Shape up!

Controlling drag through three-dimensional shape optimization

by Michiel Straathof

During the early development of aircraft, not much attention was paid to aerodynamic efficiency. Structural design did not yet allow for cantilever wings - wings that are only supported on one side. Instead, they had to be supported with numerous struts and wires, which caused huge amounts of parasitic drag. During WWI speed and range became important for fighters, bombers and observation aircraft. The drag of an aircraft increases with the square of its speed, so drag reduction came high on the agenda. The resulting advances in aerodynamic design can be seen in the Spirit of St. Louis, the aircraft with which Charles Lindbergh performed his famous flight across the Atlantic Ocean in 1927. It could fly non-stop for over 33 hours, covering a distance of almost 6500 km. Lindbergh's airplane shows a number of aerodynamic design

Fig. 8. Charles Lindbergh next to his Spirit of St. Louis.

features. First, the steel tube fuselage is covered with fabric to allow the air to flow past smoothly. Second, the struts are aerodynamically shaped for low drag and their number is kept to a minimum. In later versions a cowling (cover) was added to the propeller.

Towards the end of the 1920s wooden monocoque fuselages and wings appeared. These had much cleaner lines, but a few struts were still necessary to support the wings, as on the Fokker F.VII.

True aerodynamic optimization was first achieved in the 1930s with the Boeing 247 (1934) and the Douglas DC-2 (1935). Except for the propellers and the rear landing gear, the entire exterior of these aircraft consists of a smooth aluminum skin; even the engines are completely covered. This skin could carry part of the loads occurring during flight, so external struts were no longer required. Also, the intersection between the wings and the fuselage has been aerodynamically optimized, to prevent the air flow from separating[1]. The wings are tapered and swept backwards, which also decreases drag.

[1] FLOW TRANSITION AND SEPARATION

Air that flows past a surface can go through different stages. At the leading edge of a wing, the flow is usually laminar, meaning that it is very smooth and causes very little friction drag. For sail planes the area of laminar flow can extend all the way to the trailing edge, while for airliners it usually doesn't extend beyond about 15% of the wing chord. Instabilities in the flow – known as Tollmien-Schlichting waves - eventually cause the laminar flow to transition to a more chaotic state called turbulent flow. Turbulent flow causes considerably more friction drag than laminar flow, but it is less likely to separate due to its energetic nature. Separation generally occurs in areas where there is a strong positive pressure gradient, i.e. in areas of large curvature. This can be actual curvature in the geometry or induced curvature caused by a large angle of attack. On passenger aircraft, flow separation is always unwanted since it creates enormous amounts of pressure drag and could even lead to loss of lift and/or control of the aircraft.
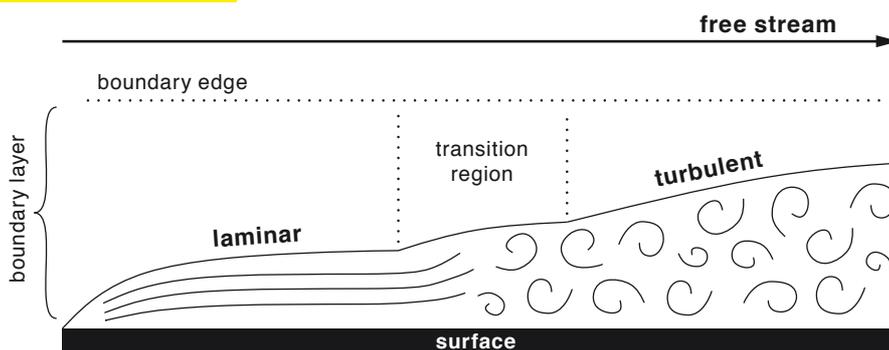
Fig. 9. Laminar and turbulent flow.

In December 2009, the latest airliner to enter production, the Boeing 787, took to the skies. Compare the B787 to the DC-2 and it is clear that over a period of 75 years, nothing changed in terms of aircraft configuration. That was not for lack of trying.

Over the years, various novel aircraft concepts have been considered, but none of them actually made it into production. One design that has been extensively studied is the joined-wing or box wing aircraft. Creating lift using two sets of wings, joined together at the tips, could dramatically reduce induced drag by weakening the wing tip vortices[2]. A promising concept, but a lot of structural challenges will have to be overcome, such as making the box-wing structure stiff enough.

Another novel configuration is the blended-wing-body aircraft. By merging the wings with the fuselage, the entire exterior surface of the aircraft contributes to the generation of lift. In a conventional aircraft, the fuselage only generates drag, without contributing to the lift. Challenges to overcome with this configuration mainly concern stability and control.
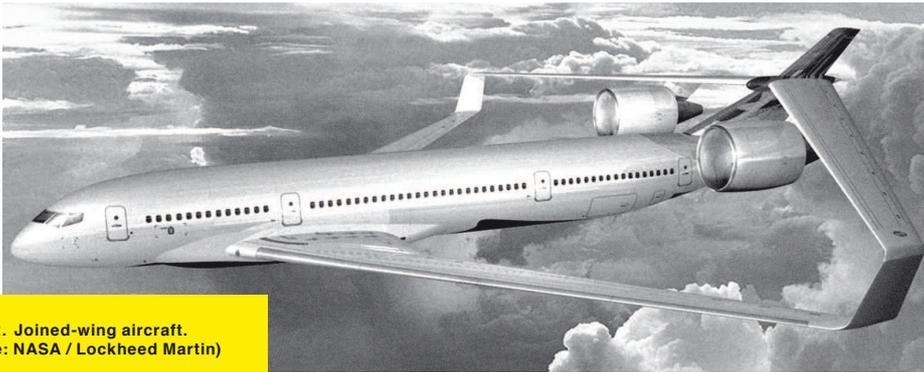

Fig. 12. Joined-wing aircraft. (image: NASA / Lockheed Martin)

Fig. 13. Blended-wing-body aircraft. (image: NASA / Boeing)

[2] LIFT DISTRIBUTION

The planform – the shape of the wing from above (or more likely, from below) – has a significant influence on the aerodynamic performance. It is one of the major factors determining the spanwise distribution of lift on the wing. During flight, the lower side of a wing experiences high pressure and the upper side low pressure; this causes an upward force: lift. At the wing tip, the high and low pressure regions come together and cause the air to flow from the lower side of the wing to the upper side and roll up into vortices. These vortices in turn cause a downwash at the leading edge of the wing, resulting in a rearward tilt of the lift vector. This vector now has a component opposite to the direction of travel, which is called induced drag. Induced drag is inevitable, but it can be reduced by modifying the wing planform.
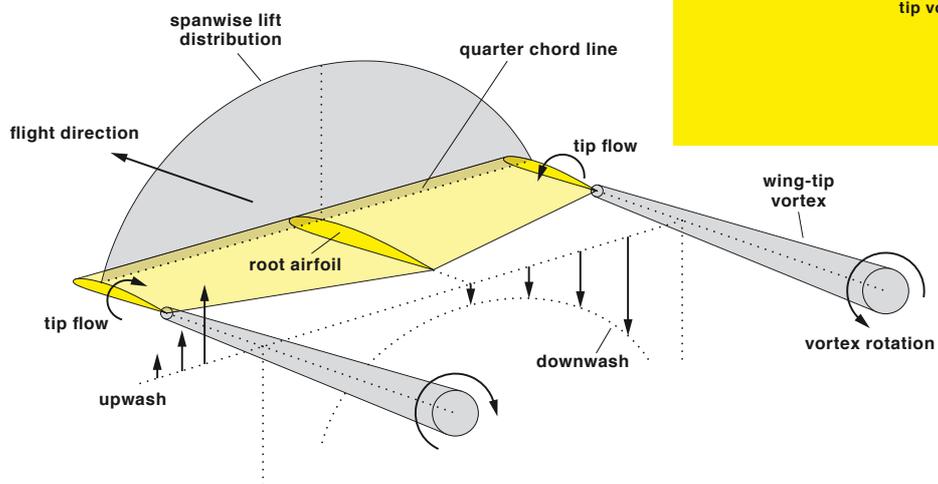
spanwise lift
distribution

quarter chord line

flight direction

tip flow

wing-tip
vortex

root airfoil

tip flow

vortex rotation

downwash

upwash

Fig. 14. Induced drag due to wing tip vortices.

A wing with an elliptical planform experiences the least amount of induced drag for a given aspect ratio. Elliptical wings are difficult to manufacture, because of the required curvature in the leading and trailing edges. Tapered wings form a good alternative, but they produce up to 15% more drag. The number by which the induced drag exceeds that of the elliptical lift distribution is called the Oswald factor.

Another way of reducing induced drag is by increasing the aspect ratio of the wing. This will reduce wing tip vortices and hence induced drag. The aspect ratio is a measure of slenderness and can be expressed as: $AR = b^2 / S$, where b is the semi-wing span and S the wing area. Sailplanes typically have very high aspect ratios, resulting in extremely low induced drag.

Despite the lack of new configurations, a number of subtle differences can be distinguished between the DC-2 and the B787. The wings and tail surfaces of the B787 are very slender and highly tapered, lowering induced[2] and wave[3] drag. Additionally, the nose section of the B787 is more aerodynamically shaped and the landing gear is fully retractable. These characteristics give the B787 a much higher aerodynamic efficiency.

In general, a number of factors have led to the superiority of modern aircraft. One is the advancement in materials. The slender wings of the B787 could simply not have been produced 70 years ago. Another one is the availability of computer power. The design of the DC-2 was purely driven by the experience of the designers, validated by wind tunnel testing. These days, computer algorithms are used to accurately model the airflow around an aircraft and then to numerically optimize aircraft shapes. These powerful tools are capable of optimizing complete aircraft.

As the speed of an aircraft increases, there comes a point where some of the air flow on the wing is supersonic, even though the aircraft itself is still flying at subsonic speed. An air particle moving over the wing will accelerate from subsonic to supersonic and decelerate back to subsonic speed again. This deceleration leads to a shock wave on top of the wing. Because it takes energy to form this shock wave, this process translates into a form of drag called wave drag. The strength of the shock wave and hence the amount of wave drag depends on the component of the flow velocity that is perpendicular to the wing. For a straight wing, this component is equal to the speed of the whole aircraft. For a swept wing it can be much smaller. That is why a swept-wing aircraft is able to travel much closer to the speed of sound, without the air flow becoming supersonic anywhere on the wing.
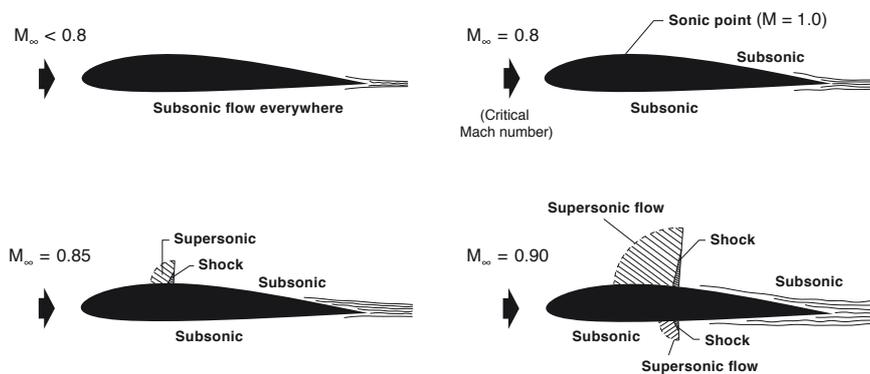
$M_\infty < 0.8$

Subsonic flow everywhere

$M_\infty = 0.8$

Sonic point (M = 1.0)

Subsonic

Subsonic

(Critical Mach number)

$M_\infty = 0.85$

Supersonic

Shock

Subsonic

Subsonic

$M_\infty = 0.90$

Supersonic flow

Shock

Subsonic

Subsonic

Shock

Supersonic flow

Fig. 15. Shock wave formation.

CFD

Calculating the flow of air around an object using computer algorithms is called computational fluid dynamics or CFD. This is done with the help of a set of equations named after French engineer Claude-Louis Navier and British mathematician George Gabriel Stokes: the Navier-Stokes (N-S) equations. These describe the motion of fluid substances. Unfortunately, no analytical solutions of the Navier-Stokes equations are known, meaning that the equations always have to be solved numerically. The most straightforward and time-consuming way of solving the N-S equations is a direct numerical simulation. This is however not (yet) feasible for use in aircraft design, because it simply takes too long to compute the flow. This problem can be solved by making a number of assumptions about the flow, such as that it is inviscid and/or incompressible.[4]

The Navier-Stokes (N-S) equations describe the behavior of all fluids (including gases, such as air) at all scales. For incompressible, Newtonian fluids, the N-S equations can be written as:

$$\rho \left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} \right) =$$
$$\rho g_x - \frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left[2\mu\frac{\partial u}{\partial x} + \lambda \nabla \cdot \mathbf{V} \right] + \frac{\partial}{\partial y}\left[\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] + \frac{\partial}{\partial z}\left[\mu \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right]$$

$$\rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) =$$
$$\rho g_y - \frac{\partial p}{\partial y} + \frac{\partial}{\partial y} \left[2\mu \frac{\partial v}{\partial y} + \lambda \nabla \cdot \mathbf{V} \right] + \frac{\partial}{\partial z} \left[\mu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \right] + \frac{\partial}{\partial x} \left[\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right]$$

$$\rho \left(\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) =$$
$$\rho g_z - \frac{\partial p}{\partial z} + \frac{\partial}{\partial z} \left[2\mu \frac{\partial w}{\partial z} + \lambda \nabla \cdot \mathbf{V} \right] + \frac{\partial}{\partial x} \left[\mu \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right) \right] + \frac{\partial}{\partial y} \left[\mu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \right]$$

As mentioned, no analytical solutions of the Navier-Stokes equations are currently known. In fact, a $1 million prize has been offered to the first person who finds an analytical solution or proves that no such solution exists. So for now, the only way to find a solution to the N-S equations is to solve them numerically. This can be done directly on the full set of equations (Direct Numerical Simulation or DNS) or one or more assumptions can be made about the flow to simplify the computation. Separating the turbulent velocity fluctuations from the mean velocity leads to the Reynolds-Averaged-Navier-Stokes (RANS) equations. Neglecting viscosity results in the Euler equations and assuming irrotational flow finally leads to the potential flow equations. Many commercial flow solvers are available to solve these simplified sets of equations. In my work I have used an Euler code that was developed at TU Delft.

Solving the N-S equations produces a velocity field; it describes the velocity of the flow at certain points in space. Interesting properties can be derived from this velocity field, such as the flow rate and aerodynamic forces and moments. CFD is also very useful for visualizing the flow around an object.

Many flow phenomena can be easily identified by looking at a plot of the pressure distribution on a wing or aircraft. Where the isobars (lines of constant pressure) lie close together and the pressure gradient is positive a shockwave is likely to form. Stagnation points can be found at locations where the pressure coefficient is equal to 1. Areas of low pressure on top of the wing
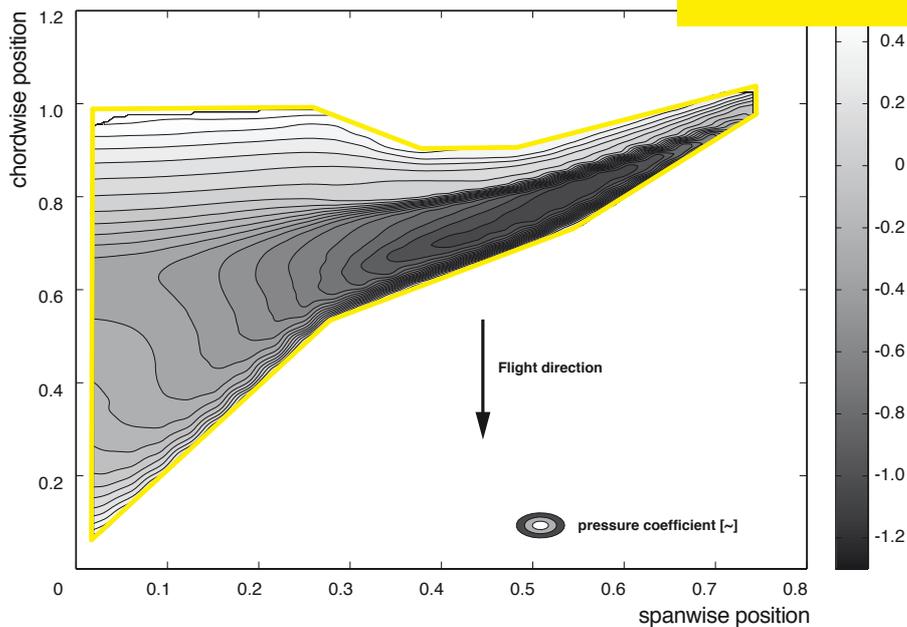
Fig. 16. Pressure contours on a blended-wing-body configuration.

(and high pressure below the wing) can give an indication about the aerodynamic moments involved. The list goes on, but an important conclusion is that computational fluid dynamics provides a powerful tool that gives insight into the flow around an aircraft in both a quantitative and a qualitative way.

PARAMETERIZATION

The shape of an object must be properly described in order to compute the flow around it. Finding a mathematical description of a shape is called parameterization. The first CFD algorithms that were used in the 1970s and 1980s were simple and thus required only simple ways of parameterizing a shape. However, as computer power grew and the flow solvers became more sophisticated, the need arose for novel parameterization methods. This is the primary focus of the CleanEra design work.
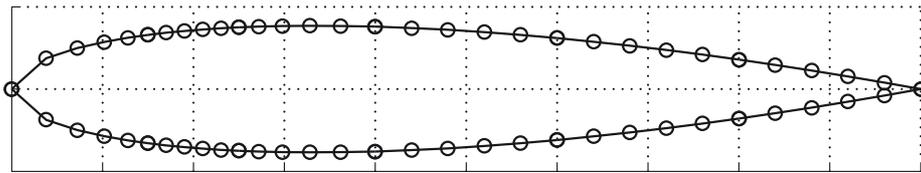
EXISTING METHODS

Fig. 17. Discrete parameterization.

The most straightforward way to parameterize a shape is by taking discrete points along its boundary and connecting those points with lines. This is not very efficient, as you need a lot of points to generate a smooth shape. Additionally, it is very difficult to maintain a smooth shape throughout the optimization process.

The number of variables required can be greatly reduced by using a polynomial representation, where the polynomial coefficients determine the shape. This results in a shape which is much smoother than with a discrete representation. A disadvantage is that in order to capture local deformations of a shape, the order of the entire polynomial needs to be increased, which could result in a high number of design variables after all.

Another alternative is to add up a number of special functions that together form the required shape. Different functions can be used for this purpose, such as Bernstein or Chebyshev polynomials[5].

[5] **POLYNOMIAL BASIS FUNCTIONS**

Instead of using a single polynomial to describe a curve, it is also possible to use a set of polynomial basis functions that form a smooth curve when added up.

One such set of basis functions are the so-called Chebyshev polynomials, which are defined by the following recurrence relationship:
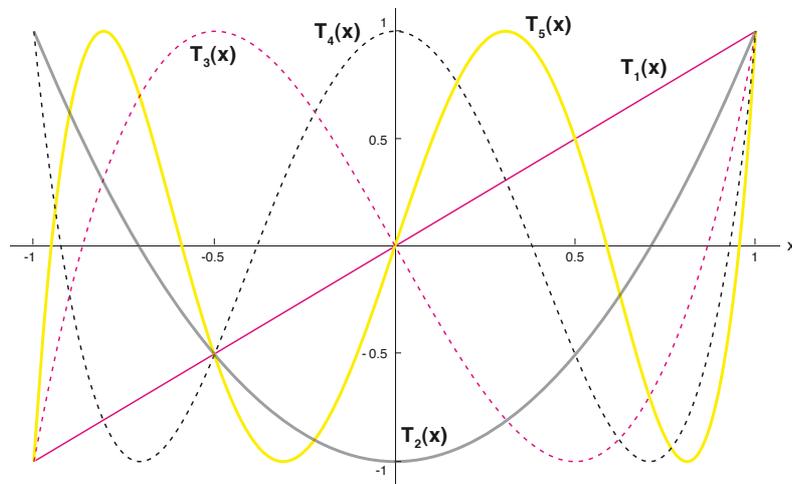
$$T_0(x) = 1$$
$$T_1(x) = x$$
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

The curve is then given by multiplying each basis function with a coefficient and then adding them all up. This is described mathematically as:

$$f(x) = \sum_{n=0}^{p} a_n T_n(x)$$

Another popular set of basis functions are the Bernstein polynomials, which have the special property that their sum is always equal to 1. They are defined as follows:
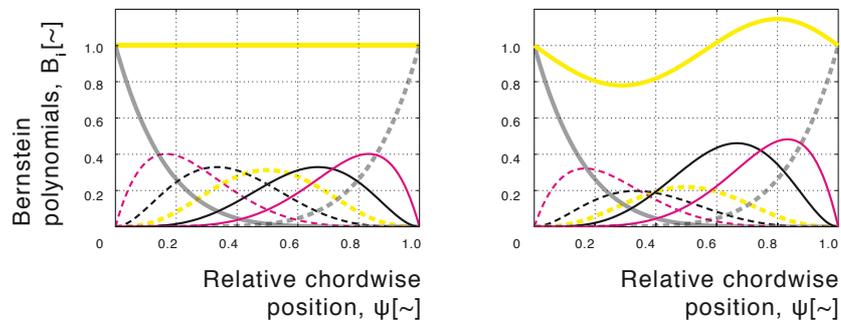
$$B_{n,p_S} = \binom{p_S}{n} x^n (1-x)^{p_S - n}$$

The Bernstein curve is then described as:

$$f(x) = \sum_{n=0}^{p_S} b_n B_{n,p_S}(x)$$

Fig. 18. (above) Chebyshev polynomials.

Fig. 19. (below) Bernstein polynomials.

CST METHOD

In 2008, a Boeing employee named Brenda Kulfan introduced a novel parameterization technique called the Class-Shape-Transformation (CST) method. This technique combines an analytical function, called the class function, and a set of Bernstein polynomials, called the shape function. The class function represents a basic class of shapes, such as an airfoil or a fuselage cross-section, while the shape function represents the deviation from this basic shape[6].

[6] CLASS-SHAPE-TRANSFORMATION (CST) METHOD

The CST method as developed by Kulfan describes the shape of a curve as the product of a class function C and a shape function S:

$$f(x) = C(x) \cdot S(x)$$

The class function is given by the following analytical function:

$$C(x) = x^{N1}(1 - x)^{N2}$$

By varying the coefficients $N1$ and $N2$, different classes of shapes can be generated, from typical round nose/sharp trailing edge airfoils to fuselage cross-sections.
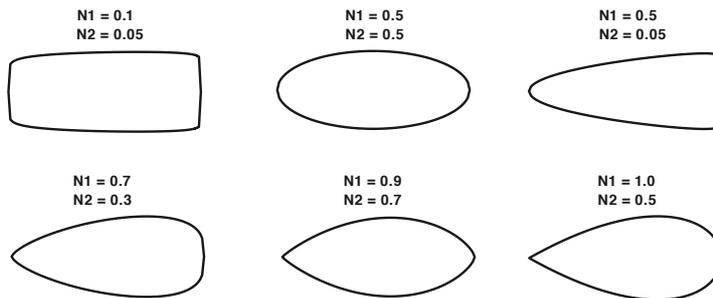
N1 = 0.1 N2 = 0.05	N1 = 0.5 N2 = 0.5	N1 = 0.5 N2 = 0.05
N1 = 0.7 N2 = 0.3	N1 = 0.9 N2 = 0.7	N1 = 1.0 N2 = 0.5

Fig. 20. Possible class functions.

The shape function consists of a set of Bernstein polynomials and can thus be described as follows:

$$S(x) = \sum_{n=0}^{p_S} b_n \binom{p_S}{n} x^n (1 - x)^{p_S - n}$$

The main advantage of the CST method is that the final shape always belongs to the class of shapes determined by the class function. For example, if $N1 = 1$ and $N2 = 0.5$, then the CST curve will always have a rounded nose and a sharp trailing edge, independent of the shape function. The shape function merely describes the deviation from the class function.

This method proved to be very useful because of its ability to handle many different airfoil and wing shapes with a relatively low number of design variables. Another advantage of the CST method is that the round nose of the airfoil is completely defined as a result of the square root term in the class function. This causes problems for most other parameterization methods.

The CST method has one big limitation: it cannot handle local deformations efficiently. When more detail is required in a specific area, the order of the entire shape function needs to be increased. This problem can be solved by adding a third function based on B-splines.

CSRT METHOD

To be able to efficiently model local shape changes, an extension to the CST technique was developed at CleanEra, called the Class-Shape-Refinement-Transformation method. As the name suggests, an extra function was added: the refinement function. This function is based on B-splines[7], which are basically strings of lower order curves. Because of the piece-wise nature of B-splines, it is possible to deform only a particular region of the curve, while keeping the rest constant. This provides the possibility to increase the detail on a specific part of the shape, without having to increase the order of the whole shape function.

[7] **B-SPLINES**

As was the case for the shape function, a B-spline curve (and hence the refinement function) consists of a set of basis functions, multiplied by a set of coefficients. For a B-spline, this set of coefficients is represented by the coordinates of so-called control points that together form a control polygon, $\bar{P}$. Mathematically, the B-spline is described as follows:

$$R(x) = \sum_{n=0}^{p_R} \bar{P}_n N_{n,k}(x)$$

The B-spline basis functions are defined iteratively:

$$\begin{aligned} N_{n,1}(x) &= 1 \quad \text{if} \quad t_n \leq x \leq t_{n+1} \\ &= 0 \quad \text{otherwise} \end{aligned}$$
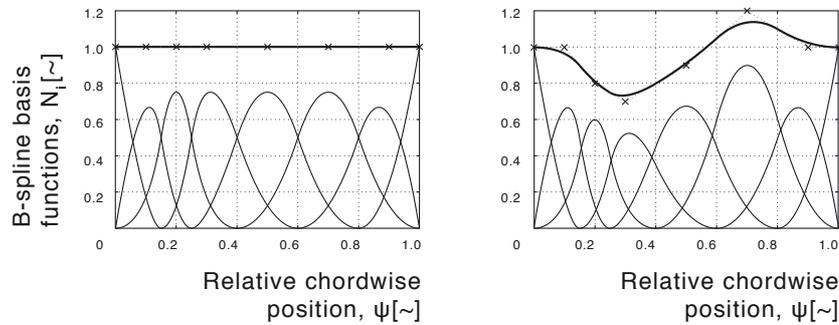
and:

$$N_{n,k}(x) = \frac{(x - t_n)N_{n,k-1}(x)}{t_{n+k-1} - t_n} + \frac{(t_{n+k} - x)N_{n+1,k-1}(x)}{t_{n+k} - t_{n+1}}$$

Where t_i are called the knot values, which relate the parametric variable x to the control points $\bar{P}$. They are defined as follows:

$$\begin{aligned} t_i &= 0 && \text{if} \quad n < k \\ t_i &= n - k + 1 && \text{if} \quad k \leq n \leq p_R \\ t_i &= p_R - k + 2 && \text{if} \quad n > p_R \end{aligned}$$

Fig. 21. B-spline basis functions.

B-spline basis functions, $N_i[\sim]$

Relative chordwise position, $\psi[\sim]$

Relative chordwise position, $\psi[\sim]$

$$f(x) = C(x) \cdot S(x) \cdot R(x) = x^{N1}(1-x)^{N2} \cdot \sum_{n=0}^{p_S} b_n B_{n,p_S}(x) \cdot \sum_{n=0}^{p_R} \bar{P}_n N_{n,k}(x)$$

The methods mentioned so far are all used to parameterize two-dimensional shapes, such as airfoils. A three-dimensional shape can be treated as a stack of two-dimensional shapes with the points in between interpolated. This is how most aircraft wings are currently defined. This is an easy solution, also because production can be done in a similar fashion, with ribs representing the airfoil sections. However, the more complex the wing shape, the more airfoil sections have to be defined to describe the wing, rendering the method less efficient.
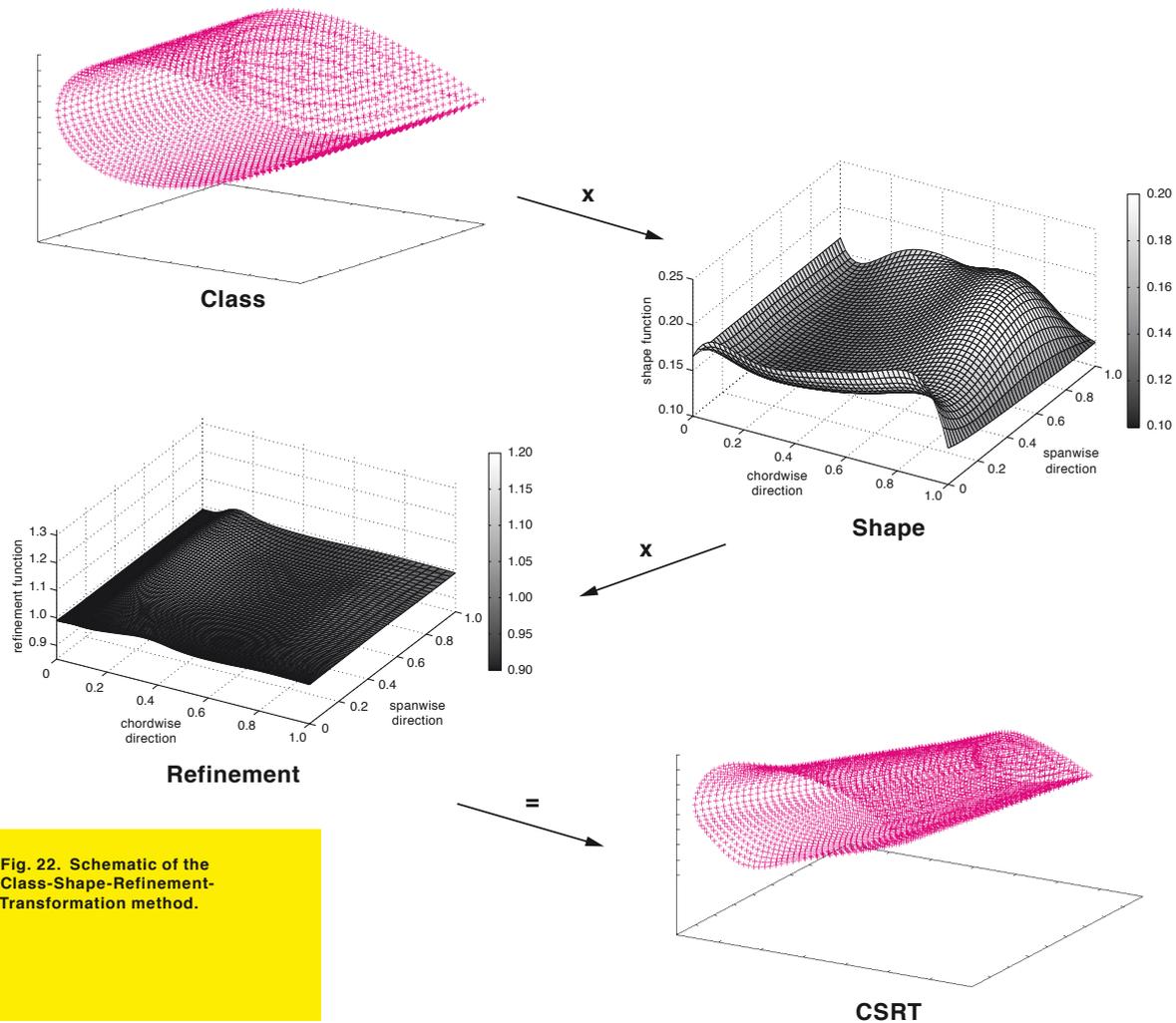

Fig. 22. Schematic of the Class-Shape-Refinement-Transformation method.

With more sophisticated production techniques and computers available, it is now possible to represent the entire three-dimensional shape as a mathematical surface. To do this with the CSRT method, the class, shape and refinement functions will have to represent surfaces instead of curves. For the class function this is straightforward, since it is an analytical function. For the shape and refinement functions this means that Bernstein and B-spline surfaces will have to be used. These require slightly more elaborate computations compared to Bernstein and B-spline curves, but they are well-defined and have the same advantageous properties.

OPTIMIZATION

Once a flow solver and a parameterization technique have been selected, they can be coupled to an optimization algorithm. At the heart of most optimization algorithms lies a so-called sensitivity analysis, which determines the gradients of the objective function with respect to the design variables. In other words, it finds out how the function to be optimized (e.g. lift-to-drag ratio) changes when you change the parameters that determine the shape. This can be done in a number of ways, but most of them require the flow solver to be run once for each gradient. This means that if a shape is parameterized using 100 variables, the flow solver will have to be run 100 times to find all gradients, which can take a very long time. However, there is one technique that can significantly reduce the required computation time: the adjoint equation method[8]. In my work, this technique has been successfully coupled to the CSRT method and to an Euler solver that was developed at TU Delft.

[8] **ADJOINT EQUATION METHOD**

First, let us assume some aerodynamic property J which is a function of the flow variables U and the geometry design variables x:

$$J = J(\mathbf{U}, \mathbf{x})$$

The derivative of J with respect to a specific design variable x_i can be written as:

$$\frac{dJ}{dx_i} = \frac{\partial J}{\partial x_i} + \frac{\partial J}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial x_i}$$

Note that this equation distinguishes between a change in objective function as a result of a variation in the flow solution $\partial \mathbf{U}$ and a variation due to the change in geometry ∂x_i. In order to solve this equation, a relationship between $\mathbf{U}$ and x is needed. Such a relationship is the steady state flow equation, i.e.:

$$\mathbf{R}(\mathbf{U}, x_i) = \mathbf{0}$$

Computing the derivative of $\mathbf{R}$ with respect to x_i gives:

$$\frac{d\mathbf{R}}{dx_i} = \frac{\partial \mathbf{R}}{\partial x_i} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial x_i} = \mathbf{0}$$

The adjoint method can be derived by introducing a vector of Lagrange Multipliers Λ. The steady state flow equation be added as a constraint to the sensitivity to obtain:

$$\frac{dJ}{dx_i} = \frac{\partial J}{\partial x_i} + \frac{\partial J}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial x_i} - \Lambda\left(\frac{\partial \mathbf{R}}{\partial x_i} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\frac{\partial \mathbf{U}}{\partial x_i}\right)$$

$$= \frac{\partial J}{\partial x_i} - \Lambda\frac{\partial \mathbf{R}}{\partial x_i} + \left(\frac{\partial J}{\partial \mathbf{U}} - \Lambda\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)\frac{\partial \mathbf{U}}{\partial x_i}$$

The vector of Lagrange Multipliers can be chosen to satisfy the following adjoint equation:

$$\Lambda\frac{\partial \mathbf{R}}{\partial \mathbf{U}} = \frac{\partial J}{\partial \mathbf{U}}$$

Combining the last two equations results in the elimination of the last two terms and hence:

$$\frac{dJ}{dx_i} = \frac{\partial J}{\partial x_i} - \Lambda\frac{\partial \mathbf{R}}{\partial x_i}$$

Finding a solution to this system only requires solving as many equations as there are flow functionals. For most aerodynamic optimization problems, this number is much lower than the number of design variables. Hence, using the adjoint equation method can dramatically reduce the time required to compute the gradients.

The CSRT method, in two as well as three dimensions, allows for a two-step optimization approach. In the first optimization step, only the Bernstein coefficients of the shape function are used as variables. In the second refinement step, the B-spline coefficients are varied. Typical results indicate that the first optimization step significantly reduces the shockwave on a wing in transonic conditions, increasing its aerodynamic efficiency by about 20-30%. The refinement step usually results in a further improvement in the order of 5%.

Fig. 23. Pressure distribution before (left) and after (right) optimization.
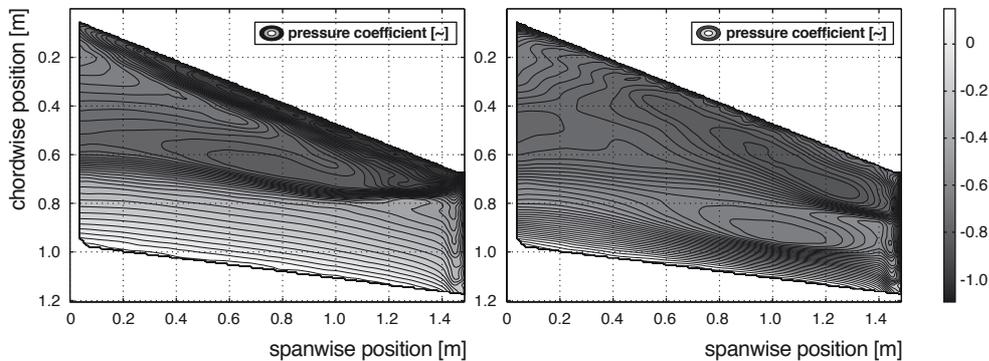
These results can be visualized by looking at the pressure distribution on a wing before and after the optimization process. Putting an ordinary wing in transonic conditions will often lead to shock waves, indicated by strong positive pressure gradients. As a result of these shock waves, most of the lift will be located near the leading edge of the wing, causing an unwanted pitch-up moment. Looking at the optimized wing, a strong reduction of the shock waves can be identified. Additionally, a more even distribution of lift over the entire wing will lead to a lower pitch-up moment.

CONCLUSIONS
The CSRT method developed at CleanEra proved to be a very intuitive and effective way of parameterizing aircraft shapes, both in two as well as in three dimensions. The method allows for a two-step approach which has the potential to significantly increase the lift-to-drag ratio of various aircraft shapes. Using an adjoint algorithm provided the computational efficiency necessary to perform true three-dimensional shape optimization.

Future research will be focused on optimizing the complete design framework and investigating the applicability of the CSRT method to more diverse aircraft shapes.