Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 224 (2007) 117–131

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Nested and self-adaptive Bézier parameterizations for shape optimization

Jean-Antoine Désidéri [a,*], Badr Abou El Majd [a], Aleš Janka [b,1]

[a] *Opale Project–Team, INRIA, 2004 Route des Lucioles, BP 93, F–06902 Sophia Antipolis Cedex, France*
[b] *Département de Mathématiques, Université de Fribourg, CH-1700 Fribourg-Pérolles, Switzerland*

Received 26 September 2006; received in revised form 21 December 2006; accepted 22 December 2006
Available online 10 January 2007

Abstract

This article is a sequel of [J.-A. Désidéri, Hierarchical optimum-shape algorithms using embedded Bézier parameterizations, in: Y. Kuznetsov et al., (Ed.), Numerical Methods for Scientific Computing, Variational Problems and Applications, CIMNE, Barcelona, 2003], in which we defined formally a hierarchical shape optimization method based on a multi-level shape representation by nested Bézier parameterizations (*FAMOSA*), and [J.-A. Désidéri, A. Janka, Multi-level shape parameterization for aerodynamic optimization – application to drag and noise reduction of transonic/supersonic business jet, in: E. Heikkola et al., (Ed.), European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004), Jyväskyla, 24–28 July 2004] where we conducted some preliminary numerical experiments of shape optimization in aerodynamics. Here, we are testing the full multi-level optimum-shape algorithm (analogous in logical structure to the classical full multigrid method). Second, we propose a technique for parameterization self-adaptivity. Both methodological enhancements are assessed by novel numerical experiments on an inverse shape model problem, confirming both are very effective.

MSC: 65C20; 76N25; 65C17

Keywords: Partial–differential equations; Computational methods; Numerical shape optimization; Compressible aerodynamics

1. Introduction

Piet Wesseling has made major contributions to the theory, applications and education of computational methods, particularly in the areas of fluid dynamics and multigrid methods [1]. In this article, we propose a

[*] Corresponding author. Tel.: +33 4 92 38 77 93; fax: +33 4 92 38 79 80.

E-mail addresses: Jean-Antoine.Desideri@sophia.inria.fr (J.-A. Désidéri), Badr.AbouElMajd@sophia.inria.fr (B. Abou El Majd), ales.janka@unifr.ch (A. Janka).

URLs: http://www-sop.inria.fr/opale/ (J.-A. Désidéri), http://www-sop.inria.fr/opale/ (B. Abou El Majd), http://commonweb.unifr.ch/math/ (A. Janka).

[1] Formerly at INRIA Sophia Antipolis, Opale Project–Team.

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.12.016

technique to extend classical multi-level strategies to parametric shape optimization, with a particular emphasis on optimum-shape design in compressible aerodynamics.

We focus on certain methodological questions in numerical shape optimization when a partial–differential equation (PDE) is solved as a state equation. Ultimately, we aim at enhancing the computational efficiency of optimum-shape algorithms in aerodynamics, in which for example, the three-dimensional Euler equations for compressible perfect gas are often used to optimize the aerodynamic coefficients of aircraft configurations. Whence, each evaluation of the cost functional is computationally costly, and improving the optimization algorithm convergence rate is a major concern.

In aerodynamics, evaluating the cost functional gradient raises non-trivial theoretical questions [12] since the flow is a weak solution. Additionally, it is another computational endeavor, which can be realized, however, by solving a discrete approximate adjoint-equation as in [14], or by means of automatic differentiation as in [5,13]. Hessians are still seldom computed, and typical gradient-based methods employ unsophisticated procedures for stepsize adjustment. More often, even more rustic optimization techniques, such as the classical *simplex search method*, or the less conventional *evolutionary algorithms* (EAs), including the *genetic algorithms* (GAs) and the *particle swarm optimizers* (PSOs), are often preferred to achieve greater *robustness* in complex nonlinear situations in which convergence, and even sometimes the optimization problem formulation itself may be problematic otherwise. In our applications, we favor the simplex method which seems to realize, for moderately complex optimization problems, an adequate compromise between simplicity, robustness and computational performance.

Our research direction for improving the convergence rate of the optimization iteration relies on the particular handling of the geometrical shape to be optimized in a *multi-scale algorithm*. The method was originally introduced in [6], where we proposed to construct a hierarchy of embedded (or nested) Bézier parameterizations to serve as a multi-level support to the shape optimization algorithm.

For example, in two-dimensional cases, we consider planar curves represented by Bézier parameterizations of the form:

$$x(t) = \sum_{k=0}^{n} B_n^k(t) x_k, \quad y(t) = \sum_{k=0}^{n} B_n^k(t) y_k \tag{1}$$

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

$$B_n^k(t) = C_n^k t^k (1-t)^{n-k} \tag{2}$$

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

$$P_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix} \quad (k = 0, 1, \ldots, n) \tag{3}$$

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

$$X = \{x_k\}, \quad Y = \{y_k\}, \quad k = 0, 1, \ldots, n \tag{4}$$

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:

$$x(t) = B_n(t)^{\mathrm{T}} X, \quad y(t) = B_n(t)^{\mathrm{T}} Y \tag{5}$$

in which the vector $B_n(t)^{\mathrm{T}} = \left(B_n^0(t), B_n^1(t), \ldots, B_n^n(t)\right)$. 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:

$$x'(t) = n \sum_{k=0}^{n-1} B_{n-1}^k(t)(x_{k+1} - x_k) = n B_{n-1}(t)^{\mathrm{T}} \Delta X \tag{6}$$

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

Our geometrical construction employs the *degree-elevation process*, well-known in the *computer-aided design* literature (see for example [3]). This process permits to cast (1) into the following equivalent Bézier parameterization of degree $n+1$:

$$x(t) = \sum_{k=0}^{n+1} B_{n+1}^{k}(t)x'_k, \quad y(t) = \sum_{k=0}^{n+1} B_{n+1}^{k}(t)y'_k \tag{7}$$

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

$$P'_0 = P_0, \quad P'_k = \frac{k}{n+1}P_{k-1} + \left(1 - \frac{k}{n+1}\right)P_k(k = 1, 2, \ldots, n), \quad P'_{n+1} = P_n \tag{8}$$

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

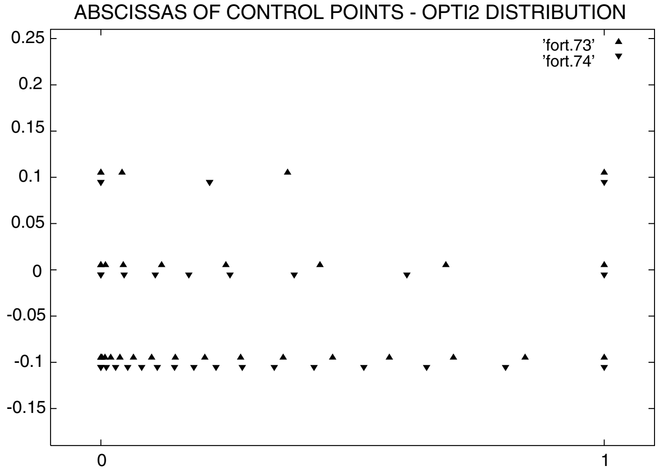

Fig. 1. One-dimensional example of embedded parameterizations: the triangles represent the supports X of three nested Bézier parameterizations of degree 4, 8 and 16 of an RAE2822 airfoil obtained from the first by 4 and 12 successive degree elevations; the symbols pointing upward (respectively downward) are associated with the upper (respectively lower) surface; the degree 4 support has been optimized to regularize the control polygon associated with the degree 16 airfoil representation (see [6], Fig. 3).

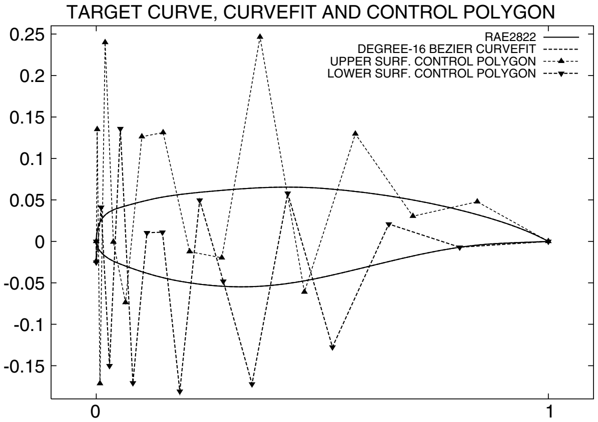

Fig. 2. RAE2822 airfoil, examples of degree 16 Bézier curvefits of the upper and lower surfaces (superimposed), and corresponding control polygons.

From a theoretical viewpoint, our construction guarantees rigorously nested search spaces, and exact upward transfer operators (from low to high-degree parameterization). This is illustrated in Fig. 1 in which the supports of three nested parameterizations of an airfoil are sketched.

Note that in the case of Fig. 1, apart from the specified endpoints, the abscissas of the degree-4 support X are not a subset of the abscissas of any support of a higher degree parameterization. Nevertheless, any Bézier curve given on the degree-4 support can be expressed exactly on any other support of higher degree provided it results, as in this example, from the degree elevation process. The parameterizations are nested, or embedded in one another in this sense precisely.

Fig. 2 represents the RAE2822 airfoil and the lower and upper control polygons of degree-16 Bézier least-squares curvefits. The RAE2822 airfoil is a classical geometry in computational aerodynamics, known for its low-drag performance in the transonic regime. This shape has been tabulated by the European Project ECARP [4].

By introducing a conceptual parallel from grid (for the purpose of *solving* a PDE) to geometrical parameterization (for the purpose of *optimizing* a shape), we were able to define formally a *full and adaptive multi-level optimum-shape algorithm* (*FAMOSA*), analogous in its multi-level logical structure to the classical *full multigrid method* (*FMG*) (known to have optimum linear complexity w.r.t. the number of degrees of freedom [1]).

In [7], we demonstrated by numerical simulation the effectiveness of certain partial steps of this construction for problems of drag reduction for transonic flight, and external noise reduction for supersonic flight. However, for non-trivial geometries discretized *a priori* by an unstructured volume mesh, another ingredient was added to the numerical method: the *free-form deformation* technique [2] has been introduced as recommended in [15]. This makes the two processes of *boundary-deformation* that pilots the optimization, and *volume-mesh deformation* that supports the flow computation, a single one. The true unknowns of the numerical algorithm are then the parameters defining the *shape deformation* rather than the *shape* itself. Much more general geometries can be handled in this versatile way since, at the initial stage of the computation, the *reference geometry* is provided in great generality by means of a three-dimensional unstructured grid, subsequently deformed by the optimization process, subject to a number of geometrical constraints.

Fig. 3, from [7], illustrates the bounding box employed to parameterize a wing deformation in a supersonic flow optimization at freestream Mach number 1.8, angle of attack 1°. Using such setting, the wing geometry of an aircraft in supersonic flow has been optimized by minimization of a functional of the form:

$$J = J_{\mathrm{SB}} + J_{C_{\mathrm{L}}} + J_{C_{\mathrm{D}}} + J_{\mathrm{Vol}} \tag{9}$$

in which

$$J_{\mathrm{SB}} = \int_{\Omega_{\mathrm{SB}}} (\nabla p)^2 \,\mathrm{d}\Omega \bigg/ \int_{\Omega_{\mathrm{SB}}} (\nabla p^0)^2 \,\mathrm{d}\Omega \tag{10}$$

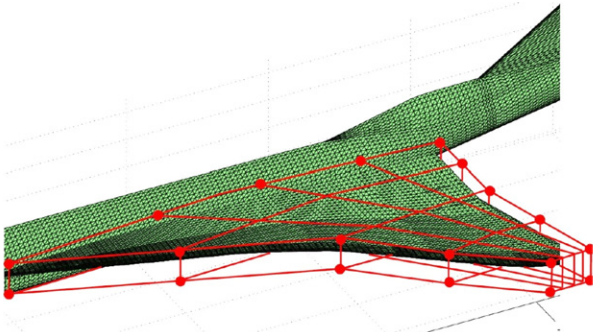

Fig. 3. Free-form deformation about a wing geometry: bounding box of a Bézier parameterization of degree 4 × 1 × 4 in the three curvilinear directions; the symbols ● indicate fixed control points; the other control points are free to move vertically.

is a measure of the pressure source of the sonic boom, and is to be reduced, whereas the other terms are penalty functions related to constraints on lift, drag and wing-box volume:

$$\begin{aligned}
J_{C_\mathrm{L}} &= \begin{cases} 10^4 \cdot |\Delta C_\mathrm{L}| & \text{if } \Delta C_\mathrm{L} < -10^{-3} \\ 0 & \text{if } \Delta C_\mathrm{L} \geqslant -10^{-3} \end{cases} \\
J_{C_\mathrm{D}} &= \begin{cases} 0 & \text{if } \Delta C_\mathrm{D} \leqslant 10^{-3} \\ 10^4 \cdot \Delta C_\mathrm{D} & \text{if } \Delta C_\mathrm{D} > 10^{-3} \end{cases} \\
J_\mathrm{Vol} &= 10^4 \cdot \max(-\Delta\mathrm{Vol}, 0)
\end{aligned} \tag{11}$$

where $\Delta g = (g - g^0)/g^0$ for $g = C_\mathrm{L}$, C_D or Vol and g^0 is the value of g for the initial design.

An unstructured mesh of 173 526 nodes and 981 822 tetrahedral elements has been used as an initial domain discretization, and iteratively deformed, without changing the mesh topology. The deformation was defined in three dimensions as a tensorial Bézier parameterization:

$$\mathbf{x}(t_1, t_2, t_3) = \mathbf{x}^0 + \sum_{k=0}^{n_1} \sum_{\ell=0}^{n_2} \sum_{m=0}^{n_3} B_{n_1}^k(t_1) B_{n_2}^\ell(t_2) B_{n_3}^m(t_3) \mathbf{p}_{k,\ell,m}, \tag{12}$$

Here, $\mathbf{x}^0$ represents a point of the original geometry, $\mathbf{x}$ its new location after deformation, and $\mathbf{p}_{k,\ell,m}$ is a generic control point.

Three embedded parameterizations were considered corresponding to $(n_1, n_2, n_3) = (4,1,2), (4,1,4)$ and $(6,1,4)$. In this preliminary test, we employed the simplex method, but not yet in the context of *FAMOSA*. Instead, we considered only a "one-way-up" algorithm, in which the vector of degrees (n_1, n_2, n_3) is successively enhanced twice, each time after completion of a partial optimization step. The convergence of the optimization iteration is indicated in Fig. 4 where the benefits of this basic hierarchical algorithm is evident.

The three-level algorithm is observed to be approximately three times faster than the original one. Although we do not have a rigorous justification of this result, it can be related to the classical result concerning the *nested iteration* in the multigrid theory, in which a progressive grid enhancement alone yields a convergence rate improvement by a factor of $\log N$ (N: number of degrees of freedom), corresponding, when the refinement is geometrical, to the number of considered levels.

Perhaps more importantly in the context of pre-industrial optimization, the improvement realized by the multi-level strategy can be viewed as an accuracy improvement for fixed amount of computational effort. The multi-level approach does seem to achieve the accuracy associated with the fine parameterization, whereas

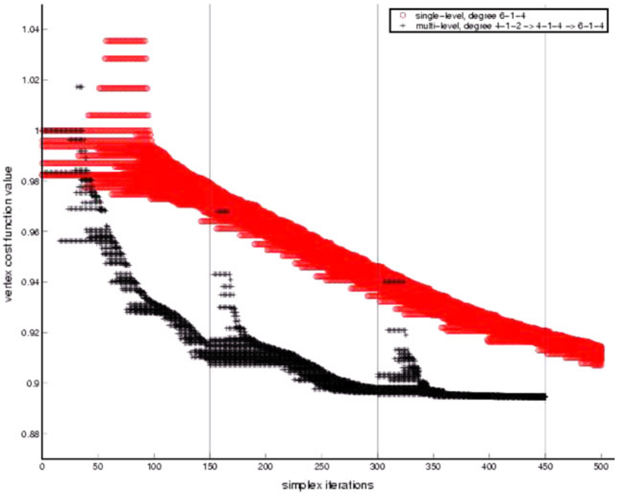

Fig. 4. Convergence history of the simplex algorithm with only finest parameterization (degree 6–1–4) vs. simplex algorithm with the degree of parameterizations gradually increased at iterations 150, 300 and 450.

with the basic algorithm, the iteration would probably be interrupted prior to full convergence, thus not fully exploiting the potential of the fine parameterization.

The purpose of this new contribution is twofold: (i) to test a *full multi-level* iteration including downward transfers as well as upward ones ("*FMOSA*"), and (ii) to propose and experiment a parameterization *adaption* (ultimately "*FAMOSA*"), and thus demonstrate the respective merits of both methodological enhancements.

2. Testing a full multi-level algorithm on a model problem

A numerical experimentation has been conducted to evaluate the gain in computational cost realized by various hierarchical algorithms over the basic optimization iteration.

The test problem is a simple model from calculus of variations in which one minimizes the shape functional

$$\mathscr{J} = \mathscr{J}(y(t)) = \frac{p^{\alpha}}{\mathscr{A}} \tag{13}$$

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

$$p = \int_0^1 \sqrt{x'(t)^2 + y'(t)^2}\,\omega(t)\,\mathrm{d}t, \quad \mathscr{A} = \int_0^1 y(t)x'(t)\omega(t)\,\mathrm{d}t \tag{14}$$

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 extensively in [8] to which we refer for a full description of the numerical test-case which corresponds to $\alpha \approx 2.03$, for which the functional is known to be convex, and a certain $\omega(t)$ for which the minimizing shape is, to rounding errors, the half-thickness distribution of the RAE2822 airfoil.

In the basic iteration, after approximation of the integrals by the trapezoidal rule with stepsize $h = \frac{1}{1000}$, and the representation of the unknown shape by a Bézier parameterization, the numerical optimum is determined by a procedure of the SCPIP package [19] (Courtesy of the University of Bayreuth), with gradient specification.

Fig. 5a indicates for different values of the degree n, the minimum value achieved for the functional $\mathscr{J}$ at full convergence, and Fig. 5(b) the corresponding suboptimal shapes (at iterations 50 and 200 in the case $n = 10$). By nature, near the optimum where $\nabla\mathscr{J} = 0$, the shapes are more sensitive to non-optimality than the functional; note that at iteration 50, the incomplete convergence degrades visibly the accuracy of the shape definition expected from the high degree $n = 10$.

Next, we examine the effect of the iterative strategy on the convergence rate. Fig. 6(a) provides the convergence history of the basic algorithm for three experiments run independently, from the same initial profile ($y = 0$), using $n = 2$, 5 or 10. Clearly, as the degree is augmented, the accuracy improves, but the stiffness increases also: more iterations are necessary to achieve convergence. Fig. 6(c) is a comparison of the basic iteration for $n = 10$, with (i) the algorithm based on two successive degree elevations ($n = 2$, then 5, then 10), each elevation being made after a partial optimization phase, and with (ii) the *full multi-level optimum-shape algorithm (FMOSA)* of [6] modeled in logical structure on the *full multigrid method*, including both downward and upward transfers of information, as indicated in Table 1.

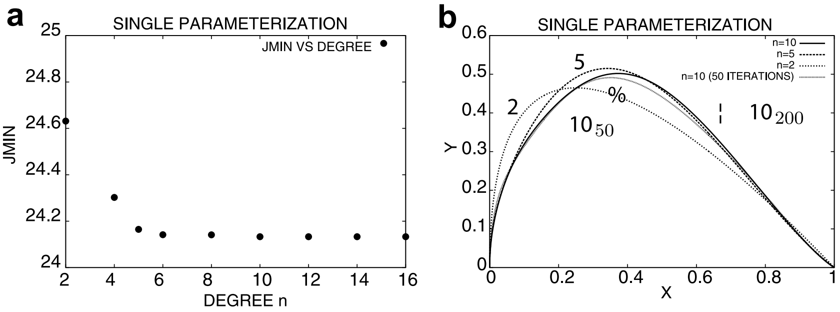

Fig. 5. Effect of degree n of Bézier parameterization on accuracy of optimum shape; (a) left: functional value $\mathscr{J}$ at convergence; (b) right: optimum shape ($n = 2$, 5 or 10).

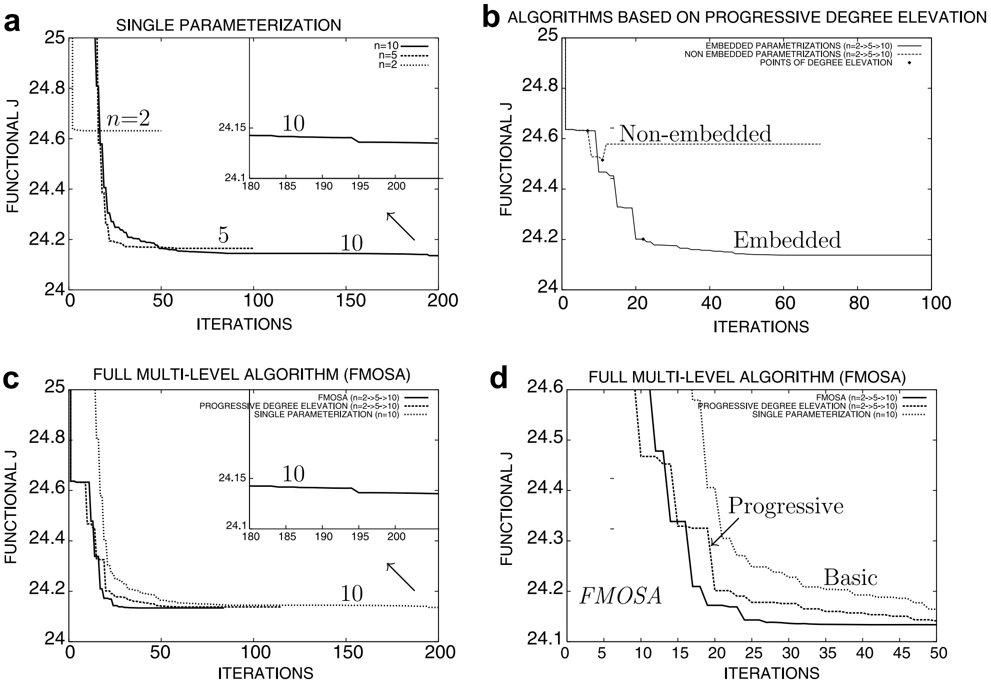

Fig. 6. Iterative performance of various algorithms; (a) top left: basic algorithm for $n = 2$, 5 or 10; (b) top right: algorithm with progressive degree elevation for proper and improper upward transfers; (c) bottom left: basic algorithm for $n = 10$, algorithm with progressive degree elevation ($n = 2$, then 5, then 10), and full multi-level algorithm (FMOSA); (d) bottom right: close-up of (c).

Table 1
Schema of the employed "saw-tooth" *FMOSA*; integers denote optimization iteration counts, enclined arrows (↗ or ↘) transfers prior to relaxation sweeps, and vertical arrows (↑) mere degree elevations; an arrow pointing downward (↘) corresponds to the reformulation of the minimization problem as the search of a shape *correction* in a nested parameterization, subsequently expressed exactly over a higher degree support by virtue of the embedding [6]; the final cycle is repeated

Fine							4						0
(degree 10)						↗		↘					↑
Medium			4		0				4				4
(degree 5)		↗		↘	↑					↘		↗	
Coarse	4				4						4		
(degree 2)													

Fig. 6(d) is a close-up of the initial stage of the same plot of iterative convergence. A close observation of Fig. 6(c) indicates that the last noticeable change in functional value occurs with the basic method near iteration 200, whereas with progressive degree elevation, it is near 60: again, with three levels considered, the algorithm based on successive degree elevations is essentially three times faster. *FMOSA* is even twice faster, realizing here an improvement in convergence rate by a factor close to 6 over the basic method.

Lastly, Fig. 6(b) is a comparison of the algorithm based on two successive degree elevations ($n = 2$, then 5, then 10) using either the advocated transfer operator (based on the classical degree-elevation process), or another transfer based on the ℓ^2-projection onto a non-embedded parameterization. Evidently, with the inexact transfer (improper in H^1-norm), both iterations at the higher degrees stabilize with shapes associated with degraded values of the criterion; in fact, the degree-10 iteration is initiated at the point of lowest value of the criterion ever achieved by the degree-5 iteration; however this transfer results in a degradation of the criterion. This experiment confirms the importance of the adequate choice to be made of the degree elevation transfer operator.

In conclusion, these experiments have confirmed the great potential of multi-level algorithms to improve significantly the convergence rate and the accuracy of the shape optimization iteration. Incomplete iterative convergence results in a poor definition of the optimum shape. Transfers by the classical process of degree elevation are strongly recommended. Qualitatively parallel between multigrid and the multi-level optimization iteration applies well.

3. Parameterization adaption by regularization

When *solving* a PDE problem numerically, a certain mesh-refinement process is always implicit, and very classically, *mesh adaptivity* instead of, or in conjunction with increasing the number of degrees of freedom, is a very effective means to accelerate grid convergence. Numerous methods exist: element division, node movement, or regeneration, in particular. Similarly, when *optimizing* a shape by means of an explicit geometrical representation, as we advocate, it is natural to seek for an analogous concept of *parameterization adaptivity*. We propose here a self-adaptive procedure for planar curves represented by Bézier parameterizations. An extension of the present technique to three-dimensional optimum-design in aerodynamics has been made [11] within the framework of the so-called *free-form deformation* approach [2] and was found effective. However here, for brevity, we restrict our discussion to a two-dimensional setting.

For *static* adaption, our algorithms attempt 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 physical criterion, and involving the numerical solution of a PDE in a domain whose boundary includes the parameterized shape; let Y^0 be the result of this phase;
(2) *Adaption:* Given the parameterization (X^0, Y^0) of an approximate optimum shape, define a better support X^1; substitute X^1 to X^0.

The above split into two separate phases, coupled via a form of Nash game, is made with the purpose of defining the adaption procedure as a simple, economical geometrical data post-processing.

Before defining our particular procedure, we discuss the basic objective that has guided its construction. In [9,10], for the purpose of a more fundamental analysis about the structure of a pertinent eigensystem, a *shape-reconstruction* or *shape-inverse problem* has been introduced, firstly by an intrinsic formulation:

$$\min_{\gamma} \mathscr{J}(\gamma) := \int_{\gamma} \frac{1}{2}[\mathsf{y}(x) - \bar{\mathsf{y}}(x)]^2 \,\mathrm{d}x \tag{15}$$

where γ is the unknown shape analytically represented by $\mathsf{y}(x)$; $\bar{\mathsf{y}}(x)$ is the analogous analytical representation of a given target curve $\bar{\gamma}$, subsequently assumed, without great loss of generality, to be a Bézier curve of degree n and support X. This problem is transformed into a *parametric optimization* by assuming Bézier representations of the curves over the support X:

$$\min_{Y\in\mathbb{R}^{n+1}} j_n(Y) := \int_0^1 \frac{1}{2}\left[B_n(t)^{\mathrm{T}}(Y - \overline{Y})\right]^2 \underbrace{nB_{n-1}(t)^{\mathrm{T}}\Delta X}_{x(t)'=\frac{\mathrm{d}}{\mathrm{d}t}B_n(t)^{\mathrm{T}}X} \mathrm{d}t \tag{16}$$

The symbol Δ represents the forward-difference operator that appears when differentiating Bernstein polynomials.

Since the functional is quadratic, the parametric gradient is linear (in Y):

$$j_n'(Y) = A(X)Y - b(X) \tag{17}$$

where

$$\begin{aligned} A(X) &= \int_0^1 \underbrace{B_n(t)B_n(t)^{\mathrm{T}}}_{\substack{(n+1)\times(n+1)\\ \text{matrix}}}\ \underbrace{nB_{n-1}(t)^{\mathrm{T}}\Delta X}_{\substack{\text{linear form}\\ \text{in } X}}\,\mathrm{d}t \\ b(X) &= \int_0^1 \underbrace{B_n(t)}_{\substack{\text{column}\\ \text{vector}}}\ \underbrace{B_n(\tau)^{\mathrm{T}}Y^0 nB_{n-1}(t)^{\mathrm{T}}\Delta X}_{\substack{\text{scalar-valued}\\ \text{nonlinear function of } X}}\,\mathrm{d}t \end{aligned} \tag{18}$$

In particular, for a uniform support X, the matrix A reduces to the simple form:

$$A = \int_0^1 B_n(t)B_n(t)^{\mathrm{T}}\mathrm{d}t = \{A_{\alpha,\beta}\} \tag{19}$$

in which the coefficients $\{A_{\alpha,\beta}\}$ are obtained by a simple calculation:

$$A_{\alpha,\beta} = \frac{1}{2n+1}\,\frac{C_n^{\alpha}C_n^{\beta}}{C_{2n}^{\alpha+\beta}} \tag{20}$$

Solving the parametric minimization problem by steepest-descent (without special preconditioning) is equivalent to applying the point-Jacobi iteration on the linear optimality condition $j_n'(Y) = 0$. Thus, the eigenstructure of the matrix A is fundamental in a *modal analysis* of the multi-level strategy. In particular, this matrix indicates how does the parameterization condition the stiffness of the optimization iteration. The diagonalization of this matrix revealed many properties discussed in [9,10]. In comparison with the usual structure of Fourier modes that classically supports the *modal analysis* of multigrid for the solution of a PDE, the matrix A exhibits, not surprisingly, the structure of an *integral* instead of a *differential* operator. Consequently, contrary to the usual situation of multigrid for PDE's, the large eigenvalues are associated with smooth modes. Hence, the minimization iteration acts as an *anti-smoother*. For this reason, here viewed in an algebraic setting, many authors have developed effective *preconditioners* to steepest-descent-type algorithms, in functional, or algebraic formulations, often resulting in the inversion of a discrete Laplacian. In a functional-space setting, it is well-known that the shape and the gradient have not the same regularity. Thus the steepest-descent algorithm, without preconditioning, is an unbounded iteration.

This observation has a very practical consequence. When following the iterative convergence, one can easily observe the control polygon becoming increasingly irregular, with large variations from point to point, sometimes with numerous sign alternations near full convergence of the optimization process (see e.g. Fig. 2). Hence, mimicking the classical regularization of the gradient by a smoothing operator, we are led to redefine, at intervals or dynamically, the geometrical representation in a way that *regularizes* the parametric representation. Among all the criteria that we tested to implement this simple concept, we observed [16] the best efficiency with a criterion that reduces the total variation of the ordinates of the control points.

Thus, the adaption separately is carried via a Stackelberg game in which the support X plays the role of *leader*, and the design vector Y of *follower*. In short: for any candidate alternate support X, a corresponding vector Y is defined so that the Bézier curve associated with the control polygon (X, Y) approximates the original one in the sense of least squares; among all possible supports X, X^1 is taken to be the one for which the total variation (TV) in the components of the corresponding vector Y^1 is minimal. The best support is thus defined to be the one that best regularizes the control polygon in the sense of least TV among all least-squares approximates. Following this principle, the main equations are now derived. We put:

$$X = E\xi + e, \quad Y = F\eta + f \tag{21}$$

in which ξ and η are reduced vectors containing the components of X and Y respectively that are free (which is problem dependent), e and f contain the other components, given by the specification of geometrical boundary conditions, E and F are rectangular matrices. For example, for an airfoil upper surface with a vertical tangent at the origin, and specified endpoints, we set:

$$x_0 = x_1 = 0, \quad x_n = 1, \quad y_0 = y_n = 0 \tag{22}$$

and thus here $\xi = (x_2, x_3, \ldots, x_{n-1})^{\mathrm{T}}$, $\eta = (y_1, y_2, \ldots, y_n)^{\mathrm{T}}$ in particular. For given X (i.e. given ξ), the vector Y (or η) is calculated to minimize the following ℓ^2-norm:

$$J = \int_0^1 \frac{1}{2}[\mathrm{y}(x) - \mathrm{y}^0(x)]^2 \mathrm{d}x \tag{23}$$

in which $\bar{y}(x)$ and $\bar{y}^0(x)$ represent the Bézier curves whose parameterizations are (X, Y) and (X^0, Y^0) respectively. Using now the parameterization indicated in (1) yields:

$$J = \int_0^1 \frac{1}{2}[y(t) - y^0(t, X)]^2 x'(t) \mathrm{d}t \tag{24}$$

in which

$$y^0(t, X) = \bar{y}^0(x(t)) = \bar{y}^0(B_n(t)^{\mathrm{T}} X) = B_n(\tau)^{\mathrm{T}} Y^0 \tag{25}$$

where the parameter value τ is related to the change of support $X^0 \to X$, and defined uniquely by the condition $x^0(\tau) = x(t)$, that is:

$$\tau = \tau(t, X) / B_n(\tau)^{\mathrm{T}} X^0 = B_n(t)^{\mathrm{T}} X \tag{26}$$

Now, since $\partial y(t)/\partial Y = B_n(t)^{\mathrm{T}}$, and $x'(t) = nB_{n-1}(t)^{\mathrm{T}} \Delta X$, differentiating J w.r.t. Y first gives:

$$\frac{\partial J}{\partial Y} = \int_0^1 \underbrace{[B_n(t)^{\mathrm{T}} Y - y^0(t, X)]}_{\text{scalar}} B_n(t)^{\mathrm{T}} \underbrace{nB_{n-1}(t)^{\mathrm{T}} \Delta X}_{\text{scalar}} \, \mathrm{d}t \tag{27}$$

or, equivalently:

$$\left(\frac{\partial J}{\partial Y}\right)^{\mathrm{T}} = \int_0^1 B_n(t)[B_n(t)^{\mathrm{T}} Y - B_n(\tau)^{\mathrm{T}} Y^0] nB_{n-1}(t)^{\mathrm{T}} \Delta X \, \mathrm{d}t = A(X)Y - b(X) \tag{28}$$

in which the matrix $A(X)$ and the vector $b(X)$ have been defined in (18).

Unsurprisingly, the matrix $A(X)$ is real-symmetric positive-definite; additionally it depends linearly upon the vector X, thus

$$A(X) = \mathscr{A} \otimes X \tag{29}$$

where $\mathscr{A} = A'(X)$ is a tensor of order 3, independent of X, and $\otimes$ stands for the contracted product implied by (18).

So, the *normal equations* implicitly defining the vector η in terms of the vector ξ, that is,

$$\frac{\partial J}{\partial \eta} = \frac{\partial J}{\partial Y} F = 0 \tag{30}$$

are written equivalently as follows:

$$\left(\frac{\partial J}{\partial \eta}\right)^{\mathrm{T}} = F^{\mathrm{T}}(A(X)Y - b(X)) = 0 \tag{31}$$

Once the above normal equations are solved for η, yielding the vector Y, the (lack of) regularity of the control polygon (X, Y) can be measured by the following criterion:

$$J_2(Y(\xi)) = TV(Y) = \sum_{k=1}^{n} |y_k - y_{k-1}| \tag{32}$$

The derivative $\partial J_2/\partial \xi$ can be calculated by successive applications of the chain rule. First let:

$$\frac{\partial J_2}{\partial Y} = p(Y)^{\mathrm{T}} \tag{33}$$

and note that this derivative is not defined when the quantity $y_k - y_{k-1}$ changes sign. This difficulty can be alleviated by substituting the following regularized criterion to J_2:

$$J_2^{\varepsilon}(Y) = \sum_{k=1}^{n} \sqrt{(y_k - y_{k-1})^2 + \varepsilon} \tag{34}$$

in which ε is a small positive number. For any index value $k = 1, 2, \ldots, n-1$, one has:

$$p_k^{\varepsilon}(Y) = \frac{\partial J_2^{\varepsilon}}{\partial y_k} = \frac{y_k - y_{k-1}}{\sqrt{(y_k - y_{k-1})^2 + \varepsilon}} - \frac{y_{k+1} - y_k}{\sqrt{(y_{k+1} - y_k)^2 + \varepsilon}} \tag{35}$$

Passing to the limit ($\varepsilon \to 0$) yields:

$$p_k(Y) = \begin{cases} +2 & \text{if } y_k > \max(y_{k-1}, y_{k+1}), \\ +1 & \text{if } y_{k-1} < y_k = y_{k+1}, \quad \text{or if } y_{k-1} = y_k > y_{k+1}, \\ 0 & \text{if } \min(y_{k-1}, y_{k+1}) < y_k < \max(y_{k-1}, y_{k+1}), \\ & \text{or if } y_{k-1} = y_k = y_{k+1}, \\ -1 & \text{if } y_{k-1} = y_k < y_{k+1}, \quad \text{or if } y_{k-1} > y_k = y_{k+1}, \\ -2 & \text{if } y_k < \min(y_{k-1}, y_{k+1}). \end{cases} \tag{36}$$

Note that the above definition applies everywhere, but the resulting derivative is not necessarily continuous; it is subject to occasional jumps (see Fig. 7). Now:

$$\frac{\partial J_2}{\partial \eta} = \frac{\partial J_2}{\partial Y} \frac{\partial Y}{\partial \eta} = p(Y)^{\mathrm{T}} F \tag{37}$$

and:

$$\frac{\partial J_2}{\partial \xi} = \frac{\partial J_2}{\partial \eta} \frac{\partial \eta}{\partial \xi} = p(Y)^{\mathrm{T}} F \frac{\partial \eta}{\partial \xi} \tag{38}$$

The matrix-valued factor

$$q = \frac{\partial \eta}{\partial \xi} \tag{39}$$

is the derivative of the design vector w.r.t. the support vector subject to the constraint of least-squares approximation of the original Bézier curve (31). It results from differentiating this constraint w.r.t. ξ. To this purpose, calculate first the differential of the constraint corresponding to an arbitrary perturbation $\mathrm{d}X = E\,\mathrm{d}\xi$:

$$F^{\mathrm{T}}[\mathrm{d}A(X)Y + A(X)\mathrm{d}Y - b'(X)\mathrm{d}X]E = 0 \tag{40}$$

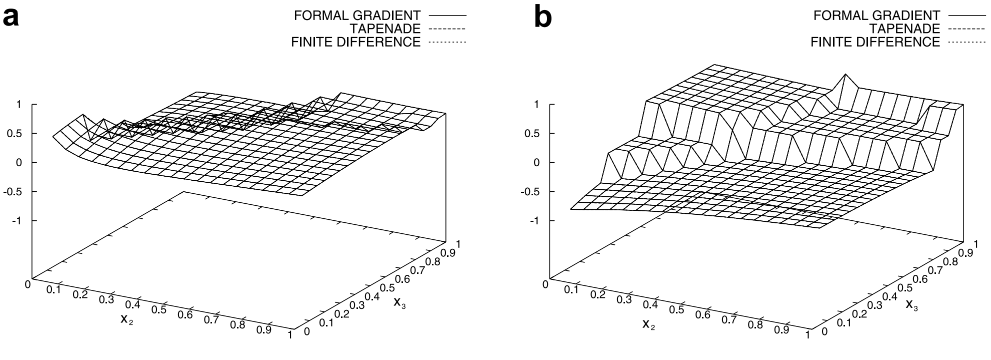

Fig. 7. Gradient of the functional J_2; here $n = 4$, $\xi = (x_2, x_3)^{\mathrm{T}}$, $(x_0 = x_1 = 0, x_4 = 1)$, and $\eta = (y_1, y_2, y_3)^{\mathrm{T}}$, $(y_0 = y_4 = 0)$ is calculated using $y = \bar{y}(x) = \sqrt{x}(1-x)/6$ as a target curve, yielding $J_2(\xi)$ and $\nabla J_2(\xi)$; three independent results (by formal calculation, program differentiation [17], and finite-difference) superimpose accurately.

in which here the superscript $'$ indicates a differentiation w.r.t. the vector X. But

$$\mathrm{d}A(X) = A'(X) \otimes \mathrm{d}X \tag{41}$$

by definition of the symbols, where $A'(X) = \mathscr{A}$ is the tensor of order 3, independent of X, introduced in (29). This gives:

$$\mathrm{d}A(X) = \mathscr{A} \otimes \mathrm{d}X = A(\mathrm{d}X) \tag{42}$$

Denote $\{e_k\}$ $(k = 0, 1, \ldots, n)$ the canonical basis of $\mathbb{R}^{n+1}$. Eq. (42) yields in particular the following expression of the partial derivative:

$$\frac{\partial A(X)}{\partial x_k} Y = A(e_k)Y \tag{43}$$

In what follows, let the symbol $\widetilde{A}_Y$ denote the matrix of dimension $(n+1) \times (n+1)$ whose kth column is equal to the vector $A(e_k)Y$. It follows that (40) is equivalent to:

$$F^{\mathrm{T}}\left[\tilde{A}_Y - b'(X) + A(X)\frac{\partial Y}{\partial X}\right]E = 0 \tag{44}$$

Additionally:

$$\frac{\partial Y}{\partial X}E = F\frac{\partial \eta}{\partial X}\frac{\partial X}{\partial \xi} = Fq \tag{45}$$

The unknown rectangular matrix q is therefore the solution of the following linear system:

$$F^{\mathrm{T}}\left[\widetilde{A}_Y - b'(X)\right]E + F^{\mathrm{T}}A(X)Fq = 0 \tag{46}$$

In summary, for a given vector ξ, the criterion J_2 and its gradient $\partial J_2/\partial\xi$ subject to the constraint of least-squares approximation of an initial Bézier parameterization, can be calculated by the following algorithm:

(1) *Regularity criterion*, J_2:
- Set $X = E\xi + e$, and compute the matrix $A(X)$ and the vector $b(X)$; perform the Choleski decomposition of the matrix $F^{\mathrm{T}}A(X)F$.
- Use this decomposition to solve the following linear system for the vector η:

$$F^{\mathrm{T}}A(X)F\eta = F^{\mathrm{T}}[b(X) - A(X)f] \tag{47}$$

set $Y = F\eta + f$, and compute $J_2 = J_2(Y)$.

(2) *Gradient*, $\partial J_2/\partial\xi$:
- Compute the vector $p(Y)$, and the matrices $\widetilde{A}_Y$ and $b'(X)$.
- Use again the same Choleski decomposition to solve the following linear system for the matrix q:

$$F^{\mathrm{T}}A(X)Fq = F^{\mathrm{T}}[b'(X) - \widetilde{A}_Y]E \tag{48}$$

and compute the gradient: $\partial J_2/\partial\xi = p^{\mathrm{T}}Fq$.

Lastly, we note that the derivative $b'(X)$ contains two terms:

$$b'(X) = b_1'(X) + b_2'(X) \tag{49}$$

where

$$b_1'(X) = \int_0^1 B_n(t)y^0(t,X)nB_{n-1}(t)^{\mathrm{T}}\Delta\,\mathrm{d}t \tag{50}$$

and since the scalar factors $y^0(t,X)$ and $nB_{n-1}(t)^{\mathrm{T}}\Delta X$ commute:

$$b_2'(X) = \int_0^1 B_n(t) y^0(t,X)' n B_{n-1}(t)^{\mathrm{T}} \Delta X \,\mathrm{d}t \tag{51}$$

in which again the superscript $'$ indicates a differentiation w.r.t. the vector X. The calculation of the term $b_1'(X)$ from the function $y^0(t,X) = B_n(\tau)^{\mathrm{T}} Y^0$ is straightforward. For the second term, $b_2'(X)$, an additional derivative of this nonlinear function is needed. It can be shown that:

$$y^0(t,X)' = \frac{B_{n-1}(\tau)^{\mathrm{T}} \Delta Y^0}{B_{n-1}(\tau)^{\mathrm{T}} \Delta X^0} B_n(t)^{\mathrm{T}} \tag{52}$$

in which again $\tau = \tau(t,X)$ is the solution of Eq. (26). To illustrate this derivation, Fig. 7 depicts the gradient in a particular case, exhibiting several fronts of discontinuity.

Equipped with this algorithm for calculating the measure of the lack of regularity of an initial Bézier parameterization (X^0, Y^0) by the criterion J_2 and its gradient $\partial J_2/\partial\xi$, the regularity can be improved by minimization of J_2 by means of a standard gradient-based procedure. For this, we have used the *FFSQP* procedure [18] (Courtesy of the University of Maryland) with gradient specification.

Fig. 8(a) demonstrates the regularizing effect on the control polygon of an initial Bézier curve by means of our adaption procedure. Here, the profile $y = \sqrt{x}(1-x)/6$, which is very much alike the RAE2822-airfoil half-thickness distribution, was defined as a target. For subsequent purpose, note that by letting $x(t) = t^2$, one gets $y(t) = t(1-t^2)/6$; therefore, this profile is the Bézier curve of degree 3 associated with the control points:

$$P_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad P_1 = \begin{pmatrix} 0 \\ \frac{1}{18} \end{pmatrix}, \quad P_2 = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{9} \end{pmatrix}, \quad P_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{53}$$

The actually considered initial parameterization is the least-squares approximate of the target associated with the degree-12 support obtained by 8 degree elevations from $\left(0, 0, \frac{1}{3}, \frac{2}{3}, 1\right)$. The corresponding polygon is the "jagged" line of Fig. 8(a).

After application of the adaption procedure, the new profile is essentially superimposed to the former, and cannot visually be distinguished from it, whereas the associated parameterization has been regularized. The new control polygon is convex, in this case however different from the result of 9 degree elevations from (53). Fig. 8(b) shows the iterative convergence of the criterion J_2.

Lastly, and most importantly, we measure the gain realized by the self-adaptive parameterization on the accuracy of the optimization. For the model problem (13) in the particular case where $\omega(t) \equiv 1$ and $\alpha = 2$, the functional is convex and the optimum shape is the circular arc $\left(x - \frac{1}{2}\right)^2 + y^2 = \frac{1}{4}$. For $n = 3$, 6 and 12, first independently, we have again used the initial condition $y = \sqrt{x}(1-x)/6$ in Bézier form, and conducted an alternating sequence of optimization and adaption. The results form a decreasing sequence of functional values represented alternatively by the symbols ● and ○ in Fig. 9(a).

As expected, the approximate minimum achieved by the first optimization from the initial shape, decreases with a larger degree n, since the search space is enlarged. The minimum achieved at convergence of the optimization-adaption sequence, is very close to the theoretical value of 2π even when n is small; however, it is not

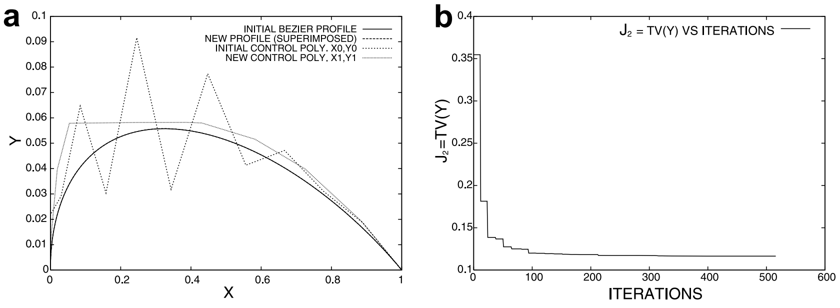

Fig. 8. Regularizing effect of Bézier parameterization adaption; (a) left: initial profile (least-squares approximate of $y = \sqrt{x}(1-x)/6$ over the support resulting from 8 degree elevations from $(0, 0, \frac{1}{3}, \frac{2}{3}, 1)$), new profile (essentially superimposed), and the two corresponding control polygons; (b) right: iterative convergence of the criterion J_2 by the procedure *FFSQP* with gradient specification.

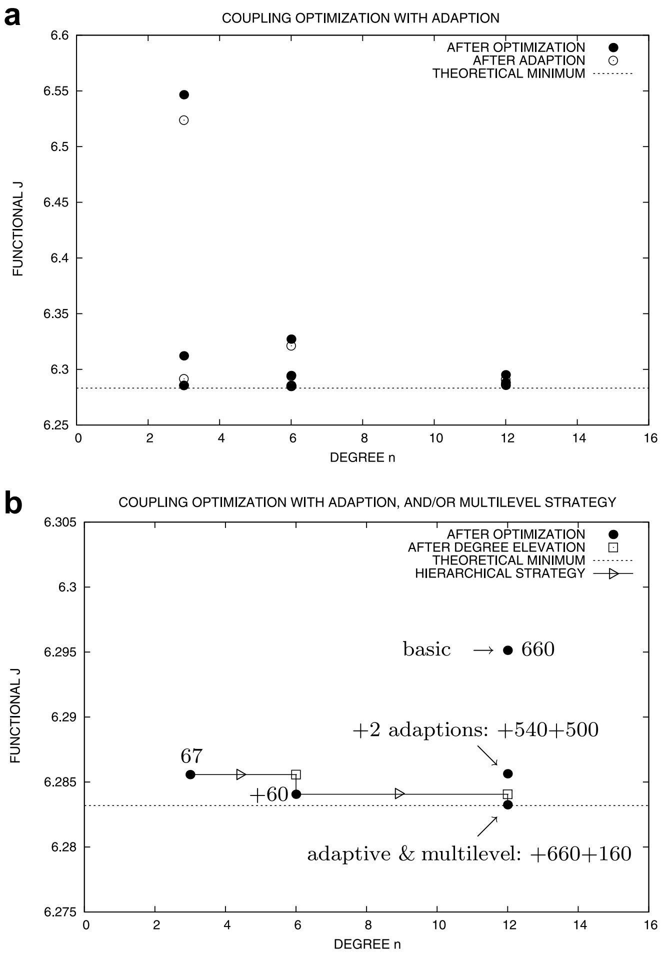

Fig. 9. Coupling optimization with adaption (a), and with a hierarchical strategy (b); the figures close to symbols on (b) are optimization iteration counts, providing an indication on the numerical stiffness.

monotonic with n. This could be due either to unequally converged stiff systems, or to the bias introduced by the Stackelberg coupling strategy. Nevertheless, the adaption, acting as a preconditioner on the optimization phase, performs remarkably well when the degree is low. Using the best point achieved with $n = 3$, as an initial

Table 2
Proposed *adaptive* saw-tooth variant: *FAMOSA*; the parameterization-support adaptions (→) are performed at fixed degree prior to degree-elevation+relaxation ↗; again ↘ indicates the formulation of a *correction* problem over the embedded support inherited from the last adaption, and ↑ a mere degree-elevation; N_1, N_2: iteration counts

Fine											N_1		0
(degree 10)										↗		↘	↑
Medium					N_1		0	⟶	0				N_2
(degree 5)				↗		↘	↑						
Coarse	N_1	⟶	0				N_2						
(degree 2)													

condition in Fig. 9(b), a sequence of degree-elevation followed by an adaptive optimization was realized twice. This basic hierarchical strategy has permitted to achieve the lowest functional value of all, realizing for $n = 12$ a minimization visibly more effective and more economical than the basic method using a parameterization of the same degree.

4. Conclusions

Multi-level strategies are better equipped to alleviate the numerical stiffness in numerical shape optimization when a parameterization of high degree is used. Additionally, parameterization self-adaptivity improves noticeably the accuracy permitting lower degrees to be used with equivalent performance. Both methodological elements combine very well, and numerous promising algorithmic variants are to be explored, such as the *FAMOSA* defined in Table 2.

In the perspective of pre-industrial optimum-shape design, the greatest benefit drawn from a reduced numerical stiffness may lie in the potential to identify the optimum shape more accurately, the diminished computational cost resulting mostly in engineering comfort.

References

[1] P. Wesseling, An Introduction to Multigrid Methods, John Wiley & Sons, 1992 (Corrected Reprint: Philadelphia, R.T. Edwards, Inc., 2004).

[2] T.W. Sederberg, Computer Aided geometric Design. <http://tom.cs.byu.edu/~tom/> (Item: Courses).

[3] G. Farin, W. Rheinboldt, D. Siewiorek (Eds.), Curves and Surfaces for Computer-Aided Geometric Design – A Practical Guide, Academic Press, Boston, 1990.

[4] J. Périaux, G. Bugeda, P.K. Chaviaropoulos, K. Giannakoglou, S. Lantéri, B. Mantel (Eds.), Optimum Aerodynamic Design & Parallel Navier–Stokes Computations, ECARP European Computational Aerodynamics Research Project, Notes on Numerical Fluid Mechanics, Vieweg, Braunschweig/Wiesbaden, Germany, 1998.

[5] F. Courty, A. Dervieux, B. Koobus, L. Hascoet, Reverse automatic differentiation for optimum design: from adjoint state assembly to gradient computation, Journal of Optimization Methods and Software 18 (5) (2003) 615–627.

[6] J.-A. Désidéri, Hierarchical optimum-shape algorithms using embedded Bézier parameterizations, in: Y. Kuznetsov et al. (Eds.), Numerical Methods for Scientific Computing, Variational Problems and Applications, CIMNE, Barcelona, 2003.

[7] J.-A. Désidéri, A. Janka, Multilevel shape parameterization for aerodynamic optimization – application to drag and noise reduction of transonic/supersonic business jet, in: E. Heikkola et al., (Ed.), European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004), Jyväskyla, 24–28 July 2004.

[8] J.-A. Désidéri, J.-P. Zolésio, Inverse shape optimization problems and application to airfoils, Control and Cybernetics 34 (1) (2005).

[9] J.-A. Désidéri, Two-level ideal algorithm for parametric shape optimization, in: W. Fitzgibbon, R. Hoppe, J. Périaux, O. Pironneau, Y. Vassilevski (Eds.), Advances in Numerical Mathematics, Proceedings of International Conferences, Moscow, Institute of Numerical Mathematics, Russian Academy of Sciences, September 16–17, 2005 and Houston, Department of Mathematics, University of Houston, 2006, pp. 65–85, in press.

[10] J.-A. Désidéri and A. Dervieux, Hierarchical Methods for Shape Optimization in Aerodynamics – I: multilevel algorithms for parametric shape optimization, in: J. Périaux and H. Deconinck, (Eds.), Introduction to Optimization and Multidisciplinary Design, Lecture Series 2006-3, Von Karman Institute for Fluid Dynamics Publish., 2006.

[11] R. Duvigneau, Adaptive Parameterization using Free-Form Deformation for Aerodynamic Shape Optimization, INRIA Research Report No. 5949, 2006. <http://hal.inria.fr/inria-00085058>.

[12] M.-B. Giles, Adjoint methods for aeronautical design, in: Proc. ECCOMAS CFD Conference, Swansea, UK, 2001.

[13] L. Hascoet, M. Vazquez, A. Dervieux, Automatic differentiation for optimum design, applied to sonic boom reduction, in: V. Kumar, et al., (Ed.), Proceedings of the International Conference on Computational Science and its Applications, ICCSA'03, Montreal, Canada, LNCS 2668, Springer Publish., 2003, pp. 85–94.

[14] A. Jameson, Aerodynamic design via control theory, Journal of Scientific Computing 3 (3) (1988) 233–260.

[15] J.A. Samareh, Multidisciplinary Aerodynamic-Structural Shape Optimization Using Deformation (MASSOUD), AIAA-2000-4911. (8th AIAA/NASA/USAF/ISS MO Symposium on Multidisciplinary Analysis and Optimization, September 6–8, 2000/Long Beach, CA.).

[16] Z.L. Tang, J.-A. Désidéri, Towards Self-Adaptive Parameterization of Bézier Curves for Airfoil Aerodynamic Design, INRIA Research Report No. 4572, 2002. <http://hal.inria.fr/inria-72016>.

[17] TAPENADE Automatic Differentiation Tool. <http://www-sop.inria.fr/tropics/tapenade/>.

[18] J.L. Zhou, A.L. Tits, C.T. Lawrence, User's Guide for FFSQP Version 3.7: A Fortran Code for Solving Optimization Programs, Possibly Minimax, with General Inequality Constraints and Linear Equality Constraints, Generating Feasible Iterates, Institute for Systems Research, University of Maryland, Technical Report SRC-TR-92-107r5, College Park, MD 20742, 1997. <http://www.aemdesign.com/>.

[19] C. Zillober, Software manual for SCPIP 2.3, Technical Report Nr. TR02-1, Informatik, Universität Bayreuth, 2002. <http://www.uni-bayreuth.de/departments/math/~czillober/abstracts/tr02-1>.