

Multilevel gradient method with Bézier parametrisation for aerodynamic shape optimisation

Massimiliano Martinelli

Scuola Normale Superiore di Pisa

Piazza dei Cavalieri, 7 - 56126 Pisa (Italy)

`max.martinelli@sns.it`

François Beux

Scuola Normale Superiore di Pisa

Piazza dei Cavalieri, 7 - 56126 Pisa (Italy)

`fbeux@sns.it`

Let us consider the classical optimal aerodynamic shape problem expressed, in a fully discrete context, as follows:

$$(0.1) \quad \text{Find } \bar{\gamma} \in \mathbb{R}^p \text{ such that } j(\bar{\gamma}) = \min_{\gamma \in \mathbb{R}^p} j(\gamma)$$

where γ represents a set of discrete control variables while j is a cost functional based on aerodynamic criteria, and thus, verifies $j(\gamma) = J(\gamma, W(\gamma))$, $W(\gamma)$ being the values at each grid-point of the flow variables obtained solving the discretised governing flow equations:

$$(0.2) \quad \Psi(\gamma, W(\gamma)) = 0$$

Since the evaluation of the cost functional requires the solution of the governing flow equations (0.2), each optimisation iteration is characterised by a high computational cost if complex flow modelling is considered. In this context, the improvement of optimisation algorithm efficiency is crucial. In [2] a multilevel strategy has been defined to make the convergence rate of a gradient-based method almost independent of the number of control parameters. More particularly, considering shape grid-point coordinates as design variables, a hierarchical parametrisation was defined considering different subsets of parameters extracted from the complete parameterisation, which can be prolonged to the higher level by linear mapping. Thus, let us consider, at level l , a linear application $P^{(l)}$ and its associated matrix $M^{(l)} \in \mathbb{R}^{p \times n_l}$:

$$\begin{aligned} P^{(l)} : \mathbb{R}^{n_l} &\longrightarrow \mathbb{R}^p \\ \alpha &\longmapsto P^{(l)}(\alpha) = M^{(l)}\alpha \end{aligned}$$

the following descent algorithm, which can be also interpreted as a preconditioned gradient method, is then obtained:

$$(0.3) \quad \gamma_{r+1} = \gamma_r - \omega_r M^{(l)}(M^{(l)})^T g_r \quad \text{with } \omega_r \in \mathbb{R} \text{ and } g_r \in \mathbb{R}^p$$

g_r being the cost functional gradient at iteration r .

In [2], $P^{(l)}$ is defined by iteratively using an hermitian interpolation of degree 3 on the nested parameterisation.

Note that using shape grid-point coordinates as design variables is a natural choice since this parameterisation allows a direct correlation with the explicit representation of the shape in the discrete cost functional. Nevertheless, in this case, non-smooth profiles can appear during the shape optimisation process, and moreover, the large number of variables involved has also a negative effect on the convergence rate. As a matter of fact, the multilevel strategy was defined exactly in order to reduce these drawbacks.

Alternatively, a polynomial representation of the shape is often used allowing a more compact description with only few control parameters. In particular, Bézier curves can be used to characterise the shape considering as design variables the Bézier control points. In [3] a multilevel strategy for shape optimisation has been already defined also for Bézier curves, even if in the context slightly different since this study was not focused on a gradient-type method. On the contrary, in the present study, a new multilevel strategy based on the use of Bézier control points, but in the context of a gradient-based method, is proposed.

A Bézier curve of degree n is defined according to the following parametrisation:

$$(0.4) \quad \begin{cases} x(t) = \sum_{q=0}^n B_n^q(t) x_q \\ y(t) = \sum_{q=0}^n B_n^q(t) y_q \end{cases}$$

in which (x_q, y_q) are the $n + 1$ control points, $B_n^q(t)$ corresponds to the Bernstein polynomials, t being a parameter varying between 0 and 1.

Let us consider y_0^S, \dots, y_m^S the ordinates of the shape grid-points as control variables and $m + 1$ parameters t_k given ($t_0 = 0 < t_1 < \dots < t_m = 1$). Then it clearly appears that the ordinates of the Bézier control points, i.e. $\alpha = (y_0, \dots, y_n)^T$, are directly related with the control variables $\gamma = (y_0^S, \dots, y_m^S)^T$ by (0.4) in which $y(t_k) = y_k^S$ for $k = 0, \dots, m$, and thus, the following linear operator from α to γ can be considered:

$$(0.5) \quad \begin{array}{ccc} P : \mathbb{R}^{n+1} & \longrightarrow & \mathbb{R}^{m+1} \\ \alpha & \longmapsto & \gamma = P(\alpha) = M\alpha \quad \text{in which } M_{ij} = B_n^j(t_i). \end{array}$$

Due to the linearity of P , a multilevel strategy using a set of subspaces of increasing dimension can be defined as done in [2]. More precisely, at a particular level l , the following steps should be performed:

1. Construction of an adequate parametrisation at level l :

Find $X^{(l)} = (x_0^{(l)}, \dots, x_{n_l}^{(l)})^T$ with $n_l > n_{l-1}$ and $T^{(l)} = (t_0^{(l)}, \dots, t_{m+1}^{(l)})^T$ such that

$$(0.6) \quad x_k^S = x(t_k^{(l)}) = \sum_{q=0}^{n_l} B_{n_l}^q(t_k^{(l)}) x_q^{(l)} \quad \text{for } k = 0, \dots, m$$

2. Computation of the descent direction:

$$(0.7) \quad d_r^{(l)} = M^{(l)}(M^{(l)})^T g_r \quad \text{with } M_{ij}^{(l)} = B_{n_l}^j(t_i^{(l)})$$

Note that in [3], the control variables are directly the Bézier control points while the level change is based on the property of degree-elevation of the Bézier curves. Indeed, this property allows to increase in a natural way the degree and the number of control points of the Bézier curves. More precisely, given a Bézier curve of degree n associated to the $n+1$ control points $P_k = (x_k, y_k)^T$, the same geometrical curve can be also understood as a Bézier curve of degree $n+1$ considering a new set of $n+2$ control points $P'_k = (x'_k, y'_k)^T$. P'_k can be obtained from P_k as follows:

$$(0.8) \quad P'_k = \frac{k}{n+1} P_{k-1} + \left(1 - \frac{k}{n+1}\right) P_k \quad \text{for } k = 0, \dots, n+1$$

The degree elevation is used, here, to define an adequate parametrisation at level l . Indeed, given $X^{(0)}$ consistent with $T^{(0)}$, i.e. such that (0.6) is verified, one can construct $X^{(l)}$ for $l > 0$ by applying successively (0.8) until to obtain $n_l + 1$ abscissae. In this way, since the distribution of the parameters t over the Bézier curve does not change by degree elevation, the parameters t_k keep unchanged on all the levels, i.e. $t_k^{(l)} = t_k^{(0)}$ for $k = 0, \dots, m+1$ and for all $l > 0$.

If the abscissae of the shape grid-points (i.e. x_k^S for $k = 0, \dots, m$) are uniformly distributed, then choosing an uniform distribution for both $X^{(0)}$ and the parameters t_0, \dots, t_{m+1} , eq. (0.6) is verified. Nevertheless, as pointed out in [4], the abscissae location can influence the quality of the optimisation, and in particular, an uniform distribution should not be, in general, a very good choice. Thus, an adaptation of the parametrisation with respect to the particular type of shape considered could be also envisaged but, here, only in a preprocessing step, i.e. before the optimisation loop.

Let us consider the operator $P^{(L)}$ which relates the finest level L of Bézier control points and the shape grid-points ordinates:

$$\begin{aligned} P^{(L)} : \mathbb{R}^{n_L} &\longrightarrow \mathbb{R}^{m+1} \\ \alpha &\longmapsto M^{(L)}\alpha \quad \text{in which } M_{ij}^{(L)} = B_{n_L}^j(t_i) \end{aligned}$$

then, at a particular level l , one can directly relate the current set of Bézier control points with the shape grid-points as done in (0.5), or first, go to finest sub-level by degree-elevation and then use $P^{(L)}$. Thus, the algorithm defined by (0.3) and (0.7) can be also rewritten as follows (the two ways should be equivalent from a theoretical view point, even if they can differ in the implementation):

$$\gamma_{r+1} = \gamma_r - \omega_r M^{(L)} D_l^L (D_l^L)^T (M^{(L)})^T g_r$$

in which D_l^L is the matrix corresponding to apply $n_L - n_l$ times the degree-elevation algorithm (0.8).

A first validation of the present optimisation procedure has been done on a classical inverse problem (see e.g. [1]) for an inviscid subsonic flow over a 2D nozzle (the flow is modelled by the Euler equations). The multilevel strategy has been coupled with an exact discrete gradient computation as well as a one-shot approach associated to a gradient approximation by divided finite difference as described in [5].

REFERENCES

1. F. Beux, "Shape optimization of an Euler flow in a nozzle", *Notes on Numerical Fluid Mechanics*, **55**, Périaux et al. eds, Vieweg Publishing, 115-131 (1994).

2. F. Beux and A. Dervieux, “A hierarchical approach for shape optimization”, *Engineering Computations* **11**, 25-48 (1994).
3. 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.
4. J.-A. Désidéri, B. Abou el Majd and A. Janka, “Nested and Self-Adaptive Bézier Parameterizations for Shape Optimization”, Proc. of International Conference on Control, Partial Differential Equations and Scientific Computing, 2004, Beijing. Sciences Press, 1-19 (2005).
5. C. Held and A. Dervieux, “One-shot airfoil optimisation without adjoint” *Computers and Fluids*, **31**/8, 1015-1049 (2002).