HIERARCHICAL METHODS FOR SHAPE OPTIMIZATION IN AERODYNAMICS
I: Multilevel algorithms for parametric shape optimization

Jean-Antoine Désidéri[*] and Alain Dervieux[†]
INRIA
2004 Route des Lucioles, BP 93, F-06902 Sophia Antipolis cedex
(France)

March 2006

Contents

[*] Opale Project Team
[†] Tropics Project Team

1 Prologue: why should we use multilevel algorithms in analysis?

Before considering the context of shape optimization, let us return to the more classical discussion about which algorithms should be used to solve the stiff system of algebraic equations that result from discretizing a partial-differential-equation (PDE) boundary-value problem. One such problem arises when solving the steady Euler (or Navier-Stokes) equations in compressible Aerodynamics.

1.1 Model problem, modal analysis, direct method versus iterations

To introduce the basic concepts, and in particular the concept of *complexity* for one such algorithm, let us consider the prototypical problem of Laplace's equation subject to homogeneous Dirichlet boundary conditions over the simple domain $\Omega = [0,1]^d$, an interval, a square or a cube of $\mathbb{R}^d$ ($d = 1$, 2 or 3):

$$\begin{cases} -\Delta u = f & (\Omega) \\ u = 0 & (\partial\Omega) \end{cases} \tag{1}$$

Symbolically, this boundary-value problem writes:

$$Au = f \tag{2}$$

Assume the interval [0,1] is discretized uniformly with $N+1$ points subscripted from 0 to N in each coordinate direction. If standard central differencing is used, a discrete analog of (1)-(2) is:

$$\boxed{A_h u_h = f_h} \tag{3}$$

where

$$h = \frac{1}{N} \tag{4}$$

is the characteristic dimension of the discrete elements, u_h is the vector of *nodal unknowns.* The two-dimensional case ($d = 2$) is really examplary, and is examined in particular; the $(N-1)^2$ unknowns can be arranged in many rational ways, by rows, columns, diagonals, etc. Assuming an arrangement by columns:

$$u_h = \{\, u_{i,j} \,\} = \{\, u_{1,1}, u_{1,2}, ..., u_{1,N-1}, u_{2,1}, u_{2,2}, ..., u_{2,N-1}, ..., u_{N-1,1}, u_{N-1,2}, ..., u_{N-1,N-1} \,\} \tag{5}$$

Otherwise, in general ($d = 1$, 2 or 3), the number of unknowns, or degrees of freedom is:

$$\mathsf{N} = (N-1)^d \sim N^d \quad (N \gg 1) \tag{6}$$

In (3), f_h a vector containing the discrete nodal values of the function f, and A_h is an *approximation matrix.* For Laplace's equation, (1), after central differencing, the matrix

A_h is the direct sum of d tridiagonal matrices of dimension $(N-1)\times(N-1)$. For example, in two dimensions ($d=2$):

$$A_h = A_x \oplus A_y = A_x \oplus I_{N-1} + I_{N-1} \oplus A_y \tag{7}$$

where the $(N-1)\times(N-1)$ matrices

$$A_x = A_y = \frac{1}{h^2}\,\mathrm{Trid}(-1,2,-1) \tag{8}$$

are tridiagonal. The matrix A_h has a classical block-tridiagonal structure, since:

$$A_x \oplus I_{N-1} = \frac{1}{h^2}\begin{pmatrix} 2I_{N-1} & -I_{N-1} & & & \\ -I_{N-1} & 2I_{N-1} & -I_{N-1} & & \\ & -I_{N-1} & 2I_{N-1} & \ddots & \\ & & \ddots & \ddots & -I_{N-1} \\ & & & -I_{N-1} & 2I_{N-1} \end{pmatrix} \tag{9}$$

whereas:

$$I_{N-1} \oplus A_y = \frac{1}{h^2}\begin{pmatrix} 2A_y & -A_y & & & \\ -A_y & 2A_y & -A_y & & \\ & -A_y & 2A_y & \ddots & \\ & & \ddots & \ddots & -A_y \\ & & & -A_y & 2A_y \end{pmatrix} \tag{10}$$

Consequently, the eigenvectors of the matrix A_h are discrete Fourier modes with the following separated-variable (or tensorial) structure [Désidéri (1998)]:

$$s_{i,j}^{(m,\ell)} = 2h\,\sin i\alpha^{(m)}\,\sin j\beta^{(\ell)} \tag{11}$$

where:

$$\alpha^{(m)} = \frac{m\pi}{N} = m\pi h\,, \quad \beta^{(\ell)} = \frac{\ell\pi}{N} = \ell\pi h \tag{12}$$

are *frequency parameters*, and the associated eigenvalue is a sum of contributions from each frequency mode:

$$\lambda^{(m,\ell)} = \frac{2-2\cos\alpha^{(m)}}{h^2} + \frac{2-2\cos\beta^{(\ell)}}{h^2} = \frac{4}{h^2}\left(\sin^2\frac{\alpha^{(m)}}{2} + \sin^2\frac{\beta^{(\ell)}}{2}\right) \tag{13}$$

In 3D (or more simply, in 1D), there would appear one more (resp. one less) $\sqrt{2h}$ sin factor in (11), and one more (resp. one less) $\frac{4}{h^2}\sin^2$ term in (13).

Note that the dimension of the matrix A_h is $\mathsf{N}\times\mathsf{N} = (N-1)^d \times (N-1)^d$. The bandwidth is the maximum difference in ranks in the ordering of the unknowns. With an arrangement similar to the above, this number is equal to twice the number of neighbors of a given node in a given hyperplane, and that is proportional to $(N-1)^{d-1} \sim \mathsf{N}/N$ (total number of unknowns/number of unknowns per coordinate direction; $N \gg 1$).

Therefore, if a direct method (Gaussian elimination) was used to solve the discrete system (3), assuming an economical algorithm memory-wise, both the computational cost and the memory requirement would be proportional to the number of elements in the smallest banded portion of the matrix containing all the nonzero elements, which is the portion that is filled in by the forward-elimination and backward-substitution. The size of this banded portion is of the order of the bandwidth times the total number of rows; therefore it is proportional to $(\mathsf{N}/N) \times \mathsf{N} \sim \mathsf{N}^2/N$. This estimate indicates than for systems originating from PDEs, even linear, when a large mesh is necessary for accuracy, the memory requirement of the direct method is completely prohibitive. With a general unstructured grid the estimate becomes N^2 (since no bound is known on the bandwidth), and the estimate is even more dramatic. Lastly, implementing a full matrix inversion is somehow contradictory with the effort to construct a local differencing scheme.

These observations direct most authors preference towards*iterative methods* instead for the solution of systems resulting from the discretization of PDEs in a general setting. Additionally, in a real-life problem, if a direct method is used, the nonlinear character of the PDE would restrict this usage to a partial solution within a global iterative formulation based on some form of linearization.

Therefore, we should qualify iterative methods in terms of computaional cost efficiency with respect to (w.r.t.) the mesh refinement which is quantified here by the parameter h. This is this notion that we refer to as the *complexity* of the algorithm. We first make this notion precise for the most basic iterative method, point-Jacobi iteration, and we then generalize to more complex algorithms.

1.2 The Jacobi iteration

The Jacobi method is the simplest iterative method for a linear system. Most other methods, such as the commonly-used Gauss-Seidel iteration, are often viewed as variants with special preconditioners. Therefore, the Jacobi method plays a major prototypical role.

For a positive-definite (neutrally-) diagonally-dominant matrix system such as (3) for Laplace's equation, point-Jacobi iteration, in our notations, and in the classical sense of Varga [Varga (2000)] consists in first rewriting each equation by solving it w.r.t. its diagonal element,

$$D_h\, u_h = (D_h - A_h)u_h + f_h \tag{14}$$

where $D_h = \text{Diagonal}(A_h)$ is made of the diagonal elements of A_h only, and defining second the following iteration:

$$u_h^{n+1} = D_h^{-1}(D_h - A_h)u_h^n + D_h^{-1} f_h = u_h^n - D_h^{-1}\left(A_h\, u_h^n - f_h\right) \tag{15}$$

For Laplace's equation discretized by central differencing, D_h is a scalar matrix with the number $(2d)/h^2$ in the diagonal (except for a few equations related to nodes near boundaries). Hence, the above equation has approximately the following form:

$$\boxed{u_h^{n+1} = u_h^n - \tau h^2\left(A_h\, u_h^n - f_h\right)} \tag{16}$$

where τ is some positive number (equal to $2d$). Subsequently, although it is slightly different from the usual convention, we will use (16) as definition for Jacobi iteration, and we will allow the positive parameter τ to be adjustable. The quantity $(A_h\, u_h^n - f_h)$ is the *discrete residual*, and thus (16) generalizes straightforwardly to the discretization of any boundary-value PDE problem, even if the corresponding *approximation operator* A_h is not linear or diagonally-dominant, without presuming actual iterative convergence.

To analyze the iterative convergence of the Jacobi iteration, one substracts the trivial equation

$$u_h = u_h - \tau h^2\,(A_h\, u_h - f_h) \tag{17}$$

to (16) to get:

$$e_h^{n+1} = e_h^n - \tau h^2 A_h\, e_h^n \tag{18}$$

in which:

$$\boxed{e_h^n := u_h^n - u_h} \tag{19}$$

is the *iterative-error nodal vector*. Hence, the convergence is described by the linear equation:

$$e_h^{n+1} = (I - \tau h^2 A_h) e_h^n \tag{20}$$

For the model problem, according to (11)-(12-(13) the matrix A_h can be diagonalized by an orthogonal transformation, corresponding to a discrete Fourier transform:

$$A_h = \frac{1}{h^2}\, S_h\, \Lambda_h\, S_h^{-1} \tag{21}$$

where $S_h^{-1} = S_h^T = S_h$. Thus defining the *iterative-error modal vector*, that is, the iterative-error vector expressed in discrete Fourier modes:

$$\boxed{\varepsilon_h^n := S_h^{-1} e_h^n} \tag{22}$$

(20) diagonalizes:

$$\varepsilon_h^{n+1} = (I - \tau\Lambda_h)\, \varepsilon_h^n \tag{23}$$

and gives:

$$\boxed{\varepsilon_h^n = (I - \tau\Lambda_h)^n\, \varepsilon_h^0} \tag{24}$$

that is, component-wise, or Fourier-mode-wise:

$$\left(\varepsilon_h^n\right)^{(m)} = \left(1 - \tau\bar{\lambda}_h^{(m)}\right)^n \left(\varepsilon_h^0\right)^{(m)} \tag{25}$$

where m refers to the mode (really (m,ℓ) in 2D, and (m,ℓ,μ) in 3D) and $\bar{\lambda}_h^{(m)} = h^2 \lambda_h^{(m)}$ is scaled eigenvalue.

Thus, every modal component of the iterative-error vector decays geometrically, being multiplied at each iteration by a factor

$$\gamma_h^{(m)} = 1 - \tau\bar{\lambda}_h^{(m)} \tag{26}$$

depending on the eigenvalue and the relaxation parameter τ. This allows us to define the *specral radius* of the iteration as fllows:

$$\boxed{\rho_h(\tau) = \max_m \left|\gamma_h^{(m)}\right|} \tag{27}$$

Evidently, the spectral radius varies piecewise linearly with τ. Putting:

$$\bar{\lambda}_{\min} = \bar{\lambda}_h^{(1)}, \quad \bar{\lambda}_{\max} = \bar{\lambda}_h^{(\mathsf{N})} \tag{28}$$

one has $\rho_h(\tau) = 1 - \tau\bar{\lambda}_{\min}$ for small enough τ and $\rho_h(\tau) = \tau\bar{\lambda}_{\max} - 1$ afterwards. Hence the optimum value for the parameter τ is:

$$\boxed{\tau^* = \left(\frac{\bar{\lambda}_{\min} + \bar{\lambda}_{\max}}{2}\right)^{-1}} \tag{29}$$

and this gives:

$$\boxed{\rho_h^* = \rho_h(\tau^*) = \frac{\bar{\lambda}_{\max} - \bar{\lambda}_{\min}}{\bar{\lambda}_{\max} + \bar{\lambda}_{\min}} = \frac{\kappa - 1}{\kappa + 1}} \tag{30}$$

where κ is the *condition number*:

$$\boxed{\kappa = \frac{\bar{\lambda}_{\max}}{\bar{\lambda}_{\min}}} \tag{31}$$

Obviously, for the model problem,

$$\kappa = O\left(\frac{1}{h^2}\right) = O\left(N^2\right) \gg 1 \tag{32}$$

where again, N is basically the number of degrees of ferredom per coordinate direction. Consequently,

$$\rho_h^* = 1 - \frac{2}{\kappa} + \cdots = 1 - O\left(h^2\right) \tag{33}$$

In conclusion, the iterative-error after n iterations of the Jacobi method is reduced by a factor of the order of $(\rho_h^*)^n$, where ρ_h^* can be related to the spectrum of eigenvalues as indicated above.

Now, the approximation error is controled by grid size h according to an estimate of the type:

$$\|u_h - u\| \sim C_A \times h^\alpha \tag{34}$$

where α is the order of approximation: $\alpha = 2$ for central differencing.

Therefore, a logical *termination criterion* for the Jacobi method is to stop whenever n is large enough to make the reduction factor of the iterative error, $(\rho_h^*)^n$, of the order of the approximation error, h^α. Using natural logarithms, this gives:

$$n \, \ln \rho_h^* \sim \ln h \tag{35}$$

(where constants have been ignored). Now

$$-\ln \rho_h^* = O\left(\frac{1}{\kappa}\right) = O\left(h^2\right) \tag{36}$$

and this results in:

$$n \sim \frac{-\ln h}{h^2} = N^2 \, \ln N \tag{37}$$

Additionally, applying the Jacobi method, (16), results in performing at each iteration a bounded number of arithmetic operations for each nodal equation. Thus, the computational cost involved in one iteration is simply proportional to the total number of unknowns $\mathsf{N} = N^d$ in the model problem.

In summary, accounting for:

- the estimate of the total number of iterations necessary to reduce the iterative error to the order of the approximation error, $N^2 \, \ln N$, and for

- the fact that the computational cost of one iteration is proportional to the total number of unknowns $\mathsf{N} = N^d$,

we conclude that solving the model problem with this accuracy by application of the Jacobi method, involves a computational cost whose order of magnitude is estimated as follows in terms of the number N of degrees of freedom per coordinate direction:

$$\boxed{\mathrm{COST}_{\mathrm{JACOBI}} \sim N^{d+2} \, \ln N = \mathsf{N} \times N^2 \, \ln N} \tag{38}$$

Equation (38) provides an estimate of the order of magnitude of the *global computational cost* relative to the Jacobi iteration when the iterative termination criterion is made consistent with the order of magnitude of the *approximation error.* We use the term *complexity* of the method for such an estimate.

We will now sketch out the principal multilevel strategies, and outline the classical arguments that support their corresponding reduced complexity estimates. The reader is directed to textbooks such as [Wesseling (1992)], [Briggs (1991)] or [Désidéri (1998)] for a more detailed analysis.

1.3 Nested iteration

The so-called *Nested Iteration* was originally introduced for elliptic problems in [Kronsjö and Dahlquist (1972)]. It relies on a simple principle: if the mesh refinement is the cause of numerical stiffnees, greater efficiency should be achieved by *progressive mesh enrichment.*

The method is sketched in Table 1 in a case where 3 grid levels are used.

Fine:					$\mathcal{M}_{h/4}$	...
				$\nearrow$		
Medium:			$\mathcal{M}_{h/2}$			
		$\nearrow$				
Coarse:	$\mathcal{M}_h$					
	Solve	Prolongate	**Smooth**	Prolongate	**Smooth**	...

Table 1: Sketch of *nested iteration* for 3 grid levels.

One first constructs a sequence of rigorously, or approximately nested grids $\mathcal{M}_{h_i}$ $(i = 0, 1, ..., K)$, and for purpose of analysis, assume that the number of gridpoints in each coordinate direction is approximately increased geometrically by a factor r each time, starting from the coarsest grid $\mathcal{M}_{h_0}$ $(i = 0)$.

At each level, the PDE is approximated consistently by the system

$$A_{h_i} u_{h_i} = f_{h_i} \qquad (i = 0, 1, ..., K) \tag{39}$$

For $i = 0$, the method is initiated by the complete solution of the above problem on the coarsest level, $h = h_0$ by a standard relaxation method, such as the Jacobi method. The iteration is interrupted as soon as the iterative error has reached the order of magnitude of the approximation on this grid, that is, h_0^α for some α (order of accuracy of the discretization scheme, say 2 for the model problem. The computational cost of this initial phase is very small if the number K of levels is large due to the relatively very coarse first grid.

For all subsequent grids, $i = 1, 2, ..., K$:

1. Prolongate the solution $u_{h_{i-1}}$ obtained on grid $\mathcal{M}_{h_{i-1}}$ to $\mathcal{M}_{h_i}$ by an appropriate interpolation operator $I_{h_{i-1}}^{h_i}$ to initialize the solution on the new grid:

$$u_{h_i}^0 = I_{h_{i-1}}^{h_i} u_{h_{i-1}} \tag{40}$$

2. Iterate on the solution $u_{h_i}^n$ $(n = 1, 2, ...)$ from the above guess $(n = 0)$ and until the termination criterion corresponding to grid-level i is satisfied.

Because on each grid, the termination criterion is adjusted to the corresponding approximation error, the error in $u_{h_{i-1}}$ is of the order of h_{i-1}^α. If the interpolation is accurate enough, this approximation error, at least in order of magnitude, is simply inherited by $u_{h_i}^0$. Therefore in the second step, the relaxation only need to be done to reduce the iterative error a factor equal to the fraction of approximation errors:

$$\frac{h_i^\alpha}{h_{i-1}^\alpha} = \frac{1}{r^\alpha} = C \tag{41}$$

where C is a known constant. Assuming again that the spectral radius ρ_{h_i} is grid-dependent according to (33), the corresponding number of necessary iterations is estimated by:

$$n_i \sim \frac{\ln C}{\ln \rho_{h_i}^*} \sim N_i^2 \tag{42}$$

where the ln factor has been removed. Therefore the cost of the computation made on grid-level i is therefore proportional to $\mathsf{N}_i \times N_i^2$, and the total cost:

$$\mathrm{COST}_{\mathrm{NESTED}} \sim \mathsf{N} \times N^2 \left(1 + \frac{1}{R} + \frac{1}{R^2} + ... + \frac{1}{R^{K-1}}\right) \leq \mathsf{N} \times N^2 \frac{1}{1 - \frac{1}{R}} \tag{43}$$

where the subscript $i = K$ corresponding to the upmost level is now omitted, and $R = r^{d+2}$ for the model problem, that is a constant >1.

Simplifying this result, we get the following uniform estimate:

$$\boxed{\mathrm{COST}_{\mathrm{NESTED}} \sim \mathsf{N} \times N^2} \tag{44}$$

This result corresponds to a partial success: indeed, the complexity of the basic iteration given by (38) has been reduced, but only by the logarithmic factor $\ln N$, equal to the number of levels if geometrical refinement is used as above. Additionally, the method to be effective requires "accurate-enough" interpolations. It turns out that linear interpolation is notably insufficient for Laplace's equation (see e.g. [Désidéri (1998), pp. 291-303).

Two final remarks before examining even more elaborate iterative strategies:

- In a nonlinear context, it is known by experience that the *nested iteration* has the additional merit of enhanced robustness.
- The *nested iteration* can be related to what is known as the *Cascadic Multigrid* [Bornemann and Deuflhard (1996)] with slightly different methodological adjustments.

1.4 Multigrid cycle

The key breakthrough realized by the multigrid cycle relies on the observation that if we split the *modal or frequential* components of the iterative-error vector into two subvectors of respectively high and low frequencies, then independently of the meshsize, the condition number related to the high-frequency modes alone is bounded.

For example for the model problem in 1D ($d = 1$, and N even):

$$\frac{\lambda_N}{\lambda_{N/2}} = \frac{4 \sin^2 \frac{(N-1)\pi}{2N}}{4 \sin^2 \frac{(N/2)\pi}{2N}} \approx 2 \tag{45}$$

The *modal analysis* [Brandt (1977)] implies that a finite number of classical relaxation sweeps suffice to reduce to a negligible level say half of the error components, those associated with the high frequencies. Because of their lower frequency spectrum, the persistent error components can be accurately represented on a coarser mesh, in a *correction problem.* Then again, a finite number of relaxation sweeps will suffice to reduce to negligible level "half" of the remaining portion of the spectrum, and so on recursively, until the coarsest level is reached. On the coarsest level, the number of degrees of freedom is so small that the related correction problem can be solved at very low cost. Then the solution is estimated by a succession of prolongations, possibly followed by additional smoothing steps (*V-cycle*).

We will admit that all the frequency-components of the iterative-error vector are subject to an effective mesh-independent reduction by this *hierarchical process of elimination.* In other words, the iterative-error vector is reduced by an effective multigrid-cycle by a factor B independent of the meshsize:

$$\boxed{\rho^* \leq B} \tag{46}$$

1.5 Full Multigrid Method, “FMG”

This method is a variant of the *Nested Iteration* obtained by replacing the classical relaxation iteration by an appropriate *Multigrid cycle.* This construction is sketched out on Table 2 for the case of 3 grids.

Fine:					$\mathcal{M}_{h/4}$	
				↗	↓↑	
Medium:			$\mathcal{M}_{h/2}$		$\mathcal{M}_{h/2}$	
		↗	↓↑		↓↑	
Coarse:	$\mathcal{M}_h$		$\mathcal{M}_h$		$\mathcal{M}_h$	
	Solve (1-grid)	Prolongate	**MG-cycle** (2-grid)	Prolongate	**MG-cycle** (3-grid)	…

Table 2: Sketch of the Full-Multi-Grid (FMG) method.

Now, let us revise the cost estimate of the nested iteration in this hypothesis. Here, (42) becomes:

$$n_i = \text{a constant} \tag{47}$$

which states that the same fixed number of multigrid cycles should be performed at every stage of the mesh refinement (evey column of Table 2). Thus the computational cost of a given stage is only proportional to the number of degrees of freedom at that stage, that is, N_i. Hence the total cost of the computation is proportional to

$$\sum_i \mathsf{N}_i = \mathsf{N}\left(1 + \frac{1}{R} + \frac{1}{R^2} + ... + \frac{1}{R^{K-1}}\right) \leq \frac{\mathsf{N}}{1 - \frac{1}{R}} \tag{48}$$

Hence, we arrive at the following remarkable result:

$$\boxed{\text{COST}_{\text{FMG}} \sim \mathsf{N}} \tag{49}$$

Effective “Full-Multi-Grid” methods have a *linear convergence rate*: the computational cost necessary to solve a problem iteratively, with a tolerance of the order of the approximation error, is only proportional to the total number of degrees of freedom.

This general result makes multigrid methods very cost-efficient for boundary-value problems for which the numerical options can be adjusted according to the theoretical criteria, onlu outlined in this short review.

2 Introduction: why should we use multilevel algorithms for shape optimization?

Our general framework is the development, mathematical analysis and experimentation of numerical methods for shape optimization for applications in which the cost functional evaluation 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 such as structural mechanics (e.g. elasticity), or electromagnetics (e.g. the Maxwell equations). These PDEs are very commonly solved by Finite Elements or Volumes, by techniques that, although becoming increasingly standard, are still very costly when the accuracy requirement is high.

The important development of multigrid methods in recent years has demonstrated that such techniques not only permit to accelerate the iterative convergence of solution procedures, but also have the more general merit of a better control on grid dependency and convergence. In fact, the linear convergence rate demonstrated in the previous section can only be achieved by applying adequate iterative termination criteria, that is, when the grid-convergence control is enforced properly.

With these concepts in mind, in a long-time perspective, we would like to generalize these convergence control criteria within a functional optmization loop, such as shape optimization in aerodynamics.

Thus, our efforts are mostly concentrated on improving the convergence rate of numerical procedures both from the viewpoint of cost-efficiency and accuracy, with the perspective of reducing the design cost, but also of mastering the election and control of the design parameters, geometrical ones in particular, in a more rational way, perhaps supported by error estimates.

Technically, our efforts tend to contribute to the following challenges:

- Construct multi-level (multi-scale) shape-optimization algorithms;
- Identify critical algorithmic ingredients (transfer operators, smoothers);
- Evaluate efficiency, theorize convergence via error estimates or an appropriate *modal analysis.*

At INRIA, a first attempt to develop such multilevel extensions to optimum-shape design was made by A. Dervieux and collaborators. They proposed a technique of *Hierarchical Smoothing* [Dervieux et al. (2001)] in which the multilevel geometrical data structure of agglomeration multigrid is exploited to define a hierarchical optimization algorithm. The major concepts related to this approach are presented in Part II of these notes.

First, in Part I, we discuss the construction of self-adaptive multilevel algorithms, in the context of *parametric shape optimization.* Embedded search spaces are defined based on a geometrical hierarchy of nested shape parameterizations of Bézier type. We provide some details on how such multilevel geometrical representations can be used to define multilevel algorithms for shape optimization, and how parameterization adaption can be devised. We present some typical results related to a model problem in calculus of variations introduced in depth in [Désidéri and Zolésio (2005)], and we refer to [Désidéri and Janka (2004)], [Désidéri et al. (2004)] and [Abou El Majd et al. (2005)] for examples of applications to aerodynamics. In particular, in these publications, the so-called *Free-Form Deformation* approach is used to extend our basic multilevel construction of parametric spaces to encompass 3D deformations in a bounding box, making our approach far more general. Second, for purpose of analysis, we present a simple conceptual model problem for shape optimization, and illustrate the corresponding eingenmodes. This model allows us to discuss a central issue in multilevel algorithms: *smoothing*.

3 An example of multilevel & self-adaptive shape-optimization algorithm

In this section, we exploit the geometrical nature of the optimization problem to devise efficient multilevel algorithms that mimick the logical structure of the geometrical multigrid method, known to have ideally, for boundary-value problems, meshsize-independent convergence rate (see e.g. [Wesseling (1992)]). To some extent, this development is not limited to a special optimizer; however, we have in mind utilizing a steepest-descent-type optimizer, on all levels except perhaps the coarsest, because evolutionary algorithms raise particular questions related to population initialization at each transfer from one level to another.

A prototype problem is the wing-shape aerodynamic optimization for a simplified aircraft geometry. Typically the Euler equations are solved by Finite Volumes over a three-dimensional mesh, which is a discretization of both the volumic domain, and also, in part, the surface subject to optimization. A first remark is that these boundary points alone are in large number in size cases. Hence, when using the coordinates of these points as design parameters, some explicit or implicit reduction of dimension is necessary to make the algorithm tractable. Additionally, the discrete geometrical scale necessary to a satisfactory representation of the smooth wing geometry, such as the size of a patch in a spline representation, may be far larger than the local geometrical scale h (local mesh size) necessary to the flow simulation, particularly in a shocked flow. Secondly, if the optimizer has the structure of a gradient-based method, some form of preconditioner or regularization is necessary, because the gradient lies in a functional space of much weaker regularity than the shape itself. We address both these requirements by the usage of a smooth parameterization involving a moderate number of parameters, and producing automatically smooth shapes. In the most complex cases, we employ tensorial Bézier parameterizations of the *shape deformation* from an initial geometry given by a three-dimensional unstructured grid (*Free-form deformation*).

Thirdly, we would like to incorporate a multilevel strategy, in the treatment of geometrical parameters, mimicking the multigrid procedure, to achieve optimal convergence rate. A solution to this requirement has been proposed by A. Dervieux *et al* [Dervieux et al. (2001)] in what they referred to as "Hierarchical Smoothing" to exploit the multilevel geometrical data structure of *agglomeration MG* that is applied to the distributed solution of the Euler equations. Our approach is somewhat similar in concept, but the realization is made via a *hierarchy of nested smooth shape-deformation parameterizations* of tensorial Bézier type in a bounding box. We provide hereafter only the most relevant features of our technique; more details can be found in [Désidéri (2003)], [Désidéri and Janka (2004)], [Désidéri and Zolésio (2005)], [Désidéri et al. (2004)].

3.1 Shape representation, Bézier parameterization

As an example, consider the Bézier parameterization of an airfoil shape to be optimized:

$$P(t) = \sum_{k=0}^{n} B_n^k(t)\, P_k \tag{50}$$

where $t \in [0,1]$ is the curve parameter, n the polynomial degree, $B_n^k(t) = C_n^k\, t^k (1-t)^{n-k}$, a Bernstein polynomial, and $P_k = (x_k, y_k)$, a control point $(k = 0, 1, ..., n)$. Such polynomial representation facilitates the explicit calculation of the successive derivatives and thus the control of contact elements.

To illustrate this, the classical RAE2822 airfoil and its superimposed curvefit are shown on FIG. 1. This airfoil [Périaux et al. (1998)] has been widely used as a test case for aerodynamic design in the transonic flow regime.

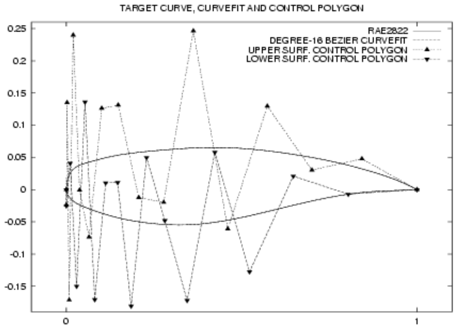


Figure 1: Curvefit of the upper and lower boundaries of the low-drag RAE2822 airfoil by two Bézier curves of degree 16 (from [Désidéri (2003)]).

When the geometry to be optimized is parameterized in such a way, a self-adaptive optimization strategy can be structured in the following steps:

1. Specify an initial *"support"*:

$$X^0 = \{x_k^0\} \ (k = 0, 1, ..., n) \tag{51}$$

2. Optimize the *"design variables"*:

$$Y^0 = \{y_k^0\} \ (k = 0, 1, ..., n) \tag{52}$$

   to minimize some physically-relevant functional of the distributed state calculated in a domain Ω dependent on (X^0, Y^0), such as the flow in the exterior of an airfoil;

3. Reconstruct a new support X^1 better adapted in some way to the shape identified by optimization; substitute X^1 to X^0, and return to Step 2. (See Subsection 3.5.)

3.2 Degree elevation

[Farin (1990)]

Multiply $P(t)$ by $(1-t)+t = 1$ and group homogeneous monomials together:

$$P(t) = \sum_{k=0}^{n} B_n^k(t)\, P_k = \sum_{k=0}^{n+1} B_{n+1}^k(t)\, P'_k \tag{53}$$

where $P'_0 = P_0$ and $P'_{n+1} = P_n$, and:

$$P'_k = \frac{C_n^{k-1}\, P_{k-1} + C_n^k\, P_k}{C_{n+1}^k} = \frac{k}{n+1}\, P_{k-1} + (1 - \frac{k}{n+1})\, P_k \qquad (1 \le k \le n) \tag{54}$$

This basic process is illustrated on FIG. 2. Note that it *preserves convexity* up to the limit $n \to \infty$ for which the control polygon converges to the Bézier curve itself. Hence a convex control polygon defines a convex Bézier curve.

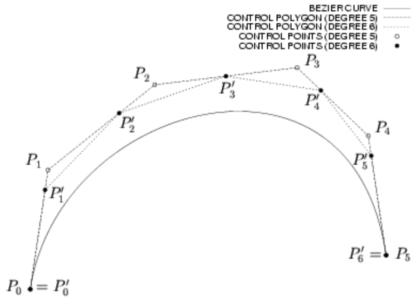

BEZIER CURVE
CONTROL POLYGON (DEGREE 5)
CONTROL POLYGON (DEGREE 6)
CONTROL POINTS (DEGREE 5) ○
CONTROL POINTS (DEGREE 6) ●
$P_0 = P'_0$
P_1
P'_1
P_2
P'_2
P'_3
P_3
P'_4
P_4
P'_5
$P'_6 = P_5$

Figure 2: Degree elevation: from a Bézier curve of degree 5, and its associated control polygon connecting the points $\{ P_k \}$, a new control polygon connecting the points $\{ P'_k \}$ is drawn, yielding an alternate parameterization, of degree 6, of the same geometrical arc (from [Désidéri (2003)]).

3.3 Nested supports and multilevel strategies

The above degree-elevation process is our building block to construct a hierarchy of rigorously-nested search spaces for the optimizer. As an example, FIG. 3 illustrate three nested supports of Bézier parameterizations of degree 4, 8 and 16.

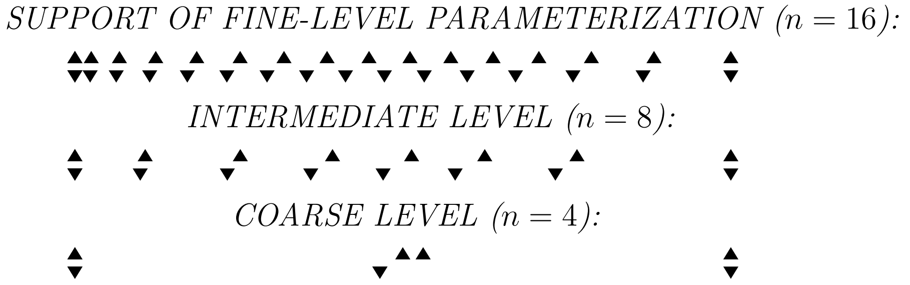

SUPPORT OF FINE-LEVEL PARAMETERIZATION ($n = 16$)*:*
INTERMEDIATE LEVEL ($n = 8$)*:*
COARSE LEVEL ($n = 4$)*:*

Figure 3: One-dimensional embedded parameterizations: the triangles represent the supports $X = \{x_k\}$ $(k = 0, 1, ..., n)$ of three nested Bézier parameterizations of degree $n = 4$, 8 and 16 of an RAE2822 airfoil, obtained from the first by 4 and 12 successive degree elevations (▲ and ▼ for upper and lower surface respectively) (from [Désidéri (2003)]).

3.4 Experimenting multilevel algorithms on a model problem

For purpose of numerical experimentation, we first consider the following model inverse-shape test problem:

$$\min \mathcal{J} := \mathcal{J}\Big(y(t)\Big) = \frac{p^{\alpha}}{\mathcal{A}} \tag{55}$$

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

$$p = \int_0^1 \sqrt{x'(t)^2 + y'(t)^2}\, \omega(t)\, dt\,,\ \mathcal{A} = \int_0^1 y(t)\, x'(t)\, \omega(t)\, dt \tag{56}$$

are, for specified $\omega(t) > 0$ and $\alpha > 1$, the pseudo-length of the arc, and the pseudo-area below the arc. Essentially all smooth unimodal graphs can be retrieved by this formulation [Désidéri and Zolésio (2005)]. In the following experiments, algorithms are tested for the case where the weight $\omega(t)$ and the exponent α are precisely those for which the unique minimum is realized by the half-thickness distribution of the RAE2822 airfoil.

In these experiments, the following optimization methods are compared in terms of iterative convergence, and accuracy:

- Basic method (single-parameterization throughout the convergence process)
- Progressive-degree elevation (from coarse to fine parameterization); analogous in logic to the so-called *"nested iteration"*;
- *FMOSA*, *"Fully Multilevel Optimum Shape Algorithm"*; analogous to the Full Multi Grid, *"FMG"*.

The employed *FMOSA* (without real optimization) has been a "saw-tooth correction algorithm" (e.g. [Wesseling (1992)] for terminology) whose skematic is depicted in Table 3. It involves both degree-elevation steps and (fine-to-coarse) *correction* steps after *smoothing* on a parameterization of upper level.

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

Table 3: Skematic of the employed *FMOSA* saw-tooth correction algorithm.

In the classical multigrid theory, it is well-known that both types of steps (from coarse level to fine, but also from fine level to coarse) are essential to the optimum linear convergence. In the context of shape optimization algorithms, the theoretical background is not so clearly established, to our understanding.

The relationship between accuracy and degree is assessed in Fig. 4 both in terms of functional value at convergence and shape sensitivity. Naturally, near the optimum, functional values are less sensitive since they are of second-order in terms of shape variations. Note that for degree $n = 10$, an incomplete convergence (50 iterations) results in a degraded shape definition, thus making the choice of a high-degree parameterization questionable, if not irrelevant.

The iterative convergence of the three methods is evaluated in Fig. 5, from which we draw the following main conclusions:

- Both multilevel algorithms are superior to the basic method.

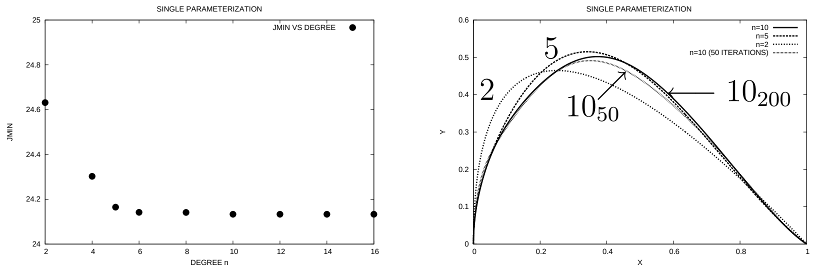


Figure 4: Accuracy vs degree at convergence of standard single-parameterization algorithm: achieved minimum functional value (left) and optimized shape (right); 10_{50} and 10_{200} refer to the results of 50 and 200 iterations for $n = 10$ (from [Désidéri et al. (2004)]).

- The progressive-degree elevation based on 3 levels in 3 times faster, as is classical for the nested iteration.
- The fully multilevel algorithm outperforms both, being here about 6 times faster than the basic method.
- Additionally, the experiment also demonstrates that the use of an improprer projection may cancel out the benefit of the multilevel strategy.

3.5 Self-adaptive multilevel algorithms

In our developments, the adaption phase is done as follows: among all L^2-approximants $\mathcal{B}_n(X,Y)$ of the current estimate of the optimized shape $\mathcal{B}_n(X^0,Y^0)$, determine the support $X = X^1$ for which the control polygon is the most regular in the sense of minimal total variation of the ordinates: $TV(Y^1)$.

The efficiency of the adaption procedure is demonstrated on Fig. 6 from which the following main conclusions are drawn:

- With a very low-degree parameterization ($n = 3$), surprisingly high accuracy can be achieved so long as the basic optimizer is coupled with adaption.
- With a high-degree parameterization ($n = 12$), the basic method fails to achieve sufficient iterative convergence to benefit from the accurate shape representation.
- Coupling a multilevel strategy with the adaption procedure results in both higher accuracy and lower computational cost (see iteration counts indicated step by step by figures on plot).

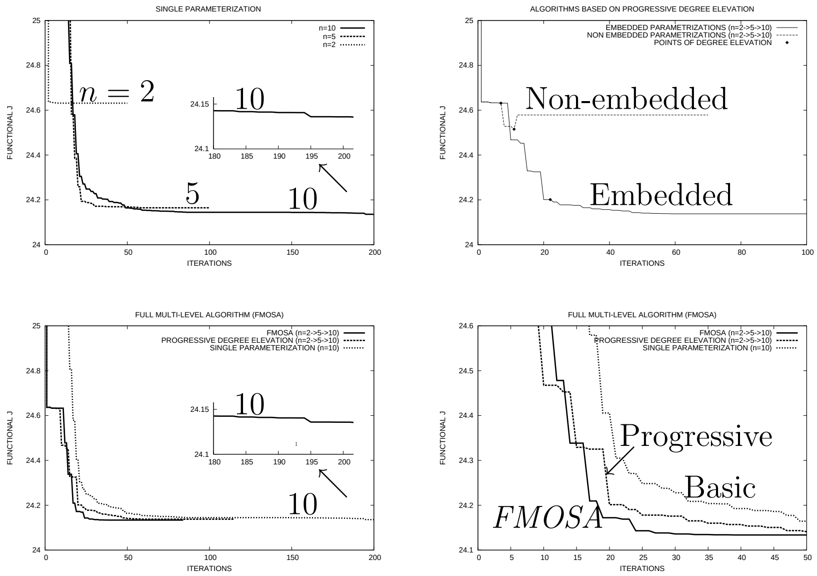


Figure 5: Iterative convergence of three methods; top: standard algorithm (left) and progressive degree-elevation (right) with proper/improper transfers; bottom: basic, progressive and *FMOSA* over 200 iterations (left) and 60 iterations (right) (from [Désidéri et al. (2004)]).

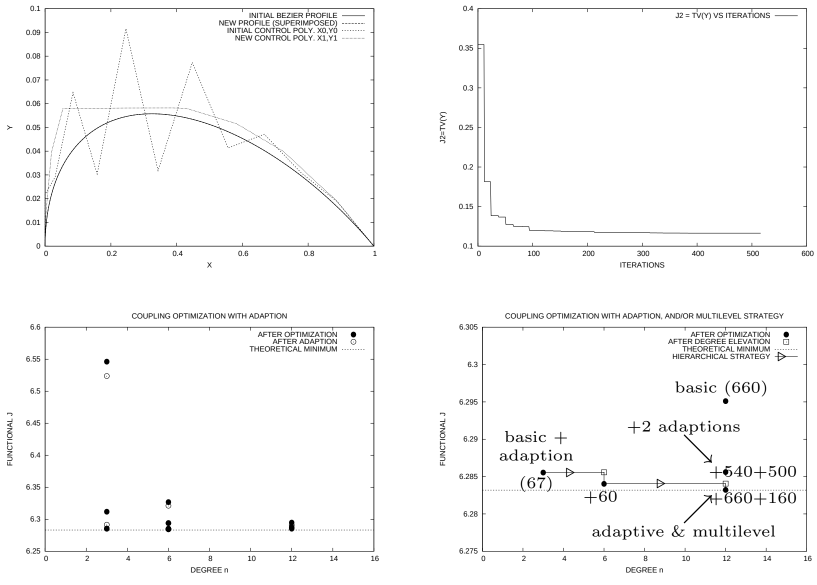


Figure 6: Parameterization adaption illustration; top: regularizing effect on control polygon (left; polygons with/without adaption), and iterative convergence of solver (right); bottom: accuracy vs degree: basic method and adaption (left), and adaption and hierarchy (right) (from [Désidéri et al. (2004)]).

4 Model problem and two-parameterization eigenmode analysis

In order to demonstrate the convergence mechanism of a basic two-level algorithm, we first set up a quadratic model reconstruction problem, and then develop an eigenmode analysis assuming a steepest-descent-type optimizer. For a more extensive presentation, we refer to [Désidéri (2006)].

4.1 Shape-reconstruction problem and basic two-level algorithm

We introduce the following general notations:
γ : shape to be optimized;
$J(\gamma)$: intrinsically-defined functional to be minimized;
X^0 , $X^{0'}$: nested supports of Bézier parameterizations of degree n , $n' \in \mathbb{N}$ $(n' < n)$;
$X^0 = E^n_{n'}\, X^{0'}$;
$E^n_{n'}$ :: matrix associated with the linear process of $n - n'$ degree elevations;
$B_n(t)^T = (B^0_n(t), B^1_n(t), ..., B^n_n(t))$: $1 \times (n+1)$-vector of Bernstein polynomials.

Particularizing the shape γ to be the parameterized arc of Bézier curve of degree n and control polygon (X^0, Y^0) $(a \le t \le b)$, here denoted:

$$\gamma = [x(t), y(t)]_a^b = [B_n(t)^T\, X^0, B_n(t)^T\, Y^0]_a^b \tag{57}$$

the *non-intrinsic*, or *parametric function* to be minimized, for fixed support $X = X^0$, is the following:

$$j_n(Y) = J\left([B_n(t)^T\, X^0, B_n(t)^T\, Y]_0^1\right) \ , \ Y \in \mathbb{R}^{n+1} \tag{58}$$

Consequently, a basic, perhaps naïve, *two-level correction-type ideal algorithm* can be defined as follows, assuming the classical steepest-descent method is used as relaxation method in the upper-level iteration:

1. Upper-level relaxation phase:
   K sweeps of descent method $(1 \le k \le K)$:

   $$Y^k = Y^{k-1} - \rho_k\, j'_n(Y^{k-1}) \tag{59}$$

   yielding approximate optimum shape associated with the design vector $Y^K \in \mathbb{R}^{n+1}$ (and support X^0).

2. Coarse-level correction phase:
   solve completely:

   $$\min_{Y' \in \mathbb{R}^{n'+1}} j_n\left(Y^K + E^n_{n'}\, Y'\right) \tag{60}$$

Consistently with the usual multigrid terminology, one such algorithm is said to be a *correction* method because the unknown of the coarse-level iteration is defined to be a correction vector, here Y', to be prolongated, here by the degre-elevation operator $E^n_{n'}$, and added to the freshest update of the unknown vector coming from the upper-level iteration, here Y^K; thus Y' is a coarse approximation of a shape correction, and not directly of the unknown shape itself. This distinction is relevant in the nonlinear case only. Besides, the algorithm is said to be *ideal* because the correction phase, when realized iteratively, is, for purpose of analysis, assumed to be continued to full convergence. Additionally, the above method is a *saw-tooth* algorithm, because the cycle is unsymmetrical, here as a simple possibility: the cycle terminates by the coarse-level iteration, in contrast for example, with the commonly-used *V-cycle* that would in the present context also include a post-relaxation similar to (59) on the upper-level.

We further particularize the model problem to be a *shape-reconstruction* or *shape-inverse problem* in which:

$$\gamma \,:\, y(x)\,;\ \text{target: } \bar{y}(x) \tag{61}$$

$$J(\gamma) := \int_\gamma \tfrac{1}{2}\left[y(x)-\bar{y}(x)\right]^2\,dx \tag{62}$$

$$j_n(Y) = \int_0^1 \tfrac{1}{2}\left[B_n(t)^T\,(Y-\bar{Y})\right]^2 \underbrace{nB_{n-1}(t)^T\,\Delta X^0}_{x^0(t)'=\frac{d}{dt}B_n(t)^T X^0}\,dt \tag{63}$$

($\bar{Y}$ is the parameter vector associated with the target curve, here assumed, without great loss of generality, to be a Bézier curve of degree n and support X^0. 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):

$$j_n'(Y) = A(X^0)Y - b(X^0,\bar{Y}) \tag{64}$$

where:

$$\boxed{A(X^0) = \int_0^1 \underbrace{B_n(t)\,B_n(t)^T}_{\substack{(n+1)\times(n+1)\\ \text{matrix}}}\quad \underbrace{nB_{n-1}(t)^T\,\Delta X^0}_{\substack{\text{linear form}\\ \text{in } X^0}}\,dt} \tag{65}$$

The matrix $A(X^0)$ is, of course, real-symmetric definite positive and can thus be diagonalized by an orthogonal transformation Ω_n:

$$\boxed{A(X^0) = \Omega_n\,\Lambda_n\,\Omega_n^T} \qquad \Lambda_n = \begin{pmatrix} \lambda_0 & & & \\ & \lambda_1 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix} \tag{66}$$

The diagonal matrix Λ_n has real-positive eigenvalues, arranged in increasing order,

$$0 \le \lambda_0 \le \lambda_1 \le \ldots \le \lambda_n \tag{67}$$

and the column-vectors of the orthogonal matrix Ω_n are the associated eigenvectors

$$\Omega_n^T\,\Omega_n = \Omega_n\,\Omega_n^T = I \tag{68}$$

In the particular case of a uniform support,

$$x^0(t)' = nB_{n-1}(t)^T \,\Delta X^0 \equiv 1 \tag{69}$$

and the matrix A reduces to the simple form:

$$\boxed{A = \int_0^1 B_n(t)\, B_n(t)^T \, dt = \{A_{\alpha,\beta}\}} \tag{70}$$

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

$$\boxed{A_{\alpha,\beta} = \frac{1}{2n+1} \, \frac{C_n^\alpha \, C_n^\beta}{C_{2n}^{\alpha+\beta}}} \tag{71}$$

4.2 Eigenvalue estimates

4.2.1 Spectral radius, $\rho(A) = \lambda_{\max} = \lambda_n$

Case of a uniform support Let $U = (1, 1, ..., 1)^T$ and $V = A\,U$, so that:

$$V = \int_0^1 B_n(t)\, B_n(t)^T \, U \, dt \tag{72}$$

But,

$$B_n(t)^T \, U = \sum_{k=0}^{n} B_n^k(t) = 1 \tag{73}$$

and:

$$V_k = \int_0^1 B_n^k(t)\, dt = C_n^k \int_0^1 t^k \,(1-t)^{n-k} \, dt = \frac{1}{n+1} = \lambda_n^0 \, U_k \tag{74}$$

which proves that the vector U is indeed an eigenvector associated with the eigenvalue λ_n^0, and second that $\lambda_n^0 = \|A\|_\infty$. Hence, λ_n^0 is also the spectral radius of A:

$$\boxed{\lambda_n^0 = \frac{1}{n+1} = \|A\|_\infty = \rho(A) = \lambda_{\max}^0} \tag{75}$$

Remark. The above calculation also yields the identity:

$$\forall n \, , \; \forall \alpha \leq n \; : \; \sum_{\beta=0}^{n} \frac{C_n^\alpha \, C_n^\beta}{C_{2n}^{\alpha+\beta}} = \frac{2n+1}{n+1} \tag{76}$$

General support, upper bound To establish this upper bound on $\lambda_{\max}(X)$ over all admissible supports $X = \{x_k\}$ $(k = 0, 1, ..., n)$ associated with monotone-increasing sequences $\{x_k\}$ for which $\forall t\,,\ x'(t) \geq 0$, we recall that since the matrix $A(X)$ is real-symmetric, definite positive, its largest eigenvalue, $\lambda_{\max}$, is equal to its induced Euclidean norm:

$$\lambda_{\max} = \|A(X)\|_2 \tag{77}$$

Therefore,

$$\lambda_{\max} = \max_{Y \in \mathbb{R}^{n+1}\,,\ \|Y\|_2=1} Y^T A(X)\, Y = Y_M^T A(X)\, Y_M \tag{78}$$

for a certain normalized vector $Y_M \in \mathbb{R}^{n+1}\,,\ \|Y_M\|_2 = 1$. But:

$$Y_M^T A(X)\, Y_M = \int_0^1 Y_M^T B_n(t)\, B_n(t)^T Y_M\, x'(t)\, dt = \int_0^1 \Big(B_n(t)^T Y_M\Big)^2 x'(t)\, dt \tag{79}$$

in which the two factors in the integrand are positive; consequently:

$$0 \leq Y_M^T A(X)\, Y_M \leq \|x'\|_\infty \int_0^1 \Big(B_n(t)^T Y_M\Big)^2 dt \tag{80}$$

But $x'(t) = n\, B_{n-1}(t)^T \Delta X$, and since the sequence $\{x_k\}$ increases from 0 to 1 so that $0 \leq x_k - x_{k-1} \leq 1$, one has:

$$0 \leq x'(t) \leq n \sum_{k=1}^{n} B_{n-1}^{k-1}(t) = n \tag{81}$$

and:

$$\|x'\|_\infty \leq n \tag{82}$$

Additionally,

$$\int_0^1 \Big(B_n(t)^T Y_M\Big)^2 dt \leq \max_{Y \in \mathbb{R}^{n+1}\,,\ \|Y\|_2=1} \int_0^1 \Big(B_n(t)^T Y\Big)^2 dt = \underbrace{\lambda_{\max}\Big(A(X^0)\Big)}_{\lambda^0_{\max}} = \frac{1}{n+1} \tag{83}$$

in which X^0 corresponds to the case of a uniform support for which the maximum eigenvalue has been established in the previous paragraph.

Ultimately, the following upper bound holds:

$$\boxed{\rho\Big(A(X)\Big) = \lambda_{max}\Big(A(X)\Big) \leq \frac{n}{n+1} < 1} \tag{84}$$

This upper bound also applies to any diagonal block of matrix $A(X)$.

4.2.2 Characterization of the smallest effective eigenvalue, λ_1

We now restrict our attention to a parametric optimization that only has $n-1$ degrees of freedom, obtained for fixed support X, by enforcing homogeneous boundary conditions to the function $y(t)$. For this, the first and last components of the vector Y are set to 0 so that:

$$Y = F\,\eta \qquad (\eta \in \mathbb{R}^{n-1}) \tag{85}$$

where the $(n+1)\times(n-1)$ matrix F is obtained by banding the identity matrix I_{n-1} above and below by rows of 0. In this context, the challenge is to estimate the smallest eigenvalue λ_1 of the $(n-1)\times(n-1)$ central diagonal block $F^T\,A\,F$.

Let $\eta_1 \in \mathbb{R}^{n-1}$ be the eigenvector, $Y_1 = F\,\eta_1$, and $y_1(t) = B_n(t)^T\,Y_1$ the associated eigenfunction. Then

$$\lambda_1\,Y_1 = A\,Y_1 = \int_0^1 B_n(t)\,B_n(t)^T\,Y_1\,x'(t)\,dt = \int_0^1 B_n(t)\,y_1(t)\,x'(t)\,dt \tag{86}$$

Multiplying by $B_n(\tau)^T$ yields:

$$\boxed{\begin{aligned}\lambda_1\,y_1(\tau) &= \int_0^1 K(t,\tau)\,y_1(t)\,dt\\ K(t,\tau) &:= B_n(\tau)^T\,B_n(t)\,x'(t)\end{aligned}} \tag{87}$$

By multiplying $\lambda_1\,y_1(\tau)$ by $y_1(\tau)$ and integrating w.r.t. τ from 0 to 1, one gets:

$$\boxed{\lambda_1 = \frac{\iint_{[0,1]\times[0,1]} K(t,\tau)\,y_1(\tau)\,y_1(t)\,d\tau\,dt}{\int_0^1 y_1(\tau)^2\,d\tau}} \tag{88}$$

In the case of a uniform support, the above Rayleigh quotient simplifies to the following:

$$\boxed{\lambda_1^0 = \frac{\left\|\int_0^1 B_n(t)\,y_1(t)\,dt\right\|_2^2}{\int_0^1 y_1(t)^2\,dt}} \tag{89}$$

in which $\|\ \|_2$ denotes the Euclidian norm in $\mathbb{R}^{n+1}$:

$$\left\|\int_0^1 B_n(t)\,y_1(t)\,dt\right\|_2^2 = \sum_{k=0}^{n+1}\left(\int_0^1 B_n^k(t)\,y_1(t)\,dt\right)^2 = \sum_{k=0}^{n+1}\left(y_1(t_k)\int_0^1 B_n^k(t)\,dt\right)^2 \tag{90}$$

for certain intermediate values $\{t_k\}$, since the mean-value theorem applies since $B_n^k(t) \geq 0$.

However, only a very accurate approximation of the unknown polynomial function $y_1(t)$, injected in (88) or (89), could yield a valuable estimate for λ_1.

4.3 Eigenmodes, frequency pairing

In the absence of an expression for the eigensystem in closed form, and for purpose of illustration, the diagonalization of the matrix has been accomplished numerically, in the case of homogeneous boundary conditions as in subsection 4.2.2, for which only the central diagonal block $F^T A F$ matters, and assuming a uniform support X.

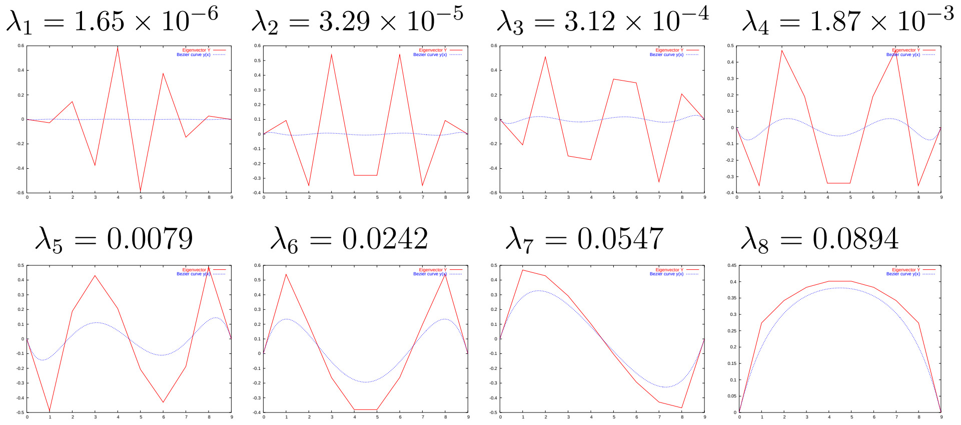
The eigenvectors of this block, associated with the fixed support X, define control polygons that are represented on FIG. 7 in the case $n-1=8$ of a *fine* parameterization, for increasing eigenvalue λ_m ($m=1,2,...,8$). The corresponding Bézier curves are also plotted on the same figure, but these are not normalized; the normalized Bézier curves are shown on FIG. 8. Since the support is uniform, these Bézier curves admit simple polynomial representations: $x(t)=t$, $y(t)=y(x)$ of degree 8. From these representations, it appears that these polynomials exhibit patterns similar to Fourier modes in terms of sign alternations. The novelty here, in contrast with the conventional setting of multigrid for a typical elliptic PDE, is that the matrix A which defines the algebraic system to be solved is a form of *discrete integration, instead of differential, operator.* As a result, as the eigenvalue λ_m increases, the apparent *frequency* diminishes, and this constitutes an unusual situation.

Second, the diagonalization is made again for the embeded *coarser* parameterization of degree $n'=5$. The corresponding control polygons are shown in raw form on top of FIG. 9. Applying degree-elevation four times to these polygons yields equivalent polygons, that is defining the same Bézier curves, but associated with the fine support X; these are shown below on the same figure, thus permitting a direct comparison with the former fine-parameterization eigensystem. Evidently, the coarse-parameterization eigenmodes are very close approximations to the upper-half eigenmodes of the fine parameterization associated with the *large* eigenvalues, that is, the *low frequencies.*

Consequently, the steepest-descent method defined in (59) and viewed as a point-Jacobi iteration with matrix A, filters out predominently the low-frequency modes. Therefore, it cannot be considered as a *smoother* in the basis of the eigenvectors of matrix A. Hence, to our understanding, the classical modal-analysis of multigrid cannot be applied straightforwardly. The evidence of effectiveness of the multilevel strategy provided in Section 3 should be supported differently. Instead of that, we propose, in the next section, a new formulation for the coarse-level iteration to be compatible with this eigensystem analysis. Before this, we provide the eigenvalue spectrum and condition number for the case $n-1=8$.

4.4 Numerical spectrum, condition number

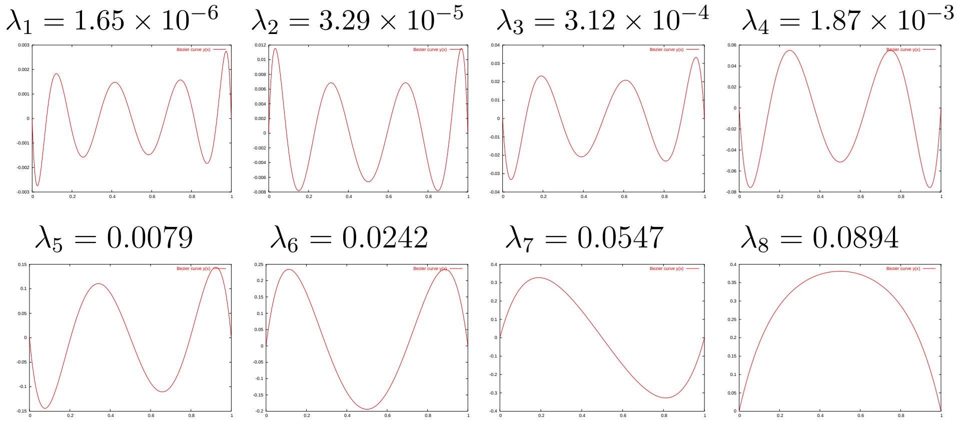
The numerical spectrum of eigenvalues of the block $F^T A F$ in case $n=9$ is given on FIG. 10. It exhibits certain very small eigenvalues even though the degree n is not very high, indicating that the condition number of the system increases very rapidly with the degree, as confirmed by the results of FIG. 11 showing an exponential explosion. From these data, one may question how many modes can actually be resolved by a standard iteration, giving support to the alledgement according which only a hierarchical approach can be effective on all the modes.



$\lambda_1 = 1.65 \times 10^{-6}$ $\lambda_2 = 3.29 \times 10^{-5}$ $\lambda_3 = 3.12 \times 10^{-4}$ $\lambda_4 = 1.87 \times 10^{-3}$

$\lambda_5 = 0.0079$ $\lambda_6 = 0.0242$ $\lambda_7 = 0.0547$ $\lambda_8 = 0.0894$

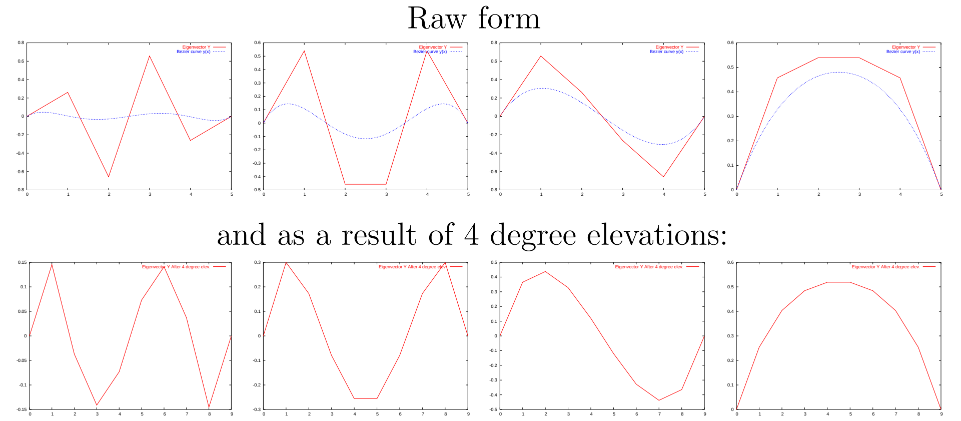
Figure 7: Upper-level eigenmodes, $m = 1, 2, ..., n - 1 = 8$: eigenvectors of the central diagonal block $F^T A F$ arranged by increasing λ_m and corresponding Bézier curves.



$\lambda_1 = 1.65 \times 10^{-6}$ $\lambda_2 = 3.29 \times 10^{-5}$ $\lambda_3 = 3.12 \times 10^{-4}$ $\lambda_4 = 1.87 \times 10^{-3}$

$\lambda_5 = 0.0079$ $\lambda_6 = 0.0242$ $\lambda_7 = 0.0547$ $\lambda_8 = 0.0894$

Figure 8: Upper-level eigenmodes, $m = 1, 2, ..., n - 1 = 8$: renormalized Bézier curves represented by the discrete eigenvectors.



Raw form

and as a result of 4 degree elevations:

Figure 9: Lower-level eigenmodes, $m = 1, 2, ..., n' - 1 = 4$: eigenvectors of the central diagonal block $F^T A F$ arranged by increasing λ'_m and corresponding Bézier curves; top: on the coarse-parameterization support; bottom: on the fine-parameterization support (after degree elevation).

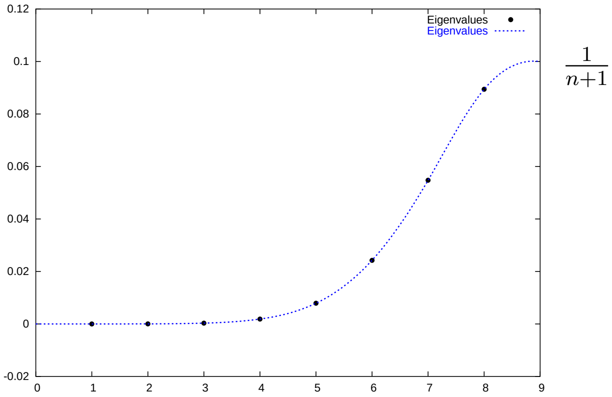


Figure 10: Spectrum of eigenvalues, $n = 9$ (corresponding condition number: $\sim 54,000$).

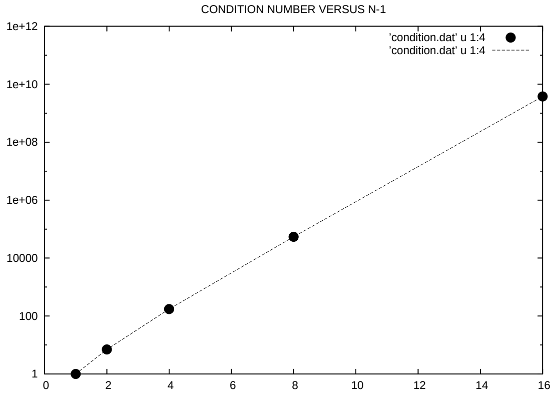


Figure 11: Condition number versus degree.

5 Conclusions

In this first part, we have reviewed some previous work on multilevel parametric optimization relying on the construction of a sequence of nested Bézier parameterizations. The approach is indeed effective. Progressive degree-elevation algorithms have been shown by numerical experiments to converge faster; the gain factor in convergence rate is equal to the number of levels, as it is classical for the *nested iteration.* This suggests that a simple and convenient algorithm would be to perform a few iterations with all the intermediate degrees progressively, but this has not been tested. A saw-tooth algorithm analogous in multilevel-structure to the Full Multi-Grid (FMG) method is even more efficient, although it is difficult to demonstrate a form of *asymptotic linear convergence* since the degrees are not large in practice. Additionally, a parametertization-adaption procedure was proposed and shown to be very effective.

A simple *shape-reconstruction model problem* has been set up for purpose of analysis. It consists of the minimization of a quadratic functional, so that the steepest-descent algorithm utilized as the optimizer could be viewed as the classical point-Jacobi iteration applied to a certain matrix system. The eigensystem of this matrix has been analyzed for two nested parameterizations, permitting us to make the following observations:

1. The eigenmodes that actually are polynomials, are so-to-speak "Fourier"-like in terms of sign alternation.

2. Lower-level eigenmodes are indeed close to the lower-frequency modes of the upper-level eigensystem.

3. But, the pairing between the frequency and the generic eigenvalue is inverse, as it is for an integration operator.

Recently, in [Désidéri (2006)], these observations have led us to define a new formulation for the ideal two-level iteration in which the reduction of dimension, in the *coarse-level correction phase*, is performed in a subspace in which the error vector is actually smooth, as a result of the upper-level relaxation. This alternative is currently being tested.

Lastly, concerning the parameterization-adaption procedure, we note that it was orginally defined to be a regularization procedure, effectively reducing the total variation associated with the control polygon of the Bézier representation. This makes sense *a posteriori* since the eigensystem analysis has shown that the optimization phase is inversely an anti-smoothing process.

Acknowledgements

The authors are indebted to many collaborators and coauthors, and particularly to Nathalie Blaszka-Marco, Badr Abou El Majd and Aleš Janka who contributed importantly to our scientific efforts.

References

Abou El Majd, B., Désidéri, J.-A., Do, T. T., Fourment, L., Habbal, A., and Janka, A. (2005). Multilevel strategies and hybrid methods for shape optimization and applications to aerodynamics and metal forming. In Schilling, R., Haase, W., Périaux, J., Baier, H., and Bugeda, G., editors, *Evolutionary and Deterministic Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems Conference (EUROGEN 2005)*, Munich, Germany. ©FLM, Munich, 2005.

Bornemann, F. and Deuflhard, P. (1996). The cascadic multigrid method for elliptic problems. *Numer. Math.*, 75:135–152. Springer International.

Brandt, A. (1977). Multi-level adaptive solutions to boundary value problems. *Math. Comput.*, 31:333–390.

Briggs, W. L. (1991). *A Multigrid Tutorial.* SIAM, Philadelphia, third printing edition.

Dervieux, A., Marco, N., Held, C., and Koobus, B. (2001). Hierarchical Principles and Preconditioning for Optimum Design and Identification. In *et al*, J. P., editor, *Innovative Tools for Scientific Computation in Aeronautical Engineering*, Handbooks on Theory and Engineering Applications of Computational Methods. CIMNE, Barcelona.

Désidéri, J.-A. (1998). *Modèles discrets et schémas itératifs. Application aux algorithmes multigrilles et multidomaines.* Editions Hermès, Paris. (352 p.).

Désidéri, J.-A. (2003). *Numerical Methods for Scientific Computing, Variational Problems and Applications, E. Heikkola, Y. Kuznetsov, P. Neittaanmäki and O. Pironneau eds.*, chapter Hierarchical Optimum-Shape Algorithms Using Embedded Bézier Parameterizations. CIMNE.

Désidéri, J.-A. (2006). Two-level ideal algorithm for parametric shape optimization. *J. Numer. Math.* to appear.

Désidéri, J.-A. and Janka, A. (2004). Multilevel shape parameterization for aerodynamic optimization – application to drag and noise reduction of transonic/supersonic business jet. In Neittaanmäki, P., Rossi, T., Korotov, S., Onate, E., Périaux, J., and Knörzer, D., editors, *European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004*, Jyväkyla, Finland.

Désidéri, J.-A., Majd, B. A. E., and Janka, A. (2004). Nested and self-adaptive bézier parameterizations for shape optimization. In *Conference on Control, PDEs and Scientific Computing*, Beijing, China. Science Press Beijing-New York. to appear.

Désidéri, J.-A. and Zolésio, J.-P. (2005). Inverse shape optimization problems and application to airfoils. *Control and Cybernatics*, 34(1).

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

Kronsjö, L. and Dahlquist, G. (1972). On the design of nested iterations for elliptic difference equations. *BIT*, 12:63–71. (also *BIT*, 11, 1971).

Périaux, J., Bugeda, G., Chaviaropoulos, P. K., Giannakoglou, K., Lantéri, S., and Mantel, B., editors (1998). *Optimum Aerodynamic Design & Parallel Navier-Stokes Computations.* Vieweg, Braunschweig/Wiesbaden, Germany. ECARP European Computational Aerodynamics Research Project.

Varga, R. S. (2000). *Matrix Iterative Analysis.* Springer. ISBN 3540663215.

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