White paper:
Airfoil Evaluator: Smoothing Function

J. Saverin MSc

Wind Energy Group: Hermann Foettinger Institute, TU Berlin

Email: Sav_flowsol@outlook.com

The online tool Airfoil Evaluator [1] allows free access to a huge database of airfoils and makes their evaluation simple and user-friendly for anybody, anywhere, whether they be students beginning their studies, or practising engineers active in development. The potential to create such a useful tool for the analysis of airfoil flow problems is the motivation for the collaboration between NOVASCIENTIA [2] and J. Saverin, the author.

The tool allows the refinement of a number of existing or user-defined aerodynamic profiles. An important quality of the profile is that geometric quantities along the surface are continuous and have smooth derivatives. These quantities include:

- Surface inclination angle θ
- Normal inclination angle θ_n
- Panel length ds (between panel points)

This can be usually accomplished with sufficient surface resolution. This is often however not the case as a result of:

- Decimal truncation (during importation)
- Input error (user-defined coordinates)
- Method of coordinate import

The purpose of the smoothing function is to remove the discontinuous, small-scale deviations on the surface while retaining the aerodynamic profile of the airfoil. This is essentially the same as filtering noise from a signal in signal processing theory. For this reason a signal filter has been used here. The smoothing function has been designed with the following criteria in mind:

- The trailing edge is incredibly important in the aerodynamic analysis and the determination of the lift coefficient c_l. These coordinates (upper and lower TE) must be retained;

- Equivalently, the tangency of the flow off the trailing edge is very important. As such the tangent vectors of the suction and pressure sides $\vec{s}_{upper}$ and $\vec{s}_{lower}$ should be retained as near as possible;

- Any harmonic behaviour which may result by interpolating the imported surface coordinates should be avoided.

For these reasons, an adaptable *Butterworth* filter has been employed. The key advantage is that there is no harmonic behaviour over the surface. Provided the parameters of the filter are optimised (particularly for the number of coordinates N), the filter appears to produce particularly good results.

Theory of the Butterworth filter

A general transfer function for any control loop can be given by:

$$H(s) = K\frac{(s-z_1)(s-z_2)\cdots(s-z_N)}{(s-p_1)(s-p_2)\cdots(s-p_N)} = K\frac{r_1 r_2 \cdots r_N}{d_1 d_2 \cdots d_N} \cdot e^{\Sigma(\psi_i)-\Sigma(\delta_i)} \tag{1}$$

where z_i and p_i are the zeros and poles respectively of the transfer function. The second equality expresses the transfer function in polar form. If the frequency input of a signal approaches a pole (p_i) of the function, as can be seen by the second equality of the equation above, it's signal is amplified, the opposite is true as it approaches a zero. With this simple principle in mind, it is possible to design a transfer function (filter) which vanishes near certain frequencies (see high-pass, low-pass etc. filters). We imagine now a form of low-pass filter where the magnitude of the transfer function is given by:

$$|H(i\omega)| = \frac{1}{\sqrt{1+\omega^{2n}}} \tag{2}$$

This has been normalised for the frequency unity. This function is plotted for increasing n values in Figure 1. It can be seen as $n \to \infty$ the transfer function works as a perfect low-pass filter, allowing only frequencies below the threshold through. As we only care about the poles which lie in the plane $Re(z) <= 0$, the poles of this function are given by:

$$p_k = e^{\frac{i\pi}{2n}(2k+n-1)} = \cos\frac{\pi}{2}(2k+n-1) + i\sin\frac{\pi}{2}(2k+n-1) \qquad k = 1, 2\cdots n \tag{3}$$

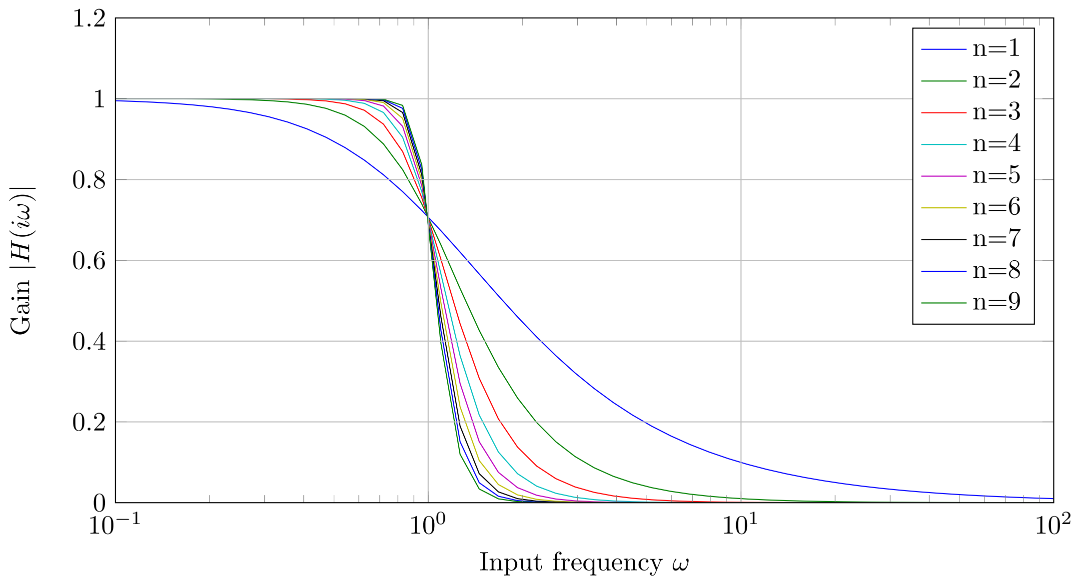

Figure 1: Low-pass filter of Butterworth type

These all lie on the unit circle in the left-half plane of the complex number plane. The normalised transfer function of this filter is hence given by:

$$H(s) = \frac{1}{(s - s_1)(s - s_2) \cdots (s - s_n)} \tag{4}$$

This is referred to as a Butterworth filter. The order of the denominator of equation 4, known as the *Butterworth polynomial* determines how steeply the filter dives at the critical frequency.

1 Improving filter input

The imported data can be pre-prepared before passing into the filter, which greatly improves the output signal. The main problems which have been seen to occur (regardless of whether they have an input on the filtering) are:

- Inconsistent data spacing
- Non-ideal data point distribution
- Multiply defined coordinates
- Repeated x coordinates

- Trailing edge inconsistencies

Each of these are described and the solution implemented in the smoothing function is elucidated.

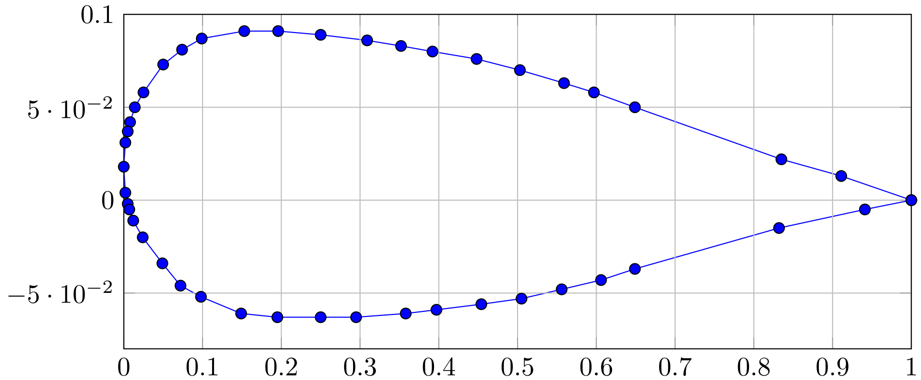

Figure 2: B737a airfoil: Imported coordinates [**?**, AE]

Rediscretisation of the surface

The imported distribution of coordinates can be completely unpredictable, as illustrated in figure 2. Should the coordinates later be passed into a potential flow model, it is important to increase the density in regions where flow curvature will be large, ie. the trailing edge and leading edge. The approach to this is generally to have a cosinus distribution of the coordinates. This increases the density of the x coordinates near the extremes of the domain and ensures that coordinate spacing in non-critical regions is relatively uniform.

Monotonic x coordinates

In order to interpolate the coordinates so that they have a cosinus distribution, the x coordinates must be made to be monotone increasing. For a given x value must correspond a unique y value, otherwise the searching algorithm won't be able to identify between whether the coordinate is on the suction or the pressure side of the airfoil (see Figure 2: Drawing a vertical line from $x = 0.6$ will produce two intersections). The airfoil coordinates are made monotonic by simply stepping through the coordinates and adding the product of the absolute difference in x value:

$$x_{i,mono} = \sum_{j=1}^{i} |x_j - x_{j-1}| \tag{5}$$

This is illustrated in Figure 3. In this process, it is recorded which coordinate reflects the *inflection* point of the profile, or the point where the sign of $(x_j - x_{j-1})$ changes, this point shall herein be referred to as x_{inf}. This is then used later for ensuring a cosine distribution on both suction and pressure sides. In addition to this, any coordinate pairs with equivalent x-coordinates are interpolated and represented by a single coordinate (with the average y value of the coordinate pair). This again overcomes difficulties which arise in trying to interpolate coordinates.

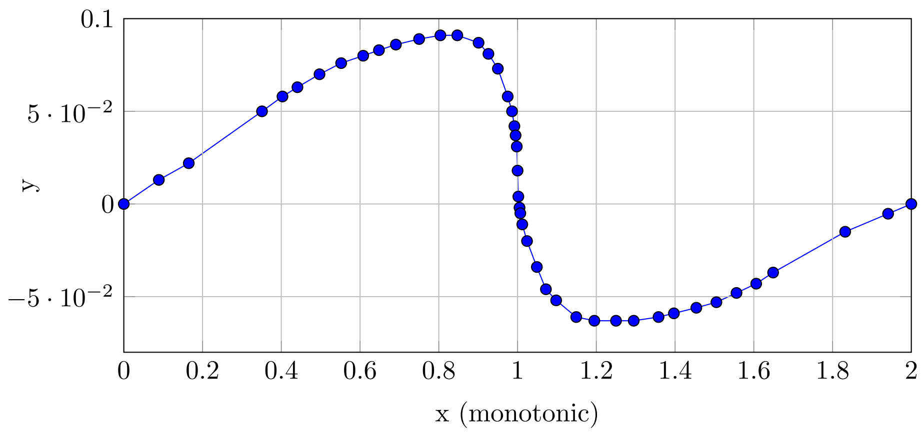

Figure 3: B737a airfoil: Monotonic x-coordinates.

Cosine distribution

Once the airfoil x coordinates are monotone increasing, the upper and lower surface y coordinates are linearly interpolated at x coordinates which have a cosinus distribution. By representing the monotonic x coordinate at the final point by x_N The suction side coordinates are on the interval: $x_{mon} \in [0, x_{inf}]$ and the pressure side coordinates are on the interval $x_{mon} \in [x_{inf}, x_N]$. The rediscretised x coordinates are then given by:

$$x_{i,mon} = \begin{cases} x_{inf}\left(\frac{1}{2} + \frac{1}{2}\cos\theta\right) & \theta \leq \pi \\ \frac{1}{2}(x_{inf} + x_N) + \frac{1}{2}(x_N - x_{inf})\cos\theta & \theta > \pi \end{cases} \quad \text{for:} \quad 0 \leq \theta \leq 2\pi \tag{6}$$

This is illustrated in Figure 4. At this stage the cosine-distributed coordinates (non-monotonic) are also stored for the final coordinates.

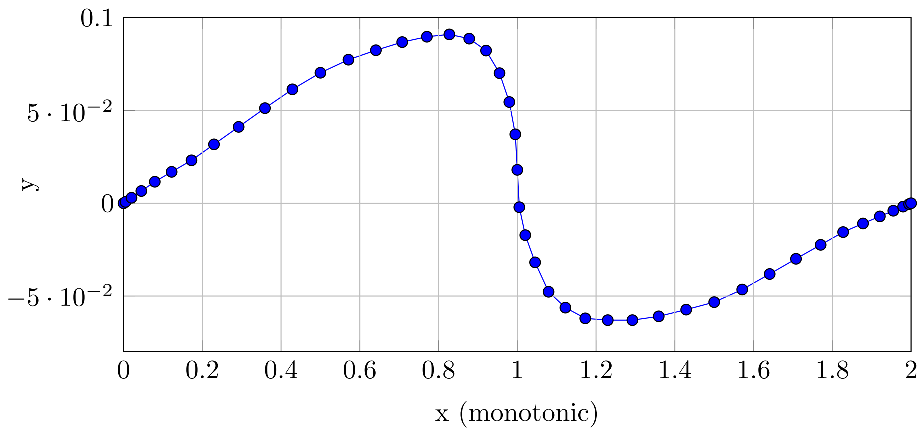

Figure 4: B737a airfoil: Monotonic x-coordinates with cosine distribution.

Interpolation

The previous steps ensure that the cosinus distribution is also monotonic increasing and to every x coordinate in the range $x \in [0, x_N]$ there exists a single y value, which is interpolated between the points of the original data with the formula:

$$y(x) = y_i + m(y_{i+1} - y_i) \qquad \text{where:} \qquad m = \frac{x - x_i}{x_{i+1} - x_i} \tag{7}$$

Although a simple linear interpolation appears crude and does not account for surface curvature, it is reminded here that the coordinates have yet to be passed through the filter, which we wish to bring as closely to the original input signal as possible. The coordinates are now re-ordered to a *non-monotonic* form and the interpolated values provide the rediscretised dataset. This is illustrated in Figure 5.

2 Trailing edge treatment

As was mentioned previously, the trailing edge is extremely important in aerodynamic calculations. The flow solver developed for the Airfoil Evaluator formulates the solution differently depending on whether the trailing edge is sharp or blunt, for example. It can generally be said that the character of the trailing edge should be preserved. The following assumption shall be made here:

- Although the geometric properties near the trailing edge are not continuous, the first and last coordinates of the imported data represent adequately the trailing edge of the airfoil.

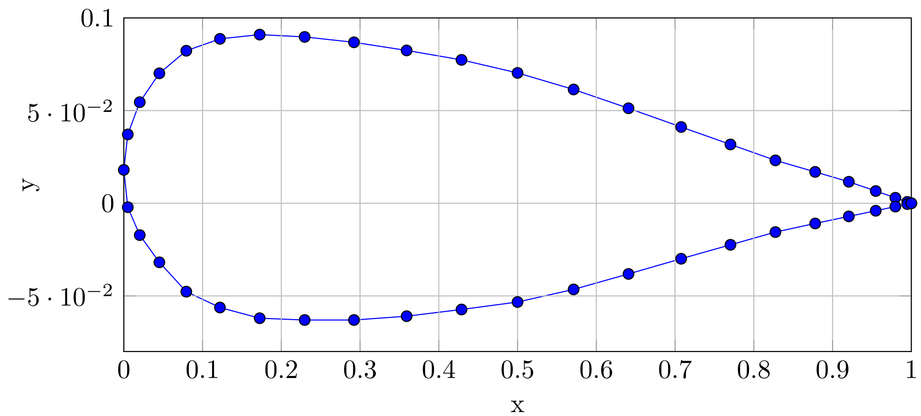

Figure 5: B737a airfoil: Cosine coordinate distribution.

Working with this assumption, any interpolation/filtering action should preserve the trailing edge coordinates. The next step is to try to remove discontinuities in the dataset near the trailing edge. It appears often that the data is discontinuous near the trailing edge owing to the method of data import, see Figure 6 . This manifests itself in two different ways:

- The slope angle of the panels are discontinuous, or:
- Sharp trailing edge coordinates collapse onto themselves before actually reaching the trailing edge.

The second of these means that the flow solver is trying to model a panel which has zero thickness, and identical panel endpoint, which causes the solution matrix to be singular. A method has been devised which attempts to *average* out the coordinates of the trailing edge. The first and last coordinates are retained while points near to the trailing edge are smoothed according to the function:

$$y_{j,new} = y_{j-1} + m(y_{j+1} - y_{j-1}) \qquad \text{where:} \qquad m = \frac{y_j - y_{j-1}}{y_{j+1} - y_{j-1}} \tag{8}$$

Which is progressively applied to coordinates approaching the trailing edge until the trailing edge is reached. The criteria for where to begin chooses the larger interval from:

- Coordinates which lie within $0.02c$ or 2% of the chord length from the trailing edge, or:
- One coordinates further than the number of coordinates which are collapsed at the TE

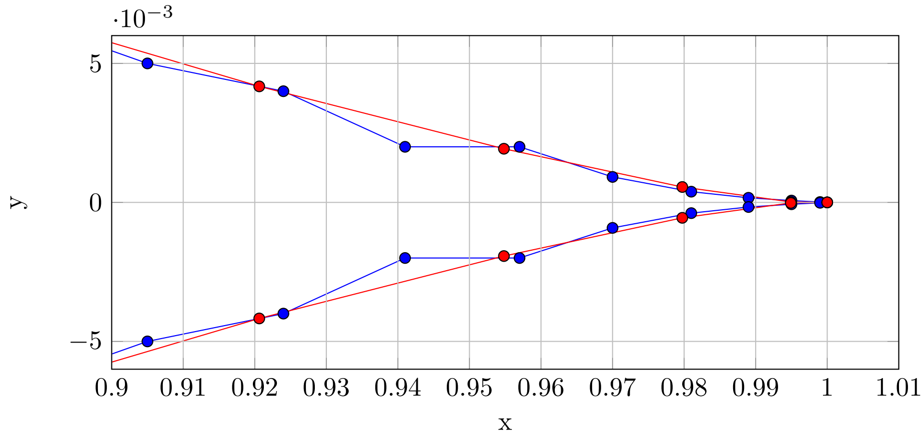

Figure 6: Joukowski airfoil trailing edge: Original, collapsed coordinates (blue), refined coordinates (red).

By applying to both the suction and pressure sides, collapsed coordinates are completely avoided and discontinuities appear to be *ironed-out* relatively well, as is illustrated for a Joukowski airfoil in Figure 6.

3 Filtering

Now that the signal has been cleaned up, it is passed through the butterworth filter. Use has been made of the `butter` function of the python package and module **scipy.signal**. This creates first the convolution coefficients for use in the implementation of the filter. The convolution coefficients are a function of O, the order of the filter (see equation 4) and Wn, the (normalised) critical frequencies for the filter (Figure 1 displays a normalised filter with only a single critical frequency). Trial and error has shown that the order of the Butterworth polynomial should be dependant upon the number of coordinates. It has been assumed here that the number of coordinates should lie in the range $10 < N < 200$ where $N = 10$ corresponds to the crudest of user-defined coordinates and $N = 200$ corresponds to a highly refined surface coordinate distribution (which is more than enough to capture most important geometric features).

If the order of the polynomial is chosen to be too high, the surface is exactly recreated. In contrast, if the order is too low, geometric features dissolve and are not recreated, for this reason the order of the filter is calculated for each data set to be:

$$O = \frac{N}{50} + 1 \tag{9}$$

Python syntax dictates that this value will always be an integer and will (for $10 < N < 200$) produce a polynomial of the order $2 < O < 4$. The normalised frequency Wn also needs to be chosen to minimise oscillation in the final solution. Although for larger N values, it can be practical to set $Wn = 0$, this literally removes all oscillations, setting this condition for datasets with smaller N values means that the airfoil contour is unsufficiently resolved, leading to oval-like forms. For this reason, again by trial and error, this value has been set to:

$$Wn = 1 - \frac{N}{200} \qquad \text{with restrictions:} \qquad 0.1 \leq Wn \leq 1.0 \tag{10}$$

These settings appear to function very well for the range of airfoils trialled. Once the butterworth coefficients have been calculated, the filter must be implemented in an appropriate way as to maintain the initial and final values (trailing edge coordinates).

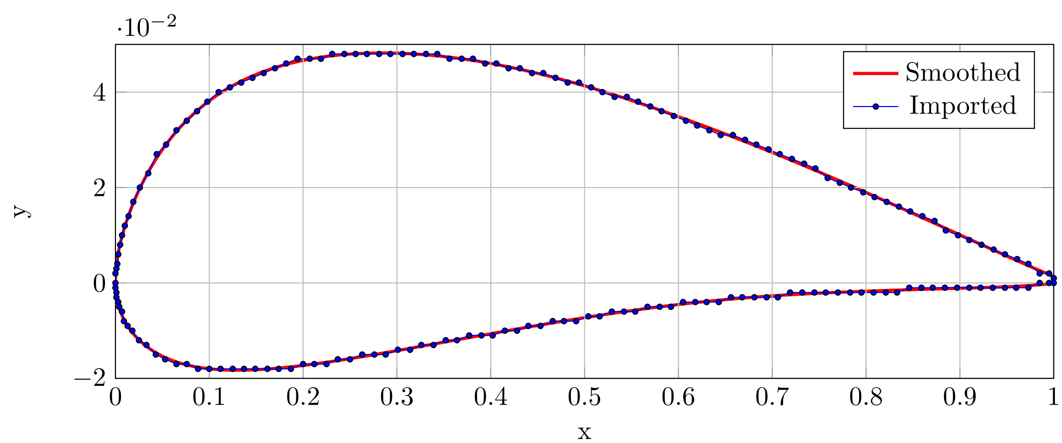

Figure 7: Drela AG-03 Airfoil: Y-axis exaggerated to emphasise geometric features.

4 Validation

Comparison to a number of airfoils has been conducted, with varying degrees of severity and a variety of N values. Four airfoils have been chosen here to illustrate the effectiveness of the developed module. These are:

- Drela AG03
- NACA2412
- Drgnfly canard
- 2032c

For each airfoil a description is given, followed by results of applying the smoothing function.

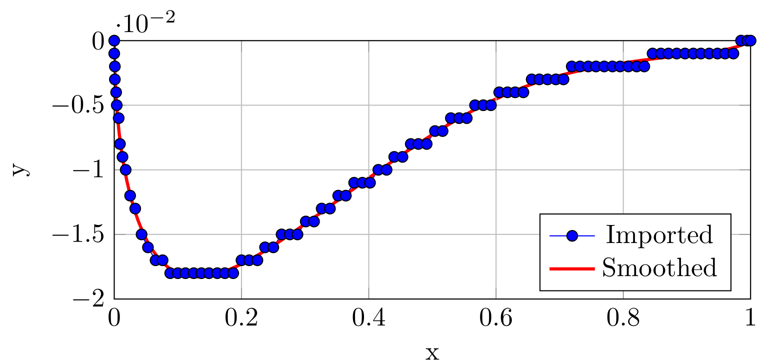

Figure 8: Drela AG-03 Airfoil: Pressure side detail.

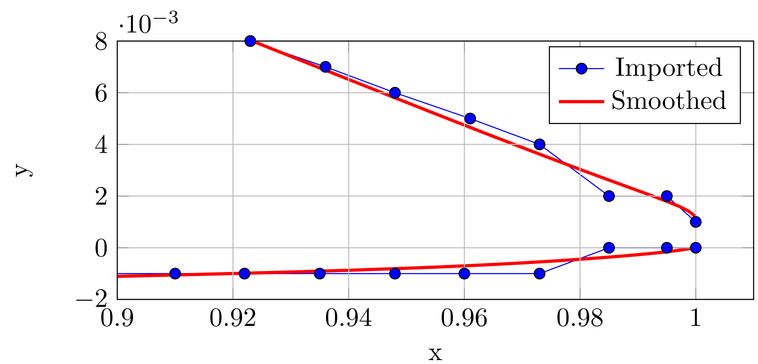

Figure 9: Drela AG-03 Airfoil: Trailing edge detail.

Drela AG03 airfoil

This is an airfoil customised within XFoil and has an essentially flat pressure side. The imported data points for this airfoil are particularly course, as can be seen in the following plots. The imported and smooth values are shown in Figure 7. It can be seen that the discontinuities in the surface have been avoided to a relatively high degree. The general shape of the airfoil has been very well preserved. Figure 8 illustrates in detail the pressure side of the airfoil and the high degree to which sharp corners are avoided, and the original shape of the airfoil is well reproduced. Figure 9 illustrates how the trailing edge has been well reproduced. The discontinuity at the third and third-from-last coordinates has been removed. A small curvature near the suction side trailing edge has been produced, it is however hoped this affects the aerodynamic analysis negligibly.

NACA2412 airfoil

A member of the family of NACA airfoils, this airfoil is asymmetrical and has a blunt trailing edge. The imported coordinates for this airfoil are particularly smooth. This airfoil was hence chosen in order to be able to demonstrate that for already smooth coordinates, the smoothing function will essentially reproduce the coordinates. The ability of the smoothing function to recreate blunt edges was demonstrated previously. For this reason, this profile has been extrapolated to have a sharp trailing edge. Figure 10 illustrates how the profile is essentially recreated perfectly. The trailing edge and leading edge are both recreated with no appreciable deviation from the imported coordinates, as was desired. Figure 11 displays how the sharp trailing edge is reproduced.

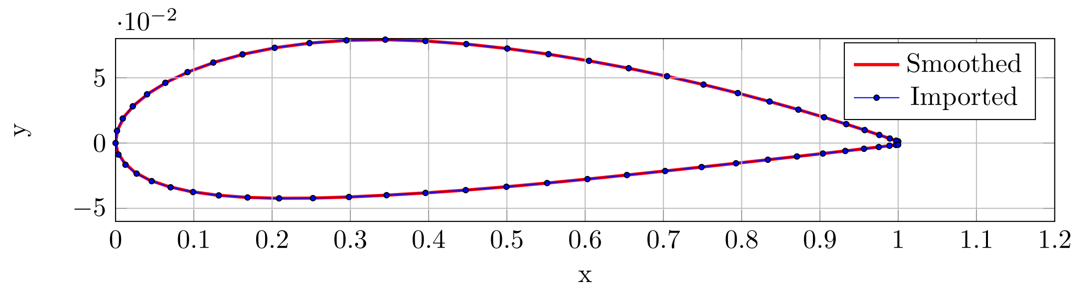

Figure 10: NACA2412 Profile.

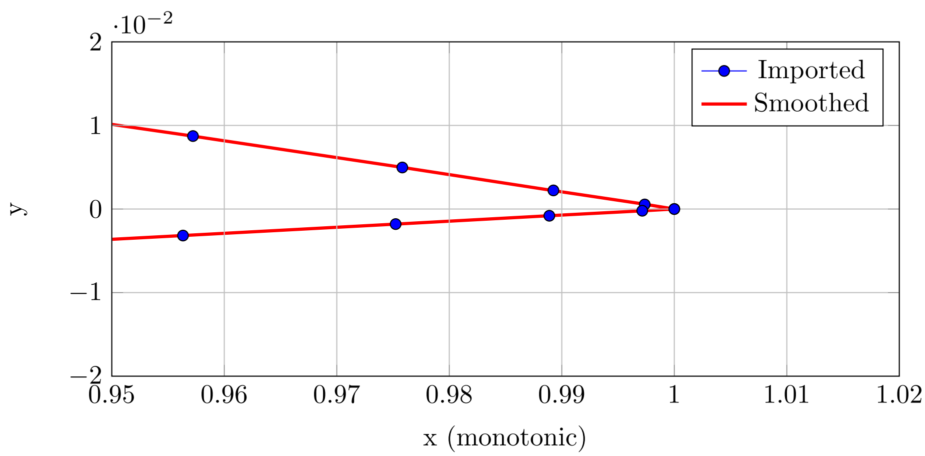

Figure 11: NACA2412: Trailing edge detail.

Drgnfly airfoil

This airfoil was chosen due to the high amount of curvature at the nose. Figure 12 shows the imported and smoothed coordinates. Although for essentially the entire profile, the coordinates were recreated perfectly, there is a very slight deviation in the regions of high curvature (near the leading edge), these can be seen in Figure 13. Although unlikely to impact heavily on the inviscid potential flow solution, these deviations may cause the boundary layer to transition at a different point.

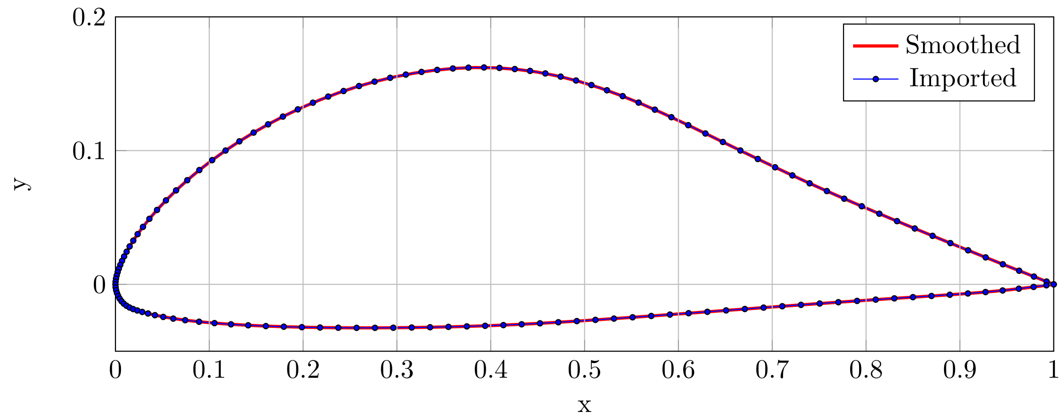

Figure 12: Drgnfly canard airfoil

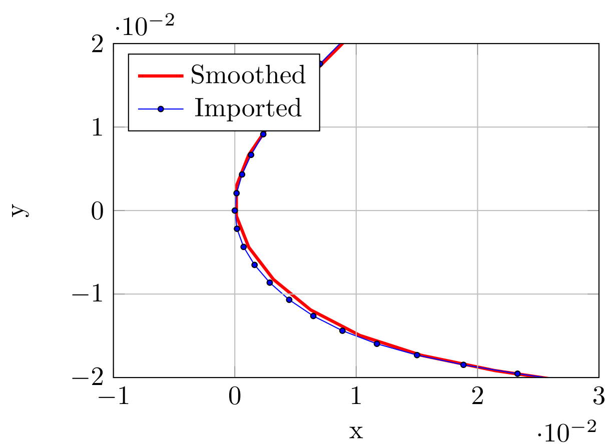

Figure 13: Drgnfly canard airfoil, leading edge detail.

2032c airfoil

This airfoil was chosen to demonstrate the ability of the smoothing function to reproduce a surface well for an airfoil with high curvature and very few coordinates (in this case $N = 35$). The imported and smoothed coordinates are illustrated in Figure 14. As can be seen almost the entire surface is again almost perfectly recreated with no appreciable difference. Inspection of Figure 15 shows that again in this case, at the section of maximum curvature, a slight deviation is seen to occur. It should be highlighted here that this seems to be an

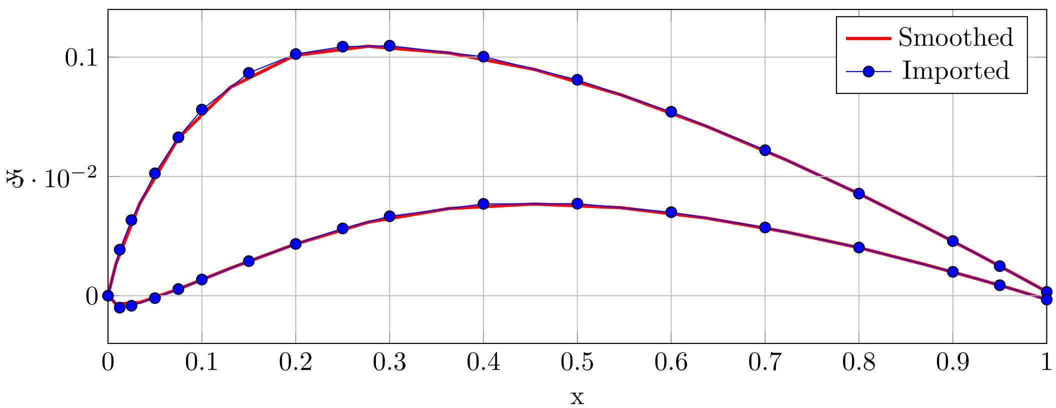

Figure 14: 2032c airfoil.

issue and can lead to irregularities in the calculation of surface curvature. A solution to this is to increase the number of refined coordinates. By re-discretising the surface with an increased number of coordinates, in this case $N_2 = 160$, the sharp edges produced by the smoothing function (which, in fact are simply the reproduced sharp edges of the imported coordinates) can be smoothed out to a certain degree. The result of this is also illustrated in Figure 15.

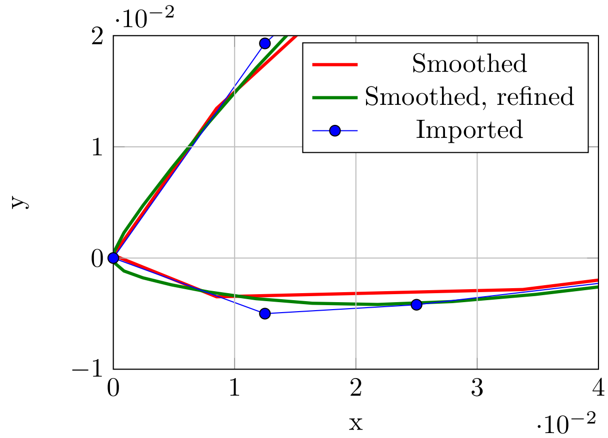

Figure 15: 2032c airfoil, leading edge detail. Notice the curvature introduced by increasing the surface refinement from $N = 35$ to $N_2 = 160$.

Conclusion

The surface of all tested airfoils appeared to be reproduced excellently. For all airfoils, the trailing edge was found to be reproduced very well, regardless of whether it was sharp or blunt. As was before stated, this has a critical influence on aerodynamic behaviour and lift production. In some cases sections of the airfoil with high surface curvature (generally the leading edge) produced very slight deviations upon being passed through the smoothing function. It should be assumed that the effect on inviscid solutions is quite small. Transition in the boundary layer however is highly sensitive to adverse pressure gradients, it should hence not be assumed that these deviations have no effect of the formation of the boundary layer (especially on the suction side of the airfoil). It is suggested that the sensitivity of these deviations is checked with a fully viscid solver such as X-Foil to assess the impact of pressure predictions. As the first implementation of the airfoil smoother however, and as the Airfoil Evaluator is, at this stage, only active as an inviscid solver, the smoothing function appears to be well validated.

References

[1] Airfoil Evaluator http://www.airfoilevaluator.com/home.php

[2] Nova Scientia http://www.novascientia.net/