

Nested and Self-Adaptive Bézier Parameterizations for Shape Optimization

Jean-Antoine DÉSIDÉRI, Badr ABOU EL MAJD *

INRIA, 2004 Route des Lucioles, BP 93, F-06902 Sophia Antipolis Cedex, France
E-mail: Jean-Antoine.Desideri@sophia.inria.fr Badr.AbouElMajd@sophia.inria.fr

and Aleš JANKA **

*Bât. de Mathématiques, Station 8, Ecole Polytechnique Fédérale de Lausanne,
CH-1015 Lausanne, Suisse*
E-mail: ales.janka@epfl.ch

This article is a sequel of [6], in which we defined formally a hierarchical shape optimization method based on a multi-level shape representation by nested Bézier parameterizations (*FAMOS*A), and [7] 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.

Keywords: Partial-Differential Equations, Computational Methods, Numerical Shape Optimization, Compressible Aerodynamics

AMS Subject classification: Primary 65C20, 76N25; Secondary 65C17

1. Introduction

We focus on certain methodological questions in numerical shape optimization when a Partial-Differential Equation (P.D.E.) 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.

* Project Team Opale: <http://www-sop.inria.fr/opale/>

** Institut d'Analyse et Calcul Scientifique: <http://iacs.epfl.ch/asn/>

In aerodynamics, evaluating the cost functional gradient raises non-trivial theoretical questions [9] 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 [11], or by means of automatic differentiation as in [5] [10]. Hessians are still seldomly 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:

$$x(t) = \sum_{k=0}^n B_n^k(t) x_k, \quad y(t) = \sum_{k=0}^n B_n^k(t) y_k \quad (1.1)$$

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

$$B_n^k(t) = C_n^k t^k (1-t)^{n-k} \quad (1.2)$$

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

$$P_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix} \quad (k = 0, 1, \dots, n) \quad (1.3)$$

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

$$X = \{x_k\}, \quad Y = \{y_k\}, \quad k = 0, 1, \dots, n, \quad (1.4)$$

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 vec-

tor 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:

$$x(t) = B_n(t)^T X, \quad y(t) = B_n(t)^T Y, \quad (1.5)$$

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:

$$x'(t) = n \sum_{k=0}^{n-1} B_{n-1}^k(t) (x_{k+1} - x_k) = n B_{n-1}(t)^T \Delta X \quad (1.6)$$

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 (1.1) into the following equivalent Bézier parameterization of degree $n+1$:

$$x(t) = \sum_{k=0}^{n+1} B_{n+1}^k(t) x'_k, \quad y(t) = \sum_{k=0}^{n+1} B_{n+1}^k(t) y'_k \quad (1.7)$$

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

$$P'_0 = P_0, \quad P'_k = \frac{k}{n+1} P_{k-1} + \left(1 - \frac{k}{n+1}\right) P_k \quad (k = 1, 2, \dots, n), \quad P'_{n+1} = P_n \quad (1.8)$$

obtained by multiplying (1.1) by $(1-t) + t$ 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). This is illustrated on Figure 1 in which the supports of three nested parameterizations of an airfoil are sketched.

Note that in the case of Figure 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

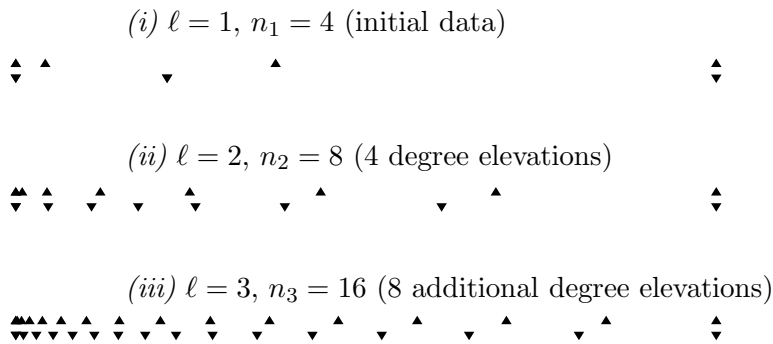


Figure 1. One-dimensional example of embedded parameterizations: the triangles represent the supports X of three nested Bézier parameterizations of degree 4, 8 and 16 of an RAE2822 airfoil obtained from the first by 4 and 12 successive degree elevations; the symbols pointing upward (resp. downward) are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

degree elevation process. The parameterizations are nested, or embedded in one another in this sense precisely.

Figure 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 P.D.E.) 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 (FAMOS)*, 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 nontrivial geometries discretized *a priori* by an unstructured volumic mesh, another ingredient was added to the numerical method: the *Free-Form Deformation* technique [2], has been introduced as recommended in [12]. This makes the two processes of *boundary-deformation* that pilots the optimization, and *volumic-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

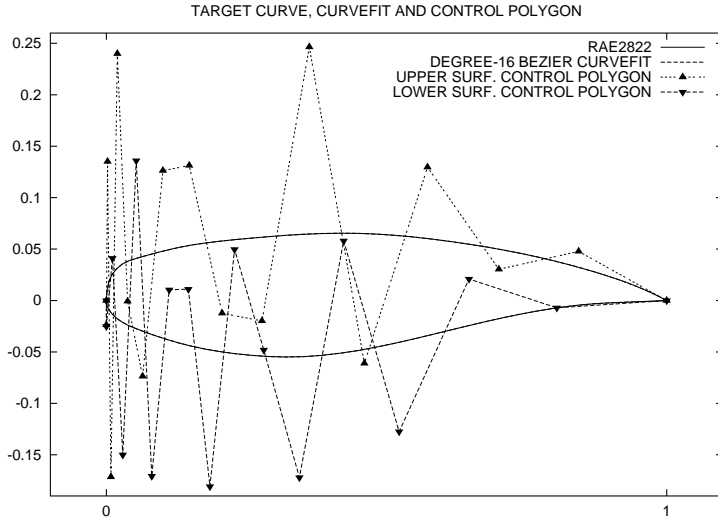


Figure 2. RAE2822 airfoil, examples of degree-16 Bézier curvefits of the upper and lower surfaces (superimposed), and corresponding control polygons.

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.

Figure 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 geom-

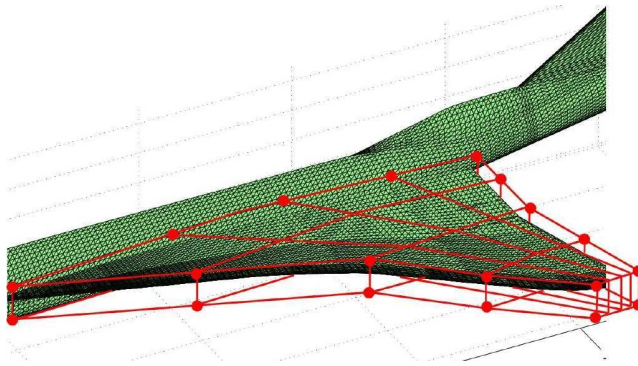


Figure 3. Free-form deformation about a wing geometry: bounding box of a Bézier parameterization of degree $4 \times 1 \times 4$ in the three curvilinear directions; the symbols \bullet indicate fixed control points; the other control points are free to move vertically.

etry of an aircraft in supersonic flow has been optimized by minimization of a functional of the form:

$$J = J_{SB} + J_{C_L} + J_{C_D} + J_{Vol} \quad (1.9)$$

in which:

$$J_{SB} = \int_{\Omega_{SB}} (\nabla p)^2 d\Omega \bigg/ \int_{\Omega_{SB}} (\nabla p^0)^2 d\Omega \quad (1.10)$$

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 volume. An unstructured mesh of 173526 nodes and 981822 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:

$$\mathbf{x}(t_1, t_2, t_3) = \mathbf{x}^0 + \sum_{k=0}^{n_1} \sum_{\ell=0}^{n_2} \sum_{m=0}^{n_3} B_{n_1}^k(t_1) B_{n_2}^\ell(t_2) B_{n_3}^m(t_3) \mathbf{p}_{k,\ell,m}, \quad (1.11)$$

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

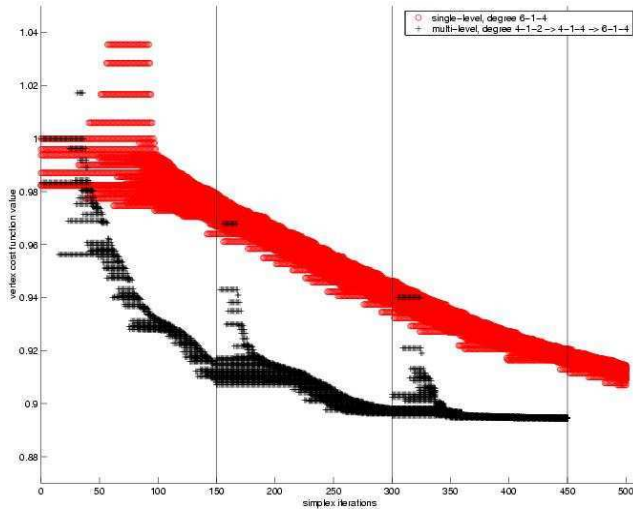


Figure 4. Convergence history of the simplex algorithm with only finest parameterization (degree 6-1-4) vs. simplex algorithm with the degree of parameterizations gradually increased at iterations 150, 300 and 450.

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

$$\mathcal{J} = \mathcal{J}(y(t)) = \frac{p^\alpha}{\mathcal{A}} \tag{2.1}$$

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

$$p = \int_0^1 \sqrt{x'(t)^2 + y'(t)^2} \omega(t) dt, \quad \mathcal{A} = \int_0^1 y(t) x'(t) \omega(t) dt \tag{2.2}$$

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 SCIP package [15] (Courtesy of the University of Bayreuth), with gradient specification.

Figure 5a indicates for different values of the degree n , the minimum value achieved for the functional \mathcal{J} at full convergence, and Figure 5b the corresponding suboptimal shapes (at iterations 50 and 200 in the case $n = 10$). By nature, near the optimum where $\nabla \mathcal{J} = 0$, 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 $n = 10$.

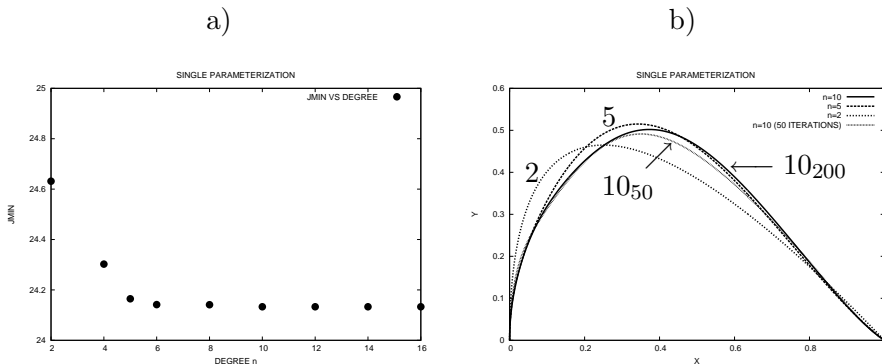


Figure 5. Effect of degree n of Bézier parameterization on accuracy of optimum shape; a) left: functional value \mathcal{J} at convergence; b) right: optimum shape ($n = 2, 5$ or 10).

Next, we examine the effect of the iterative strategy on the convergence rate. Figure 6a 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. Figure 6c 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.

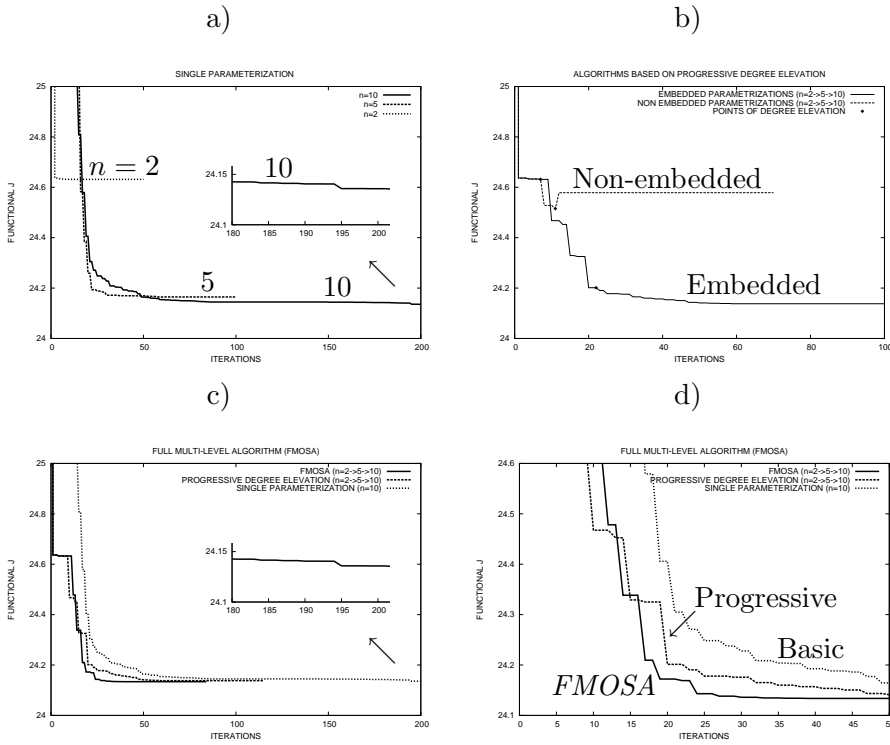


Figure 6. Iterative performance of various algorithms; a) top left: basic algorithm for $n = 2, 5$ or 10 ; b) top right: algorithm with progressive degree elevation for proper and improper upward transfers; c) bottom left: basic algorithm for $n = 10$, algorithm with progressive degree elevation ($n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

Figure 6d is a close-up of the initial stage of the same plot of iterative convergence. A close observation of Figure 6c 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, Figure 6b 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

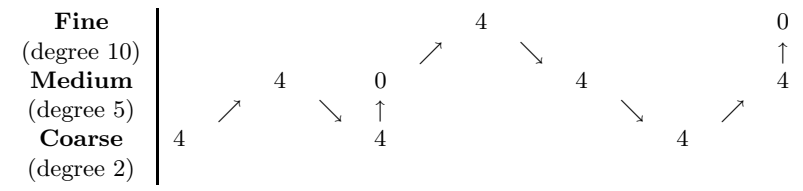


Table 1

Skematic of the employed “saw-tooth” *FMOSA*; integers denote optimization iteration counts, enclined arrows (\nearrow or \searrow) transfers prior to relaxation sweeps, and vertical arrows (\uparrow) mere degree elevations; an arrow pointing downward (\searrow) 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.

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. Quantitatively, the parallel between multigrid and the multi-level optimization iteration applies well.

3. Parameterization adaption

When *solving* a P.D.E. 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. Extensions to a higher dimension and, or more general parameterizations are possible in numerous ways, but not considered here for brevity.

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 so-

lution of a P.D.E. 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.

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 approximants. Following this principle, the main equations are now derived. We put:

$$X = E \xi + e, \quad Y = F \eta + f \tag{3.1}$$

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:

$$x_0 = x_1 = 0, \quad x_n = 1, \quad y_0 = y_n = 0, \tag{3.2}$$

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

$$J = \int_0^1 \frac{1}{2} [\bar{y}(x) - \bar{y}^0(x)]^2 dx \tag{3.3}$$

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 (1.1) yields:

$$J = \int_0^1 \frac{1}{2} [y(t) - y^0(t, X)]^2 x'(t) dt \tag{3.4}$$

in which:

$$y^0(t, X) = \bar{y}^0(x(t)) = \bar{y}^0(B_n(t)^T X) = B_n(\tau)^T Y^0 \tag{3.5}$$

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

$$\boxed{\tau = \tau(t, X) / \quad B_n(\tau)^T X^0 = B_n(t)^T X} \quad (3.6)$$

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

$$\frac{\partial J}{\partial Y} = \int_0^1 \underbrace{[B_n(t)^T Y - y^0(t, X)]}_{\text{scalar}} B_n(t)^T \underbrace{nB_{n-1}(t)^T \Delta X}_{\text{scalar}} dt \quad (3.7)$$

or, equivalently:

$$\begin{aligned} \left(\frac{\partial J}{\partial Y}\right)^T &= \int_0^1 B_n(t) [B_n(t)^T Y - B_n(\tau)^T Y^0] nB_{n-1}(t)^T \Delta X dt \\ &= A(X)Y - b(X) \end{aligned} \quad (3.8)$$

in which the matrix $A(X)$ and the vector $b(X)$ are defined as follows:

$$\boxed{\begin{aligned} A(X) &= \int_0^1 \underbrace{B_n(t) B_n(t)^T}_{\substack{(n+1) \times (n+1) \\ \text{matrix}}} \underbrace{nB_{n-1}(t)^T \Delta X}_{\substack{\text{linear form} \\ \text{in } X}} dt \\ b(X) &= \int_0^1 \underbrace{B_n(t)}_{\substack{\text{column} \\ \text{vect.}}} \underbrace{B_n(\tau)^T Y^0 nB_{n-1}(t)^T \Delta X}_{\substack{\text{scalar-valued} \\ \text{nonlinear funct. of } X}} dt \end{aligned}} \quad (3.9)$$

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

$$A(X) = \mathcal{A} \otimes X \quad (3.10)$$

where $\mathcal{A} = A'(X)$ is a tensor of order 3, independent of X , and \otimes stands for the contracted product implied by (3.9).

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

$$\frac{\partial J}{\partial \eta} = \frac{\partial J}{\partial Y} F = 0 \quad (3.11)$$

are written equivalently as follows:

$$\left(\frac{\partial J}{\partial \eta}\right)^T = F^T (A(X)Y - b(X)) = 0 \quad (3.12)$$

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:

$$J_2(Y(\xi)) = TV(Y) = \sum_{k=1}^n |y_k - y_{k-1}| \quad (3.13)$$

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

$$\frac{\partial J_2}{\partial Y} = p(Y)^T \quad (3.14)$$

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 :

$$J_2^\varepsilon(Y) = \sum_{k=1}^n \sqrt{(y_k - y_{k-1})^2 + \varepsilon} \quad (3.15)$$

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

$$p_k^\varepsilon(Y) = \frac{\partial J_2^\varepsilon}{\partial y_k} = \frac{y_k - y_{k-1}}{\sqrt{(y_k - y_{k-1})^2 + \varepsilon}} - \frac{y_{k+1} - y_k}{\sqrt{(y_{k+1} - y_k)^2 + \varepsilon}} \quad (3.16)$$

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

$$p_k(Y) = \begin{cases} +2 & \text{if } y_k > \max(y_{k-1}, y_{k+1}), \\ +1 & \text{if } y_{k-1} < y_k = y_{k+1}, \text{ or if } y_{k-1} = y_k > y_{k+1}, \\ 0 & \text{if } \min(y_{k-1}, y_{k+1}) < y_k < \max(y_{k-1}, y_{k+1}), \\ & \text{or if } y_{k-1} = y_k = y_{k+1}, \\ -1 & \text{if } y_{k-1} = y_k < y_{k+1}, \text{ or if } y_{k-1} > y_k = y_{k+1}, \\ -2 & \text{if } y_k < \min(y_{k-1}, y_{k+1}). \end{cases} \quad (3.17)$$

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

$$\frac{\partial J_2}{\partial \eta} = \frac{\partial J_2}{\partial Y} \frac{\partial Y}{\partial \eta} = p(Y)^T F \quad (3.18)$$

and:

$$\frac{\partial J_2}{\partial \xi} = \frac{\partial J_2}{\partial \eta} \frac{\partial \eta}{\partial \xi} = p(Y)^T F \frac{\partial \eta}{\partial \xi} \quad (3.19)$$

The matrix-valued factor

$$\boxed{q = \frac{\partial \eta}{\partial \xi}} \quad (3.20)$$

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 (3.12). 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 $dX = E d\xi$:

$$F^T [dA(X)Y + A(X)dY - b'(X)dX] E = 0 \quad (3.21)$$

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

$$dA(X) = A'(X) \otimes dX \quad (3.22)$$

by definition of the symbols, where $A'(X) = \mathcal{A}$ is the tensor of order 3, independent of X , introduced in (3.10). This gives:

$$dA(X) = \mathcal{A} \otimes dX = A(dX) \quad (3.23)$$

Denote $\{e_k\}$ ($k = 0, 1, \dots, n$) the canonical basis of \mathbb{R}^{n+1} . Equation (3.23) yields in particular the following expression of the partial derivative:

$$\frac{\partial A(X)}{\partial x_k} Y = A(e_k) Y \quad (3.24)$$

In what follows, let the symbol \tilde{A}_Y denote the matrix of dimension $(n+1) \times (n+1)$ whose k th column is equal to the vector $A(e_k)Y$. It follows that (3.21) is equivalent to:

$$F^T \left[\tilde{A}_Y - b'(X) + A(X) \frac{\partial Y}{\partial X} \right] E = 0 \quad (3.25)$$

Additionally:

$$\frac{\partial Y}{\partial X} E = F \frac{\partial \eta}{\partial X} \frac{\partial X}{\partial \xi} = F q \quad (3.26)$$

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

$$F^T \left[\tilde{A}_Y - b'(X) \right] E + F^T A(X) F q = 0 \quad (3.27)$$

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 $F^T A(X) F$.
- Use this decomposition to solve the following linear system for the vector η :

$$F^T A(X) F \eta = F^T [b(X) - A(X) f], \quad (3.28)$$

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 \tilde{A}_Y and $b'(X)$.
- Use again the same Choleski decomposition to solve the following linear system for the matrix q :

$$F^T A(X) F q = F^T [b'(X) - \tilde{A}_Y] E, \quad (3.29)$$

and compute the gradient: $\partial J_2 / \partial \xi = p^T F q$.

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

$$b'(X) = b'_1(X) + b'_2(X) \quad (3.30)$$

where

$$b'_1(X) = \int_0^1 B_n(t) y^0(t, X) n B_{n-1}(t)^T \Delta dt \quad (3.31)$$

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

$$b'_2(X) = \int_0^1 B_n(t) y^0(t, X)' n B_{n-1}(t)^T \Delta X dt \quad (3.32)$$

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 $y^0(t, X) =$

$B_n(\tau)^T Y^0$ is straightforward. For the second term, $b'_2(X)$, an additional derivative of this nonlinear function is needed. It can be shown that:

$$y^0(t, X)' = \frac{B_{n-1}(\tau)^T \Delta Y^0}{B_{n-1}(\tau)^T \Delta X^0} B_n(t)^T \tag{3.33}$$

in which again $\tau = \tau(t, X)$ is the solution of equation (3.6). To illustrate this derivation, Figure 7 depicts the gradient in a particular case, exhibiting several fronts of discontinuity.

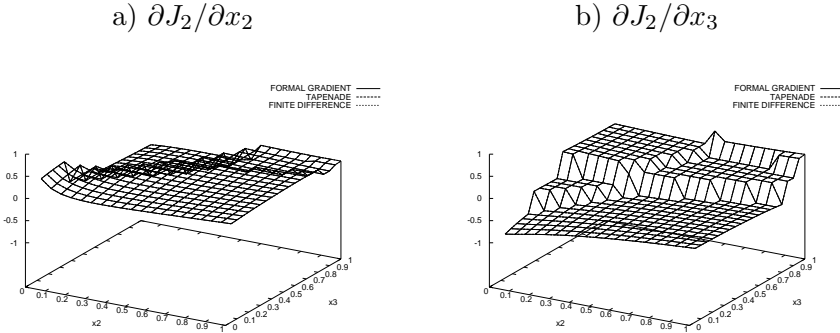


Figure 7. Gradient of the functional J_2 ; here $n = 4$, $\xi = (x_2, x_3)^T$ ($x_0 = x_1 = 0$, $x_4 = 1$), and $\eta = (y_1, y_2, y_3)^T$ ($y_0 = y_4 = 0$) is calculated using $y = \bar{y}(x) = \sqrt{x(1-x)}/6$ as a target curve, yielding $J_2(\xi)$ and $\nabla J_2(\xi)$; three independent results (by formal calculation, program differentiation [13], and finite-difference) superimpose accurately.

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 [14] (Courtesy of the University of Maryland) with gradient specification.

Figure 8a 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:

$$P_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad P_1 = \begin{pmatrix} 0 \\ \frac{1}{18} \end{pmatrix}, \quad P_2 = \begin{pmatrix} \frac{1}{3} \\ \frac{1}{9} \end{pmatrix}, \quad P_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{3.34}$$

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

the “jagged” line of Figure 8a. After application of the adaption procedure, the new profile is essentially superimposed to the former, whereas the associated parameterization has been regularized. The new control polygon is convex, however different from the result of 9 degree elevations from (3.34). Figure 8b shows the iterative convergence of the criterion J_2 .

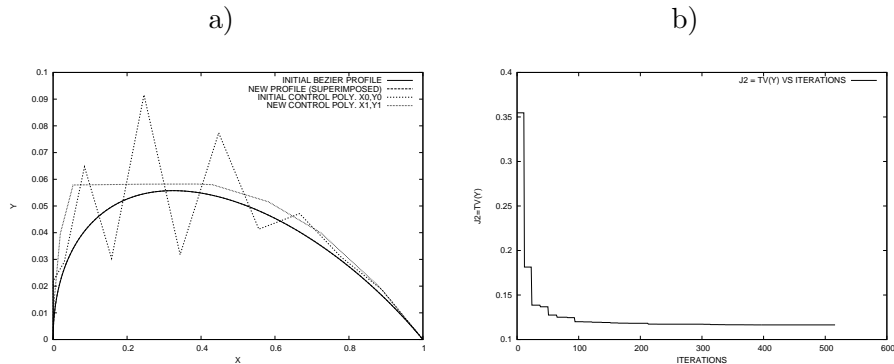
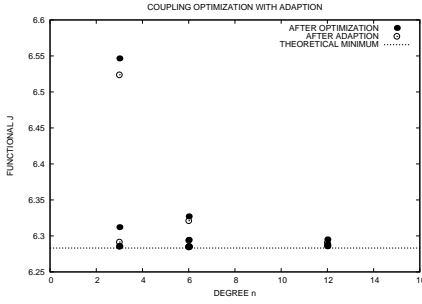


Figure 8. Regularizing effect of Bézier parameterization adaption; a) left: initial profile (least-squares approximant of $y = \sqrt{x}(1-x)/6$ over the support resulting from 8 degree elevations from $(0, 0, \frac{1}{3}, \frac{2}{3}, 1)$), new profile (essentially superimposed), and the two corresponding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

Lastly, and most importantly, we measure the gain realized by a self-adaptive parameterization on the accuracy of an optimization. For this we have considered the model problem (2.1) in the particular case where $\omega(t) \equiv 1$, and $\alpha = 2$ for which the functional is convex and the optimum shape is the circular arc $(x - \frac{1}{2})^2 + y^2 = \frac{1}{4}$. 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, whose results form a decreasing sequence of functional values represented alternatively by the symbols \bullet and \circ on Figure 9a.

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 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 condition in Figure 9b, a sequence of degree-elevation followed by an adaptive optimization was realized twice. This basic hierarchical

a) Basic method and adaption



b) Adaption and hierarchy

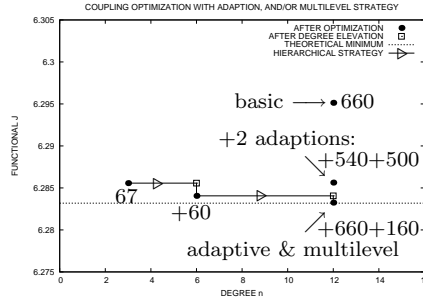


Figure 9. Coupling optimization with adaption, and with a hierarchical strategy; on the right, figures are optimization iteration counts, indicating how stiffness increases with n .

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.

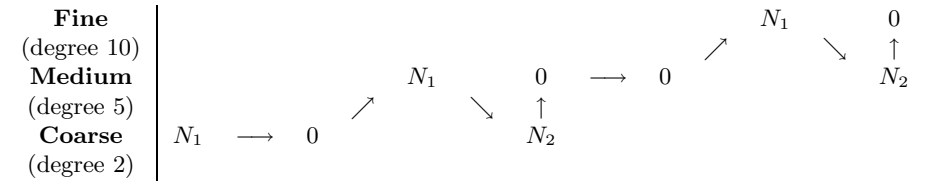


Table 2

Proposed *adaptive* saw-tooth variant: *FAMOSA*; the parameterization-support adaptations (\rightarrow) are performed at fixed degree prior to degree-elevation+relaxation \nearrow ; again \searrow indicates the formulation of a *correction* problem over the embedded support inherited from the last adaption, and \uparrow a mere degree-elevation; N_1, N_2 : iteration counts.

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] *An Introduction to Multigrid Methods*, P. Wesseling, John Wiley & Sons Ltd, 1992. Corrected Reprint: Philadelphia, R.T. Edwards, Inc., 2004.
- [2] *Computer Aided geometric Design*, T.W. Sederberg, <http://tom.cs.byu.edu/~tom/> (Item: Courses).
- [3] *Curves and Surfaces for Computer-Aided Geometric Design – A Practical Guide*, G. Farin. W. Rheinboldt and D. Siewiorek eds., Academic Press, Boston, 1990.
- [4] *Optimum Aerodynamic Design & Parallel Navier-Stokes Computations, ECARP European Computational Aerodynamics Research Project*, J. Périaux and G. Bugeđa and P. K. Chaviaropoulos and K. Giannakoglou and S. Lantéri and B. Mantel eds., Notes on Numerical Fluid Mechanics, Vieweg, Braunschweig/Wiesbaden, Germany, 1998.
- [5] F. Courty, A. Dervieux, B. Koobus, and L. Hascoet, L., *Reverse automatic differentiation for optimum design: from adjoint state assembly to gradient computation*, J. Optimization Methods and Software, Taylor & Francis Publish., 18(5), pp. 615-627, 2003.
- [6] J.-A. Désidéri, *Hierarchical Optimum-Shape Algorithms Using Embedded Bézier Parameterizations*, Numerical Methods for Scientific Computing, Variational Problems and Applications, Y. Kuznetsov et al eds., 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*, European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2004), E. Heikkola et al eds., Jyväskylä, 24-28 July 2004.
- [8] J.-A. Désidéri, J.-P. Zolésio, *Inverse Shape Optimization Problems and Application to Airfoils*, Control and Cybernetics, to appear.
- [9] M.-B. Giles, *Adjoint methods for aeronautical design*, Proc. ECCOMAS CFD Conference, Swansea, U.K., 2001.
- [10] L. Hascoet, M. Vazquez, and A. Dervieux, *Automatic Differentiation for Optimum Design, Applied to Sonic Boom Reduction*, Proc. International Conference on Computational Science and its Applications, ICCSA'03, Montreal, Canada, V. Kumar et al eds., LNCS 2668, Springer Publish., pp. 85-94, 2003.
- [11] A. Jameson, *Aerodynamic Design via Control Theory*, Journal of Scientific Computing 3(3), pp. 233-260, 1988.
- [12] 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.)
- [13] *TAPENADE Automatic Differentiation Tool*, <http://www-sop.inria.fr/tropics/tapenade/>
- [14] J. L. Zhou, A. L. Tits and 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/>
- [15] 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>