Aerodynamic Shape Parameterisation and Optimisation of Novel Configurations

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

Barry Koren§
Leiden University, Leiden, 2333 CA, The Netherlands

The Multi-Disciplinary Design Optimisation (MDO) process can be supported by partial automation of analysis and optimisation steps. Design and Engineering Engines (DEE) are useful concepts to structure this type of automation. Within the DEE, a product can be parameterically defined using Knowledge Based Engineering (KBE). This parameteric product model needs to be initiated before global multidisciplinary optimisation can be performed. This paper presents the first phase in the development of an aerodynamic initiator tool. This tool combines all aspects of the aerodynamic design process, thereby allowing the designer to efficiently determine a feasible aircraft shape that can be used as an initial state for the MDO. This research is performed as part of the CleanEra project at the Faculty of Aerospace Engineering at Delft University of Technology.

The Class-Shape-Transformation (CST) method is used to create a parameteric description of a blended-wing-body (BWB) aircraft. The method is then expanded to allow for more local control of the aircraft shape. The mathematical description of the BWB is then translated to an input for the panel method program VSAERO, which outputs a number of relevant flow characteristics. These are then fed into an optimisation algorithm which generates a new aircraft shape and the process is repeated.

The CST method has proven to be very useful as a parameterisation tool. VSAERO is considered sufficient for the flow analysis for now, but should later be replaced by a Euler or Navier-Stokes code. Work on the optimiser will start later this year.

1 Introduction

Aviation is one of society's great contributors, bringing people and cultures together and creating economic growth across the globe. At the same time, the air transport industry does not ignore the growing concerns about the environment, related to air pollution, noise and contribution to climate change. Although today the contribution of air transport to man-made greenhouse gases is only 2%, it is expected to increase to 3-10% by 2050, [1].

This is however not the only problem. It is obvious that at some point, the oil will be depleted and long before that, the oil prices will have risen enormously. Due to the economic growth and the growing population, air transportation is expected to triple in the next 20 years. Looking back at the

*Ph.D. Researcher, Faculty of Aerospace Engineering, Kluyverweg 1, 2629 HS Delft
†Professor, Faculty of Aerospace Engineering, Kluyverweg 1, 2629 HS Delft
‡Assistant Professor, Faculty of Aerospace Engineering, Kluyverweg 1, 2629 HS Delft
§Professor, Mathematical Institute, Niels Bohrweg 1, 2333 CA Leiden, The Netherlands

development in aircraft history, the aircraft configuration did not change in the last 50 years (since the Boeing 707).

For this reason, 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 plane. A special team of engineers, primarily Ph.D. students, is developing new technologies for a revolutionary conceptual aircraft optimised for environmental and passenger friendliness. The team also actively investigates the possibility of integrating these different technologies.

One of the technologies to be developed is a fully integrated design tool, combining generic shape parameterisation, analysis and optimisation. The first step in developing this tool is to create an aerodynamic initiator for the DEE. The current paper therefore focuses purely on the aerodynamic optimisation, i.e., on finding the outer shape of the aircraft for which the lift-over-drag ratio is optimal, subject to constraints.

Section 2 describes the complete process that is executed by the initiator tool. The subsequent sections then focus on the specific steps in this process. Section 3 describes the parameterisation, section 4 looks into the aerodynamic analysis and section 5 explains the optimisation process. Finally, the conclusions and recommendations can be found in section 6.

2 Process overview

The initiator process consists of three main parts: shape parameterisation, aerodynamic analysis and optimisation. An initial input to this process is required, which can consist of any aircraft shape. In this preliminary state of the research, the shape parameterisation is done using MATLAB, because of easy implementation. For the aerodynamic analysis, the panel method program VSAERO is used. This commercially available software couples integral methods for potential and boundary layer flows to achieve low runtimes and adequate accuracy for preliminary design applications. The mathematical description of the aircraft shape as defined in MATLAB has to be translated to a VSAERO input file. This is accomplished by modifying a software algorithm developed by Frank Dircken from Delft University of Technology. Once the VSAERO input file has been created, the program can be run and an output file containing all the flow analysis results is created. Another MATLAB algorithm interprets these results and forwards them to the optimisation algorithm. Once this optimisation algorithm has converged, the output of the process forms the input to the multi-disciplinary design optimisation. The whole process is depicted in figure 1.

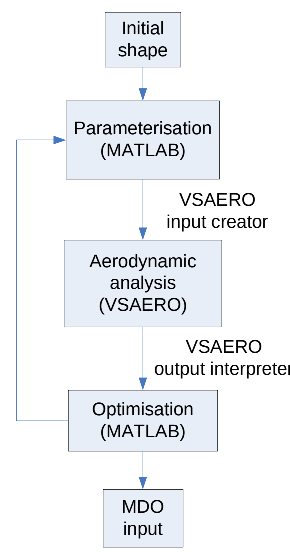

Figure 1: The design process

3 Parameterisation

Airfoil geometry can be modelled in a number of ways. The most straightforward way is to describe the shape using a cloud of points. This however requires a huge amount of design variables in order to guarantee a smooth and accurate airfoil shape. Additionally, modifying the airfoil shape by changing the positions of the individual points is very counter-intuitive and has no real physical meaning. In order to limit the amount of design variables, an airfoil section can also be modeled as a curve. Such a curve can be described as a set of parameteric polynomial equations. One way to increase the complexity of a curve is to increase the order of the polynomials. A type of curve that makes use of this principle is the Bézier curve, as described in section 3.1. Another way is to combine multiple (low-order) curve segments that together form the airfoil shape. B-splines make use of this principle, as explained in section 3.2.

In section 3.3 a geometry representation method called the Class-Shape-Transformation method is briefly explained. It was developed by Brenda Kulfan, [2]. A new addition to this method is presented in section 3.3.2. How to implement the method in three dimensions is briefly explained in section 3.5.

3.1 Bézier curves

A parameteric representation of a quadratic curve is given by:

$$\begin{aligned} x(u) &= a_x u^2 + b_x u + c_x, \\ y(u) &= a_y u^2 + b_y u + c_y, \\ z(u) &= a_z u^2 + b_z u + c_z. \end{aligned} \tag{1}$$

The a-, b- and c-coefficients determine the shape of the curve. Equations (1) can be written in vector form as:

$$\mathbf{p}(u) = \mathbf{a}u^2 + \mathbf{b}u + \mathbf{c}. \tag{2}$$

For a Bézier curve, equation (2) can be expressed as:

$$\mathbf{p}(u) = \sum_{i=0}^{n} \mathbf{p}_i B_{i,n}(u), \qquad u \in [0,1], \tag{3}$$

where the basis functions $B_{i,n}$ are defined as:

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i}, \tag{4}$$

in which n is the number of sides of the control polygon. The number of vertices then equals $n+1$.

3.2 B-splines

Multiple (lower order) Bézier curves can be combined to form a B-spline. For B-splines, equation (3) changes to:

$$\mathbf{p}(u) = \sum_{i=0}^{n} \mathbf{p}_i N_{i,k}(u). \tag{5}$$

The basis functions $N_{i,k}$ are no longer a function of the number of vertices, but instead are dependent on a separate parameter k. This parameter k determines the degree of the basis function polynomial. The basis functions are defined iteratively as:

$$N_{i,1}(u) = \begin{cases} 1, & \text{if} \quad t_i \le u < t_{i+1}, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

and

$$N_{i,k}(u) = \frac{(u-t_i)N_{i,k-1}(u)}{t_{i+k-1}-t_i} + \frac{(t_{i+k}-u)N_{i+1,k-1}(u)}{t_{i+k}-t_{i+1}}, \tag{7}$$

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

$$t_i = \begin{cases} 0, & \text{if} \quad i < k, \\ i-k+1, & \text{if} \quad k \le i \le n, \\ n-k+2, & \text{if} \quad i > n. \end{cases} \tag{8}$$

3.3 Class-Shape-Transformation (CST) method

In [2] Brenda Kulfan presents the Class-Shape-Transformation (or CST) method. In two dimensions, the method combines two special functions, called "Class Functions" and "Shape Functions". The Class Function defines the basic class of general shapes. The Shape Function can be defined in different ways but has to guarantee an analytically smooth geometry. Mathematically the method is defined as:

$$\zeta(\psi) = C_{N2}^{N1}(\psi) \cdot S(\psi), \tag{9}$$

with $\zeta = z/c$, $\psi = x/c$ and c the chord length. $C_{N2}^{N1}(\psi)$ and $S(\psi)$ represent the Class and Shape Function respectively. The Class Function is defined as:

$$C_{N2}^{N1}(\psi) = (\psi)^{N1}(1-\psi)^{N2}. \tag{10}$$

For a NACA type round nose and pointed aft end airfoil the $C_{1.0}^{0.5}(\psi)$ Class Function is used. In other words: $C(\psi) = \sqrt{\psi}(1-\psi)$.

3.3.1 Bernstein polynomials as Shape Function

As was mentioned before, the shape function can be defined in a number of ways. One possibility is to use a Bernstein polynomials representation. In order to do this, the shape function is first written as the product of a coefficient vector A and the Bernstein polynomial terms S_i. For the upper side of the airfoil the formula for the Shape Function then becomes:

$$Su(\psi) = \sum_{i=1}^{n} Au_i \cdot S_i(\psi). \tag{11}$$

The Bernstein polynomial terms are typically defined as:

$$S_i = \binom{n}{i} \psi^i (1-\psi)^{n-i}, \tag{12}$$

where n is the order of the Bernstein polynomial. Note that this definition is identical to that of the Bézier basis functions from section 3.1. Comparing equations (11) and (12) one sees that the order of the polynomial is always equal to the number of control points (or length of the vector A) minus one. For six control points ($n = 5$) the following terms are found:

$$\begin{aligned} S_1 &= (1-\psi)^5, \\ S_2 &= 5(1-\psi)^4\psi, \\ S_3 &= 10(1-\psi)^3\psi^2, \\ S_4 &= 10(1-\psi)^2\psi^3, \\ S_5 &= 5(1-\psi)\psi^4, \\ S_6 &= \psi^5. \end{aligned} \tag{13}$$

Plotting these polynomial terms results in figure 2. A special feature of Bernstein polynomials is that for every value of ψ they add up to one, which is shown by the black line in figure 2. This means that if the coefficients of the vector A are all equal to one, equation (11) will also be equal to one and equation (9) will simplify to:

$$\zeta(\psi) = \sqrt{\psi}(1-\psi). \tag{14}$$

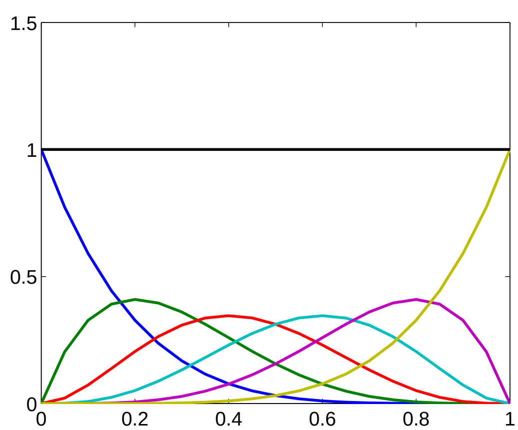

Figure 2: Bernstein polynomial terms for $n = 5$

This equation describes the so-called unit $C_{1.0}^{0.5}$ airfoil, which is plotted in figure 3.

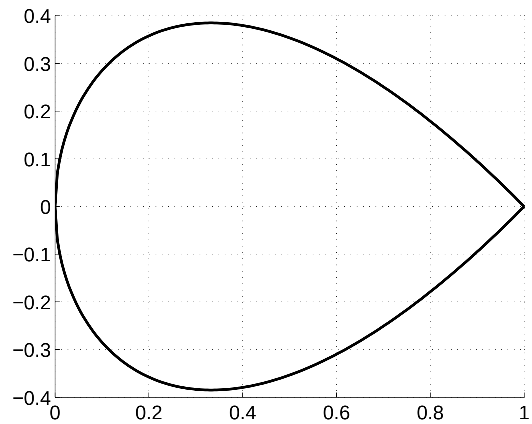

Figure 3: Unit airfoil for $C_{1.0}^{0.5}$

Changes in the coefficient vector A will lead to a variation in shape around the unit airfoil. Figure 4 shows the unit airfoil (black line) and the airfoil belonging to a coefficient vector of $[1 \quad 2 \quad 1/2 \quad 2 \quad 1/2 \quad 1]$ (blue line).

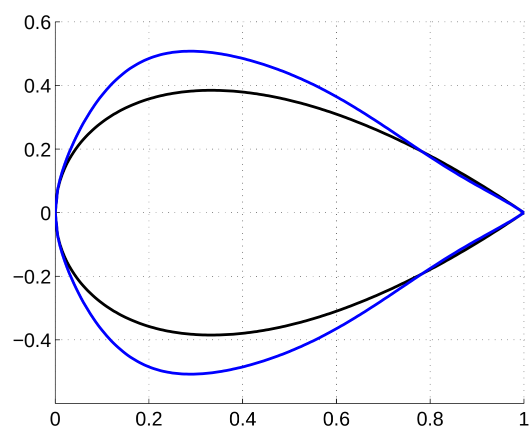

Figure 4: Airfoil shape belonging to $A = [1 \quad 2 \quad 1/2 \quad 2 \quad 1/2 \quad 1]$ (blue line) and $A = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]$ (black line)

A special feature of the CST method is that the values of the Shape Function at $\psi = 0$ and $\psi = 1$ are directly related to the leading edge nose radius and the boat-tail angle respectively, [2]. In the example of figure 4 this means that both the leading edge nose radius and the boat-tail angle should be the same, since the first and last value of the A vector are equal for both cases. From the plot it appears that this is indeed the case.

The advantage of using the CST method in combination with Bernstein polynomials is that every reasonable coefficient vector A leads to a realistic, smooth shape of the airfoil. However, any change in A will lead to a global change in airfoil geometry, thereby not allowing for any local modifications. Another disadvantage is that an increase in control points automatically leads to an increase in the order of the polynomial, which might not be required. These disadvantages can be overcome by using B-splines instead of Bernstein polynomials for the Shape Function.

3.3.2 B-splines as Shape Function

An alternative approach is to use B-splines as a Shape Function, instead of Bernstein polynomials. In this method the ζ-coordinates of the control points are used as the control variables. The ψ-coordinates can be chosen as required. It is usually advantageous to have more control points on the leading edge of the airfoil. Using B-splines, this can be accomplished without having to increase the order of the curve.

The main difference between a conventional B-spline and a B-spline coupled to the CST method is that the ζ-component of the curve is multiplied by $\sqrt{\psi}(1-\psi)$. This means that if the ζ-components are all equal to one, the airfoil will be the unit airfoil. This is a nice similarity with the CST/Bernstein method. However, changing one component of the ζ-vector will only lead to a local change in airfoil geometry. Figure 5 shows the airfoil shape determined using the CST/Bernstein method for $A = [1 \quad 1.1 \quad 1.2 \quad 1.2 \quad 1.1 \quad 1]$ (blue line) and the CST/B-spline method for $\zeta = [1 \quad 1.1 \quad 1.2 \quad 1.2 \quad 1.1 \quad 1]$ (red line). Also shown are the unit airfoil (black line) and the control points for the B-spline (blue crosses). There is no significant difference between the two methods. However, when the A and ζ coefficients are chosen in a less 'smooth' manner, say $A = [1 \quad 1.1 \quad 0.8 \quad 0.8 \quad 1.1 \quad 1]$ and $\zeta = [1 \quad 1.1 \quad 0.8 \quad 0.8 \quad 1.1 \quad 1]$, then the situation changes, as can be seen in figure 6.

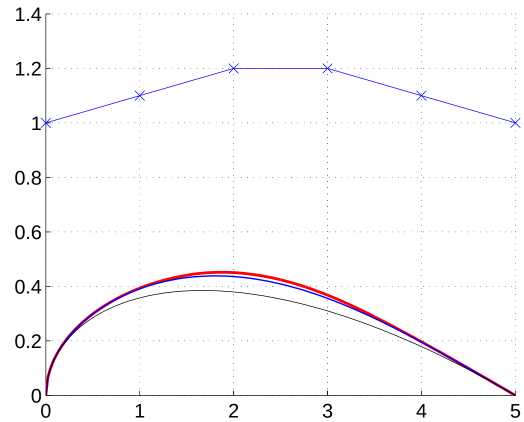

Figure 5: CST/Bernstein (blue line) and CST/B-spline (red line) airfoil belonging to $A/\zeta = [1 \quad 1.1 \quad 1.2 \quad 1.2 \quad 1.1 \quad 1]$

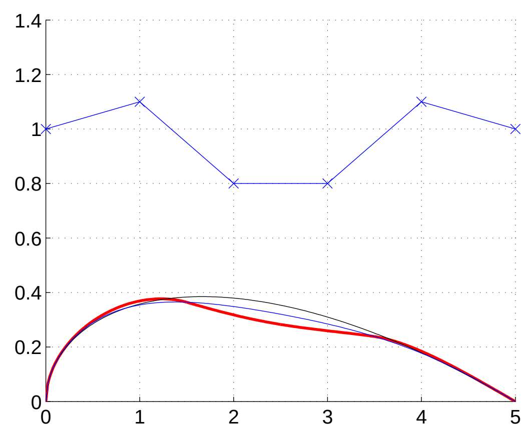

Figure 6: CST/Bernstein (blue line) and CST/B-spline (red line) airfoil belonging to $A/\zeta = [1 \quad 1.1 \quad 0.8 \quad 0.8 \quad 1.1 \quad 1]$

Now it is clear that the effect of changing the coefficients is much more local for the CST/B-spline method than for the CST/Bernstein method. The Bernstein curve still looks like a smooth airfoil, while the B-spline curve shows a clear 'dent'. Having only local control is also not practical, so the best solution seems to be a combination of both methods.

3.4 Class-Shape-Refinement-Transformation method

A useful way of combining both methods is to start with the CST/Bernstein approach and use B-splines to further refine the shape. This method will be referred to as the Class-Shape-Refinement-Transformation (CSRT) method. This method contains three functions: the Class Function, the Shape Function and the Refinement Function. Symbolically this is defined as:

$$\zeta(\psi) = C_{N2}^{N1}(\psi) \cdot S(\psi) \cdot R(\psi). \tag{15}$$

$C_{N2}^{N1}(\psi)$ is defined according to equation (10), $S(\psi)$ are the Bernstein polynomial terms and $R(\psi)$ are the B-spline basis functions.

3.5 CST method in three dimensions

In order for the CST method to be usable in preliminary aircraft design, it must be extended to three dimensions. In [2] the 3D version of equation (9) is given as:

$$\zeta(\psi, \eta) = C_{N2}^{N1}(\psi) \cdot S(\psi, \eta), \tag{16}$$

with $\eta = \frac{2y}{b}$ the non-dimensional semi-span station and b the wing span. The Class Function C_{N2}^{N1} is defined according to equation (10).

3.5.1 Bernstein polynomials as Shape Function

A Bernstein polynomials representation, like explained in section 3.3.1, can also be used in three dimensions. In order to guarantee a smooth surface, Bernstein polynomials are necessary in both x- and y-direction. This leads to the following definition of the Shape Function for the upper surface of the airfoil:

$$Su(\psi) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} Bu_{i,j} \cdot Sx_i(\psi) \cdot Sy_j(\eta), \tag{17}$$

with

$$Sx_i(\psi) = \binom{n_x}{i} \psi^i (1-\psi)^{n_x - i}, \tag{18}$$

and

$$Sy_j(\eta) = \binom{n_y}{j} \eta^j (1-\eta)^{n_y - j}. \tag{19}$$

Comparing equation (17) to equation (11) shows that the coefficient vector A changed to a coefficient matrix B of size $n_x \times n_y$. The Bernstein polynomials now represent a set of surfaces instead of curves, as is illustrated in figure 7.

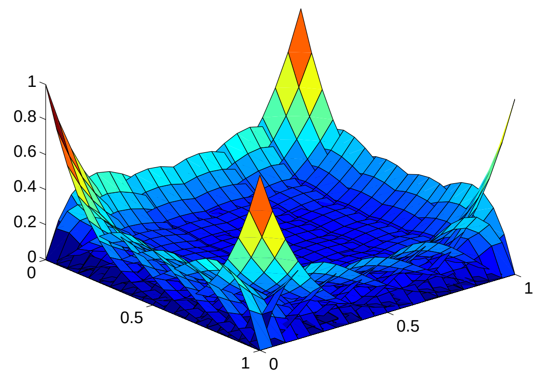

Figure 7: 3D Bernstein polynomial terms for $n_x = n_y = 5$

These surfaces again add up to one everywhere in the domain. Because the Class Function is only defined in the x-direction, this implies that if all coefficients in the matrix B are equal to one, the cross section of the surface will be the unit airfoil as shown in figure 3 and it will be constant in spanwise direction. This is shown in figure 8.

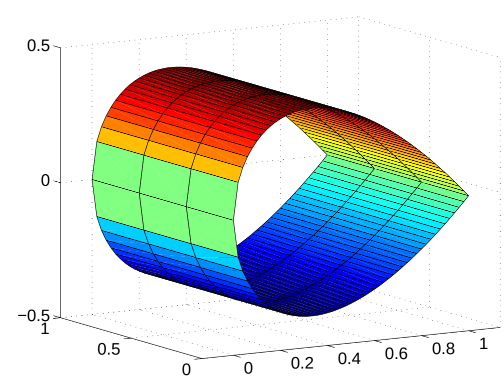

Figure 8: Unit airfoil in 3D for $C_{1.0}^{0.5}$

To show that the surface generated by this method is always smooth, figure 9 shows the surface resulting from randomly selecting the coefficients of the matrix B.

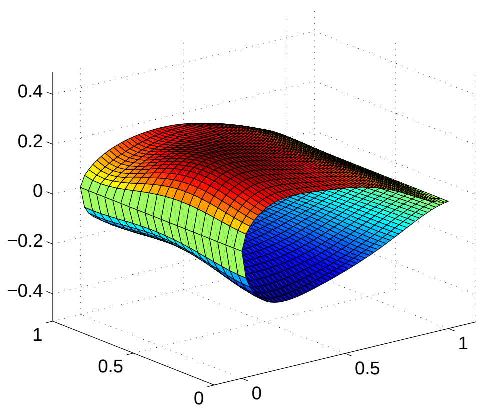

Figure 9: Surface resulting from randomly selecting the coefficients of B

3.5.2 Deviation from the unit airfoil

In real aircraft, a wing generally does not look like the unit airfoil from figure 8. Hence it should be modifiable by a tangible set of parameters. These parameters are chosen to be chord, taper, sweep, twist, dihedral, span and the airfoil sections. This section explains the definition of these parameters.

The chord length (c) is the length of the root airfoil (of a particular section) measured horizontally along the x-axis. The taper ratio (λ) is the ratio between the tip chord and the root chord (of the section), again measured horizontally along the x-axis. The taper is also applied to the thickness of the wing, to keep the airfoil shape constant. The sweep angle (Λ) is the angle about the z-axis between the leading edge of the wing and the y-axis. Sweep is applied by linearly translating the wing along the x-axis, as shown in figure 10.

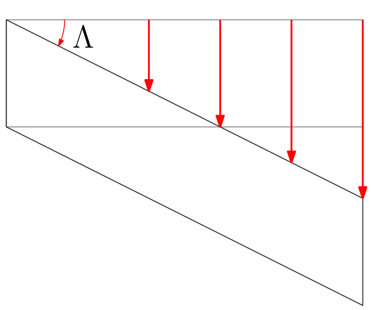

Figure 10: The definition of the sweep angle

The twist angle (ϵ) is the angle about the y-axis between the line joining the leading and trailing edge and the x-axis. Twist is applied by linearly translating the wing along the z-axis, as shown in figure 11. In spanwise direction, the twist increases linearly from the root twist angle (θ_0) over the given twist angle (θ).

The dihedral (Γ) is the angle about the x-axis between the wing leading edge and the y-axis. The dihedral is applied similarly to the sweep angle, only in (negative) z-direction. The span (s) is the length between the section root and section tip measured horizontally along the y-axis.

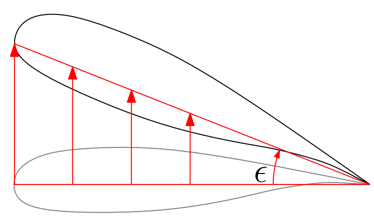

Figure 11: The definition of the twist angle

Finally, the root and tip airfoils have to be specified in terms of their Bernstein polynomial coefficients (see section 3.3.1). An easy way to ensure a smooth transition from the root to the tip is to assume a linear development from the root coefficients to the tip coefficients. It is of course possible to use any specific transition between the two.

3.5.3 Building a BWB using the CST method

To test the capabilities of the CST method and to provide a basis for an optimisation, a CST model is created of a blended-wing-body (BWB) aircraft. The aircraft geometry that is used comes from [3]. This design was originally created for the Aerospace Vehicle Design project at Cranfield College of Aeronautics in 1998, [4]. A slightly modified design has later been used as a basic configuration for the MOB project, [5]. For this reason the design will from here on be referred to as the MOB aircraft.

The model will consist of different CST surfaces that will be connected to form the MOB aircraft. These surfaces will be defined by the parameters as listed in the previous section.

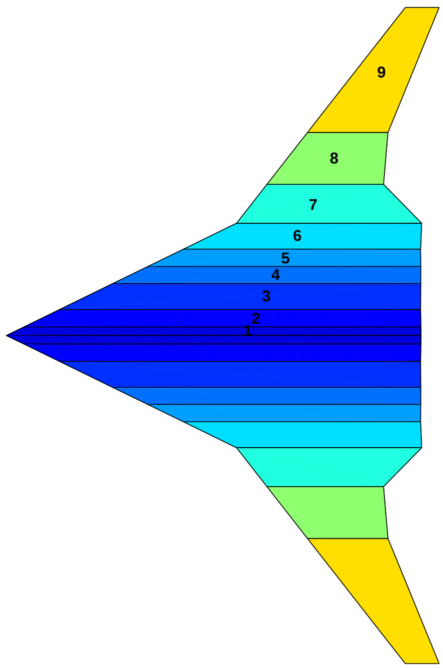

Figure 12: The MOB sections

In order to allow for the implementation of all the airfoil sections specified in [3], the MOB model consists of nine sections (see figure 12). Each section is defined using two fifth order Bernstein polynomials, one for the upper side and one for the lower side. In order to translate the known airfoil coordinates to the correct Bernstein coefficients, a simple optimisation was performed. Note that the coefficients have been made non-dimensional by dividing them by the local chord length of the section. All section data is listed in tables 1 and 2.

	c(m)	λ**(-)**	$\Lambda(^0)$	$\theta_0(^0)$	$\theta(^0)$	$\Gamma(^0)$	**s(m)**
sec 1	48.0	0.957	64	-1.6	0.0	4.4	1.0
sec 2	45.9	0.910	64	-1.6	0.0	4.4	2.0
sec 3	41.8	0.853	64	-1.6	0.2	4.4	3.0
sec 4	35.7	0.885	64	-1.4	0.2	4.4	2.0
sec 5	31.6	0.870	64	-1.2	0.2	4.4	2.0
sec 6	27.5	0.780	64	-1.0	0.5	4.4	3.0
sec 7	21.4	0.630	38	-0.5	0.5	4.4	4.5
sec 8	13.5	0.690	38	0.0	1.0	5.9	6.0
sec 9	9.3	0.420	38	1.0	-3.5	7.4	14.5

Table 1: MOB section data

Notice that not all these parameters are independent. The chord length of section 2 for example equals the chord length of section 1 multiplied by the taper of section 1. Care has to be taken in applying these relationships to guarantee a perfectly closed model.

	B1	**B2**	**B3**	**B4**	**B5**	**B6**
sec 1 root						
upper	0.2307	0.2916	0.1678	0.2550	0.1871	0.2080
lower	0.2143	0.1791	0.1637	0.2386	0.2126	0.2952
sec 2 root						
upper	0.2421	0.2829	0.1874	0.2435	0.1918	0.2090
lower	0.2219	0.1853	0.1951	0.2020	0.2560	0.2845
sec 3 root						
upper	0.2582	0.2621	0.2254	0.2143	0.2030	0.2058
lower	0.2362	0.2062	0.2141	0.2101	0.2506	0.2803
sec 4 root						
upper	0.2611	0.2363	0.2557	0.1778	0.2086	0.2004
lower	0.2553	0.2140	0.2650	0.1683	0.2680	0.2531
sec 5 root						
upper	0.2526	0.2193	0.2591	0.1579	0.2059	0.1937
lower	0.2521	0.2141	0.2774	0.1442	0.2614	0.2411
sec 6 root						
upper	0.2355	0.1981	0.2468	0.1415	0.1949	0.1848
lower	0.2295	0.2104	0.2534	0.1349	0.2359	0.2136
sec 7 root						
upper	0.1714	0.1640	0.1602	0.1342	0.1483	0.1574
lower	0.1612	0.1740	0.1548	0.1179	0.1564	0.1512
sec 8 root						
upper	0.1364	0.0781	0.1412	0.1133	0.1437	0.1469
lower	0.1312	0.0972	0.0898	0.1972	0.0305	-0.0964
sec 9 root						
upper	0.1210	0.0699	0.1246	0.1018	0.1270	0.1309
lower	0.1166	0.0869	0.0789	0.1763	0.0263	-0.0854
sec 9 tip						
upper	0.1210	0.0699	0.1246	0.1018	0.1270	0.1309
lower	0.1166	0.0869	0.0789	0.1763	0.0263	-0.0854

Table 2: MOB Bernstein coefficients

The MOB model that follows from this input is shown in figure 13.

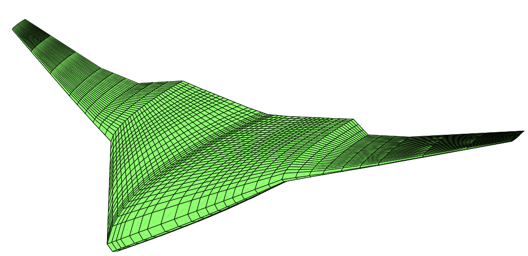

Figure 13: The MOB geometry

3.5.4 B-splines as Shape Function

As was the case for the Bernstein polynomials, B-splines can also be applied in 3D. Equation (5) is expanded with an extra set of basis functions. This leads to the following parameteric equation for the B-spline surface:

$$\mathbf{p}(u,w) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{p}_{ij} N_{i,k}(u) N_{j,l}(w). \tag{20}$$

The control points $\mathbf{p}_{ij}$ are now the vertices of a control polyhedron. The basis functions $N_{i,k}(u)$ and $N_{j,l}(w)$ are defined in the same way as for 2D. For $k = 3$, equation (20) can be expanded as follows for the interior points of the control polyhedron:

$$\begin{aligned} \mathbf{p}_{i,j}(u) = & \\ & \mathbf{U}\mathbf{M}_{S_i}\mathbf{P}\mathbf{M}_{S_j}^T\mathbf{W}^T \\ = \tfrac{1}{2} & \begin{bmatrix} u^2 & u & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdots \\ \cdots & \begin{bmatrix} \mathbf{p}_{i-1,j-1} & \mathbf{p}_{i-1,j} & \mathbf{p}_{i-1,j+1} \\ \mathbf{p}_{i,j-1} & \mathbf{p}_{i,j} & \mathbf{p}_{i,j+1} \\ \mathbf{p}_{i+1,j-1} & \mathbf{p}_{i+1,j} & \mathbf{p}_{i+1,j+1} \end{bmatrix} \cdots \\ \cdots \tfrac{1}{2} & \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}^T \begin{bmatrix} w^2 \\ w \\ 1 \end{bmatrix}. \end{aligned} \tag{21}$$

If $i = 1$ and/or $j = 1$ then the matrices $\mathbf{M}_{S_i}$ and/or $\mathbf{M}_{S_j}$ change to:

$$\mathbf{M}_{S_0} = \begin{bmatrix} 1 & -\frac{3}{2} & \frac{1}{2} \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}. \tag{22}$$

Similarly, if $i = n-1$ and/or $j = m-1$ then the matrices $\mathbf{M}_{S_i}$ and/or $\mathbf{M}_{S_j}$ change to:

$$\mathbf{M}_{S_e} = \frac{1}{2} \begin{bmatrix} 1 & -3 & 2 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}. \tag{23}$$

Now it is again possible to model the MOB aircraft from the previous section. However, the coefficients describing the MOB shape will be different from those in table 2. The coefficients for a B-spline representation of the MOB with six control points for each B-spline are listed in table 3.

	B1	**B2**	**B3**	**B4**	**B5**	**B6**
sec 1 root						
upper	0.2503	0.2447	0.2256	0.2161	0.2074	0.2014
lower	0.2092	0.1863	0.1918	0.2159	0.2489	0.2884
sec 2 root						
upper	0.2568	0.2480	0.2291	0.2170	0.2079	0.2033
lower	0.2163	0.1969	0.2003	0.2197	0.2491	0.2865
sec 3 root						
upper	0.2626	0.2513	0.2328	0.2166	0.2067	0.2039
lower	0.2311	0.2158	0.2153	0.2258	0.2481	0.2807
sec 4 root						
upper	0.2580	0.2461	0.2295	0.2094	0.2005	0.2019
lower	0.2478	0.2349	0.2293	0.2287	0.2392	0.2602
sec 5 root						
upper	0.2467	0.2362	0.2209	0.2005	0.1928	0.1967
lower	0.2449	0.2361	0.2282	0.2203	0.2285	0.2505
sec 6 root						
upper	0.2282	0.2182	0.2052	0.1866	0.1813	0.1878
lower	0.2266	0.2225	0.2129	0.2026	0.2061	0.2221
sec 7 root						
upper	0.1714	0.1644	0.1549	0.1472	0.1478	0.1568
lower	0.1658	0.1639	0.1525	0.1445	0.1450	0.1546
sec 8 root						
upper	0.1233	0.1097	0.1181	0.1279	0.1377	0.1485
lower	0.1233	0.1089	0.1143	0.1115	0.0341	-0.0946
sec 9 root						
upper	0.1095	0.0976	0.1049	0.1137	0.1224	0.1320
lower	0.1096	0.0969	0.1016	0.0991	0.0303	-0.0840
sec 9 tip						
upper	0.1095	0.0976	0.1049	0.1137	0.1224	0.1320
lower	0.1096	0.0969	0.1016	0.0991	0.0303	-0.0840

Table 3: MOB B-spline coefficients

4 Analysis

In this preliminary design stage a panel method program called VSAERO is chosen for the aerodynamic analysis. This section briefly explains the workings and a validation of VSAERO.

4.1 Flow analysis

4.1.1 Laplace's equation

Panel methods are based on Laplace's equation for the velocity potential:

$$\nabla^2 \Phi = 0. \tag{24}$$

The control volume to which equation (24) is applied is shown in figure 14. The boundary of the control volume consists of three parts: the body surface S_B, the wake surface S_W and the outer boundary S_∞. The potential is split into two parts:

the potential in the flow field Φ and the potential in the inner region Φ_i.

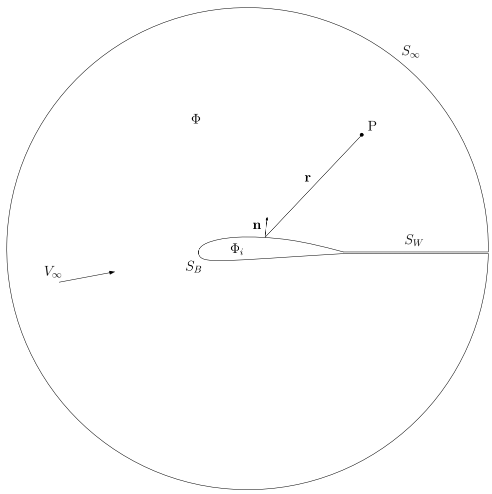

Figure 14: Control volume

Applying Green's theorem to equation (24) for both the flow field and the inner region results in:

$$\Phi_P = \frac{1}{4\pi}\iint_{S_B+S_W+S_\infty}(\Phi-\Phi_i)\mathbf{n}\cdot\nabla(\tfrac{1}{r})dS - \frac{1}{4\pi}\iint_{S_B+S_W+S_\infty}\tfrac{1}{r}\mathbf{n}\cdot(\nabla\Phi-\nabla\Phi_i)dS. \quad (25)$$

This expression describes the velocity potential in any point P expressed in terms of surface integrals. The first integral represents the local jump in potential and the second integral represents the local jump in the normal component of the velocity. In terms of singularities, the first integral represents the doublet distribution, while the second integral represents the source distribution, [6].

4.1.2 Boundary conditions

The boundary condition that is chosen needs to specify a zero normal velocity on the surface. This can be done either directly by applying the Neumann condition ($\nabla\Phi\cdot\mathbf{n}=0$) or indirectly by applying the Dirichlet condition (Φ specified on the boundary). VSAERO makes use of the former.

In VSAERO the user has the possibility to specify a certain normal velocity (V_n) to model for example boundary layer suction. Furthermore, boundary layer displacement (δ^*) can be modelled by assuming a certain transpiration velocity. Finally, a normal velocity component due to rotation (ω) can be specified. This changes the boundary condition to, [7]:

$$\nabla\Phi\cdot\mathbf{n} = -V_n + (V_\infty - \omega\times r_g)\cdot\mathbf{n} + \frac{\partial}{\partial s}(V\delta^*) \quad (26)$$

4.1.3 Singularity model

Equation (25) does not have a unique solution. There is still an infinitely large number of combinations of source and doublet distributions that satisfy the flow problem. A number of techniques are possible to find a unique combination of singularities. It is for example possible to specify the source distribution and solve for the doublet distribution, or vice versa. Another possibility is to apply boundary conditions to the inner flow, either to the velocity or the velocity potential. VSAERO specifies the velocity potential of the inner flow to be equal to the freestream velocity potential, i.e. $\Phi_i=\Phi_\infty$. An alternative is to assume zero velocity potential in the inner region, but this method is more sensitive to bad panelling.

4.1.4 Compressibility

In the derivation of Laplace's equation it is assumed that the flow is incompressible. However, some methods are available to approximate the effects of compressibility on the solution. The best method available in VSAERO for external flows is the Karman-Tsien method. This method relates the compressible and incompressible velocities as:

$$\frac{V}{V_\infty} = \frac{(\frac{V}{V_\infty})_{inc}(1-\lambda)}{1-\lambda(\frac{V}{V_\infty})^2_{inc}}, \quad (27)$$

where

$$\lambda = \frac{M_\infty^2}{(1+\sqrt{1-M_\infty^2})^2}. \quad (28)$$

4.1.5 Viscosity

In order to model aircraft drag, viscous effects need to be included. Since the potential flow model itself neglects viscosity, it needs to be coupled to a boundary layer method. VSAERO uses a method developed by Thwaites, later adapted by Curle. This method, which is explained in detail in [7], is used to calculate the displacement thickness δ^*, which can then be substituted into equation (26).

In the transonic regime, viscous drag is not just caused by the displacement thickness of the boundary layer. Shockwave/boundary layer interaction effects also play an important role. VSAERO is not capable of modelling these effects. Using a Euler code it is possible to calculate viscous drag caused by shock wave/boundary layer interaction, as long as the shock wave is weak. Generally this is the case for transonic transport aircraft.

4.2 VSAERO Validation

In order to determine whether VSAERO will provide a reliable analysis for preliminary design optimisation, the program is validated using reference data available from the AGARD-AR-303 study, [8].

4.2.1 AGARD

AGARD, or Advisory Group for Aerospace Research & Development, is a NATO initiative aimed at bringing together knowledge in the fields of science and technology relating to aerospace within its member nations. One particular study, AGARD-AR-303, was performed to provide the scientific community with a selection of experimental test cases for the validation of CFD codes. For these tests the geometry of the DLR-F4 model was used. The model was tested in three different wind tunnels (NLR-HST, ONERA-S2MA, DRA-8ft x 8ft DRA Bedford) in order to distinguish and correct the wind tunnel effects as much as possible.

4.2.2 DLR-F4 Model

The DLR-F4 model consists of a swept wing connected to a fuselage. No wing/body fairing is used and no tail is present. The geometry of the model and wing can be seen in figures 15 and 16. A complete geometry data set for the DLR-F4 model was freely available online from the proceedings of the 1st AIAA CFD Drag Prediction Workshop held in Anaheim, California in 2001. For this workshop, the data supplied with the AGARD-AR-303 report was used to produce a complete point by point description of the wind tunnel model.

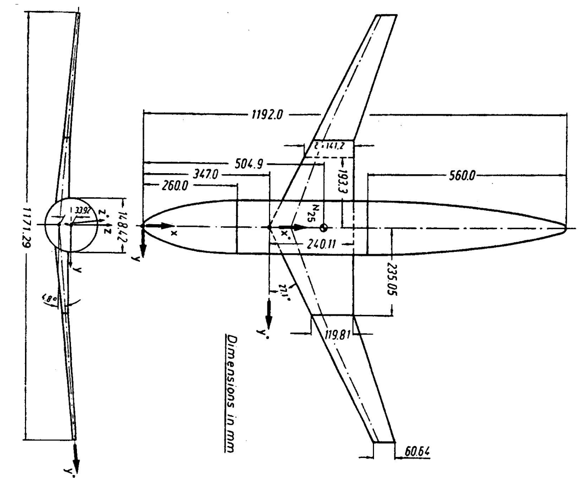

Figure 15: Definition of the DLR-F4 wind tunnel model

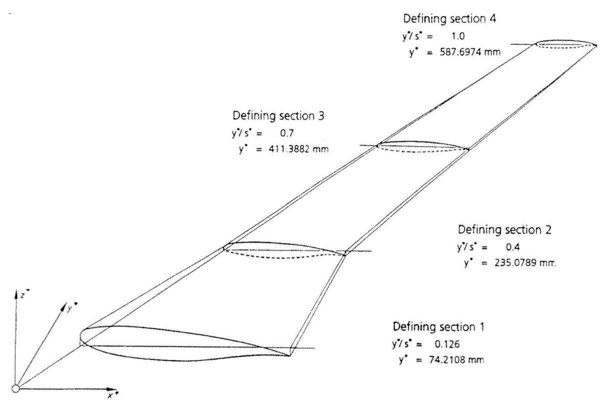

Figure 16: Definition of the DLR-F4 wing

4.2.3 Results

The model was analyzed for $M = 0.75$ and $Re = 3 \cdot 10^6$. The angle of attack was varied from -6 to +4 degrees. From the output data, a lift curve and lift-drag polar were constructed and compared to the wind tunnel data from the AGARD-AR-303 report. The results are shown in figures 17 and 18.

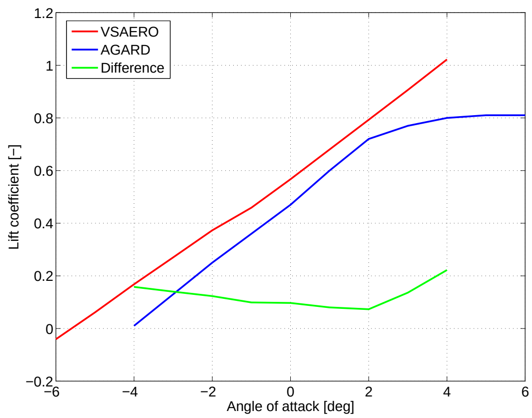

Figure 17: Lift curve comparison between AGARD and VSAERO

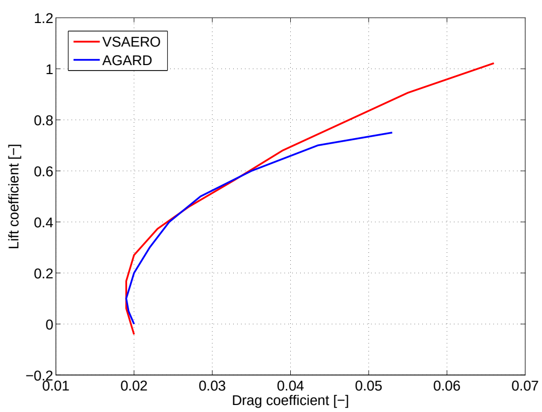

Figure 18: Lift-drag polar comparison between AGARD and VSAERO

Figure 17 shows the wind tunnel results (blue), the VSAERO results (red) and the difference between the two (green). VSAERO clearly overpredicts the lift coefficient. The reason for this is yet unclear, although the same phenomenon has been found for other codes in literature, [9] and [10].

Figure 18 shows the lift-drag polars from AGARD and VSAERO. For a lift coefficient up to about 0.7 the computational results are close to the experimental results. There is a small discrepancy around $C_l = 0.3$, but overall these results are very satisfactory, especially considering the fact that the tests were performed at a Mach number of 0.75. Although this is still below the drag rise Mach number for this aircraft configuration ($M_{div} \approx 0.8$), transonic effects will likely not be negligible.

For values of the lift coefficient above 0.7 the computational results for both C_l and C_d start to deviate from the experimental results. This is also visible in figure 19, which depicts the lift-to-drag ratio versus the lift coefficient. This deviation is mainly caused by separation which is not accurately modelled by VSAERO.

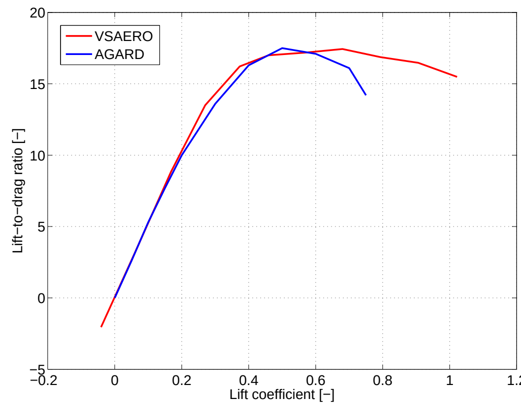

Figure 19: Lift-to-drag ratio vs lift coefficient

5 Optimisation

The optimisation module of the aerodynamic initiator still needs to be developed. This work will start in the coming months. This section will give a short description of the optimisation methods that are being considered. Also an example is given of what happens to the aircraft drag when one of the variables of the MOB aircraft is varied.

5.1 Optimisation methods

Optimisation problems can be divided into continuous, discrete and mixed continuous-discrete problems. All of these problems are common in preliminary aircraft design. Optimising the sweep angle for example would be a continuous problem, while determining the optimum amount of tail surfaces would be a discrete problem. Doing both simultaneously would naturally result in a mixed problem.

Continuous optimisation methods are generally based on calculating the gradient of some objective function. Most of these methods require more function evaluations as the number of design variables increases. An exception to this rule is 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 optimisation process is independent of the number of design variables. A disadvantage of a continuous optimisation method is that it is only capable of finding a local optimum.

Discrete optimisation methods do not make use of the gradient of a function and are often capable of finding a global optimum. They are however considerably less efficient than continuous methods. Some discrete methods that seem promising are genetic algorithms, [12] and particle swarm optimisation, [13].

Mixed optimisation problems can be solved in many different ways. Probably the most straightforward way is to relax the integer values as real values and apply a real-valued optimisation method to the relaxed problem. The optimised real-valued solution can then be rounded back to an integer solution. Although simple, this method is usually inaccurate and can even lead to infeasible solutions. An interesting alternative could be the so-called mixed-integer hybrid differential evolution (MIHDE) method, [14]. This method makes use of evolutionary algorithms to efficiently find a global solution to a mixed optimisation problem.

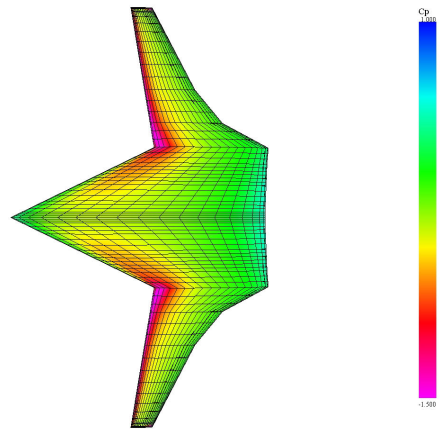

Figure 20: MOB with an outer sweep angle of -10 degrees

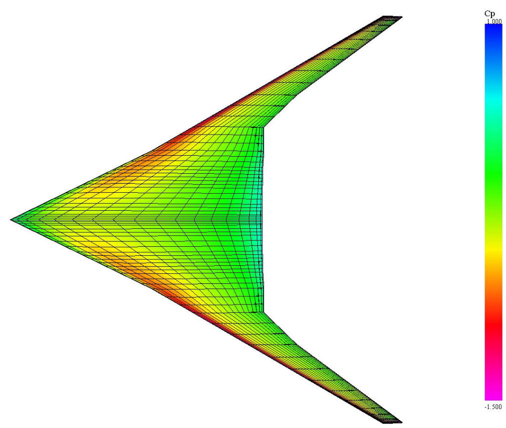

Figure 21: MOB with an outer sweep angle of +55 degrees

5.2 Varying the sweep

To show that the combination of parameterisation and analysis is capable of producing an optimum shape, a one-dimensional example is given in this section. The sweep angle of the outer wing of the MOB aircraft is varied from -10 to +55 degrees, as is shown in figures 20 and 21. Because of the way in which the sweep is varied (see section 3.5.2) the surface area and wing span will stay the same and hence the reference values that are required by VSAERO do not change.

Figure 22 shows the lift-to-drag ratio calculated by VSAERO as a function of the sweep angle. Although this example is one-dimensional and therefore not nearly as complex as a real design problem, it does show that the developed software is capable of producing a clear optimum value that would be found by any gradient based optimisation algorithm. The calculated optimum of 35 degrees is close to the sweep angle of the actual MOB aircraft, which is 38 degrees.

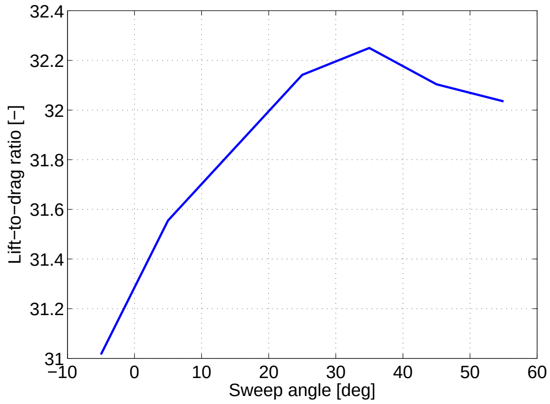

Figure 22: Lift-to-drag ratio as a function of the sweep angle

6 Conclusions and recommendations

This paper presented the work done so far on an aerodynamic initiator for the Design Engineering Engine being developed at Delft University of Technology. The Class-Shape-Transformation method has proven to be very useful and intuitive as a parameterisation tool, especially after adding the Refinement Function based on b-splines. The panel method program VSAERO is considered sufficient for the aerodynamic analysis now, but should eventually be replaced by a Euler or Navier-Stokes code for better capturing the high-speed flow characteristics, such as shock wave/boundary layer interaction. For the optimiser part of the initiator some research has been done, but most of the actual work still needs to be performed. This work will start later this year.

References

[1] Intergovernmental Panel on Climate Change, *IPCC Special Report on Aviation and the Global Atmosphere*, 1999

[2] B.M. Kulfan, *Universal Parameteric Geometry Representation Model*, Journal of Aircraft, Vol. 45, No. 1, January-February 2008

[3] H. Smith, A. Yarf-Abbasi, *The MOB Blended Body Reference Aircraft*, Multidisciplinary Optimisation for Blended Wing Body Project, Technical Report MOB/4/CU/Treport/No4, Cranfield University, March 2001

[4] H. Smith, *Blended Wing Body Preliminary Design Study BW-98*, Aerogram, Vol. 9, No. 4, December 1999

[5] A. Morris, P. Arendsen, G. LaRocca, M. Laban, R. Voss, H. Hönlinger, *MOB - A European Project On Multidisciplinary Design Optimisation*, 24th International Congress of the Aeronautical Sciences, 2004

[6] J. Katz, A. Plotkin, *Low Speed Aerodynamics*, Cambridge University Press, Cambridge, 2001

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

[8] AGARD, *A Selection of Experimental Test Cases for the Validation of CFD Codes*, Volume II, Virginia, U.S.A., August 1994

[9] G. Waller, *Prediction of the DLR-F4 Model Drag Using MGAERO*, AIAA Drag Prediction Workshop, Anaheim, California, U.S.A., 9-10 June 2001

[10] U. Goldberg, *CFD++ - Unstructured, Mixed Element Cell-based Solver*, AIAA Drag Prediction Workshop, Anaheim, California, U.S.A., 9-10 June 2001

[11] J.R.R.A. Martins, J.J. Alonso, J.J. Reuther, *A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design*, Optimization and Engineering, Vol. 6, Pages 33-62, 2005

[12] M. Green, R. Cheng, *Genetic Algorithms & Engineering*, John Wiley & Sons, Inc., New York, 1997

[13] G. Venter, J. Sobieszczanski-Sobieski, *Particle Swarm Optimization*, AIAA 2002-1235, 2002

[14] Y. Lin, F. Wang, K. Hwang, *A Hybrid Method of Evolutionary Algorithms for Mixed-Integer Nonlinear Optimization Problems*, IEEE, 1999