AIAA JOURNAL
Vol. 49, No. 4, April 2011

Extension to the Class-Shape-Transformation Method Based on B-Splines

Michiel H. Straathof[*] and Michel J. L. van Tooren[†]
Delft University of Technology, 2629 HS Delft, The Netherlands

DOI: 10.2514/1.J050706

A new parameterization method to enhance shape optimization for aircraft design was developed based on the class-shape-transformation method. This method was implemented in a tool that combined all aspects of the aerodynamic design process: parameterization, aerodynamic analysis, and optimization. The parameterization method used a combination of Bernstein polynomials and B-splines to allow for both local and global control of a shape. Additionally, the use of B-splines made it possible to efficiently handle volume constraints, which are very common in aircraft design. The parameterization method was coupled to two different aerodynamic analysis tools: the commercial panel method code VSAERO and an in-house Euler code. As a first test case, a wing was optimized for maximum lift-to-drag ratio in subsonic conditions using VSAERO and various optimization algorithms. It was shown that the optimizer was capable of arriving at wing shapes that demonstrated laminar flow up to 80%, without violating the implemented volume constraints. As a second test case, an airfoil was optimized for lift-to-drag ratio in transonic conditions using the in-house Euler solver and a gradient-based optimizer. Refining the shape using B-splines proved to be an efficient way of increasing the design freedom, resulting in additional improvements in the drag divergence Mach number in the order of 0.02.

Nomenclature

a	=	first class function scaling factor
b	=	second class function scaling factor
C	=	class function
c	=	chord length, m
C_d	=	three-dimensional drag coefficient
C_l	=	three-dimensional lift coefficient
C_p	=	pressure coefficient
c_d	=	two-dimensional drag coefficient
c_{d_w}	=	wave drag coefficient
c_l	=	two-dimensional lift coefficient
c_r	=	root chord, m
c_t	=	tip chord, m
L/D	=	lift-to-drag ratio
L/D_w	=	lift-to-wave drag ratio
M	=	Mach number
N	=	B-spline basis function
n	=	number of B-spline control points $+ 1$
$N1$	=	first exponent of class function
$N2$	=	second exponent of class function
$\mathbf{p}$	=	vector of B-spline control points
R	=	refinement function
S	=	shape function
S_{in}	=	inner section span, m
S_{out}	=	outer section span, m
t	=	B-spline knot value
u	=	parametric variable
x	=	chordwise position, m
y	=	spanwise position, m
z	=	distance from airfoil centerline, m
α	=	angle of attack, deg
ζ	=	relative distance from airfoil centerline
η	=	relative spanwise position
Λ	=	sweep angle, deg
λ	=	taper ratio
ψ	=	relative chordwise position

I. Introduction

THE three main components of the tool presented in this paper are shape parameterization, aerodynamic analysis, and optimization. The goal of the work done is to optimize wing shapes for maximum lift-to-drag ratio subject to specific (volume) constraints; hence, it is solely focused on aerodynamics. To design a complete aircraft shape, other disciplines, such as structures, propulsion, and stability and control, should be included. This is currently being investigated. However, the use of the aerodynamic part of the tool is not limited to wings only. It could be used for the aerodynamic optimization of any aircraft surface.

One promising implementation of the design tool would be as an initiator for a multidisciplinary optimization (MDO) algorithm. At the faculty of Aerospace Engineering at Delft University of Technology, a distributed design system called the design and engineering engine (DEE) has been under development. The DEE consists of a multidisciplinary collection of design and analysis tools able to automatically interface and exchange data and information. One of the components of this DEE is the initiator, as shown in Fig. 1. The main function of this initiator is to provide a feasible initial solution to facilitate the MDO process [1]. For aircraft parts where aerodynamic considerations drive the design, an aerodynamic optimization algorithm could fulfill this function.

To numerically optimize any shape, it is necessary to express it into a finite number of variables, preferably as little as possible to minimize computation cost during optimization. This process is called shape parameterization or geometric representation and is an essential part of any shape optimization problem. Kulfan [2] composed a list of desirable features that any geometric representation technique should possess, including features such as smoothness, mathematical efficiency, appropriate design space, and ability to handle constraints. It was then observed that most conventional

Presented as Paper 2010-3004 at the 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Orlando, FL, 12–15 April 2010; received 25 June 2010; revision received 2 October 2010; accepted for publication 11 December 2010.

[*] Ph.D. Researcher, Faculty of Aerospace Engineering, Kluyverweg 1. Young Professional Member AIAA.

[†] Professor, Faculty of Aerospace Engineering, Kluyverweg 1. Senior Member AIAA.

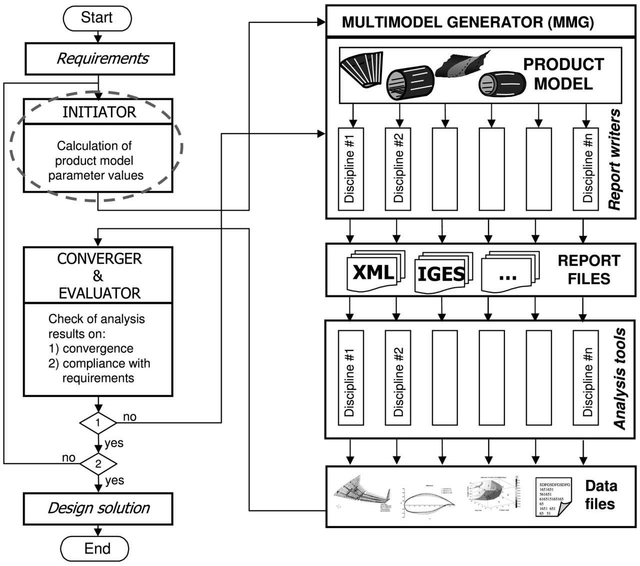


Fig. 1 Position of initiator within the DEE framework.

methods fail to meet all of these criteria, and a novel technique was presented that did fulfill the complete set of requirements. Kulfan [3] later extended the method to more complex and three-dimensional shapes. This class-shape-transformation (CST) technique combined a class function, representing a specific class of shapes, and a shape function, which defined the deviation from the class function. The shape function was based on Bernstein polynomials, and their coefficients formed the design variables. An alternative parameterization method for use in the conceptual design phase was presented by Rodriguez and Sturdza [4]. Samareh [5] evaluated a number of shape parameterization techniques and concluded that polynomial and spline representations are the best approach in terms of 1) ability to handle surfaces, 2) consistency across disciplines, 3) ability to handle large geometry changes, and 4) availability of sensitivity derivatives

To allow for local modifications, Straathof et al. [6] expanded the CST method with a refinement function based on B-splines. By using B-splines, it became possible to locally modify the shape of a curve without having to increase the order of the Bernstein polynomials of the shape function. This local control also provided a larger range of possibilities for handling geometric constraints. A number of different techniques for implementing constraint handling was investigated in literature. Juhász and Hoffmann [7] made use of the knot values of the B-spline to create a virtual envelope that the shape could not cross. A similar technique was presented by Berglund et al. [8]. These techniques are elegant. However, they only work for external constraints, whereas aircraft constraints are usually internal (e.g., fuel tanks, passenger cabin, and baggage volume). Pourazady and Xu [9] ascribed physical properties, such as stretching stiffness and bending stiffness, to the B-spline curves. Satisfying the constraints was accomplished by minimizing the deformation energy of the curves. This technique was shown to be especially useful in design cases where the physical properties actually played a role. Painchaud-Ouellet et al. [10] used a constraint on the maximum thickness to guarantee a certain internal volume. However, the latter two methods did not allow the geometric constraints to be translated to bounds on the design variables. This is desirable, because it limits the required computation time.

Also significantly influencing the computation time is the choice of aerodynamic solver, which is usually made based on the required complexity of the computed flow. Reynolds-averaged Navier–Stokes solvers use a time average of the full Navier–Stokes equations and provide the most accurate description of a flowfield that can feasibly be used for complete aircraft design applications. Neglecting viscosity leads to the Euler equations. Not taking into account viscosity means that the only drag sources that Euler solvers can compute are induced drag and wave (pressure) drag, unless they are coupled to boundary-layer equations. Adding the assumption of irrotational flow leads to the full potential equations, and linearizing these finally results in the linearized potential equations. For this paper, use was made of the commercial panel method program VSAERO [11], which employs the linearized potential equations coupled to boundary-layer equations, as well as an in-house Euler solver [12].

The most cost efficient way of optimizing a certain objective function is by making use of its gradients with respect to the design variables. Usually, such methods, like sequential quadratic programming (SQP), require more function evaluations as the number of design variables increases. There are, however, exceptions to this rule, like the adjoint equation method [13]. Coupling this method to an aerodynamic analysis means that only the derivatives of the objective function with respect to the flow variables are necessary; hence, the number of function evaluations required during the optimization process is independent of the number of design variables. This is very promising. However, a disadvantage of a gradient-based optimization method is that it is only capable of finding local optima. These can be local optima in the actual design space, but in the case of finite difference gradients, they can also be caused by noise in the gradient values. Finally, gradient-based algorithms usually cannot handle discrete variables.

To overcome these problems, a global optimization method could be employed. A disadvantage of such a method, however, is that it usually requires a high number of function evaluations. Since, in aerodynamic optimization, function evaluations are often expensive in terms of computation time, this is highly undesirable. Another problem of global optimization methods is that they cannot guarantee that a global optimum has been found. A solution to the former drawback could be the use of response surfaces [14]. A response surface is basically a surrogate model of the design space created using the information from a limited number of sample points. Evaluating a design point using this surrogate model is much less expensive than performing the actual aerodynamic analysis and, therefore, could make global optimization feasible.

Choosing the right set of sample points and generating an accurate response surface can be done in many ways. Techniques for achieving this are widely described in literature: e.g., by Forrester et al. [14,15] and Jones et al. [16]. There is also a wide choice in global optimization algorithms. For example, two promising techniques are genetic algorithms and particle swarm optimization (PSO); see, for example, Green and Cheng [17] and Venter and Sobieszczanski-Sobieski [18].

In the present investigation, a sample set was created using the Latin hypercube sampling process [14]. The Kriging method was used to generate the response surface [16]. SQP and PSO were chosen as optimization algorithms. The algorithms were used separately and not in hybrid mode.

A description of the techniques used for parameterization, aerodynamic analysis, and optimization is given in Sec. II. Section III presents the definition and results of a first test case that was performed to investigate handling volume constraints in subsonic conditions using B-splines. The definition and results of a second test case can be found in Sec. IV. This test case demonstrates how to decrease wave drag in transonic conditions, making use of a B-spline refinement technique. The conclusions and recommendations can be found in Sec. V.

II. Techniques Employed for Parameterization, Analysis, and Optimization

This section describes the techniques employed for parameterization, analysis, and optimization. An explanation of the employed parameterization technique and a novel way of implementing volume constraints using B-splines are presented in Sec. II.A. Section II.B gives an overview of the aerodynamic analysis tools used and their coupling to the parameterization method. A description of the optimization routines used is given in Sec. II.C.

A. Parameterization of Wing Shape

1. *Class-Shape-Transformation Method*

The CST method, as developed by Kulfan [3], combines an analytical function (the class function) with a parametric curve (the shape function), and it is defined as follows in two dimensions:

$$\zeta(\psi) = C_{N2}^{N1}(\psi) \cdot S(\psi) \tag{1}$$

with $\zeta = z/c$, $\psi = x/c$, and c as the chord length of the wing. $C_{N2}^{N1}(\psi)$ and $S(\psi)$ represent the class and shape functions, respectively. The class function is defined as

$$C_{N2}^{N1}(\psi) = (\psi)^{N1}(1-\psi)^{N2} \cdot [a(1-\psi) + b\psi] \tag{2}$$

The last term of Eq. (2) was not included in Kulfan's [3] definition of the class function. It has been added here as part of the volume constraint handling method presented in Sec. II.A.2. If $N1$ and $N2$ are chosen equal to 0.5 and 1.0, respectively, then the class function describes a basic airfoil shape. The shape function then represents the deviation from this basic airfoil. Kulfan used Bernstein polynomials as a shape function. The advantage of this is that any choice of Bernstein coefficients leads to a realistic, smooth airfoil geometry. A limitation, however, is that any change in the coefficient vector leads to a global change in shape, therefore not allowing for local modifications. The use of B-splines remedies this problem.

A B-spline consists of multiple (lower order) Bézier curves and is defined as follows:

$$\mathbf{p}\,(u) = \sum_{i=0}^{n} \mathbf{p}_i N_{i,k}(u) \tag{3}$$

The basis functions $N_{i,k}(u)$ are defined iteratively as

$$N_{i,1}(u) = \begin{cases} 1, & \text{if } t_i \leq u < t_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

and

$$N_{i,k}(u) = \frac{(u - t_i)N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u)N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}} \tag{5}$$

where t_i are the knot values that relate the parametric variable u to the control points $\mathbf{p}_i$. They are defined as

$$t_i = \begin{cases} 0, & \text{if } i < k \\ i - k + 1, & \text{if } k \leq i \leq n \\ n - k + 2, & \text{if } i > n \end{cases} \tag{6}$$

In the extension to the CST method, the B-spline is called the refinement function $R(\psi)$. The method is now referred to as the class-shape-refinement-transformation (CSRT) method and can be defined as follows:

$$\zeta(\psi) = C_{N2}^{N1}(\psi) \cdot S(\psi) \cdot R(\psi) \tag{7}$$

Naturally, adding an extra function to the method requires a higher number of design variables compared with the original CST method, which contributes to the complexity of the design problem. However, it is believed that the benefits outweigh this disadvantage.

One of the main benefits of the CSRT method is the possibility to perform a shape optimization in a two-step approach. In the first general step, the Bernstein coefficients of the shape function are varied. In the second regional step, the B-spline coefficients of the refinement function are varied to refine the shape. This technique is applied in Sec. IV.

For a more thorough explanation of the B-spline expansion, called the CSRT method, the reader is referred to Straathof et al. [6].

2. *Volume Constraint Handling in Two Dimensions*

To demonstrate the principle of volume constraint handling using B-splines, a box of fixed dimensions was selected as a constraint. This box constraint can be seen as an extension to the commonly used thickness constraint. With the development of novel aircraft configurations, such as blended-wing bodies, it is believed that a simple thickness or volume constraints will no longer suffice. A passenger cabin that should fit in a wing requires the constraints to be more flexible.

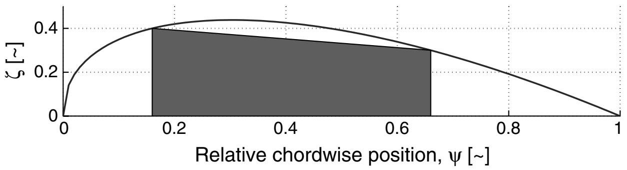


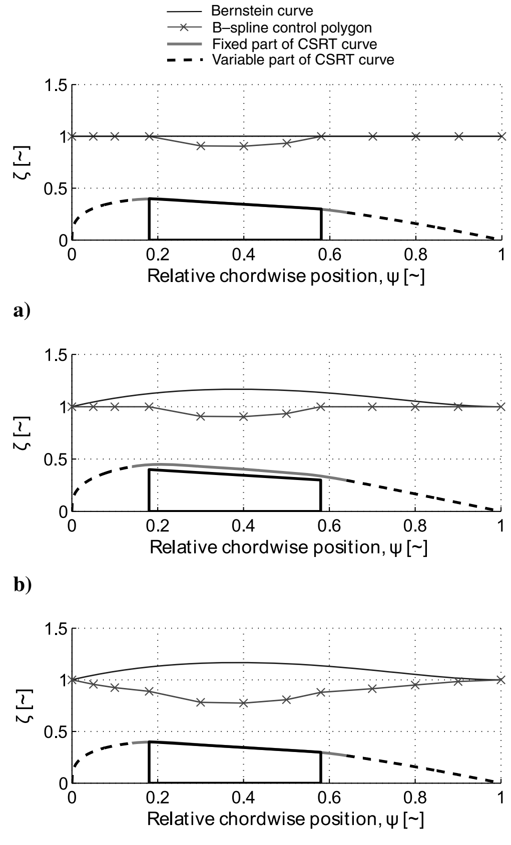
Fig. 2 Making the curve go through the vertices of the box.

The goal of this demonstration is to find a set of rules governing the coefficients of the curves used in the CSRT method that prevent the curve to violate the box constraint during the optimization process. This is accomplished by first finding an airfoil shape that fits tightly around the box constraint and subsequently allowing some of the control points to move only in a specific direction.

The first step in fitting the curve around the box constraint is to find a class function that goes through the vertices of the box. If the coordinates of the two upper vertices are (ψ_1, ζ_1) and (ψ_2, ζ_2), then the following system of equations has to be solved for the scaling coefficients a and b:

$$\begin{cases} (\psi_1)^{N1}(1-\psi_1)^{N2} \cdot [a(1-\psi_1) + b\psi_1] = \zeta_1 \\ (\psi_2)^{N1}(1-\psi_2)^{N2} \cdot [a(1-\psi_2) + b\psi_2] = \zeta_2 \end{cases} \tag{8}$$

This leads to the curve shown in Fig. 2. Alternatively, the class function could be kept constant and the box constraint could be translated along the x axis to a point where Eq. (8) is satisfied (with $a = b = 1$). Generally, however, the position of the box will be predefined.



a)

b)

c)

Fig. 3 Fitting the curve around the constraint.

Equation (8) guarantees that the curve has the right shape but not that it goes through the vertices of the box constraint. This is accomplished by scaling the curve vertically, resulting in Fig. 2.

As can be seen in Fig. 2, there is still some vertical space between the class function and the box constraint. This can be solved by adjusting the control points of the refinement function B-spline such that the gap between the curve and the box is minimized. Any gradient-based optimization algorithm can be used to accomplish this. The result is shown in Fig. 3a.

The straight solid line represents the shape function Bernstein polynomial. It is straight because all Bernstein coefficients are equal to one. The crossed line represents the control polygon of the refinement function B-spline. The region of the curve influenced by the fitting process is mathematically defined. This is the solid part of the curve. The curve as depicted in Fig. 3a essentially forms a minimum shape for the optimization process. The question now is how the coefficients of the shape and refinement functions should be allowed to vary without violating the box constraint. The class function is fixed during the optimization.

Changing the coefficients of the shape function has a global effect on the shape of the curve. This means that decreasing the values of the coefficients without changing the refinement function will always lead to a violation of the box constraint. For this reason, the Bernstein coefficients are only allowed to increase. The situation for the refinement function is slightly more complicated. If the shape function would not be changed, then the B-spline coefficients governing the part of the curve above the box constraint would only be allowed to increase as well. However, if some of the Bernstein coefficients have been changed, then a situation can occur where some of the B-spline coefficients can be decreased without violating the constraint. Such a situation is shown in Fig. 3b.

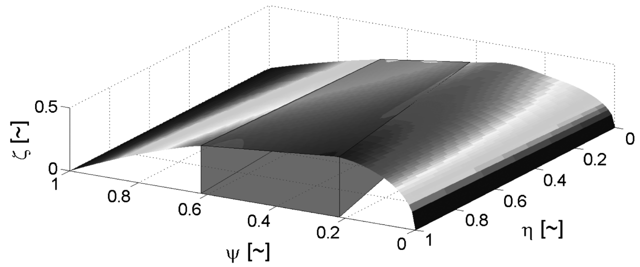


Fig. 4 Fitting the three-dimensional wing around the constraint.

The question now is what set of constraints can be applied to the B-spline coefficients that allows them to be decreased without violating the box constraint. One way to do this is to divide the values of the B-spline coefficients by the magnitude of the Bernstein polynomial at their specific locations. The number of B-spline control points is usually limited; therefore, this does not require a large number of function evaluations. Since the shape of the B-spline curve is generally quite smooth, it follows the control polygon closely enough to assume that, in that case, the product of the shape function and the refinement function gives a good approximation of the curve that fits tightly around the box constraint. In any case, the convex hull property of the B-spline guarantees that it will not violate the box constraint. Figure 3c shows that after applying this procedure, the curve, indeed, does not cross the constraint.

A specific characteristic of a B-spline is that its coefficients only influence a specific part of the curve. As a result of this, for the control points that govern the part of the curve that is above the box constraint, the positions in Fig. 3c represent their minimum values. The control points that do not influence the part of the curve above the constraint are free to move up and down. It should be noted, however, that the original CST method allows for the leading-edge nose radius and the trailing-edge boat-tail angle to be expressed analytically in terms of the Bernstein coefficients. To retain this useful property, the B-spline coefficients that control the leading- and trailing-edge regions should not be changed.

3. *Volume Constraint Handling in Three Dimensions*

In order for the volume constraint handling method, as described in Sec. II.A.2, to be of use for real wings, it has to be expanded to three dimensions. This requires two main adaptations compared with the method in two dimensions. First, the shape function now consists of a Bernstein surface. Second, the refinement function is now given by a B-spline surface. The reasoning behind handling a volume constraint is completely analogous to the situation in two dimensions and will thus not be explained here again. Figure 4 shows the three-dimensional equivalent of Fig. 3a: it describes the minimum wing shape as used during the optimization process.

B. Aerodynamic Analysis Codes

To demonstrate the new parameterization technique in an optimization routine, use is made of two different types of flow solvers. For the low-speed regime, the panel method program VSAERO [11] was used. For the transonic flow regime, an in-house Euler code [12] was used.

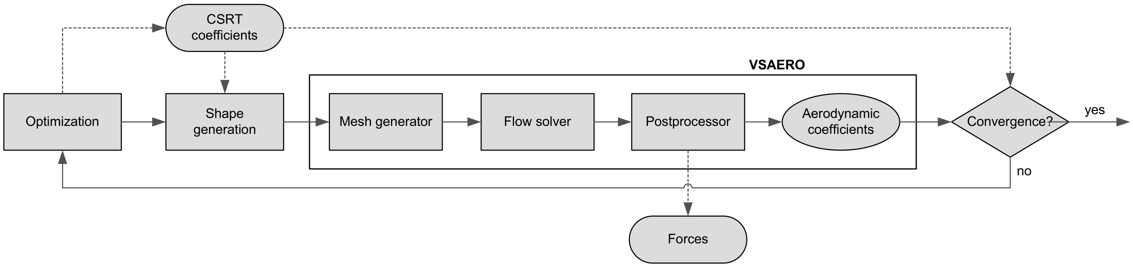


Fig. 5 Schematic of the three-dimensional subsonic optimization.

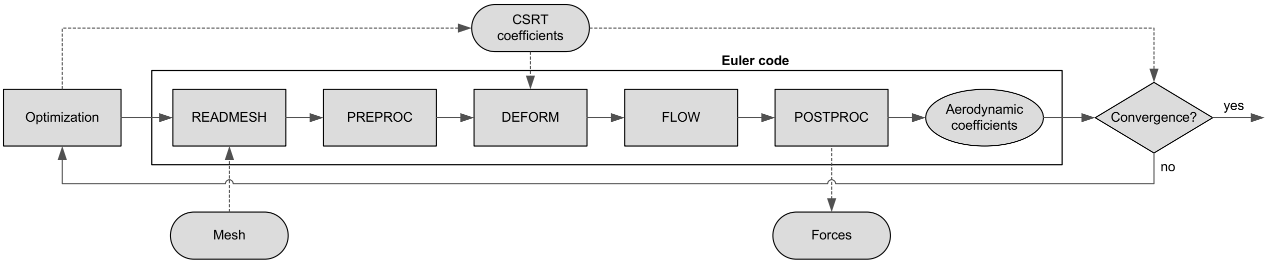


Fig. 6 Schematic of the two-dimensional transonic optimization.

Panel method programs based on Laplace's equation always assume an incompressible, inviscid flow; hence, they are not capable of capturing all aspects of a flowfield. However, they can be improved by correcting for compressibility and by coupling the potential flow equations to a set of boundary-layer equations. For this paper, the Karman–Tsien rule was used for compressibility correction, and transition was computed by using the Granville criterium for transition prediction [11]. As a result of these improvements, the lift, induced drag, and friction drag could be determined with reasonable accuracy [6]. Finally, VSAERO is capable of handling complex three-dimensional shapes relatively fast; thus, it is very suitable for use in any preliminary design application. Figure 5 shows how the CSRT method has been coupled to VSAERO.

Euler solvers also neglect viscosity, but they do include rotational flow. Consequently, they are capable of modeling shock waves, and hence transonic flow. Carpentieri et al. [12] developed an Euler code based on an unstructured finite volume formulation that discretizes the Euler equations on a median-dual mesh. It was shown by Carpentieri et al. that this solver is capable of accurately predicting wave drag. In three-dimensional cases, induced drag can also be computed. The code is written in Fortran and consists of five main modules, as can be seen in Fig. 6: a mesh reading module (READMESH), a preprocessor (PREPROC), a deformation module (DEFORM), the flow solver (FLOW), and a postprocessor (POSTPROC). An effective way of implementing the CSRT method is to create a mesh of the class function and subsequently deform the mesh using the shape and refinement functions, which were programmed into the deformation module. This method was used to model a NACA0012 airfoil and calculate the flowfield for different values of the angle of attack and Mach number. Table 1 shows the results of this test case compared with the results calculated by Carpentieri et al. [12] and Viviand [19]. Viviand provided a series of reference cases for inviscid flowfield methods.

Table 1 shows that the CSRT model performed similarly (within 5%) to the code as programmed by Carpentieri et al. [12]. The density and shape of the mesh were approximately the same for all models (~12, 500 nodes). From this test case, it could be concluded that the implementation of the CSRT method into the Euler code was successful. Note that the goal of this verification was not to show that the Euler solver itself was valid. A thorough validation of the Euler code has been provided by Carpentieri [20].

C. Optimization Routines

Using the CSRT method to model aircraft surfaces could have an influence on the design space. To investigate this effect, a number of different optimization schemes were tested. Two different optimization algorithms were considered: SQP and PSO [18]. For the subsonic test case presented in Sec. III, SQP was performed using the aerodynamic analysis directly. In addition, a response surface (RS) was created on which both SQP and PSO were employed. For the transonic test case of Sec. IV, only direct SQP was used. The initial solutions were also varied. The SQP method used employs a trust-region reflective algorithm and uses finite differences to compute the Hessian [21]. In the PSO method, the local and global learning factors were chosen equal to 0.5 and 0.3, respectively. The inertia factor was varied between 0.3 and 0.7 during the optimization. A total number of 10 particles was used.

To build the response surface, a sample set of 500 analysis runs was created. To check the accuracy of the surrogate model, 10 design points were randomly generated and evaluated using both VSAERO and the response surface. The result is shown in Fig. 7. Considering the fact that only 500 sample runs were performed to recreate a 12-dimensional design space, the results shown in the figure are very good. The difference between the VSAERO and response surface evaluations is never more than 4%. Additionally, and more important, the gradients and minimum and maximum values of the design points coincide. It seems that the response surface generated for this test case is a realistic model of the design space.

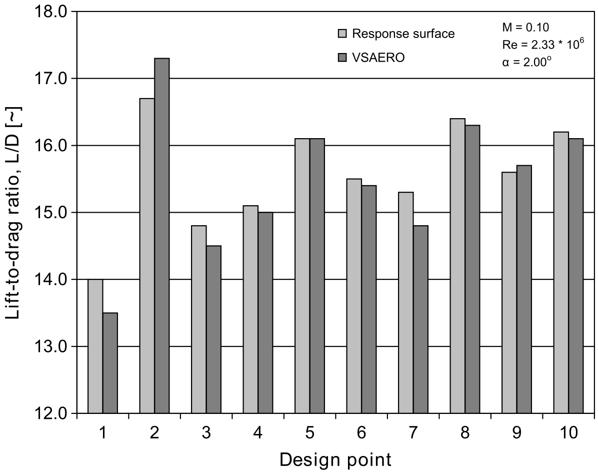


Fig. 7 Test results of the response surfaces.

III. Handling Volume Constraints in Subsonic Conditions Using B-Splines

This first test case focused on handling volume constraints in subsonic conditions using B-splines. The goal of this test case was to show that the three-dimensional CSRT method, including volume constraint handling, could be successfully coupled to a flow solver and to investigate how using the CSRT method influences the optimization problem. The objective function was the lift-to-drag ratio (L/D).

A. Test Case Definition

For this test case, a symmetric wing was used that consisted of an inner and an outer section, both with equal width ($S_{\text{in}} = S_{\text{out}} = 1.0$). The outer section had a taper ratio of 0.50 and a variable sweep angle. Applying sweep is usually done to improve the transonic characteristics of a wing. However, since this was a subsonic test case, this was not necessary. The reason it was decided to vary the sweep was to show that the coupling between parameterization and analysis tools can accommodate such changes in geometry. The inner section had no taper and no sweep. The twist and dihedral of both wing sections were also zero. The wing was described using the three-dimensional CSRT method in which only the coefficients of the shape function were design variables. The box constraint consisted of a prismatic beam with a constant rectangular cross section and was only applied to the inner wing section. It could represent, for example, a fuel tank or other internal volume.

The most accurate procedure would have been to optimize for constant values of the lift coefficient. However, this would have

Table 1 NACA0012 results compared with Carpentieri et al. [12] and Viviand [19]

		c_l			c_d		
M_∞	α, deg	CSRT	Carpentieri et al. [12]	Viviand [19]	CSRT	Carpentieri et al. [12]	Viviand [19]
0.80	1.25	0.356	0.349	0.359	0.023	0.022	0.023
0.85	1.00	0.392	0.375	0.353	0.059	0.058	0.055
0.95	0.00	0.000	0.000	0.000	0.109	0.109	0.108
1.20	0.00	0.000	0.000	0.000	0.096	0.095	0.095
1.20	7.00	0.521	0.524	0.521	0.155	0.154	0.154

required considerably more analysis runs, since the aerodynamic analysis tool could only be run for a specific angle of attack. It was therefore decided to use a constant angle of attack of 2 deg. Analysis showed that the resulting error was small.

As was stated earlier, the wing was described using the three-dimensional CSRT method. However, only one airfoil was used for the entire wing, and this airfoil can be described using the two-dimensional CSRT method. This makes explaining the test case and the results more straightforward. Using the two-dimensional analogy, it can be explained how the shape function was allowed to vary during the optimization. As was explained in Sec. II.A.2, the first step was to find the B-spline coefficients of the refinement function that fitted the airfoil tightly around the box constraint. For this test case, this resulted in the airfoil shown in Fig. 8.

During the optimization, the refinement function was fixed and the shape function was allowed to vary. For both the upper and the lower sides of the airfoil, a fifth-order Bernstein polynomial was used. Kulfan showed that this is sufficient to approximate a typical airfoil [2]. A fifth-order Bernstein polynomial requires six coefficients, so the total number of coefficients governing the airfoil shape would have been 12. However, to keep the nose of the airfoil round, the first coefficients of both Bernstein polynomials were kept equal. This left as design variables 11 Bernstein coefficients and the outer section sweep angle.

Finally, the constraints on the design variables had to be determined. As was explained in Sec. II.A.2, the Bernstein coefficients were not allowed to decrease; hence, their lower bounds were equal to their initial values. Their upper bounds were set to twice their initial values, which translates to a maximum thickness-to-chord ratio of about 16%. The reason the thickness had to be constrained in this way is that an inviscid optimizer has no inherent way of doing this. It was thus fully expected that at least some of the coefficients would reach these bounds. This was not a problem, however, since the goal of this test case was merely to show a correct coupling and the functioning of the parameterization, aerodynamic analysis, and optimizer. Finally, the sweep angle was initialized at 0 deg and was allowed to vary between 0 and 50 deg. For higher sweep angles, the panel mesh would have become too distorted to give accurate results. The definition of the sweep angle Λ can be found in Fig. 9. The taper ratio λ was defined as c_t/c_r.

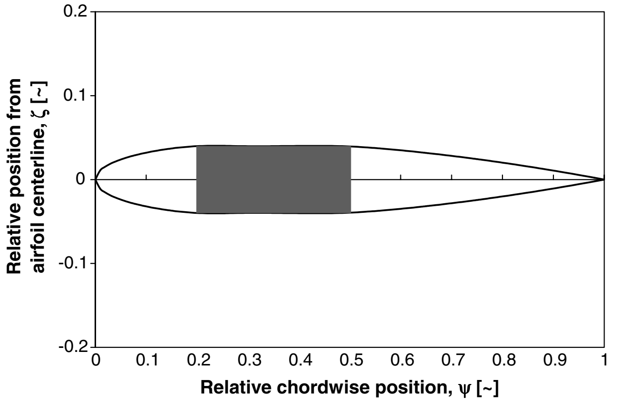


Fig. 8 Three-dimensional test case airfoil fit around the box constraint.

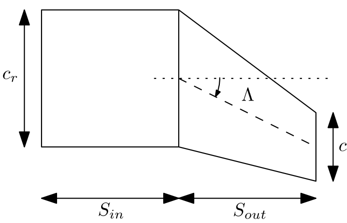


Fig. 9 Definition of the sweep angle Λ.

Table 2 L/D results of various optimization schemes

Test case	C_l	C_d	L/D	Increase in L/D, %	Λ, deg
SQP (direct)	0.726	0.0356	20.4	21	10
SQP on response surface	0.591	0.0299	19.7	17	44
PSO on response surface	0.619	0.0313	19.7	17	44

Table 3 L/D results of various optimization schemes for constant lift coefficient

Test case	α, deg	C_l	C_d	L/D	Increase in L/D, %
SQP (direct)	1.5	0.655	0.0327	20.0	13
SQP on response surface	2.5	0.659	0.0323	20.4	15
PSO on response surface	2.3	0.661	0.0328	20.1	14

B. Test Case Results

Table 2 presents the results for three optimization schemes. For the first scheme, a direct SQP optimization was performed using VSAERO to directly calculate the aerodynamic coefficients. For the second and third schemes, a response surface was created and optimizations were performed on this response surface using SQP and PSO, respectively.

Figure 10 presents the shape of the airfoils following from the test cases presented in Table 2. The airfoil shapes are represented by the values of their Bernstein coefficients. The graph clearly shows that the SQP solution got trapped in a local optimum. Some of the variables stayed at or close to 1.0, while the others went to their extreme values. The other two solutions show a thickness peak on the upper aft part of the airfoil. The actual shape of the optimized profiles can be found in Figs. 11a–11c. The initial airfoil is also shown in the plots.

The best profile following from the test cases is shown in Fig. 12. Its most striking feature is the aft location of the point of maximum thickness. This can easily be explained by considering the test conditions as implemented in VSAERO. The program was run at a very low speed ($M = 0.1$) and a Reynolds number of only 2.33 million. For these conditions, the most straightforward way of reducing drag was by postponing the transition point of the boundary layer by moving the region of adverse pressure aft. This is exactly what the optimizer did by moving the point of maximum thickness aft. In fact, comparing the optimized airfoil to the laminar NACA67-110 profile shows that they are very similar. This is shown in Figs. 13 and 14. Both profiles show laminar flow up to about 80% of the airfoil.

It is not clear at this point why two of the results presented in Table 2 show high sweep angles. One would expect an unswept wing to give the best performance in subsonic conditions. However, this does not influence any of the conclusions drawn in this section.

As was mentioned in III.A, the optimization was performed at a constant angle of attack of 2 deg. To see if this approach was justified,

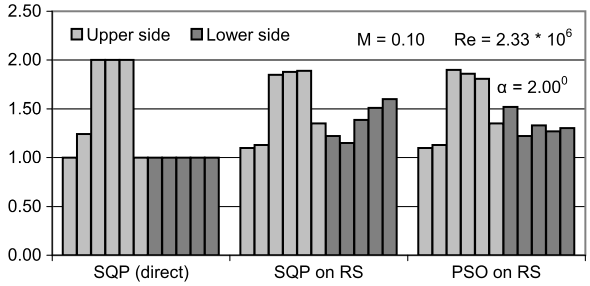


Fig. 10 Optimized airfoils expressed in terms of their Bernstein coefficients.

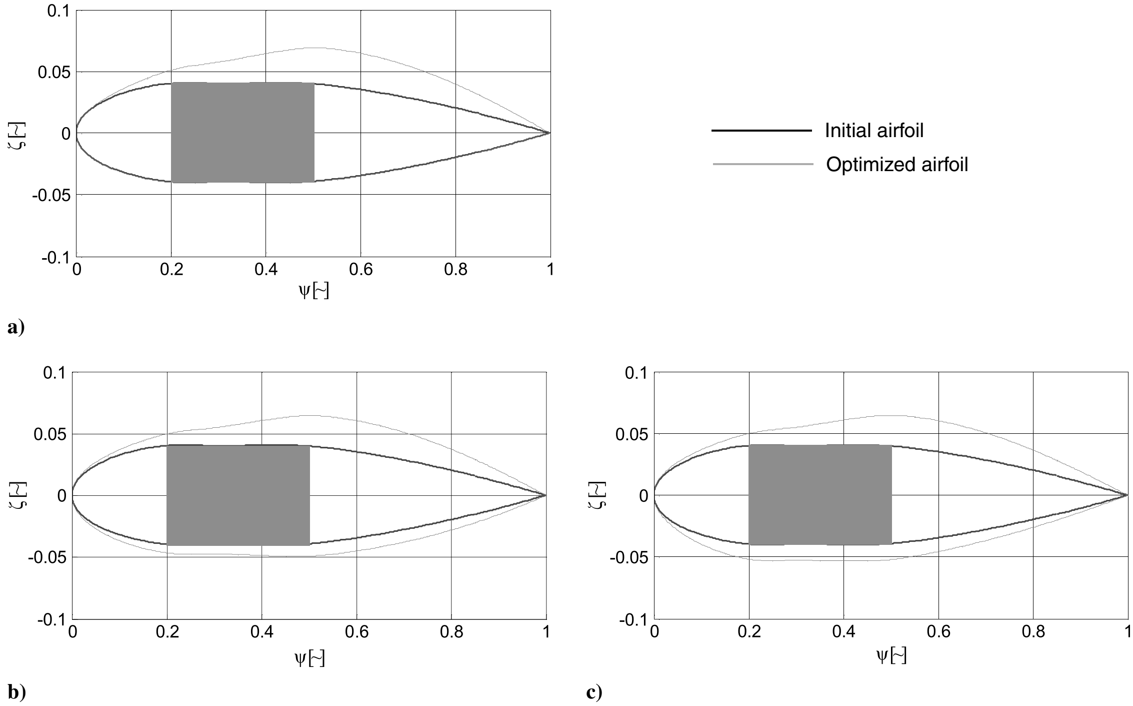


Fig. 11 Optimized airfoils compared with the unit airfoil: a) SQP (direct), b) SQP on RS, and c) PSO on RS.

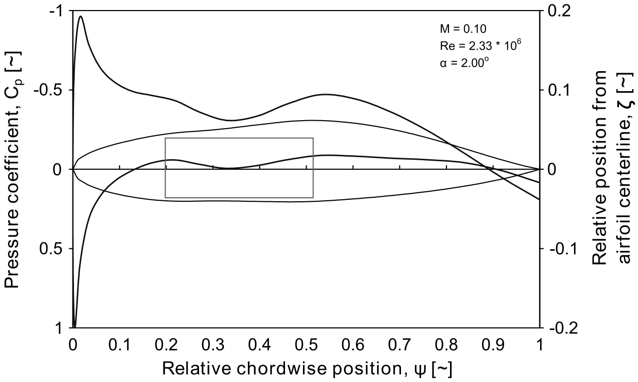


Fig. 12 Solution for the SQP test case, including the box constraint.

the airfoils were also analyzed for a near-constant lift coefficient. The results are shown in Table 3. The table also shows the angle of attack required to reach the given lift coefficients. The last column presents the increase in lift-to-drag ratio with respect to the initial solution at a similar value of the lift coefficient ($C_l = 0.654$).

As was to be expected, the results differ from the ones presented in Table 2. However, the conclusions for this test case remain the same. The three optimization methods perform similarly, and all create a good improvement over the initial solution.

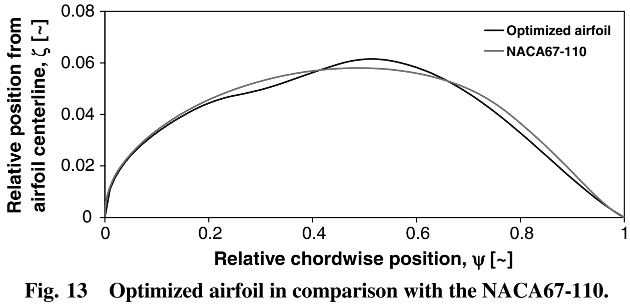


Fig. 13 Optimized airfoil in comparison with the NACA67-110.

IV. Reducing Wave Drag in Transonic Conditions Using a B-Spline Refinement Technique

This second test case modeled an airfoil in transonic conditions using the in-house Euler solver. The goal of this test case was to demonstrate that the solver is capable of modeling shock waves and to demonstrate that using a B-spline refinement leads to a better airfoil without significantly increasing the computation time.

A. Test Case Definition

For this test case, the NACA0012 subsonic airfoil was used as an initial guess. Using a subsonic airfoil for a transonic test case guaranteed that the starting point of the optimization was far away from the optimal solution. This way, it could be shown that the optimizer was capable of finding the global optimum of the design problem.

The airfoil was first optimized for a Mach number of 0.80 and an angle of attack of 1.25 deg. The validation in Sec. II.B showed that the Euler solver performs very well for this flow condition. The optimization was performed in two steps. First, a general optimization was performed using only the Bernstein coefficients of the shape function as design variables. Second, a regional refinement was

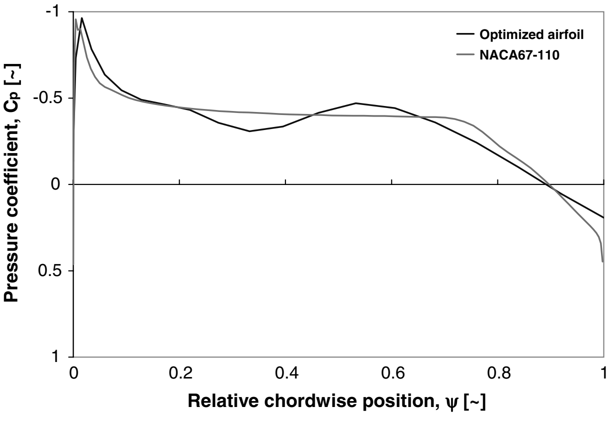


Fig. 14 Pressure distribution of the optimized airfoil in comparison with that of the NACA67-110.

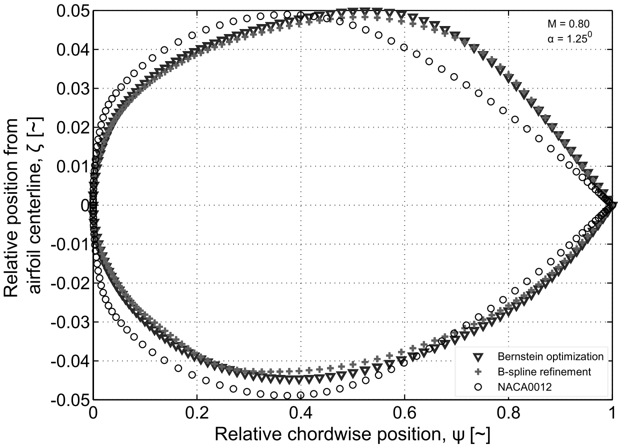


Fig. 15 Airfoils resulting from the two-dimensional test case.

performed using the B-spline coefficients of the refinement function. For both steps, 20 shape variables were used: 10 for the upper and 10 for the lower sides of the airfoil. The bounds on the Bernstein coefficients were set to $[0.1, 0.2]$, which amounts to about $\pm 35\%$ compared with the Bernstein coefficients that describe the NACA0012 airfoil. The B-spline coefficients were allowed to vary between 0.8 and 1.2. Since the B-splines were only meant to locally refine the airfoil shape, these bounds should not be reached.

Table 4 Lift and (wave) drag results for $M = 0.80$ and $\alpha = 1.25°$

	c_l	c_{d_w}	L/D_w
NACA0012	0.356	0.0234	15
Bernstein optimization	0.403	0.0025	161
B-spline refinement	0.422	0.0019	222
B-spline-only optimization	0.292	0.0018	162

After this initial optimization, another batch of runs was performed for Mach numbers ranging from 0.75 to 0.86 and an angle of attack of 1 deg. This was done to investigate the capability of the algorithm to improve the lift-to-drag ratio for different Mach numbers.

B. Test Case Results

Figure 15 shows the airfoils resulting from the general (Bernstein) and regional (B-spline) optimizations, as explained in Sec. IV.A. Both optimizations were performed using SQP. The figure also shows the NACA0012 airfoil that was used as an initial solution. From the figure, it is clear that the Bernstein optimization transformed the subsonic NACA0012 profile into a typical transonic airfoil. This is very well illustrated by the pressure distributions over

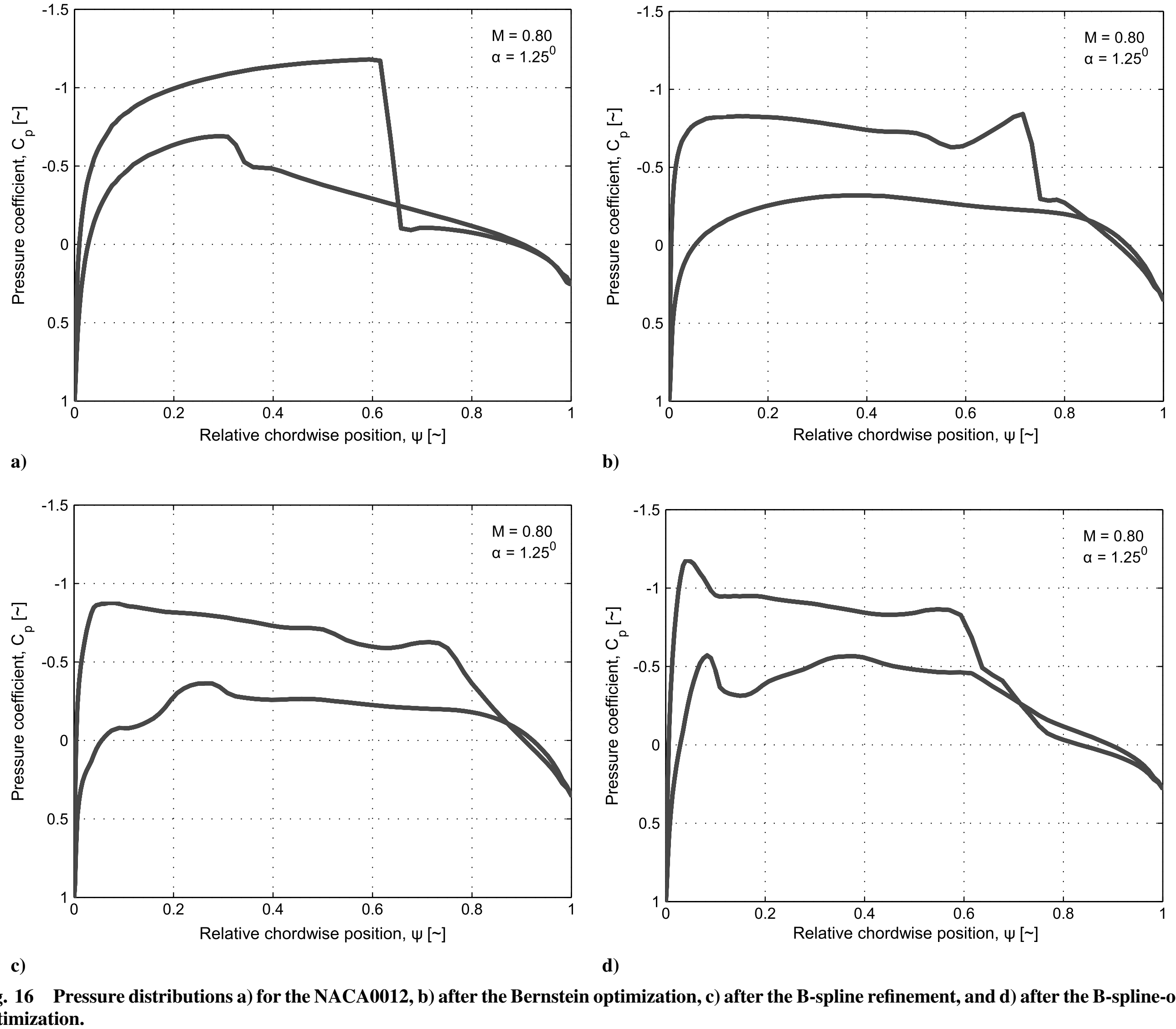


Fig. 16 Pressure distributions a) for the NACA0012, b) after the Bernstein optimization, c) after the B-spline refinement, and d) after the B-spline-only optimization.

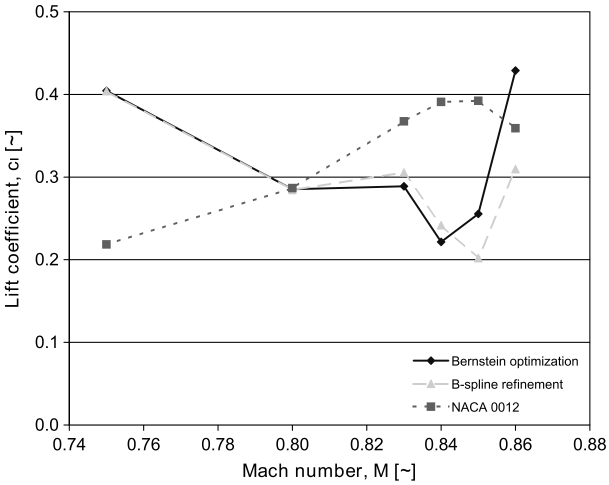


Fig. 17 Lift coefficient versus Mach number.

both profiles, which can be seen in Figs. 16a and 16b. Figure 16a shows a large shock wave on the upper aft part of the airfoil. In Fig. 16b, it can be seen that this shock wave decreased considerably after the first optimization. Figure 16c shows the pressure distribution after the second optimization that employed B-splines. Compared with Figs. 16a and 16b, the shock wave has largely disappeared. In Sec. IV.A, it was mentioned that the B-spline coefficients should not reach their bounds of 0.8 and 1.2. With a minimum coefficient of 0.935 and a maximum coefficient of 1.058, they stayed well within these bounds.

Table 5 Lift-to-drag ratio results for $\alpha = 1.00^\circ$

	L/D_w NACA0012	L/D_w general	Δ, %	L/D_w regional	Δ, %	$\sum \Delta$, %
$M = 0.75$	66.2	274.4	315	275.6	0.4	316
$M = 0.80$	15.6	205.7	1220	214.3	4	1275
$M = 0.83$	9.2	144.7	1475	189.5	31	1963
$M = 0.84$	7.9	76.6	870	140.3	83	1677
$M = 0.85$	6.6	34.3	418	93.3	172	1311
$M = 0.86$	5.2	19.9	282	30.9	55	493

The preceding drawn conclusions can be verified by looking at the values of the wave drag, which are presented in Table 4. After the B-spline refinement, the wave drag decreased by another 25%, which is a significant improvement. The computation times for the Bernstein optimization and the B-spline refinement were approximately equal. As a last test case, the NACA0012 airfoil was optimized using B-splines only. Figure 16d shows the pressure distribution resulting from this optimization. It is clear that the graph is much less smooth than in Figs. 16a–16c. This is likely to have a negative effect on the accuracy of the solver. Looking at Table 4, it can be seen that, even though the resulting wave drag is similar to the B-spline refinement case, the lift is much lower.

Figures 17–19 show the lift coefficient, the drag coefficient, and the lift-to-drag ratio as a function of Mach number for an angle of attack of 1 deg. Each figure shows the results from the general (Bernstein) and regional (B-spline) optimizations; hence, each point on these graphs represents a different airfoil. The NACA0012 results

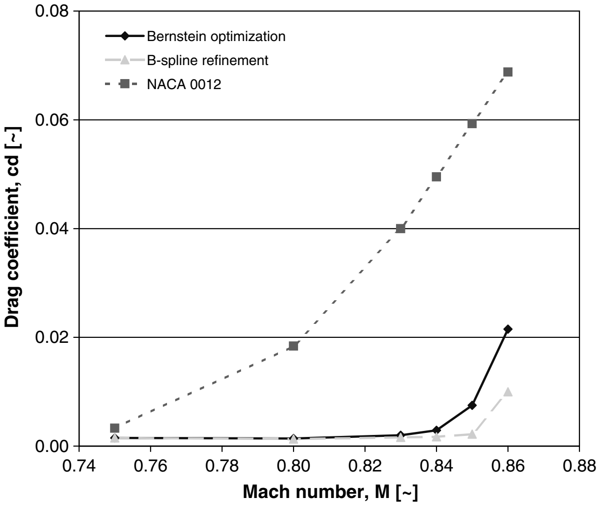


Fig. 18 Drag coefficient versus Mach number.

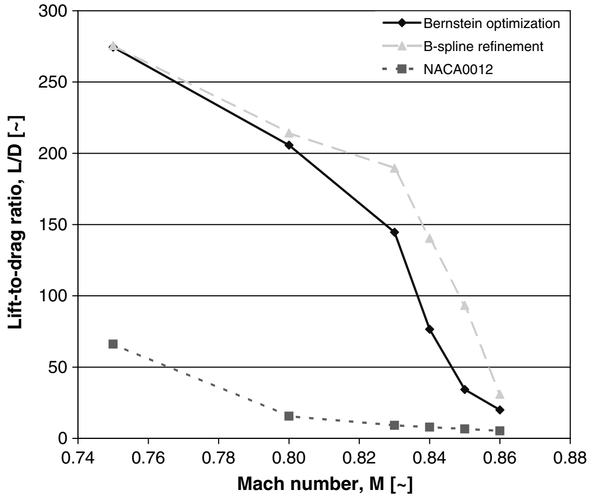


Fig. 19 Lift-to-drag ratio versus Mach number.

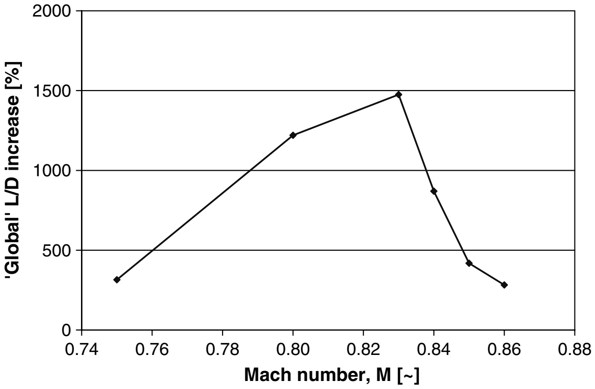


Fig. 20 Relative improvement in lift-to-drag ratio versus Mach number after general optimization.

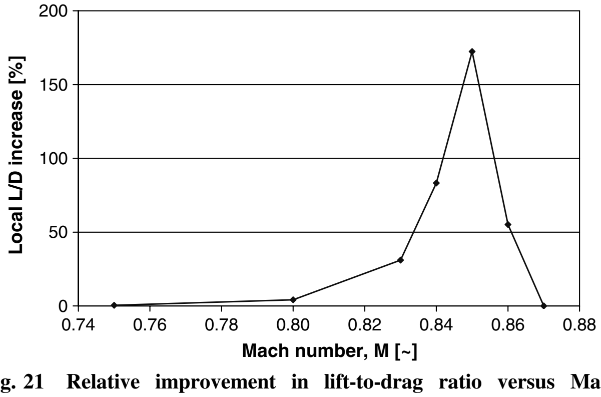


Fig. 21 Relative improvement in lift-to-drag ratio versus Mach number after regional refinement.

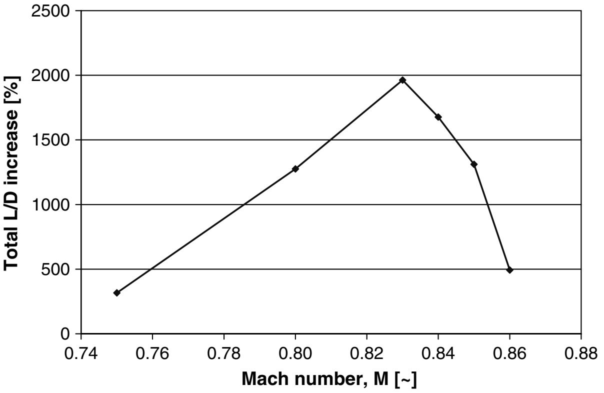


Fig. 22 Total relative improvement in lift-to-drag ratio versus Mach number.

are also included for reference. In Fig. 17, it can be seen that the lift coefficient varies significantly with Mach number. The reason for this is that the objective function of the optimization is the lift-to-drag ratio. Hence, for each case, the optimizer will find the value of the lift coefficient for which this lift-to-drag ratio is minimum. Figure 18 clearly shows the effects of both the general and regional optimizations. Not only do they considerably decrease the magnitude of the drag, they also postpone the drag divergence Mach number from about 0.70 for the NACA0012 to 0.83 after the Bernstein optimization, and to 0.85 after the B-spline refinement. It should be emphasized that, for each case, a different airfoil was considered and, as a result, this comparison is not completely fair. However, the trend is very promising. Finally, Fig. 19 confirms the increase in lift-to-drag ratio after each optimization step. These results have been summarized numerically in Table 5.

Figures 20 and 21 show the performance of each optimization step expressed in terms of the relative increase in lift-to-drag ratio compared with the previous solution. Figure 22 depicts the combined result of both optimizations. As is clear from these figures, there is a certain Mach range for which the optimizations seem to be most effective. The reason for this lies in the nature of the flow solver and the parameterization. Since the only way for the algorithm to reduce the lift-to-drag ratio is to decrease the size and strength of the shock wave, no real gain is expected for Mach numbers lower than about 0.75. As the Mach number increases into the transonic regime ($0.75 < M < 0.85$), both the Bernstein optimization and the B-spline refinement show significant improvements. However, as the Mach number grows even further, a bow shock starts to form on the airfoil and the optimizer is not capable of considerably increasing the performance of the airfoil anymore. This is not a shortcoming of the parameterization method, however. Instead, it demonstrates how the required parameterization depends on the flow regime for which the airfoil or wing is to be optimized.

V. Conclusions

A new parameterization technique was presented based on B-splines. This CSRT method was successfully coupled to the panel method program VSAERO and an in-house Euler solver. It was shown that including B-splines in the parameterization of the aircraft shape provides the ability to handle volume constraints by applying bounds directly on the design variables. This was not possible with previous parameterization methods based on polynomial and spline representations. The volume constraint handling method, as presented in this paper, can be applied equally well in two and three dimensions. Using VSAERO, it was shown that the CSRT method can be used to achieve large portions (up to 80%) of laminar flow on a subsonic wing. The test case also showed that local minima in the design space could pose a problem for the optimization algorithm, possibly requiring the use of non-gradient-based methods. In a second test case, it was demonstrated that B-splines can also be used as an extra refinement in the shape optimization process. Using an in-house Euler solver, it was shown that this refinement can lead to significant L/D improvements on the order of 25 to 100% by eliminating the shock wave and, consequently, improving the drag divergence Mach number significantly. A disadvantage of adding the refinement function is that more function variables are required. However, it is believed that the benefits outweigh this disadvantage.

A recommendation for future work is to include the lift coefficient in the shape optimization, thereby eliminating the need to optimize for constant angle of attack. This would allow a fairer comparison of the optimization results, albeit at the expense of extra computation time. Another step that needs to be taken is to expand the tool to be able to handle not just airfoils and simple wings, but also more complex shapes, such as blended-wing bodies and wing–fuselage combinations. Once this is accomplished, a more extensive and more realistic range of test cases can be performed.

Acknowledgments

The authors would like to thank Roelof Vos and Mark Voskuijl for their valuable feedback. They also owe gratitude to the CleanEra project for providing the necessary facilities to do the research.

References

[1] Schut, E. J., and van Tooren, M. J. L., "A Knowledge Based Engineering Approach to Support Automatic Generation of FE Models in Aircraft Design," 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 2007-0968, 2007.

[2] Kulfan, B. M., "'Fundamental' Parametric Geometry Representation for Aircraft Component Shapes," 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, VA, AIAA Paper 2006-6948, Sept. 2006.

[3] Kulfan, B. M., "Universal Parametric Geometry Representation Method," *Journal of Aircraft*, Vol. 45, No. 1, 2008, pp. 142–158. doi:10.2514/1.29958,

[4] Rodriguez, D. L., and Sturdza, P., "A Rapid Geometry Engine for Preliminary Aircraft Design," 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 2006-0929, Jan. 2006.

[5] Samareh, J. A., "Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization," *AIAA Journal*, Vol. 39, No. 5, 2001, pp. 877–884. doi:10.2514/2.1391

[6] Straathof, M. H., van Tooren, M. J. L., Voskuijl, M., and Koren, B., "Aerodynamic Shape Parameterisation and Optimisation of Novel Configurations," *Proceedings of the RAeS Aerodynamic Shape Parameterisation and Optimisation of Novel Configurations Conference*, London, Royal Aeronautical Soc., London, Oct. 2008, pp. 1–14.

[7] Juhász, I., and Hoffmann, M., "Constrained Shape Modification of Cubic B-Spline Curves by Means of Knots," *Computer Aided Design*, Vol. 36, No. 5, 2004, pp. 437–455. doi:10.1016/S0010-4485(03)00116-7

[8] Berglund, T., Jonsson, H., and Söderkvist, I., "An Obstacle-Avoiding Minimum Variation B-spline Problem," *Proceedings of the 2003 International Conference on Geometric Modeling and Graphics*, London, IEEE Publ., Piscataway, NJ, July 2003, pp. 156–161.

[9] Pourazady, M., and Xu, X., "Direct Manipulation of B-Spline and NURBS Curves," *Advances in Engineering Software*, Vol. 31, No. 2, 2000, pp. 107–118. doi:10.1016/S0965-9978(99)00026-5

[10] Painchaud-Ouellet, S., Tribes, C., Trépanier, J-Y., and Pelletier, D., "Airfoil Shape Optimization Using a Nonuniform Rational B-Splines Parameterization Under Thickness Constraint," *AIAA Journal*, Vol. 44, No. 10, 2006, pp. 2170–2178. doi:10.2514/1.15117

[11] *VSAERO: A Code for Calculating the Nonlinear Aerodynamic Characteristics of Arbitrary Configurations User's Manual Version 7.1*, Analytical Methods, Redmond, WA, 2005, pp. 2.1–2.44.

[12] Carpentieri, G., Koren, B., and van Tooren, M. J. L., "Development of the Discrete Adjoint for a Three-Dimensional Unstructured Euler Solver," *Journal of Aircraft*, Vol. 45, No. 1, 2008, pp. 237–245. doi:10.2514/1.32871

[13] Carpentieri, G., Koren, B., and van Tooren, M. J. L., "Adjoint-Based Aerodynamic Shape Optimization on Unstructured Meshes," *Journal of Computational Physics*, Vol. 224, No. 1, 2007, pp. 267–287. doi:10.1016/j.jcp.2007.02.011

[14] Forrester, A. I. J., Sóbester, A., and Keane, A. J., *Engineering Design via Surrogate Modelling*, Wiley, West Sussex, England, U.K., 2008,

pp. 1–76.

[15] Forrester, A. I. J., Bressloff, N. W., and Keane, A. J., "Optimization Using Surrogate Models and Partially Converged Computational Fluid Dynamics Solutions," *Proceedings of the Royal Society A*, Vol. 462, No. 2071, 2006, pp. 2177–2204. doi:10.1098/rspa.2006.1679

[16] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, 1998, pp. 455–492. doi:10.1023/A:1008306431147

[17] Green, M., and Cheng, R., *Genetic Algorithms and Engineering Design*, Wiley, New York, 1997, pp. 1–97.

[18] Venter, G., and Sobieszczanski-Sobieski, J., "Particle Swarm Optimization," 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, CO, AIAA Paper 2002-1235, April 2002.

[19] Viviand, H., "Numerical Solutions of Two-Dimensional Reference Test Cases," *Test Cases for Inviscid Flow Field Methods*, AGARD Rept. AR-211, May 1985.

[20] Carpentieri, G., *An Adjoint-Based Shape-Optimization Method for Aerodynamic Design*, Delft Univ. of Technology, Delft, The Netherlands, 2009, pp. 1–148.

[21] *Optimization Toolbox^{TM} User's Guide*, MathWorks, Natick, MA, 2010, pp. 3.29–3.37.

R. Kapania
Associate Editor