51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference<BR> 18th
12 - 15 April 2010, Orlando, Florida

AIAA 2010-3004

Development and Implementation of a Novel Parametrization Technique for Multidisciplinary Design Initialization

Michiel H. Straathof[*] , Michel J.L. van Tooren [†] , M. Voskuijl [‡] and R. Vos [§]

Delft University of Technology, Delft, 2600 AA, Netherlands

A new parametrization method for aircraft shapes is presented to enhance shape optimization for aircraft design. This parametrization method was implemented in a tool that creates feasible initial solutions for multidisciplinary design optimization problems. The tool combines all aspects of the aerodynamic design process: parametrization, aerodynamic analysis and optimization. The novel parametrization method presented in this paper makes use of the Class-Shape-Refinement-Transformation (CSRT) method. This method employs a combination of Bernstein polynomials and B-splines to allow for both global and local control of the shape. Additionally, the use of B-splines makes it possible to efficiently handle volume constraints, which are very common in aircraft design. The parametrization method was coupled to two different aerodynamic analysis tools. The commercial panel method code VSAERO was used for the low-speed regime and an in-house Euler code was used for transonic and supersonic flight conditions. Various different optimization schemes were investigated and compared. A number of test cases were performed. For the first set of test cases, a three-dimensional geometry was optimized for subsonic conditions, using VSAERO and various optimization algorithms. For the second set of test cases, an airfoil was optimized for transonic and supersonic conditions, using the in-house Euler solver and a gradient-based optimizer. From this work it can be concluded that a combination of stochastic and gradient-based optimization algorithms works best together with the CSRT method. Additionally, refining the shape using B-splines proved to be an efficient way of increasing the design freedom, while the design space remained smooth enough to employ gradient-based optimization.

Nomenclature

c	Chord length, [m]
C	Class function, [$\sim$]
c_l	Lift coefficient, [$\sim$]
c_d	Drag coefficient, [$\sim$]
c_r	Root chord, [m]
c_t	Tip chord, [m]
L/D	Lift-to-drag ratio, [$\sim$]
M	Mach number, [$\sim$]
n	Number of B-spline control points + 1, [$\sim$]
N	B-spline basis function, [$\sim$]
$N1$	First exponent of class function, [$\sim$]
$N2$	Second exponent of class function, [$\sim$]
$\mathbf{p}$	Vector of B-spline control points, [$\sim$]
R	Refinement function, [$\sim$]

[*] Ph.D. Researcher, Faculty of Aerospace Engineering, Kluyverweg 1, 2629 HS, Delft, AIAA Young Professional Member.

[†] Professor, Faculty of Aerospace Engineering, Kluyverweg 1, 2629 HS, Delft, AIAA Senior Member.

[‡] Assistant Professor, Faculty of Aerospace Engineering, Kluyverweg 1, 2629 HS, Delft, AIAA Young Professional Member.

[§] Assistant Professor, Faculty of Aerospace Engineering, Kluyverweg 1, 2629 HS, Delft, AIAA Young Professional Member.

S	Shape function, [$\sim$]
S_{in}	Inner section span, [m]
S_{out}	Outer section span, [m]
t	B-spline knot value, [$\sim$]
u	Parametric variable, [$\sim$]
x	Chordwise position, [m]
y	Spanwise position, [m]
z	Distance from airfoil centerline, [m]
α	Angle of attack, [deg]
ζ	Relative chordwise position, [$\sim$]
η	Relative spanwise position, [$\sim$]
λ	Taper ratio, [$\sim$]
Λ	Sweep angle, [deg]
ψ	Relative distance from airfoil centerline, [$\sim$]

I. Introduction

The Faculty of Aerospace Engineering at Delft University of Technology has embarked on a four-year project, called CleanEra, to develop the ‘ultra-eco-friendly airplane’ of the future. This ambitious goal can only be achieved by developing innovative technologies in all fields of aircraft design. Examples are: plasma actuators for aerodynamic drag reduction, reconfigurable flight control systems and adaptive aerostructures.

This paper focuses on the development of the aerodynamic segment of an initialization tool that will enable the designer to intuitively create a feasible aircraft shape that can then be used as initial input for a multi-disciplinary design optimization (MDO). Its three main components are shape parametrization, aerodynamic analysis and optimization. Figure 1 shows the position of the tool in the MDO framework. The goal of the work presented in this paper is to optimize wing shapes for maximum lift-to-drag ratio subject to specific (volume) constraints, hence it is solely based on aerodynamics. In order to initiate a complete aircraft shape, other disciplines, such as 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.

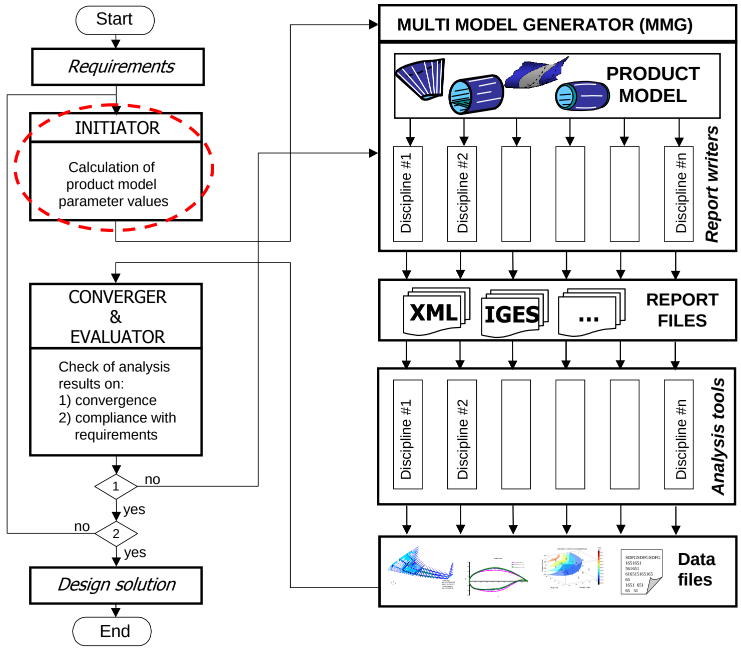


Figure 1. Position of the initialization tool within the MDO framework

In order to 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 parametrization' or 'geometric representation' and is an essential part of any shape optimization problem. Kulfan[1] composed a list of desirable features that any geometric representation technique should possess, including features such as smoothness, mathematical efficiency, size of design space and ability to handle constraints. It was then observed that most conventional methods fail to meet all of these criteria and a novel technique was presented that did fulfill the complete set of requirements. Kulfan[2] 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. Samareh[3] evaluated a number of shape parametrization techniques and concluded that polynomial and spline representations are the best approach in terms of:

- Ability to handle surfaces
- Consistency across disciplines
- Ability to handle large geometry changes
- Availability of sensitivity derivatives

In order to allow for local modifications, Straathof[4] 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 have been investigated in literature. Juhsz and Hoffmann[5] 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.[6] These techniques are elegant, however, they only work for external constraints, whereas aircraft constraints are usually internal (e.g. fuel tanks, passenger cabin, bagage volume). Pourazady and Xu[7] 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.[8] 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.

This choice is usually made based on the required complexity of the computed flow. Reynolds-Averaged-Navier-Stokes (RANS) solvers use a time-average of the full Navier-Stokes equations and provide the most accurate description of a flow field 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 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 called VSAERO,[9] which employs the linearized potential equations, as well as an in-house Euler solver.[10]

The most cost efficient way of optimizing a certain objective function is by making use of its gradient with respect to the design variables. Usually, such methods, like sequential quadratic programming, require more function evaluations as the number of design variables increases. There are however exceptions to this rule, like the adjoint equation method.[11] 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, but a disadvantage of a gradient-based optimization method is that it is only capable of finding local optima. It is essential that the aircraft shape found by the tool is close to the global optimum of the design space, because most MDO algorithms employ gradient-based optimization. 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 is the use of response surfaces.[12] 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 hence this makes 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.[12,13] and Jones et al.[14] There is also a wide choice in global optimization algorithms, two promising techniques are genetic algorithms and particle swarm optimization, see e.g. Green and Cheng[15] and Venter and Sobieszczanski-Sobieski.[16] For this paper, a sample set was created using the Latin hypercube sampling process.[12] The Kriging method was used to generate the response surface.[14] Sequential quadratic programming and particle swarm optimization were chosen as optimization algorithms.

A description of the methodologies used and test cases performed is given in Section II. Section III presents the results of the test cases. The conclusions can be found in Section IV.

II. Description of methodologies and test cases

This section describes the research methodologies used for parametrization, analysis and optimization and gives a definition of two design test cases. An explanation of the employed parametrization technique and a novel way of implementing volume constraints using B-splines are presented in Section II.A. Section II.B gives an overview of the aerodynamic analysis tools used and their coupling to the parametrization method. A description of the optimization routines used is given in Section II.C. Finally, the definition of the test cases that were performed can be found in Section II.D.

II.A. Parametrization of the wing shape

II.A.1. Class-Shape-Transformation method

The CST method, as developed by Kulfan,[2] combines an analytical function (the class function) with a parametric curve (the shape function) and 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 the chord length of the wing. $C_{N2}^{N1}(\psi)$ and $S(\psi)$ represent the Class and shape function respectively. The class function is defined as:

$$C_{N2}^{N1}(\psi) = (\psi)^{N1}(1-\psi)^{N2} \tag{2}$$

If $N1$ and $N2$ are chosen as 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:

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

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:

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

The expanded CST method can now be defined as follows:

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

For a more thorough explanation of this expansion, called the CSRT method, the reader is referred to Straathof.[4]

II.A.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. The goal is to find a set of rules governed by the coefficients of the curves used in the CSRT-method that do not enable the curve to lie within the box. 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. In doing this, the first exponent $N1$ is varied and the second exponent $N2$ is kept constant at 1.0. If the coordinates of the two upper vertices are (ψ_1, ζ_1) and (ψ_2, ζ_2) then first we find the value of $N1$ for which the ratio of the ζ-coordinates at ψ_1 and ψ_2 is equal to ζ_1/ζ_2. This means that the following equation has to be solved for $N1$:

$$\psi_1^{N1}(1 - \psi_1) = \frac{\zeta_1}{\zeta_2}\psi_2^{N1}(1 - \psi_2) \tag{8}$$

leading to the following expression for $N1$:

$$N1 = \frac{\log\left[\frac{\zeta_1(1-\psi_2)}{\zeta_2(1-\psi_1)}\right]}{\log\left(\frac{\psi_1}{\psi_2}\right)} \tag{9}$$

Alternatively, the class function could be kept constant and the box constraint could be translated along the x-axis to a point where Equation (8) is satisfied. Generally however, the position of the box will be predefined for stability reasons.

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 Figure 2.

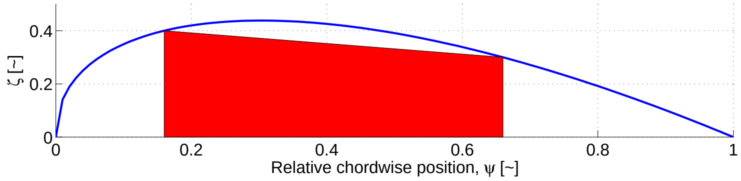


Figure 2. Fitting the box in the curve

As can be seen in Figure 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 Figure 3.

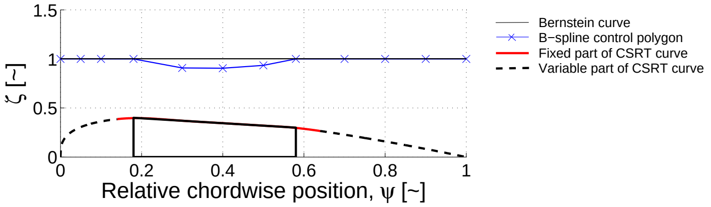


Figure 3. Fitting the curve around the constraint by moving the B-spline control points

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. It is theoretically known which region of the curve is influenced by the fitting process. This is the solid part of the curve. The curve as depicted in Figure 3 forms essentially a minimum shape for the optimization process. The question is now 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 Figure 4.

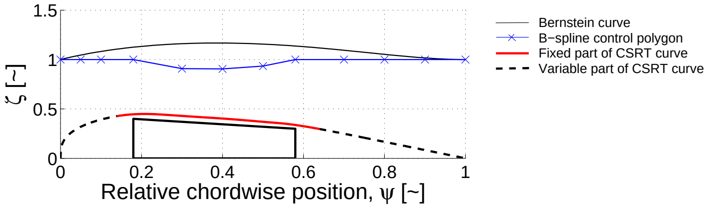


Figure 4. Situation where the B-spline control points could be lowered without violating the constraint

The question now is what set of constraints can be applied to the B-spline coefficients that allow 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 shape function and 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 5 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 Figure 5 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. In order to retain this useful property, the B-spline coefficients that control the leading and trailing edge regions should not be changed.

II.A.3. Volume constraint handling in three dimensions

In order for the volume constraint handling method as described in Section II.A.2 to be of use for real wings, it has to be expanded to three dimensions. This requires three main adaptations compared to the method in two dimensions. First, the class function has to be variable in span wise direction. Second, the shape

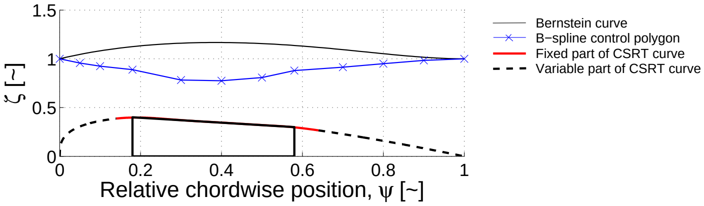


Figure 5. Constructing a new minimum shape

function now consists of a Bernstein surface. Third, 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 6 shows the three-dimensional equivalent of Figure 3: it describes the minimum wing shape as used during the optimization process.

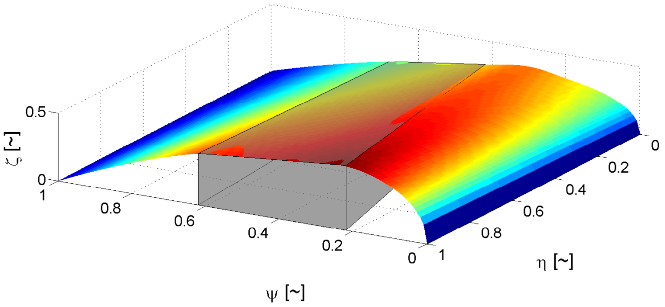


Figure 6. Fitting the three-dimensional wing around the constraint

II.B. Aerodynamic analysis codes

To demonstrate the new parametrization technique in an optimization routine, use was made of two different types of flow solvers. For the low speed regime, the panel method program VSAERO[9] was used. For the transonic and supersonic flow regimes, an in-house Euler code[10] was used.

Panel method programs always assume an incompressible, inviscid flow and hence are not capable of capturing all aspects of a flow field. 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.[9] As a result of these improvements, the lift, induced drag and friction drag could be determined with reasonable accuracy.[4] Finally, VSAERO is capable of handling complex three-dimensional shapes relatively fast and is thus very suitable for use in any preliminary design application. Figure 7 shows how the CSRT method has been coupled to VSAERO.

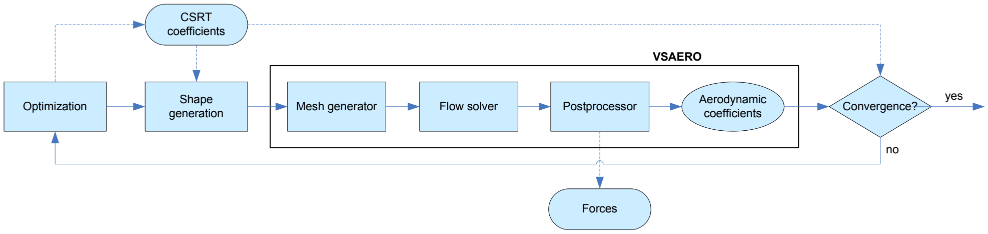


Figure 7. Schematic of the three-dimensional subsonic optimization

Euler solvers also neglect viscosity, but they are capable of modeling shock waves and hence transonic and supersonic flows. Carpentieri et al.[10] 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, also induced drag can be computed. The code is written in Fortran and consists of 5 main modules, as can be seen in Figure 8: a mesh read 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 was 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 flow field for different values of the angle of attack and Mach number. Table 1 shows the results of this test case, compared to the results calculated by Carpentieri et al.[10] and Viviand.[17] Viviand provided a series of reference cases for inviscid flow field methods.

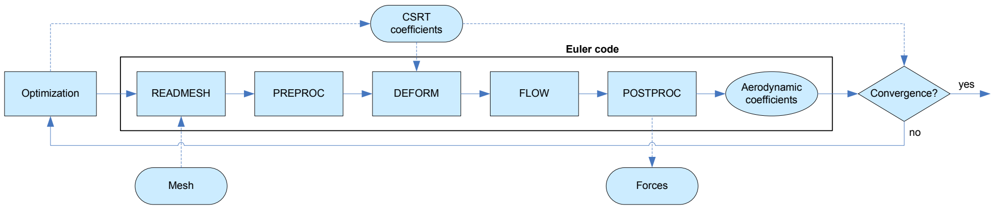


Figure 8. Schematic of the two-dimensional transonic/supersonic optimization

Table 1. NACA0012 results compared to Carpentieri et al.[10] and Viviand[17]

$M_\infty [\sim]$	α [deg]	$c_l [\sim]$			$c_d [\sim]$		
		CSRT	Carpentieri[10]	Viviand[17]	CSRT	Carpentieri[10]	Viviand[17]
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	-	-	-	0.109	0.109	0.108
1.20	0.00	-	-	-	0.096	0.095	0.095
1.20	7.00	0.521	0.524	0.521	0.155	0.154	0.154

The table shows that the CSRT model performed similarly (within 5 %) to the code as programmed by Carpentieri. The density and shape of the mesh were approximately the same for all models ($\sim$ 12500 nodes). From this test case it could be concluded that the implementation of the CSRT method into the Euler code was successful.

II.C. Optimization routines

Using the CSRT method to model aircraft surfaces could have an influence on the design space. To find out which optimization method is most compatible with the new parametrization technique, five different schemes were investigated. They are listed below:

- Sequential quadratic programming (SQP)
- Sequential quadratic programming on a response surface (SQP/RS)
- Particle swarm optimization on a response surface (PSO/RS)
- Sequential quadratic programming using the solution from the second scheme as initial input (SQP/RS → SQP)
- Sequential quadratic programming using the best solution from a sample set as initial input (sample set best → SQP)

In order 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 Figure 9.

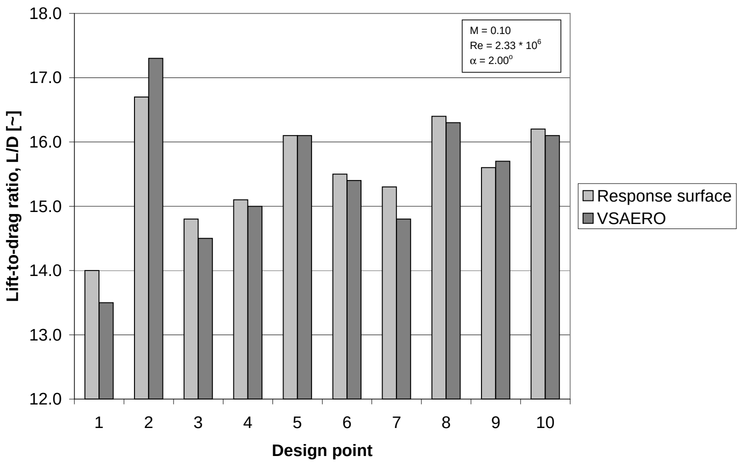


Figure 9. Test results of the response surfaces

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 importantly, the gradients and minimum and maximum values coincide. It can be concluded that the response surface generated for this test case is a realistic model of the design space.

II.D. Test cases

In order to demonstrate the functionality of the design algorithm (parametrization, analysis, optimization), two test cases were performed. The first test case focused on the optimization of a three-dimensional wing in subsonic flow and made use of VSAERO. The goal of this test case was to show that the three-dimensional CSRT method, including volume constraint handling, can be successfully coupled to a flow solver and to find a suitable (combination of) optimization algorithm(s). The second test case modeled an airfoil in transonic and supersonic 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 function would lead to a better airfoil without significantly increasing the computation time. The objective function for both cases was the lift-to-drag ratio (L/D). The next step in this research will be to couple both flow solvers and create one integrated optimization algorithm that can handle all flow regimes.

In Section II.D.1 the definition of the three-dimensional subsonic test case is given. The definition of the two-dimensional transonic/supersonic test case can be found in Section II.D.2.

II.D.1. Definition of the three-dimensional subsonic test case

For this test case a symmetric wing was used that consisted of an inner and an outer section, both with equal width. 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 anyway is to show that the coupling between parametrization and analysis tool 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 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. A possible wing shape is shown in Figure 10.

The most accurate procedure would have been to optimize for constant values of the lift coefficient. However, this would have 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 degrees. This caused an error in the results, but it is believed to be small.

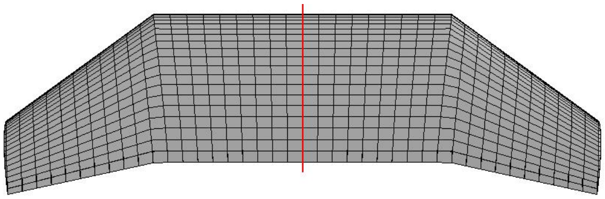


Figure 10. Possible three-dimensional test case wing shape

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 Section 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 Figure 11.

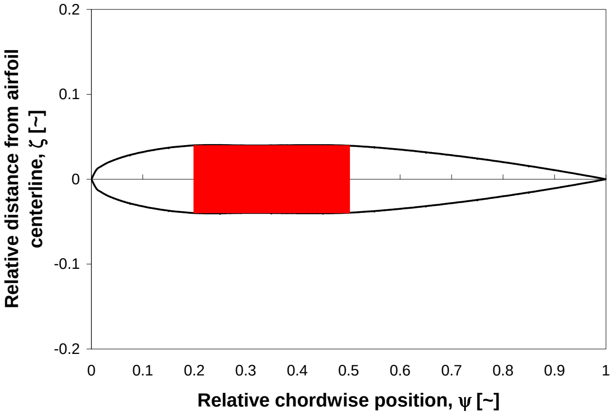


Figure 11. Two-dimensional test case airfoil fit around the box constraint

During the optimization, the refinement function was fixed and the shape function was allowed to vary. For both the upper and the lower side of the airfoil a fifth order Bernstein polynomial was used. Kulfan showed that this is sufficient to approximate a typical airfoil.[1] A fifth order Bernstein polynomial requires six coefficients, so the total number of coefficients governing the airfoil shape would have been twelve. However, to keep the nose of the airfoil round, the first coefficients of both Bernstein polynomials were kept the same. 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 Section 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 functioning of parametrization, aerodynamic analysis and optimizer. Finally, the sweep angle was initialized at 0 degrees and was allowed to vary between 0 and 50 degrees. 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 Figure 12. The taper ratio λ was defined as c_t/c_r.

II.D.2. Definition of the two-dimensional transonic/supersonic test case

For this test case the NACA0012 subsonic airfoil was used as an initial guess. Using a subsonic airfoil for a transonic/supersonic 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.

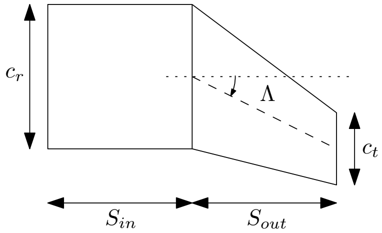


Figure 12. Definition of the sweep angle Λ

The airfoil was optimized for a Mach number of 0.80 and an angle of attack of 1.25 degrees. The validation in Section II.B showed that the Euler solver performs very well for this flow condition. The optimization was performed in two steps. First, a 'global' optimization was performed using only the Bernstein coefficients of the shape function as design variables. Second, a 'local' refinement was 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 side of the airfoil. The bounds on the Bernstein coefficients were set to [0.1,0.2], which amounts to about ± 35 % compared to 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.

III. Results

The results of the three-dimensional subsonic test case can be found in Section III.A. The results of the two-dimensional transonic/supersonic test case are presented in Section III.B.

III.A. Results of the three-dimensional subsonic test case

The results of the optimization runs are presented in Table 2. Also included in the table are the L/D values and sweep angles of the unit airfoil (i.e. with all Bernstein coefficients equal to 1.0) and of a randomly generated airfoil. The best result from the sample set is also incorporated in the table.

Table 2. L/D results of the various optimization schemes

Test case	$L/D[\sim]$	increase in L/D [%]	Λ [deg]
unit airfoil	16.8	0	0
random airfoil	15.6	-7	30
sample set best	19.8	18	44
SQP	20.4	21	10
SQP/RS	19.7	17	44
PSO/RS	19.7	17	44
SQP/RS → SQP	21.8	30	50
sample set best → SQP	21.8	30	50

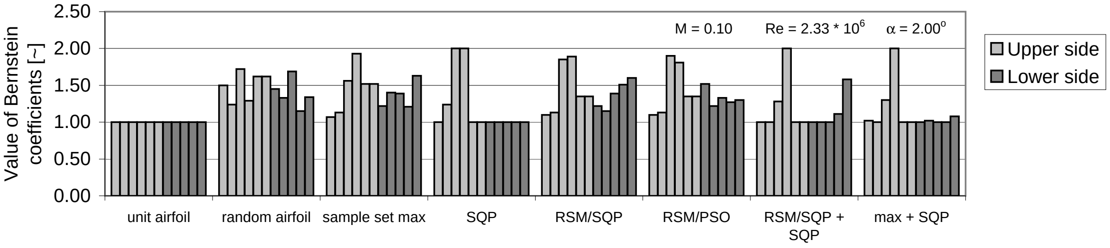


Figure 13. Optimized airfoils expressed in terms of their Bernstein coefficients

Figure 13 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. For the unit airfoil all coefficients are by definition equal to one. It can be seen that the best solution of the sample set shows a thickness peak on the upper aft part of the airfoil. This peak can also be distinguished in all subsequent solutions. The graph also shows clearly 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. All the other optimization schemes converged to similar profiles. The actual shape of the optimized profiles can be found in Figure 14 (a) to (h). The unit airfoil is also shown in all plots.

The best profile following from the test cases is shown in Figure 15. 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 for 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 trying to postpone 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. It should be noted that, especially for these test conditions, the optimized wing is not of much use for any real aircraft. However, it does show that the parametrization and optimization techniques worked properly in combination with an aerodynamic solver.

Finally, to investigate the effect of outer section sweep angle on lift-to-drag ratio, a wing was analyzed for sweep angles varying from 0 to 50 degrees. The profile from Figure 15 was used throughout the wing. The result is presented in Figure 16. From the figure, a clear trend can be distinguished. However, it is straightforward to see how a gradient-based optimizer could get trapped in a local optimum caused by error in the VSAERO output. It is not clear at this point why the lift-to-drag ratio would increase with higher values of Λ.

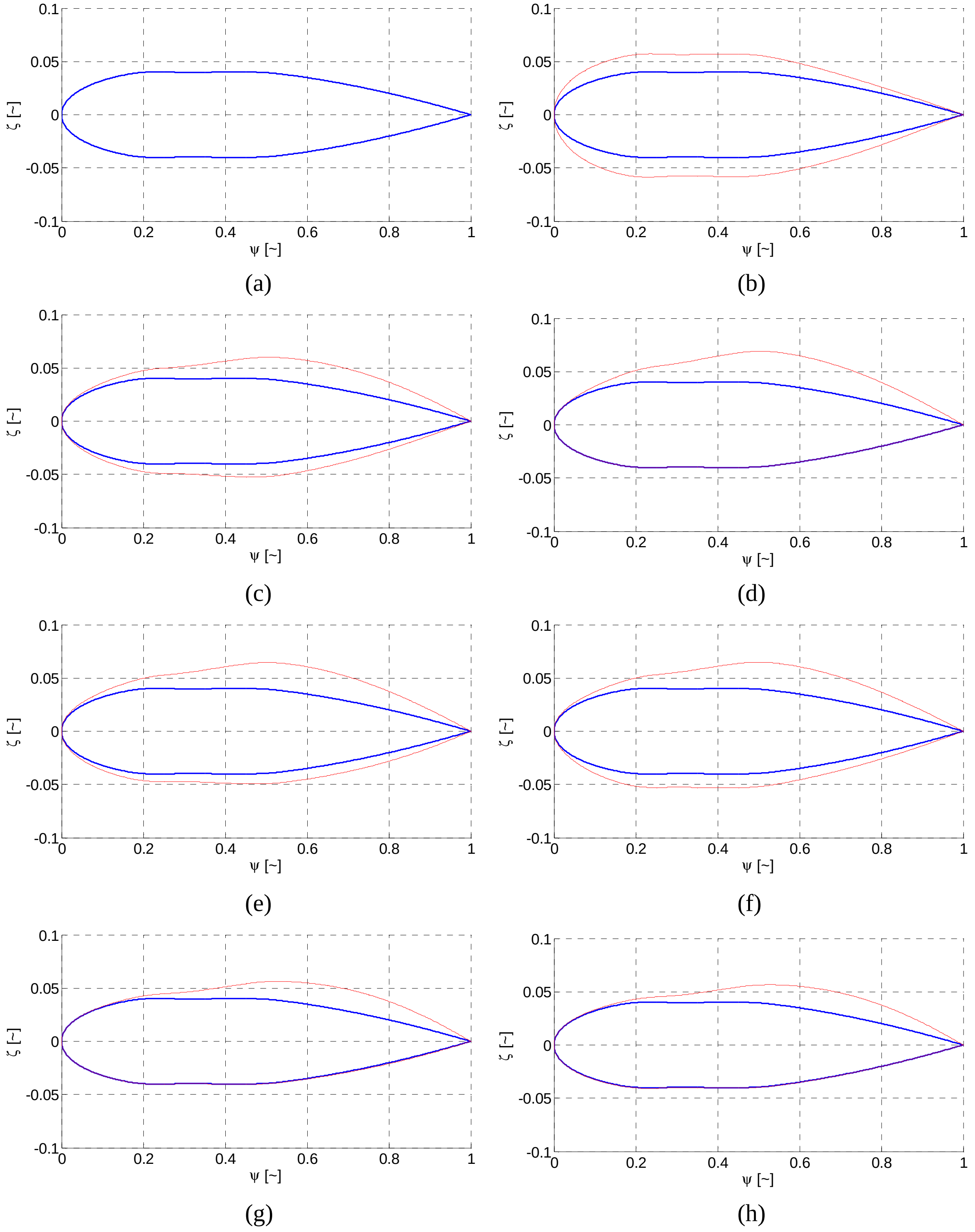
III.B. Results of the two-dimensional transonic/supersonic test case

Figure 17 shows the airfoils resulting from the 'global' (Bernstein) and 'local' (B-spline) optimizations, as explained in Section II.D.2. It 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 both profiles, which can be seen in Figures 18 and 19. Figure 18 shows a large shock wave on the upper aft part of the airfoil. In Figure 19 it can be seen that this shock wave decreased considerably after the first optimization. Figure 20 shows the pressure distribution after the second optimization, that employed B-splines. Compared to Figures 18 and 19, the shock wave has largely disappeared. In II.D.2 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.

The conclusions drawn above can be verified by looking at the values of the wave drag, which are presented in Table 3. 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 21 shows the pressure distribution resulting from this optimization. It is clear that the isobars are much less smooth than in Figures 18, 19 and 20. This is likely to have a negative effect on the accuracy of the solver. Looking at Table 3 it can be seen that even though the resulting wave drag is similar to the B-spline refinement case, the lift is much lower.

Table 3. Lift and (wave) drag results for M = 0.80, $\alpha = 1.25^o$

	$c_l[\sim]$	$c_d[\sim]$	L/D $[\sim]$
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



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 14. Optimized airfoils compared to the unit airfoil, (a) unit airfoil, (b) random airfoil, (c) sample set best, (d) SQP, (e) SQP/RS, (f) PSO/RS, (g) SQP/RS → SQP, (h) sample set best → SQP

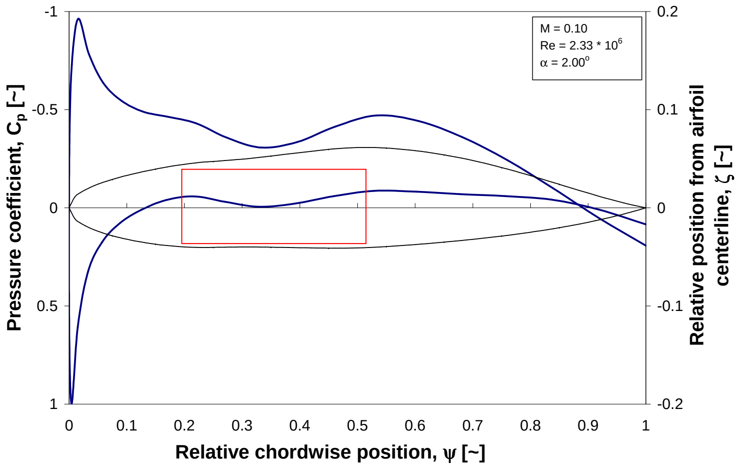


Figure 15. Solution for the RSM/SQP + SQP test case, including the box constraint

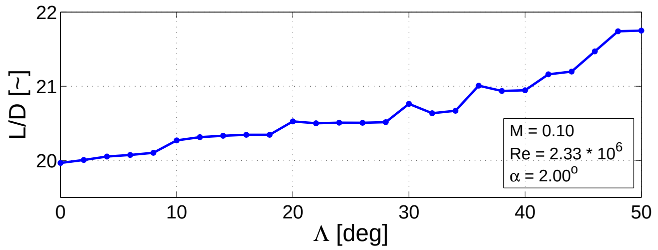


Figure 16. Effect of sweep on lift-to-drag ratio

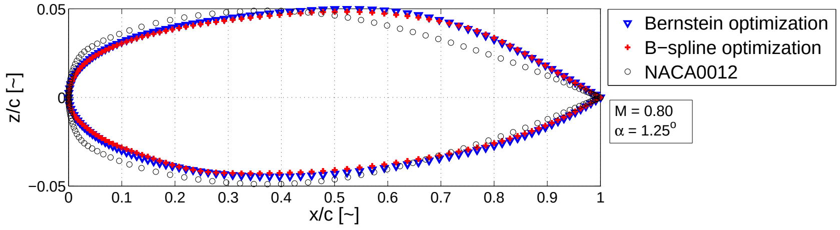


Figure 17. Airfoils resulting from the two-dimensional test case

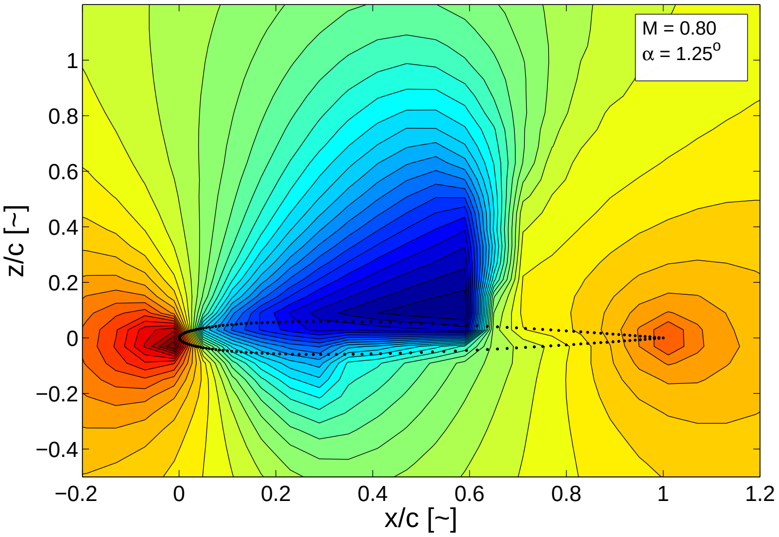


Figure 18. NACA0012 pressure distribution

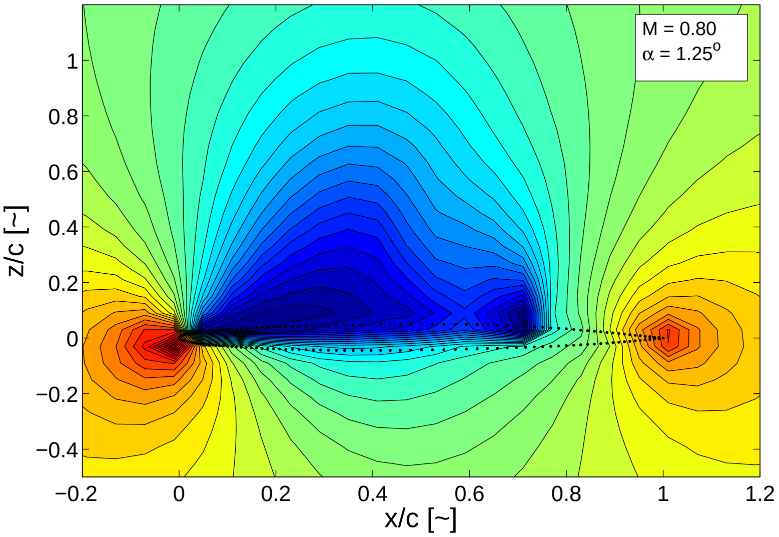


Figure 19. Pressure distribution after the Bernstein optimization

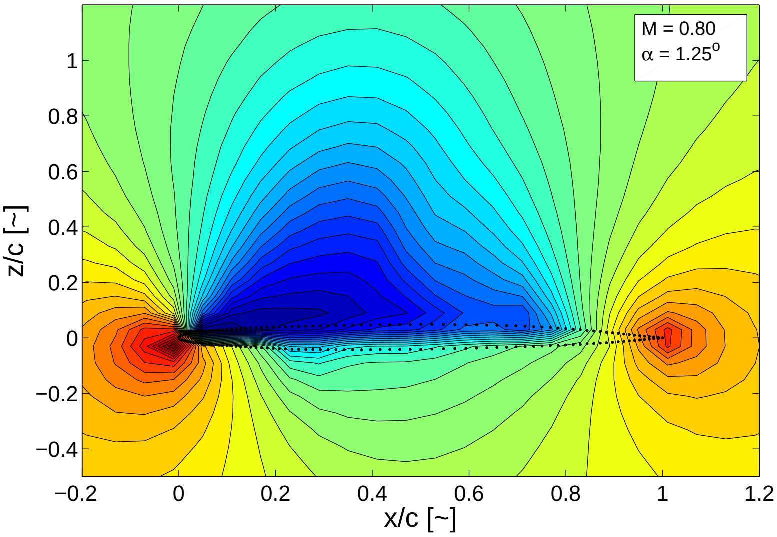


Figure 20. Pressure distribution after the B-spline refinement

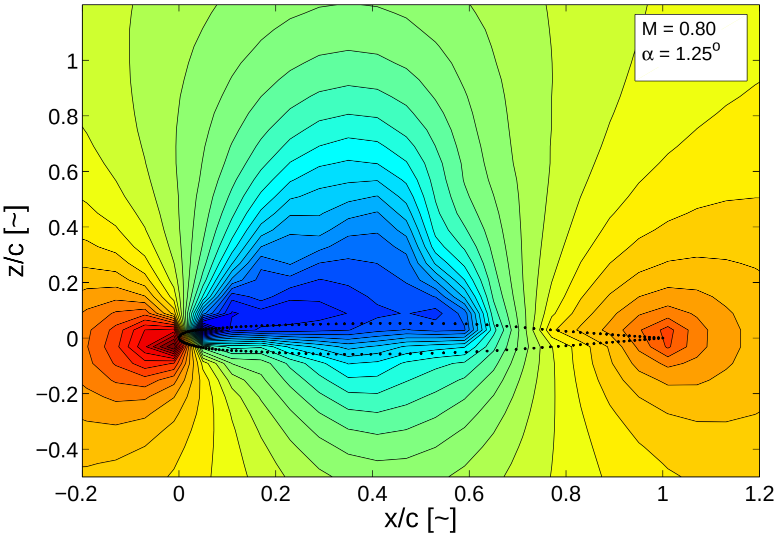


Figure 21. Pressure distribution after the B-spline only optimization

IV. Conclusions

A new parametrization technique was presented based on B-splines. This Class-Shape-Refinement-Transformation (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 parametrization 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 parametrization 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. The B-splines can also be used as an extra refinement in the shape optimization. It was shown that this can lead to significant improvements in the order of 25 %, while keeping the computation time relatively low. A study was also performed on the most suitable optimization algorithm or combination of algorithms for a three-dimensional geometry. It was concluded that using the CSRT method, the combination of a sampling process and a gradient-based optimization is the best approach in terms of required computation time and ability to find the global optimum.

References

[1] Kulfan, B.M., ""Fundamental" Parametric Geometry Representation for Aircraft Component Shapes", *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, Virginia, 6-8 September 2006

[2] Kulfan, B.M., "Universal Parametric Geometry Representation Method", *Journal of Aircraft*, Vol. 45, No. 1, pp. 142-158, 2008

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

[4] Straathof, M.H., Tooren, M.J.L. van, 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, October 2008

[5] Juhász, I. and Hoffmann, M., "Constrained Shape Modification of Cubic B-Spline Curves by Means of Knots", *Computer-Aided Design*, Vol. 36, No. 5, pp. 437-455, 2004

[6] 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, 16-18 July 2003

[7] Pourazady, M. and Xu, X., "Direct Manipulation of B-Spline and NURBS Curves", *Advances in Engineering Software*, Vol. 31, Nr. 2, pp. 107-118, 2000

[8] 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, pp. 2170-2178, 2006

[9] Anonymous, "VSAERO A Code for Calculating the Nonlinear Aerodynamic Characteristics of Arbitrary Configurations User's Manual Version 7.1", *Analytical Methods Inc.*, Redmond, Washington, 2005

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

[11] Carpentieri, G., Koren, B. and Tooren, M.J.L. van, "Adjoint-Based Aerodynamic Shape Optimization on Unstructured Meshes", *Journal of Computational Physics*, Vol. 224, No. 1, pp. 267-287, 2007

[12] Forrester, A.I.J., Sóbester, A. and Keane, A.J., "Engineering Design via Surrogate Modelling", *John Wiley & Sons Ltd.*, West Sussex, 2008

[13] 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, pp. 2177-2204, 2006

[14] 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, pp. 455-492, 1998

[15] Green, M. and Cheng, R., "Genetic Algorithm & Engineering", *John Wiley & Sons, Inc.*, New York, 1997

[16] Venter, G. and Sobieszczanski-Sobieski, J., "Particle Swarm Optimization", *Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Denver, Colorado, 22-25 April 2002

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