposite is true. Does this say something about the initial efficiency of each of these methods? Would it be instructive to push each of these algorithms to convergence?

In Tables 1 and 2 we used a fixed number of objective function evaluations to stop the algorithms: 100 (1000) times the number of design variables, i.e., 600 (6000) objective function evaluations, respectively. What is clear from Table 1 is that DDFPSO and DIRECT show close performances when one has only a reduced number of objective function evaluations available; indeed, they obtain nearly the same result in terms of design variable values and objective function reduction. FILLDIR is faster but converges to a slightly less attractive solution (anyhow the loss in ship performance with respect to the other two algorithms is only about 3.5%). When more objective function evaluations are available we got a reversed situation: DDPFSO is the best: the solution is almost coincident with the other two but the convergence is faster. Anyhow it is clear that different global methods can converge to different results; indeed, there is no mathematical proof that, for non-convex problems, one has reached the absolute global minimum.

Question 3b

In one case where you investigated two different objective functions for the application to a sailing yacht keel, you found that for the conditions investigated that the initial gradient from your low fidelity model (potential flow) was almost exactly opposite to that given by your high fidelity model (RANS). Is this because of significant separation in the original geometry? Or was the low fidelity potential *model ignoring lift? Or was there some other characteristic of the flow that led to this result?*

The example reported has been designed for illustrating the qualities of the Variable Fidelity Modeling, and was selected as a good example of the differences between two different models. The low fidelity potential model includes straight trailing vortices living the fin keel at an undetermined angle (Bollay W., 1936). In the example, however, the bulb geometry of the yacht keel is changing, and the Laplace solver is not able to produce accurate results in these conditions, because the frictional resistance is brutally estimated by considering the thin plate resistance, while the flow details are much more complex and have a great influence on the drag. We believe that this situation is relatively common for flows around bodies with an angle of attack.

Question 3c

Have you looked at the effect of different forms of geometry modification on the optimization result? In the HSSL SWATH example you discuss using initial sensitivity studies to down select design variables. Could you offer some more information on how this was done?

We have not performed yet any study on the effect of different shape modification strategies, but we fully agree with Dr. Hendrix that it would be of great interest. About the HSSL SWATH question, we may say that in general, sensitivity studies can be useful in determining if a design variable is really affecting the objective function or not. The approach is based on the computation of the partial derivative of the objective function with respect to the variable: if it is small, this variable could be excluded from the optimization. The problem is that the gradient is computed around a given solution, and the gradient component might be completely different in a different location of the design space. The use of surrogate models of the objective function can overcome this difficulty, but the effort in computing the surrogate model is not negligible. So, a common approach is based on the evaluation of the gradient at the original design.

Question 3d

Do you have any opinion on whether increasing the number of design variables for different types of hull parameterizations converge to the same result?

Different numbers of variables give different freedom to the optimizer. In our experience (Kim et al., 2008) any algorithm shows better results when the number of variables increases.

Question 4

What are the sources of the differences in the 5415 optimizations presented (solver, hull discretization, how RAOs are evaluated, other)?

The shapes reported in the paragraph "Single-Objective Application: DTMB Model 5415 Optimization" are obtained with two completely different frameworks. SDB-A uses (1) a genetic algorithm for the optimization, (2) CFDShip-Iowa as RANSE solver and (3) a CAD-based approach for the parametrization (that is, a CAD system is connected with the optimizer, and it is used for the deformation of the original hull). Conversely, SBD-B uses (1) a VFM approach, using two different grid levels as high and low fidelity, (2) MGShip as RANSE solver (the in-house INSEAN RANS solver developed by Andrea Di Mascio), and (3) the parametrization is obtained by Béziér patches superimposition. The seakeeping solver is the same, but it was used for the computation of constraints only. So, all the three constitutive SBD elements are different. Nevertheless, the two final solutions show similar geometrical trends.

Reply to Dr. Kuhn Questions

The authors thank Dr. Kuhn for the stimulating discussion. We think that the problem raised by the question of Dr. Kuhn can be addressed under the general framework of robust design optimization (RDO) methods. RDO methods are developed to prevent the effects of uncertainties. The effects of considering uncertainty consist in (i) a loss in specialization of the system and (ii) a gain in robustness (i.e., in the expectation and/or variance of the performances against the variation of the probabilistic parameters). Robust design can be formulated (as described in the paper) as an optimization problem by considering the Bayes principle and replacing the objective function f with a more complex function $\phi(d)$ which includes a probability density function.

A possible solution is to take into account headings and sea states by assuming their probability density p (Figure 2) whereas motions for given speeds can be retained as objectives. In this way one can obtain a reduction of the number of objectives.

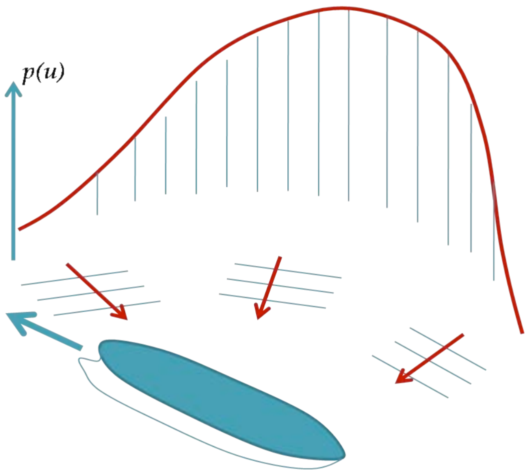


Figure 2 A sketch of the problem of with multiple headings and sea states

The integration of the objective function is, however, expensive. To reduce the computational burden, one can choose a number of (sea) states and evaluate the objective at each state, approximating the integral by some quadrature formula. The exact Kernel of the integral can also be replaced by an approximation obtained by interpolating values of the Kernel at some given state.

References

Bollay, W. 1936 A new theory for wings of small aspect ratio, PhD thesis, California Institute of Technology.

Kim, H. J., Chun, H. H., Peri, D., and Campana, E. F. Optimizing using Parametric Modification Functions and Global Optimization Methods, 27th ONR Symp. on Naval Hydrodynamics, Seul (Korea), 2008.

Statnikov, R. B. and Matusov, J. H. 1995 Multicriteria Optimization and Engineering, Chapman and Hall, New York.