Multi-Level Parameterization for Shape Optimization in Aerodynamics and Electromagnetics using a Particle Swarm Optimization Algorithm

Régis Duvigneau, Benoît Chaigne, Jean-Antoine Désidéri

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Multi-Level Parameterization for Shape Optimization in Aerodynamics and Electromagnetics using a Particle Swarm Optimization Algorithm

Régis Duvigneau, Benoît Chaigne and Jean-Antoine Désidéri

N° 6003

Octobre 2006

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--6003--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Multi-Level Parameterization for Shape Optimization in Aerodynamics and Electromagnetics using a Particle Swarm Optimization Algorithm

Régis Duvigneau*, Benoît Chaigne† and Jean-Antoine Désidéri‡

Thème NUM — Systèmes numériques
Projet Opale

Rapport de recherche n° 6003 — Octobre 2006 — 27 pages

Abstract: A multi-level parameterization approach for shape optimization problems is developed in the framework of Particle Swarm Optimization (PSO) algorithms. The Free-Form Deformation (FFD) technique based on a third order Bézier tensor product is used to construct a set of nested parameterizations. Then, optimization is carried out in the corresponding search spaces of variable dimension using a PSO algorithm. A method is developed to transfer the information collected from the optimization using a given parameterization, to the optimization procedure using a finer level. The proposed method is applied to shape optimization problems in electromagnetics and aerodynamics. Improvements of the convergence rate and the fitness obtained are reported.

Key-words: Shape optimization, Particle Swarm Optimization, Multi-level approach, Parameterization, Aerodynamics, Electromagnetics.

* Opale Project-Team
† Opale Project-Team
‡ Opale Project-Team

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Paramétrisation multi-niveau pour l'optimisation de forme en aérodynamique et électromagnétisme par un algorithme d'optimisation par essaim de particules

Résumé : Une méthode de paramétrisation multiniveau est développée pour les problèmes d'optimisation de forme, dans le cadre des algorithmes d'optimisation par essaim de particules (PSO). La méthode des boites englobantes (FFD) basée sur un produit tensoriel triple de Bézier est employée pour construire une famille de paramétrisations emboitées. Ensuite, l'optimisation est réalisée dans les espaces correspondants, de dimension variable, avec un algorithme PSO. Une méthode est développée pour transférer les informations collectées durant l'optimisation avec une paramétrisation donnée, vers la procédure d'optimisation utilisant une paramétrisation plus fine. La méthode proposée est appliquée à des problèmes d'optimisation de forme en électromagnétique et aérodynamique. Des améliorations du taux de convergence et de la performance obtenue sont rapportées.

Mots-clés : Optimisation de forme, Optimisation par essaim de particules, approche multiniveau, paramétrisation, aeérodynamique, électromagnétisme.

Contents

Introduction

Shape optimization problems for systems governed by PDEs, such as those encountered in aerodynamics or electromagnetics, are challenging problems, because of the multimodality of the objective function and the computational cost related to its evaluation. Hence, the use of sophisticated optimization algorithms is required, which are able to avoid local optima using a low number of evaluations.

Prior works[2, 16, 19] have shown that a multi-level parameterization strategy applied to a gradient-based optimization method can significantly speed up the convergence. Basically, it consists in using different shape representation levels during the optimization procedure, ranging from a coarse level parameterization including a small number of design variables to a fine level parameterization which implies a large number of design variables. Hence, the search for the optimal shape is carried out in several embedded design spaces of variable dimension. Several strategies can be considered, which are inspired from the multi-grid theory for Partial Differential Equations (PDEs) solving, ranging from simple level increase to V-cycle or Full Multi-Grid (FMG) approaches.

However, these strategies are dedicated to descent methods, which can easily be trapped into local optima. Their use in the framework of semi-stochastic optimization methods, such as genetic algorithms (GAs) or evolutionary algorithms (EAs), is not straightforward. Therefore, an other semi-stochastic optimization algorithm, so-called Particles Swarm Optimization (PSO), is studied here and a new scheme is proposed to include this algorithm into a multi-level parameterization approach.

The resulting method, called Multi-level Particle Swarm Optimization Algorithm (MPSOA), is applied to shape optimization problems in electromagnetics and aerodynamics, and its efficiency is demonstrated. Especially, an inverse problem in electromagnetics is first studied, which consists in conforming reflector antennas with respect to radiation diagrams. Then, the aerodynamic design of a wing of a transsonic business jet is considered, whose objective is the drag minimization under a constant lift constraint.

1 Parametric shape optimization

1.1 Principles

A shape optimization problem consists in minimizing a cost function $\mathcal{J}$, which depends on a shape Γ and state variables W. A parametric approach is adopted here, since the shape Γ is defined by a small number of design variables x_c, which are considered as optimization variables. The use of such a parametric approach allows to replace the initial shape optimization problem of infinite dimension by a problem with a finite number n of unknowns. State variables W (i.e. physical fields) depend implicitly on the design variables through the state equations $\mathcal{E}$, which are considered as constraints. Finally, a general parametric shape optimization problem can be expressed as :

$$\begin{aligned} \text{Minimize } & \mathcal{J}(x_c, W(x_c)) \quad x_c \in \Re^n, \\ \text{Subject to } & \mathcal{E}(x_c, W(x_c)) = 0, \\ & \mathcal{C}(x_c, W(x_c)) \leq 0, \end{aligned} \tag{1}$$

where $\mathcal{C}$ represents additional (physical or geometrical) constraints. Several methods can be carried out to solve such a problem. The numerical implementation of these methods often depend on the field of interest. Nevertheless, a typical algorithm to solve such a problem can be described as follows:

1. choose initial design variables $x_c^{(0)}$;
   $k \leftarrow 0$;
2. begin iteration k of the optimization loop ;
3. solve the state equations $\mathcal{E}(x_c^{(k)}, W(x_c^{(k)})) = 0$ yielding the state variables $W(x_c^{(k)})$;
4. estimate the cost function $\mathcal{J}(x_c^{(k)}, W(x_c^{(k)}))$;
5. update the design variables to $x_c^{(k+1)}$ according to the optimization algorithm ;
6. finish iteration k of the optimization loop ;
   if a stopping criterion is reached then STOP ;
   else $k \leftarrow k+1$ GOTO step (2).

The shape parameterization method, which describes how the shape Γ is defined by the design variables x_c, is detailed in the next section. Then, the optimization algorithm used for both applications in aerodynamics and electromagnetics is described.

1.2 Parameterization using the Free-Form Deformation approach

A critical issue in parametric shape optimization is the choice of the shape parameterization. The objective of the parameterization is to describe the shape, or the shape modification, by a set of parameters which are considered as design variables during the optimization procedure. Parameterization techniques in shape optimization have to fulfill several practical criteria :

- the parameterization should be able to take into account complex geometries, possibly including constraints and singularities ;
- the number of parameters should be as small as possible, since the stiffness of the shape optimization numerical formulation increases abruptly with the number of parameters ;
- the parameterization should allow to control the smoothness of the resulting shapes ;

A survey of shape parameterization techniques for multi-disciplinary optimization, which are analyzed according to the previous criteria, is proposed by Samareh [21]. In accordance with his conclusions, the Free-Form Deformation (FFD) technique [22] is adopted in the present study, since it provides an easy and powerful framework for the deformation of complex shapes, as those encountered in aerodynamics or electromagnetics.

The FFD technique originates from the Computer Graphics field [22]. It allows the deformation of an object in a 2D or 3D space, regardless of the representation of this object. Instead of manipulating the surface of the object directly, by using classical B-Splines or Bézier parameterization of the surface, the FFD techniques defines a deformation field over the space embedded in a lattice which is built around the object. By transforming the space coordinates inside the lattice, the FFD technique deforms the object, regardless of its geometrical description.

More precisely, consider a three-dimensional hexahedral lattice embedding the object to be deformed. Figure (1) shows an example of such a lattice built around a realistic wing. A local coordinate system (ξ, η, ζ) is defined in the lattice, with $(\xi, \eta, \zeta) \in [0,1] \times [0,1] \times [0,1]$. During the deformation, the displacement Δq of each point q inside the lattice is here defined by a third-order Bézier tensor product:

$$\Delta q = \sum_{i=0}^{n_i} \sum_{j=0}^{n_j} \sum_{k=0}^{n_k} B_i^{n_i}(\xi_q) B_j^{n_j}(\eta_q) B_k^{n_k}(\zeta_q) \Delta P_{ijk}. \tag{2}$$

$B_i^{n_i}$, $B_j^{n_j}$ and $B_k^{n_k}$ are the Bernstein polynomials of order n_i, n_j and n_k (see for instance [9]):

$$B_p^n(t) = C_n^p \, t^p \, (1-t)^{n-p}. \tag{3}$$

$(\Delta P_{ijk})_{0 \le i \le n_i, 0 \le j \le n_j, 0 \le k \le n_k}$ are weighting coefficients, or control points displacements, which are used to monitor the deformation and are considered as design variables during the shape optimization procedure.

The FFD technique described above is well suited to complex shape optimization, thanks to the following properties:

- the initial shape can be exactly represented (no deformation occurs when all weighting coefficients are zero) ;
- the deformation is performed whatever the complexity of the shape (this is a free-form technique) ;
- geometric singularities can be taken into account (the initial shape including its singularities is deformed) ;
- the smoothness of the deformation is controlled (the deformation is ruled by Bernstein polynomials) ;
- the number of design variables depends on the user's choice (the deformation is independent of the shape itself) ;
- it nicely deals with multi-level representation (thanks to the Bézier degree elevation property).

The FFD technique is implemented in the shape optimization procedure and is used to control the shape deformation for applications in both aerodynamics and electromagnetics.

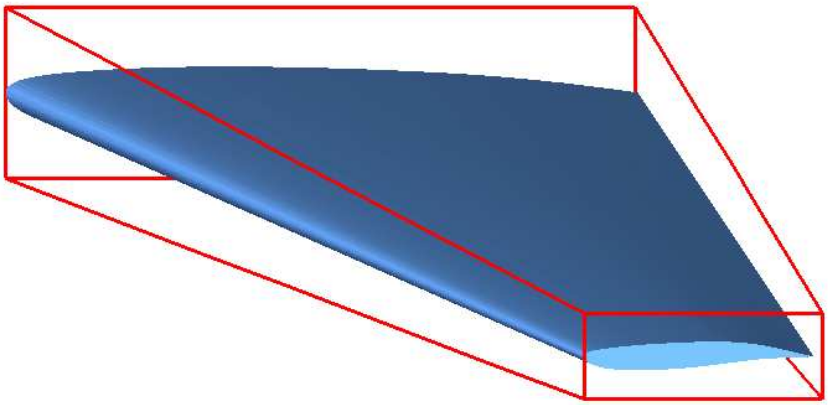


Figure 1: Example of FFD lattice (red) around a wing.

1.3 Particle Swarm Optimization

As explained above, several optimization strategies can be implemented to solve the problem (1), whatever the field of interest is. Gradient-based optimization methods, such as Quasi-Newton methods [11], are well suited to local and convex optimization, whereas semi-stochastic optimizers are devoted to global and multi-modal problems. Shape optimization problems in aerodynamics or electromagnetics usually exhibit non-linear behaviors, which often yield multi-modal functionals. Therefore, the present study is focused on the use of semi-stochastic methods for global optimization, although gradient-based approaches can also be useful for local search.

Most semi-stochastic optimization methods are inspired from natural phenomena or physical processes, such as Genetic Algorithms [12, 13] (GAs), Evolutionary Algorithms [3, 4] (EAs) or Simulated Annealing [1] (SA). The efficiency of these methods has been demonstrated in the past for several applications in different fields. On the contrary, Particle Swarm Optimization (PSO) is an arising method that is still maturing. Nevertheless, PSO algorithm exhibits some characteristics that are highly valuable for multi-level approaches, as will be explained latter. PSO algorithms was first introduced by Kennedy and Eberhart [14], as a simplified social model. It mimics the behavior of birds flocking and is based on underlying rules that enable sudden direction changes, scattering, regrouping, etc. These moves are motivated in nature by food seeking or predators avoiding and can be implemented in a simple algorithm for global optimization purpose [10, 23, 24].

The algorithm consists in building the trajectories of the particles of a swarm in the search space of dimension n. These trajectories are adjusted dynamically to take into account the information collected about the cost function value. Consider a set of p particles that constitutes the swarm. The location of each particle i at the time step k is determined by its coordinates $(x_i^k)_{1\leq i\leq p} \in \Re^n$. The behavior of each particle is then characterized by its velocity $(v_i^k)_{1\leq i\leq p} \in \Re^n$. Initially, particule locations are randomly chosen in a given interval $[x_c^{MIN}, x_c^{MAX}]$. Velocities are initialized randomly with a maximum modulus $(x_c^{MAX} - x_c^{MIN})/2$ Hence, particles can cover half of the initialization interval during the first time step. At each time step, the velocity of each particle is then updated according to some criteria:

- the particle inertia : parameter ω ;
- the best location ever found by the particle i : $(x_i^\star)_{1\leq i\leq p}$ (individual memory) ;
- the best location ever found by the swarm : $x^\star$ (collective memory).

Hence, the velocity of the particle i is defined in practice by:

$$v_i^k = \underbrace{\omega v_i^{k-1}}_{\text{term } ①} + \underbrace{c_1 r_1 (x_i^\star - x_i^k)}_{\text{term } ②} + \underbrace{c_2 r_2 (x^\star - x_i^k)}_{\text{term } ③} \quad 1 \leq i \leq p, \tag{4}$$

where r_1 and r_2 are uniformly distributed random numbers $\in [0, 1]$. The term ① corresponds to the inertia, whereas the terms ② and ③ correspond to attractions towards the individual and collective memories. The

particle location is then updated by time integration during a time step of length unity:

$$x_i^{k+1} = x_i^k + v_i^k \quad 1 \leq i \leq p. \tag{5}$$

c_1 and c_2 in (4) are called trust coefficients and are set to the value two [14]. Then, the two related attraction terms in (4) yield the particle to overfly the locations corresponding to the individual and collective memories with a probability one half every time step. This algorithm does not exactly correspond to the original one proposed by Kennedy and Eberhart [14], which did not use inertia. Moreover, velocity were updated by taking into account the best location found by the swarm at each time. The use of the inertia term was proposed by Shi and Eberhart [23], whereas the use of the best location ever found by the swarm to update the velocity was introduced by Fourie and Groenwold [10]. These works reported some improvements related to these modifications.

Moreover, these works reported the critical influence of the inertia parameter ω on the results. Indeed, for large ω values the exploration in the search space is promoted, whereas a small ω value favors a straight convergence towards the best locations ever found. A mathematical analysis of the dynamical system, including stability conditions, can be found in the work of Clerc and Kennedy [5]. Shi and Eberhart [23] proposed practically that ω value be selected such that $\omega \in [0.8, 1.4]$. They also reported improved convergence rates when ω is decreased linearly during the optimization. The present implementation is based on a dynamic update of ω proposed by Fourie and Groenwold [10]. A starting ω^0 is chosen with a quite large value in order to promote an exploratory search. Then, its value is decreased by a factor α if no improved solution is found within h consecutive time steps:

$$\text{If } \mathcal{J}(x^\star)|_k = \mathcal{J}(x^\star)|_{k-h} \text{ then } \omega^k = \alpha\, \omega^{k-1}, \tag{6}$$

with $\alpha \in]0, 1[$. Therefore, if the exploratory search fails, the convergence towards the best locations ever found is promoted. An other dynamical update is also proposed by Venter and Sobieski [24], whose criterion is based on the coefficient of variation in a subset of particles.

Some authors include in the basic algorithm a craziness operator whose role is to add random perturbations. This operator is inspired from the mutation operator in GAs. First, Kennedy and Eberhart [14] proposed to randomly perturbate the velocity of some particles, but they concluded that this operator was not mandatory. Fourie and Groenwold [10] reintroduced a craziness operator that randomly modifies the velocities with a predetermined probability and with a decreasing amplitude of perturbation. Venter and Sobieski [24] also used such an operator, but they decided to perturbate both particles location and velocity. In the current work, a craziness operator is implemented in the spirit of Fourie and Groenwold operator. Especially, a probability of craziness $p_c \in [0, 1]$ is predetermined. Then, at each time step and for each particle, the velocity direction is randomly modified with the probability p_c, but the velocity modulus is kept constant. Therefore, large random perturbations occur at the beginning of the optimization procedure, which promote random global search, whereas small random perturbations are performed when the swarm is close to the solution, which promote random local search. This approach is inspired from the so-called non-uniform mutation operator in GAs [20].

Finally, Shi and Eberhart [23] proposed to introduce a maximum velocity value v^{MAX} for stability purpose and observed an improvement of the convergence rate. This strategy is also used in the present study.

Finally, the algorithm implemented in the current study can be described as follows:

1. initialize randomly the locations $(x_i^0)_{1 \leq i \leq p}$ in the interval $[x_c^{MIN}, x_c^{MAX}]$;
   initialize the inertia ω^0 ;
   $k \leftarrow 0$;
2. begin time step k ;
3. estimate the fitness of all particles $1 \leq i \leq p$;
4. update the individual memory $(x_i^\star)_{1 \leq i \leq p}$ and the collective memory $x^\star$;
5. Check if the inertia should be updated

$$\text{If } \mathcal{J}(x^\star)|_k = \mathcal{J}(x^\star)|_{k-h} \text{ then } \omega^k = \alpha\, \omega^{k-1};$$

6. compute the velocity of the particles

$$v_i^k = \omega^k v_i^{k-1} + c_1 r_1 (x_i^\star - x_i^k) + c_2 r_2 (x^\star - x_i^k) \quad 1 \leq i \leq p;$$

7. check if the velocities are not above v^{MAX} ;

8. apply craziness with a probability p_c : random perturbation of the velocity with a constant velocity modulus

9. update the location of the particles

$$x_i^{k+1} = x_i^k + v_i^k;$$

10. if termination condition reached then stop ;
else $k \leftarrow k+1$ and goto (2);

The main weakness of such semi-stochastic optimizers is that the number of cost function evaluations increase dramatically with the dimension of the problem. This observation has motivated the current work on multi-level algorithms.

2 Multi-level parameterization algorithm

2.1 Principles

Recent works in parametric shape optimization applied to shape reconstruction [8, 15] or aerodynamic design [2, 16, 19] reported that a significant improvement of the convergence rate can be obtained using a hierarchical parameterization approach. It consists in using different shape representation levels during the optimization procedure, ranging from a coarse level parameterization including a small number of design variables to a fine level parameterization which implies a large number of design variables. Hence, the search for the optimal shape is carried out in several embedded design spaces of variable dimension. Several strategies can be considered, which are inspired from the multi-grid theory for Partial Differential Equations (PDEs) solving, ranging from simple level increase to V-cycle or Full Multi-Grid (FMG) approaches [8].

The present study is focused on the simplest approach (simple level increase), which consists in using first a coarse level parameterization and increasing progressively the number of design parameters. This approach can be easily implemented using a gradient-based optimizer. In that case, some iterations are first carried out during a first optimization step with a coarse level parameterization. Then, the best shape found is projected into the space corresponding to the finer level parameterization and a second optimization step can be performed from this new starting shape, etc. The convergence rate increase can be explained by the fact that the search in the coarse level space is quickly performed (thanks to the low dimensionality) and yields a fairly good estimate of the optimum shape that can be used as a starting guess for the search in the fine level space. Then, the resulting multi-level algorithm exhibits a higher convergence rate than that obtained using a simple search performed for the finest level only (with a large dimensionality).

It was shown in prior works [8] that a critical issue for such a strategy is related to the accuracy of the transfer of the information from a given level to the finer one. Indeed, after some iterations with a coarse level parameterization, the best shape found has to be projected into a space of higher dimension corresponding to the finer level parameterization. If this projection modifies the shape characteristics too much, the starting shape for the second optimization step can exhibit a poor fitness, resulting in an useless first optimization step. Hopefully, the Bernstein polynomials used for the FFD technique allow an exact information transfer from a given level to the finer one, thanks to the *degree-elevation property* [9]. Indeed, any polynomial of degree n is also a polynomial of degree $n+1$. Then, any deformation using the FFD technique ruled by the Bernstein polynomials basis of degree n along one direction can be expressed as another deformation ruled by the Bernstein polynomials basis of degree $n+1$. Consider the following deformation of degree n_i along direction ξ defined by the weighting coefficients $(\Delta P_i)_{0 \leq i \leq n_i}$:

$$\Delta q = \sum_{i=0}^{n_i} B_i^{n_i}(\xi) \Delta P_i. \tag{7}$$

Then, the same deformation can be obtained using the Bernstein polynomials basis of degree n_i+1 by considering the weighting coefficient $(\Delta P'_i)_{0\leq i\leq n_i+1}$ defined as follows [9]:

$$\begin{aligned} \Delta P'_0 &= \Delta P_0; \\ \Delta P'_i &= \frac{i}{n_i+1}\Delta P_{i-1} + (1-\frac{i}{n_i+1})\Delta P_i \quad 1\leq i\leq n_i; \\ \Delta P'_{n_i+1} &= \Delta P_{n_i}. \end{aligned} \tag{8}$$

The extension to the three-dimensional FFD technique is straightforward. Hence, this construction guarantees rigorously *nested search spaces* and *exact upward transfer operators*.

Finally, the multi-level parameterization approach used within a gradient-based framework in prior works can be described as follows:

1. initialize design variables $(\Delta P^{(0)}_{ijk}|_0)_{0\leq i\leq n^0_i,0\leq j\leq n^0_j,0\leq k\leq n^0_k}$ for level 0 ;
   $l \leftarrow 0$;
2. begin search for level l ;
3. perform k^l gradient-based optimization iterations yielding the shape defined by $(\Delta P^{(k^l)}_{ijk}|_l)_{0\leq i\leq n^l_i,0\leq j\leq n^l_j,0\leq k\leq n^l_k}$;
4. use de degree-elevation process to determine a starting shape on the finer level $l+1$ defined by $(\Delta P^{(0)}_{ijk}|_{l+1})_{0\leq i\leq n^{l+1}_i,0\leq j\leq n^{l+1}_j,0\leq k\leq n^{l+1}_k}$;
5. end of the search for level l ;
   $l \leftarrow l+1$;
   if finest level reached then stop ;
   else $l \leftarrow l+1$ and goto (2).

This algorithm provided promising results [2, 8, 15, 16] for different applications, but using a gradient-based optimization method. In the next section, the use of such a multi-level strategy for semi-stochastic optimizers is examined.

2.2 Multi-level approach and stochastic optimization

The use of the above described multi-level strategy for semi-stochastic optimizers is not straightforward. Indeed, the above algorithm relies on two main principles:

- the optimization path is split in several optimization steps that are carried out in spaces of increasing dimension ;
- the information collected from a given level is transfered into the finer level as a starting shape.

According to the well-established status of GAs and to our experience in GAs based optimization [17, 18], the use of GAs for the multi-level approach with a semi-stochastic optimizer was first envisaged. However, one can notice that none of the two above mentioned principles can be verified using GAs. Indeed, GAs do not follow an optimization path in the search space but use samples to construct the optimum by combination of potential solutions. Moreover, semi-stochastic optimizers such as GAs do not have a starting point but use a random initialization of the samples.

Nevertheless, some tests were carried out [2] using GAs and a multi-level parameterization inspired from the described above approach. In these tests, the samples (individuals of the population) are randomly initialized and then evolve during a given number of iterations (generations) for a coarse parameterization level. Then, the best individual found is transfered into the space corresponding to a fine parameterization, whereas all other individuals are randomly re-initialized in this space of higher dimension. Poor results were reported, since one single individual is not able to influence the evolution of the whole population due to the recombination process. A second strategy was then tested, by transferring all individuals into the finer level parameterization. In that case, the lack of random initialization yield a poor exploration of the space of higher dimension. Finally, authors conclude that GAs are not well suited to the multi-level parameterization approach, due to the difficulty to build an efficient strategy to transfer the information from a given level to the finer one.

Hopefully, PSO algorithm offers an alternative search approach, that relies on different principles, which can be used in an efficient manner to build a multi-level PSO algorithm.

2.3 Multi-level Particle Swarm Optimization Algorithm (MPSOA)

Contrary to GAs which employ *sampling* and *combination* to evolve, PSO algorithm uses *particles displacement* and *memory* to progress towards the solution. These characteristics constitute a nice framework to build a semi-stochastic multi-level optimization method.

The core of the proposed approach consists in using the best shape found in the space corresponding to a given parameterization level as the initial collective memory for the search in the space of higher dimension, whereas the particle locations are randomly re-initialized. Since the collective memory is an initial guess of the optimum shape, it acts as an attractor towards a region of the space of higher dimension where high fitness shapes are expected. Moreover, the random initialization promotes the global search in the space corresponding to the finer parameterization. Hence, this strategy yields *a balance between the global search in the space of high dimension and the use of prior searches within subspaces of lower dimension.* Using such a strategy, the role of the search for a given level is to determine a region in the space where the optimum shape is expected, that can be used to lead and speed up the search for the finer level.

An improvement is also proposed, which is related to the way the particle locations are initialized. Indeed, the interval $[x_c^{MIN}, x_c^{MAX}]$ used for the random initialization can be modified to benefit from the search performed for the coarser level parameterization. Since the optimum shape is expected to be found in a region of the space of higher dimension close to the initial collective memory, it is proposed to define a new initialization interval, whose center corresponds to the initial collective memory and whose extent is smaller than that of the coarser level. Then, the search in regions of the space where low fitness shapes are expected is avoided.

Finally, the proposed Multi-level Particle Swarm Optimization Algorithm (MPSOA) can be described as follows:

1. begin the search for level l ;
2. initialize randomly the locations $(x_i^0)_{1\leq i\leq p}$ and velocities $(v_i^0)_{1\leq i\leq p}$ in the interval $[x_c^{MIN}|_l, x_c^{MAX}|_l]$;
   initialize the inertia ω^0 ;
   $k \leftarrow 0$;
3. begin time step k ;
4. estimate the fitness of all particles $1 \leq i \leq p$;
5. update the individual memories $(x_i^\star)_{1\leq i\leq p}$ and the collective memory $x^\star|_l$;
6. Check if the inertia should be updated
$$\text{If } \mathcal{J}(x^\star|_l)|_k = \mathcal{J}(x^\star|_l)|_{k-h} \text{ then } \omega^k = \alpha\,\omega^{k-1};$$
7. compute the velocity of the particles
$$v_i^k = \omega^k v_i^{k-1} + c_1 r_1 (x_i^\star - x_i^k) + c_2 r_2 (x^\star|_l - x_i^k) \quad 1 \leq i \leq p;$$
8. check if the velocities are not above v^{MAX} ;
9. apply craziness with a probability p_c : random perturbation of the velocity with a constant velocity modulus ;
10. update the location of the particles
$$x_i^{k+1} = x_i^k + v_i^k;$$
11. if termination condition reached then goto (12) ;
    else $k \leftarrow k+1$ and goto (3);
12. end of the search for level l ;
13. use de degree-elevation process from $x^\star|_l$ to determine the initial collective memory $x^\star|_{l+1}$ for the finer level $l+1$;

14. determine the new initialization interval $[x_c^{MIN}|_{l+1}, x_c^{MAX}|_{l+1}]$ such that

$$x^\star|_{l+1} = \frac{1}{2}(x_c^{MIN}|_{l+1} + x_c^{MAX}|_{l+1}),$$

and

$$x_c^{MAX}|_{l+1} - x_c^{MIN}|_{l+1} = \beta(x_c^{MAX}|_l - x_c^{MIN}|_l);$$

if finest level reached then stop ;
else $l \leftarrow l+1$ and goto (1).

A critical parameter to enhance the overall convergence rate of the procedure is the termination criterion of each optimization step, which determines when the transfer into a space of higher dimension should be carried out. One conjectures that a full convergence is not mandatory for the coarsest levels. However, the search in the space corresponding to a coarse parameterization should be performed during a number of time steps sufficient to ensure that valuable information is obtained for this level and is useful for the finer level search. Since the role of the search on a given level is to determine a region in the space where the optimum shape is expected, the search should be stopped as soon as a promising region is detected. Therefore, a criterion based on the location of the individual memories is proposed. The search for a given level is stopped as soon as all individual memories are localized into a fraction of the initialization interval. If the following condition stands for all components of the vector of design variables:

$$\max_{1\le i\le p} x_i^\star - \min_{1\le i\le p} x_i^\star \le \gamma(x_c^{MAX}|_l - x_c^{MIN}|_l), \tag{9}$$

then the search on the level l is stopped. This condition means that, whatever the current location of the particles, the best shape found by each particle is localized within a particular region in the space. Moreover, this approach allows to control the extent of this region and the computational effort required, by adjusting the numerical parameter γ.

One should underline that the above described MPSOA is essentially different than the multi-level algorithm using a gradient-based optimizer. In the later case, the search for a coarser level provides a *starting* shape for the search in the space of higher dimension. In the former case, the search on a coarser level provides an *attractor* towards a region where high fitness shapes are expected. Although both algorithms rely on the same multi-level parameterization representation, their underlying mechanisms are totally different.

The behavior of the above described MPSOA depends on some numerical parameters, that are reminded in the following table:

T^{MAX}	max number of time steps
p	number of particles
ω^0	initial inertia
h	inertia reduction criterion
α	inertia reduction rate
c_1,c_2	trust coefficients
p_c	craziness probability
v^{MAX}	maximum velocity
β	interval initialization coefficient
γ	termination criterion

For most of them, the literature provides guidelines to determine satisfactory values. Some of them are problem dependent, then some numerical experiments should be carried out to determine suitable values. In the next sections, two shape optimization problems in different fields are studied, in order to demonstrate the capability of the MPSOA and quantify the influence of the main parameters for the multi-level approach.

3 Application in electromagnetics

An application of the MPSOA to electromagnetics is presented in this section. The electromagnetic (EM) field is assumed to be time-harmonic dependent.

A classical problem of EM waves propagation is to conform reflector antennas w.r.t. radiation diagrams. A radiation diagram describes the electric field (or transmitted power) in some domain Ω. Let $\vec{E}^t$ denote a target or ideal electric field in Ω. The conformation problem consists in finding the reflector geometry $\mathcal{S}$ whose radiation corresponds to the ideal field. Thus, a natural functional is to measure the discrepancy between $\vec{E}(\mathcal{S})$ and $\vec{E}^t$ in Ω as follows:

$$\mathcal{J}(\mathcal{S}) = \frac{1}{2}\int_\Omega \|\vec{E}(\mathcal{S}) - \vec{E}^t\|^2 d\Omega \tag{10}$$

where $\|.\|$ is a norm on a complex space since the electric field is complex valued.

This kind of problem, known as inverse scattering, is highly multimodal. Hence, gradient-based methods are easily trapped into local minima. In other words, without any good *a priori* initial shape, such methods will fail. The following subsections are devoted to show that the MPSOA improves robustness and convergence in shape optimization using the *FFD* technique described in section 1.2 for minimizing the functional (10). Namely, considering the space of feasible deformations $\mathbf{X}$, which is determined by the degree of parameterization, and a shape $\mathcal{S}^0$, the problem reads:

$$\min_{\Delta\mathcal{S}\in\mathbf{X}} \mathcal{J}(\mathcal{S}^0 + \Delta\mathcal{S}). \tag{11}$$

When Ω is far enough from the reflector, A set of directions $\{(\theta,\varphi)\}$ (resp. polar angle and azimuth) is considered. In this case the radiation diagram depends on the so-called *far field* $\vec{E}$ (see [6] for more details about EM waves and Antennas).

3.1 Test-case description

The geometry is assumed to be axisymmetric around the Oz axis. Thus the problem reduces to a 2D problem in the xOz plan. The geometry is a curve, also called meridian. Moreover only deformations in the z direction will be considered. The source of EM waves is an elementary dipole and is located at the origin of the system of coordinates. Let the target field be the *far field* computed from a known geometry $\mathcal{S}^t$, that is, $\vec{E}^t = \vec{E}(\mathcal{S}^t)$, for the directions $\Omega = \{(\theta,\varphi) \in [0,\pi]\times\{0,\frac{\pi}{2}\}\}$ (see Figure 2). The initial shape $\mathcal{S}^0$ is a *Free-Form Deformation* of $\mathcal{S}^t$ for some parameter x with degree $n_x = 15$, $n_z = 0$ (see Figure 3). Trivially, there exists a solution for this problem.

Experimentations are conducted to investigate the benefits of the MPSOA compared to a simple PSO algorithm. A fine tuning of basic parameters, which is problem dependent, can improve the results, but this improvement is not related to the hierarchical property. Therefore these parameters will remain constant for all tests so that they will not interfere with the properties of the multilevel algorithm (see Table 1).

Parameter	Value
w_0	1.1
h	3
α	0.98
c_1	2
c_2	2
p_c	0
v^{MAX}	0.04

Table 1: basic parameters of PSO.

3.2 Parameterization

The *FFD* lattice is constructed around the initial meridian as depicted in Figure 3. Three levels are considered. More precisely, the considered degrees for all experimentations are in Table 2. Note that the finest level is chosen to correspond to the degree used to define $\mathcal{S}^0$ from $\mathcal{S}^t$, which ensures that the target shape is reachable. In addition the reflector has to be fixed to the symmetry axis in order to preserve the distance to the source. This means that for all degrees, the basis function $B_0^{n_x}$ should not be used, hence let $x_0 = 0$.

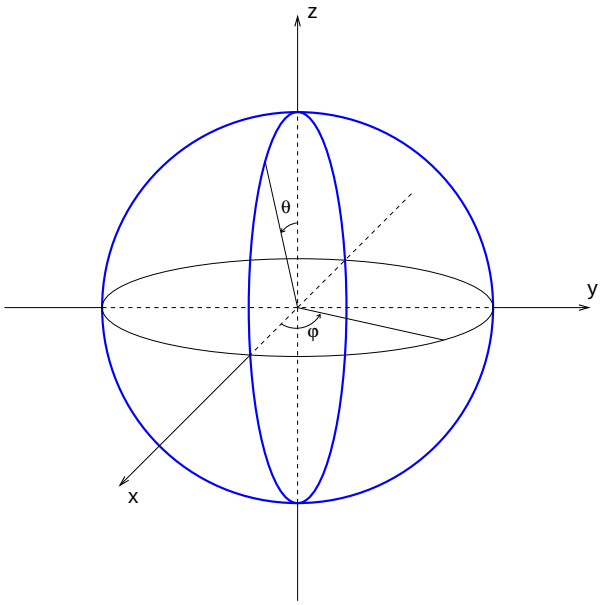


Figure 2: set of directions Ω where the field is computed, i.e. $\varphi = 0$ and $\varphi = \pi$ planes (bold circles).

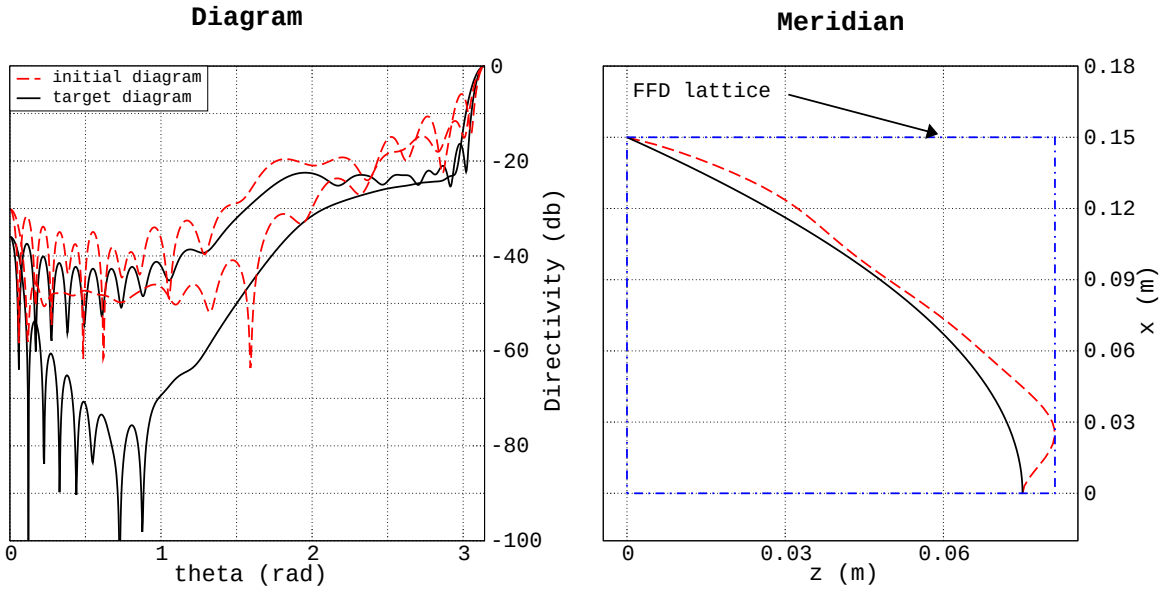


Figure 3: diagrams and shapes of initial (dotted line) and target (plain line) configurations; the reference of the decibel scale is the maximum value of the electromagnetic power (here for $\theta{=}\pi$).

	n_x	n_z
coarse	4	0
medium	8	0
fine	15	0

Table 2: parameterization degrees.

3.3 Fitness evaluation

The electric field for the fitness evaluation is computed according to an explicit function of the geometry. This function is given by a simple model known as *Physical Optics* which is accurate for the considered configurations. In this way, heavy computations are avoided.

3.4 Results

3.4.1 Role of memory, influence of contraction coefficient β

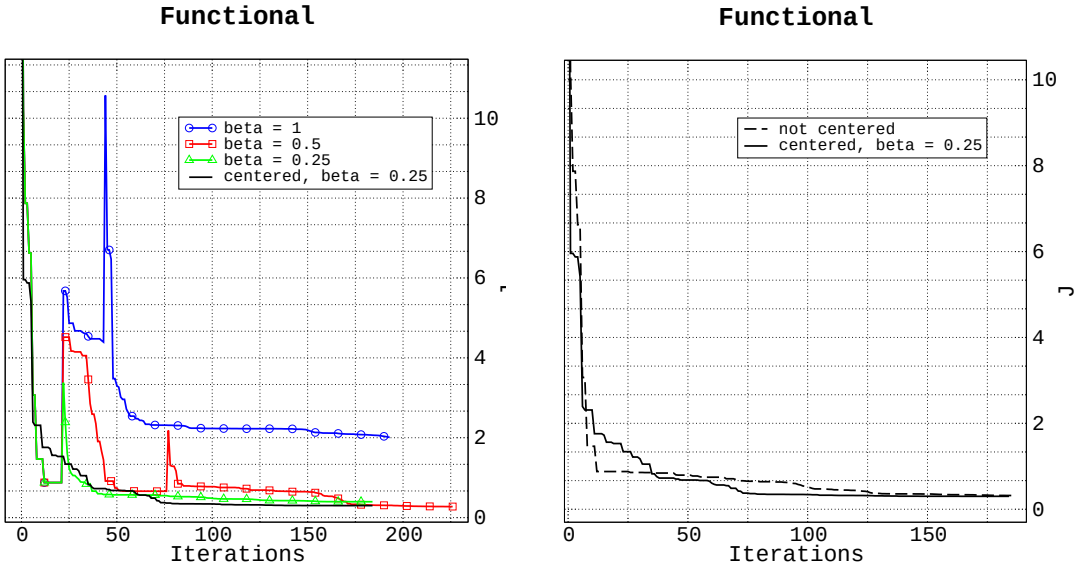
In the MPSOA described in section 2.3 one can see that the global memory has two roles. First, the best particle at convergence on a coarse level is kept as global memory for the next level. Then the new initialization domain is centered around this point. The following tests are run in order to understand each role independently:

1. the global memory is reset to the null parameters vector $\mathbf{0}$ at each level; the initialization domain is centered around the solution of previous level and the radii are reduced by coefficient β for values 1.0, 0.5 and 0.25
2. the best particle is kept on the next level as global memory by degree elevation; the initialization domain remains the same, centered around $\mathbf{0}$
3. (control experiment) the best particle is kept on the next level as global memory by degree elevation; the initialization domain is centered around the solution of previous level and the radii are reduced by coefficient $\beta = 0.25$

Results of experiments 1 and 2 are compared with the control experiment 3 in Figure 4. It appears that both roles are important. Without any change in the initialization domain, one can see that after few steps the swarm is highly influenced by the global memory. That is, when no better solution is found, the search domain is naturally reduced.

When the domain is just centered without any reduction nor global memory conservation, the algorithm fails. However, when the domain reduction is sufficient, it seems to act as a global memory, though there is an obvious transition after each degree elevation.

Finally the control experiment is the most efficient since the domain reduction helps to converge faster.



(a) experiment 1

(b) experiment 2

Figure 4: roles of memory, 30 particles, $\gamma = 0.5$, $T^{MAX} = 150$.

3.4.2 Single-level parameterization

In order to compare with the multilevel algorithm, results using basic PSO are presented. Several runs with different random numbers have been considered in order to test the robustness w.r.t. the inherent stochastic behavior of PSO. However the sample size, $N = 6$, is small, because one experiment lasts from 6 to 10 hours, depending on the number of particles. Table 3 summarizes single level results for some statistical values (best, mean and median values obtained, standard deviation), for different numbers of particles.

Nb of particules	$\mathcal{J}_{best}$	Mean	Median	Standard Deviation	Nb of evaluations
40	$0.142047 \cdot 10^{-1}$	0.734330	0.717097	0.611958	8000
50	$0.123409 \cdot 10^{-1}$	0.852612	0.713756	0.870882	10000
80	$0.446563 \cdot 10^{-3}$	0.294568	0.132269	0.419132	16000

Table 3: single level, influence of p, $T^{MAX} = 200$.

As can be observed, the best and the median values obtained for the different runs decrease as the number of particles is increased, as expected. However, one can notice that the mean value using 50 particules is slightly higher than that using 40 particules. This surprising result is due to the use of a too small sample for statistical estimations. Actually, a bad result obtained for one run using 50 particules degrades significantly the mean and increases the standard deviation.

3.4.3 Multi-level parameterization

For the multi-level experimentations, according to section 3.4.1, let $\beta = 0.25$. The results are presented under three different formats:

- in Table 4 numerical indicators are shown
- in Figure 5 functional values for the best run of each sample (single and multilevel) are represented
- in Figure 6 a synthesis of the results is proposed: one point corresponds to the final value of the functional for each run and a line represents the median value for each sample; hence the whole sample is represented and can be compared with the median value

It appears that the multi-level algorithm is globally better than the single one. First, the best functional value is always smaller, although the improvement is not always significant (see for 80 particles). However, the mean and the median values of the functional are clearly much smaller. This means that the MPSOA is not trapped as many times as the basic PSO algorithm into local minima. In addition, the standard deviation value is smaller, indicating that the MPSOA is more stable. This observations are represented graphically in Figure 6.

It is important to notice that this results are obtained with the same computational cost than with the single parameterization case. In fact the number of iterations over the first two levels does not exceed 50. Thus the global number iterations is about 200.

Nb of particles	$\mathcal{J}_{best}$	Mean	Median	Standard Deviation	Nb of evaluations (best)
40	$0.203276 \cdot 10^{-2}$	0.256483	0.133515	0.320331	7440
50	$0.234657 \cdot 10^{-2}$	0.311201	0.344843	0.277711	9550
80	$0.301865 \cdot 10^{-3}$	0.097700	0.00820857	0.145369	15920

Table 4: multi-level, influence of p, $\gamma = 0.5$, $T^{MAX} = 150$.

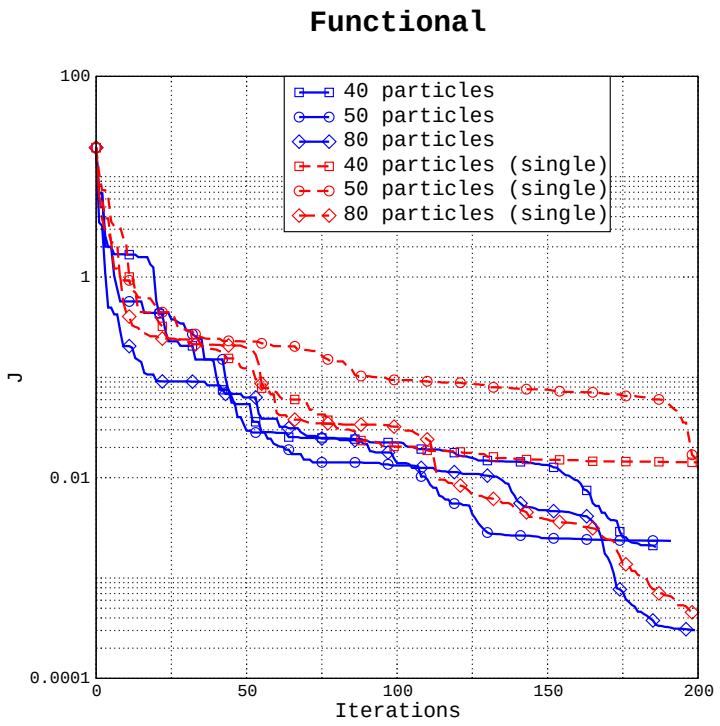


Figure 5: single and multi-level best functionals (logarithmic scale).

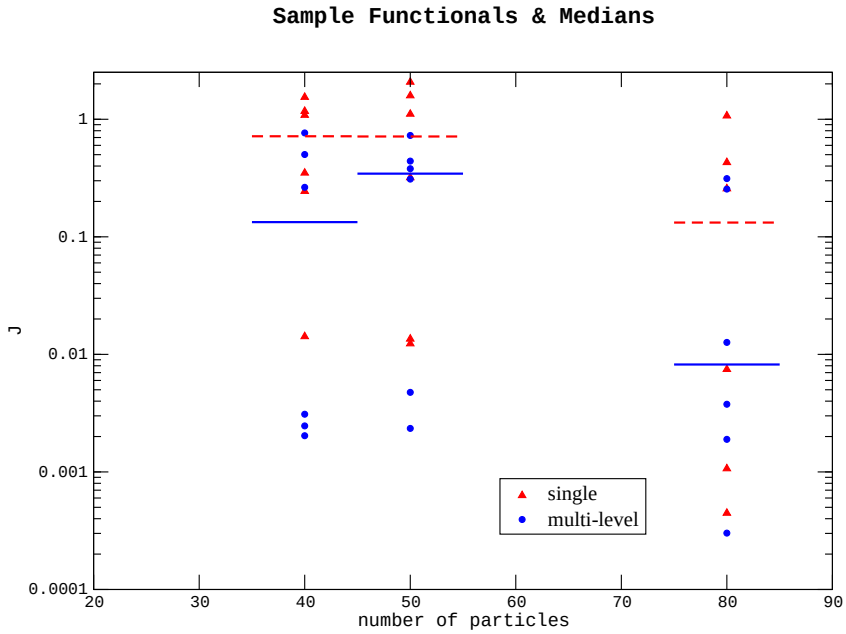


Figure 6: sample representation; each point correponds to a single run; the line represents the median value of each sample.

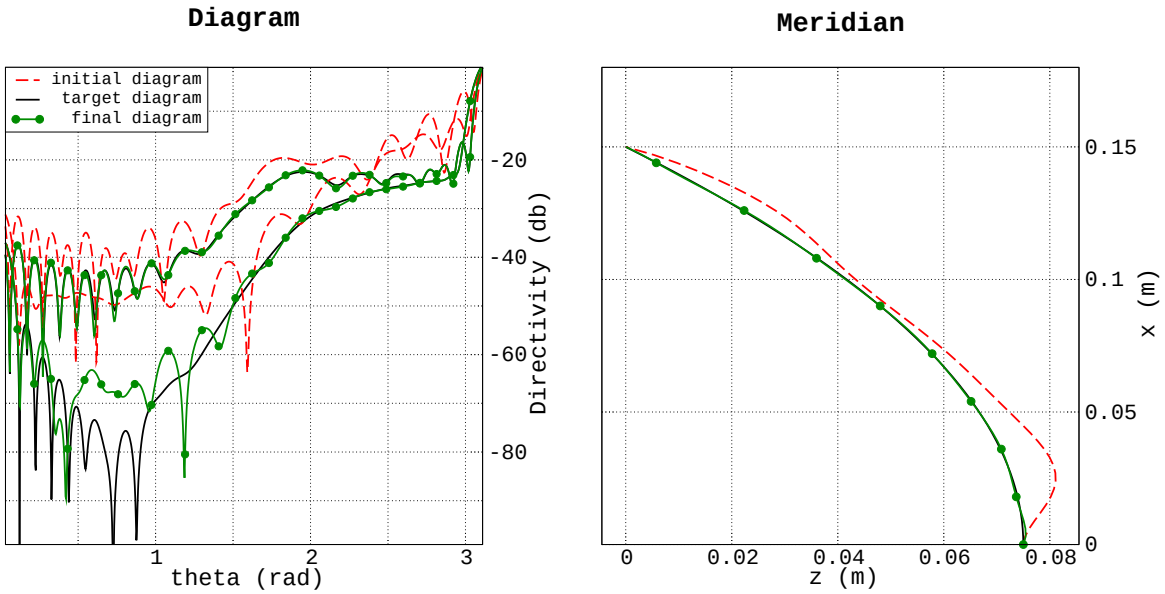


Figure 7: final diagram; the target shape and the final shape are superimposed; the difference between the target and the final diagram for $\theta \in [0.5\,1.5]$ is negligible since the power is smaller than 50 db in these directions.

4 Application in aerodynamics

The proposed MPSOA is now faced with a realistic shape optimization problem in aerodynamics. The following sections describe the test-case and the numerical methods involved in the aerodynamic fitness evaluation.

4.1 Test-case description

The test-case considered here corresponds to the optimization of the shape of the wing of a business aircraft (courtesy of Piaggio Aero Ind.), for a transonic regime. The test-case is described in depth in [2]. The overall wing shape can be seen in figure (8). The free-stream Mach number is $M_\infty = 0.83$ and the incidence $\alpha = 2°$. Initially, the wing section is supposed to correspond to the NACA 0012 airfoil.

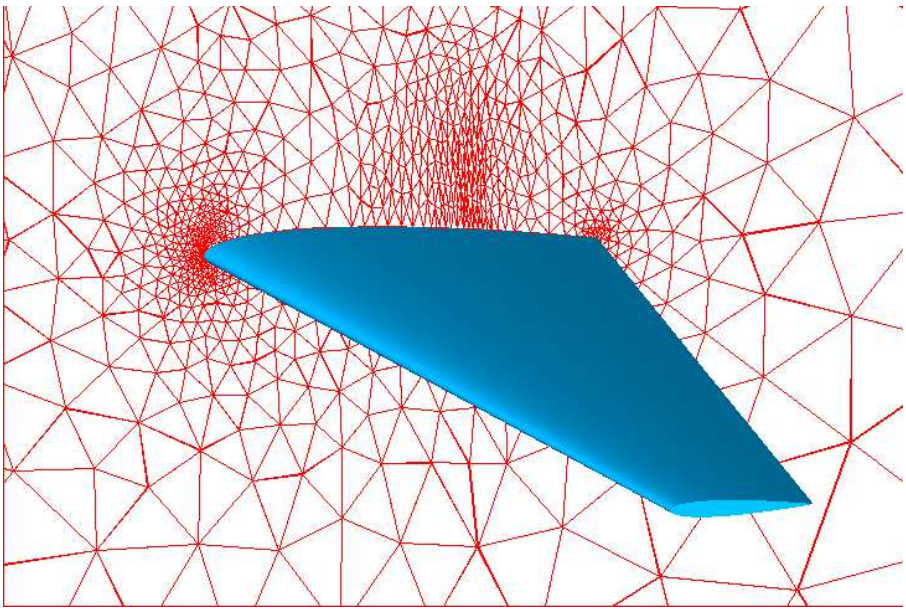


Figure 8: Initial wing shape (blue) and mesh in the symmetry plane (red).

The goal of the optimization is to reduce the drag coefficient C_D subject to the constraint that the lift coefficient C_L should not decrease more than 0.1%. The constraint is taken into account using a penalization approach. Then, the resulting cost function is :

$$\mathcal{J}_{OPT} = \frac{C_D}{C_{D0}} + 10^4 \max(0, 0.999 - \frac{C_L}{C_{L0}}). \tag{12}$$

C_{D0} and C_{L0} are respectively the drag and lift coefficients corresponding to the initial shape (NACA 0012 section).

4.2 Parameterization

The FFD lattice is built around the wing with ξ, η and ζ in the chordwise, spanwise and thickness directions respectively. The lattice is chosen in order to fit the planform of the wing (see figure 1). Then, the leading and trailing edges are kept fixed during the optimization by freezing the control points that correspond to $i = 0$ and $i = n_i$. Moreover, control points are only moved vertically. Three parameterization levels are under consideration. The coarsest one corresponds to $n_i = 3$, $n_j = 1$ and $n_k = 1$. Therefore, $(4-2) \times 2 \times 2 = 8$ degrees of freedom are taken into account in the optimization. The medium parameterization corresponds to $n_i = 6$, $n_j = 1$ and $n_k = 1$ and counts $(7-2) \times 2 \times 2 = 20$ degrees of freedom. Finally, the finest parameterization corresponds to $n_i = 9$, $n_j = 1$ and $n_k = 1$ and counts $(10-2) \times 2 \times 2 = 32$ degrees of freedom.

4.3 Aerodynamic fitness evaluation

Modeling This study is restricted to three-dimensional inviscid compressible flows governed by the Euler equations. Then, the state equations can be written in the conservative form :

$$\frac{\partial W}{\partial t} + \frac{\partial F_1(W)}{\partial x} + \frac{\partial F_2(W)}{\partial y} + \frac{\partial F_3(W)}{\partial z} = 0, \tag{13}$$

where W are the conservative flow variables $(\rho, \rho u, \rho v, \rho w, E)$, with ρ the density, $\overrightarrow{U} = (u, v, w)$ the velocity vector and E the total energy per unit of volume. $\overrightarrow{F} = (F_1(W), F_2(W), F_3(W))$ is the vector of the convective fluxes, whose components are given by :

$$F_1(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix} \quad F_2(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix} \quad F_3(W) = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{pmatrix}. \tag{14}$$

The pressure p is obtained from the perfect gas state equation :

$$p = (\gamma - 1)(E - \frac{1}{2}\rho \|\overrightarrow{U}\|^2), \tag{15}$$

where $\gamma = 1.4$ is the ratio of the specific heat coefficients.

Spatial discretization An unstructured mesh, composed of 31124 nodes and 173 445 tetrahedral elements, is generated around the wing, including a refined area in the vicinity of the shock (figure (8)). Provided that the flow domain Ω is discretized by a tetrahedrization $\mathcal{T}_h$, a discretization of equation (13) at the mesh node s_i is obtained by integrating (13) over the volume C_i, that is built around the node s_i by joining barycenters of the tetrahedra and triangles containing s_i and midpoints of the edges adjacent to s_i :

$$Vol_i \frac{\partial W_i}{\partial t} + \sum_{j \in N(i)} \Phi(W_i, W_j, \overrightarrow{\sigma}_{ij}) = 0, \tag{16}$$

where W_i represents the cell averaged state and Vol_i the volume of the cell C_i. $N(i)$ is the set of the neighboring nodes. $\Phi(W_i, W_j, \overrightarrow{\sigma}_{ij})$ is an approximation of the integral of the fluxes (14) over the boundary ∂C_{ij} between C_i and C_j, which depends on W_i, W_j and $\overrightarrow{\sigma}_{ij}$ the integral of a unit normal vector over ∂C_{ij}. These numerical fluxes are evaluated using upwinding, according to the approximate Riemann solver of Roe :

$$\Phi(W_i, W_j, \overrightarrow{\sigma}_{ij}) = \frac{\overrightarrow{F}(W_i) + \overrightarrow{F}(W_j)}{2} \cdot \overrightarrow{\sigma}_{ij} - |A_R(W_i, W_j, \overrightarrow{\sigma}_{ij})| \frac{W_j - W_i}{2}. \tag{17}$$

A_R is the jacobian matrix of the fluxes for the Roe average state and verifies:

$$A_R(W_i, W_j, \overrightarrow{\sigma}_{ij})(W_j - W_i) = (\overrightarrow{F}(W_j) - \overrightarrow{F}(W_i)) \cdot \overrightarrow{\sigma}_{ij}. \tag{18}$$

A high order scheme is obtained by interpolating linearly the physical variables from s_i to the midpoint of $[s_i s_j]$, before equation (16) is employed to evaluate the fluxes. Nodal gradients are obtained from a weighting average of the P1 Galerkin gradients computed on each tetrahedron containing s_i. In order to avoid spurious oscillations of the solution in the vicinity of the shock, a slope limitation procedure using the Van-Albada limiter is introduced. The resulting discretization scheme exhibits a third order accuracy in the regions where the solution is regular.

Time integration A first order implicit backward scheme is employed for the time integration of (16), which yields :

$$\frac{Vol_i}{\Delta t} \delta W_i + \sum_{j \in N(i)} \Phi(W_i^{n+1}, W_j^{n+1}, \overrightarrow{\sigma}_{ij}) = 0, \tag{19}$$

with $\delta W_i = W_i^{n+1} - W_i^n$. Then, the linearization of the numerical fluxes provides the following integration scheme :

$$\left(\frac{Vol_i}{\Delta t} + J_i^n\right) \delta W_i = - \sum_{j \in N(i)} \Phi(W_i^n, W_j^n, \overrightarrow{\sigma}_{ij}). \tag{20}$$

Here, J_i^n is the jacobian matrix of the first order numerical fluxes, whereas the right hand side of (20) is evaluated using high order approximations. The resulting integration scheme provides a high order solution of the problem. More details can be found in [7].

4.4 Results

4.4.1 Single-level parameterization

The baseline PSO algorithm is first used to carry out some optimization exercises for single-level parameterizations of increasing degrees. Numerical parameters are set using results from the literature and prior tests:

$T^{MAX} = 200$	max number of time steps
$\omega^0 = 1.2$	initial inertia
$h = 3$	inertia reduction criterion
$\alpha = 0.98$	inertia reduction rate
$c_1 = c_2 = 2$	trust coefficients
$p_c = 0.05$	craziness probability
$v^{MAX} = 0.25\ (x_c^{MAX} - x_c^{MIN})$	maximum velocity

These values are kept fixed during the whole computational campaign. First, the baseline PSO algorithm using $p = 30$ particles is tested for the three parameterizations described above. Results obtained are sumarized in the following table:

Parameterization	Best cost function value	Nb of evaluations required
coarse (8 d.o.f.)	0.542	6000
medium (20 d.o.f.)	0.513	6000
fine (32 d.o.f.)	0.662	6000

The evolutions of the cost function is depicted in figure (9). As can be seen, the results obtained using the medium parameterization are better than those obtained using the coarse one. This behavior was expected since the number of design variables is increased. However, the best fitness reached using the fine parameterization is far worse than the one obtained for coarser parameterizations. This unexpected behavior is due to the fact that for the fine parameterization too few particles ($p = 30$) are considered for a search in a space of high dimension ($n = 32$).

Some additional optimization exercises are then carried out for the fine parameterization, by considering increasing numbers of particles. Results are provided in the following table:

Nb of particles	Best cost function value	Nb of evaluations required
30	0.662	6000
60	0.537	12000
120	0.473	24000
240	0.449	48000

The evolutions of the cost function are depicted in figure (10). Increasing the number of particles results in the improvement of the convergence rate and the increase of the fitness reached. Nevertheless, the computational cost increases linearly (if the number of time steps is maintained). This optimization exercise puts in light the difficulty to use a semi-stochastic optimizer when the dimension of the design space becomes quite high.

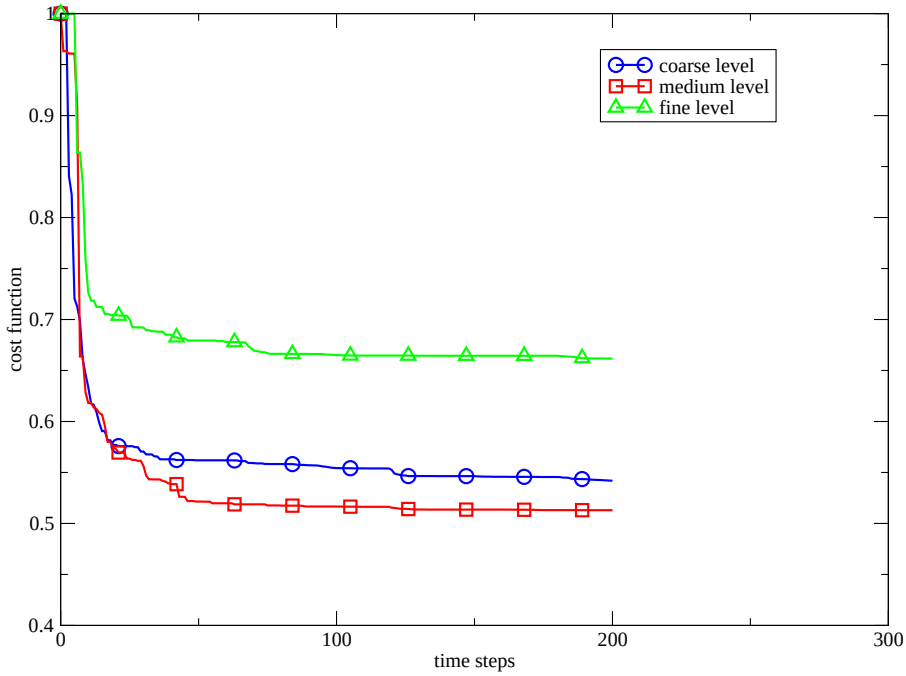


Figure 9: Evolutions of the cost function for the three single-level parameterizations (30 particles).

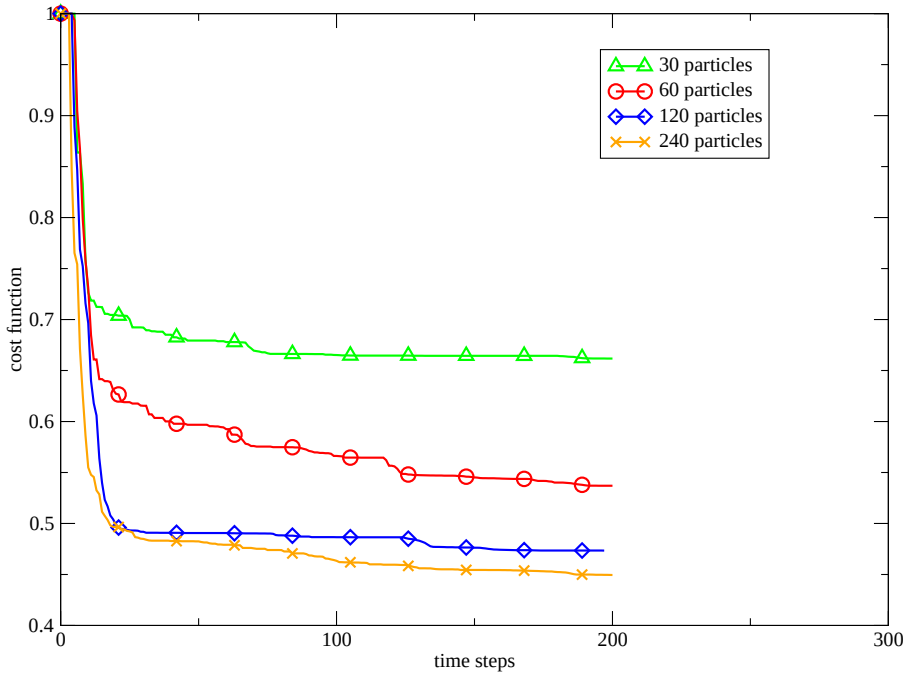


Figure 10: Evolutions of the cost function for the fine parameterization (increasing number of particles).

4.4.2 Multi-level parameterization

The MPSOA efficiency is tested using the numerical parameters provided above for a swarm including $p = 30$ particles. For the parameters γ and β that rule the multi-level strategy, two values are tested. The results for the four resulting calculations are provided in the next table:

Parameters	Best cost function value	Nb of evaluations required
$\gamma = 0.5$ $\beta = 0.5$	0.459	6630
$\gamma = 0.25$ $\beta = 0.5$	0.458	8550
$\gamma = 0.5$ $\beta = 0.25$	0.458	7110
$\gamma = 0.25$ $\beta = 0.25$	0.444	10890

The evolution of the cost function can be seen on figure (11). As observed, these four calculations provide solutions of satisfactory fitness for the final fine parameterization, although the number of particles is rather low. Hence, the multi-level strategy allows the swarm to perform an efficient search with a moderate number of particles. The result obtained using the parameters $\gamma = 0.25$ $\beta = 0.25$ is slightly better than the others. This indicates that it is more beneficial to perform a quite long search on a given level and then reduce significantly the search domain for the upper level.

If these results are compared to those obtained with the single-level parameterization, one can conclude that the multi-level approach allows to reach shapes of similar fitness for a computational cost five times lower (figure 12).

Figures (12) shows a comparison of the best shapes obtained for the single-level parameterization and the multi-level parameterization. As can be observed, shapes are very close to each other for the suction side, which is the most sensitive part of the shape. However, some discrepancies exist, that underline the multimodality of the cost function. Obviously, physical fields exhibit also some discrepancies (see figures (14) to (17)).

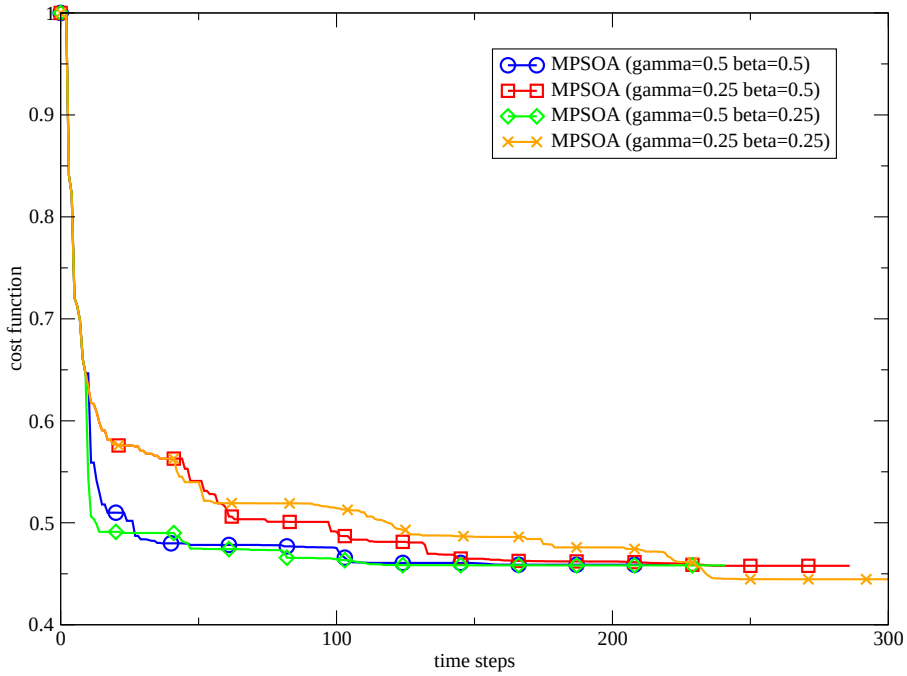


Figure 11: Evolutions of the cost function for the MPSOA (30 particles).

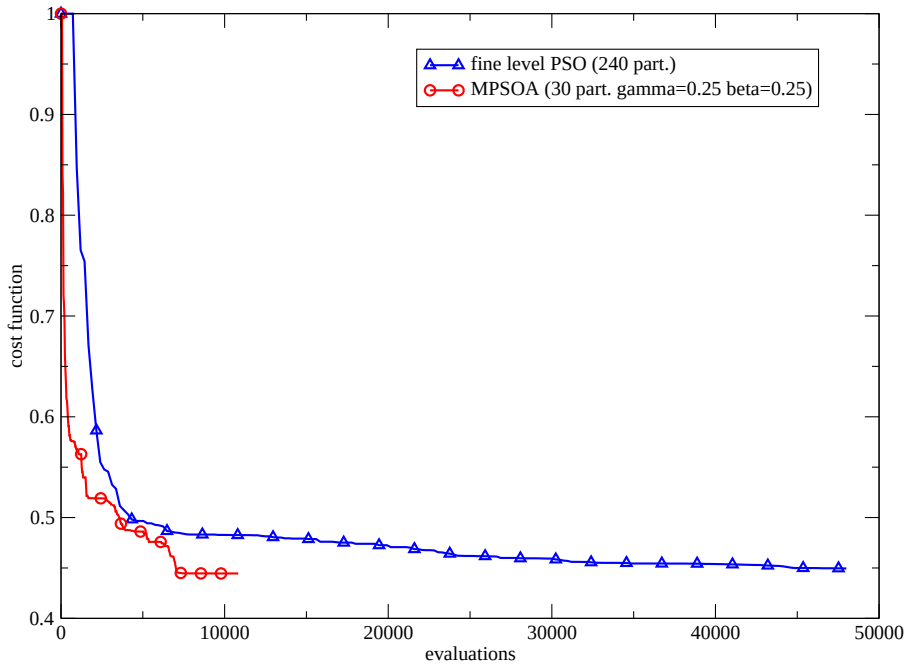


Figure 12: Evolutions of the cost function for the MPSOA (30 particles) and the basic PSO (240 particules), in terms of evaluations.

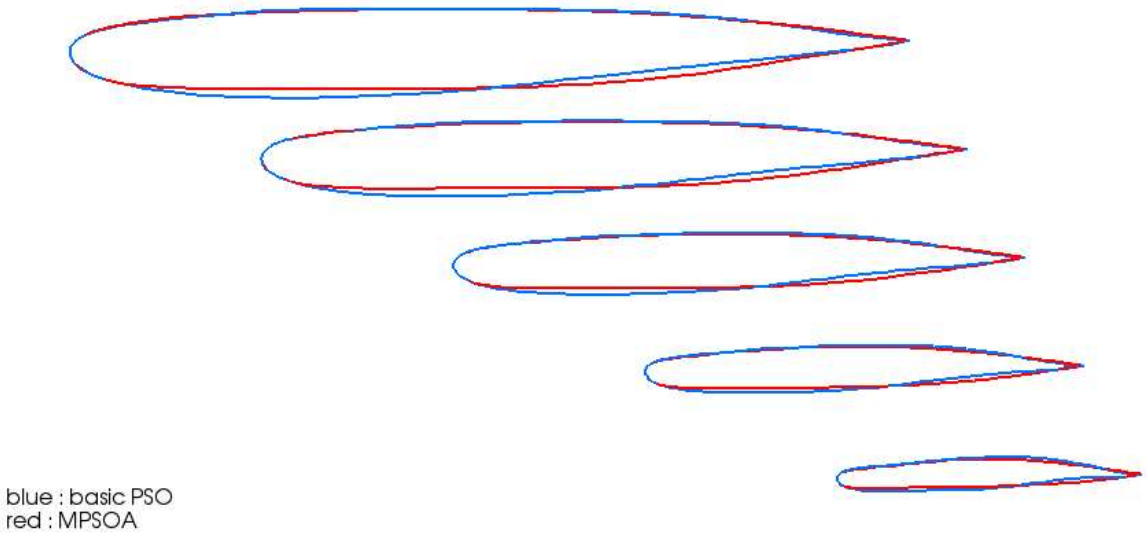


Figure 13: Comparison of the best shapes obtained for the MPSOA (30 particles) and the basic PSO (240 particules).

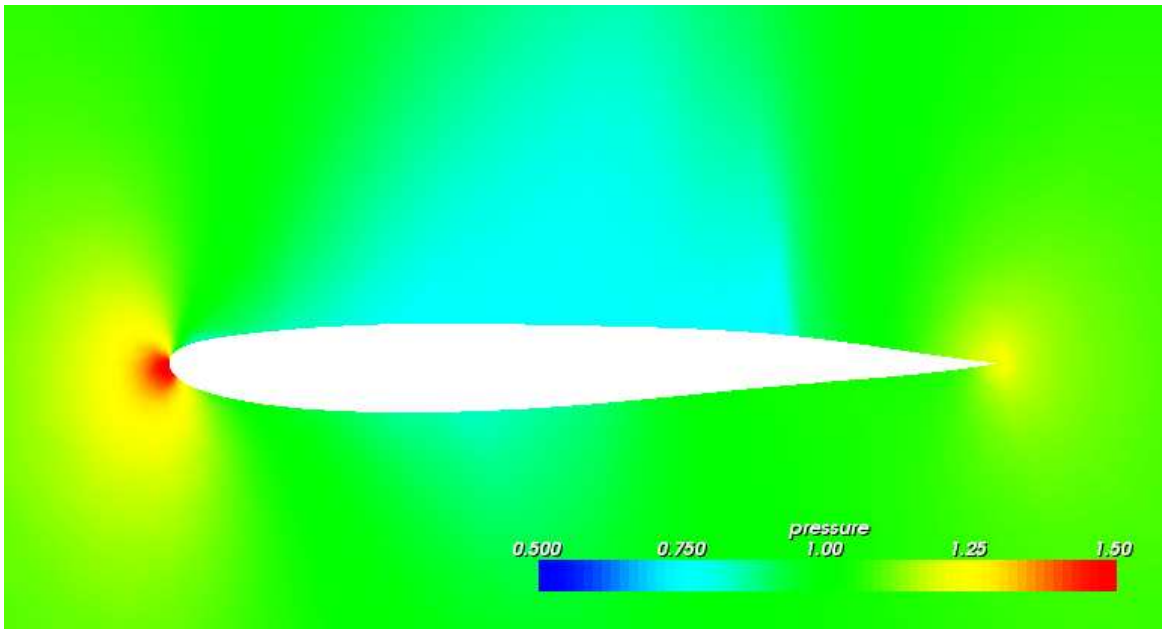


Figure 14: Pressure field at the root section for the basic PSO (240 particules).

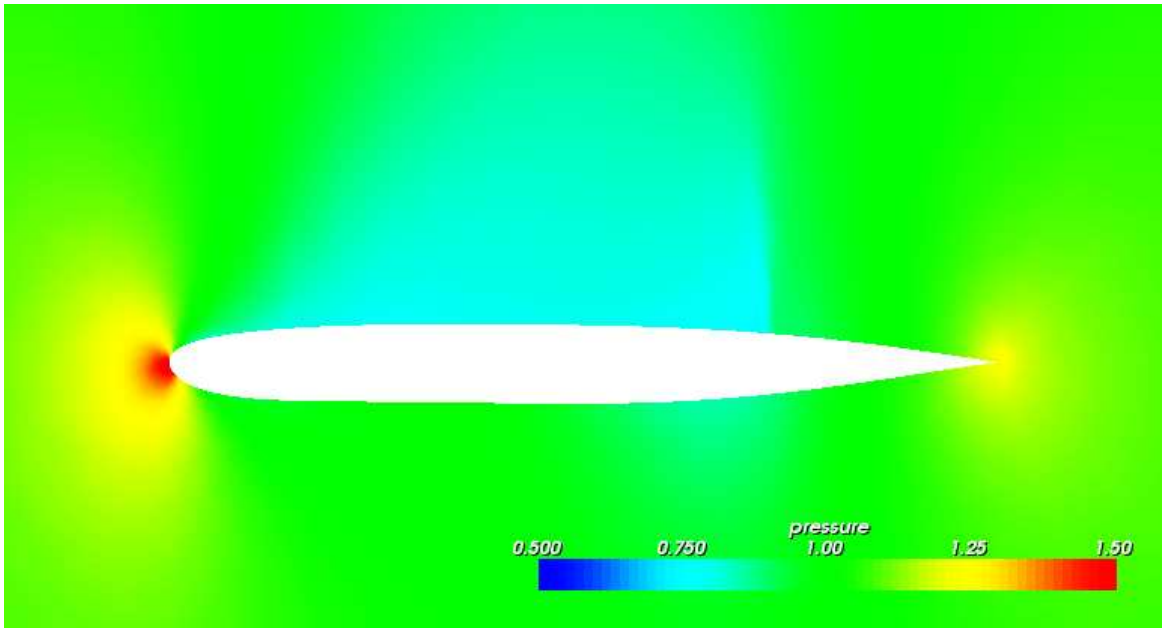


Figure 15: Pressure field at the root section for the MPSOA (30 particles).

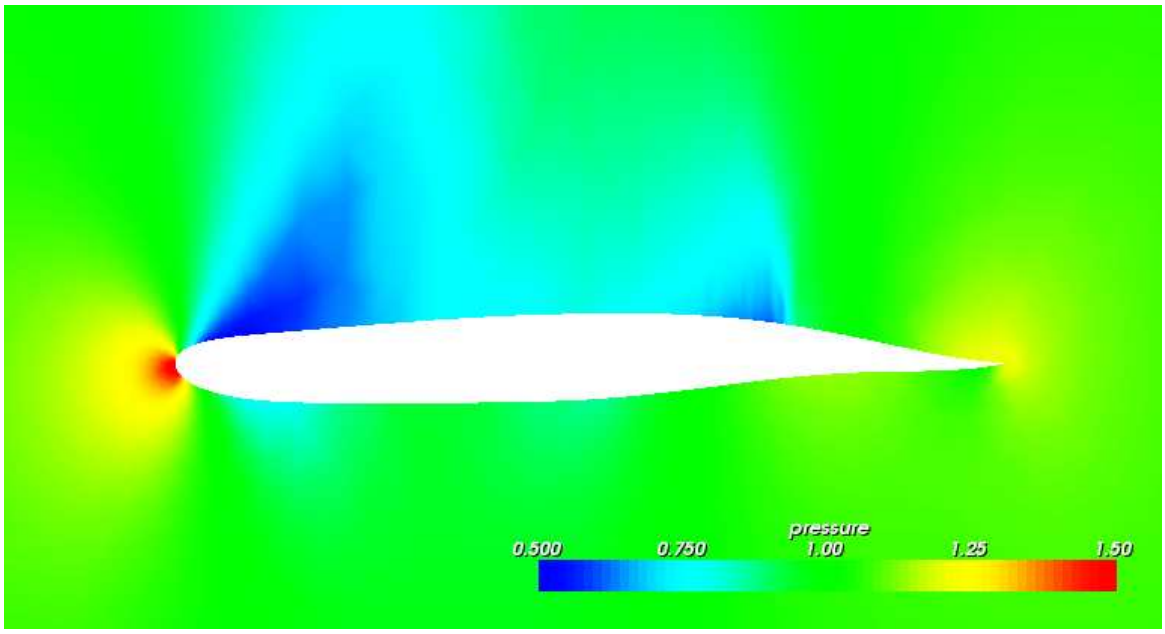


Figure 16: Pressure field at the tip section for the basic PSO (240 particules).

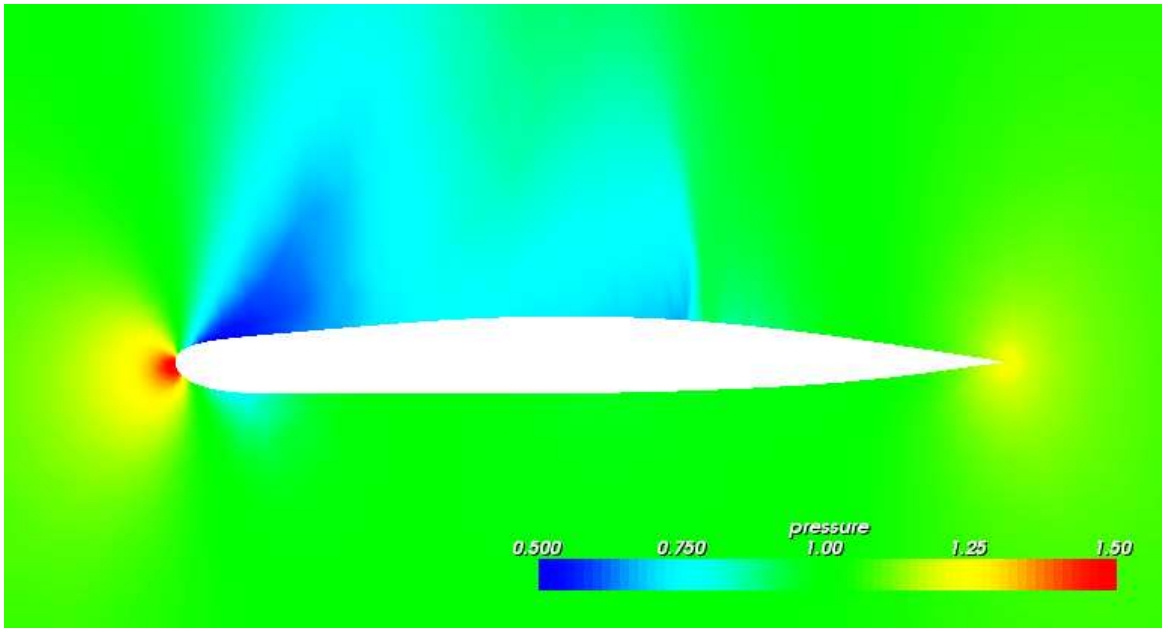


Figure 17: Pressure field at the tip section for the MPSOA (30 particles).

Conclusion

A multi-level parameterization strategy is proposed in the framework of a Particle Swarm Optimization algorithm. In this strategy, at a given parameterization level, the collective memory of the swarm and the initialization domain of the particles are defined to benefit from the search already performed at the lower parameterization level.

The efficiency of the resulting approach is demonstrated for shape optimization problems in electromagnetics and aerodynamics. Especially, the proposed scheme improves the robustness of the optimization procedure, increasing the capability of the PSO algorithm to avoid local optima and reducing the variance of the results due to the inherent stochastic behavior of PSO. Moreover, the proposed method yields a significant decrease of the computational cost, since it allows the use of a smaller swarm without altering the final fitness obtained.

The simplest multi-level strategy was employed in this study, namely the simple level increase. Future works will be focused on the possibility to use a more sophisticated approach, such as V-cycles or Full Multi-Grid (FMG) methods.

References

[1] Aarts, E., and Korst, K. *Simulated annealing and Boltzmann machines.* John Wiley, 1989.

[2] Andreoli, M., Janka, A., and Désidéri, J.-A. Free-form deformation parameterization for multilevel 3D shape optimization in aerodynamics. INRIA Research Report 5019, November 2003.

[3] Bäck, T., and Schwefel, H. *Genetic algorithms in engineering and computer science.* John Wiley & sons, 1995, ch. Evolution strategies I: Variants and their computational implementation, pp. 111–125.

[4] Bäck, T., and Schwefel, H. *Genetic algorithms in engineering and computer science.* John Wiley & sons, 1995, ch. Evolution strategies I: Theorical aspects, pp. 127–140.

[5] Clerc, M., and Kennedy, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation 6*, 1 (February 2002), 58–73.

[6] Combes, P. F. *Micro-ondes.* Dunod, 1997.

[7] Dervieux, A., and Désidéri, J.-A. Compressible flow solvers using unstructured grids. INRIA Research Report 1732, June 1992.

[8] Désidéri, J.-A., Majd, B. A. E., and Janka, A. Nested and self-adaptive bézier parameterization for shape optimization. In *International Conference on Control, Partial Differential Equations and Scientific Computing, Beijing, China* (September 13-16 2004).

[9] Farin, G. *Curves and Surfaces for Computer-Aided Geometric Design.* Academic Press, 1989.

[10] Fourie, P., and Groenwold, A. The particle swarm optimization algorithm in size and shape optimization. *Structural and Multidisciplinary Optimization 23*, 4 (May 2002), 259–267.

[11] Gill, P. E., Murray, W., and Wright, M. H. *Practical Optimization.* Academic Press, 1981.

[12] Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley Company Inc., 1989.

[13] Holland, J. *Adaptation in natural and artificial systems.* University of Michigan Press, 1975.

[14] Kennedy, J., and Eberhart, R. Particle swarm optimization. In *1995 IEEE International Conference on neural networks, Perth, Australia* (1995).

[15] Majd, B. A. E., Désidéri, J.-A., and Habbal, A. A fully multilevel and adaptive algorithm for parameterized shape optimization. In *European Multigrid Conference, EMG 2005, Scheveningen, The Netherlands* (September 27-30 2005).

[16] Majd, B. A. E., Duvigneau, R., and Désidéri, J.-A. Aerodynamic shape optimization using a full and adaptive multilevel algorithm. In *ERCOFTAC Conference Design Optimization : Methods and Applications, Canary Island, Spain* (April 2006).

[17] Marco, N., Désidéri, J., and Lanteri, A. Multi-objective optimization in CFD by genetic algorithms. Tech. rep., INRIA Research Report No 3686, april 1999.

[18] Marco, N., and Lanter, S. A two-level parallelization strategy for genetic algorithms applied to shape optimum design. Tech. rep., INRIA Research Report No 3463, July 1998.

[19] Martinelli, M., and Beux, F. Optimum shape design through multilevel gradient-based method using bézier parametrisation. In *4th ICCFD Conference, Ghent, Belgium* (July 2006).

[20] Michalewics, Z. *Genetic algorithms + data structures = evolutionary programs.* AI series. Springer-Verlag, New York, 1992.

[21] Samareh, J. A survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA Journal 39*, 5 (2001), 877–884.

[22] Sederberg, T., and Parry, S. Free-from deformation of solid geometric models. *Computer Graphics 20*, 4 (1986), 151–160.

[23] Shi, Y., and Eberhart, R. A modified particle swarm optimizer. In *International Conference on Evolutionary Computation* (1998), pp. 69–73.

[24] Venter, G., and Sobieszczanski-Sobieski, J. Particle swarm optimization. *AIAA Journal 41*, 8 (August 2003), 1583–1589.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399