

Hessian transfer for multilevel and adaptive shape optimization

Badr Abou El Majd^{1,*}, Ouail Ouchetto², Jean Antoine Désidéri³, and Abderrahmane Habbal⁴

¹ LIMSAD Lab., Faculty of Science Ain Chock, Hassan II University, Casablanca 20100, Morocco

² LR2I Lab., FSJES Ain Chock, Hassan II University, Casablanca 20100, Morocco

³ Opale Project-Team, INRIA, Sophia-Antipolis Méditerranée Centre, Sophia Antipolis 06902, France

⁴ JAD Lab., University of Nice-Sophia-Antipolis, Nice 06000, France

Received 5 February 2016 / Accepted 9 January 2017

Abstract – We have developed a multilevel and adaption parametric strategies solved by optimization algorithms which require only the availability of objective function values but no derivative information. The key success of these hierarchical strategies refer to the quality of the downward and upward transfers of information. In this paper, we extend our approach when using a derivative-based optimization algorithms. The aim is to better re-initialize the Hessian and the gradient during the optimization process based on our construction of the downward and upward operators. The efficiency of this proposed approach is demonstrated by numerical experiments on an inverse shape model.

Key words: Shape parametric optimization, Hessian and gradient transfer, Multilevel, Adaption, BFGS method.

1 Introduction

We focus on improving the computational efficiency on optimum shape algorithms for application in which the cost functional relies on the prior solution of a complex set of partial-differential equations (PDEs), such as those governing compressible aerodynamics (e.g. the Euler equations), or related coupled disciplines [1] such as structural mechanics (e.g. elasticity), or electromagnetics [2] (e.g. the Maxwell equations). Hence, each evaluation functional is computationally costly. In several papers [3–8], we introduced a conceptual parallel from grid to geometrical parametrization by constructing a full and adaptive multi-level optimum-shape algorithm (FAMOS), analogous to the classical full multigrid method (FMG) known to have optimum linear complexity with respect to the number of degrees of freedom). We demonstrated the efficiency of this construction for problems of drag reduction for transonic flight, external noise reduction for supersonic flight, and aerodynamic and structural optimization of a business-jet wingshape. Figure 1 illustrates an unstructured mesh, composed of 31,124 nodes and 173,445 elements, generated around the wing, including a refined area in the vicinity of the shock. Figure 2 shows a comparison of the convergence for the three strategies; FMOSA is the most efficient, yielding a shape of better fitness using a smaller

computational effort. Using such setting, the wing geometry of an aircraft in transonic flow has been optimized by the minimization of a functional of the form:

$$\mathcal{J}_{\text{OPT}} = \frac{C_D}{C_{D0}} + 10^4 \max \left(0, 0.999 - \frac{C_L}{C_{L0}} \right). \quad (1)$$

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

Until now, free-optimization techniques, such as the classical simplex search method, the genetic algorithms (GAs) and the particular swarm optimizers (PSOs), has been employed to reach the optimum design. Despite their greater robustness, they require a great number of function evaluations and the cost increases with the number of design variable. Another alternative, according to the literature, consists to utilize a derivative-based optimization algorithms that are able to locate the optimum solutions in the expense of only few function evaluations (i.e. with reasonable CPU time). They are often supported by tools that compute the derivatives of the objective function; such as the adjoint techniques [9, 10], the Automatic Differentiation (AD) [11, 12], the surrogate models (Radial Basis Function(RBF), Artificial Neural Networks (ANN), etc.) [13].

We note that prior to us, we find many applications employing the derivative-based methods with Bézier parametrization. In particular, Marino et al. [14] has used triangular Bézier surface and the optimization is numerically

*e-mail: b.abouelmajd@fsac.ac.ma

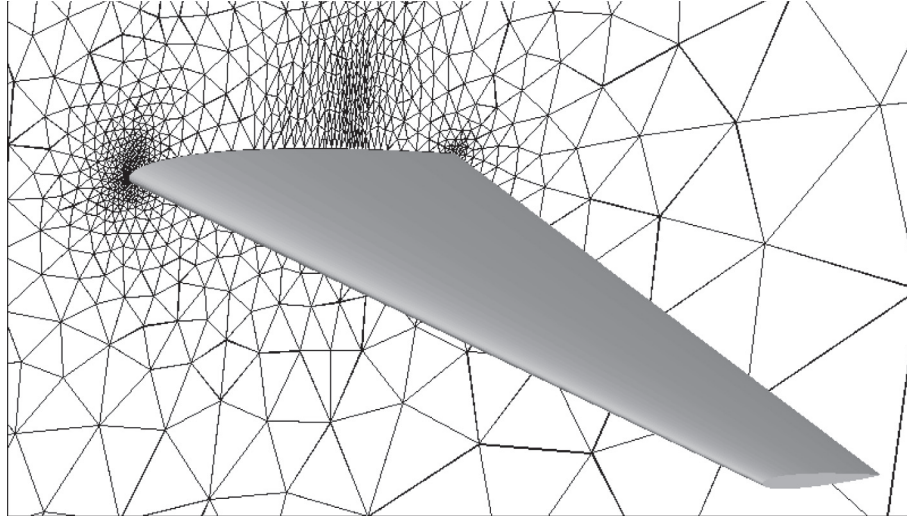


Figure 1. Initial wing shape and mesh in the symmetry plane.

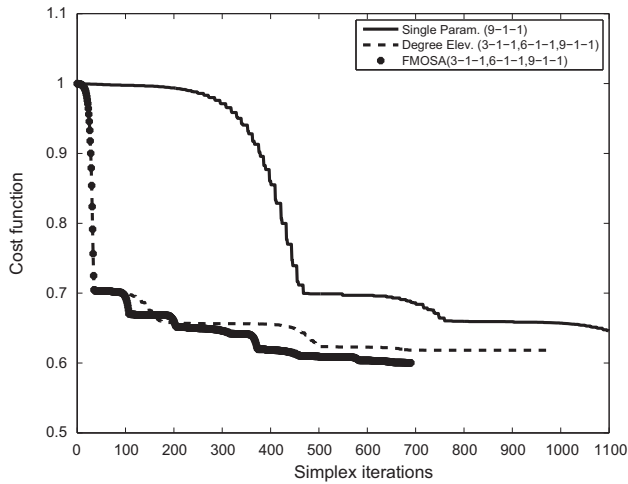


Figure 2. Comparison of the convergence history for the three strategies.

solved by a constrained gradient-based minimization algorithm. In this work, Marino et al. find the gradient-based algorithm more suitable than the evolutionary algorithm. In reference [15], we find a combination of NURBS (Non-Uniform Rational B-Spline) and a gradient-based optimization algorithm to conduct shape optimization of stratospheric airships.

In this paper, we propose to solve the shape optimisation problem, within multilevel and adaptive strategies, by the BFGS (Broyden Fletcher Goldfarb and Shanno) Quasi Newton method. One of the key properties of BFGS method is its superlinear convergence, which is guaranteed under reasonable assumptions [16]. In addition, it works well in practice, even when the initial iterate is far from the solution. Despite these advantages, the initialization of BFGS algorithm by identity may affect the performance of our hierarchical and adaptation strategies; this later depends on the quality of transfer operators in order to rigorously construct embedded and nested

search space. Thus, our proposed approach consists to well re-initialize the Hessian and the gradient at each change of the search parametric space.

This paper is organized as follows. First, we review the construction of multilevel algorithms, in the context of *parametric shape optimization*. Second, we present the proposed approach for multilevel and adaptive parametric shape optimization. We then present numerical results on a model problem from calculus of variations. Finally, we conclude and mention some perspectives.

2 Nested Bézier parameterizations for multilevel shape representation

We begin with the simplest situation of a two-dimensional geometry for which we employ a Bézier shape representation:

$$x(t) = \sum_{k=0}^n B_n^k(t) x_k, \quad y(t) = \sum_{k=0}^n B_n^k(t) y_k \quad (2)$$

in which the parameter t varies from 0 to 1, n is the degree of the parameterization,

$$B_n^k(t) = C_n^k t^k (1-t)^{n-k} \quad (3)$$

is a Bernstein polynomial, $C_k^n = \frac{n!}{k!(n-k)!}$, and

$$P_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix} \quad (k = 0, 1, \dots, n) \quad (4)$$

is the generic control point. The coordinates of these control points are split into two vectors

$$X = \{x_k\}, \quad Y = \{y_k\}, \quad k = 0, 1, \dots, n, \quad (5)$$

and we refer to the vector X as the *support* of the parameterization, and the vector Y as the *design* vector. Typically, we optimize the design vector for fixed support according to some physical criterion, such as drag reduction in aerodynamics. The somewhat unsymmetrical roles dispensed to the vectors X and Y are chosen to reduce (to n essentially)

the dimension of the search space in the optimization phase, which is the most numerically costly and subject to numerical stiffness.

We also use the notation:

$$x(t) = B_n(t)^T X, \quad y(t) = B_n(t)^T Y, \quad (6)$$

in which the vector $B_n(t)^T = (B_n^0(t), B_n^1(t), \dots, B_n^n(t))$. In all this article, only supports for which the sequence $\{x_k\}$ is monotone increasing are said to be admissible and considered throughout. Thus, the function $x(t)$ is monotone-increasing and defines a one-to-one mapping of, say, $[0,1]$ onto itself. Recall also the simple formula for the derivative:

$$\frac{dx(t)}{dt} = n \sum_{k=0}^{n-1} B_{n-1}^k(t) (x_{k+1} - x_k) = n B_{n-1}(t)^T \Delta X \quad (7)$$

in which Δ denotes the forward-difference operator ($\Delta x_k = x_{k+1} - x_k$) as well as the associated $n \times (n+1)$ matrix.

In the prototypical case of an airfoil, we use such a parametric representation for both the upper and lower surfaces separately. The vertical slope at the leading edge is enforced by the conditions:

$$x_0 = x_1 = 0, \quad y_0 = 0 \quad (8)$$

for both surfaces which assures a smooth match; at the trailing edge, we simply have:

$$x_n = 1, \quad y_n = 0 \quad (9)$$

for a continuous match.

Our geometrical construction employs the *degree-elevation process*, well-known in the *Computer-Aided Design* literature (see, e.g. [17]). This process permits to cast (Eq. (6)) into the following equivalent Bézier parameterization of degree $n+1$:

$$x(t) = \sum_{k=0}^{n+1} B_{n+1}^k(t) x'_k, \quad y(t) = \sum_{k=0}^{n+1} B_{n+1}^k(t) y'_k \quad (10)$$

in which the new control points $P'_k = (x'_k, y'_k)$ are obtained from the former by convex combinations:

$$P'_0 = P_0, P'_k = \frac{k}{n+1} P_{k-1} + \left(1 - \frac{k}{n+1}\right) \times P_k \quad (k = 1, 2, \dots, n), P'_{n+1} = P_n \quad (11)$$

obtained by multiplying equation (6) by $(1-t)^{n+1-k}$ and grouping together the monomials in $t^k(1-t)^{n+1-k}$, for each k .

From a theoretical viewpoint, our construction guarantees rigorously nested search spaces, and exact upward transfer operators (from low to high-degree parameterization). Note that, apart from the specified endpoints, the abscissas of the initial support X are not a subset of the abscissas of any support of a higher degree parameterization. Nevertheless, any Bézier curve given on the initial support can be expressed exactly on any other support of higher degree provided it results from the degree elevation process. The parameterizations are nested, or embedded in one another in this sense precisely.

Based on this concept of nested Bézier parameterizations, the following theorem introduces the upward and downward transfer operators.

Theorem 1. *Let $X' = \{x'_k\}$ and $X = \{x_k\}$ two nested (by degree elevation process) parametrization supports of degree $n' = n + 1$ and n respectively, t_i ($i = 0, \dots, n'$) and q_i ($i = 0, \dots, n$) two arbitrary partitions of the interval $[0, 1]$. Then, the upward relation (elevation) between X' and X is given by the relation:*

$$X' = (B_{n'})^{-1} B_{n',n} X \quad (12)$$

where the generalized Vandermonde matrix $B_{n'}$ and the rectangular matrix $B_{n',n}$ are:

$$B_{n'} = (B_{n'}^j(t_i)) \quad (i = 0, \dots, n'; j = 0, \dots, n') \quad (13)$$

$$B_{n',n} = (B_n^j(t_i)) \quad (i = 0, \dots, n'; j = 0, \dots, n) \quad (14)$$

and the downward relation (reduction) between X' and X is given by the relation:

$$X = (\bar{B}_n)^{-1} \bar{B}_{n,n'} X' \quad (15)$$

where the generalized Vandermonde matrix \bar{B}_n and the rectangular matrix $\bar{B}_{n,n'}$ are:

$$\bar{B}_n = (B_n^j(q_i)) \quad (i = 0, \dots, n; j = 0, \dots, n) \quad (16)$$

$$\bar{B}_{n,n'} = (B_{n'}^j(q_i)) \quad (i = 0, \dots, n; j = 0, \dots, n') \quad (17)$$

Proof. Let $X = \{x_k\}$ and $X' = \{x'_k\}$ two nested Bézier parametrization supports of degree n' and n respectively. It means that:

$$B_n(t)^T X = B_{n'}(t)^T X' \quad \text{for each } t \in [0, 1] \quad (18)$$

Let $\{t_i\}$ ($i = 0, \dots, n'$) arbitrary partition of $[0,1]$ and $\{e_j\}$ ($i = 0, \dots, n'$) a canonic base of $\mathbb{R}^{n'}$. We first multiply the equation (18) by e_k ($k = 0, \dots, n'$), then:

$$e_k B_n(t_k)^T X = e_k B_{n'}(t_k)^T X' \quad (19)$$

and by adding these equations (19) with respect to k , we find the following formula:

$$B_{n'} X' = B_{n',n} X \quad (20)$$

where

$$B_{n'} = (B_{n'}^j(t_i)) \quad (i = 0 \dots, n'; j = 0 \dots, n') \quad (21)$$

and

$$B_{n',n} = (B_n^j(t_i)) \quad (i = 0 \dots, n'; j = 0 \dots, n) \quad (22)$$

Since the generalized Vandermonde matrix $B_{n'}$ is invertible, it follows that:

$$X' = (B_{n'})^{-1} B_{n',n} X \quad (23)$$

by the same way, we prove the second statement (15) by interchanging n and n' . \square

In what follows, we note by E_n^N the upward operator from the lower level n to N ($N > n$) and by R_n^N the downward operator. By construction, we have this following property:

$$R_n^N E_n^N = I_n \quad (24)$$

To test these two transfer operators, we use the following model problem:

$$\min_Y j_n(Y) = \frac{p^\alpha}{\mathcal{A}} \quad (25)$$

in which $x(t)$ is given, smooth and monotone increasing,

$$p = \int_0^1 \sqrt{x'(t)^2 + y'(t)^2} \omega(t) dt, \quad (26)$$

$$\mathcal{A} = \int_0^1 y(t)x'(t)\omega(t)dt \quad (\omega > 0, \alpha > 1)$$

and

$$x(t) = B_n(t)^T X, \quad y(t) = B_n(t)^T Y, \quad (27)$$

p and \mathcal{A} are, for specified $\omega(t) > 0$ and $\alpha > 1$, the pseudo-length of the arc, and the pseudo-area below the arc. This model problem has been studied in [18] to which we refer for a full description of the numerical test-case which corresponds to $\alpha \approx 2.03$, for the functional is known to be convex, and a certain $\omega(t)$ for which the minimizing shape is the half-thickness distribution of the RAE2822 airfoil.

Figure 3a (resp. 3b) illustrates the upward process using E_6^8 (resp. the downward process R_6^8). The curves corresponding to the both parametrizations are superimposed.

3 Hessian and gradient transfer

In this section, we present our approach within a multilevel algorithms and adaption procedure. The subject is to good initialize the optimization process when the parametric search space is changed (Sect. 3.1) and the control polygon is regularized (Sect. 3.2).

3.1 First case

As explained in the previous section, the transfer operators guarantee the construction of nested control polygon with higher degree ($N > n$). To achieve a good transfer of informations, we need to better update the Hessian and the gradient in order to conserve the efficiency of our multilevel strategies.

For fixed support X^0 , let (X^0, Y^0) the solution of the minimization of the problem (25) and γ_0 the corresponding shape. By using the upward operator, we construct a new control polygon (X, Y) of degree N corresponding to the same shape γ_0 . It follows that:

$$\begin{cases} E_n^N X^0 = X \\ E_n^N Y^0 = Y \end{cases} \quad (28)$$

and

$$j_n(Y = E_n^N Y^0) = j_n(Y^0) \quad (29)$$

where j_N is the cost function of the problem (25) corresponding to the Bézier parametrization of degree N .

To express the Hessian and the gradient on the coarse level (of degree n), we differentiate two times the equation (29) with respect to Y^0 :

$$\frac{dj_n(Y^0)}{dY^0} = \frac{dj_n(Y)}{dY} E_n^N \quad (30)$$

$$\frac{d^2 j_n(Y^0)}{d^2 Y^0} = (E_n^N)^T \frac{d^2 j_n(Y)}{d^2 Y} (E_n^N) \quad (31)$$

where $\frac{dY}{dY^0} = E_n^N$.

By differentiating the equation (28) with respect to Y , we find:

$$E_n^N \frac{dY^0}{dY} = I_N \quad (32)$$

it follows that

$$\frac{dj_n(Y)}{dY} = \frac{dj_n(Y^0)}{dY^0} R_n^N \quad (33)$$

and

$$\frac{d^2 j_n(Y)}{d^2 Y} = (R_n^N)^T \frac{d^2 j_n(Y^0)}{d^2 Y^0} (R_n^N) \quad (34)$$

where $R_n^N = \frac{dY^0}{dY}$. We note that (33) and (34) can also be obtained using the property (24).

Proposition 1. *The Hessian matrix given by (34) is positive definite.*

Proof. Let

$$B_n = \frac{d^2 j_n(Y^0)}{d^2 Y^0} \quad \text{and} \quad B_N = \frac{d^2 j_N(Y)}{d^2 Y} \quad (35)$$

The matrix B_n is the BFGS approximation corresponding to the first optimization phase on the coarse level; thus, it is positive definite.

Let $x \in \mathbb{R}^N$ ($x \neq 0$), then

$$\begin{aligned} x^T B_N x &= x^T (R_n^N) B_n (R_n^N)^T x \\ &= W_n^T B_n W_n \end{aligned} \quad (36)$$

where $W_n = (R_n^N)^T x$.

Since $W_n \neq 0$ and B_n is positive definite, then $x^T B_N x > 0$. It follows that B_N is positive definite. \square

As consequence, the following expression holds:

$$\left(\frac{d^2 j_n(Y)}{d^2 Y} \right)^{-1} = (E_n^N) \left(\frac{d^2 j_n(Y^0)}{d^2 Y^0} \right)^{-1} (E_n^N)^T \quad (37)$$

Finally, we can update the Hessian by the matrices B_N or B_N^{-1} , it depends on the BFGS method used, instead of identity when the parametrization degree is augmented or reduced. The second phase of the optimization in the fine level of degree N is initialized by an exact Hessian matrix.

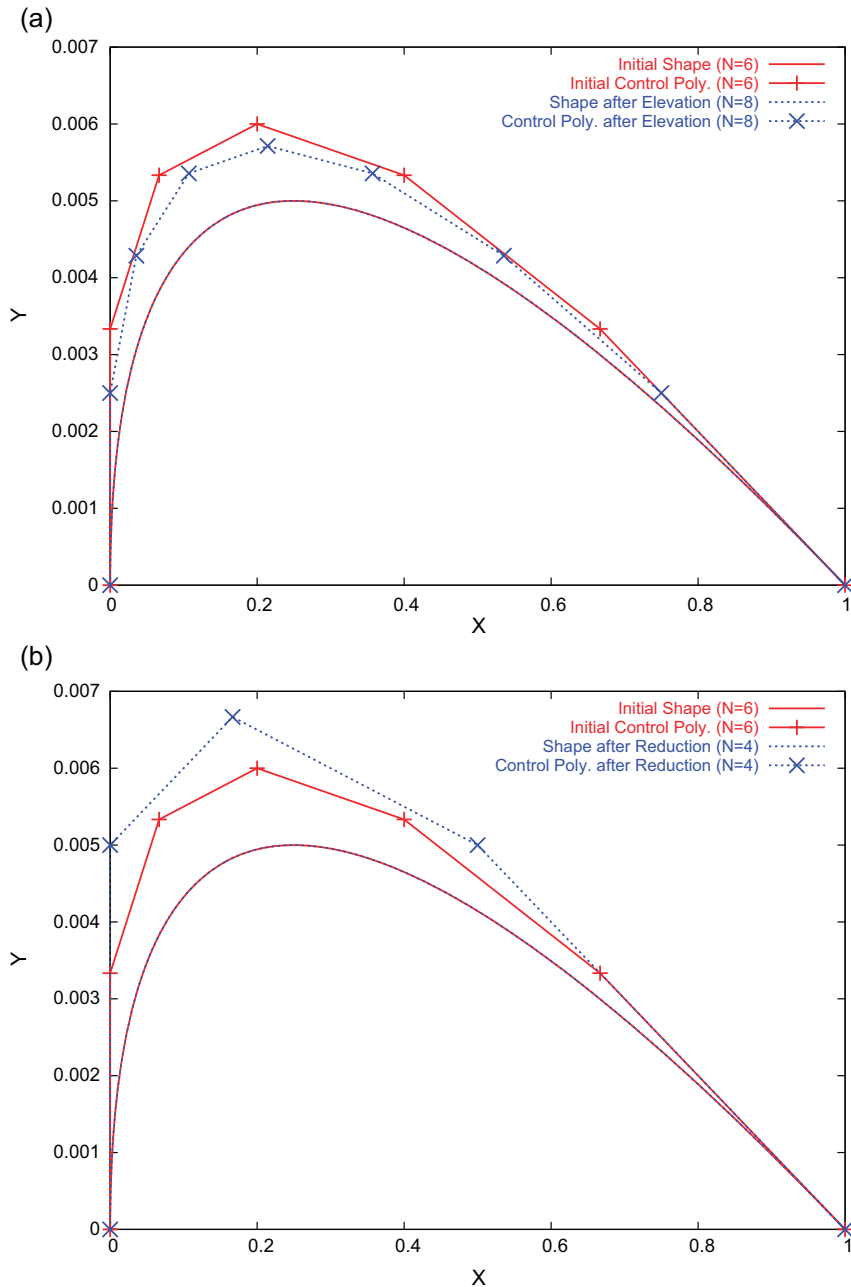


Figure 3. Illustration of E_n^N and R_n^N by using two embedded control polygons of degree $n = 6$ and $N = 8$. (a) Upward process, (b) downward process.

3.2 Second case

Suppose that we are placed on the search space of degree n . the parametrisation adaption consists to adapt the support X by alternating two complementary phases:

1. Optimization: optimize the design vector Y for fixed support $X = X^0$ according to some criterion; let Y^0 be the result of this phase.
2. Regularization: given the parametrization (X^0, Y^0) of an approximate optimum shape γ^0 , the new support X^1 is taken to be the better support for which the total

variation (TV) in the components of the corresponding vector Y^1 is minimal, such that the correspondent shape γ^1 to (X^1, Y^1) approximates γ^0 in the sense of least squares; substitute X^1 to X^0 .

This adaption procedure has been studied extensively in [3] for two-dimensional parametrization. The extension to three-dimensional optimum design in aerodynamics within the framework of the so-called free-form deformation has been proposed in reference [19]. It improves noticeably the convergence rate and reduces the numerical stiffness. The main idea

consists to redefine the geometrical representation by using regularization techniques. Thus, a new formulation of the Hessian and the gradient is required in order to re-initialize the adaption process by a good regularized solution with equivalent performance.

As the both control polygons (X^0, Y^0) and (X^1, Y^1) correspond to the same performance, it follows that:

$$\mathcal{J}(\gamma^0) = \mathcal{J}(\gamma^1) \quad (38)$$

i.e.

$$\mathcal{J}([B_n(t)^T X^0, B_n(t)^T Y^0]_0^1) = \mathcal{J}([B_n(t)^T X^1, B_n(t)^T Y^1]_0^1) \quad (39)$$

This intrinsic formulation can be transformed into a parametric optimization by:

$$\mathcal{J}_n^X(Y) = \int_0^1 [B_n(t)^T Y - \bar{y}_T(t, X)]^2 B_{n-1}(t)^T \Delta X dt \quad (40)$$

where $\bar{y}_T(t, X) = B_n(\tau)^T Y^0$ and $\tau = \tau(t, X)$ is related to the change of support $X^0 \rightarrow X$, and defined by this condition:

$$B_n(\tau)^T X^0 = B_n(t)^T X \quad (41)$$

To satisfy the unicity of the solution, we impose that the components of X^0 are monotone increasing.

The derivative of the cost function \mathcal{J}_n^X is given by:

$$\begin{aligned} \left(\frac{\partial \mathcal{J}_n^X}{\partial Y} \right)^T &= \int_0^1 B_n(t) [B_n(t)^T Y - \bar{y}_T(t, X)] n B_{n-1}(t)^T \Delta X dt \\ &= A(X) Y - b(X) \end{aligned} \quad (42)$$

where

$$A(X) = \int_0^1 B_n(t) B_n(t)^T n B_{n-1}(t)^T \Delta X dt \quad (43)$$

and

$$b(X) = \int_0^1 B_n(t) \bar{y}_T(t, X) n B_{n-1}(t)^T \Delta X dt \quad (44)$$

The matrix $A(X)$ is real-symmetric positive definite and depends linearly upon the vector X , thus

$$A(X) = \mathcal{A} \otimes X \quad (45)$$

where $\mathcal{A} = A'(X)$ is a tensor of order 3, independent of X and \otimes stands for the contracted product.

The minimization of \mathcal{J}_n^X with respect to Y is equivalent to $\frac{\partial \mathcal{J}_n^X}{\partial Y} = 0$. It follows that:

$$A(X) Y = b(X) \quad (46)$$

In particular, (X^0, Y^0) and (X^1, Y^1) verify this linear system.

For $X = X^0$, the first and second derivative of (40) with respect to Y^0 are given by:

$$\frac{d \mathcal{J}_n^{X^0}(Y^0)}{dY^0} = \frac{d \mathcal{J}_n^{X^1}(Y^1)}{dY^1} \frac{dY^1}{dY^0} \quad (47)$$

and

$$\begin{aligned} \frac{d^2 \mathcal{J}_n^{X^0}(Y^0)}{d^2 Y^0} &= \left(\frac{dY^1}{dY^0} \right)^T \frac{d^2 \mathcal{J}_n^{X^1}(Y^1)}{d^2 Y^1} \left(\frac{dY^1}{dY^0} \right) \\ &+ \frac{d \mathcal{J}_n^{X^1}(Y^1)}{dY^1} \frac{d^2 Y^1}{d^2 Y^0} \end{aligned} \quad (48)$$

By differentiating the linear system (46) with respect to Y^0 , we obtain:

$$A(X^1) \frac{dY^1}{dY^0} = \frac{db(X^1)}{dY^0} \quad (49)$$

where

$$\frac{db(X^1)}{dY^0} = \int_0^1 B_n(t) B_n(\tau)^T n B_{n-1}(t)^T \Delta X dt \quad (50)$$

As $A(X^1)$ and $\frac{db(X^1)}{dY^0}$ do not depend on Y^0 , then:

$$A(X^1) \frac{d^2 Y^1}{d^2 Y^0} = 0 \quad (51)$$

Proposition 2. *If all the components of the support X are monotone increasing, then the matrix $A(X)$ given by the formula (43) is positive definite.*

Proof. Let $Y \in \mathcal{R}^n$ ($Y \neq 0$). We have

$$\begin{aligned} Y^T A(X) Y &= \int_0^1 Y^T B_n(t) B_n(t)^T Y n B_{n-1}(t)^T \Delta X dt \\ &= \int_0^1 Z^T Z n B_{n-1}(t)^T \Delta X dt \end{aligned} \quad (52)$$

where $Z = B_n(t)^T Y$.

As the components of X are monotone increasing, then $\Delta X > 0$ (Δ is the forward difference operator). So, its enough to prove that $Z^T Z \neq 0$. We proceed by contradiction and assume that $Z^T Z = 0$, so $Z = 0$. As $B_n(t)^T$ form a basis of \mathbb{R}^n , then $Y = 0$, which contradicts the first assumption. \square

Since X^1 satisfy the condition of the Proposition 2, then the matrix $A(X^1)$ is invertible. Thus:

$$\frac{d^2 Y^1}{d^2 Y^0} = 0 \quad (53)$$

As consequence,

$$\frac{d^2 \mathcal{J}_n^{X^1}(Y^1)}{d^2 Y^1} = \left(\frac{dY^1}{dY^0} \right) \frac{d^2 \mathcal{J}_n^{X^0}(Y^0)}{d^2 Y^0} \left(\frac{dY^1}{dY^0} \right)^T \quad (54)$$

and

$$\left(\frac{d^2 \mathcal{J}_n^{X^1}(Y^1)}{d^2 Y^1} \right)^{-1} = \left(\frac{dY^1}{dY^0} \right) \left(\frac{d^2 \mathcal{J}_n^{X^0}(Y^0)}{d^2 Y^0} \right)^{-1} \left(\frac{dY^1}{dY^0} \right)^T \quad (55)$$

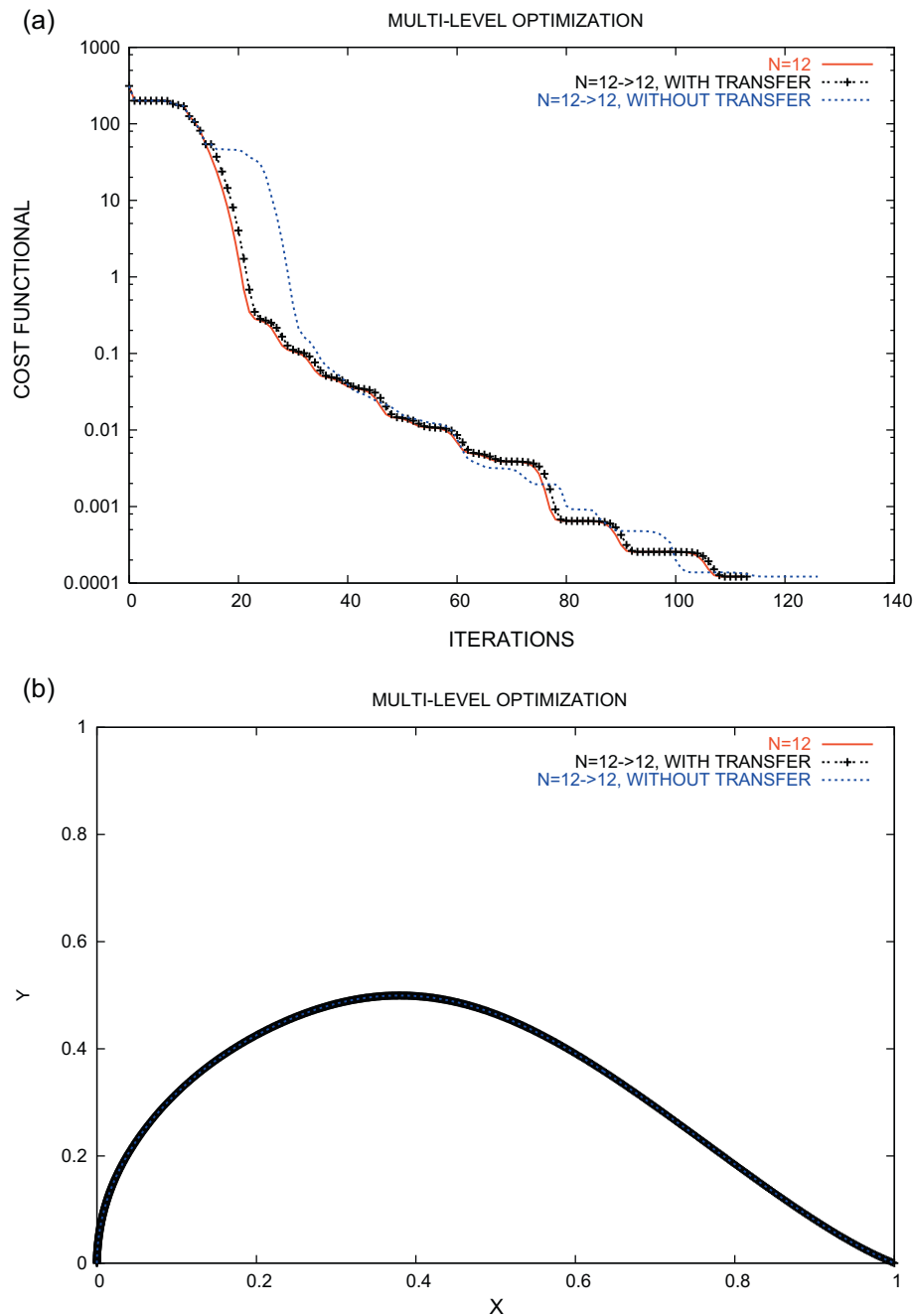


Figure 4. Basic algorithm for $N = 12$; Comparison between three strategies: full convergence, with Hessian transfer, without Hessian transfer. (a) Iterative performance, (b) optimum shape.

The formulas express the relation between the Hessian matrices corresponding to the initial control polygon (X^0, Y^0) and the regularized one (X^1, Y^1) . It permits to re-initialize the optimization process after each parametrization adaption.

4 Numerical experiments

The proposed approach is tested using the model problem, described in Section 3.1, for the first case and the

shape-reconstruction problem, described in Section 3.2, for the second one. The numerical optimum is determined by the Quasi-Newton algorithm [20] based on BFGS (Broyden-Fletcher-Goldfarb-Shanno) updated formula. This algorithm is a line search family method, one of the most powerful methods to solve unconstrained optimization problem. It exposes superlinear convergence; resource-intensity is estimated as $O(n^2)$ per iteration for n -component argument vector. The search direction is given by $s^k = -\mathbf{H}^k \mathbf{G}^k$, where \mathbf{G}^k and \mathbf{H}^k are respectively the gradient and the Hessian of the

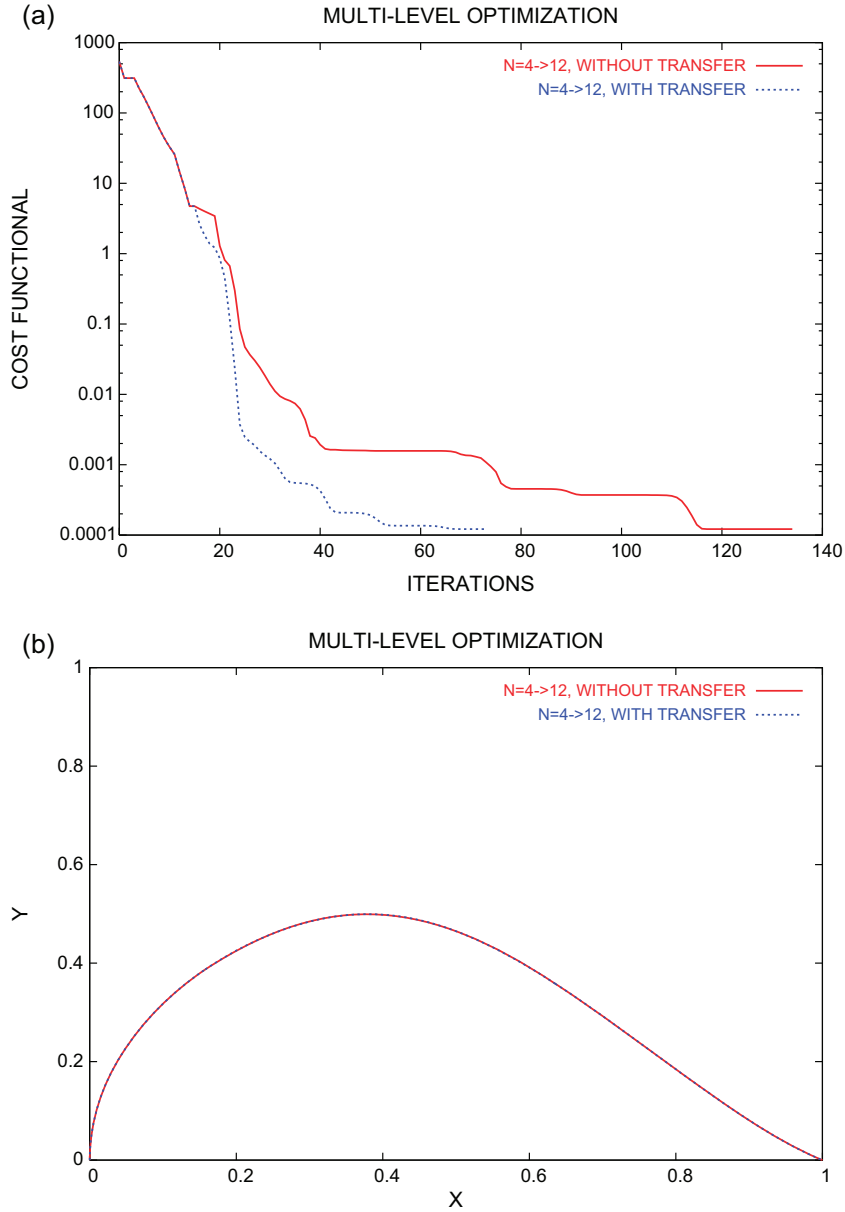


Figure 5. Multilevel algorithm for $n = 4$ and $N = 12$; Comparison between two strategies: with Hessian transfer, without Hessian transfer. (a) Iterative performance, (b) optimum shape.

objective function. \mathbf{H}^k a symmetric definite matrix, constructed at each iteration, which approximates the Hessian matrix and/or its inverse. In the literature, there are several formulas for the calculation of \mathbf{H}^{k+1} from \mathbf{H}^k . We use here, the BFGS update formula [16],

$$\mathbf{H}^{k+1} = \mathbf{H}^k - \frac{d^k (y^k)^T \mathbf{H}^k + \mathbf{H}^k y^k (d^k)^T}{(y^k)^T d^k} + \left(\frac{1 + (y^k)^T \mathbf{H}^k y^k}{(y^k)^T d^k} \right) \frac{d^k (d^k)^T}{(d^k)^T y^k} \quad (56)$$

where $y^k = \mathbf{G}^{k+1} - \mathbf{G}^k$ and $d^k = x^{k+1} - x^k$.

As a starting point, \mathbf{H}^0 can be set to any symmetric positive definite matrix and very often the identity matrix.

This initialization has a bad impact on the efficiency of our multilevel and adaption strategies.

In the following numerical experiments, the gradient is calculated analytically and we focus only on the impact of the Hessian transfer. For the model problem, the derivative of the cost function in the search space of degree n is given by:

$$\frac{d\mathcal{J}_n(Y)}{dY} = \frac{p^{\alpha-1} (\alpha \mathcal{A} \frac{\partial p}{\partial Y} - p \frac{\partial \mathcal{A}}{\partial Y})}{\mathcal{A}^2}$$

with

$$\frac{\partial p}{\partial Y} = \int_0^1 \frac{n B_{n-1}^T(t) y'(t) \Delta}{\sqrt{x'^2(t) + y'^2(t)}} dt$$

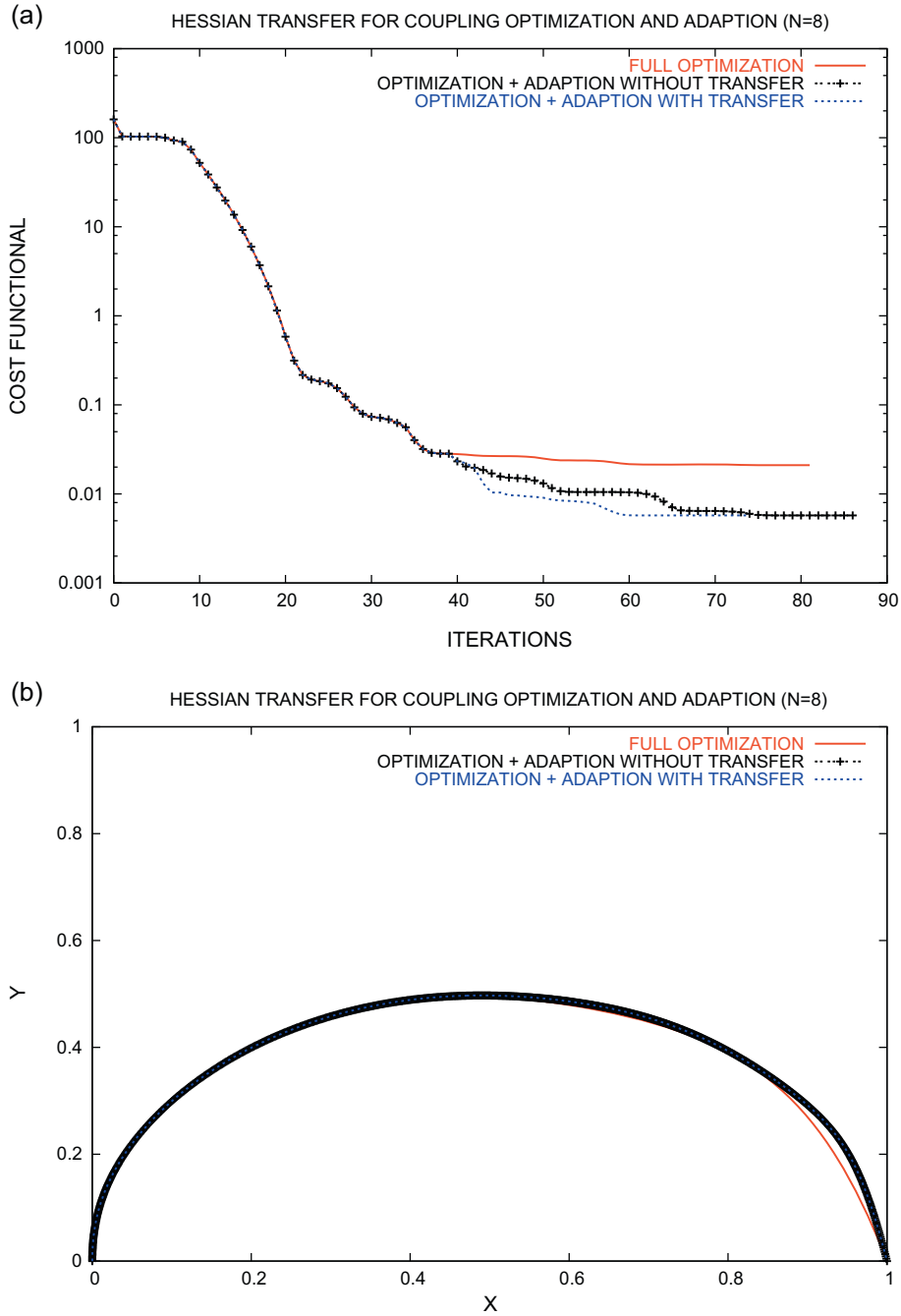


Figure 6. Adaption algorithm for $n = 8$; Comparison between two strategies: with Hessian transfer, without Hessian transfer. (a) Iterative performance, (b) optimum shape.

and

$$\frac{\partial \mathcal{A}}{\partial Y} = \int_0^1 B_n^T(t) x'(t) \omega(t) dt$$

We use here the formula of the derivative of $y'(t)$ given by $nB_{n-1}^T \Delta Y$, in which Δ is the $n \times (n + 1)$ matrix associated with the forward-difference operator ($\Delta Y = y_{k+1} - y_k$).

For the shape-reconstruction problem, the gradient corresponds to the linear system (46):

$$\frac{d \mathcal{J}_n^X(Y)}{dY} = A(X) Y - b(X)$$

To Calculate the gradient on the search space of degree $N > n$ (resp. $N < n$), we apply the upward operator E_n^N (resp. the downward operator R_n^N) to X and Y and we substitute.

Figure 4 provides the convergence history of the basic algorithm for three optimization strategies: at full convergence, with Hessian transfer after 20 iterations and without Hessian transfer (the Hessian is initialized automatically by identity).

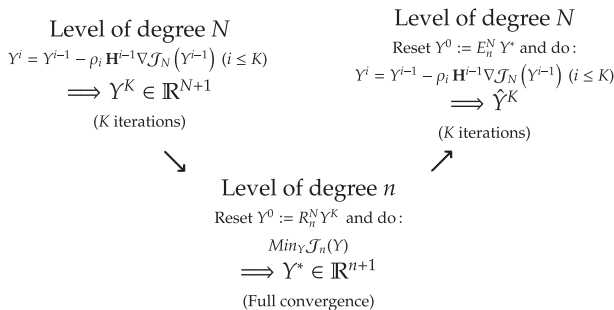
And Figure 4 gives the corresponding optimal shapes. Clearly, all the optimization achieve the convergence and the main point is the fact that the strategy based on Hessian transfer has no effect on the convergence rate; so, this experiment validate the formulas of the Hessian and gradient.

In this experiment, the multilevel optimization is considered here for $n = 4$ and $N = 12$. Figures 5a and 5b give respectively the iterative performance and the corresponding optimal shapes for the multilevel algorithm with and without Hessian transfer. Clearly, the algorithm based on Hessian transfer is faster in term of the convergence rate. This experiment confirms the impact on the performance of the multilevel algorithm when the Hessian matrix is initialized by identity.

Lastly, we measure the effect of the hessian transfer on the accuracy of the optimization when using the adaption procedure. Figure 6a shows a comparison of the basic algorithm for $n = 8$, with the adaption procedure with and without Hessian transfer. The Figure 6b gives the corresponding optimal shape. As expected, the Hessian transfer guarantees a best convergence rate of the optimization.

5 Conclusion

The Hessian transfer is essential to preserve and enhance the quality of our multilevel strategies and adaption procedure. This approach permits to better initialize the BFGS Quasi-Newton method. The numerical experiments demonstrate clearly its efficiency in the case of two-dimensional model problem. This study, encourage developing a new multilevel geometrical structure in which, for example in the case of two-level ideal V-cycle [22, 23], the problem is solved to complete convergence on the coarse level by robust optimization algorithms (Genetic algorithm [GA], Particle swarm optimization [PSO], simplex method, etc.) and few iterations are enough (acts as preconditionners) in the fine level to alleviate the numerical stiffness; BFGS Quasi-Newton can be used in this case. This variant is illustrated by the following schematic. Others strategies can be elaborated mimicking the multigrid-type strategies (saw-tooth, V and W cycle, FMG, etc.) [21].



The proposed approach will be generalized for 3D optimum design in aerodynamics, structural mechanics, or coupled disciplines within the framework of the so-called

Free-Form deformation (FFD) [24]. Testing a great number of promising algorithmic strategies is also a part of our ongoing work.

References

1. Abou El Majd B, Desideri J-A, Habbal A. 2010. Aerodynamic and structural optimization of a business-jet wingshape by a Nash game and an adapted split of variables. *Mécanique & Industries*, 11(3-4), 209-214.
2. Ouchetto O, Abou El Majd B, Ouchetto H, Essakhi B, Zouhdi S. 2016. Homogenization of periodic structured materials with chiral properties. *IEEE Transactions on Antennas and Propagation*, 64(5), 1751-1758.
3. Abou El Majd B, Désidéri J-A, Janka A. 2004. Nested and self-adaptive Bézier parameterizations for shape optimization, *International Conference on Control, Partial Differential Equations and Scientific Computing (dedicated to late Professor J.-L. Lions)*, Beijing, China, 13-16, September, 2004.
4. Abou El Majd B, Désidéri J-A, Duvigneau R. 2008. Multilevel strategies for parametric shape optimization in aerodynamics. *European Journal of Computational Mechanics*, 17(1-2).
5. Désidéri J-A, Duvigneau R, Abou El Majd B, Tang Z. 2007. Algorithms for efficient shape optimization in aerodynamics and coupled disciplines. 42nd AAAF Congress on Applied Aerodynamics, Sophia-Antipolis, France.
6. Abou El Majd B, Désidéri JA, Do TT, Fourment L. 2005. Multilevel strategies and hybrid methods for shape optimization and application to aerodynamics and metal forming. *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems Conference (EUROGEN 2005)*, 2005, September, pp. 12-14.
7. Duvigneau R, Abou El Majd B, Désidéri J-A. 2008. Towards a self-adaptive parameterization for aerodynamic shape optimization. *ESAIM: Proceedings*. Vol. 22, pp. 169-174, EDP Sciences, 2008.
8. Duvigneau R, Abou El Majd B, Désidéri J-A. 2008. Aerodynamic design using hierarchical shape parameterizations for descent and Particle Swarm Optimization Methods. *Numerical Analysis and Scientific Computing for Partial Differential Equations and Their Challenging Applications*. CIMNE.
9. Giannakoglou KC, Papadimitriou DI. 2008. Adjoint Methods for Shape Optimization. *Archives of Computational Methods in Engineering (State of the Art Reviews)*, 15(4), 447-488.
10. Jameson A, Hu R, Wang Q. 2012. Adjoint-based aerodynamic optimization of supersonic biplane airfoils. *Journal of Aircraft*, 49(3), 802-814.
11. Courty F, Dervieux A, Dervieux B, Hascoet L. 2003. Hascoet, Reverse automatic differentiation for optimum design: from adjoint state assembly to gradient computation. *Optimization Methods and Software*, 18(5), 615-627.
12. Hascoet L. 2004. TAPENADE: A Tool for Automatic Differentiation of Programs. *Proceedings of the ECCOMAS Conference*, Jyväskylä, Finland.
13. Duvigneau R, Chandrashekar P. 2012. Kriging-based optimization applied to flow control. *Int. J. for Numerical Methods in Fluids*, 69(11), 1701-1714.

14. Marino E, Salvatori L, Orlando M, Borri C. 2016. Two shape parametrizations for structural optimization of triangular shells. *Computers & Structures*, 166, 1–10.
15. Marino E, Salvatori L, Orlando M, Borri C. 2016. Two shape parametrizations for structural optimization of triangular shells. *Comput. Struct.*, 166, 1–10.
16. Nocedal J, Wright SJ. 2006. *Numerical Optimization*, 2nd edn. Springer-Verlag: Berlin, Germany.
17. Farin G. 1990. *Curves and surfaces for computer-aided geometric design – A practical guide*, Rheinboldt W, Siewiorek D, Editors. Academic Press, Boston.
18. Désidéri J-A, Zolésio J-P. 2005. Inverse shape optimization problems and application to airfoils. *Control and Cybernetics*, 34(1), 165.
19. Abou El Majd B. 2014. Parameterization adaption for 3D shape optimization in aerodynamics. *International Journal of Science and Engineering*, 6(1), 61–69.
20. Nielsen HB. 2000. UCMINF – An Algorithm For Unconstrained, Nonlinear Optimization, Informatics and Mathematical Modelling (IMM). Technical University of Denmark.
21. Wesseling P. 1992. *An introduction to Multigrid Methods*. John Wiley & Sons: Chichester.
22. Zhao J, Abou El Majd B, Desideri JA. 2015. Two level correction algorithm for parametric shape inverse optimization. *International Journal of Engineering and Mathematical Modelling*, 2(1), 17–30.
23. Zhao J, Abou El Majd B, Desideri JA. 2007. Two level correction algorithms for model problems. INRIA.
24. Sederberg T, Parry S. 1986. Free-from deformation of solid geometric models. *ACM SIGGRAPH Computer Graphics*, 20(4), 151–160.

Cite this article as: El Majd BA, Ouchetto O, Désidéri JA & Habbal A: Hessian transfer for multilevel and adaptive shape optimization. *Int. J. Simul. Multisci. Des. Optim.*, 2017, 8, A9.