

---

# Penalty Function approaches for Ship Multidisciplinary Design Optimization (MDO)

---

**Abstract:** This paper focuses on the solution of difficult Multidisciplinary Optimization formulations arising in ship design. The latter schemes are by nature the result of the interaction among several optimization problems. Each optimization problem summarizes the issues related to a specific aspect (discipline) of the formulation, and it may be hardly solved by stand-alone methods which ignore the other disciplines. This usually yields very challenging numerical optimization problems, due to the *simultaneous* solution of different schemes.

In particular, in our ship design applications we stress the strong interaction between fluid-dynamics and optimization, in order to get remarkable achievements. The ordinary stand-alone methods from mathematical programming prove to be often unsatisfactory on the latter multidisciplinary problems. This scenario requires a specific integration of both Fluid-dynamics and Optimization, where constrained optimization schemes frequently arise. We give evidence that the proper use of Penalty Methods, combined with Global Optimization techniques, may both be a theoretically correct approach, and may yield a fruitful class of techniques for the solution of Multidisciplinary problems.

We provide numerical results with different penalty functions, over difficult multidisciplinary formulations from ship design. Here, the introduction of penalty methods proved to be a valuable tool since feasibility issues strongly affect the formulation.

**Keywords:** Multidisciplinary Design Optimization; Fluid-Dynamics; Constrained Optimization; Penalty methods; Global Optimization.

---

## 1 Introduction

Efficient solvers for Computational Fluid-Dynamics (CFD) play a key-role in ship design problems, where both the number and length of experiments is a crucial issue PeRoCa01, 2001. In addition, CFD analysis has been recently used in a larger context, where different CFD solvers are combined in an optimization framework. Indeed, traditional approaches to shape design often required only feasibility, rather than focusing on the solution of a full optimization problem. As a consequence, in case different disciplines were involved in the design problem, heuristic approaches were applied to achieve each individual disciplinary feasibility AlHu97, 1997; Ra02, 2002. However, the application of heuristics risks to be unsatisfactory, since it may be based more on human skills and experience, rather than on quantitative methods.

The latter facts motivate our interest for the systematic approach to numerical MDO. In our problems the *multidisciplinarity* refers to the design of a ship, and includes physical phenomena where hydrodynamics is coupled with structural me-

chanics, and control. Furthermore, the coupling among disciplines might be essential, in order to provide the convergence properties of the numerical techniques adopted. Thus, typical issues from nonlinear programming as feasibility, optimality conditions, sensitivity analysis, duality theory, require a suitable adaptation in our MDO framework.

In particular we want to solve an MDO formulation of a sailing yacht keel fin design problem, where the derivatives of the objective function and constraints are unavailable. This problem is a hydroelastic design optimization problem for an *America's Cup* race yacht. The fin is used to hold the bulb, so that yacht stability increases during a competition.

The keel fin design is both affected by the weight and the shape of the bulb (*structure*), as well as the hydrodynamic forces arising during the competitions (*hydrodynamics*). The interaction between the two disciplines yields an optimization formulation, which is solved with a combined approach including Penalty Methods and derivative-free techniques. We remark that, as in several MDO problems, the derivatives are unavailable, since the functions involved are essentially obtained by solving coupled PDE systems, by black-box tools. As a result, automatic differentiation or adjoint methods may not be adopted. Moreover, each function evaluation may be really time consuming, requiring up to one day of computation on a parallel machine.

In particular, we apply an approach which encompasses the use of suitable penalty methods and global optimization algorithms. The paper is organized as follows. Section 2 describes some generalities of MDO. Then, in Section 3 we detail our keel fin design problem. Section 4 addresses, with specific emphasis, the use of suitable algorithms from nonlinear programming, in order to solve our MDO formulation. Finally, Section 6 provides a numerical experience.

## 2 Generalities on MDO

In this section we introduce a formal description of a general MDO problem, which includes several approaches from the literature AlHu97, 1997; AlLe09, 1999; AlLe00b, 2000; AlLe00a, 2000. The section details the optimization framework and the difficulties related to our fin design problem.

Let us consider the  $r$  disciplines  $D_1, \dots, D_r$ , along with the vectors of design variables  $x_i \in \mathbb{R}^{n_i}$  and state variables  $s_i \in \mathbb{R}^{m_i}$ , associated with the discipline  $D_i$ . With the latter positions, and introducing also the vector  $x_0 \in \mathbb{R}^{n_0}$  (design variables shared by the  $r$  disciplines), we want to model a general MDO ship design problem. On this purpose, let us consider the quantities

$$x = (x_0^T \ x_1^T \ \dots \ x_r^T)^T \in \mathbb{R}^n, \quad n = n_0 + n_1 + \dots + n_r,$$

$$s = (s_1^T \ \dots \ s_r^T)^T \in \mathbb{R}^m, \quad m = m_1 + \dots + m_r,$$

and the following position (see also AlHu97, 1997; AlLe09, 1999; AlLe00a, 2000).

**Assumption 2.1.** *Suppose the MDO problem involves the  $r$  disciplines  $D_i$ ,  $i = 1, \dots, r$ . Consider the vectors  $x \in \mathbb{R}^n$  and  $s \in \mathbb{R}^m$ ; assume that for each discipline  $D_i$ :*

1. we can describe the feasible set  $B_i \subseteq \mathbb{R}^{n_0 \times n_i \times m}$  (possibly empty), through equalities and inequalities, i.e.

$$B_i = \{(x_0, x_i, s), x_0 \in \mathbb{R}^{n_0}, x_i \in \mathbb{R}^{n_i}, s \in \mathbb{R}^m : \\ g_i(x_0, x_i, s) \geq 0, A_i(x_0, x_i, s) = 0\},$$

where  $g_i : \mathbb{R}^{n_0} \times \mathbb{R}^{n_i} \times \mathbb{R}^m \rightarrow \mathbb{R}^{\text{in}_i}$ , with  $\text{in}_i \geq 1$ , and  $A_i : \mathbb{R}^{n_0} \times \mathbb{R}^{n_i} \times \mathbb{R}^m \rightarrow \mathbb{R}^{\text{eq}_i}$ , with  $\text{eq}_i \geq 1$ ;

2. there exists the nonlinear function  $f_i(x_0, x_i, s)$ , with  $f_i : \mathbb{R}^{n_0} \times \mathbb{R}^{n_i} \times \mathbb{R}^m \rightarrow \mathbb{R}$ , such that the formulation associated with the discipline  $D_i$  is

$$\min_{(x_0, x_i, s) \in B_i} f_i(x_0, x_i, s);$$

3. there exist the set  $B = B(x, s) = \{(x, s), x \in \mathbb{R}^n, s \in \mathbb{R}^m : g_0(x, s) \geq 0, (x_0, x_i, s) \in B_i, i = 1, \dots, r\}$ , and the functions  $f(x, s) = \varphi[f_1(x_0, x_1, s), \dots, f_r(x_0, x_r, s)]$  and  $g_0(x, s)$  (either explicitly or implicitly defined), with  $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$  and  $g_0 : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^{\text{in}_0}$ , with  $\text{in}_0 \geq 1$ , such that the ship design problem becomes

$$(1) \quad \min_{(x, s) \in B} f(x, s).$$

**Definition 2.1.** Let the Assumption 2.1 hold; then, we say that (1) is a **nonlinear MDO formulation** for the MDO problem.

In our ship design MDO problems the Assumption 2.1 holds; moreover, box constraints on a subset of the design variables may be also included. The latter constraints may be included in the block of inequalities  $g_0(x, s) \geq 0$ . The formulation (1) is aimed to distinguish between *tractable* MDO problems (for which we say that at least a nonlinear MDO formulation exists), and *intractable* MDO problems (whose formulation is hardly described, either explicitly or implicitly, by a nonlinear programming formulation).

We observe that though (1) is formally a nonlinear program, standard techniques from numerical optimization are not immediately applicable for its solution. Indeed, the main difficulty of solving (1) is often the fact that  $B$  includes the so called *MultiDisciplinary Analysis (MDA)*

$$MDA = \begin{cases} A_1(x_0, x_1, s) = 0 \\ \vdots \\ A_r(x_0, x_r, s) = 0. \end{cases}$$

Apparently  $MDA$  is a system of nonlinear equations. However, in practice the  $i$ -th block of nonlinear equalities  $A_i(x_0, x_i, s) = 0$  is often a black-box, which only implicitly defines a map among  $x_0, x_i$  and  $s$ . Moreover, in ship design problems the block of nonlinear equalities  $A_i(x_0, x_i, s) = 0$  frequently corresponds to a discretized PDE system. Thus, the *implicit function theorem* cannot be exploited to retrieve  $s = s(x)$  from  $MDA$ , in order to solve (1) as

$$(2) \quad \min_{x \in B(x, s(x))} f(x, s(x)).$$

As a consequence, (1) may be hardly solved as a nonlinear program all at once. Moreover, optimization techniques based on Karush-Kuhn-Tucker (KKT) conditions for (1) may be of difficult application, since *constraint qualification conditions* often do not hold. This proves the intrinsic difficulty to develop complete convergence analysis along with effective algorithms for our ship design problem, formulated as (1).

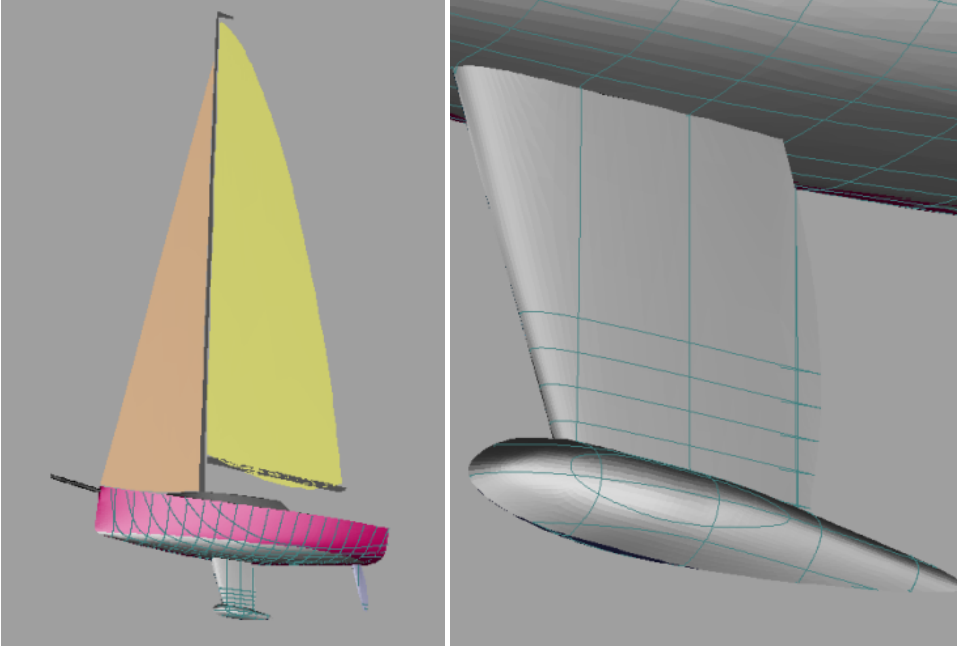
### 3 Our keel fin design problem

The MDO application we selected to apply our approach is the optimization of a keel fin of a race sailing yacht. A sailing yacht is a ship which travels using the wind power only. As a consequence, we have not a complete control of our propulsive system, since the speed and direction of the wind are unpredictable to a large extent. Thus, the yacht often travels in a different direction than the wind, and this causes lateral forces acting on the ship (see also PeMa05, 2005). This generates the following two effects: the lateral forces tend both to roll the ship along the longitudinal axis, and to shift it along the side direction. In order not to capsize, an heavy body (usually called *bulb*, due to its streamlined shape) is placed in depth: it is particularly heavy, since it is used to contrast the heeling moment generated by the wind. The bulb is connected with the yacht by a thin fin, usually called *keel fin*. When the wind comes from one side, the yacht tends to roll around the longitudinal axis, as soon as an equilibrium position between the heeling moment generated by the wind and the moment caused by the weight of the bulb is obtained. As a consequence, the keel fin is subject to a strong bending moment due to the large weight of the bulb. On the other hand, the keel fin also provides the lateral forces contrasting the lateral forces produced by the wind: for this purpose, it is shaped as a wing. A second equilibrium condition is required for these forces. The shape of the keel fin is important because we need to obtain an high side force with a moderate induced resistance. The shape of the keel is dynamically influenced by the system of forces acting on it, and the hydrodynamic characteristics as well. This is an example in which the solution of a multidisciplinary design optimization problem is claimed. In Figure 1 a perspective view of a sailing yacht with a keel fin and bulb is presented.

Since our objective is the design of a keel fin for a race sailing yacht, the performances of the yacht are our primary goal. What we need to design is a keel fin which is able to induce a side force, with a minimum expense in terms of resistance. In fact, when a wing generates a lift force, this is payed by an induced drag. A correct shaping of the keel fin is expected to reduce the induced drag without losing lifting force. We want to solve the formulation (1), where our objective function  $F$  is the cost of side force in terms of induced drag, say the ratio

$$(3) \quad F = \frac{F_{side}}{F_{forw}}.$$

In (3) ‘*side*’ indicates the lateral direction and ‘*forw*’ the advancing direction. The overall formulation will be detailed more widely in Section 6.



**Figure 1** Fisheye view of a typical sailing yacht with keel fin and bulb: global perspective view and detail view.

#### 4 NonLinear Programming (NLP) for ship design MDO

In this section we consider some generalities for the solution of the following constrained optimization problem

$$(4) \quad \min_{(x,s) \in B} f(x, s),$$

which arises in several MDO formulations of ship design, as described in Section 2. The *feasible set*  $B \subseteq \mathbb{R}^{n \times m}$  is defined throughout equalities and inequalities, i.e.

$$B : \begin{cases} h_j(x, s) = 0, & j = 1, \dots, p, \\ g_i(x, s) \leq 0, & i = 1, \dots, m. \end{cases}$$

Observe that (4) has a very general structure, where no specific assumption is considered, e.g. any convexity/concavity assumption on  $f(x, s)$ , or the feasible set  $B$ . In particular, we also assume that the functions  $f(x, s)$ ,  $h_j(x, s) = 0$ ,  $j = 1, \dots, p$ ,  $g_i(x, s) \leq 0$ ,  $i = 1, \dots, m$ , are *continuously differentiable* over the set  $B$  (though the derivatives are unavailable).

Further information on the feasible set  $B$ , as the convexity, allows for the use of specific and efficient methods for the solution of problem (4) (e.g., *feasible direction methods* BaShSh93, 1993; Be95, 1995). In the latter case, if  $f(x, s)$  is also convex over  $B$ , then an interesting amenity holds: every local minimum of  $f(x, s)$  over  $B$  is also a global minimum of  $f(x, s)$  over  $B$ . This would simplify the search of global

minima over  $X$ , since most of the gradient-based algorithms for local optimization could be fruitfully used.

In principle the solution of (4) requires the simultaneous solution of two separate problems: the minimization of  $f(x, s)$  and the feasibility of the solution found (if any exists).

Excluding the use of specific methods to solve (4), *general approaches* adopted in the literature of numerical optimization may be essentially reduced to the following categories (see Be95, 1995; NoWr06, 2006; FiMc68, 1968; Be82, 1982):

- Methods which solve a *sequence of constrained subproblems*, each of which must be simpler than (4). In this class we find the well known SQP (*Sequential Quadratic Programming*) methods, which basically represent the natural extension of Newton's method to the constrained problem (4).
- Methods which solve a *sequence*  $\mathcal{K}$  (possibly containing just one element) of *unconstrained subproblems*. Under specific assumptions, the sequence  $\{x_k^*\}_{\mathcal{K}}$ , where  $x_k^*$  solves the  $k$ -th unconstrained subproblem, converges to a solution  $x^*$  of (4).

In the latter class of methods we include the following subclasses:

1. **Penalty Methods**, where at once the sum of  $f(x)$  and a penalty term measuring the infeasibility (constraints violation), is minimized. Thus, the minimization of  $f(x)$  and the feasibility of the solution are simultaneously sought. The penalty term always depends on a *parameter*  $\mu \in \mathbb{R}$ , which is often crucial for the efficiency of the overall method. Depending on the nature of the penalty function, we distinguish between two different approaches.
  - **Exact Penalty Methods**: the sequence of subproblems for (4) includes a *unique* unconstrained subproblem  $UN$ . There exists a value  $\bar{\mu}$  for the parameter, such that if  $\mu \in (0, \bar{\mu}]$ , then a solution  $x^*$  of  $UN$  is also a solution of (4).
  - **(Sequential) Penalty Methods**: we have an infinite sequence  $\{UN_k\}$  of subproblems, where the subproblem  $UN_k$  corresponds to a different value  $\mu_k$  of the parameter  $\mu$ . Convergence may be proved when  $\mu_k \rightarrow 0$ .
2. **Barrier Methods**: here a sequence  $\{x_k^*\}$  of points in the interior of  $B$  is generated. Under suitable assumptions  $\{x_k^*\} \rightarrow x^*$ , where  $x^*$  is a solution of (4).
3. **Augmented Lagrangian Methods**: in this class of algorithms, which may be both *Exact* and *Sequential*, the idea of a penalty function is essentially combined with the Lagrangian function associated with (4).

The next section will be devoted to describe and motivate the use of a proper algorithm to solve (4), among the approaches described in this section.

#### 4.1 The proper NLP method for our ship design problem

Consider our ship design problem (1) in Section 3. A careful modelling of that problem yields a formulation as in (4), where in particular we obtain for the feasible set  $B$  the modified expression<sup>a</sup>

$$(5) \quad \bar{B} : \begin{cases} h_H(x, \bar{s}_x) = 0, \\ h_S(x, \bar{s}_x) = 0, \\ g_i(x) \leq 0, \end{cases} \quad i = 1, \dots, m.$$

The set of equality constraints  $h_H(x, s) = 0$  is obtained by discretization of a PDE system, associated with the hydrodynamic properties of the fin. Similarly, the set of equality constraints  $h_S(x, s) = 0$  is obtained by discretization of another PDE system, associated with the structural properties of the fin. In the constraints  $h_H(x, s) = 0$  and  $h_S(x, s) = 0$  we distinguish between the *design variables*  $x$  (which affect the optimization) and the *state variables*  $s$  (which are shared just by the equalities of the hydrodynamic and the structural blocks). Moreover, with  $\bar{s}_x$  we indicate a specific value of the state variables, depending on ‘ $x$ ’ obtained after coupling the *structural* and *hydrodynamic* disciplines, as detailed in Section 2. Our NLP formulation of the MDO problem becomes

$$(6) \quad \min_{x \in \bar{B}} f(x, \bar{s}_x)$$

with

$$\bar{B} : \begin{cases} \bar{h}(x, \bar{s}_x) = 0 \\ g_i(x) \leq 0, \quad i = 1, \dots, m, \end{cases} \quad \bar{h}(x, \bar{s}_x) = 0 : \begin{cases} h_H(x, \bar{s}_x) = 0, \\ h_S(x, \bar{s}_x) = 0, \end{cases}$$

and the optimization is performed only with respect to the design variables  $x$ . Now we motivate the choice of the NLP techniques to be used for solving (6), according with the description in Section 4. In particular, we describe both advantages and drawbacks we met for each class of optimization methods listed in the previous section. The latter analysis will motivate and support our choice for the *penalty methods*.

We remark that since the source codes of PDE solvers are usually black-box solvers, Automatic Differentiation cannot be adopted for computing derivatives (see also Gr00, 2000).

##### 4.1.1 SQP Methods

*SQP Methods* are iterative methods introduced since '63 (Wi63, 1963), and widely studied in the '70s and '80s (Po78, 1978; GPMa76, 1976; PoYu86, 1986). They

---

<sup>a</sup>With a little abuse of symbols we use here the same letters  $m$  and  $n$  in (1), but possibly with different meanings.

solve (6) by generating the sequence of iterates  $\{x_k\}$ . At the  $k$ -th iteration they compute the direction  $d_k$  which solves the (simpler) quadratic constrained problem

$$(7) \quad \min_d \frac{1}{2} d^T \nabla_{xx}^2 \mathcal{L}(x_k, \bar{s}_{x_k}, \lambda_k, \sigma_k) d + \nabla_x f(x_k, \bar{s}_{x_k})^T d$$

$$W_x(x_k, \bar{s}_{x_k})d + \bar{h}(x_k, \bar{s}_{x_k}) = 0$$

$$Z_x(x_k, \bar{s}_{x_k})d + g(x_k) \leq 0,$$

where  $\mathcal{L}(x_k, \bar{s}_{x_k}, \lambda_k, \sigma_k)$  is the *Lagrangian function*<sup>b</sup>

$$\mathcal{L}(x_k, \bar{s}_{x_k}, \lambda_k, \sigma_k) = f(x_k, \bar{s}_{x_k}) + \lambda_k^T \bar{h}(x_k, \bar{s}_{x_k}) + \sigma_k^T g(x_k),$$

of the problem (6),  $\lambda_k \in \mathbb{R}^p$  is the vector of multipliers associated with the equality constraints  $\bar{h}(x, \bar{s}_{x_k}) = 0$ ,  $\sigma_k \in \mathbb{R}^m$  is the vector of multipliers associated with the inequality constraints  $g(x) \leq 0$ , and  $W_x(x_k, \bar{s}_{x_k})$ ,  $Z_x(x_k, \bar{s}_{x_k})$  are the Jacobians of the constraints at  $x_k$ . Then, the direction  $d_k$  is used to compute the new iterate  $x_{k+1} = x_k + d_k$ . The main advantage of SQP methods is undoubtedly their appealing rate of convergence, which is under mild assumptions *quadratic* or at least *superlinear* for the sequence  $\{x_k\}$  (see NoWr06, 2006). On the other hand, several disadvantages discourage us from the application of these methods.

First, the derivatives of both  $f(x, \bar{s}_x)$  and the constraints are required in (7), and since we compute  $f(x, \bar{s}_x)$  along with the constraints by simulations, derivatives are in general unavailable. Even when we tried to compute them by *finite differences*, we faced other two problems:

1. in case  $f(x, \bar{s}_x)$ ,  $\bar{h}_j(x, \bar{s}_x)$  and  $g_i(x, \bar{s}_x)$  are noisy functions (the usual case), finite differences suffered from serious inaccuracy and instability in the computation;
2. the quadratic subproblem (7) may be infeasible or even unbounded from below. This represents the main serious drawback of SQP, and may require the introduction of additional trust region constraints to formulation (7).

#### 4.1.2 Barrier Methods

Barrier methods, which are also addressed as Interior Point methods, are parameter dependent techniques which generate a sequence of iterates  $\{x_k\}$  in the interior of the feasible set  $\bar{B}$ . These methods are usually attractive for their simplicity and easy implementation, along with their efficiency (*primal-dual* methods with the predictor-corrector variant by Mehrotra NoWr06, 2006). However, apart from possible instabilities which may arise when convergence comes up, these methods require that the interior of  $\bar{B}$  is non-empty. Moreover, at least the starting feasible point  $x_0$  should be provided by the user, which is often very difficult to generate.

---

<sup>b</sup>Observe that the Hessian matrix  $\nabla_{xx}^2 \mathcal{L}(x_k, \bar{s}_{x_k}, \lambda_k, \sigma_k)$  is often computed by Quasi-Newton approximations, in order to reduce the computational effort. This leads to a wide range of SQP methods.



#### 4.1.3 (Sequential) Penalty Functions

Penalty Functions were introduced for the first time by Courant (Co43, 1943), then extended by Fiacco & McCormick (FiMc68, 1968). Given the problem (6), they consider the solution of the unconstrained problem

$$(8) \quad \min_{x \in \mathbb{R}^n} P(x, \bar{s}_x; \varepsilon), \quad \varepsilon > 0,$$

where

$$P(x, \bar{s}_x; \varepsilon) = f(x, \bar{s}_x) + \frac{1}{\varepsilon} p(x), \quad p(x) : \begin{cases} = 0, & \forall x \in \bar{B}, \\ > 0, & \forall x \notin \bar{B}. \end{cases}$$

$P(x, \bar{s}_x; \varepsilon)$  is usually addressed as a Penalty Function, and the term  $p(x)$  substantially measures the violation of the constraints by the vector  $x$ . Moreover, when both  $f(x, \bar{s}_x)$  and  $p(x)$  are *convex* over  $\mathbb{R}^n$ , then  $P(x, \bar{s}_x; \varepsilon)$  is also convex, so that its local minima coincide with its global minima.

To solve (6) we iteratively considered in (8) the sequence  $\{\varepsilon_k\}$  of the parameter  $\varepsilon$ , with  $\varepsilon_k \rightarrow 0$ ; then, for any  $\varepsilon_k$  the solution  $x_k$  of (8) was computed. The function  $P(x, \bar{s}_x; \varepsilon)$  is also known as an Exterior Penalty Function for (6), since its minimizer  $\hat{x}$  is often infeasible for (6), i.e.  $\hat{x} \notin \bar{B}$ . Among the simplest penalty functions proposed in the literature, to solve our ship design problems we have considered the *Quadratic Penalty Function* for (6):

$$(9) \quad P(x, \bar{s}_x; \varepsilon) = f(x, \bar{s}_x) + \frac{1}{\varepsilon} \{ \|\bar{h}(x, \bar{s}_x)\|^2 + \|\max\{0, g(x)\}\|^2 \},$$

where

$$(10) \quad \max\{0, g(x)\} \doteq \begin{pmatrix} \max\{0, g_1(x)\} \\ \vdots \\ \max\{0, g_m(x)\} \end{pmatrix} \in \mathbb{R}^m,$$

and  $\|\cdot\|$  is the Euclidean norm. We sketch in Table 4.1.3 the algorithm we used to implement the Quadratic Penalty Function approach. As regards the scheme in Table 4.1.3, the following comments apply:

- in the scheme **Alg\_Quadratic\_Penalty** we test if the Karush-Kuhn-Tucker (KKT) conditions are satisfied for (6), at the current minimizer  $y_k$  of  $P(x, \bar{s}_x; \varepsilon_k)$ . However, since the Lagrangian function for (6) is hardly available in our ship design problems, in the practical implementation of scheme **Alg\_Quadratic\_Penalty** we stopped when either no significant progress was obtained, or a fixed maximum number of iterations was performed.
- the parameter  $\varepsilon_k$  is updated so that if the condition  $\|\nabla_x P(y_k, \bar{s}_x; \varepsilon_k)\| \leq \rho_k$  is hardly met (which usually occurs when illconditioning arises, too), then  $\varepsilon_{k+1}$  is set according with  $\varepsilon_{k+1} \approx \varepsilon_k$ . Otherwise, when the condition  $\|\nabla_x P(y_k, \bar{s}_x; \varepsilon_k)\| \leq \rho_k$  is easily met (i.e. (12) is easily solved), then  $\varepsilon_{k+1} \ll \varepsilon_k$ .
- illconditioning may arise when we try to satisfy  $\|\nabla_x P(y_k, \bar{s}_x; \varepsilon_k)\| \leq \rho_k$  (i.e. when we minimize the penalty function  $P(x, \bar{s}_x; \varepsilon_k)$ ). As a consequence, the

**Table 1** Our algorithm for the Quadratic Penalty Function: `Alg_Quadratic_Penalty`

```

Given  $\{\varepsilon_k\} \rightarrow 0, \varepsilon_{k+1} < \varepsilon_k, \{\rho_k\}, \rho_k > 0, \forall k \geq 1$ .
Choose  $x_0 \in \mathbb{R}^n$ 
FOR  $k = 1, 2, \dots$ 
    start from  $x_{k-1}$  and find
    (12)  $y_k = \operatorname{argmin}_{x \in \mathbb{R}^n} P(x, \bar{s}_x; \varepsilon_k)$ 
    such that  $\|\nabla_x P(y_k, \bar{s}_x; \varepsilon_k)\| \leq \rho_k$ .
    IF ( $y_k$  satisfies KKT for (6)).OR.(a stopping rule holds) THEN
        set  $x_k = y_k$ , STOP.
    ELSE
        choose  $\varepsilon_{k+1} \in (0, \varepsilon_k)$ 
        set  $x_k$  from  $y_k$  (possibly  $x_k = y_k$ )
    ENDIF
END FOR

```

efficiency of methods based on Penalty Functions relies on the proper technique we adopt to minimize  $P(x, \bar{s}_x; \varepsilon_k)$ . These techniques should avoid to use the illconditioned matrix  $\nabla_{xx}^2 P(x, \bar{s}_x; \varepsilon_k)$  (e.g. Newton's method, or even the Gradient method whose progress depends on the ratio  $\lambda_M/\lambda_m$ , where  $\lambda_M$  [ $\lambda_m$ ] is the absolute value of the largest [smallest] eigenvalue of  $\nabla_{xx}^2 P(x, \bar{s}_x; \varepsilon_k)$ ). In our optimization problem (6), first and second order derivatives are unavailable as recalled above. Furthermore, exact derivative-free methods as pattern search, model-based or linesearch-based techniques (see MoWi07, 2007 for a comparison) are definitely too expensive for our problems (we recall that on our problems one function evaluation may take even one day of computation). Thus, Penalty Functions combined with heuristics to solve (12), are to us appealing, since

1. they yield unconstrained formulations (which require simpler algorithms)
  2. they do not suffer from infeasibility (unlike Barrier Methods or SQP)
  3. when  $\varepsilon_k \rightarrow 0$  we can deflate the illconditioning of  $\nabla_{xx}^2 P(x, \bar{s}_x; \varepsilon_k)$  by choosing a derivative-free method to minimize  $P(x, \bar{s}_x; \varepsilon_k)$
- in order to further reduce the effects of illconditioning, in `Alg_Quadratic_Penalty` we 'set  $x_k$  from  $y_k$ ' by means of extrapolation techniques, which generate  $x_k$  from the trajectory described by the iterates  $\{x_0, x_1, \dots, x_{k-1}\}$ .

We decided to mainly focus on Penalty Methods for ship design, also for the following appealing result (see also Fl87, 1987).

**Proposition 4.1.** *Consider the problem (6), with  $f$ ,  $h$  and  $g$  continuous on  $\mathbb{R}^n$ ; consider the quadratic penalty function (9), where  $\varepsilon_k \rightarrow 0$ . Let  $x_k$  be a global minimum of  $P(x, \bar{s}_x; \varepsilon_k)$  on  $\mathbb{R}^n$ . Then, every limit point  $\bar{x}$  of the sequence  $\{x_k\}$*

generated by `Alg_Quadratic_Penalty`, is also a global minimum for the problem (6).

The latter proposition substantially suggests that the global minima of (6), may be detected by iteratively computing the global minima of (9), for  $\varepsilon_k \rightarrow 0$ . To the latter purpose, we used the heuristics *Particle Swarm Optimization (PSO)* for derivative-free problems (see KeEb95, 1995; CaFaPi09, 2009). PSO proved to be a satisfactory compromise between the computational burden and the accuracy to find a global minimum of  $P(x, \bar{s}_x; \varepsilon_k)$ .

#### 4.1.4 Exact Penalty Functions

We have also considered Exact Penalty functions for solving (6), where a unique unconstrained minimization problem was solved.

In the literature the term ‘Exact’ is used to indicate the precise correspondence between the solutions of (6) and the minimizers of the Exact Penalty Function.

The latter correspondence holds as long as the parameter  $\varepsilon$  satisfies  $\varepsilon \in (0, \varepsilon^*]$ , for a suitable (and usually unknown) threshold  $\varepsilon^*$ . As detailed in Section 6 we didn’t experience problems to set this parameter. A wide range of Exact Penalty Functions has been introduced in the literature. We mainly focus on the so called  $\ell_1$  Exact Penalty Function, which is the following *nondifferentiable* penalty function proposed by Zangwill (see Za67, 1967; Fl87, 1987)

$$(13) \quad \Phi(x, \bar{s}_x; \varepsilon) = f(x, \bar{s}_x) + \frac{1}{\varepsilon} \left\{ \|\max\{g(x), 0\}\|_1 + \|\bar{h}(x, \bar{s}_x)\|_1 \right\}.$$

Clearly relation (13) is nondifferentiable because of the norm  $\|\cdot\|_1$ . Thus, the minimization of (13) requires a specific safeguard. Since in our ship design problem (6) the derivatives are unavailable, we again minimized (13) by using Particle Swarm Optimization (see Section 4.1.3). This choice had two remarkable advantages:

- for any  $\varepsilon \in (0, \varepsilon^*]$  PSO was applied to minimize the nondifferentiable function (13), without being affected by possible illconditioning;
- PSO again provided a suitable compromise between the performance (i.e. a satisfactory estimate of the minimizer for (13)) and the computational cost.

Under mild assumptions we can prove, also for Exact Penalty functions, a theoretical result similar to Proposition 4.1.

Due to the nature of our ship design problem, we did not consider also the use of *continuously differentiable* exact penalty functions defined in the literature (see for example Fl70, 1970).

#### 4.1.5 Augmented Lagrangian Functions

Augmented Lagrangian Functions are iterative methods introduced by Powell in 1969, for equality constrained problems (see Po69, 1969). Then, Rockafellar extended the approach to inequality constrained problems (see Ro73, 1973). These methods have some similarities with the Penalty Functions and Exact Penalty Functions, and proved to be even more efficient and less sensible to illconditioning. However, for our purpose at least a couple of motivations discourage from their use: they

introduce an additional set of variables (*dual variables*), one for any constraint in (6); they require the use of a Lagrangian function associated with problem (6), to be minimized, in place of  $f(x, \bar{s}_x)$ . Considering that the derivatives of  $f(x, \bar{s}_x)$ ,  $\bar{h}(x, \bar{s}_x)$  and  $g(x)$  in (6) are unavailable, the latter choice could be a complication with respect to both Sequential and Exact Penalty Functions. Anyway, the use of derivative-free techniques could possibly alleviate the latter drawback. The adoption of Augmented Lagrangian Functions could be an interesting extension for future work.

## 5 Notes on Particle Swarm Optimization (PSO)

Here we provide a brief description of the iterative heuristics Particle Swarm Optimization. It has been recently widely adopted for the approximate solution of global optimization problems (see KeEb95, 1995). It belongs to the class of Evolutionary Methods and was originally inspired by the flight of birds in a flock. The basic idea of PSO (see also BIKeP07, 2007 for a tutorial) is an attempt to model the behavior of flocks of birds when they cooperate to search for food.

In particular, the paradigm of a flying flock may be reformulated as the search of a global minimum of a nonlinear function. For each member (namely a *particle*) of the flock (or a swarm), a pair of vectors is considered: the *position* and the *velocity*. More formally, suppose we apply PSO, where  $M$  particles are included, to solve an unconstrained optimization problem in  $\mathbb{R}^t$ ,  $t \geq 1$ . At the  $k$ -th step of the PSO algorithm we introduce the following quantities for any particle  $j \in \{1, \dots, M\}$ :

- $x_j^k \in \mathbb{R}^t$ , the current position;
- $v_j^k \in \mathbb{R}^t$ , the current velocity;
- $p_j^k \in \mathbb{R}^t$ , with  $p_j^k = \operatorname{argmin}_{1 \leq h \leq k} \{f(x_h^j)\}$  (i.e., the best position visited in the previous steps by the  $j$ -th particle);
- $pbest^k \in \mathbb{R}^t$ , with  $pbest^k = \operatorname{argmin}_{1 \leq h \leq k, 1 \leq \ell \leq M} \{f(x_\ell^h)\}$  (i.e., the best position visited in the previous steps by all the particles).

A standard version of the PSO method is summarized by the following algorithm (see also ClKe02, 2002)

1. Set  $k = 1$ . Set  $\chi > 0$  and  $w \in [w_{\min}, w_{\max}]$ , compute  $f(x_j^k)$ ,  $j = 1, \dots, M$ , and  $pbest^k = +\infty$ .
2. For any  $j = 1, \dots, M$ , if  $f(x_j^k) < p_j^k$  then set  $p_j^k = f(x_j^k)$ . Set  $pbest^k = \operatorname{argmin}_{1 \leq \ell \leq M} \{p_\ell^k\}$ .
3. For any  $j = 1, \dots, M$  update the position and the velocity of the  $j$ -th particle, as

$$v_j^{k+1} = \chi [wv_j^k + c_1 u_1 \otimes (p_j^k - x_j^k) + c_2 u_2 \otimes (pbest^k - x_j^k)]$$

$$x_j^{k+1} = x_j^k + v_j^{k+1}$$

where  $u_1, u_2 \in \mathbb{R}^t$  are uniformly randomly distributed vectors in  $[0, 1]$  (however, we set in our experiments  $u_1 = u_2 = 1$  as detailed in Section 6), and the symbol  $\otimes$  denotes the component-wise product.

4. If a *convergence test* is satisfied then STOP, else  $k = k + 1$  and Goto 2.

For typical values of the constriction coefficient  $\chi$  see also BIKeP07, 2007 and therein references. Roughly speaking, at step  $k$  the  $j$ -th particle updates its position as subject to a pair of attractive directions: the vector heading its previous best position (i.e.  $(p_j^k - x_j^k)$ ) and the vector heading the swarm best position (i.e.  $(pbest^k - x_j^k)$ ).

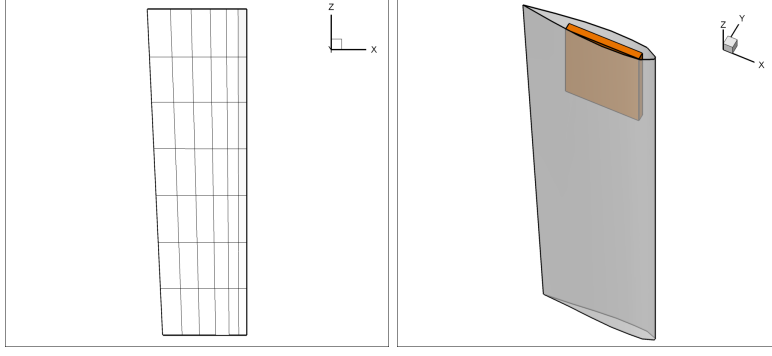
Since PSO is a heuristics it is not endowed with the standard convergence properties provided by *exact Derivative Free Optimization methods* like Pattern Search methods, Polynomial Model-based methods, Linesearch-based methods, etc. However, it often provides a satisfactory approximation of a solution within a very few iterations. In other words, PSO often yields a reasonable compromise between the computational cost and the quality of the final approximate solution found.

## 6 Numerical results and conclusions

As previously recalled, our application is the optimization of the keel fin of a sailing yacht. It is a slender body: at the end of the fin, an heavy body is connected, namely the keel bulb. The keel bulb provides the stability of the yacht. The fin is designed to provide the side force needed to contrast the lateral forces generated by the sails (see Figure 1): this effect is obtained by adequately shaping the fin, producing a non-symmetric pressure distribution on the two sides of the fin. The pressure distribution on the fin is origin of the required side force, but also of a side force and a bending moment on the fin itself, whose geometry is deformed under these actions. Furthermore, the yacht travels in an oblique position, so that the fin bends under the action of the keel bulb plus the pressure acting on the fin itself. As a consequence of the unknown fin deformation, two different solvers are needed for the evaluation of the performances of the fin: a *fluid dynamic* solver and a *structural* solver. With the structural solver, the real geometry of the deformed fin is computed, while the fluid dynamic solver computes the hydrodynamic pressure on the fin, based on the real geometry computed. The final result is obtained by the multidisciplinary interactions of the two different phenomena: the hydrodynamic flow around the fin and the deformation of the fin. These two aspects are mutually related, and this is classically the nature of a multidisciplinary problem.

The state variables (see Section 2) are taken into account in an indirect way. In fact, the pressure and the deformation on each element of the fin, created after a discretization of its surface, represent the state variables  $\bar{s}_x$ . Since the deformed geometry of the fin is a function of the hydrodynamic actions plus the bulb weight, an iterative procedure is used for the derivation of the real deformed geometry: a first hydrodynamic solution is computed, the pressure on the fin is calculated and it is passed to the structural solver as an input. The deformation is computed and the new deformed geometry is analyzed by the fluid dynamic solver. At step  $k$  of this iterative procedure, our objective function (3) will be different than at step  $k - 1$ . We can now define the quantity

$$(14) \quad \eta = \frac{F^k - F^{k-1}}{F^k},$$



**Figure 2** Keel fin geometry: on the left, side view of the adopted keel fin, on the right, visualization of the geometrical constraint.

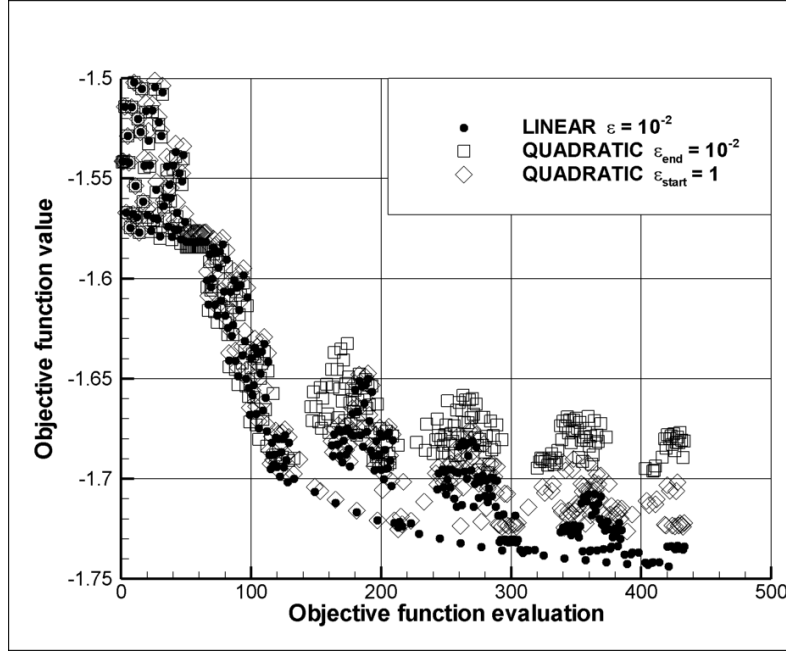
where  $F^k$  is the value of the objective function (3) at step  $k$ , and  $\eta$  is an indicator of the steady-state value of the objective function when performing the MDA. If  $\eta \rightarrow 0$  the iterative procedure of the MDA converges, i.e. the equality constraints in (4) are satisfied. The value of  $\eta$  is therefore an indicator of the degree of coupling between the disciplines. In the current application we set a *satisfactory* value for  $\eta$ , namely  $\bar{\eta}$ , and stop the iterative procedure for the MDA if  $\eta < \bar{\eta}$ .

The fin is substantially represented by a NACA (National Advisory Committee on Aeronautics) profile with a large aspect ratio. A picture is reported in Figure 2: on the left a side view of the fin is presented; on the right a representation of our geometrical constraints is visible (i.e., the inequality constraints in (4)). In order to allow the connection of the fin with the yacht hull, and also in order to include a weak structural constraint (the fin sustains a heavy weight, and the section of the fin cannot be too small), we define a minimum volume to be included into the fin (the parallelepiped in Figure 2).

The parametrization of the fin is obtained by superimposing an analytical surface to the original fin surface. Only one side of the fin is deformed, and then the new geometry is mirrored. The fin can be enlarged/shrunk along the lateral direction, while the bottom section can be shifted forward and backward with respect to the top section of the fin. Four parameters are needed to model this deformation.

The objective function selected for this application is the efficiency of the fin, defined as the ratio between the side force generated by the fin and the total resistance. Since we need a high efficiency of the fin, in this application we solve a maximization problem, or equivalently we minimize the negative value of the efficiency.

To this end, as motivated in Sections 4.1.3 and 4.1.4, PSO algorithm is applied (see KeEb95, 1995; CaFaPi09, 2009), and 16 particles are randomly distributed at the outset of the algorithm. Up to  $100 \cdot \text{NDV}$  evaluations of the objective function are allowed (computational cost), where  $\text{NDV} = 4$  is the number of design variables (i.e., the parameters used to model the fin deformation). Since the four unknowns represent displacements of physical measurements, before applying PSO the objective function is also evaluated in the point  $(0, 0, 0, 0)^T$ . The latter choice provides a reference value for the objective function, corresponding to zero displacements. For  $\eta$  in (14) we set a threshold value of  $\bar{\eta} = 10^{-3}$ . As a consequence, the fulfilment of

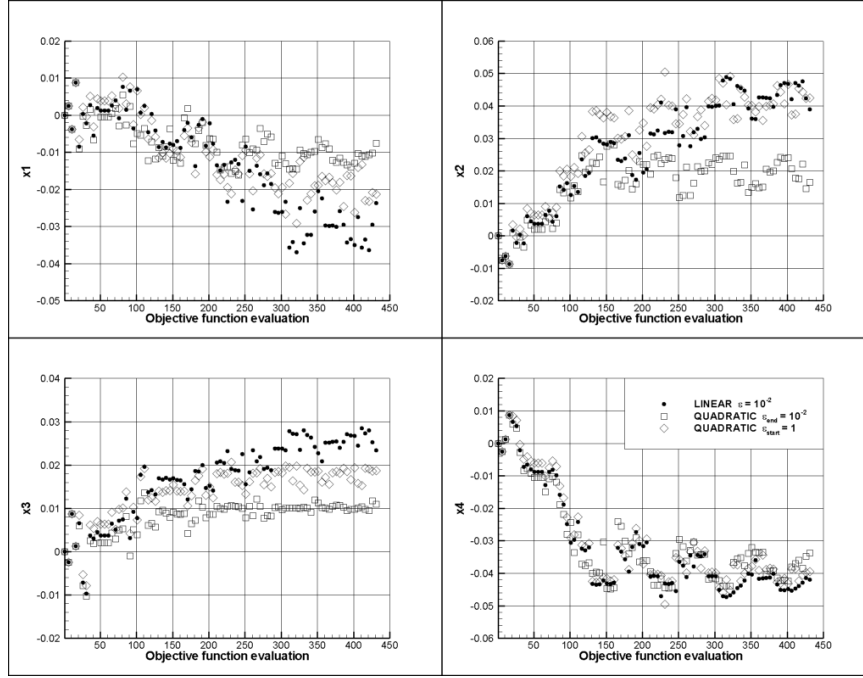


**Figure 3** Comparison among different penalty methods for the minimization of the objective function.

the MDA is achieved along with the convergence of the optimization problem (for further details see also CaFaPePi07, 2007).

Different choices for the penalty function have been experienced on our problem. According with Section 4.1 we chose: the  $\ell_1$  non differentiable penalty function (13) (addressed therein as ‘*linear*’, for short), with a fixed penalty parameter  $\varepsilon = 10^{-2}$ , and a quadratic penalty function (9) with  $\varepsilon_{start} = 1$  and  $\varepsilon \rightarrow 10^{-2}$ . In addition, for the quadratic penalty function we also experienced  $\varepsilon_{start} > 1$  at the outset of the procedure `Alg_Quadratic_Penalty`, then  $\varepsilon \rightarrow 10^{-2}$ , according with the theory which requires  $\varepsilon \rightarrow 0$ . At each step, the penalty parameter for the quadratic penalty function is halved. Results of the comparisons are reported in Figures 3 and 4. The  $\ell_1$  non differentiable penalty function (*bullets*) yields a better value of the objective function, using a fairly large value of the parameter  $\varepsilon$  in (13). We remark that much smaller values of  $\varepsilon$  (say  $\varepsilon < 10^{-6}$ ) may cause a possible numerical instability when minimizing the  $\ell_1$  penalty function in (13). Moreover, the quadratic penalty function starting with  $\varepsilon > 1$  (*squares*) is not amenable, according with the theory. Each symbol in the picture represents a particle of PSO. By observing the areas of Figure 3 where the particles tend to cluster, we immediately deduce that the objective function is likely nonlinear and nonconvex. Indeed, particles clusters are associated with several local minima of the objective function. On the overall, the  $\ell_1$  penalty function provides the best approximation of a global minimum, using the same number of function evaluations.

Observe that from Figure 4 the three penalty functions considered provide very different final values for the 4 design variables, i.e. very different geometries of the



**Figure 4** Final values of the 4 design variables (i.e., the 4 parameters used to model the deformation of the fin), adopting different penalty functions.

final fin. This reveals that unlike heuristic methods may suggest, *feasible* geometries of the fin may be very different from the *optimal* geometries computed by *ad hoc* nonlinear programming approaches. Each symbol in the picture again represents a particle of PSO, and all the final geometries in Figure 4 are feasible.

For the PSO parameters we adopted the values  $c_1 = 0.4$ ,  $c_2 = 0.8$ ,  $w = 0.9$ ,  $\chi = 1$ . Moreover, we set  $u_1 = u_2 = 1$  (see also CaLiLuPePi09, 2009) and unlike the common literature we did not consider  $u_1, u_2$  uniformly randomly chosen in  $[0, 1]$ . The latter partial modification of a PSO scheme has a twofold motivation. On one hand we wanted to adopt PSO since it is relatively fast to locate a satisfactory approximation of a global minimum. On the other hand, the choice of random coefficients for  $u_1$  and  $u_2$  would have required several reruns, in order to provide a significant statistics of the numerical results. Since at the iterate  $x_j^k$  the computation of  $f(x_j^k)$  may take several hours, on our problem setting  $u_1, u_2$  uniformly in  $[0, 1]$  was not allowed in practice (even using a simplified computation for  $f(x_j^k)$ ).

We remark that other papers consider the implicit use of penalty functions for MDO formulations (e.g. BeHaYuSi00, 2000). However, here we explicitly motivate, select and compare different penalty function approaches, in order to match theoretical results with the application, as detailed in Section 1.



## 7 Conclusions

In this paper we have pursued a twofold purpose in a Multidisciplinary Optimization framework. First, we have briefly listed some renowned approaches from constrained nonlinear programming, which can represent useful tools for the solution of complex multidisciplinary optimization schemes. Then, we have focused on a specific application in ship design, where two disciplines are involved (namely *fluid dynamic* and *structural* disciplines), each requiring a specific and different solver. The strategy adopted to combine the two disciplines shows that the overall scheme yields very challenging numerical optimization problems, which require a fine and careful use of mathematical programming tools. In particular, considering that the minimization of non-differentiable functions was required, we have also integrated the heuristics PSO in our approach. Numerical results proved the effectiveness of our proposal, though further experiments may be fruitful to test the robustness of the approach. On this guideline, we are also considering to include robust optimization methods in our multidisciplinary optimization framework.

## References

- Bazaraa, M.S., Sherali, H.D. and Shetty, C.M. (1993) ‘Nonlinear Programming - Theory and Algorithms (2nd edition)’, *John Wiley & Sons*.
- Bertsekas, D.P. (1995) ‘Nonlinear Programming’, *Athena Scientific*, Belmont, Massachusetts, USA.
- Nocedal, J. and Wright, S.J. (2006) ‘Numerical Optimization’, *Springer-Verlag*, New York, Inc.
- Fiacco, A.V. and McCormick, G.P. (1968) ‘Nonlinear Programming: Sequential Unconstrained Minimization Techniques’, *John Wiley & Sons*.
- Bertsekas, D.P. (1982) ‘Constrained Optimization and Lagrange Multiplier Methods’, *Academic Press*.
- Wilson, R.B. (1963) ‘A Simplicial Algorithm for Concave Programming’, Ph.D. Thesis, Harvard University, USA.
- Powell, M.J.D. (1978) ‘The convergence of Variable Metric Methods for Nonlinear Constrained Optimization Calculation’, in O.L.Mangasarian, R.Meyer, S.M.Robinson, editors, *Nonlinear Programming 3*, Academic Press.
- Garcia-Palomares, U.M. and Mangasarian, O.L. (1976) ‘Superlinearly Convergent Quasi-Newton methods for Nonlinearly Constrained Optimization Problems’, *Mathematical Programming*, n.11.
- Powell, M.J.D. and Yuan, Y. (1986) ‘A recursive quadratic programming algorithm that uses differentiable exact penalty functions’, *Mathematical Programming*, n. 35, pp. 265-278.
- Courant, R. (1943) ‘Variational Methods for the Solution of Problems with Equilibrium and Vibration’, *Bull. Amer. Math. Soc.*, n.49, pp. 1-23.

- Zangwill, W.I. (1967) 'Nonlinear Programming via penalty functions', *Management Science*, n.13, pp. 344-358.
- Fletcher, R. (1970) 'A Class of Methods for Nonlinear Programming with Termination and Convergence Properties', In J.Abadie, editor, *Integer and Nonlinear Programming*, pp. 157-173, North-Holland.
- Powell, M.J.D. (1969) 'A Method for Nonlinear Constraints in Minimization Problems', in R.E.Fletcher, editor, *Optimization*, pp. 283-298, Academic Press.
- Rockafellar, R.T. (1973) 'The multiplier method of Hestenes and Powell applied to convex programming', *Journal of Optimization Theory and Applications*, n.12, pp. 555-562.
- Peri, D., Rossetti, M. and Campana, E.F. (2001) 'Design optimization of ship hulls via CFD techniques', *Journal of Ship Research*, n.45, pp. 140-149.
- Alexandrov, N.M. and Hussaini, M.Y. (Eds.) (1997) 'Multidisciplinary Design Optimization - state of the art', *Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization, SIAM Proceedings Series*.
- Raymer, D. (2002) 'Enhancing Aircraft Conceptual Design using Multidisciplinary Optimization', *Doctoral Thesis Report 2002-2*, May 2002, ISBN 91-7283-259-2, Department of Aeronautics Royal Institute of Technology SE-100 44, Stockholm, Sweden.
- Alexandrov, N.M. and Lewis, R.M. (1999) 'Comparative properties of Collaborative Optimization and other approaches to MDO', *Proceedings of the First ASMO UK/ISSMO CONFERENCE on Engineering Design Optimization*, July 8-9, 1999, MCB Press.
- Alexandrov, N.M. and Lewis, R.M. (2000) 'Algorithmic Perspectives on Problem Formulations in MDO', *AIAA 2000-4719*.
- Alexandrov, N.M. and Lewis, R.M. (2000) 'Analytical and Computational Properties of Distributed Approaches to MDO', *NASA Langley Research Center*, Hampton, Virginia, *AIAA 2000-4718*.
- Mor, J.J. and Wild, S. (2007) 'Benchmarking derivative-free optimization algorithms', *Technical Report Preprint ANL/MCSP1471- 1207*, Argonne National Laboratory.
- Kennedy, J. and Eberhart, R.C. (1995) 'Particle swarm optimization', *Proceedings of the 1995 IEEE International Conference on Neural Networks (Perth, Australia)*, IEEE Service Center, Piscataway, NJ, IV, pp. 1942-1948.
- Campana, E.F., Fasano, G. and Pinto, A. (2009) 'Dynamic system analysis for the selection of parameters and initial population, in Particle Swarm Optimization', *Journal of Global Optimization*, DOI: 10.1007/s10898-009-9493-0.
- Campana, E.F., Fasano, G., Peri, D. and Pinto A. (2007), 'Nonlinear Programming Approaches in the Multidisciplinary Design Optimization of a Sailing Yacht Keel Fin', *9th International Conference on Numerical Ship Hydrodynamics Ann Arbor, Michigan*, August 5-8, 2007.

- Belegundu, A.D., Halberg, E., Yukish, M.A. and Simpson, T.W. (2000) 'Attribute-Based Multidisciplinary Optimization of Undersea Vehicles', AIAA-2000-4865.
- Blackwell, T., Kennedy, J. and Poli, R. (2007) 'Particle swarm optimization - An overview', *Swarm Intelligence*, n. 1, pp. 33-57.
- Clerc, M. and Kennedy, J. (2002) 'The particle swarm - explosion, stability, and convergence in a multidimensional complex space', *IEEE Transaction on Evolutionary Computation*, n. 6, pp. 58-73.
- Fletcher, R. (1987) 'Practical methods of Optimization - Second Edition', John Wiley & Sons Ltd.
- Griewank, A. (2000) 'Automatic Differentiation', SIAM Philadelphia.
- Campana, E.F., Liuzzi, G., Lucidi, S., Peri, D., Piccialli, V. and Pinto, A. (2009) 'New global optimization methods for ship design problems', *Optimization Engineering*, n. 10, pp. 533-555.
- Peri, D. and Mandolesi, F. (2005) 'Multiobjective Optimization of an IACC Sailing Yacht by Means of CFD High-Fidelity Solvers', The 17<sup>th</sup> Chesapeake Sailing Yacht Symposium, Annapolis Maryland, March 2005.

