JAVAFOIL User's Guide

Martin Hepperle

22-December-2017

Contents

JAVAFOIL

JAVAFOIL is a relatively simple program, which uses several traditional methods for the analysis of airfoils in subsonic flow.

The main purpose of JAVAFOIL is to determine the lift, drag and moment characteristics of airfoils. The program will first calculate the distribution of the velocity on the surface of the airfoil. For this purpose it uses a potential flow analysis module which is based on a higher order panel method (linear varying vorticity distribution). This local velocity and the local pressure are related by the Bernoulli equation. In order to find the lift and the pitching moment coefficient the distribution of the pressure can be integrated along the surface.

Next JAVAFOIL will calculate the behavior of the flow layer close to the airfoil surface (the boundary layer). The boundary layer analysis module (a so called integral method) steps along the upper and the lower surfaces of the airfoil, starting at the stagnation point. It solves a set of differential equations to find the various boundary layer parameters. The boundary layer data is then be used to calculate the drag of the airfoil from its properties at the trailing edge.

Both analysis steps are repeated for each angle of attack, which yields a complete polar of the airfoil for one fixed Reynolds number.

Additional tools for the creation and modification of airfoils have been added to fill the toolbox. These tools are wrapped in a Graphical User Interface (GUI) which was designed to be easy to use and not overly complicated. The GUI is organized into a stack of cards, which will be described later.

All calculations are performed by a computer code of my own. JAVAFOIL is neither a rewrite of Eppler' PROFIL nor of Drela's XFOIL program. The boundary layer module is based on the same equations which are also used in the initial version of the Eppler program. Additions include new stall and transition models. The panel method was developed with the help of the extensive survey of panel methods found in [14].

Compared with similar programs, JAVAFOIL can also handle multi-element airfoils and also simulate ground effect.

Limitations

As already noted, JAVAFOIL is a relatively simple program with some limitations. Like with all engineering computer codes, it is up to the user to judge and to decide how far he wants to trust a program.

As JAVAFOIL does not model laminar separation bubbles and flow separation, its results will become inaccurate if such effects occur. The boundary layer method does not include any feedback to the potential flow solution, which means that it is limited to mostly attached flows. Flow separation, as it occurs at stall, is modeled to some extent by empirical corrections, so that maximum lift can be estimated for conventional airfoils. If you analyze an airfoil beyond stall, the results will be quite inaccurate. On the other hand, it is somewhat questionable, whether any two-dimensional analysis method can be used at all in this regime, as the flow field beyond stall becomes fully three dimensional with spanwise flow and strong vortices developing.

In the case of multi element airfoils, one must be aware, that in the real world very complex flows can develop due to interaction of trailing wakes and the boundary layers of the individual elements or if the boundary layers separate locally. An accurate analysis would require a more sophisticated solver for the Navier-Stokes equations, which would also imply an increase in computer time in the order of 1000. Nevertheless a simple tool like JAVAFOIL can be helpful to estimate the main effects and to improve a design to avoid suction peaks and flow separation.

JAVAFOIL's Cards

The user interface of JAVAFOIL is divided into a stack of cards. Each card contains interface elements for a specific task. The content of some cards is also relevant for actions executed on other cards, for example the Mach number specified on the Options card affects the analyses on all other cards.

The Geometry Card

The Geometry card is used to store and prepare the geometry of your airfoil. It contains the "current" or "working" airfoil. The geometry is described by a set of coordinate points, each having an x and a y value. The working airfoil is used and modified by the actions you perform in JAVAFOIL.

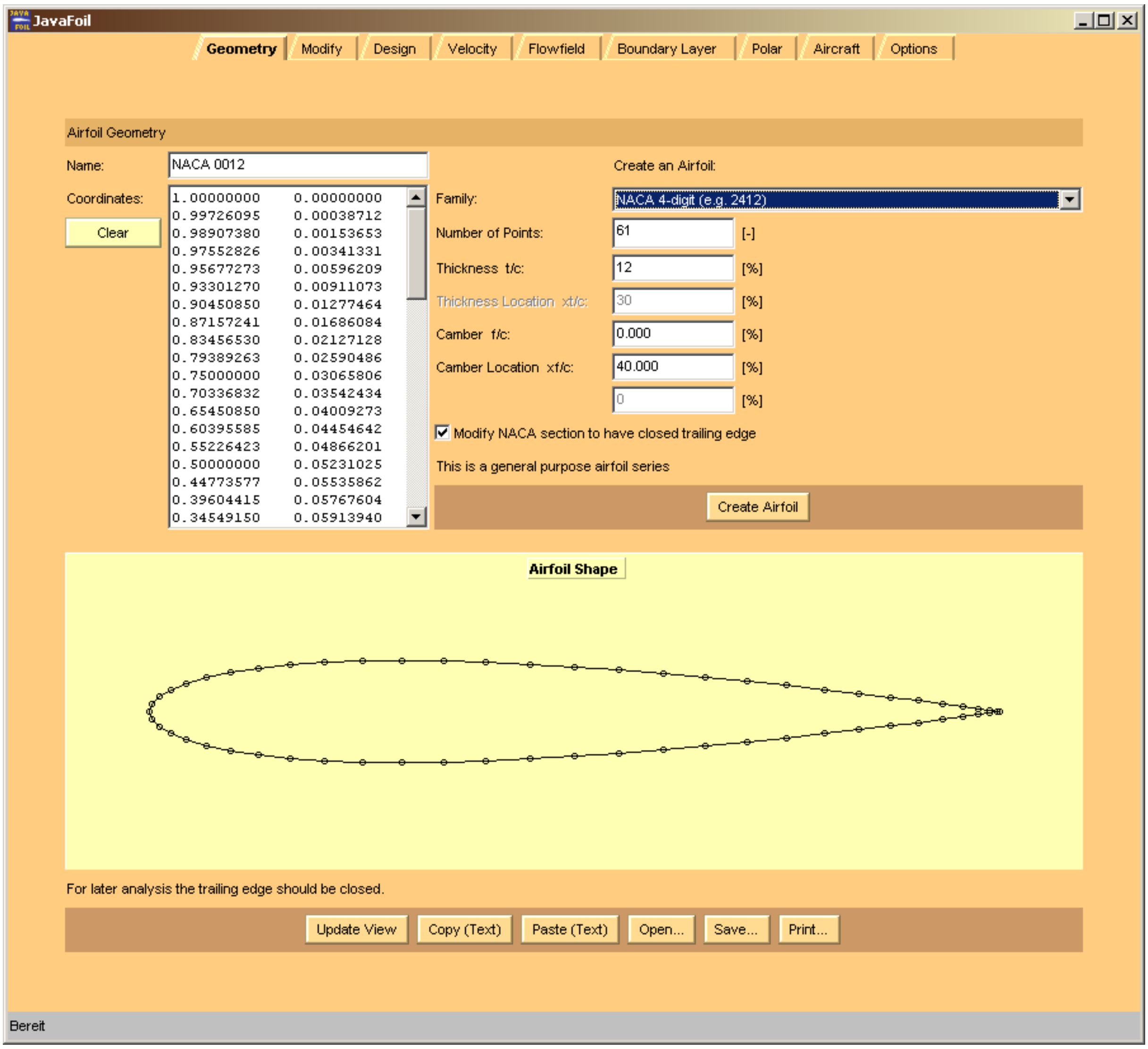
The Geometry card shows a list of x- and y-coordinate pairs and a plot of the resulting airfoil shape. You can enter or paste arbitrary coordinates into this field and press the "Update View" button to copy the coordinates into the working airfoil.

The coordinates must be ordered so that they describe the shape in a continuous sequence. The order must be "trailing edge" → "upper surface" → "nose" → "lower surface" → "trailing edge".

JAVAFOIL comes with a set of shape generators for a variety of airfoils which is accessible from this card. These airfoils represent classical airfoil sections for which analytical descriptions exist (e.g. NACA sections) or which can be constructed from geometrical constraints (e.g. wedge sections). Despite their age, many classical airfoil sections are still applicable to many problems or form a good starting point for new developments.

Today, modern airfoil sections are usually developed for specific purposes and their shapes are usually not published. More recent developments lead towards the direct design of three-dimensional wing shapes, eliminating the classical steps of two-dimensional airfoil design and three-dimensional wing lofting. In most cases, modern airfoil sections are not described anymore by analytical formulas, just by a set of points.

The row of buttons at the bottom allows for copying, saving, loading and printing of airfoil coordinate sets.



Airfoil Geometry

Name: NACA 0012

Coordinates:

```
1.00000000   0.00000000
0.99726095   0.00038712
0.98907380   0.00153653
0.97552826   0.00341331
0.95677273   0.00596209
0.93301270   0.00911073
0.90450850   0.01277464
0.87157241   0.01686084
0.83456530   0.02127128
0.79389263   0.02590486
0.75000000   0.03065806
0.70336832   0.03542434
0.65450850   0.04009273
0.60395585   0.04454642
0.55226423   0.04866201
0.50000000   0.05231025
0.44773577   0.05535862
0.39604415   0.05767604
0.34549150   0.05913940
```

Create an Airfoil:

Family: NACA 4-digit (e.g. 2412)

Number of Points: 61 [-]

Thickness t/c: 12 [%]

Thickness Location xt/c: 30 [%]

Camber f/c: 0.000 [%]

Camber Location xf/c: 40.000 [%]

0 [%]

☑ Modify NACA section to have closed trailing edge

This is a general purpose airfoil series

For later analysis the trailing edge should be closed.

Figure 1: View of JAVAFOIL's Geometry card.

Exporting airfoil geometry

JAVAFOIL can write airfoil geometry to the following file types:

- `*.txt`
  multi-element airfoil geometry in form of simple x-y coordinate sets arranged in two columns. Multi-element airfoils must be separated by a pair of x and y-values each being larger than 999.
- `*.xml`
  multi-element airfoil geometry in JAVAFOIL's hierarchically structured `xml` format.
- `*.dxf`
  multi-element airfoil geometry in AutoCad drawing exchange format. Many CAD programs can read this file format, but the interpretation is not always perfect.
- `*.igs or *.iges`
  multi-element airfoil geometry in Initial Graphics Exchange Standard format. Many CAD programs can read this file format.

Note that JAVAFOIL selects the output file format according to the file name extension.

Importing airfoil geometry

JAVAFOIL can read airfoil geometry from the following file types:

- `*.txt`
  multi-element airfoil geometry in form of simple x-y coordinate sets arranged in two columns. The elements of multi-element airfoils must be separated by a pair of x- and y-values each having a value of 999.0 or larger.
- `*.xml`
  multi-element airfoil geometry in JavaFoil's hierarchically structured xml format.
- `*.png, *.gif, *.bmp, *.jpg`
  single element airfoil geometry from an image file. For details, see next section.

Note that JavaFoil selects the input file format according to the file name extension.

Importing scanned images

You can also load an airfoil from a bitmap image in GIF, PNG, BMP or JPG format. JavaFoil then tries to find an airfoil shape in this image by comparing the image points with the color found in the upper left corner of the image. Therefore the image should have no border, and a monochrome background. Before scanning the image, a smoothing filter is applied to remove spurious points from the image. To achieve acceptable results an image width of 1000 or more pixels is recommended. The interior of the airfoil shape can be empty or arbitrarily filled, because the algorithm searches from the top and bottom edges of the image and stops when it detects the border of the shape. The resulting points are filtered again to improve the smoothness of the shape. Nevertheless the results will not be perfect, but this method can be considered as a last resort to quickly determine airfoil coordinates if only a scanned image is available. It is recommended to inspect the resulting velocity distribution and to use the inverse design method for smoothing the airfoil shape.

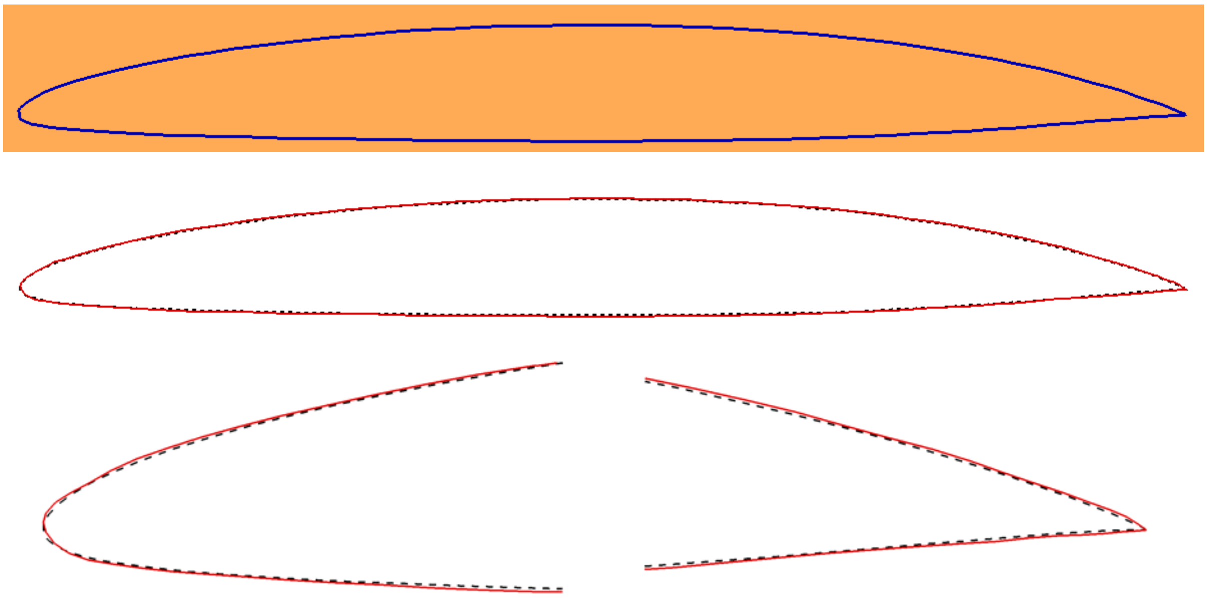


Figure 2: Airfoil image (top) and comparison of the original (dashed) and the reconstructed airfoil shape (solid) using JavaFoil's bitmap import capability on the Geometry card.

JavaFoil's Geometry Generators

Note on NACA airfoils

The construction of the cambered NACA airfoil sections requires that the thickness distribution is erected at right angles to the camber line. Some computer programs do not follow this construction principle and add the thickness just to the y-coordinates of the camber line. This leads to larger deviations from the true airfoil section when the camber line is inclined, e.g. close to the leading edge or close to the trailing edge of airfoils with a high amount of aft camber.

The correct construction method may lead to points extending slightly into the negative x-range, when a large amount of camber is located close to the leading edge. This is a correct behavior and an expected result.

Note also that most NACA sections have a thick trailing edge by definition. In order to produce a thin, sharp trailing edge, JAVAFOIL has an option to close the airfoil shape by bending the upper und lower surfaces to close the trailing edge.

NACA 4-digit airfoils

The calculation of these classical airfoils is easy because their shape and the associated camber lines are defined by rather simple formulas. The maximum thickness is located at $x/c = 30\%$, whereas the maximum camber is typically located at $x/c = 40\%$. See [3] and [4] for more details.

The camber lines are composed of two parabolic arcs, which are joined with equal tangents, but a kink in the curvature. This kink can be seen in the velocity distributions, especially when the position of the maximum camber is different from the common 40% chord station.

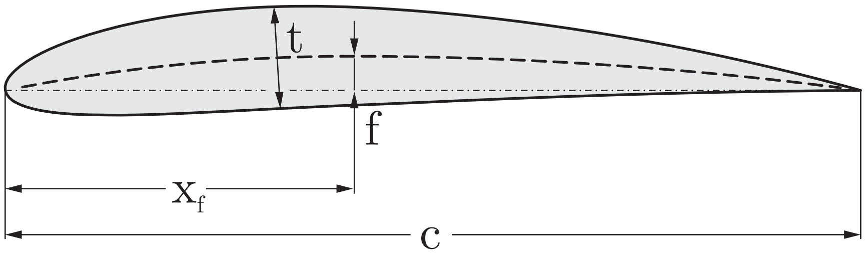


Figure 3: Parameters of NACA 4-digit airfoil sections.

Parameters:

- Free: t/c, f/c, x_f/c
- Fixed $x_t/c = 0.3$

Naming Scheme

The first two integers define the camber line, while the last two integers define the thickness.

- 1st digit: maximum ordinate of camber $100 \cdot f/c$
- 2nd digit: location of maximum camber $10 \cdot x_f/c$
- 3rd and 4th digit: maximum thickness $100 \cdot t/c$

Example:

NACA 2412: 2% camber at 40% chord, 12% thickness

The thickness distribution for the 10% thick section is given by the polynomial:

$$y = \pm\left(0.29690 \cdot \sqrt{x} - 0.12600 \cdot x - 0.35160 \cdot x^2 + 0.28530 \cdot x^3 - 0.10150 \cdot x^4\right)$$

The coefficients of this thickness distribution had been chosen according to the following four constraints [4] (for a 10% thick section):

- maximum thickness occurs at $x/c = 0.3 \rightarrow \frac{\partial y}{\partial x}(0.3) = 0$,

- finite thickness at trailing edge of $\frac{y}{c} = 0.004$,
- finite trailing edge angle at $x / c = 1.0 \rightarrow \frac{\partial y}{\partial x}(1.0) = -0.234$,
- nose shape defined by $y / c = 0.078$ at $x / c = 0.1$.

Modified NACA 4-digit airfoils

The modification adds the position of the maximum thickness as well as the nose radius to the parameter set of the 4-digit series (see [3]).

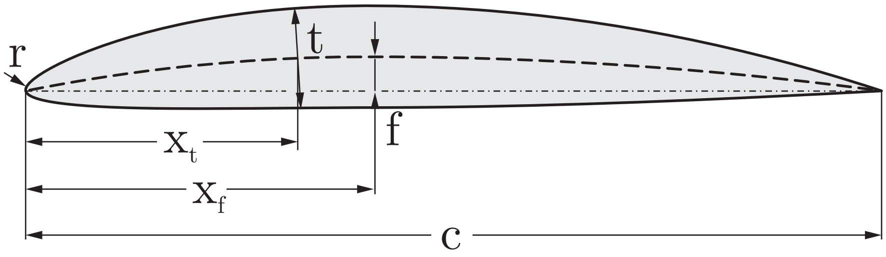


Figure 4: Parameters of the modified NACA 4-digit airfoil sections.

Parameters:

- Free: t / c, f / c, x_t / c, x_f / c, r

Naming Scheme

The name consists of a 4 digit prefix which is identical to the 4-digit series designation, followed by a dash and two additional digits.

- 1st digit: maximum ordinate of camber $100 \cdot f / c$
- 2nd digit: position of the maximum camber $10 \cdot x_f / c$
- 3rd and 4th digit: maximum thickness $100 \cdot t / c$
- dash
- 5th digit: indicates the leading edge radius and is usually one of 0, 3, 6, or 9:
  - 0: sharp leading edge,
  - 3: ¼ normal radius,
  - 6: the normal radius of the 4-digit series,
  - 9: 3 times the normal radius.
- 6th digit: position of the maximum thickness $10 \cdot x_t / c$

Example:
NACA 1410-35: 1% camber at 40% chord, 10% thickness, reduced leading edge radius, maximum thickness at 50% x/c

NACA 5-digit airfoils

These sections use the same thickness distributions as the 4-series, but have new camber lines leading to lower pitching moments. The camber line is composed of a cubic curve in the forward part to which a straight line is attached which extends to the trailing edge. Instead of the camber f / c, a design lift coefficient $C_{\ell\ \mathrm{design}}$ is now used to define the maximum height of the camber line. In practical applications, these airfoils are often used with a maximum camber at $x / c = 0.15$, i.e. relatively far forward.

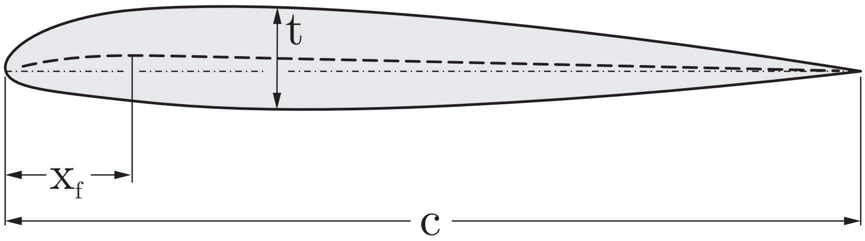


Figure 5: Parameters of NACA 5-digit airfoil sections.

Parameters:

- Free: t/c, x_f/c, $C_{\ell\,\text{design}}$
- Fixed $x_t/c = 0.3$

Naming Scheme

- 1st digit: design $10 \cdot 2/3 \cdot C_{\ell\,\text{design}}$
- 2nd and 3rd digit: $2 \cdot 100 \cdot x_f/c$. Note that the 3rd digit is usually a zero, i.e. the position of the maximum camber is a multiple of 5%.
- 4th and 5th digit: maximum thickness $100 \cdot t/c$

Example:
NACA 23012: design lift coefficient 0.3, maximum camber at 15% chord, 12% thickness.

Modified NACA 5-digit airfoils

The rear part of the camber line of these sections has been modified to a cubic curve which provides a reflexed camber line. Therefore the pitching moments are reduced further or may become even positive.

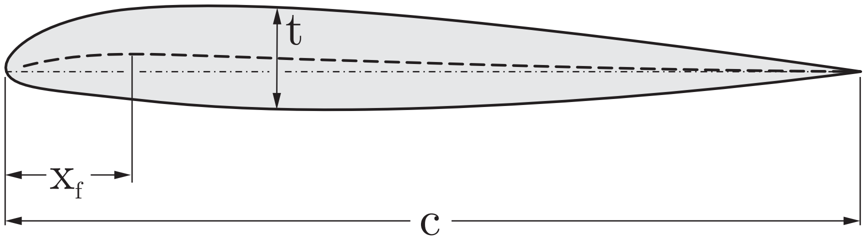


Figure 6: Parameters of the modified NACA 5-digit airfoil sections.

Parameters:

- Free: t/c, x_f/c, $C_{\ell\,\text{design}}$
- Fixed $x_t/c = 0.3$

Naming Scheme

- 1st digit: design $10 \cdot 2/3 \cdot C_{\ell\,\text{design}}$
- 2nd and 3rd digit: $2 \cdot 100 \cdot x_f/c$ plus 1. Assuming that the position of the maximum camber is a multiple of 5% the 3rd digit is always a 1.
- 4th and 5th digit: maximum thickness $100 \cdot t/c$

Example:
NACA 23112: like NACA 23012: design lift coefficient 0.3, maximum camber at 15% chord, 12% thickness, but with a reflexed aft camber line.

NACA 1-series airfoils

The development of these airfoils was aiming at high subsonic speed applications like propellers (see [5]). Their shape was designed with the help of the new (that is, in the 1930s) numerical design methods. JAVAFOIL can create airfoils of the NACA-16 type, which are the only members of the 1-series published by NACA. The maximum thickness and the maximum camber are located at 50% chord, whereas the minimum pressure is reached at 60% of the chord length.

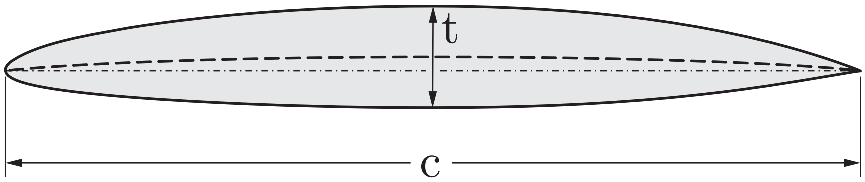


Figure 7: Parameters of NACA 1-series airfoil sections.

Parameters:

- Free: t/c, $C_{\ell\text{ design}}$
- Fixed $x_f/c = 0.5$, $x_t/c = 0.5$

Naming Scheme

- 1st digit “1”: series designation
- 2nd digit: position of minimum pressure of the thickness distribution $10 \cdot x/c$
- a dash
- 3rd digit: $10 \cdot C_{\ell\text{ design}}$
- 4rd and 5th digit: maximum thickness $100 \cdot t/c$

Example:
16-212: 1-series, minimum pressure at 60% chord, design lift coefficient 0.2, 12% thickness.

While these airfoil shapes are not based on analytical expressions, the published coordinates have been approximated to produce an accurate representation of these airfoils. The camber lines used are of the uniform load type (a=1.0, see next section about NACA 6-series airfoils).

NACA 6- and 6A-series airfoils

These airfoils were the first NACA airfoils which had been systematically developed with the inverse design method by Theodorsen. The conformal mapping algorithm was able to deliver a shape for a given pressure distribution. This means that no closed form equations describing the thickness distributions exist.

Earlier JAVAFOIL versions used a very approximate algorithm which had been lifted from the "Digital Datcom" programs, but it was discovered that this produced very inaccurate representations of the 6-series airfoils. Therefore, since version 2.09 (August 2009) JAVAFOIL uses a more elaborate algorithm, which is based on the work of Ladson [6]. This new method is using quite accurate tables of the stream function for most of the 6-series airfoils. JAVAFOIL can generate individual members of the 63, 64, 65, 66 and 67 as well as the 63A, 64A, and 65A families. The "A" modification leads to a less cusped trailing edge region.

The 63, 64, 65, 66 and 67 families can be combined with camber lines of the $a = 0$ to $a = 1$ type.

The 63A, 64A, and 65A sections use a modified $a = 0.8$ camber line which is straight aft of $x / c = 0.8$. The thickness distribution of these airfoils has also been modified to yield straight lines from $x / c = 0.8$ to the trailing edge.

The “a” camber line shapes are specified in terms of the design lift coefficient and the position x/c where the constant loading ends. This is indicated with an additional $a = x.y$ label in the airfoil name.
If you specify $a \geq 1$ in JAVAFOIL’s input, the camber line has a constant loading from leading edge to trailing edge. The resulting airfoils do not carry the “a” label.

Note that officially no intermediate airfoils (e.g. a NACA64.5-012) exist.

Naming Scheme

- 1st digit “6”: series designation
- 2nd digit: chordwise position of minimum pressure of the thickness distribution $10 \cdot x / c$
- single digit suffix following a comma, which is $10 \cdot \Delta C_\ell$. It represents the range ΔC_ℓ above and below $C_{\ell \text{ design}}$ where favorable (accelerating) pressure gradients for laminar flow exist (therefore ΔC_ℓ is approximately the semi-width of laminar bucket)
- a dash
- 3rd digit: design $10 \cdot C_{\ell \text{ design}}$
- 4rd and 5th digit: maximum thickness $100 \cdot t / c$

A camber line shape different from $a = 1.0$ is indicated by the additional designation $a = x.y$, where $x.y$ is replaced by the location x / c where the constant part of the loading ends and the linear drop towards the trailing edge starts..

TsAGI "B" airfoils

The TsAGI (also ZAGI, CAGI) was and is Russia's leading aeronautical research organization. Not much is known about early airfoil development, but the available literature [6], [9] shows that similar to other nations Russia has developed airfoil families based on analytical shape descriptions. The TsAGI series-B is just one such airfoil family. The very simple shape description is using just the maximum thickness. The resulting sections have a reflexed camber line and hence low pitching moment.

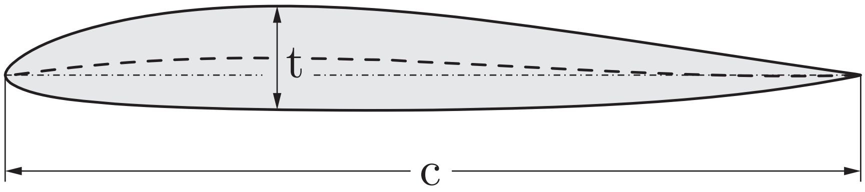


Figure 8: Parameters of TsAGI “B” airfoil sections.

Parameters:

- Free: t / c
- Fixed $x_t / c = 0.3388$, maximum (positive) camber at $x_f / c = 0.3018$, minimum (negative) camber at $x_f / c = 0.9204$.

- The maximum camber is linked to the thickness by the expression $f/c = 0.168 \cdot t/c$.

!!! I am still looking for more information about Russian airfoil developments.

NPL-EC, ECH and EQH airfoils

These British symmetrical airfoil sections are composed of an elliptical forward portion (E) and a cubic (C) or quartic (Q) rear end. The tail closure is built from a hyperbolic curve (H series). The location of the maximum thickness can be varied between 30 and 70% of the chord length. A limited description is contained in [10], [13].

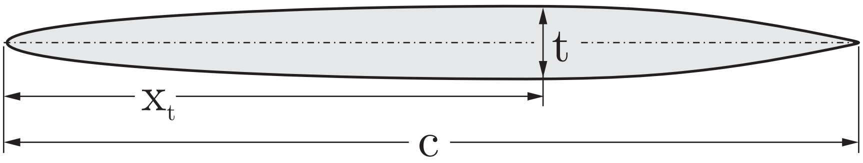


Figure 9: Parameters of NPL airfoil sections.

Parameters:

- Free: t/c, x_t/c

After some reverse engineering, I have used the following assumptions for these airfoils:

- the trailing edge thickness is 2% of the airfoil thickness,
- in case of the "C" and "Q" series the rear end is attached with C0, C1, C2 continuity (position, tangent, curvature) to the elliptic front part,
- in case of the "Q" series the second derivative at the trailing edge is set to -0.2, (this gave the best approximations for 1240 to 1260 airfoils),
- the "H" modification closes the thick trailing edge by a hyperbolic curve which is attached with C0, C1 continuity (position, tangent) to the thickness distribution at $x/c = 0.965$.

Camber lines are 3rd order polynomials which allow to place the location of the maximum camber approximately between 30 and 60% of the chord length.

Note: I am still looking for the "official" description of the airfoil geometry of the EQ and EQH aerofoils, especially how the quartic curve was defined and how the hyperbolic closure was attached to the quartic curve. It seems to be that the procedure to generate these shapes was not published.

Biconvex airfoils

These are symmetrical airfoils, formed by two arcs. They can be represented by the following formula:

$$y = a \cdot \left(x - x^{b}\right)$$

The exponent b can be found from the location of the maximum thickness, i.e. the point where $\partial y / \partial x = 0$

$$x_{t\,max} = \left(\frac{1}{b}\right)^{\frac{1}{b-1}},$$

while the factor a depends on the value of the maximum thickness:

$$t_{max} = 2 \cdot a \cdot \left(x_{t\,max} - x_{t\,max}^{b} \right)$$

If the maximum thickness is placed at $x / c = 0.5$, the airfoil is composed of two equal circular arcs. These airfoils are normally used for application in supersonic flow.

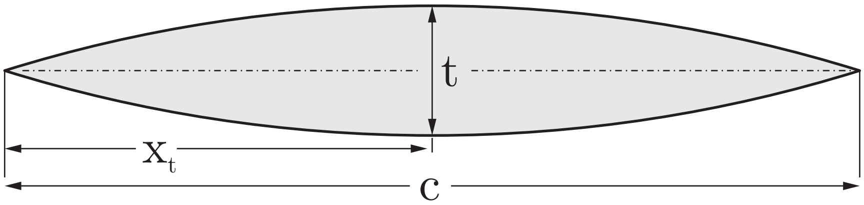


Figure 10: Parameters of biconvex airfoil sections.

Parameters:

- Free: t / c, x_t / c

Double Wedge airfoils

These are symmetrical airfoils composed of straight lines. They are intended for supersonic flow.

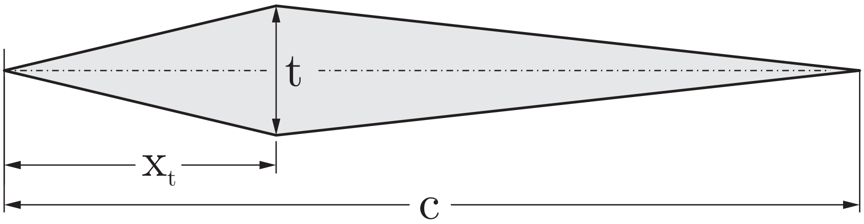


Figure 11: Parameters of double wedge airfoil sections.

Parameters:

- Free: t / c, x_t / c

Plate airfoils

The thickness distribution of these sections represents a plate with a rounded nose and a sharp trailing edge. The nose shape is formed by a so called Cassini curve, which provides a smooth curvature transition to the flat part of the surface. The trailing edge closure is modeled by a cubic parabola. This thickness distribution is superimposed over a NACA 4-series camber line to produce a cambered plate.

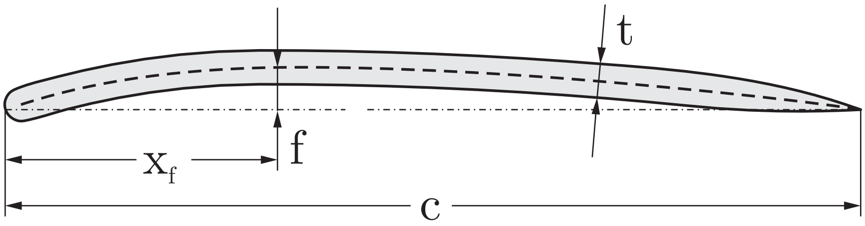


Figure 12: Parameters of cambered plate airfoil sections.

Parameters:

- Free: t/c, f/c, x_f/c
- Fixed: Leading edge shape. The trailing edge closure begins at $x/c = 0.8$

Newman airfoils

These sections consist of a circular nose to which straight tapered tail is attached. It can be manufactured easily, but has a curvature jump at the junction between the nose and the trailing wedge, leading to suction peaks and a risk of flow separation.

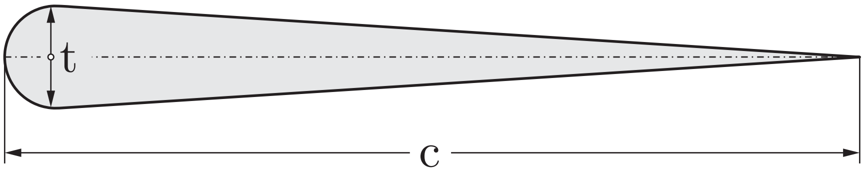


Figure 13: Parameters of Newman airfoil sections.

Parameters:

- Free: t/c

Joukowsky airfoils

These classical airfoil sections are generated by applying a conformal mapping procedure. They were the first practical airfoils developed on a theoretical model. Besides producing the airfoil shape, the mapping procedure was also used to find the flow field around the airfoil as well as the force and the moment acting on the wing section. The airfoils have very thin cusped trailing edges and are therefore difficult to analyze with panel methods and difficult to manufacture.

The conformal mapping is performed using the *Joukowsky* transformation of the complex points z_{circle} on a unit circle with is center at (x_0, y_0).

$$z_{airfoil} = z_{circle} + \frac{\lambda^2}{z_{circle}}, \text{ where } \lambda = -x_0 + \sqrt{1 - y_0^2}\,.$$

In order to match the prescribed airfoil thickness and camber, JAVAFOIL performs an iterative search for the center of the circle. As usual, the resulting coordinates are scaled to unit length.

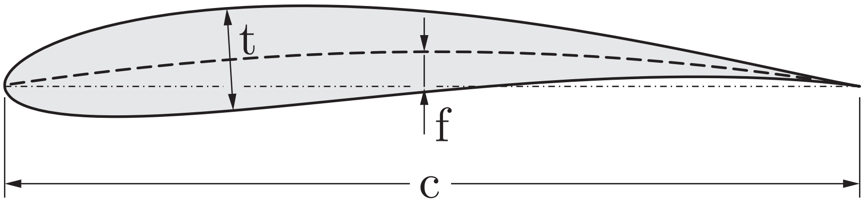


Figure 14: Parameters of Joukowsky airfoil sections.

Parameters:

- Free: t/c, f/c

Van de Vooren airfoils

In contrast to the classical *Joukowsky* airfoils, these airfoils have a finite trailing edge angle. The transformation function is of the type

$$z_{\mathrm{airfoil}} = \frac{\left(1 - z_{\mathrm{circle}}\right)^{k}}{\left(\varepsilon - z_{\mathrm{circle}}\right)^{k-1}} \ .$$

They can be used to create sections with thick trailing edge regions e.g. for fairings. A description of this shape can be found in [14].

Note that not all thicknesses can be achieved for all trailing edge angles; therefore the final maximum thickness may not be what was desired. Also only symmetrical sections are generated in JAVAFOIL.

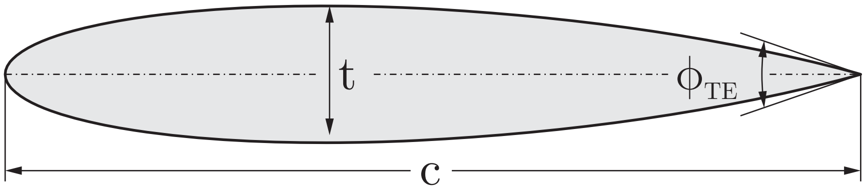


Figure 15: Parameters of Van de Vooren airfoil sections.

Parameters:

- Free: t/c, ϕ_{TE}

Helmbold-Keune airfoils

In the 1940s many attempts were made to extend the then classical NACA airfoil section methodology to more general airfoil shapes. *Helmbold* and *Keune* [15] developed elaborate methods to characterize and parameterize airfoil sections. While the mathematical approach allowed for representation of a wide range of shapes, the methodology was not really successful in these years of manual calculation. Later in the age of numerical shape optimization similar methods have been developed, e.g. the Parsec shape functions.
The parameters of the symmetrical airfoil must be carefully chosen to generate a realistic airfoil shape. The center curvature must be large enough to avoid self-crossing of the outline.

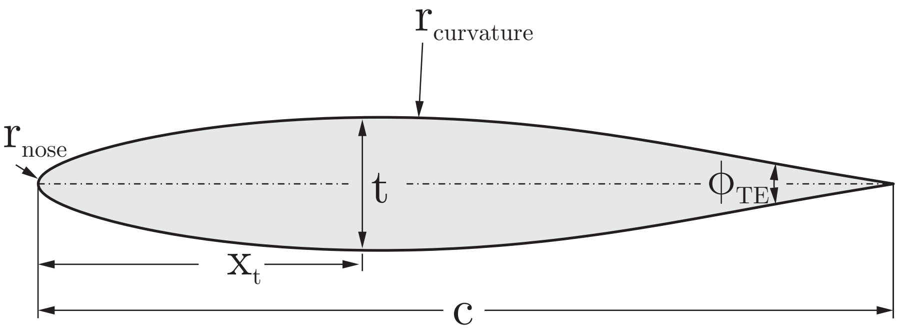


Figure 16: Parameters of Helmbold-Keune airfoil sections.

Parameters:

Free: t/c, x_t/c, trailing edge angle, curvature radius at middle, nose radius.

Roßner airfoils

Another algorithm to generate analytical airfoil shapes based on conformal mapping was published by *Roßner* [16]. Like all methods using conformal mapping, his solution also allowed for the exact analytical determination of the corresponding pressure distributions.

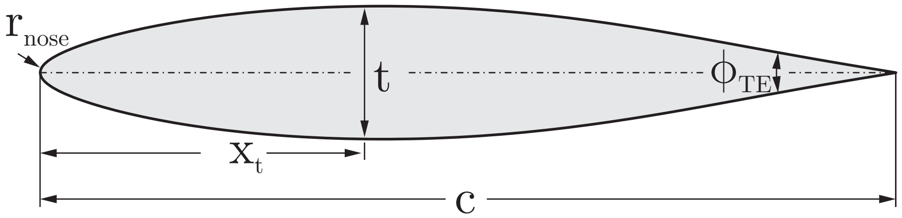


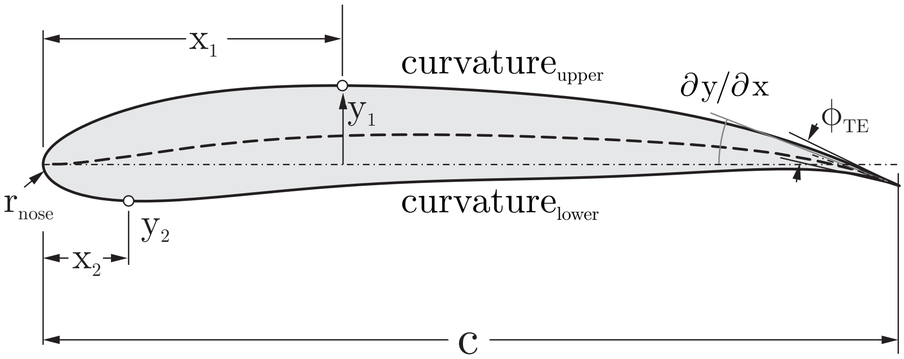
Figure 17: Parameters of Roßner airfoil sections.

Parameters:

Free: t/c, x_t/c, trailing edge angle, nose radius.

Parsec airfoils

The Parsec geometry parameterization was developed by *H. Sobietzky* in the 1990s. It tries to model airfoil shapes by superposition of selected polynomial terms. The parameters resemble the *Helmbold-Keune* approach and are mainly intended to be used for numerical shape optimization. JAVAFOIL implements the so called Parsec-11 formulation which uses 11 parameters. The parameters of the airfoil must be carefully chosen to generate realistic airfoil shapes. The center curvature parameters, the nose radius as well as the trailing edge wedge angle must be carefully adjusted to avoid self-crossing of the outline.



Parameters:

Free: leading edge radius parameter P_1, (P_2, P_3) and (P_4, P_5) the coordinates of control points on the upper and the lower surfaces, P_6 and P_7 curvature control parameters for upper and lower surfaces, P_8 the trailing edge vertical position, P_9 the gradient of the camber line at the trailing edge, P_{10} the trailing edge slenderness, and P_{11} the bluntness angle at the trailing edge.

Horten airfoils

The *Horten* brothers are well known for their development of flying wing airplanes. For most of their wings, they used airfoil sections with a reflexed camber line. These were based on a camber line of low or zero pitching moment (following the thin airfoil theory of *Birnbaum*) to which a thickness distribution was added. A description of these rather simple functions can be found in [17].

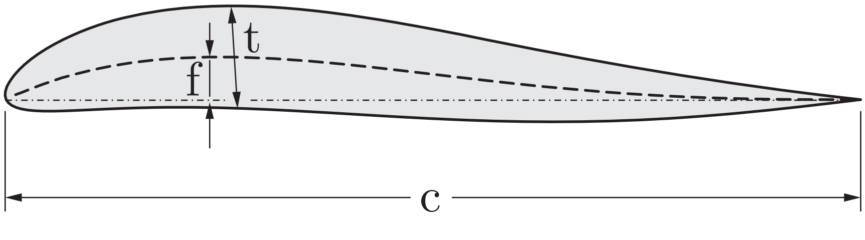


Figure 18: Parameters of Horten airfoil sections.

Parameters:

- Free: t/c, f/c
- Fixed $x_t/c = 0.293$, maximum camber at $x_f/c = 0.25$.

DHMTU airfoils

The stable flight of ground effect vehicles depends on wing planform and airfoil shape. This family of flat bottom airfoils has been developed for this specific application at the Department of Hydromechanics of the Marine Technical University in Saint Petersburg, Russia. Similar to the NACA 4-digit series, the airfoil shape is composed of polynomial segments and a straight lower surface. Their camber line is slightly reflexed and the outline between points 2 and 3 is a straight line segment. A description of the shape can be found in [18].

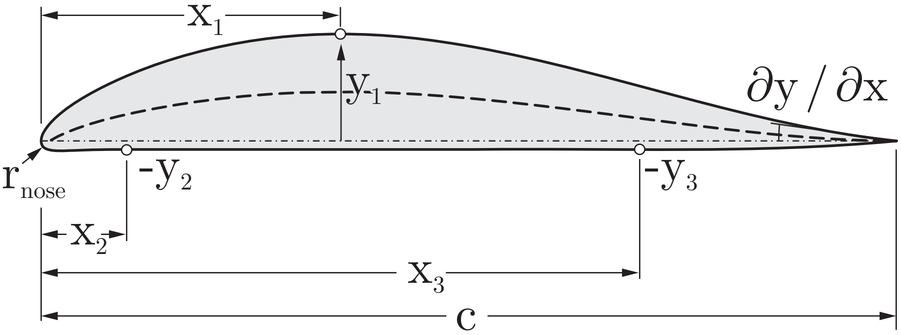


Figure 19: Parameters of DHMTU airfoil sections.

Parameters:

- Free: point x_1/c, y_1/c on the upper and points x_2/c, y_2/c and x_3/c, y_3/c on the lower surface, a nose radius parameter $k = (r/c)/(y_1/c)^2$ and the gradient $\partial y/\partial x$ on upper surface at the trailing edge.

Guderley airfoils

This shape was derived from theoretical considerations of sonic flow and is of mostly academic interest. It is characterized by an accelerating flow with a linear pressure distribution in the forward portion followed by a set of expansion waves. The maximum thickness is located at $x/c = 3/5$. A description of the shape can be found in [26].

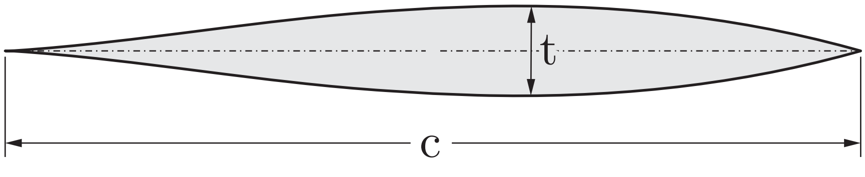


Figure 20: Parameters of Guderley airfoil sections.

Parameters:

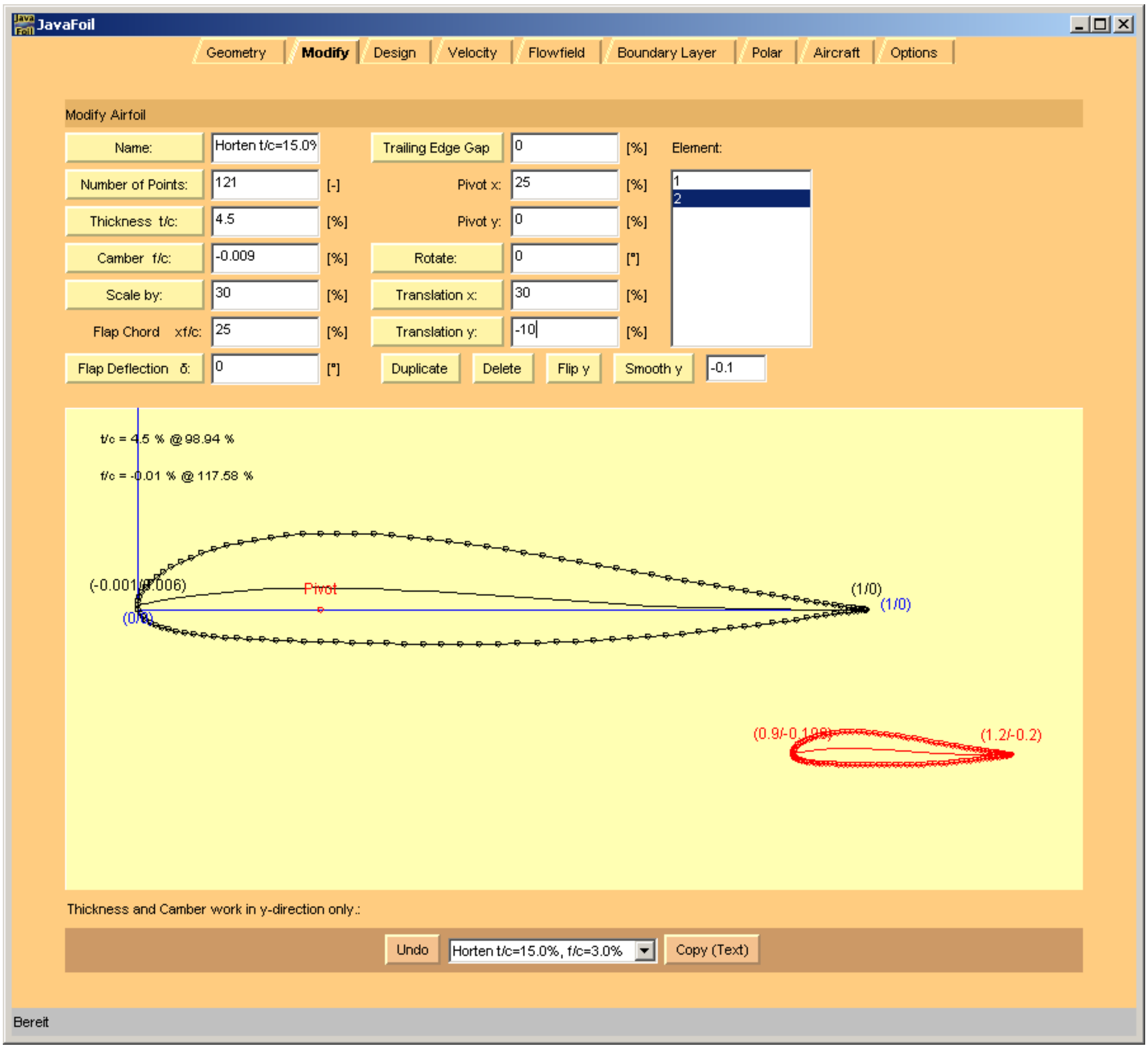
Free: t/c.

The Modify Card

This card can be used to perform various modifications to the airfoil geometry. It consists of an input and action area and a geometry view below.

The modification of parameters is performed by entering new values into a text field and then pressing the button at the left of the text field or pressing the “Enter” key while the focus is still in the text field. Thus it is easy to apply certain operations several times. Any modification will only be applied to the airfoil elements which are currently selected in the “Element” list box.

The geometry view is automatically scaled to fit all airfoil elements. The currently selected elements are highlighted in red.



JavaFoil

Modify Airfoil

Name: Horten t/c=15.0%

Number of Points: 121 [-]

Thickness t/c: 4.5 [%]

Camber f/c: -0.009 [%]

Scale by: 30 [%]

Flap Chord xf/c: 25 [%]

Flap Deflection δ: 0 [°]

Trailing Edge Gap 0 [%]

Pivot x: 25 [%]

Pivot y: 0 [%]

Rotate: 0 [°]

Translation x: 30 [%]

Translation y: -10 [%]

Element: 1, 2 (2 selected)

Smooth y -0.1

t/c = 4.5 % @ 98.94 %

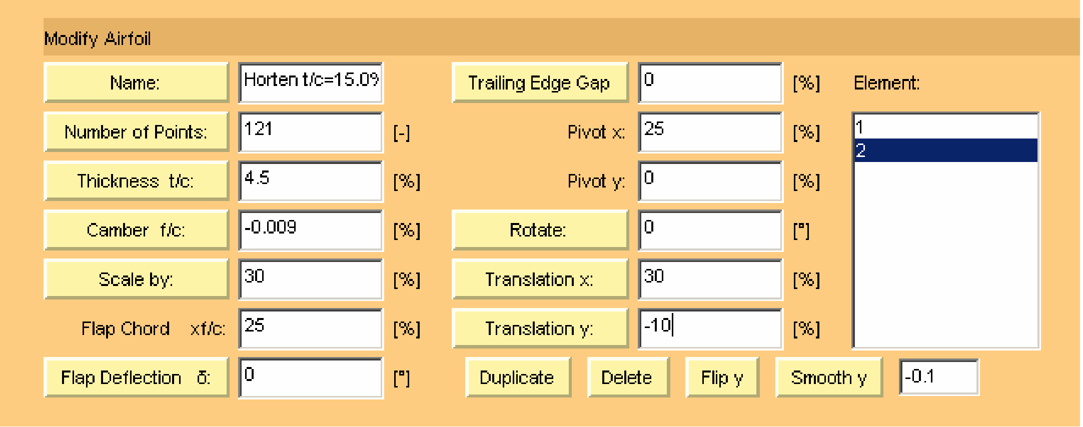
f/c = -0.01 % @ 117.58 %

Thickness and Camber work in y-direction only.:

Horten t/c=15.0%, f/c=3.0%

Bereit

Figure 21: View of the Modify card showing a two-element airfoil with element #2 selected.


Modify Airfoil

Name: Horten t/c=15.0

Number of Points: 121 [-]

Thickness t/c: 4.5 [%]

Camber f/c: -0.009 [%]

Scale by: 30 [%]

Flap Chord xf/c: 25 [%]

Flap Deflection δ: 0 [°]

Trailing Edge Gap 0 [%]

Pivot x: 25 [%]

Pivot y: 0 [%]

Rotate: 0 [°]

Translation x: 30 [%]

Translation y: -10 [%]

Element: 1, 2

Smooth y -0.1

Figure 22: View of the controls on the Modify card.

The following modifications can be performed:

- Name
  Changes the name of the airfoil
- Number of Points
  Changes the number of coordinate points of the selected element(s).
- Thickness
  Scales the thickness of the selected element(s) by decomposing the shape into a thickness distribution and a camber line. Only the thickness distribution is scaled, so that the camber line is maintained. Note that small changes to the camber may occur due to numerical errors.
- Camber
  Scales the camber line to a new height. This works only if the airfoil is already cambered. Scaling the camber line of a symmetrical airfoil accomplishes nothing.
- Scale by
  Scales the airfoil shape by multiplying the coordinates with the given scaling factor.
- Flap deflection
  Modifies the coordinates by deflecting a plain flap of the given chord length. The axis of rotation is always the middle between upper and lower surface.
- Trailing Edge Gap
  Modifies the shape so that the prescribed trailing edge gap is produced. Generally it is recommended to use closed trailing edges for analysis, except if the airfoil is extremely thin towards the trailing edge. This function can also be applied before exporting airfoil shapes suitable for manufacturing.
- Rotate
  Rotates the selected airfoil element(s) around the specified Pivot point.
- Translate x
  Moves the selected airfoil element(s) by the given distance horizontally.
- Translate y
  Moves the selected airfoil element(s) by the given distance vertically.
- Duplicate
  Creates a copy of the currently selected element(s). Note that you have to move the new element from its initial location so that it is not overlapping with other elements.
- Delete
  Deletes the selected element(s)

- FLIP Y
  Reflects the selected elements across a horizontal line passing through the pivot point.
- SMOOTH Y
  This command uses the smoothing factor specified in the text field to the right of the button. Currently it supports two smoothing variants:
  If the smoothing factor is positive, the coordinates are approximated by a smoothing spline curve. A reasonable smoothing factor is 0.1.
  If the smoothing factor is negative, a filter is applied to the y-coordinates to reduce waviness. This filter applies a weighted average to each point and its two neighbor points. If for example the smoothing factor is -0.1, the y coordinate of the smoothed point is 90% of its initial value and 10% of the linear interpolation between the two neighboring points according to:

  $$y_i \leftarrow (1-f)\cdot y_i + f\cdot\left(y_{i-1}+\left(y_{i+1}-y_{i-1}\right)\cdot\frac{s_i - s_{i-1}}{s_{i+1}-s_{i-1}}\right)$$

  This filter can be applied several times, but subsequent application will also smooth out details like a pointed airfoil nose.

You can also modify individual points by dragging them up or down with the left mouse button depressed. This modification method is restricted to movements in the y-direction. If you need more freedom, you have to modify the numerical coordinate values on the Geometry card.

The COPY (TEXT) command button at the bottom of the card copies the airfoil geometry to the clipboard with the following data:

- a table with the x-y coordinates, similar to the copy on the GEOMETRY card, but additionally with the local curvature 1/r, and
- a second table with the coordinates of the camber line and the thickness distribution as reconstructed from the airfoil shape.

Note concerning multi-element airfoils
Modifications are applied only to the airfoil element(s) selected in the “Elements” list box. The selection is also used by other cards. Only selected elements are taken into account when total force, moment and drag coefficients are determined.

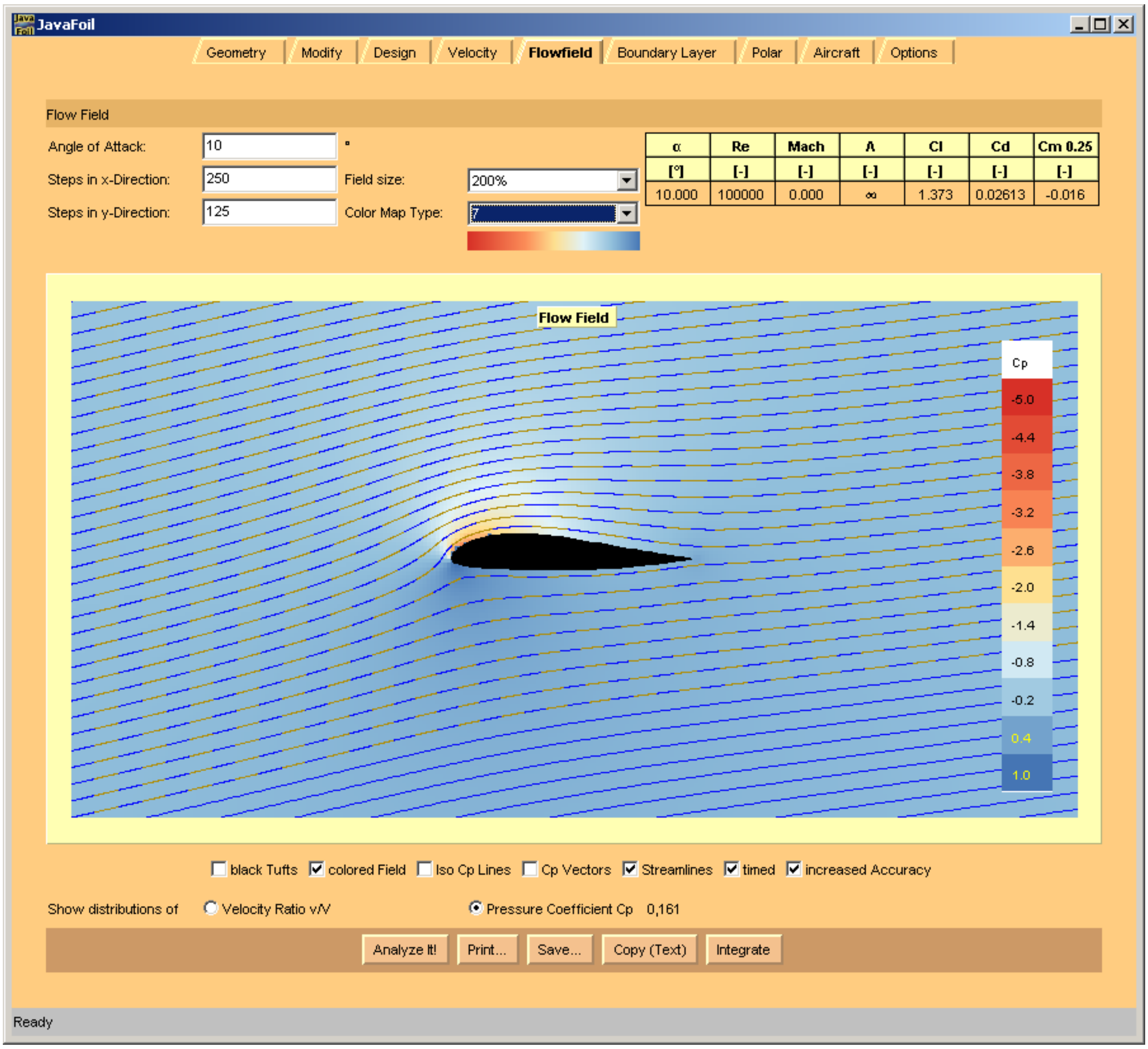
The Flowfield Card

This card is intended to visualize the flow around the airfoil in various ways. Basically the “Analyze It!” command first performs an analysis of the airfoil for the given angle of attack. The results are presented in form of the global coefficients in the table. In order to be consistent with the display on the Boundary layer card, these results include friction using parameter taken from the boundary layer card (Reynolds number and transition location) as well as from the Polar card (stall model).

Then the local velocity over a rectangular grid of points is calculated. This calculation uses the vorticity distribution on the surface and neglects friction. Therefore you will not see flow separation or a viscous wake behind the airfoil.
It is possible to display either the ratio of the local velocity to the freestream velocity v/V or the local pressure coefficient.

Moving the mouse pointer over the colored field shows the corresponding pressure coefficient or velocity ratio at the bottom of the screen.

- ANALYZE IT! Performs the analysis of the flow field and if selected the integration of the path of streamlines. Note that the classical Runge-Kutta scheme used to integrate streamlines with increased accuracy can take quite some time. Standard accuracy uses a fast but simple forward stepping Newton algorithm which introduces larger errors in regions of high curvature. Progress is indicated in the status line.
- COPY (TEXT) copies the field data to the clipboard in tabular format suitable for plotting with the Tecplot software. Remember that you can copy or export the picture using the context menu of the graph window.
- INTEGRATE uses the momentum equation to integrate the momentum and pressure field along a circular path with a radius of 50 around the airfoil. As we neglect friction and hence follow d'Alembert's thoughts, the result should produce zero drag, but a lift coefficient close to what we obtain from the integration of the surface pressure. Remember that the lift coefficient given in the table includes friction and the effect of the stall model, so that both results can only agree if the Reynolds number is rather high. The result is displayed in a message box and also copied to the clipboard.



Flow Field

Angle of Attack: 10 °

Steps in x-Direction: 250

Field size: 200%

Steps in y-Direction: 125

Color Map Type: 7

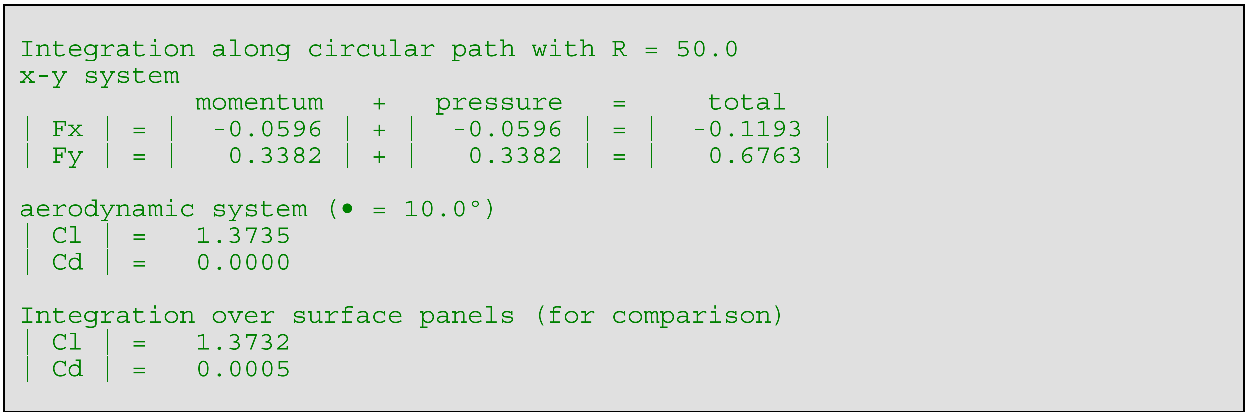
α	Re	Mach	Λ	Cl	Cd	Cm 0.25
[°]	[-]	[-]	[-]	[-]	[-]	[-]
10.000	100000	0.000	∞	1.373	0.02613	-0.016

□ black Tufts ☑ colored Field □ Iso Cp Lines □ Cp Vectors ☑ Streamlines ☑ timed ☑ increased Accuracy

Show distributions of ○ Velocity Ratio v/V ● Pressure Coefficient Cp 0,161

Ready

Figure 23: View of the Flowfield card showing an airfoil with times streamlines.



```
Integration along circular path with R = 50.0
x-y system
           momentum   +   pressure   =     total
| Fx | = |  -0.0596 | + |  -0.0596 | = |  -0.1193 |
| Fy | = |   0.3382 | + |   0.3382 | = |   0.6763 |

aerodynamic system (• = 10.0°)
| Cl | =   1.3735
| Cd | =   0.0000

Integration over surface panels (for comparison)
| Cl | =   1.3732
| Cd | =   0.0005
```

Figure 24: Result of momentum and pressure integration over a circular path around the airfoil. Compare the lift coefficient with the value obtained from surface pressure integration (The stall model was set to “none” on the Polar card). The total force is the result of the change of the momentum passing through the control volume and the pressure acting on its surface. Note that the pressure part is very important, even if the integration boundary is rather far away from the airfoil.

The Aircraft Card

This card is similar to the Polar card but is intended to be used to analyze the airfoil under conditions with are close to the application on aircraft or hydrofoil wings. It is assumed that the wing has to carry a certain load (the weight of the aircraft) at all flight speeds. In order to produce the same lift the lift coefficient of the airfoil section at low speed must be higher than at high speed. Thus lift coefficient and flight speed (and hence Reynolds number) depend on each other.

To establish the relations using aircraft design parameters we start with the definition of the lift coefficient C_L for the complete wing

$$C_L = \frac{g}{\frac{\rho_\infty}{2}\cdot v_\infty^2}\cdot\frac{m}{S}.$$

Solving the definition of the Reynolds number with the chord length c

$$Re = \frac{v_\infty \cdot c}{\nu}$$

for the flight speed v_∞ and inserting into the lift coefficient gives µs

$$C_L = \frac{g}{\frac{\rho_\infty}{2}\cdot\left(\frac{Re\cdot\nu}{c}\right)^2}\cdot\frac{m}{S}.$$

Solving for the Reynolds number

$$Re = \frac{c}{\nu}\cdot\sqrt{\frac{2\cdot g}{\rho_\infty\cdot C_L}\cdot\frac{m}{S}}$$

Yields the desired relation between lift coefficient and Reynolds number as it is seen by and aircraft with a rectangular wing planform. The design parameters directly related to the aircraft and its wing are wing loading m/S and chord length c.

The Panel Method

JAVAFOIL implements a classical panel method to determine the linear potential flow field around single and multi-element airfoils. In JAVAFOIL the airfoil surfaces carry a linearly varying vorticity distribution. This is the same type of distribution as used in *XFOIL* but simpler than the higher order (parabolic) distribution used in Eppler's *PROFIL* code. The resulting equation system consists therefore of a (# of panels +1)² sized matrix and two right hand sides. These are for 0° and 90° angle of attack and can be solved efficiently at the same time for the two corresponding vorticity distributions. The vorticity distribution for any arbitrary angle of attack is then derived from these two solutions (remember that potential theory is linear and allows for superposition). There is no interaction with the boundary layer, as in *XFOIL*, though.

For a shape discretization by N panels, the equation system of this classical panel method consists of the matrix of influence coefficients, the unknown circulation strength at each panel corner point and the two right hand side vectors. These represent the "no flow through the surface" conditions for 0° and 90° angle of attack. Each coefficient $C_{i,j}$ reflects the influence of the triangular vorticity distribution due to the vortex strength γ_i at each corner point on the center point of each panel j. The last row contains the tangential flow condition at the trailing edge (the "Kutta-condition"). This is needed to obtain a solution which is compatible with the experience that the flow normally separates smoothly at the sharp trailing edge. Note that this assumption will not be correct when large regions of flow separation occur.

$$\begin{bmatrix} C_{1,1} & \cdots & C_{N+1,1} \\ \vdots & \ddots & \vdots \\ C_{1,N} & \cdots & C_{N+1,N} \\ 1 & \cdots 0 \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} \gamma_{1,0^\circ} & \gamma_{1,90^\circ} \\ \gamma_{2,0^\circ} & \gamma_{2,90^\circ} \\ \vdots & \vdots \\ \gamma_{N+1,0^\circ} & \gamma_{N+1,90^\circ} \end{bmatrix} = \begin{bmatrix} RHS_{1,0^\circ} & RHS_{1,90^\circ} \\ RHS_{2,0^\circ} & RHS_{2,90^\circ} \\ \vdots & \vdots \\ RHS_{N+1,0^\circ} & RHS_{N+1,90^\circ} \end{bmatrix}$$

Like with most panel methods the solution time for the system of linear equations increases with the square of the number of unknowns. Therefore it is advisable to limit the number of points to values between 50 and 150. This relatively small number already yields sufficient accuracy of the results (in contrast to more complex CFD methods for solving the Navier-Stokes equations, where you may need several 100 points on the airfoil surface and many more points to fill the space around the shape).

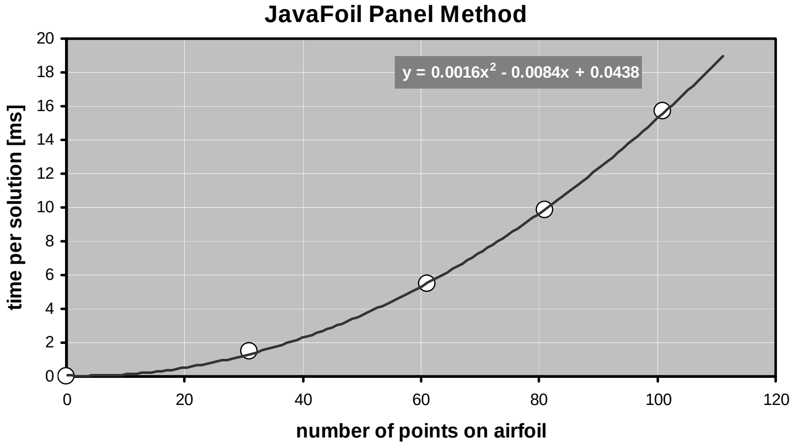


Figure 25: Graph of the solution time versus number of points on the airfoil (Pentium 4, 2.4 GHz).

Boundary Layer Analysis

The boundary layer analysis module implements an integral boundary layer integration scheme following publications by Prof. Richard Eppler. Such integral methods are based on differential equations describing the growth of boundary layer parameters depending on the external local flow velocity. These equations are then integrated starting at the stagnation point. While accurate analytical formulations are available for laminar boundary layers, some empirical correlations are needed to model the turbulent part.

Note: the local skin friction coefficient as given on the Boundary Layer card is twice the value as used by Eppler to follow the more common convention $C_f = \tau_0 / \left(\frac{\rho}{2} \cdot v_\infty^2\right)$.

In JAVAFOIL there is no interaction between the boundary layer and the external flow, as in *XFOIL*, though. Therefore largely separated flows cannot be analyzed – a short flow separation ($s_{separated} / c < 10\%$) at the trailing edge does not affect the results very much. Also laminar separation bubbles are not modeled; when laminar separation is detected the code switches to turbulent flow.

Transition Criteria

Methods to predict transition from laminar to turbulent flow have been developed by many authors since the early days of Prandtl’s boundary layer theory. While it is possible to analyze the stability of a boundary layer numerically, all methods which are practical and fast are more or less approximate and rely on empirical relations (usually derived from experiments). Because the local boundary layer parameters at a station s are the result of an integration process starting at the stagnation point, they contain a “history” of the flow.

Local Criteria

Many methods predict transition by applying a criterion based on local boundary layer parameters. These criteria are based on relations, which can be evaluated at any station along the surface. They do not need an extra integration of some instability parameter, but of course are affected by the “history” of the flow. Most of these criteria are relating Re_{δ_2} to the shape of the boundary layer profile.

Eppler

Transition is assumed to occur when $Re_{\delta_2} \geq e^{18.4 \cdot H_{32} - 21.74 - 0.36 \cdot r}$.

Eppler enhanced

Transition is assumed to occur when $Re_{\delta_2} \geq e^{18.4 \cdot H_{32} - 21.74 + 125 \cdot (H_{32} - 1.573)^2 - 0.36 \cdot r}$.

Michel (1)

This simple criterion assumes transition to occur when $Re_{\delta_2} \geq 1.535 \cdot Re_s^{0.444}$.

Michel (2)

Transition is assumed to occur when $Re_{\delta_2} \geq 1.174 \cdot \left(1 + 22400 / Re_s\right) \cdot Re_s^{0.46}$. See [24].

H_{12}-Re_s

Transition is assumed to occur when $2.1<H_{12}<2.8$ and

$\log_{10}(\mathrm{Re}_s) \geq -40.4557 + 64.8066 \cdot H_{12} - 26.7538 \cdot {H_{12}}^2 + 3.3819 \cdot {H_{12}}^3$. See [25].

Criteria based on a region of instability, $n - \mathrm{Re}_{\delta_2}$ envelopes

These methods first determine a local point of instability and then begin at this point to integrate a measure for the amplification of instability.

Drela approximates the envelopes of the amplification rate n versus Re_{δ_2} by straight lines of the form $\tilde{n} = f(\mathrm{Re}_{\delta_2}, H_{12})$. Two versions of this approximation were used in his codes of the *XFOIL* and *MSES/ISES* family.

The approximation is expressed by

$$\tilde{n} = \frac{\partial \tilde{n}}{\partial \mathrm{Re}_{\delta_2}} \cdot \left(\mathrm{Re}_{\delta_2} - \mathrm{Re}_{\delta_2,\mathrm{crit}}\right).$$

Transition can occur when $\mathrm{Re}_{\delta_2} > \mathrm{Re}_{\delta_2,\mathrm{crit}}$ and $\tilde{n} > n_{\mathrm{crit}}$. In JAVAFOIL transition is assumed to occur when the value $n_{\mathrm{crit}} = 9 - r$ is exceeded.

Drela, XFOIL 1.1 and 5.4

$$\frac{\partial \tilde{n}}{\partial \mathrm{Re}_{\delta_2}} = 0.01 \cdot \sqrt{\left(2.4 \cdot H_{12} + 2.5 \cdot \tanh(1.5 \cdot H_{12} - 4.65) - 3.7\right)^2 + 0.25}$$

$$\log_{10}\left(\mathrm{Re}_{\delta_2,\mathrm{crit}}\right) = \left(\frac{1.415}{H_{12} - 1} - 0.489\right) \cdot \tanh\left(\frac{20}{H_{12} - 1} - 12.9\right) + \frac{3.295}{H_{12} - 1} + 0.44$$

These approximations can be found in [1] and [2].

Drela, XFOIL 5.7

Modification in 1991

$$\frac{\partial \tilde{n}}{\partial \mathrm{Re}_{\delta_2}} = 0.028 \cdot (H_{12} - 1) - \frac{0.0345}{e^{-\left(\frac{3.87}{H_{12}-1} - 2.52\right)^2}}$$

$$\log_{10}\left(\mathrm{Re}_{\delta_2,\mathrm{crit}}\right) = 0.7 \cdot \tanh\left(\frac{14}{H_{12} - 1} - 9.24\right) + 2.492 \cdot \left(\frac{1}{H_{12} - 1}\right)^{0.43} + 0.66$$

Drela, XFOIL 6.8

only a tiny modification (term 0.66 → 0.62)

$$\frac{\partial \tilde{n}}{\partial \mathrm{Re}_{\delta_2}} = 0.028 \cdot (H_{12} - 1) - \frac{0.0345}{e^{-\left(\frac{3.87}{H_{12}-1} - 2.52\right)^2}}$$

$$\log_{10}\left(\mathrm{Re}_{\delta_2,\mathrm{crit}}\right) = 0.7 \cdot \tanh\left(\frac{14}{H_{12} - 1} - 9.24\right) + 2.492 \cdot \left(\frac{1}{H_{12} - 1}\right)^{0.43} + 0.62$$

Method of Arnal:

A set of tables produced by D. Arnal has been approximated by W. Würz with polynomials:

$$\frac{\partial \tilde{n}}{\partial \mathrm{Re}_{\delta_2}} = a_1 + a_2 \cdot H_{12} + a_3 \cdot H_{12}^2$$

$$\log_{10}\left(\mathrm{Re}_{\delta_2,\mathrm{crit}}\right) = b_1 + b_2 \cdot H_{12} + b_3 \cdot H_{12}^2$$

Here the envelope is not a straight line as in Drela's method. For details see [21].
In JAVAFOIL transition is assumed to occur when the value $n_{crit} = 9 - r$ is exceeded.

Method of Granville

This method is not described here. It also works by integrating a stability parameter starting from a point of instability.

Abbreviations:

approximation of *n*	$\tilde{n}$
roughness factor (0 = smooth)	r
displacement thickness	δ_1
momentum thickness	$\delta_2 = \theta$
shape factor displacement thickness / momentum thickness	$H_{12} = \frac{\delta_1}{\delta_2}$
Reynolds number based on local momentum thickness	$\mathrm{Re}_{\delta_2} = \mathrm{Re}_\theta$
Reynolds number based on local arc length	Re_s

Effect of Roughness

The effect of roughness on transition and drag is complex and cannot be simulated accurately. Even modern resource hungry direct numerical simulation methods have difficulties to simulate the effect.
In JAVAFOIL two effects of surface roughness are modeled:

- laminar flow on a rough surface will be destabilized leading to premature transition,
- laminar as well as turbulent flow on rough surfaces produce a higher skin friction drag.

The effect on toughness is modeled in the following transition models

Eppler Standard	Transition is assumed to occur when $\mathrm{Re}_{\delta_2} \geq e^{18.4 \cdot H_{32} - 21.74 - 0.36 \cdot r}$.
Eppler enhanced	Transition is assumed to occur when $\mathrm{Re}_{\delta_2} \geq e^{18.4 \cdot H_{32} - 21.74 + 125 \cdot (H_{32} - 1.573)^2 - 0.36 \cdot r}$.
Drela, e^n approx.	Transition is assumed to occur when the value $n_{crit} = 9 - r$ is exceeded.
Arnal (Würz)	Transition is assumed to occur when the value $n_{crit} = 9 - r$ is exceeded.

The global effect on drag is taken into account by a simple scaling of the total drag coefficient

$$C_d = C_d \cdot (1 + r/10)$$

The roughness factor r is meant to represent the following surface conditions

$r = 0$	perfect smooth surface as for example on a composite material sailplane wing

$r = 1$	smooth, but slightly rough surface as for example a painted cloth surface
$r = 2$	similar to the NACA standard roughness
$r = 3$	dirty surface with spots of dirt, bugs and flies

Note that the NACA standard roughness is usually applied to the leading edge only. It consists of a sparse (5-10% of the area) leading edge coating up to 8% x/c. The grain size is about 0.45‰ of the chord length. Thus for a wing chord length of 1m the grain size would be 0.45mm.

Stall Corrections

Empirical Stall Correction #1 („CalcFoil")

```
if ($\alpha > 0$ )
{
        // handle separation on upper surface
        // drag increment
        $C_{d,\text{ upper}} = C_{d,\text{ upper}} + \left|\sin^2\alpha \cdot \left(x_{TE} - x_{\text{sep, upper}}\right)^2 + 0.025 \cdot \cos\alpha \cdot \left(x_{TE} - x_{\text{sep, upper}}\right)^2\right|$
        // lift multiplier reduces lift linearly with length of separated length
        $C_\ell = C_\ell \cdot \left(1 - 0.2 \cdot \left(x_{TE} - x_{\text{sep, upper}}\right)\right)$
}
else if ($\alpha < 0$ )
{
        // handle separation on lower surface
        // drag increment
        $C_{d,\text{ lower}} = C_{d,\text{ lower}} + \left|\sin^2\alpha \cdot \left(x_{TE} - x_{\text{sep, lower}}\right)^2 + 0.025 \cdot \cos\alpha \cdot \left(x_{TE} - x_{\text{sep, lower}}\right)^2\right|$
        // lift multiplier reduces lift linearly with length of separated length
        $C_\ell = C_\ell \cdot \left(1 - 0.2 \cdot \left(x_{TE} - x_{\text{sep, lower}}\right)\right)$
}

// moment multiplier
$C_{m,\text{ corrected}} = C_{m,\text{ panel method}} \cdot 0.9 \cdot x^2_{\text{sep, lower}} \cdot x^2_{\text{sep, upper}}$
```

// lift multiplier due to suction peak criterion

$$C_\ell = C_\ell \cdot \frac{1}{\left(\frac{\Delta C_{P,\text{ max}}}{20}\right)^2 + 1}$$

, where $\Delta C_{P,\text{ max}}$ is the difference between the minimum pressure coefficient close to the nose of the airfoil and the pressure close to the trailing edge.

Empirical Stall Correction #2 („Eppler")

```
if ($\alpha > 0$)
{
        // handle separation on upper surface
```

```
    if ( $x_{\text{sep, upper}} < x_{\text{TE}}$ )
    {
        // trailing edge angle of upper surface
        $\theta_{\text{TE}} = \arctan\left(-\frac{y_{\text{sep, upper}} - y_{\text{TE}}}{x_{\text{sep, upper}} - x_{\text{TE}}}\right)$
    }
    else
    {
        $\theta_{\text{TE}} = 0$
    }
    // drag increment
    $C_{\text{d, upper}} = C_{\text{d, upper}} + 0.2 \cdot \sin\left(\alpha + \theta_{\text{TE}}\right) \cdot \left(x_{\text{TE}} - x_{\text{sep, upper}}\right)^2$

    $\Delta C_\ell = C_{\text{l, max, fudge}} \cdot \left(\alpha + \theta_{\text{TE}}\right) \cdot \pi \cdot \left(x_{\text{TE}} - x_{\text{sep, upper}}\right)$

    if ( $\Delta C_\ell > 0$ )
    {
        // lift reduction
        $C_\ell = C_\ell - \Delta C_\ell$
    }
    else
    {
        // lift multiplier
        $C_\ell = C_\ell \cdot \left(1 - \sin\alpha \cdot \left(x_{\text{TE}} - x_{\text{sep, upper}}\right)\right)$
    }

    // moment increment
    $C_m = C_m - \sin\alpha \cdot \left(x_{\text{TE}} - x_{\text{sep, upper}}\right) \cdot \left(0.5 \cdot \left(1 + x_{\text{sep, upper}}\right) - 0.25\right)$
}
else if ( $\alpha < 0$ )
{
    // handle separation on lower surface
    if ( $x_{\text{sep, lower}} < x_{\text{TE}}$ )
    {
        // trailing edge angle of lower surface
        $\theta_{\text{TE}} = \arctan\left(-\frac{y_{\text{sep, lower}} - y_{\text{TE}}}{x_{\text{sep, lower}} - x_{\text{TE}}}\right)$
    }
    else
    {
        $\theta_{\text{TE}} = 0$
    }
    // drag increment
```

```
        $C_{d,\text{ lower}} = C_{d,\text{ lower}} - 0.2 \cdot \sin\left(\alpha + \theta_{TE}\right) \cdot \left(x_{TE} - x_{\text{sep, lower}}\right)^2$

        $\Delta C_\ell = C_{l,\text{ max, fudge}} \cdot \left(\alpha + \theta_{TE}\right) \cdot \pi \cdot \left(x_{TE} - x_{\text{sep, lower}}\right)$

        if ( $\Delta C_\ell < 0$ )
        {
                // lift reduction
                $C_\ell = C_\ell - \Delta C_\ell$
        }
        else
        {
                // lift multiplier
                $C_\ell = C_\ell \cdot \left(1 - \sin\alpha \cdot \left(x_{TE} - x_{\text{sep, lower}}\right)\right)$
        }

        // moment increment
        $C_m = C_m - \sin\alpha \cdot \left(x_{TE} - x_{\text{sep, lower}}\right) \cdot \left(0.5 \cdot \left(1 + x_{\text{sep, lower}}\right) - 0.25\right)$
}
```

// lift multiplier due to modified suction peak criterion

$C_\ell = C_\ell \cdot \dfrac{1}{\left(\dfrac{\Delta C_{P,\text{ max}}}{30}\right)^2 + 1}$, where $\Delta C_{P,\text{ max}}$ is the difference between the minimum pressure coefficient close to the nose of the airfoil and the pressure close to the trailing edge.

Compressible Flow

JavaFoil analyzes airfoils in incompressible flow, which means low Mach numbers as they are common in model aircraft of general aviation airplanes. In practical application this means Mach numbers below $M = 0.25$. It is possible however to extend the Mach number range somewhat by applying compressibility corrections to the incompressible results. This is only possible, as long as the flow speed is subsonic all over the surface of the airfoil and compressibility effects are small.

Critical Pressure Coefficient

The character of the flow changes dramatically when sonic speed is exceeded anywhere on the surface. The pressure coefficient associated with sonic speed is called “critical pressure” coefficient ($C_{p,\text{ crit}}$). In most cases pressure recovery from supersonic to subsonic speeds (from $C_p < C_{p,\text{ crit}}$ to $C_p > C_{p,\text{ crit}}$) is leading to an abrupt recompression with a shock. The analysis of such flows requires more complex methods than implemented in JavaFoil. Such methods must be capable of handling compressible flows (for example by solving the full, compressible potential equations or by solving the Euler equations).

In order to indicate how close the local flow is to supersonic speeds, JAVAFOIL calculates the critical pressure coefficient if a Mach number is specified on the Options card. The critical limit is drawn as a wavy line in the graph on the Velocity card. Additionally, a compressibility correction is applied to the incompressible solution to model first order compressibility effects. Note however, that the theory becomes invalid, when flow reaches or exceeds sonic speed.

In JAVAFOIL, the critical pressure coefficient is calculated from the relation

$$\mathrm{C}_{\mathrm{p,\,crit}} = \frac{2}{\kappa \cdot \mathrm{M}_\infty^2} \cdot \left(\left(\frac{2}{\kappa + 1} \cdot \left(1 + \frac{\kappa - 1}{2} \cdot \mathrm{M}_\infty^2 \right) \right)^{\frac{\kappa}{\kappa - 1}} - 1 \right)$$

In terms of the velocity ratio the critical limit is found from

$$\left(\frac{\mathrm{v}}{\mathrm{v}_\infty} \right)_{\mathrm{crit}} = \sqrt{1 + 2 \cdot \frac{1 + \mathrm{M}_\infty^2}{(\kappa + 1) \cdot \mathrm{M}_\infty^2}}$$

Compressibility Corrections

There are different ways to correct incompressible flow results for compressibility effects. One should keep in mind that these are only corrections – they can never produce the correct physical effects when the flow locally reaches or exceeds supersonic speed. Therefore the applicability of all compressibility corrections is limited to cases where the local flow velocity (which can be much higher than the onset flow velocity) is well beyond the speed of sound. In practical application one can use such corrections well up to about $\mathrm{M} = 0.5$, the error grows very rapidly when the onset Mach number exceeds 0.7.

In JAVAFOIL, the incompressible panel analysis is always performed for the given airfoil – the shape is never geometrically distorted. The compressibility correction is applied later to the local surface pressure according to the Kármán-Tsien approximation

$$C_{P,c} = \frac{2 \cdot C_{P,i} \cdot \left(1 + \sqrt{1 - M_\infty^2}\right)}{2 \cdot \left(1 + \sqrt{1 - M_\infty^2} - M_\infty^2\right) + M_\infty^2 \cdot C_{P,i}} \quad .$$

The corrected pressure coefficient is then used to calculate the lift and moment coefficients.

Finite Wings in *JAVAFOIL*

In the 1920s it has been found by Prandtl and also by Lanchester that the finite span of wings affects their aerodynamic performance. They found that the effects could be expressed as a function of the aspect ratio (a.k.a. “slenderness” or “finesse”) of the wing. Prandtl’s “Lifting Line” theory was developed and successfully applied to design wings up to the 1940s and even today it is useful for unswept wings of relatively high aspect ratio ($\Lambda > 5$). The aspect ratio can be determined from $\Lambda = \mathrm{b} / \ell = \mathrm{b}^2 / \mathrm{S}$ (span b divided by the mean chord length ℓ or span squared divided by wing area S). The main result of this theory is that the airfoil drag is increased by an additional drag force (“induced drag” a.k.a. “vortex drag”) which is caused by the finite wing span and the associated wake downwash behind the wing. It is physically unavoidable when a wing produces lift. The vortex drag coefficient of a wing can be expressed by $\mathrm{C}_{\mathrm{D,\,induced}} = \mathrm{k} \cdot \mathrm{C}_\mathrm{L}^2 / (\pi \cdot \Lambda)$, where C_L is the lift coefficient of the whole wing and k is a factor to account for the shape of the lift force distribution along the span (for good high aspect ratio wing designs and for very low aspect ratios $k \approx 1$). Linked to the downwash is also a loss of lift at the same angle of attack and a reduction of the pitching moment.

Now, JAVAFOIL is a program for the analysis of two dimensional airfoils. Nevertheless it supports a very simple model of finite wings to allow for a more realistic comparison of airfoils. When the user supplies a value for the aspect ratio on the Options card classical wing theory formulas are used to determine an approximation of the 3D effects on lift, drag and pitching moment. These effects can applied to the polars produced by JAVAFOIL and make it possible to get a first impression of the relations between induced drag and airfoil drag. For example the importance of the airfoil drag is diminishing for higher lift coefficients and lower aspect ratios.
These three dimensional corrections can also be applied to the results for constant Reynolds number (Polar card) as well as more realistically for the results associated with a constant wing loading (Aircraft card).

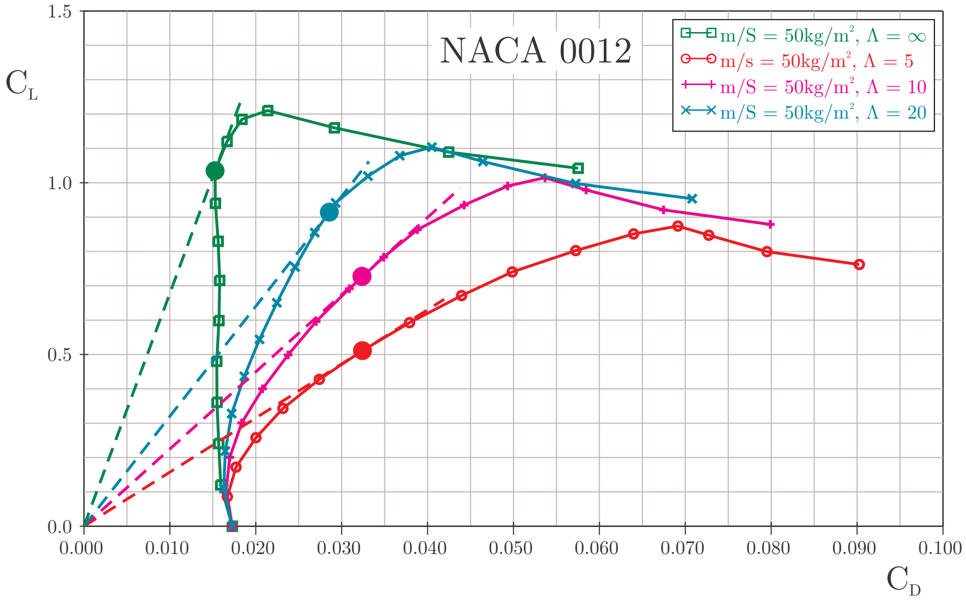


Figure 26: Lift versus drag coefficient polars for a NACA 0012 airfoil and wings of different aspect ratio.

The graph above shows the effect of the wing aspect ratio on lift over drag coefficient. Starting with infinite aspect ratio (aspect ratio = 0 on the Options card) three wings with increasing aspect ratio have been analyzed. For each curve the maximum of the lift over drag (L/D) ratio is indicated by a filled circle. It can be clearly seen, that depending on the aspect ratio the additional induced drag distorts the polar so that the optimum L/D ratio is shifted to lower lift coefficients. While the two dimensional airfoil achieves its maximum of L/D at slightly above $C_\ell = 1.0$, the low aspect ratio wing of $\Lambda = 5$ requires to operate the airfoil at $C_\ell = 0.5$ because this is the optimum C_L of the whole wing. If we compare with another airfoil we would better compare the airfoils at the lift coefficients corresponding to the wing aspect ratios.

Note that the results as shown above are accurate for a wing having an elliptical lift distribution and an elliptical, untwisted planform. Due to the spanwise lift distribution on a generic wing, the airfoils along the span of the wing will operate somewhat above and below the total lift coefficient of the wing. To analyze such effects requires a more sophisticated three dimensional wing analysis code (e.g. lifting line, vortex lattice or panel methods). Also no additional wing effects (like Reynolds number variation due to taper) are taken into account.

Polars for Constant Wing Loading

Airfoil data has traditionally been presented in form of graphs and tables for constant Reynolds numbers. This form results from the typical way wind tunnel experiments and numerical analyses are conducted. In a wind tunnel it is relatively easy to maintain a constant wind speed and Reynolds number.
Now the lift coefficient of a real airplane depends on the speed because the wing loading is usually fixed during flight – flying at low lift coefficients results in high speeds (and high Reynolds numbers) and vice versa. Therefore the operating points during flight would slice through a set of polars having constant Reynolds numbers.
It is possible to create polars more closely related to the conditions during flight. This would require adjusting the wind speed to each lift coefficient, which is cumbersome and expensive in a wind tunnel, but feasible in a numerical tool like JAVAFOIL. Here you can use the Aircraft card to calculate polars for a given wing loading.

Abbreviations:

mass of aircraft	m	kg
gravity constant	g	m/s^2
density of medium	ρ_∞	m/s^2
kinematic viscosity	ν	m^2/s
flight speed	v_∞	m/s
wing area	S	m^2
chord length	c	m
Reynolds number	Re	-

Basic Equations

The definition of the lift coefficient is $C_L = \frac{m \cdot g}{\frac{\rho_\infty}{2} \cdot v_\infty^2 \cdot S}$. Solving the definition of the Reynolds number $Re = \frac{v_\infty \cdot c}{\nu}$ for the velocity v_∞ yields $v_\infty = \frac{Re \cdot \nu}{c}$. Inserting this result into the definition of the lift coefficient produces

$$C_L = \frac{m \cdot g \cdot c^2}{\frac{\rho_\infty}{2} \cdot Re^2 \cdot \nu^2 \cdot S} .$$

Solving for the Reynolds number yields

$$Re = \frac{c}{\nu} \cdot \sqrt{\frac{2 \cdot g}{\rho_\infty \cdot C_L} \cdot \frac{m}{S}} .$$

Note that this equation can also be written $Re \cdot \sqrt{C_L} = \frac{c}{\nu} \cdot \sqrt{\frac{2 \cdot g}{\rho_\infty} \cdot \frac{m}{S}}$, which means that we can also calculate polars of constant $Re \cdot \sqrt{C_L}$ to match a given aircraft.

Using these results one can derive an aircraft oriented airfoil polar for a given wing loading $\frac{m}{S}$ and given mean chord length c. Due to the dependency between lift coefficient and Reynolds number an iterative calculation procedure is used:

- prescribe the environmental condition density ρ_∞ and kinematic viscosity ν.
- prescribe a wing loading $\frac{m}{S}$ and a reference chord length c.
- perform the following calculation sequence:

```
inital value
$Re^* = 10^6$
for ($\alpha = \alpha_0$ to $\alpha_1$ step $\Delta\alpha$)
{
        iterate
        {
                $Re = Re^*$
                $C_L = f(\alpha, Re)$
                $Re^* = \sqrt{\frac{g}{\frac{\rho_\infty}{2} \cdot C_L \cdot \nu^2} \cdot \frac{m}{S} \cdot c^2}$
        }
        while$\left(\frac{\left|Re^* - Re\right|}{Re} > \varepsilon\right)$
}
```

Note that the result still is an airfoil polar, even if wing loading and chord length are involved. Only when you additionally specify an aspect ratio on the options card, the polars include the induced drag and approximate a finite wing.

A precaution must be undertaken to handle cases where $C_L \to 0$. Here JAVAFOIL limits the Reynolds number to a value corresponding to a small lift coefficient, e.g. $C_L = 0.02$.

Note: One can also derive the Reynolds number for a constant ratio $\frac{m}{\Lambda}$, eliminating the chord length c. This has not been implemented in JAVAFOIL as it was considered more abstract to think in terms of $\frac{m}{\Lambda}$ instead of the aircraft design parameters $\frac{m}{S}$ and c. But as the relation is $\frac{m}{S} \cdot c^2 = \frac{m}{\Lambda}$ it would be sufficient to use $\frac{m}{\Lambda}$ instead of $\frac{m}{S}$ in JAVAFOIL while setting $c = 1$.

Lift Correction for given Aspect Ratio and Mach number

For a given angle of attack, a 3D wing of finite aspect ratio produces less lift than the 2D airfoil section, which corresponds to an infinite aspect ratio. Another correction has to be applied when the Mach number is larger than zero. In subsonic flight more lift is produced when the Mach numbers is increased.
The 3D wing correction is applied only if you specify a value for the aspect ratio of the wing $\Lambda = b^2 / S$ (span bb and wing area S) on the Options card.
The following correction is applied to the lift coefficient of a 2D airfoil C_ℓ in order to approximate the lift coefficient C_L of the 3D wing in compressible flow. The correction is divided into two regimes of aspect ratios.

For small aspect ratios ($\Lambda < 4$) the following formula is used:

$$C_L = \frac{C_\ell}{\sqrt{1 - M_\infty^2 + \left(\frac{2 \cdot \pi}{\Lambda \cdot \pi}\right)^2} + \frac{2 \cdot \pi}{\Lambda \cdot \pi}}$$

If the aspect ratio is larger, $\Lambda \geq 4$, the simplified approximation is applied:

$$C_L = \frac{C_\ell}{\sqrt{1 - M_\infty^2} + \frac{2 \cdot \pi}{\Lambda \cdot \pi}}$$

Implementation in *JavaFoil*

```
public final static double LiftForAspectRatio(double dCl,
                                              double dAspectRatio,
                                              double dMachNumber)
    {
        double dReturn = dCl;

        // correction for finite wings
        if (dAspectRatio > 0.1)
        {
            // Source: Anderson, "Aircraft Performance and Design"
            // lift gradient reduction factor
            // a 0 / (pi*AR)
            double dGradientRatio = 2.0 * Math.PI / (Math.PI *
                                    dAspectRatio);
            if (dAspectRatio < 4.0)
            {
                // low aspect ratio, compressible (Anderson [2.18b])
                dReturn /=
                        (Math.sqrt(1.0 - Math.pow(dMachNumber, 2.0) +
                                   Math.pow(dGradientRatio, 2.0)) +
                         dGradientRatio);
            }
            else
            {
                // high aspect ratio, compressible (Anderson [2.16])
                dReturn /=
                        (Math.sqrt(1.0 - Math.pow(dMachNumber, 2.0)) +
                         dGradientRatio);
            }
        }
        return (dReturn);
    }
```

Moment Correction for given Aspect Ratio and Mach number

The pitching moment of a 3D wing of finite aspect ratio is reduced due to the loss of pressure difference towards the wing tips. Again, this 3D wing correction is applied only if you specify a value for the aspect ratio of the wing on the Options card.

The following correction is applied to the pitching moment coefficient of the 2D airfoil C_m in order to approximate the moment coefficient C_M of the 3D wing:

$$C_M = C_m \cdot \frac{\Lambda}{\Lambda + 4}$$

This relation has been determined by a series of vortex lattice analyses of rectangular wings.

Drag Correction for given Aspect Ratio and Mach number

After the lift coefficient of the 2D airfoil for a given angle of attack α has been corrected to the effect of the 3D wing, an approximation of the induced drag is added to the airfoil drag (for the same angle of attack α). Like the lift correction this correction is only applied if you specify a value for the aspect ratio of the wing $\Lambda = b^2 / S$ (span b and wing area S) on the Options card.
As no information about the real wing shape is available, the assumption of having a “good” wing planform is made. Therefore the induced drag component is calculated by using the classical formula derived in the lifting line theory (Prandtl).

$$C_{D,i} = k \cdot \frac{C_L^2}{\pi \cdot \Lambda}$$

In JavaFoil the “k-Factor” is assumed to be 1.0 (planar wing with elliptical lift distribution).

Note that the idea of these simple corrections is to give you a feeling for the relative importance of the induced drag in relation to the airfoil drag only. For real wing design you should use a more appropriate 3D aerodynamic analysis tool, e.g. a vortex lattice or panel method.

Implementation in *JavaFoil*

```
    public final static double DragForAspectRatio(double dCd, double dCl,
                                                  double dAspectRatio,
                                                  double dMachNumber)
    {
        double dReturn = dCd;

        if (dAspectRatio > 0.1)
        {
            // add the induced drag of finite wing according to Prandtl
            dReturn += dCl * dCl / (Math.PI * dAspectRatio);
        }

        return (dReturn);
    }
```

Note that all finite wing results are only approximations. If you need more accurate results, you must use a 3D wing analysis code, which ideally can also handle friction effects.

Swept Wings in *JavaFoil*

While JavaFoil is and remains a tool for analyzing two-dimensional wing sections, it is capable of analyzing an airfoil section as part of a swept wing. This is no replacement for the three-dimensional analysis of the three-dimensional wing using appropriate tools, but it helps to understand the main effects of wing sweep on the velocity respectively pressure distribution. You have to keep in mind, that a three dimensional wing has wing tips, one or more kinks in the spanwise sweep angle distribution and that it may be tapered. Also the boundary layer on a three-dimensional swept wing deviates from the ideal two-dimensional models used in JavaFoil.

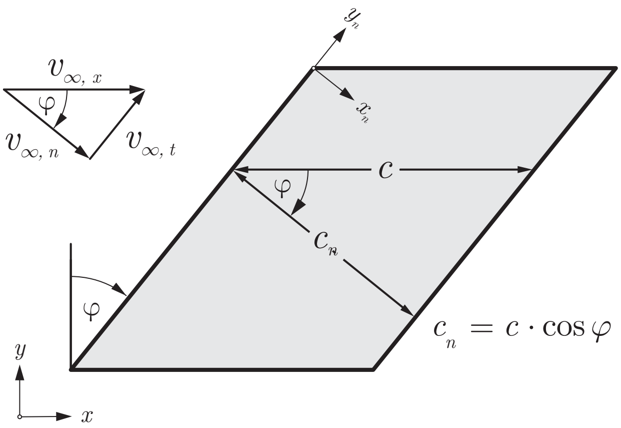
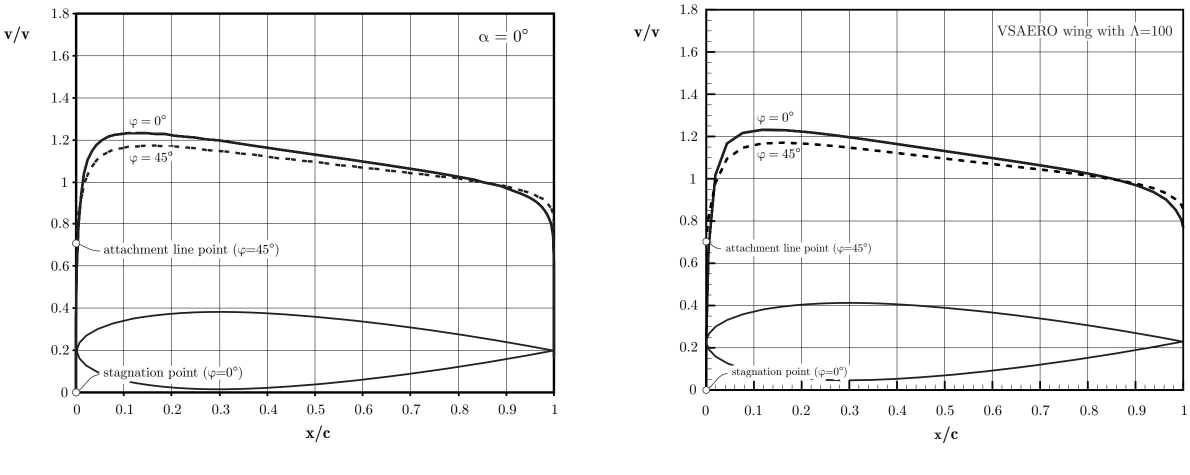


Figure 27: Top view of a swept wing with sections in the $x - z$ and in the normal plane $x_n - z$.

JAVAFOIL assumes that the wing has no taper and that it is infinitely long. In practical application, these assumptions are not too far from the conditions at mid-span of a low taper ratio wing having an aspect ratio of 10 or higher.

Note that JAVAFOIL can predict the velocity or pressure distribution over the section quite accurately, but all of its transition criteria are not made for swept wings. Typically wing sweep introduces additional disturbances due to crossflow and spanwise flow at the leading edge of the wing, which both may lead to premature transition. Therefore the drag coefficients should not be taken too serious when the sweep angle is larger than, say, 20° except if transition is fixed close to the leading edge.

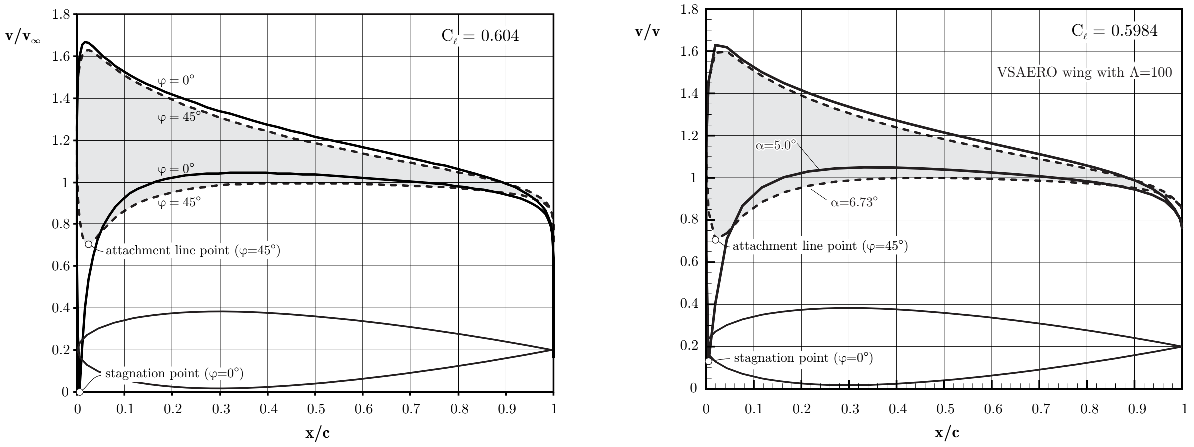
In order to analyze an airfoil on a swept wing we align the defining wing section with the streamwise direction (this is also called a “sheared” wing planform). The sweep angle must be specified on the Options card and then the analyses on all other cards (including the Design card) take this angle into account. Following classical sweep theory as devised by the German aerospace engineer Busemann in the 1930s, JAVAFOIL internally analyzes the airfoil section at right angles to the leading edge at an increased angle of attack and combines the result with the tangential flow past the wing. The resulting velocity resp. pressure distribution matches perfectly with the result of a three-dimensional analysis produced with a 3D-panel method. The following figures show a comparison between JAVAFOIL and VSAERO.



a) Results of two dimensional analysis with sweep correction.

b) Results of three dimensional analysis.

Figure 28: Velocity ratio past a NACA 0015 airfoil for sweep angles of 0° and 45° at $\alpha = 0°$. Left: two dimensional analysis; right: three dimensional analysis of a finite wing of very high aspect ratio.



a) Results of two dimensional analysis with sweep correction.

b) Results of three dimensional analysis.

Figure 29: Velocity ratio past a NACA 0015 airfoil for sweep angles of 0° and 45° and the same lift coefficient as obtained at $\alpha = 5^\circ$. Left: two dimensional analysis, data see Table 1; right: three dimensional analysis of a finite wing of very high aspect ratio.

φ	α	C_ℓ	C_d	$C_{m\,0.25}$	C_p^*	$M_{crit.}$
[°]	[°]	[-]	[-]	[-]	[-]	[-]
0	5.00	0.604	0.01710	-0.008	-1.787	0.564
45	6.72	0.604	0.01611	-0.009	-1.650	0.579

Table 1: Two dimensional analysis results for a NACA 0015 airfoil at same lift coefficient.

As can be seen in the table above, the swept wing requires a larger angle of attack to achieve the same lift. Due to the changed velocity distribution this example exhibits lower drag and, the main reason for sweeping wings, a higher critical Mach number allowing higher flight speeds.

The Aerodynamic Center

The output of the “Polars” and “Aircraft” cards contains a column with the position of the aerodynamic center (A.C.). The aerodynamic center is a point on the airfoil at which the pitching moment is constant (not necessarily zero) for all angles of attack.
It can be calculated from the gradient of the pitching moment over lift coefficient curve:

$$x_{\mathrm{A.C.}} = 0.25 - \frac{\partial C_{\mathrm{m}\,0.25}}{\partial C_\ell}.$$

According to thin airfoil theory the aerodynamic center is located at 25% of the chord length and does not move when the angle of attack is changed. In real life airfoils are thick and the location typically can vary about ±2 % around this location.

The aerodynamic center is not to be confused with the center of pressure (C.P.), which is the point at which the total aerodynamic force acts. This total force produces the same effect as the lift and pitching moment. The location of the center of pressure changes with angle of attack and can even move in front or behind the airfoil shape. The center of pressure can be calculated from lift and pitching moment coefficients:

$$x_{C.P.} = 0.25 - \frac{C_{m\,0.25}}{C_\ell}.$$

Note that both, center of pressure as well as the aerodynamic center, are for the airfoil only, not for the complete aircraft with tailplanes.

Effect of a Ground Surface

When a wing is brought close to the ground, its characteristics are changed considerably. First the pressure distribution around the two dimensional airfoil shape (a wing of infinite span) is affected by the presence of the ground. Second, the lift and the induced drag of a wing of finite span are affected also.

JAVAFOIL simulates the ground effect on the flow around the two dimensional airfoil by using a mirror image of the airfoil section. The mirror plane is always located at y = 0. Note that for a proper simulation, the baseline airfoil must be translated into the positive y-direction so that it does not intersect the horizontal line y = 0. This translation can be performed using the Modify card.
In contrast to the flow around a free airfoil, where the flow field can be constructed from a superposition of the solutions for zero and 90 degrees angle of attack, the ground effect case cannot be created by superposition. Any change of angle of attack also changes the geometry of the airfoil and mirror airfoil pair. Therefore a new panel solution is required for each angle of attack, which slows down the calculation of a polar somewhat. More slowdown will be noticed during the analysis on the Flowfield card because here an analysis is required for each single flow field point and for each step on each streamline.

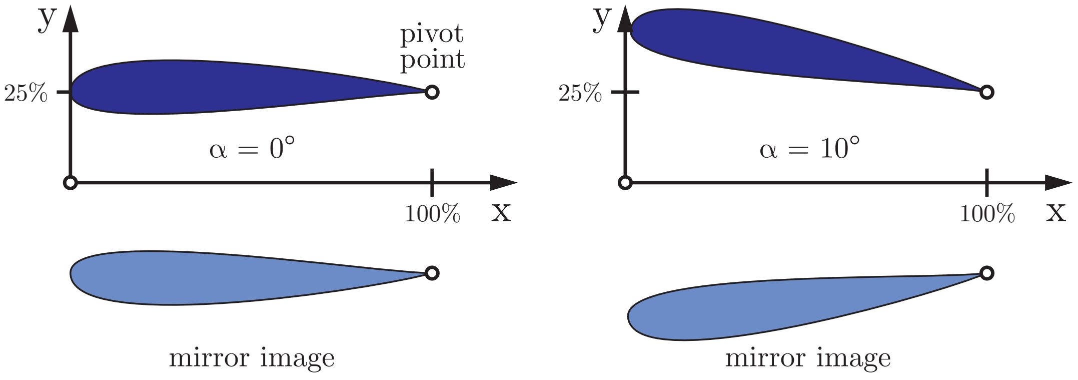


Figure 30: Changing the angle of attack in ground effect rotates around the pivot point.

The angle of attack of the airfoil is always changed by rotating the section around the pivot point specified on the Modify card. If you want to analyze an airfoil at a height of 25% of the chord length and want to maintain the trailing edge point, you would first translate the airfoil in y-direction by 25% and then set the pivot point to x=100%,y=25%. Then any subsequent change of angle of attack would maintain the trailing edge point and elevate the nose of the airfoil above the y=25% line. Note that the airfoil is rotated and thus its projection on the x/c axis becomes shorter, but pressure, velocity or Mach number distributions on the Velocity card are still plotted over x/c.

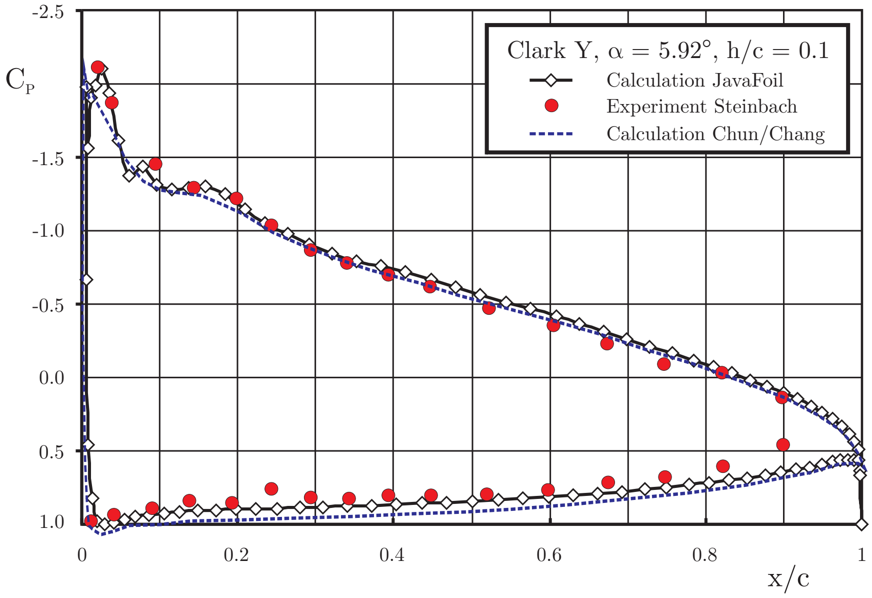


Figure 31: Distributions of the pressure coefficient on a Clark Y airfoil in ground proximity.

Results of the numerical solution of the Navier-Stokes equations have been taken from [19], the experimental results have been reproduced from [20]. The experiments were carried out with a fixed ground board, equipped with a suction system. The results of JAVAFOIL match the experimental results quite well. The Navier-Stokes solutions should model boundary layer displacement effects more accurately.

The ground effect on a wing of a finite span is approximated by applying a modified calculation of the induced drag. If you specify the aspect ratio of the wing $\Lambda = b^2 / S$ (span b and wing area S), and the height of the wing above ground h / b (height h over wing span b) on the Options card, these values are used to calculate an approximation of the induced drag using

$$C_{D,i} = \frac{C_L^2}{\pi \cdot \Lambda} \cdot \left(1 - \frac{33 \cdot (h/b)^{1.5}}{1 + 33 \cdot (h/b)^{1.5}} \right).$$

Effect of a Water Surface

Another application of the “airfoil” is the hydrofoil, an airfoil moving under water. The water surface acts similar to a ground plane, but when the hydrofoil is operating close to the surface there is a subtle difference. The pressure field of the airfoil affects the shape of the surface – the surface is pushed upwards where an overpressure occurs and sucked downwards where the local pressure is lower than the ambient pressure. Therefore a wave is forming on the surface. This wave affects the curvature of the flow around the airfoil.
JAVAFOIL includes a simple water surface model which can be activated on the Options card. This “Froude-effect”model is valid for high (to be precise: infinite) Froude numbers.
The Froude number, which is an important parameter for hydrodynamic analyses (similar to the Reynolds number in aerodynamics), is usually[1] defined as

[1] Sometimes the Froude number is given as the square of this expression – be careful when comparing data.

$$\mathrm{Fr} = \frac{v_\infty}{\sqrt{g \cdot c}} \; .$$

In JAVAFOIL you have to move the airfoil below the water surface to the appropriate depth, which is always located at y = 0. This translation can be performed using the Modify card. Note that the surface effect diminishes with immersion depth and it can be assumed that it can be neglected when the airfoil is submersed deeper than 10 chord lengths below the surface.

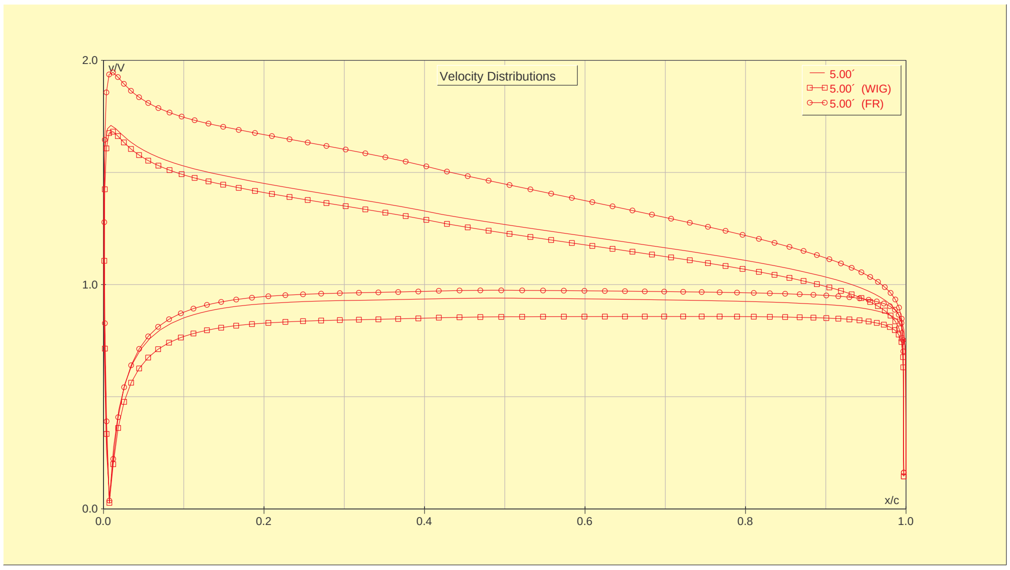


Figure 32: Velocity distribution past a NACA 2412 at an angle of attack of 5°. The results are for free flow, at 50% c above the ground and submerged 50% c below the water surface. Note that the pivot point has to be set to y=0, +50%, -50% respectively to minimize distortion of the x-axis die to angle of attack for the ground effect and Froude effect analysis.

Case	**α**	**Cl**	**Cd**	**Cm 0.25**	**Reference Point**
NACA 2412	**[°]**	**[-]**	**[-]**	**[-]**	
free	5.0	0.843	0.01610	-0.089	25% / 0%
ground effect	5.0	0.875	0.01403	-0.064	25% / +50%
Froude effect	5.0	1.234	0.02354	-0.091	25% / -50%

Force and moment coefficients for the three cases depicted above.

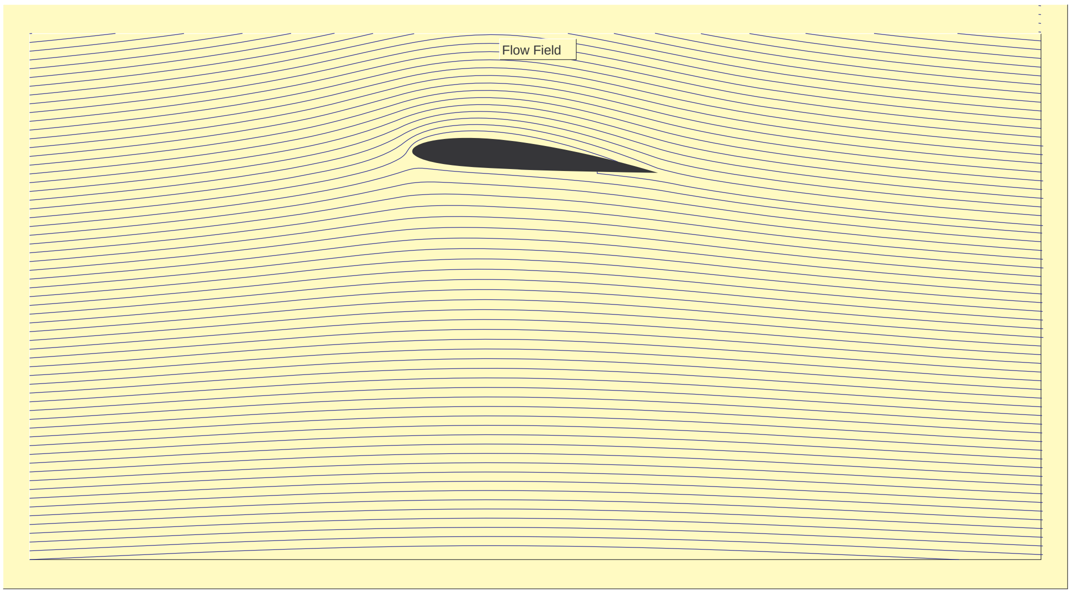


Figure 33: Streamlines around the submersed hydrofoil (note that image is clipped at y=0) but the generated surface wave are extending above this border.

Multi-Element Airfoils

The maximum lift of single element airfoils is limited by onset of flow separation. The achievable limit for single element airfoils seems to be at lift coefficients between 2 and 3. For maximum lift it can be beneficial to split an airfoil unto several elements, arranged to form a slotted cascade. Each element then develops its own fresh boundary layer and positive interference effects between the elements allow for a higher lift loading per element. JavaFoil can handle such multi-element airfoils to a limited extent. Limitations are imposed by the fact that boundary layer effects are not modeled. Therefore inaccurate results must be expected when slots are very narrow (less than twice the displacement thickness of the boundary layer and when the wake of a leading element interacts with a following element. Nevertheless JavaFoil should be useful to produce a reasonable first design for a slotted airfoil with appropriate gap, overlap and element angle settings. Also one can design such sections so that suction peaks and too steep pressure gradients are avoided.

The following script shows how a two element downforce wing section can be generated starting with a basic NACA 4-digit section.

```
//
//
// A simple JavaFoil example which creates a two element
// airfoil for downforce generation.
//
// switch to US country settings
Options.Country(0);
// create a cambered NACA airfoil for starting
var dThickness = 0.15;
var dCamber = 0.02;
var dCamberLocation = 0.40;
var params = new Array ( dThickness,0.0, dCamber, dCamberLocation );
Geometry.CreateAirfoil(0,61,params,true);
//
// create a copy of this first airfoil element to be used later
Modify.Select(1);
Modify.Duplicate();
```

```
// now select the first element again
Modify.Select(1);
// ... flip it upside down
Modify.Flip(0.0);
// ... scale it to 75%
Modify.Scale(0.75);
// ... rotate it 5 degrees trailing edge up around its nose (0;0)
Modify.Rotate(0,0,-5);
//
// now select the second element
Modify.Select(2);
// ... flip it upside down
Modify.Flip(0.0);
// ... scale this copy to 30%
Modify.Scale(0.30);
// ... move it back so that there is 5% overlap
Modify.Move(0.70,0.12);
// ...rotate it by 30 degrees around a point at (70%/12%)
Modify.Rotate(0.70,0.12,-30);
// finally: (THIS IS IMPORTANT!)
// select both elements again for all further analyses
// if only one element is selected on the Modify card, only this element
// will be considered during the calculation of polars etc.!
arr = new Array(2);
arr[0] = 1;
arr[1] = 2;
Modify.Select(arr);
// and move both elements up by 25% for ground clearance
// (note that the airfoil may not cross y=0 i.e. the ground plane)
Modify.Move( 0.0, 0.25 );

/*
 * prepare for analysis
 */
// switch ground effect simulation ON
Options.GroundEffect(1);
Options.MachNumber(0);
Options.StallModel(0);
Options.TransitionModel(1);
Options.AspectRatio(0);
Modify.SetPivot(0.25,0);
// velocity versus x/c should show no strong suction peaks in the
// nose region of 2nd element
Velocity.Analyze(0,0,1,0,false);
//
// polar for Re=500'000, alfa=-15 to +10 degrees
// with ground present a strong suction force is generated
// (even for single element airfoils)
// therefore a Cl min in the order of -4 to -5 can be seen.
Polar.Analyze(500000,500000,500000,-15,10,1,1.0,1.0,0,false);
//
// finally: export coordinates in XML format
Geometry.Save("Z:/groundforce-example.xml");
```

Automating JavaFoil with a Script

JavaFoil had a rather simple but useful scripting interface since 2002. In 2012 this has been rewritten to make use of JavaScript as the scripting language. This scripting engine is included with Java since version 1.5 and allows for more complex scripts.

Scripts can be recorded, modified and stored for later reuse in script files. Basically these scripts mimic the actions you perform in the graphical user interface. The syntax must follow the JavaScript rules, which are not detailed here. There is no direct interaction with JavaFoil, i.e. you cannot directly ask for certain values. If you want to build up something like an

optimization loop, you have to write results to output files and parse these results to extract data.
The easiest way to start with scripting is to use the script recorder built into the Script dialog window, which can be opened on the Options card.

Sometimes it is useful to have JavaFoil execute a command sequence in a script automatically and then terminate. This allows running JavaFoil inside a parameter sweep or as part of an optimization loop. For this purpose you prepare the script file and start JavaFoil with the name of the script file on the command line. Then JavaFoil runs invisible, without opening a window, and executes the commands in the script file.

The name of the script file can be transferred to JavaFoil in two ways. First you can define a "system property" using the "-D" command line option of the java command, look like so:

```
java.exe -DScript="Path\Script" -cp "Path\mhclasses.jar" -jar "Path\javafoil.jar"
```

Alternatively, you can specify the script file using as a command line argument like this:

```
java.exe -cp "Path\mhclasses.jar" -jar "Path\javafoil.jar" Script="Path\Script"
```

Both ways methods are equivalent and produce the same result.

Note that this example in Windows style uses the backslash as file separator, for Unix-like systems you have to use the appropriate separator, usually a forward slash.
JavaScript is uses the backslash character as an "escape" character e.g. to specify a "newline" character by "\n". In Windows the backslash character is also used as a path separator. Inside a script you must therefore make sure that path names inside strings are written with double backslashes to separate directory elements (e.g. `filename = "C:\\dir1\\dir2\\file.name";`).

Note:
As JavaFoil is running without showing a window, you must make sure that the script ends with an `JavaFoil.Exit()` command to terminate the run of JavaFoil properly. Otherwise, the JavaFoil process will continue to run in the background. In Windows you can check for running JavaFoil processes using the Task Manager window. In Unix-like operating systems you can use the "ps" command to list all processes running under your user account.

References

[1] Mark Drela, Michael B. Giles, "Viscous-Inviscid Analysis of Transonic and Low Reynolds Number Airfoils", AIAA-86-1786-CP, 1986

[2] Xiao-liang Wang, Xue-xiong Shan, "Shape Optimization of Stratosphere Airship", Journal of Aircraft V43N1, 2006.

[3] Ira Abbott and Albert Von Doenhoff, "Theory of Wing Sections", Dover Publications, New York, 1959

[4] Eastman N. Jacobs, Kenneth E. Ward, Robert M. Pinkerton, "The Characteristics of 78 Related Airfoil Sections from Tests in the Variable-Density Wind Tunnel", NACA Rep. No. 460, 1933.

[5] John Stack, "Tests of Airfoils Designed to Delay the Compressibility Burble", NACA Rep. No. 763, 1943.

[6] Charles L. Ladson and Cuyler W. Brooks, Jr., “Development of a Computer Program To Obtain Ordinates for NACA 6- and 6A-Series Airfoils”, NASA Technical Memorandum TM X-3069, September, 1974.

[7] Charles L. Ladson, Cuyler W. Brooks, Jr. and Acquila S. Hill, “Computer Program To Obtain Ordinates for NACA Airfoils”, NASA Technical Memorandum 4741, December, 1996.

[8] A. K. Martynov, “Practical Aerodynamics”, Pergamon Press 1965.

[9] A. C. Kravets, “Characteristics of Aircraft Profiles”, Moscow 1939.

[10] Anonymous, “Measurement of Force Coefficients on the Aerofoils EC 1240 and EC 1240/0640 in the High-Speed Tunnel at the National Physical Laboratory”, British ARC R&M 2246.

[11] Sidney Goldstein, “Approximate Two-Dimensional Aerofoil Theory. Part I. Velocity Distributions for Symmetrical Aerofoils”, British A.R.C. Technical Report C.P. No. 68, 1952.

[12] British A.R.C. R&M 4726.

[13] British A.R.C. R&M 4978.

[14] Joseph Katz, Allen Plotkin, “Low-Speed Aerodynamics”, McGraw-Hill, 1991.

[15] H. B. Helmbold, F. Keune, “Beiträge zur Profilforschung, II. Geometrie der Profilsystematik”, Luftfahrt-Forschung, Band XX, 1943, pp 81 ff.

[16] G. Roßner, “Über eine Klasse von theoretischen Profilen mit vier frei wählbaren geometrischen Parametern”, Jahrbuch 1942 der Deutschen Luftfahrtforschung, p.I/141-I/159, 1942.

[17] K. G. Wilkinson, „The Horten Tailless Aircraft - How the Hortens Design Their Aircraft“, Farnborough, ca. 1946, [http://www.twitt.org/Farnborough.html], retrieved 13 January 2012.

[18] Chun Ho-Hwan, Chang Chong-Hee, “Development of S-shaped section”, Pusan national university, 1996.

[19] H. H. Chun, R. H. Chang, “Turbulence Flow Simulation for Wings in Ground Effect with two Ground Conditions: fixed and moving Ground”, International Journal of Maritime Engineering, Royal Institution of Naval Architects, 2003.

[20] Dieter Steinbach, “Comment on Aerodynamic Characteristics of a Two-Dimensional Airfoil with Ground Effect”, Journal of Aircraft, Col. 34, No. 3, pp. 455-456, 1996.

[21] Werner Würz, “Hitzdrahtmessungen zum laminar-turbulenten Strömungsumschlag in anliegenden Grenzschichten und Ablöseblasen sowie Vergleich mit der linearen Stabilitätstheorie und empirischen Umschlagskriterien”, Dissertation, Institut für Aero- und Gasdynamik, Universität Stuttgart, 1995.

[22] Richard Eppler, D. Somers, „A Computer Program fort he Design and Analysis of Low-Speed Airfoils“, NASA TM-80210, 1980.

[23] Richard Eppler, „Airfoil Design and Data“, Springer-Verlag, 1990.

[24] T. Cebeci, P. Bradshaw, “Momentum Transfer in Boundary Layers”, Hemisphere Publishing Corporation, Washington, London, 1977

[25] A. R. Wazzan, C. Gazley, A. M. O. Smith, “H-Rx method for Predicting Transition”, AIAA Journal, Vol. 19, No. 6, June 1981, pp.810-811.

[26] K. G. Guderley, “Theorie schallnaher Strömungen”, p. 235 ff., Springer, 1957.

Version History

```
/**
 *  1.00      ... 2001  initial version
 *  1...                various small undocumented changes over the months
 *  1.46    18.04.2002  setGUILocale enhanced, printing errors repaired
 *  1...                various small undocumented changes over the months
 *  1.49       07.2002  experimental implementation of MGM design method.
 *  1.50    20.08.2002  Lots of changes and improvements concerning the handling
 *                      of multi element airfoils. Most changes in MH.AeroTools.*
 *  1.51    22.08.2002  Added simple handling of ground effect on all cards.
 *                      Most changes in MH.AeroTools.*
 *  1.52    23.08.2002  Moved Mach and ground effect settings to Options-Card
 *                      because these are common for all cards
 *  1.60    24.08.2002  new scripting mechanism added
 *  1.61    11.09.2002  finite wing corrections for Cl and Cd added
 *  1.70    10.11.2002  design card released
 *  1.71    10.11.2002  mouse drag of y-coordinates on modify card added
 *  1.72    10.01.2003  Spanish translation added
 *  1.73    22.02.2003  normal vector plot added to flow field
 *  1.74    10.05.2003  boundary layer added to design card (transition, separation)
 *  1.75    30.05.2003  transition models added, Cd<->Cm in table on boundary layer card swapped
 *  1.76    20.06.2003  normal vector Cp-plot in flow field changed to spectrum filled outline
 *  1.77    10.08.2003  update to remedy problem with growth of XY canvases
 *  1.78    16.02.2004  application command line arguments are now used,
 *                      Exit() script command added
 *                      AFL polar export for XPlane 7.00 added
 *  1.79    17.04.2004  NACA 6-series approximative shape generation added
 *  1.80    16.05.2004  rotational flow field feature added (experimental/Script only)
 *  1.81     7.10.2004  small changes to XML export to Polar card
 *  1.82     3.01.2005  TsAGI "B" airfoils added to Geometry card
 *  1.83    12.01.2005  Design card improved by adding arc length display option
 *                      Text direction record in CGM export in MH.AeroTools.* added
 *  1.84    13.03.2005  circular arc, wedge, plate airfoil generators added to Geometry card
 *  1.85    27.03.2005  Finnish translation added
 *  1.86     6.07.2005  NPL EQH airfoil shape added to Geometry card (symmetrical)
 *  1.87    26.11.2005  H-Rx transition criterion added
 *                      Granville transition criterion added
 *  1.88    17.12.2005  Van de Vooren airfoil added
 *  1.90    31.01.2006  Newman airfoil added
 *  1.91    21.09.2006  display of Mach number versus x/c to "Velocity" card added,
 *                      compressibility corrections moved into new classes and
 *                      MH.AeroTools.Airfoils.PanelAnalysis cleaned up and corrected
 * 1.92     1.03.2007   all cards enlarged in size
 * 1.93    25.03.2007   Cf plot added to BoundaryLayer card
 * 1.94     6.05.2007   Cf output multiplied by 2 to adhere to common conventions
 *                      Additions to allow for embedding of JavaFoil
 * 1.95     9.06.2007   Mach influence on velocity vector in space corrected (in MHClasses)
 * 1.96    19.06.2007   Error with incorrect Cp display on Velocity card corrected
 * 1.97    23.07.2007   NACA 4 digit inactive option  x_t/c disabled, minimum aspect ratio
 *                      set to 0.01 even if this makes not much sense.
 * 1.98    13.09.2007   MHClasses: XYCanvas Import Dialog repaired for Java 1.6
 * 1.99    17.11.2007   Geometry card: airfoil type dropdown box selection is
 *                      restored when country is changed, airfoils of old Selig
 *                      format can be imported (first line contains nUpper nLower)
 * 2.00    13.01.2008   Aircraft card for aircraft oriented airfoil polars added
 * 2.01    13.02.2008   Aerodynamic center position added to polar listings
 * 2.02    21.04.2008   Dutch language version completed
 * 2.03    18.06.2008   NACA 5 digit definition improved (m factor)
 * 2.04    28.06.2008   Inverse design card enhanced by CpTarget specification window
 * 2.05    28.08.2008   Center of pressure position added to polar listings.
 *                      Robustness of number format parsing of coordinate on
 *                      geometry card improved.
 *                      Input and output of polars etc now uses UTF-8 Unicode
 *                      charset for alfa etc.
 * 2.06    22.11.2008   Incorrect TsAGI "B" generator repaired.
 *                      Trailing edge handling in panel method cleaned up.
 * 2.07    13.12.2008   Joukovsky airfoil generator repaired.
 * 2.08    07.02.2009   Froude free surface approximation added (not yet working correctly).
 * 2.09    16.08.2009   Added values to tables on Flowfield and BoundaryLayer cards.
 *                      Replaced NACA 6 and 6A series generator by more accurate
 *                      algorithm according to Ladson's NASA report (with tables).
 * 2.10    06.01.2010   A negative number of points on Modify card now creates a
 *                      spline interpolation of strong tension, almost a polyline.
 *                      Clean NPL EC and ECH airfoil definitions added.
 *                      FlowField filled color plot improved.
 *                      Autoscaling for geometry graph on Modify card as well as x-y-axes added
 * 2.11    21.02.2010   MHSpinner controls added, MHText controls added,
 *                      more robustness in airfoil generators against nonsense input.
 * 2.12    13.03.2010   minor modifications, check for Java 1.5, transition and separation
 *                      location labels added to BoundaryLayer card
 * 2.13    12.04.2010   added functions to Design card. Fixed a AC calculation bug when a
 *                      single point was analyzed on 2D polar or airfoil polar cards.
 * 2.14    15.05.2010   added Roßner airfoils.
 * 2.15    30.05.2010   fixed a bug in Prandtl-Glauer correction and added robustness.
 * 2.16    30.06.2010   XML export format to Velocity card added.
 *                      Option for colored v/V on Flow field card added.
 * 2.17    26.11.2010   smoothing option changed
 * 2.18    02.07.2011   Bitmap import on Geometry card added,
 *                      colorramp display on Flowfield card added.
 * 2.19    05.09.2011   Automatic saving on restoring of current settings added.
 *                      Added scripting access to Height/Span ratio on Options card.
 * 2.20    01.10.2012   Fixed scripting access to FlipY and Smooth commands on Modify card.
 *                      Ground and Froude effect analysis now uses the pivot point
 *                      from Modify card for angle of attack.
 *                      Froude simulation rewritten correctly, some code unification with ground effect.
 *                      Cp and airfoil geometry added to Tecplot output from Flowfield card.
 *                      Mach number corrections corrected.
 *                      Added swept wing section analysis feature.
 * 2.21    02.03.2014   Option to enter the number of decimal places added to the Geometry card.
 *                      Momentum and pressure field integration on Flowfield card added.
 * 2.22    04.11.2014   Finite wing approximation extended to pitching moment correction.
 * 2.23    15.09.2015   Bug in DXF export on geometry card fixed.
 * 2.24    11.04.2016   Bug in scripting interface (missing JavaFoil.Exit() method) fixed.
 * 2.25    18.01.2017   Script recorder now escapes backslashes in path names on Windows.
 * 2.26    02.04.2017
 * 2.27    18.08.2017   Improvement in Calcfoil stall model.
 * 2.28    22.12.2017   Reversion of “Improvement” in Calcfoil stall model introduced in 2.27.
 */
```