



**HAL**  
open science

## Multiparameter Shape Optimization

Abderrahmane Benzaoui, Régis Duvigneau

► **To cite this version:**

Abderrahmane Benzaoui, Régis Duvigneau. Multiparameter Shape Optimization. Piotr Breitkopf and Rajan Filomeno Coelho. Multidisciplinary Design Optimization in Computational Mechanics, ISTE - Wiley, 2010. inria-00537337

**HAL Id: inria-00537337**

**<https://inria.hal.science/inria-00537337>**

Submitted on 18 Nov 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Multidisciplinary Design Optimization in Computational Mechanics

OMD Partners

June 12, 2009



# Multiparameter Shape Optimization

## 0.1. Introduction

Optimization tools in engineering design often require a high computational cost. This cost originates from two main sources: First, the evaluation of the objective function involved in such problems is in general very expensive. Then, depending on the method employed and on the dimension of the design vector, the optimization procedure requires a high number of evaluations of the objective function to reach the final solution. For instance, when studying an optimization problem in fluid dynamics, many difficulties occur. First, the physical problem is modeled by complex equations (such as Navier-Stokes equations), so that the evaluation of the objective function is cumbersome. Then, the numerical computation of the physical problem is expensive. And finally, the fine parameterization of the object to optimize leads to a high dimension design vector which results in a stiff optimization search. Many authors proposed hierarchical techniques to make the optimization algorithm cheaper. Among these techniques, we can cite the use of a simplified model of the physical problem (for example, the use of Euler equations instead of the Navier-Stokes ones), the use of a metamodel instead of the exact model, or the use of a hierarchical parameterization instead of a single level one. Here we do not describe nor give an exhaustive list of these hierarchical techniques. The interested reader can refer for instance to [GIA 06]. In our study, we are interested only by techniques concerning the parameterization level of the object to optimize (Multilevel algorithms).

To improve the efficiency of the optimization algorithms, some authors propose to use a multilevel approach instead of a direct one. This means that the optimization is carried out, at some steps, on a coarse level where not all the design parameters are considered. This idea is inspired from the multigrid theory used to solve problems

with differential equations. Indeed, it is well known in such problems, that the computation of a fine solution is expensive, not only because of the increase of the cost of a single iteration (resolution of a high dimension linear system), but also because of the low rate of convergence of the iterative algorithm. This is why the multigrid techniques use a coarse level to accelerate the fine level resolution. Several strategies can be considered, ranging from simple level increase to V-cycle or Full Multi-Grid approaches [DES 04].

The idea is similar in optimization. Earlier studies reproduced successfully the multigrid concept and strategies to optimization problems. In particular, Jameson and Martinelli [JAM 98] used the multigrid concept to solve simultaneously the flow equations and the optimization problems. Lewis and Nash [LEW 00] used the same concept to solve optimization problems of systems governed by differential equations. Désidéri et al [DES 04, DES 05, ABO 06] generalized this approach to elaborate a multilevel shape optimization algorithm where the shape is parameterized using Bézier curves or a Free-Form Deformation (FFD) technique [SED 86].

In these studies, the coarse level optimization is carried out on subspaces that depend on the parameterization of the geometry of interest. Moreover, when the Bézier or FFD parameterization is employed, the transfer from a coarse level to a finer one is done using the Bézier degree-elevation process, which is a straightforward technique that permits to add some control points without any modification on the shape. Eventhough, it can not be generalized to any optimization problems.

In the present study, we propose a more efficient and more general method that can accelerate the convergence of the optimization algorithm and can be employed for any kind of problem. This method combines the multigrid concept with the spectral decomposition of the Hessian matrix of the cost function. Indeed, the smallest eigenvalues of the Hessian matrix correspond to directions where the convergence of the optimization algorithm is very slow, while the highest eigenvalues correspond to directions where the convergence is fast when descent optimization algorithms are used. Thus, instead of iterating on the entire design space, our algorithm search for the solution in a selected subspace in order to accelerate the resolution in the directions of low convergence rate. Then it pursues the search on the entire design space. This can be done by several strategies analogous to those of the multigrid methods.

This chapter is organized as follows: we start first by presenting our multilevel approach. Then, we describe in details the implementation of this method in the case of the Multidirectional-search Algorithm [DEN 91]. The method is validated on an analytical test function and then applied to the shape reconstruction problem and to the design of a SuperSonic Business Jet (SSBJ).

## 0.2. Multilevel optimization

### 0.2.1. Description of the Multilevel algorithm

In this section, we present a multilevel algorithm that combines the multigrid concept with the spectral decomposition of the Hessian matrix. As mentioned in the introduction, the idea of this algorithm is motivated by the fact that the smallest eigenvalues of the Hessian matrix correspond to directions where the convergence of the optimization algorithm is very slow, while the highest eigenvalues correspond to directions where the convergence is fast (see §0.3.1). Thus, instead of iterating on the entire design space, this algorithm search for the solution in a selected subspace in order to accelerate the resolution in the directions of low convergence rate. Then, it pursues the search on the entire design space.

Let  $E$  be the entire design space and  $X_0 \in E$  is a starting design vector. The multilevel algorithm starts first by computing the Hessian matrix  $H(X_0)$  and its eigenvectors  $V = (v_1, \dots, v_n)$ . These vectors are ranked from the smallest to the highest corresponding eigenvalues<sup>1</sup>. Suppose that  $X^{(l)} \in E$  is the design vector obtained at a level  $l$ . Then, the multilevel algorithm search for the new design vector  $X^{(l+1)} \in E$  at the new level  $l + 1$  by adding a correction term which minimizes the cost function in the current optimization subspace. This is done as follows:

Suppose that the new level is characterized by  $m = m_{l+1}$  parameters ( $m \leq n$ ) and consider the basis  $V_m$  and the subspace  $E_m$  defined by:

$$V_m = (v_1, \dots, v_m) \quad [1]$$

$$E_m = \{z = V_m y \mid y \in \mathbb{R}^m\} \quad [2]$$

The correction term must be a vector from  $E_m$ . Hence, to find the best correction, the optimization algorithm search for a vector  $y \in \mathbb{R}^m$  so that the cost function:

$$g(y) = f(X^{(l)} + V_m y) \quad [3]$$

is minimized. In this way, we can either go from a coarse to a finer level or from a fine to a coarser level without loosing any information obtained from the former level. Let

---

1. In the cas of non positive matrices, these vectors are ranked from the smallest to the highest absolute values of the corresponding eigenvalues

$y^*$  be the optimum value that minimizes (3). Then, the design vector of the new level is given by:

$$X^{(l+1)} = X^{(l)} + V_m y^* \quad [4]$$

The resolution at any level can be full or incomplete. The algorithm can be carried out using several levels, ordered in some way called a cycle, by analogy to the multigrid terminology. So a cycle is defined by a sequence of levels of dimensions  $(m_1, m_2, \dots, m_{nl})$  and by the corresponding numbers of iterations  $(it_1, it_2, \dots, it_{nl})$  where  $nl$  is the number of levels in the cycle. Then, the cycle can be repeated many times if necessary.

The multilevel algorithm is summarized below:

- 1) Read the input data and initialize the design variable  $X_0$ ;
- 2) initialize the number of cycles at  $k = 0$ ;
- 3) while  $k < nc$  and stopping criterion not satisfied, perform a multilevel cycle:
  - a) compute the Hessian matrix  $H(X_k)$  and evaluate its eigenvectors;
  - b) initialize the level number at  $l = 1$ ;
  - c) let  $X^l = X_k$
  - d) while  $l \leq nl$  do:
    - i. select a basis  $V_m$  with  $m = m_l$  eigenvectors,
    - ii. use an optimization algorithm to perform a correction vector  $y^* \in \mathfrak{R}^m$  that minimizes the cost function  $g(y) = f(X^{(l)} + V_m y)$ ;
    - iii.  $X^{(l+1)} \leftarrow X^{(l)} + V_m y^*$ ;
    - iv.  $l \leftarrow l + 1$  and goto (3d);
  - e)  $X_{k+1} \leftarrow X^{(l)}$ ;
  - f)  $k \leftarrow k + 1$  and goto (3);
- 4)  $X^* \leftarrow X_{(k)}$ .

The stopping criterion in the step (3) of the above algorithm is the relative decrease in the cost function. While this decrease is above a given value, the algorithm pursues the computation. Furthermore, in step (3d), the  $m_l$  eigenvectors correspond to the smallest eigenvalues of the Hessian. In this way, on a coarse level, the algorithm focuses only on directions with a low convergence rate.

### 0.2.2. Multilevel optimization using the Multi-directional-Search Algorithm

The Multi-directional-Search Algorithm (MSA), developed by Dennis and Torczon [DEN 91], uses a simplex approach to find the optimum. This simplex is composed of  $m + 1$  design vectors  $y_0, y_1, \dots, y_m$ . After being initialized, the simplex in each new iteration is obtained as follows:

Suppose that  $y_0^{(k)}$  is the vertex that gives the smallest cost function among all the simplex vertices, where  $k$  is the iteration number. Then the simplex is reflected with respect to  $y_0^{(k)}$  according to the following equation:

$$\widetilde{y}_i^{(k)} = (1 + \alpha)y_0^{(k)} - \alpha y_i^{(k)} \quad i = 0, \dots, m \quad [5]$$

where  $\widetilde{y}_i^{(k)}$  represent the reflected vertices and  $\alpha$  is a positive real usually equal to 1. If the reflection is successful, i.e. if one of the new vertices has a smallest cost function than that of  $y_0^{(k)}$ , this means that perhaps the solution is in this direction. So the new simplex is expanded to pursue the search in this direction. The simplex of the new iteration is then obtained by:

$$y_i^{(k+1)} = (1 - \gamma)\widetilde{y}_0^{(k)} + \gamma\widetilde{y}_i^{(k)} \quad i = 0, \dots, m \quad [6]$$

where  $\gamma > 1$  is the expansion coefficient. Usually  $\gamma$  is set to 2. In the other case, if the reflection fails, the simplex is contracted so that the vertices become closer to the best one. The simplex of the new iteration is then obtained by:

$$y_i^{(k+1)} = (1 - \beta)\widetilde{y}_0^{(k)} + \beta\widetilde{y}_i^{(k)} \quad i = 0, \dots, m \quad [7]$$

where  $\beta$  is the contraction coefficient. Usually  $\beta$  is set to  $-\frac{1}{2}$ .

In the present study, for the stopping criterion, we compute the distance (using the Euclidean norm) of vertices  $y_i^{(k)}$  from the best one  $y_0^{(k)}$ . Then, the stopping criterion is given by:

$$\frac{\max_i \left( \| y_i^{(k)} - y_0^{(k)} \| \right)}{\| y_{ref} \|} < \epsilon \quad [8]$$

where  $y_{ref}$  is a given reference vector. While this condition is not satisfied and the number of iterations is below the maximum value for the selected level, the MSA algorithm pursues the iterations.

In the case of the Multilevel optimization, the MSA algorithm is used in the step (3dii) of the algorithm described above. In this case,  $y$  is the correction vector written



in a basis of the Hessian eigenvectors and  $m$  the dimension of the subspace of optimization. In this way, the number of vertices in a coarse level is less than that of the fine level. This means that the optimization cost is lower, which is an other advantage of the multilevel approach.

The simplex can be initialized in any way provided that the vertices are linearly independent. But to insure the transfer of information from the previous levels, one of the vertices need to be the null vector so that no correction is added to the previous design vector  $X^{(l)}$ . Moreover, to take advantage of the directions of the Hessian eigenvectors, we choose the other vertices to be colinear with the unit vectors. Furthermore, for an efficient search, the initial size of the simplex need to be large at the beginning of the multilevel procedure (the first cycles). However, at the end of this procedure (the last cycles), when  $X^{(l)}$  is near the final solution, we need to have a small simplex since the correction to add to  $X^{(l)}$  to reach the optimum is small.

### 0.3. Validation

The Multilevel algorithm is tested and validated on a quadratic function for which the search of the optimum is straightforward. This function is given by the following expression:

$$f(X) = a + B^T X + X^T C X \quad [9]$$

where  $X \in \mathbb{R}^n$  is the optimization parameter. In this function,  $a \in \mathbb{R}$ ,  $B \in \mathbb{R}^n$  and  $C \in \mathbb{R}^{n \times n}$  are chosen arbitrarily, but to guarantee the existence of a minimum,  $C$  is taken as a symmetric positive definite matrix. In this case, the Hessian of  $f$  is simply equal to  $2C$  and is also positive definite. Since we would like to use this function to test the multilevel optimization with a Hessian decomposition approach,  $C$  is chosen such that we control its eigenvalues and eigenvectors. Details about the values of  $a$ ,  $B$  and  $C$  can be found in [BEN 09].

Table (1) gives the condition number of the matrix  $C$  and the optimal value of the tested function. As we will see, the condition number has an effect on the convergence rate of the optimization algorithm and is thus an important parameter.

Function	Condition number	Optimal value
$f$	18000	-32.517335

**Table 1.** Condition number and optimal value for the tested function.

In the following section, we try to find the above optimum using the multilevel algorithm. The efficiency of the multilevel strategy is measured by the number of evaluations of the function required to reach the optimum.

### 0.3.1. Single level optimization

Before testing the multilevel algorithm we try to find the optimum value of the above function using the MSA method without any multilevel nor Hessian decomposition strategy. For this purpose, we carry out 35000 iterations of the MSA optimizer in order to find a good approximation of the value given in table (1). The optimum value obtained is  $f_{min} = -32.516920$ . We have so a relative error of 0.001%. The corresponding number of evaluations is 910014, which is very high. This means that using the same algorithm in a problem with the same parameterization size, and where the evaluation of the cost function is expensive can be very stiff, if possible. So a limitation in the accuracy is then necessary. If we limitate the iteration number to 5000, which is still a high number, the optimum value obtained is  $-29.2075$  and the relative error is 10%.

This exemple shows the necessity to improve the optimization procedure to become faster. As we mentioned in §0.2.1, the convergence of the optimization algorithm is quick in directions of high eigenvalues of the Hessian and slow in directions of low eigenvalues. To justify this idea, we compute the components  $e_i$  of the error,  $|X - X^*|$ , where  $X^*$  is the optimum design vector. These components are calculated in a basis of the Hessian eigenvectors and ordered from the lowest to the highest corresponding eigenvalue. Figure (1) shows a histogram of these components at iterations 0, 5 and 50 respectively. In this figure, the horizontal axis corresponds to the indices of the compenents. From this figure, we can notice that the compenents of the error that correspond to high eigenvalues fall down quickly, whereas those corresponding to small eigenvalues are still at high order after 50 iterations. This confirms our idea that directions of the smallest eigenvalues require more effort to achieve the optimum. This is what we hope to overcome using our multilevel algorithm.

In order to validate the multilevel algorithm and to prove its efficiency, we start by optimizing the above function using a simple level strategy with a spectral decomposition of the Hessian. This consists in working in the eigenvectors basis. Therefore, the optimized function is equivalent to that of equation (3) where  $m = n$ . This is done by setting  $nl = 1$  and  $m_1 = n$  in the algorithm described in §0.2. The evolution of the cost function with respect to the number of evaluations is presented in figure (2). The result of this optimization is spectacular. The simple use of the Hessian eigenvectors basis permits the MSA optimizer to converge very fast. Indeed, the optimum value  $f_{min} = -32.517335$  was reached after 130 iterations only with 3394 evaluations of the cost function. This method is thus 268 times faster then the first one! As we will see, this performance depends on the condition number of the Hessian.

### 0.3.2. Multilevel optimization

As mentioned above, the multilevel strategy can improve the performance of the optimization algorithm when it is used in an adequate manner. For the function  $f$ , we carry out the algorithm described in §0.2 with two and three levels using the strategy described in table (2). To be efficient, it is recommended to use a few number of iterations at each level and to repeat the multilevel cycles until convergence.

Cycles	cycle 1	cycle 2	cycle 3
Fine level resolution	•	•	•
Coarse level resolution	↘	↘	↘
	•	•	•

**Table 2.** schematic description of the two-level cycles.

Table (3) presents the multilevel tests using the strategy described in table (2), whereas figure (3) shows the evolution of the cost function with respect to the number of evaluations for the best cases. In table (3), the case 1 corresponds to the single level optimization using the spectral decomposition of the Hessian.

As we can see in table (3) and in figure (3), the multilevel strategy allows a faster optimization than the single level one. Indeed, if we compare cases 1 and 7 for instance, we can see that the multilevel strategy allows a reduction in the number of evaluations of about 27%. It is not necessary to use a larger number of iterations by level because this will result in an increased number of evaluations without any acceleration of the convergence. However, if we reduce the number of iterations, the convergence becomes slow resulting in an increased number of cycles and so an increased number of evaluations, while the accuracy on the optimum value is worst.

Furthermore, from previous tests (see [BEN 09]), we can state that when the condition number of the Hessian is small, the Hessian decomposition is not interesting and the number of evaluations does not decrease sensitively. Eventhough, the multilevel approach is still interesting. However, when the condition number is high, the Hessian decomposition seems to be necessary and very efficient. The multilevel approach is also interesting in this case.

From these observations we can conclude that the multilevel strategy permits to accelerate the convergence of the optimization algorithm and to reduce its cost. This is true whatever is the condition number of the Hessian matrix. If this number is high, it becomes necessary to search for the solution in subspaces generated by the eigenvectors of the Hessian. In this case, the gain in the optimization cost is significant.

Case	levels	number of iterations by cycle	number of cycle	number of evaluations	optimal value
1	12	130	1	3394	-32.517335
2	12	3	19	2661	-32.517335
	6	↘ 3			
3	12	3	22	2773	-32.517307
	6	↘ 2			
4	12	3	18	2773	-32.517335
	6	↘ 4			
5	12	2	22	2817	-32.517302
	6	↘ 4			
6	12	4	18	2737	-32.517335
	6	↘ 2			
7	12	3	17	2483	-32.517335
	6	↘ 2			
	3	↘ 2			

**Table 3.** Description of the Multilevel optimization for the function  $f$ .

## 0.4. Applications

### 0.4.1. First application: Shape reconstruction problem

#### 0.4.1.1. problem description

In this application, we would like to approach a given target function  $F_0(t)$  by a Bézier curve  $F(t, X)$ , where  $X = (x_1, \dots, x_n)^T$  is the design vector whose components are the control points of the Bézier curve. So, the approximated function is given by:

$$F(t, X) = \sum_{k=0}^{n-1} B_{n-1}^k(t) x_{k+1} \quad [10]$$

for  $t \in [0, 1]$ . In equation (10),  $n$  is the parameterization level and  $B_{n-1}^k(t)$  are the Bernstein polynomials given by:

$$B_n^k(t) = C_n^k t^k (1-t)^{n-k} \quad [11]$$

where

$$C_n^k = \frac{n!}{k!(n-k)!} \quad [12]$$

The target function selected in this study is also a Bézier curve<sup>2</sup> of degree  $n_0 > n$ . It is given by:

$$F_0(t) = \sum_{k=0}^{n_0-1} B_{n_0-1}^k(t) x_{0k+1} \quad [13]$$

where  $x_{01}, \dots, x_{0n_0}$  are the control points of the target function. Details about this function can be found in [BEN 09].

The squared error of the approximation of the target function  $F_0(t)$  by the Bézier curve  $F(t, X)$  is given by:

$$e(X) = \int_0^1 (F(t, X) - F_0(t))^2 dt \quad [14]$$

For a given parameterization level  $n$ , the approximation problem is equivalent to the search of a design vector  $X$  that minimizes the squared error. This is hence an unconstrained optimization problem where the cost function is  $f(X) \equiv e(X)$ . A theoretical solution of this problem can be found easily by considering that the gradient of the cost function vanishes at the optimum. This leads to the following linear system:

$$A \cdot X = \mathcal{B} \quad [15]$$

---

2. This is just an exemple and we can select any kind of functions. The resolution method and the behaviour of the optimization algorithm do not depend on the nature of the target shape.

where  $A = (a_{kj})_{1 \leq k \leq n, 1 \leq j \leq n}$ ,  $B = (b_k)_{1 \leq k \leq n}$  and:

$$a_{kj} = 2 \int_0^1 B_{n-1}^{k-1}(t) B_{n-1}^{j-1}(t) dt \quad [16]$$

$$b_k = 2 \int_0^1 B_{n-1}^{k-1}(t) F_0(t) dt \quad [17]$$

Note that the matrix  $A$  is equal to the Hessian of the cost function and is symmetric positive definite. This quadratic problem has a unique solution which can be obtained by solving the system (15).

To evaluate the cost function, the interval  $[0, 1]$  is subdivided into  $n_p$  points. The squared error (14) is then given by:

$$e(X) \approx \frac{1}{n_p} \sum_{p=1}^{n_p} \left( F \left( \frac{p-1}{n_p-1}, X \right) - F_0 \left( \frac{p-1}{n_p-1} \right) \right)^2 \quad [18]$$

A similar discretization is done to compute the Hessian matrix  $A$ , given by (16), and the right hand side term of equation (15) which is given by (17).

As application of our multilevel algorithm, we employ it to approximate two target functions  $F_{01}$  and  $F_{02}$  of parameterization levels 14 and 18 by respectively functions  $F_1$  and  $F_2$  of levels 8 and 16. The corresponding cost functions are so  $f_1$  and  $f_2$  respectively. This allows us to test at the same time the ability to approximate different shapes and the influence of the condition number of the Hessian matrix on the algorithm. Table (4) gives the optimum values and the condition numbers corresponding to these cost functions for  $n_p = 20$ , whereas figure (4) gives the shape of the target and the approximated functions. These optimum values are obtained from the resolution of (15).

Cost Function	$n_0$	$n$	Condition number	Optimal value
$f_1$	14	8	4617.6	$1.751264 \times 10^{-5}$
$f_2$	18	16	$3.6 \times 10^9$	$1.037 \times 10^{-9}$

**Table 4.** Condition numbers and optimal values for the two tested functions of the shape reconstruction problem.

0.4.1.2. *Experimentation and results*

In this section, we try to obtain the approximated function using the multilevel optimization algorithm with the MSA method. For this purpose, we test many cases in order to find the best optimization strategy. Tables (5) and (6) describe some cases for cost functions  $f_1$  and  $f_2$  respectively (corresponding to the target functions  $F_{01}$  and  $F_{02}$ ) whereas figure (5) shows the evolution of these functions with respect to the number of evaluations for the studied cases. In these tables, cases 0 correspond to a classical MSA optimization without the Hessian decomposition, whereas the other cases use a basis of the Hessian eigenvectors.

Case	levels	number of iterations by cycle	number of cycle	number of evaluations	optimal value $\times 10^5$
0	8	200	1	3610	2.303
1	8	135	1	2440	1.751264
2	8	3	18	2125	1.751264
	4	5			
3	8	3	19	2053	1.751264
	4	4			
4	8	3	15	1471	1.751264
	4	3			
5	8	3	19	1958	1.751264
	4	2			
	2	2			

**Table 5.** *The tested cases for the multilevel optimization of the cost functions  $f_1$  of the shape reconstruction problem.*

From tables (5) and (6) and figure (5) we can see that the behaviour of the optimization algorithm is the same as that for the analytical function test case of §0.3. For medium Hessian condition, the simple use of the Hessian decomposition allows the algorithm to be faster. The multilevel strategy permits to accelerate the convergence. The best acceleration is obtained with few iterations by level and by cycle (cases 4 and 5 for function  $f_1$  and 2 and 4 for function  $f_2$ ). This acceleration is more important for a high condition number.

Case	levels	number of iterations by cycle	number of cycle	number of evaluations	optimal value
0	16	200	1	6818	0.00041
1	16	140	1	4778	$1.037 \times 10^{-9}$
2	16	3	20	3641	$1.037 \times 10^{-9}$
	8	3			
3	16	4	20	4321	$1.037 \times 10^{-9}$
	8	3			
4	16	3	20	3781	$1.037 \times 10^{-9}$
	8	2			
	4	2			

**Table 6.** The tested cases for the multilevel optimization of the cost functions  $f_2$  of the shape reconstruction problem.

Thus, the multilevel optimization algorithm can be employed to solve the shape reconstruction problem. For high quality approximations, a high order Bézier curve is required, and then the condition number is large. In this case, the multilevel approach using the spectral decomposition of the Hessian is very efficient.

#### 0.4.2. Second application: Optimum design of a SuperSonic Business Jet

The optimum design of a SuperSonic Business Jet (SSBJ) is a very challenging optimization problem because of its complexity. This problem, which involves aerodynamics, propulsion and structural mechanics has been used by Sobieszczanski-Sobieski et al to test their "BLISS" (Bi-Level Integrated System Synthesis) methods [SOB 00]. Their objective was the maximization of the aircraft range subject to some constraints. In our case, the SSBJ design problem is used to test the multilevel optimization algorithm in a complex context. The aim of our study is the minimization of the total weight of the aircraft at the take-off (TOW) subject to constraints on its range  $R$ , the take-off length  $L_{take}$  and the landing speed  $v_{land}$ . More precisely, these constraints are:

$$\begin{cases} R > 6500 \text{ km} \\ L_{take} < 1828 \text{ m} \\ v_{land} < 70 \text{ m/s} \end{cases} \quad [19]$$



The set of the 15 design variables is composed of geometrical parameters as well as flight conditions. Thus, additional constraints on the range of these variables are introduced. All these constraints are taken into account in our optimization code by means of penalty terms. Moreover, the design of the aircraft and the evaluation of its performances are made using routines provided by Dassault Aviation which are considered here as black boxes. This optimization problem is hard due to the multimodality of the objective function and to the complexity of the constraints.

Table (7) presents the different tests carried out to optimize the aircraft weight, whereas figure (6) shows the evolution of the weight with respect to the number of cost function evaluations for these tests. In table (7), case 0 refers to the classical single level MSA optimization without Hessian decomposition, whereas case 1 refers to the single level MSA optimization working in the Hessian eigenvectors basis (see §0.3.1). Case 2 is similar to case 1 but with many cycles and few iterations by cycle. This consists in updating the Hessian and reinitializing the simplex after each 10 MSA iterations. This is obtained by staying  $nc > 1$ ,  $nl = 1$  and  $m_l = 15$  in the algorithm presented in §0.2. Finally, Cases 3 and 4 corresponds to a multilevel MSA optimization as described in table (2). Note that, in these tests, the Hessian matrix is computed locally using the finite-difference method and is updated at each multilevel cycle. Furthermore, the starting point of the optimization procedure is taken at the center of the design space. This arbitrary choice does not satisfy the constraints, nevertheless the multilevel algorithm is able to reach a satisfactory solution. The design parameters of the best solution (case 4 of table (7)) are given in table (8) which gives also the bounds of the design space. Table (9) shows the corresponding performances of the aircraft. We can see that this solution is very close to the bounds of the design space and satisfies barely the constraints. This fact makes the optimization procedure difficult because the simplex tries to go on forbidden regions. Even though, the algorithm is able to reach the solution.

Case	levels	number of iterations by cycle	number of cycle	number of evaluations	optimal weight (kg)
0	15	800	1	25617	38964
1	15	225	1	7764	35390
2	15	10	27	21122	34731
3	15	3	24	14857	35073
	7	3			
4	15	5	10	7311	34389
	7	6			

**Table 7.** The tested cases for the multilevel MSA optimization of SSBJ design problem.

Variable	lower limite	upper limite	Optimal value
Cruise altitude	8000 <i>m</i>	18500 <i>m</i>	15306.12 <i>m</i>
Cruise Mach number	1.6	2	1.64
Reference wing area	100 <i>m</i> <sup>2</sup>	200 <i>m</i> <sup>2</sup>	100.52
Leading edge sweep of the wing	40°	70°	49.10°
Trailing edge sweep of the wing	-10°	20°	14.45°
Wing taper ratio	0.05	0.5	0.15
Thickness/chord ratio for the wing	0.04	0.08	0.044
Leading edge sweep of the vertical stabilizer	40°	70°	69.91°
Trailing edge sweep of the vertical stabilizer	0°	10°	1.75°
Vertical stabilizer taper ratio	0.05	0.5	0.058
Thickness/chord ratio for the vertical stabilizer	0.05	0.08	0.075
Fuselage diameter	2 <i>m</i>	2.5 <i>m</i>	2.44 <i>m</i>
Fuel weight	15000 <i>kg</i>	40000 <i>kg</i>	15000 <i>kg</i>
Maximum incidence angle	10°	15°	14.99°
Landing/Take-off weight ratio	0.85	0.95	0.87

**Table 8.** The optimal Design parameters (case 4 of table (7)) and the corresponding bounds for the SSBJ problem.

Performance	Value
Take-off weight	34389.47 <i>kg</i>
Range	6505.52 <i>km</i>
Take-off length	1806.76 <i>m</i>
Landing speed	69.92 <i>m/s</i>

**Table 9.** Performances of the aircraft for the optimal design (case 4 of table (7)).

On figure (6), we can see that the multilevel approach allows, not only to reduce the optimization cost, but also to achieve better performances. However, since the objective function is multimodal, the MSA optimiser can be easily trapped by a local minimum. Therefore, it is difficult to tune the computational parameters and to find the best multilevel strategy that permits to get good performances.

Note that, in the above results, we take into account the number of evaluations of the cost function due to the Hessian computation by the finite-difference method. This number represents, in most cases, more than 50% of the total number of evaluations. This means that these results could be better if the Hessian is computed by a chipper method. For instance, it could be interesting to estimate the Hessian using

a least-square approach on the basis of points already evaluated during a stochastic optimization.

## 0.5. Conclusion

In this chapter we present an efficient optimization method based on an algebraic multilevel approach. This method uses the eigenvectors of the Hessian of the objective function to define search subspaces. Therefore, this method is general and can be applied to any kind of problems. Moreover, this multilevel approach can be used with any optimization algorithm, but in this study it is implemented with the Multidirectional Search Algorithm.

In order to validate the multilevel algorithm, it is tested with a quadratic function. This test case shows that the use of the eigenvectors of the Hessian as basis of the design space accelerates the convergence of the optimization algorithm. This acceleration is significant for medium and high condition number of the Hessian. Additional acceleration is obtained when the Hessian decomposition is combined to an adequate multilevel strategy. A decreasing level with few iterations by level and by cycle is found to be a satisfactory strategy.

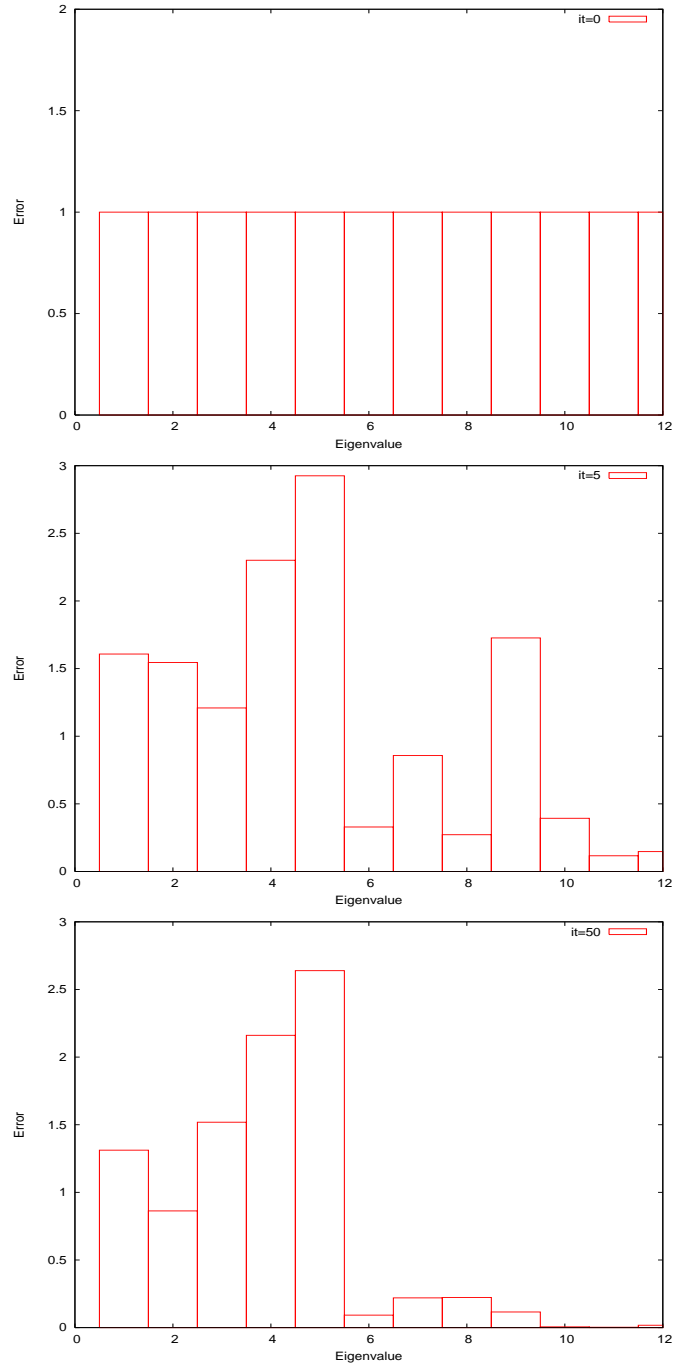
The method is then applied to two problems: shape reconstruction and Supersonic Business Jet design. The first application consists in a least-square approximation of a given shape by a Bezier curve. The cost function is therefore quadratic. In this case, the behaviour of the multilevel algorithm is similar to that of the validation test: The use of the Hessian eigenvectors permits to well approximate the target shape with a moderate cost. This cost is lower when using the same multilevel strategy as that for the validation case.

In the second application, the problem is multimodal and constrained. Moreover, the cost function evaluation is considered as a black box, which means that the Hessian is not available. Then, it is computed using a finite-difference discretization. Many experiments are carried out using the multilevel algorithm with the MSA method. It is difficult to tune the computation parameters. Nevertheless, the multilevel algorithm gives satisfactory results. Indeed, using the multilevel approach and the eigenvectors of the Hessian we obtain a lower aircraft weight than that of the classical MSA method and with a lower cost.

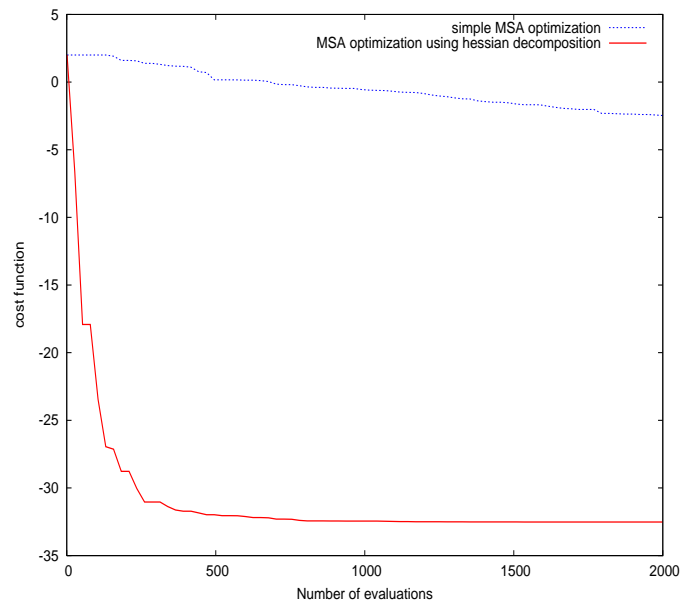
As mentioned above, using the MSA method, it was difficult to find a satisfactory multilevel strategy for the SSBJ problem. For such problem, a stochastic algorithm could be more suitable. A priori, the Multilevel approach can be used with any optimization method. Presently, a Particle Swarm Optimization method is implemented in our multilevel procedure and the resulting algorithm is under study.

## 0.6. Bibliography

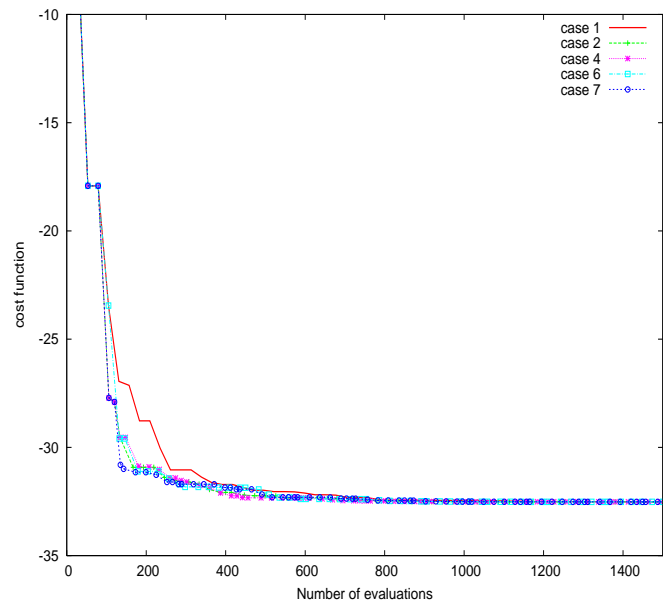
- [ABO 06] ABOU-ELMAJD B., DUVIGNEAU R., DÉSIDÉRI J. A., “Aerodynamic Shape Optimization using a Full Adaptive Multilevel Algorithm”, *ERCOFTAC Conference Design Optimization : Methods and Applications*, Canary Island, Spain, April 2006.
- [BEN 09] BENZAOUI A., DUVIGNEAU R., Efficient Hierarchical Optimization using an Algebraic Multilevel Approach, 2009.
- [DEN 91] DENNIS J. E., TORCZON V., “Direct Search Methods On Parallel Machines”, *SIAM Journal on Optimization*, vol. 1 (4), p. 448-474, 1991.
- [DES 04] DESIDÉRI J. A., ABOU-ELMAJD B., JANKA A., “Nested and Self-adaptive Bézier Parameterization for shape Optimization”, *International Conference on Control, Partial Differential Equations and Scientific Computing*, Beijing, China, September 2004.
- [DES 05] DESIDÉRI J. A., “Two-level ideal algorithm for shape optimization”, *International Conference on Advances in Numerical Mathematics*, Moscow, September 2005.
- [GIA 06] GIANNAKOGLU K. C., KARAKASIS M. K., “Hierarchical and Distributed Metamodel-Assisted Evolutionary Algorithms”, *Introduction to Optimization and Multidisciplinary Design*, von Karman Institute of Fluid Dynamics, Lecture Series, March 2006.
- [JAM 98] JAMESON A., MARTINELLI L., “Optimum Aerodynamic Design using the Navier-Stokes Equation”, *Theoretical and Computational Fluid Dynamics*, vol. 10, p. 213-237, 1998.
- [LEW 00] LEWIS R., NASH S., “A Multigrid Approach to the Optimization of Systems Governed by Differential Equations”, *American Institute of Aeronautics and Astronautics, Reston*, vol. VA, AIAA-2000-4890, 2000.
- [SED 86] SEDERBERG T., PARRY S., “Free-form deformation of solid geometric models”, *Computer Graphics*, vol. 20 (4), p. 151-160, 1986.
- [SOB 00] SOBIESZCZANSKI-SOBIESKI J., AGTE J. S., SANDUSKY R. R., “Bilevel Integrated System Synthesis”, *AIAA Journal*, vol. 38 (1), p. 164–172, 2000.



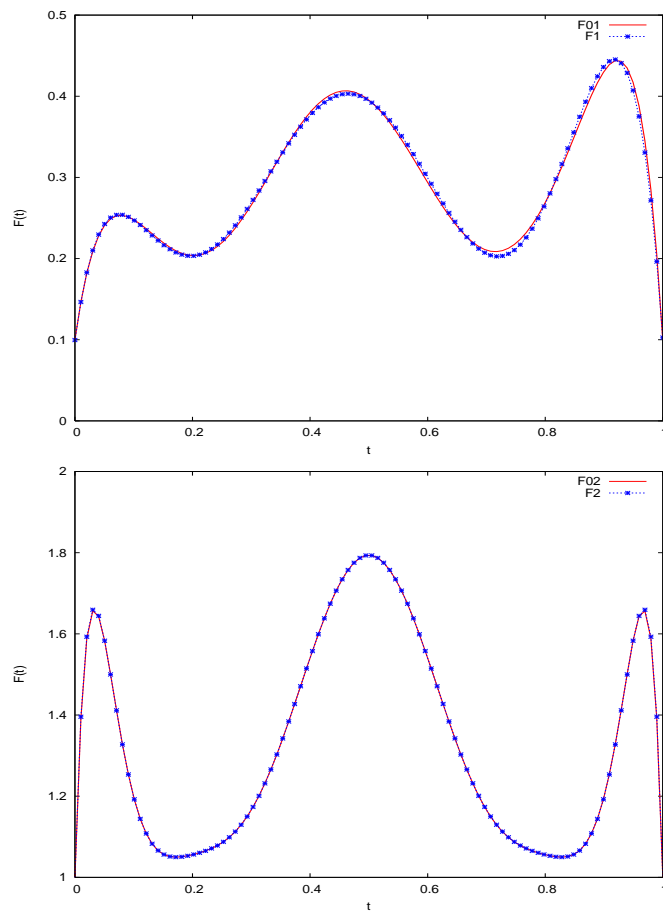
**Figure 1.** Histogram of the error components in a basis of the Hessian eigenvectors for a single level optimization of the function  $f$ .



**Figure 2.** Evolution of the cost function  $f$  for a single level optimization.

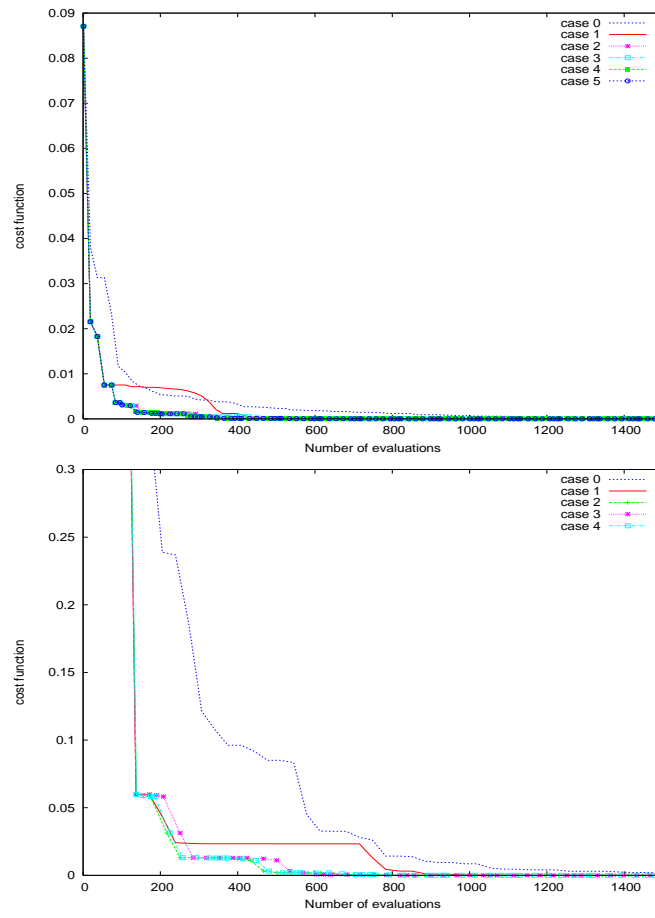


**Figure 3.** Evolution of the cost function  $f$  for a multilevel optimization.

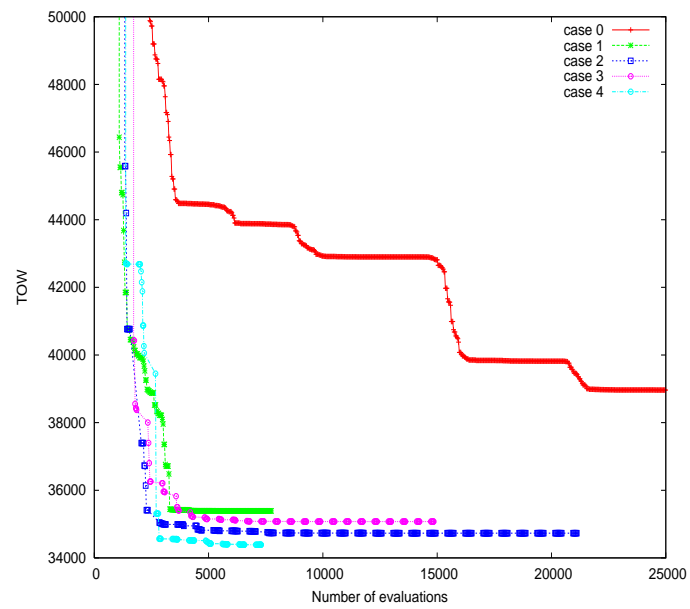


**Figure 4.** Target and approximated curves for functions  $F_{01}$  (up) and  $F_{02}$  (down) of the shape reconstruction problem.





**Figure 5.** Evolution of the cost functions  $f_1$  (up) and  $f_2$  (down) of the shape reconstruction problem for several optimization strategies.



**Figure 6.** Evolution of the aircraft weight for single level and multilevel MSA optimization

