# Global Optimization Algorithms in Multidisciplinary Design Optimization

Daniele Peri,[*]

*INSEAN, Italian Ship Model Basin.*

Giovanni Fasano,[†]

*University Ca' Foscari of Venice, Italy*

Daniele Dessi[*] and Emilio F. Campana[*]

**While Multidisciplinay Design Optimization (MDO) literature focuses mainly on the development of different formulations, through the manipulation of design variables, less attention is generally devoted to the combination of specific MDO formulations with existing nonlinear optimization algorithms.**

**In this paper, the focus is on the application of a Global Optimization (GO) algorithm to an MDO problem. We first introduce and describe some MDO approaches from the literature. Then, we consider our MDO formulation where we deal with the GO box-constrained problem**

$$\min_{a \le x \le b} f(x), \qquad f : \mathbb{R}^n \to \mathbb{R}.$$

**We assume that the solution of the latter problem requires the use of a derivative-free methods since the derivatives of $f(x)$ are unavailable and/or the function must be treated as a 'black-box'. Within this framework we study some globally convergent modifications of the evolutionary Particle Swarm Optimization (PSO) algorithm, suitably adapted for box-constrained optimization. Finally, we report our numerical experience. Preliminary results are provided for a simple hydroelastic problem. Two different numerical tools are involved: a fluid dynamic solver, to simulate the flow around hydrofoils traveling in proximity of the air-water interface, and a simplified torsion-flexional wing model.**

## I. Introduction

Together with developments on numerical solvers and computer power, the recent years have seen some progress in the development of Simulation Based Design examples for ships too, and in the formulation and solution of some MDO problems. The ship design problem complexity, has however so far prevented from assessing a satisfactory reformulation of the overall design problem into a unique mathematical programming formulation. As a result, when several disciplines are involved in the design problem, different heuristic methods have been sequentially used, one for each discipline, which address individual disciplinary optimization. The growing complexity of modern engineering systems has spurred designers to provide more efficient heuristics. Unfortunately, the latter are often based on designers personal skills on the *specific problem* treated, instead of relying on self-adaptive techniques. Thus, the application of heuristics to new instances of general real problems, may be unsatisfactory.

These reasons motivate our interest for the systematic numerical approach to MDO. In our case the *multidisciplinarity* refers to the design of a ship, which encompasses interacting physical phenomena as hydrodynamics, structural mechanics, and control, to name a few.

Recently a larger number of real industrial applications have included complex optimization approaches, where efficient solutions were claimed. Aircraft and spacecraft engines design are among the latter applications, which intrinsically yield challenging MDO formulations (see e.g.[1]). One may observe that in most of the cases, MDO methodologies substantially imply a process of parallelization and coupling of different independent optimization schemes (*disciplines*). Moreover, a distinguishing feature of MDO formulations is that the interaction among the standard optimization approaches, each related to a discipline, is non-trivial.

Furthermore, the accurate coupling of disciplines might be essential to guarantee the convergence properties of the overall framework. Thus, a specific care should be paid to provide the *correctness* of the MDO formulation, for the problem in hand. On the other hand, both the theoretical results and the methods provided by nonlinear programming, for standard optimization problems, must be coupled accordingly. This suggests that those MDO formulations, which strongly rely on the abilities of optimization methods, may efficiently gain advantage from conventional optimization.[2,3]

On this guideline, observe that most of the typical issues considered for nonlinear programming formulations (e.g. feasibility, optimality conditions, sensitivity analysis, duality theory, etc.), require a suitable adaptation when considered in an MDO framework.

The first attempts to give a taxonomy of MDO formulations, simply relied on managing standard nonlinear programming schemes in a sequential fashion. I.e., no real coupling among the disciplines was considered, and the interaction among optimization codes was often non-significative. The latter scenario was essentially the consequence of the early incapability to match several numerical codes, independently studied for each discipline. In addition, large scale MDO problems were even tougher, so that coarse solutions had to be allowed and the coupling among disciplines was possibly weakened. As a consequence, the early MDO formulations often described quite poorly several non-convex real challenging problems.

This work briefly reviews the main results of MDO literature, including some more recent MDO formulations based on multilevel programming. Starting for this literature review, we are stressing how some particular global optimization algorithms can be efficiently applied in this context.

Then, we study and solve the MDO formulation of a vertical surface-piercing fin design problem, where the derivatives of the objective functions are unavailable. The latter problem is a hydroelastic design optimization problem, where the performances of the fin are depending from both the hydrodynamic and elastic features. We highlight that for several MDO problems the real functionals to be minimized are described by expensive simulations and the derivatives are unavailable. This strongly motivates the interest for effective derivative-free techniques.

We apply a modified Particle Swarm Optimization (PSO,[11]) method, which belongs to the family of evolutionary algorithms. PSO[16] owes its popularity to the reasonable balance between its overall computational cost and the quality of the final solution it provides. More specifically, the PSO algorithm is an *iterative method*, which is tailored to detect a global minimum for the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

i.e., a point $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$, for any $x \in \mathbb{R}^n$. For a computationally costly function $f(x)$, exact iterative methods are possibly too expensive or they may not provide a current satisfactory approximation of the solution, after a finite number of iterations. In these scenarios heuristics may be fruitfully used, whenever the computational resources and/or the time allowed for the computation are severely bounded. On this guideline, PSO proved to be both effective and efficient on several practical applications from real life.[22]

In-house solvers for the flow and elastic computations are here adopted: they are intentionally simple in order to reduce the computational cost, shifting the main effort on the algorithmic side. The great advantage of the self-developed solvers is reflected on the easiness in producing the interfaces between the codes and/or between the single code and the optimizer, guaranteing in the same time the complete control on the quality of the obtained results, being absolutely clear the kind of assumptions/approximations introduced in the solvers.

## II. Summary of MDO formulations

To clarify the details of the integration strategy between the MDO formulations and the GO algorithm, some MDO formulations will be re-formulated according to a line suggested by[13] and further by.[2,3]

The general formulation of an MDO problem is partially rewritten as:

$$\min_{(x,s,t) \in \hat{B}} \hat{f}(x,s,t), \qquad \hat{B} = \Gamma_1 \cap \Gamma_2 \cap \Gamma_3, \tag{2}$$

where the sets $\Gamma_1$, $\Gamma_2$, $\Gamma_3$ are respectively given by

*Design Constraints*:

$$\Gamma_1 = \begin{cases} g_0(x, s) \geq 0 \\ g_1(x_0, x_1, s) \geq 0 \\ \qquad \vdots \\ g_p(x_0, x_p, s) \geq 0 \end{cases}$$

*Disciplinary Analysis Constraints (MDA)*:

$$\Gamma_2 = \begin{cases} A_1(x_0, x_1, s_1, t_2, \ldots, t_p) = 0 \\ \qquad \vdots \\ A_p(x_0, x_p, s_p, t_1, \ldots, t_{p-1}) = 0 \end{cases}$$

*Interdisciplinary Consistency Constraints*:

$$\Gamma_3 = \begin{cases} t_1 = \mathcal{C}_1(s_1) \\ \quad \vdots \\ t_p = \mathcal{C}_p(s_p). \end{cases}$$

being $x_i$ the design variables, $x^T = (x_0^T, x_1^T, \ldots, x_p^T)$, $s_i$ the state variables, $g_i$ the design constraints, $A_i$ the set of PDEs governing each discipline and $t_i$ the interdisciplinary consistency constraints.

(*R1*) *MultiDisciplinary Feasible (MDF).* It is an MDO reformulation of (2) also known as *FIO* or *AIO*,[13] which represents the most trivial approach to the solution. It consists of using the implicit function theorem to explicit the vectors $s = s(x)$ and $t = t(x)$ from the Disciplinary Analysis and the Interdisciplinary Constraints. Then, the resulting MDO reformulation reduces to

$$\begin{aligned} \min_{x} \quad & \hat{f}(x, s(x), t(x)) \\ & g_0(x, s(x)) \geq 0 \\ & g_1(x_0, x_1, s(x)) \geq 0 \\ & \qquad \vdots \\ & g_p(x_0, x_p, s(x)) \geq 0, \end{aligned} \tag{3}$$

which may be treated as a nonlinear program depending on the vector of unknowns $x \in \mathbb{R}^n$. As we said, the equality constraints in (2) can be hardly inverted to provide $s = s(x)$, so that the reformulation (3) turns to be quite unusual. The MDF scheme is an OD/CDA/CIC reformulation.[2,3]

(*R2*) *Simultaneous Analysis and Design (SAD).* Also known with the acronyms *AAO* or *SAND*,[15] is the counterpart of MDF. Indeed, now $x$, $s$ and $t$ must be treated as independent unknowns, so that the overall reformulation to be solved is

$$\begin{aligned} \min_{x,s,t} \quad & \hat{f}(x, s, t) \\ & g_0(x, s) \geq 0 \\ & g_1(x_0, x_1, s) \geq 0 \\ & \qquad \vdots \\ & g_p(x_0, x_p, s) \geq 0 \\ & A_1(x_0, x_1, s_1, t_2, \ldots, t_p) = 0 \\ & \qquad \vdots \\ & A_p(x_0, x_p, s_p, t_1, \ldots, t_{p-1}) = 0 \\ & t_1 = \mathcal{C}_1(s_1) \\ & \qquad \vdots \\ & t_p = \mathcal{C}_p(s_p). \end{aligned} \tag{4}$$

Observe that the number of unknowns for SAD is relatively larger with respect to MDF. The SAD scheme is an OD/ODA/OIC reformulation.

(*R3*) *Distribute Analysis Optimization (DAO).* This is an intermediate approach (see Section VI) between the previous two; that is why it is often addressed as the *In Between*[13,18] reformulation, or alternatively it is the *IDF* approach.[13] Here, a subset of the equality constraints is used to explicit a sub-vector of the unknowns in terms of the remaining variables (i.e., the implicit function theorem may be partially applied). Considering the following partition of vectors $s^T = (\tilde{s}^T \ \ \hat{s}^T)$ and $t^T = (\tilde{t}^T \ \ \hat{t}^T)$, the resulting optimization problem becomes (for simplicity we have compounded the Disciplinary Analysis and the Interdisciplinary Consistency constraints)

$$\begin{aligned}
\min_{x,\tilde{s},\tilde{t}} \quad & \hat{f}\left[x, (\tilde{s}^T \ \ \hat{s}^T(x,\tilde{s}))^T, (\tilde{t}^T \ \ \hat{t}^T(x,\tilde{s}))^T\right] \\
& g_0\left[x, (\tilde{s}^T \ \ \hat{s}^T(x,\tilde{s}))^T\right] \geq 0 \\
& g_1\left[x_0, x_1, (\tilde{s}^T \ \ \hat{s}^T(x,\tilde{s}))^T\right] \geq 0 \\
& \qquad \vdots \\
& g_p\left[x_0, x_p, (\tilde{s}^T \ \ \hat{s}^T(x,\tilde{s}))^T\right] \geq 0 \\
& A\left[x, (\tilde{s}^T \ \ \hat{s}^T(x,\tilde{s}))^T, (\tilde{t}^T \ \ \hat{t}^T(x,\tilde{s}))^T\right] = 0 \\
& \tilde{t} = \tilde{C}(\tilde{s}).
\end{aligned} \tag{5}$$

Finally observe that the DAO scheme is an OD/CDA/OIC reformulation.

(*R4*) *Optimization by Linear Decomposition (OLD).* This is a true *bilevel reformulation* of (2).[14] Indeed, the first (*upper*) level of minimization (the master level) has the role of coordinating the results coming from the second (*lower*) level of minimization, which is the disciplines level. The resulting overall nonlinear program is

$$\begin{aligned}
\min_{x_0,t} \quad & \hat{f}\left[x_0, x_1, .., x_p, s_1(x_0,x_1,t), .., s_p(x_0,x_p,t)\right] \\
& g_0\left[x_0, x_1, .., x_p, s_1(x_0,x_1,t), .., s_p(x_0,x_p,t)\right] \geq 0 \\
& m_i(x_0,x_i,t) \leq 0, \qquad i \leq p \\
& \min_{x_i} \quad m_i(x_0,x_i,t) \\
& \qquad t_i = C_i\left[s_i(x_0,x_i,t)\right], \qquad i \leq p
\end{aligned} \tag{6}$$

where

$$m_i(x_0,x_i,t) = \left\| g_i^+\left[x_0, x_i, s_i(x_0,x_i,t)\right]\right\|^2 \tag{7}$$

and

$$\begin{aligned}
g_i^+\left[x_0, x_i, s_i(x_0,x_i,t)\right] \quad = & \\
& \min\left\{0, g_i\left[x_0, x_i, s_i(x_0,x_i,t)\right]\right\},
\end{aligned}$$

and the last equality is intended componentwise. Observe that here the sub-vector $s_i(x_0,x_i,t)$ is supposed to be computed by the implicit function theorem, applied to the $i$-th block of MDA constraints (i.e., $A_i(x_0,x_i,s_i,t_1,\ldots,t_{i-1},t_{i+1},\ldots,t_p) = 0$). The function $m_i()$ (the so called *discrepancy function*[3]) in both the upper and lower level of (6) substantially measures a penalization for infeasible solutions. Note that the exponent 2 in (7) is introduced in order to yield a continuously differentiable objective function, for the lower level.

(*R5*) *Collaborative Optimization (CO).* Similarly to OLD, CO is a multilevel optimization reformulation.[7] Here, the different role played by the system level and the disciplines level is strongly remarked. In particular, we allow the dependency of the Interdisciplinary Constraints from both the design variables and the state variables. The overall nonlinear program is described by introducing the so called *surrogates* $y_1, \dots, y_p$ of vector $x_0$. Observe that for each discipline, the latter unknowns are used to de-couple the upper level and the lower level, i.e. they play a role similar to that of vector $t$.

$$\begin{aligned}
&\min_{x_0,t} \quad \hat{f}(x_0, x_1, \dots, x_p, t) \\
&\qquad \|t_i - C_i\left(y_i - x_0, s_i(y_i, x_i, t)\right)\|_* = 0, \quad i \le p \\
&\qquad \min_{y_i, x_i} \quad \frac{1}{2}\left[\|y_i - x_0\|^2 + \|s_i(y_i, x_i, t) - t_i\|^2\right] \\
&\qquad\qquad g_i\left(y_i, x_i, s_i(y_i, x_i, t)\right) \ge 0, \quad i \le p.
\end{aligned} \tag{8}$$

The reformulation (8) strongly requires that the sub-vector $s_i = s_i(y_i, x_i, t)$ can be computed, by applying the implicit function theorem to the $i$-th block of MDA constraints $A_i(y_i, x_i, s_i, t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_p) = 0$. As reported above, the bilevel structure of $CO$ may be fruitfully exploited by using suitable nonlinear programming techniques.[14] Finally, the choice of the norm '$*$' is substantially arbitrary; however, common choices are '$*$'$= 1$ ($CO_1$) and '$*$'$= 2$ ($CO_2$).

With the choice '$*$'$= 1$ in (8), the constraints of the upper level are not differentiable. This implies that the Lagrange multiplier rule could not be applied.[23] On the other hand, the choice '$*$'$= 2$ is appealing because it gives a smooth feasible region of the upper level in (8). Unfortunately, in case the feasible region of the upper level is open, the KKT optimality conditions may fail, since the Jacobian matrix (of the upper level constraints) vanishes in any feasible point. Thus, the Lagrange multiplier rule[5] may fail as well.

## III. A generalized PSO scheme for GO

As described in Section I, PSO is an iterative heuristics for the solution of (1). It generates subsequences of points in $\mathbb{R}^n$ which possibly converge eventually to a stationary point of $f(x)$.

At the current iteration $k$ the PSO algorithm generates the $P$ sequences $\{x_j^k\}$, $j = 1, \dots, P$, according with (see[9]):

$$\begin{aligned}
v_j^{k+1} &= \chi\left[w^k v_j^k + c_j r_j (p_j^k - x_j^k) + c_g r_g (p_g^k - x_j^k)\right], \\
x_j^{k+1} &= x_j^k + v_j^{k+1}.
\end{aligned} \tag{9}$$

PSO belongs to the wide class of *evolutionary algorithms* and follows the natural paradigm of a bee swarm, where the trajectories of the bees (so called *particles*) are represented by the $P$ sequences $\{x_j^k\}$. On the other hand, the vector $v_j^k \in \mathbb{R}^n$ represents the so called *speed* of the $j$-th particle at iteration $k$. Finally, the $n$-real vectors $p_j^k$ and $p_g^k$, for any $k$, satisfy the conditions

$$\begin{aligned}
&1) \quad p_j^k \in \{x_j^\ell\} && \ell \le k, \; j = 1, \dots, P, \\
&2) \quad f(p_j^k) \le f(x_j^\ell) && \forall \ell \le k, \; j = 1, \dots, P,
\end{aligned} \tag{10}$$

and

$$\begin{aligned}
&1) \quad p_g^k \in \{x_1^\ell, \dots, x_P^\ell\} && \ell \le k, \\
&2) \quad f(p_g^k) \le f(x_j^\ell) && \forall \ell \le k, \; \forall j = 1, \dots, P.
\end{aligned} \tag{11}$$

Furthermore, $\chi, w^k, c_j, r_j, c_g, r_g$ are real bounded coefficients. Observe that we use the subscript $j$ to indicate the subsequence, while the superscript $k$ indicates the iterate in the subsequences $\{x_j^k\}$. Also note that

$p_j^k$ represents the 'best position' in the $j$-th subsequence, while $p_g^k$ is the 'best position' among all the subsequences. The choice of the coefficients is often problem dependent; however, several values for them were proposed in the literature.[9,12,21,24] In particular, the parameters $r_j$ and $r_g$ are often random parameters with uniform distribution in $[0,1]$.

Observe that in relation (9) the speed $v_j^{k+1}$ depends only on the vectors $p_j^k - x_j^k$, $p_g^k - x_j^k$. However, for the $j$-th particle an obvious generalization of (9) could be the following[9]

$$\begin{aligned} v_j^{k+1} &= \chi_j^k \left[ w_j^k v_j^k + \sum_{h=1}^{P} c_{h,j} r_{h,j} (p_h^k - x_j^k) \right], \\ x_j^{k+1} &= x_j^k + v_j^{k+1}, \end{aligned} \tag{12}$$

where the speed $v_j^{k+1}$ depends on the $P$ vectors $p_h^k - x_j^k$ (see also[19]), $h = 1, \ldots, P$.

Now, assuming $\chi_j^k = \chi_j$ and $w_j^k = w_j$, for any $k \geq 0$, the iteration (12) is equivalent to the *discrete stationary (time-invariant) system*

$$\begin{aligned} X_j(k+1) &= \begin{pmatrix} a_j I & -\omega_j I \\ & \\ a_j I & (1-\omega_j) I \end{pmatrix} X_j(k) \\ &\quad + \begin{pmatrix} \displaystyle\sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j} p_h^k \\ \displaystyle\sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j} p_h^k \end{pmatrix}, \end{aligned} \tag{13}$$

where $a_j = \chi_j w_j, \;\; \omega_j = \sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j}$ and

$$X_j(k) = \begin{pmatrix} v_j^k \\ \\ x_j^k \end{pmatrix} \in \mathrm{I\!R}^{2n}, \qquad k \geq 0. \tag{14}$$

The sequence $\{X_j(k)\}$ identifies the trajectory of the $j$-th particle in the real space $\mathrm{I\!R}^{2n}$. In addition, this trajectory can be split into the *free response* $X_{j\mathcal{L}}(k)$ and the *forced response* $X_{j\mathcal{F}}(k)$ (see also[20]). In other words, for any $k \geq 0$, $X_j(k)$ may be rewritten according with

$$X_j(k) = X_{j\mathcal{L}}(k) + X_{j\mathcal{F}}(k), \tag{15}$$

where

$$X_{j\mathcal{L}}(k) = \Phi_j(k) X_j(0), \qquad X_{j\mathcal{F}}(k) \;=\; \sum_{\tau=0}^{k-1} H_j(k-\tau) U_j(\tau), \tag{16}$$

and (after few calculations[9])

$$\Phi_j(k) \;=\; \begin{pmatrix} a_j I & -\omega_j I \\ & \\ a_j I & (1-\omega_j) I \end{pmatrix}^{k}, \tag{17}$$

$$H_j(k-\tau) \;=\; \begin{pmatrix} a_j I & -\omega_j I \\ & \\ a_j I & (1-\omega_j) I \end{pmatrix}^{k-\tau-1}, \tag{18}$$

$$U_j(\tau) \;=\; \begin{pmatrix} \displaystyle\sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j} p_h^\tau \\ \displaystyle\sum_{h=1}^{P} \chi_j c_{h,j} r_{h,j} p_h^\tau \end{pmatrix}. \tag{19}$$

The issue of the convergence of the free response of the system has been partially investigated in[17] and finalized in.[11]

# IV. PSO algorithm in a DAO formulation

As from Section II, DAO formulation is open with respect to design variable and interdisciplinary consistency constraints. The latter are satisfied once we approach the set of optimal design variables. In fact, if we consider a small variation of a prescribed design, we can assume that the values of the interdisciplinary variables do not change dramatically. This sentence is untrue, for example, if we are in the vicinity of a particular transitional state for the phenomena under analysis. If we are also in the position to obtain a small modification in the design object for a small variation of the design variables, what we need are optimization algorithms than naturally search the optimal solution by widely exploring the design space. This is not the typical behavior of a global optimization algorithm: as a consequence, in the literature of MDO we find a large number of applications in which the different objective function of the problem are re-casted into a single merit function, and this problem is solved by using gradient-based optimization algorithms. By the way, using the DAO formulation, the approximated computation of the gradient of the objective function may be further affected by the uncertainty given by the incomplete convergence of the single MDA.

This approach heavily penalizes the final solution, since we are in the position to discover only local minima of the problem in hands. Viceversa, the application of a global optimization algorithm can produce a larger improvement because global optima are possibly found. If we observe the behavior of the PSO algorithm, we can see how a number of interacting elements explore the design space in a coordinate way, all looking for the global optimum of the objective function. The design space is globally investigated, but the pattern of every single particle is potentially continuous, depending on the maximum speed the particle is allowed to assume at each iteration. If we consider a DAO formulation for each swarm element separately, we can obtain the convergence of the interdisciplinary constraints individually.

Furthermore, the PSO algorithm is also suitable for multi-objective problems: under this perspective, using this algorithm a solution of a multi-objective problem, in which the objective function is not recast into a single merit function, can be preformed. In this case, the full Pareto front of the problem can be determined, instead of a single optimal solution valid just for the adopted recombination of the objective functions.

The classical approach of the gradient-based algorithm for the solution of (5) has been utilized in,[8] and a comparison between the application of a gradient-based local optimization algorithm and a derivative-free algorithm has been proposed in,[10] giving also evidence of the convergence of the interdisciplinary consistency constraints, as expected. In the following application, the PSO algorithm is tested on an MDO problem: a vertical fin is travelling across the free surface, also deflecting due to the drift angle. While in the previous papers an open-source code has been adopted for the solution of the elastic problem, here an in-house code is applied.

# V. Numerical Tools

For the solution of the hydro-elastic problem, we need two separate solvers, one for the solution of the hydrodynamic problem, one for the elastic problem. A very simple code is utilized for the elastic problem solution, and the fin is modelled using the thin beam approximation. For the flow solution, the Laplace equations are solved by using a Boundary Element Method (BEM) for incompressible flows.

The MDA is performed iteratively: starting form the undeformed fin geometry, the pressures computed by the fluid dynamic solver are the input loads for the structural solver. Conversely, the displacements computed by the structural solver are applied to the fin and the deformed fin is computed again by the flow solver. The procedure stops once the loads and deformation converge to stable values.

## A. BEM for free-surface computations

The hypothesis of potential flows is assumed in this case. Vorticity is confined in zero-volume regions, managing the wake as a thin surface. Two different plane elements are used for the distribution of unknowns on the body and on the free surface, whose are limiting the computational domain. A system of Hess and Smith plane sources, distribution on the exact body surface, is used in conjunction with vortex rings, in order to include lifting surfaces. The Poincaré formulation is giving the expression of the speed due to a vortex ring, or and horseshoe vortex representing the wake.[6] Wake position is an unknown of the problem, and the deformation of the free surface as well. The problem is solved iteratively: at each step the wake is aligned with the local speed, and the free surface collocation points are moved according to the actual local wave elevation. Impermeability condition

$$\nabla\Phi \cdot \vec{n} = 0$$

is enforced on the body while

$$\varphi_{ll}\varphi_l^2 + \varphi_z/Fr^2 = 0$$

is the free surface non linear condition, where $Fr$ is the *Froude number* and the derivatives are also performed along the streamline $l$.[4]

### B. FEM for elastic computations

An equivalent beam is produced basing on the actual fin geometry. A number of strips is considered spanwise: on these locations, the 3D loads distribution is converted into a set of concentrated forces and moments. For each strip, the equilibrium of the loads is enforced, and displacements are computed accordingly.

## VI. Numerical Experiments

A surface piercing vertical fin is the argument of this numerical experiment. The original profile is a NACA0012, with a chord of 10 cm and a span of 60 cm. It is travelling at the speed of 10 m/s with an angle of attack of 5 degrees. Objective function is the efficiency of the fin, defined as the ratio between vertical and longitudinal forces. This function is to be maximized.

The *Free Form Deformation* approach has been used for the purpose of shape deformation.[10] The fin is deformed maintaining the cylindrical shape along the vertical axis. 4 design variables are applied all on the same side of the profile: geometrical variation is transmitted to the opposite side of the profile in order to enforce symmetry.

In order to check the convergence properties of the interdisciplinary design constraints, the value of the hydrodynamic functions are traced trough the MDA iterations. If we compute the longitudinal (Fx) and lateral (Fy) forces at two successive iteration of the MDA, we define

$$\delta = \sqrt{\left(\frac{Fx_k - Fx_{k-1}}{Fx_{k-1}}\right)^2 + \left(\frac{Fy_k - Fy_{k-1}}{Fy_{k-1}}\right)^2}$$

where the index $k$ is counting the MDA internal iterations. We stop the MDA if $\delta$ is lower than a prescribed limit. At least two successive evaluations of the hydrodynamic function are needed for the computation of $\delta$.

In figure 1 the objective function and the convergence parameter $\delta$ for the original configuration are reported as a function of the number of inner iteration of the MDA, achieving a multidisciplinary equilibrium. We can see that a number of 13 inner steps are needed for the full convergence of the MDA: here a value of $\delta$ close to $10^{-5}$ is obtained, that means that the difference in the objective function prediction between two iterations is lover than 0.001%. In the following computations, a limit of $10^{-3}$ is adopted in order to consider the MDA converged.

In figure 2, a front-up view of the fin and the free surface is presented. On left, the wave elevation produced by the rigid (undeformed) fin is shown, whereas on right the same picture but for the fin deformed under loads is reported. Only small differences are visible: this is connected with the small deformation of the fin in the region across the water surface. The free surface is colored according to the local wave elevation: red is representing high crest, blue is for deep trough.

The best solution of the MDO problem is provided by the profile reported in figure 3. The profile is thicker in the front part and thinner at the trailing edge. The efficiency of the profile is increasing passing from the value of 0.545 (original profile) to 0.653 (optimal profile). The final gain is around 20%. We remember that this improvement is obtained considering throughout the optimization process the real deformation of the fin under the hydrodynamic loads.

The elastic response of the optimal fin appear to be increased if we compare the deformations reported in figure 4: in particular, an higher torsion of the fin is producing a local reduction of the angle of attack.

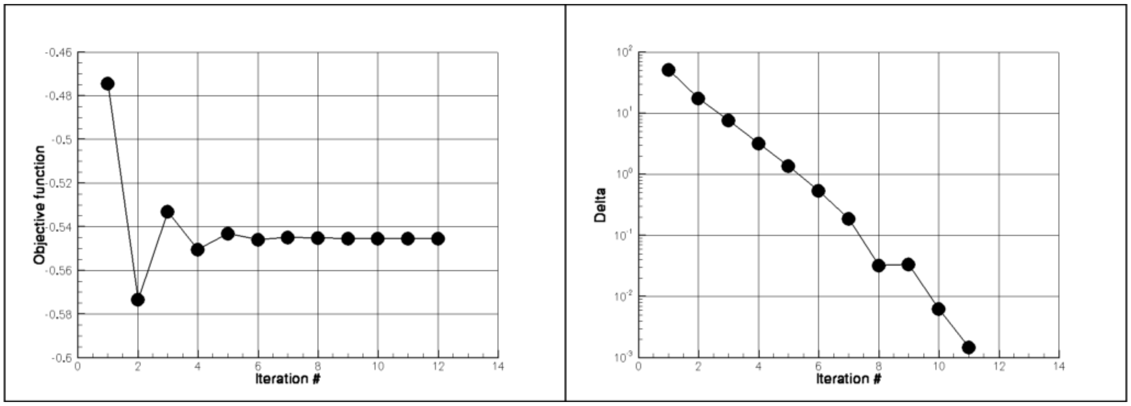

**Figure 1. Convergence history of the MDA of the original design. In abscissa the iteration number of the MDA is reported. On left, the vertical axis reports the objective function values. Good convergence is obtained in 13 steps. On right, the vertical axis reports the values assumed by the parameter $\delta$.**

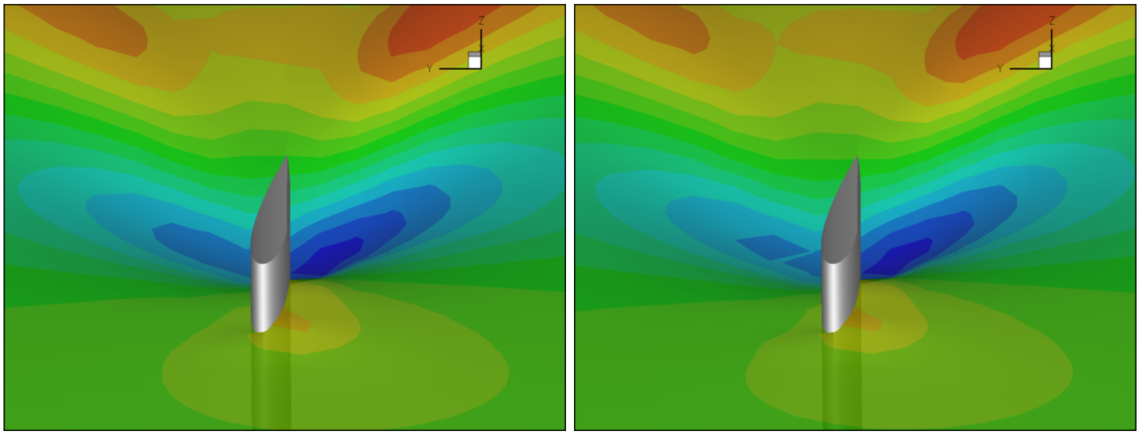

**Figure 2. Comparison of the free surface deformation for the fin with and without elastic deformation. On left, the free surface elevation past the undeformed fin, on right the same but with the fin under loads. Colors are indicating the water level: red is representing high crest, blue is for deep trough. Small differences are perceptible due to the small deformation of the fin near the free surface.**

If we compute the flow around the original and optimized geometry without taking into account the effect of the loads on the structure, the efficiency for the original profile would be 0.714, whereas it would be 0.936 for the optimal profile, producing an ideal gain of about 31%. This means that the improvement is not obtained by reducing the deformation of the fin: the larger part of the improvement is obtained just by enhancing the hydrodynamic performances of the shape.

The maximum number of objective function evaluations has been fixed at $100 * N$, being $N$ the number of design variables (4 in this example). This number is evidently too low for this numerical experiment: if we observe the results in figure 5, we can see how the algorithm is producing an initial large exploration of the design variable space, and than it start focusing on a limited region. This behavior is also evident by looking at the convergence history reported on figure 6, where a sudden steepness change is indicating a rapid improvement of the objective function. Unfortunately, the process is stopped before a real convergence is achieved. Further computations are needed to produce a complete exploration of the capacities of the method.

A different way to observe this behavior is from the design variable evolution, reported in figure 5. Here the values of the design variables for all the particles are reported together in a single graph, one for each design variable. Here we can see the initial large exploration phase, and than a sudden diversion of all the swarm toward a single direction. One particle is not converging together with all the others: it will be

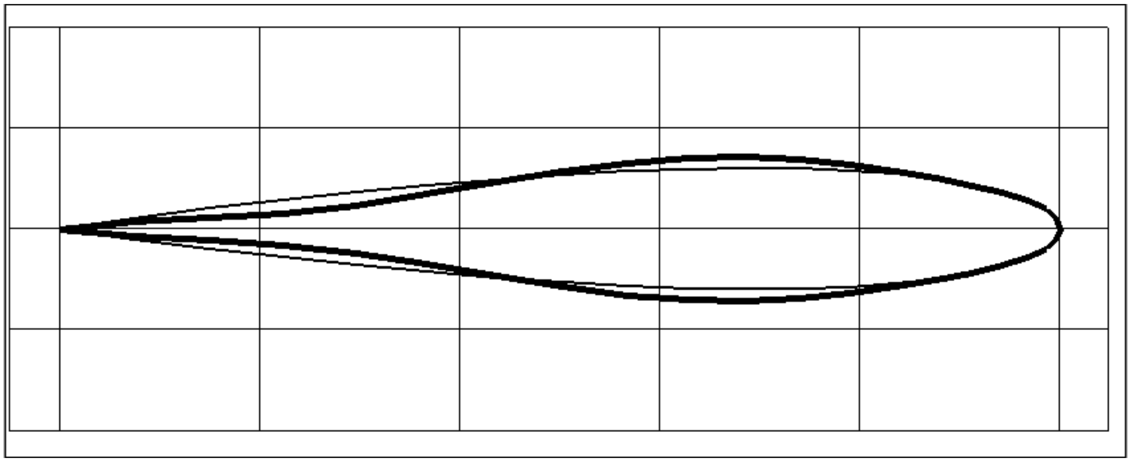

**Figure 3. Optimal shape at the end of the optimization process. Thick solid line is representing the optimal solution, while thin line is showing the initial shape.**

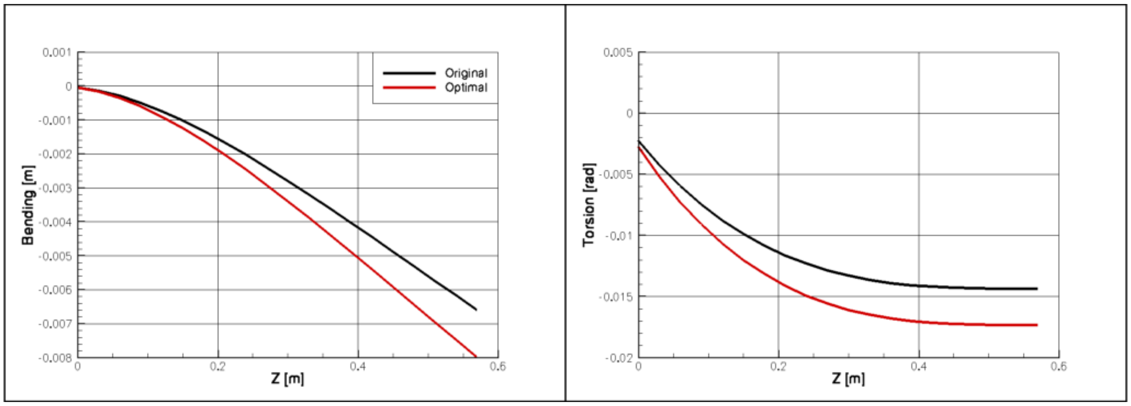

**Figure 4. Comparison of bending and torsion of the fin for the original and optimal shape. Bending is reported on left as a function of the span, while the torsion of the fin is shown on right.**

probably recovered at the end of the optimization cycle, when all the particles are supposed to converge into an unique location.

## VII. Conclusions

Optimization problems in which the solution depends on more *disciplines* may be tackled with MDO. In this paper we present results for the MDO optimal shape of a vertical fin travelling across the free surface, simultaneously accounting for hydrodynamics and elasticity. In the MDO framework the fin is assumed to be elastic and hence it can be modified by the hydrodynamic loads.

The optimal design problem is tackled considering a Global Optimization (GO) problem within a MDO framework. Some numerical experiments have been performed on a test problem, showing how global optimization algorithms can be applied in this context.

Future work will include the application of this technique to multi-objective problems, a more articulated reshaping of the profile (as to the optimizer), the modelling of the fin using non-isotropic materials (like carbon fiber) for the structural analysis, and the use of a RANS solver for the hydrodynamic analysis.

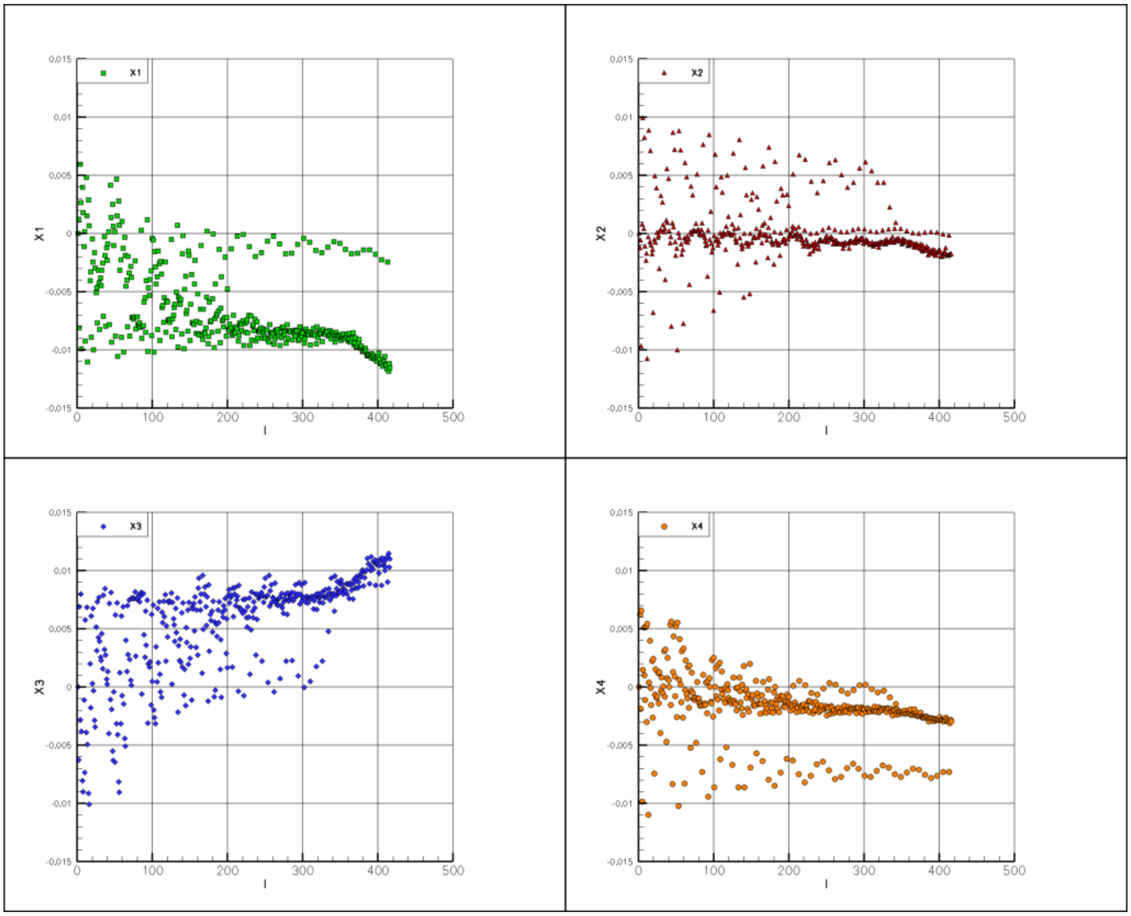

**Figure 5. Convergence history of all the 8 particles forming the swarm.**

## Acknowledgments

EFC, DP and DD have been partially supported by the research program '*ViSIR*' and '*Sicurezza*'. GF wishes to thank '*Programma di Ricerca INSEAN 2007-2009*' for the support received.

## References

[1] NM Alexandrov, MY Hussaini (Eds.), 1997, *Multidisciplinary Design Optimization - state of the art*, Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization, SIAM Proceedings Series.

[2] NM Alexandrov, RM Lewis, 2000, *Algorithmic Perspectives on Problem Formulations in MDO*, AIAA 2000-4719.

[3] NM Alexandrov, RM Lewis, 2000, *Analytical and Computational Properties of Distributed Approaches to MDO*, NASA Langley Research Center, Hampton, Virginia, AIAA 2000-4718.

[4] P Bassanini, UP Bulgarelli, EF Campana, F Lalli, 1994, *The wave resistance problem in a boundary integral formulation*, Surveys on Mathematics for Industry, **4**, 151-194.

[5] DP Bertsekas, 1995, *Nonlinear Programming*, Athena Scientific, Belmont, Massachusetts.

[6] W Bollay, 1939, *A Non-linear Wing Theory and its Application to Rectangular Wings of Small Aspect Ratio*, Z. angew. Math. Mech, Bd 9 Nr. 1 Febr. 1939

[7] RD Braun, 1996, *An architecture for large-scale distributed design*, PhD thesis, Stanford University, Department of Aeronautics and Astronautics.

[8] EF Campana, D Peri, G Fasano, 2005, *Numerical experience in MDO for ship design*, 2005 SIAM Conference on Optimization, Stockholm (Sweden)

[9] EF Campana, G Fasano, A Pinto, 2005, *Particle Swarm Optimization: dynamic system analysis for parameter selection in global optimization frameworks*, Technical Report INSEAN 2005-023, Italy.

[10] Campana, E.F., Fasano G., Peri D., Issues of Non-Linear Programing for Multidisciplinary Design Optimization (MDO) framework, NuTTS '05 Symposyum, Varna (Bulgaria), 2005

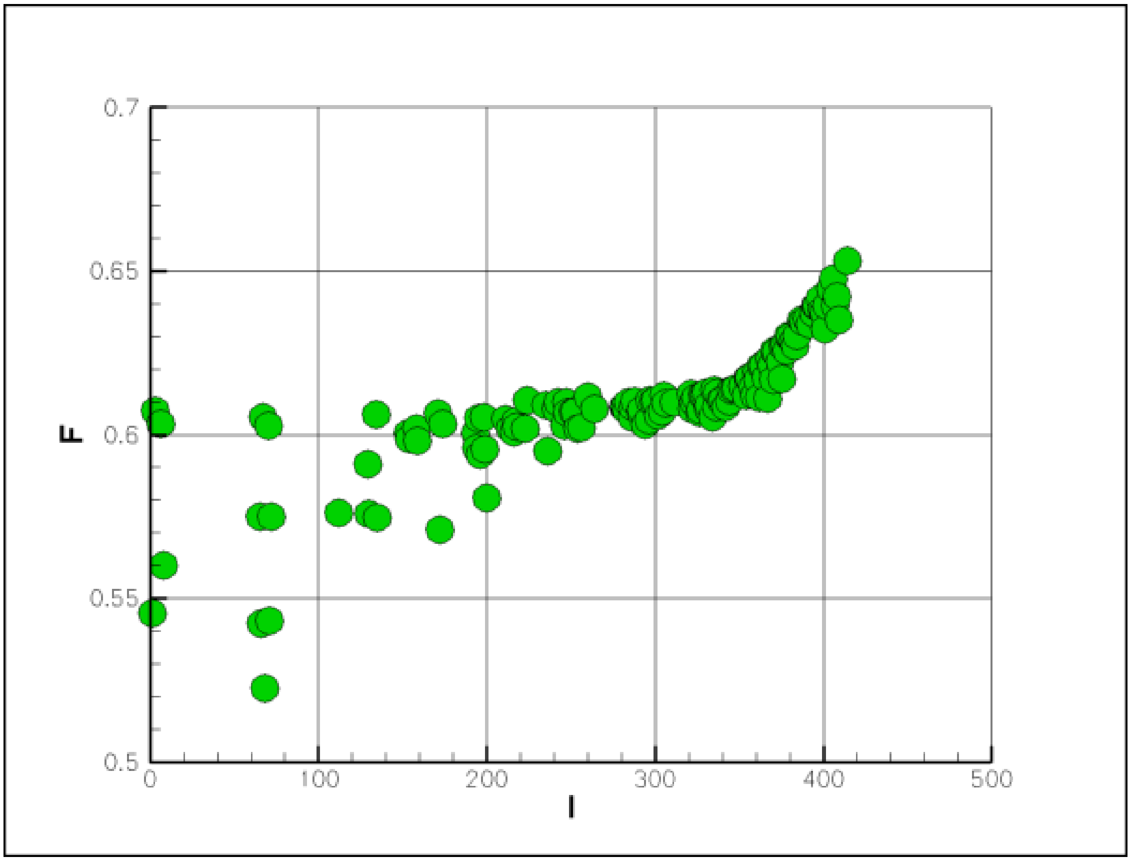

**Figure 6. Convergence history of the objective function.**

[11] EF Campana, G Fasano, A Pinto, 2006, *Dynamic system analysis and initial particles position in Particle Swarm Optimization*, IEEE Swarm Intelligence Symposium, Indianapolis

[12] M Clerc, J Kennedy, 2002, *The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space,* IEEE Transactions on Evolutionary Computation, Vol. 6, No. 1.

[13] EJ Cramer, JE Dennis Jr., PD Frank, RM Lewis, GR Shubin, 1994, *Problem Formulation for Multidisciplinary Optimization*, SIAM Journal on Optimization, Vol. 4., Issue 4, pp. 754-776.

[14] S Dempe, 2002, *Foundations of Bilevel Programming*, Kluwer Academic Publishers, Dordrecht, Boston, London, The Netherlands.

[15] RT Haftka, Z Gurdal, MP Kamat, 1990, *Elements of Structural Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherland.

[16] J Kennedy, RC Eberhart, 1995, *Particle swarm optimization*, Proceedings of the 1995 IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948, 1995.

[17] TG Kolda, RM Lewis, V Torczon, 2003, *Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods* SIAM Review Vol. 45, No. 3, pp. 385-482.

[18] RM Lewis, 1997, *Practical aspects of variable reduction formulations and reduced basis algorithms in Multidisciplinary Design Optimization*, in Natalia M. Alaxandrov, M. Y. Hussaini (Eds.), 1997, *Multidisciplinary Design Optimization - state of the art*, Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization, SIAM Proceedings Series.

[19] R Mendes, 2004, *Population Topologies and Their Influence in Particle Swarm Performance,* PhD Dissertation, University of Minho, Departamento de Informtica Escola de Engenharia Universidade do Minho.

[20] PE Sarachik, 1997, *Principles of linear systems,* Cambridge University Press.

[21] JF Schutte, AA Groenwold, 2005, *A Study of Global Optimization Using particle Swarms,* Journal of Global Optimization, No. 31, pp. 93-108.

[22] Y Shi, R Eberhart, 1998, *Parameter Selection in Particle Swarm Optimization,* The seventh Annual Conference on Evolutionary Computation, 1945-1950.

[23] R Thareja, RT Haftka, 1986, *Numerical Difficulties associated with using equality constraints to achieve multi-level decomposition in structural optimization*, AIAA Paper 86-0854.

[24] YL Zheng, LH Ma, LY Zhang, JX Qian, 2003, *On the convergence analysis and parameter selection in particle swarm optimization,* Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an.