

# A GUI for computing flows past general airfoils

S. T. Simakov, A. S. Dostovalova and E. O. Tuck

Departments of Applied and Pure Mathematics,  
University of Adelaide, Adelaide, SA 5005

## Abstract

We report a GUI for computing flows past airfoils of arbitrary shape. The tool consists of a single m-file and integrates computer-aided design of airfoils, choice of flow parameters, computation and visualisation of the outer flow, and generation of data for a boundary-layer module. The numerical algorithm employs the Theodorsen-Naiman method, which gives an accurate approximation to the exact flow and is based on conformal mapping and the use of finite trigonometric series. We suggest visual procedures for convergence improvement, and discuss limitations of the technique. The program could be an aid to teaching Aerodynamics and/or Complex Analysis.

## 1 Introduction

Computation of an unseparated flow past an airfoil using Prandtl's boundary layer equations requires determination of an ideal (outer) flow. This is often done using conformal mapping of the outer-airfoil region onto the exterior of a standard domain (usually a disc) for which an analytic form of the solution is known. There are cases when this approach leads to exact solutions (flows past the Zhukovskii profiles, Karman-Trefftz profiles or some simple polygonal airfoils) but commonly the solution is approximate in the sense that it describes an exact flow past a shape approximating the airfoil. When, for example, the Schwarz-Christoffel transformation is used [6] such a shape is a polygon to which the wing section is reduced. If the ideal flow solution is needed for boundary-layer computation, then, in addition to closely fitting the airfoil, the approximating shape must preserve its smoothness properties. This requirement stems from the fact that the boundary-layer separation

algorithms are sensitive to singularities of the boundary, or, in physical terms, to fast changes in the pressure gradient, and it is necessary to avoid false predictions of separation from corners which are in fact numerical artefacts.

The Theodorsen-Naiman technique [9, 10, 7, 8, 1] allows us to explicitly construct a mapping that meets such a requirement for a wide class of airfoils. We report a tool that implements this procedure in MATLAB, which provides simplicity of coding and flexibility in handling graphics and mathematics and seems to be one of the most suitable environments for such algorithms.

In Section 2 we provide a concise and self-contained theoretical background of the method. In Section 3, a summary and details of how we implement the technique are given. We also discuss limitations of the method. One of limitations is the requirement that the near-circle obtained from the airfoil by an auxiliary Zhukovskii mapping must be a star-shaped region. Another limitation is connected with the restriction on the geometry of the trailing edge, which must be a cusp of the first kind. Violation of this restriction may destroy a one-to-one correspondence between angular parameters in the circle and the near-circle planes and result in an inability to satisfy the Kutta-Zhukovskii condition. The trailing-edge geometry limitation is not present if a preliminary Karman-Trefftz transformation is used for mapping to a near-circle plane. Section 4 outlines features of the graphical user interface.

## 2 Ideal flow past general airfoils

We will use a methodology developed and published in [9, 10, 7, 8, 1]. Specifics of our procedure require notational adjustments and the theoretical material below is intended to provide this.

Suppose that an airfoil contour is given in the  $\zeta$ -plane. One of our goals will be to find a transformation  $z(\zeta)$ , or equivalently  $\zeta(z)$ , that satisfies

$$\zeta(z) = ze^{i\alpha_1} + O(1) \quad \text{as } |z| \rightarrow \infty, \quad (1)$$

for some  $\alpha_1$  and conformally maps the exterior of the airfoil onto the exterior of a circle  $|z| < R$ . The flow past such a circle is described by the complex velocity potential

$$w(z; U_\infty, \alpha_0, R, \Gamma) = U_\infty \left( ze^{-i\alpha_0} + \frac{R^2 e^{i\alpha_0}}{z} \right) + \frac{i\Gamma}{2\pi} \log \frac{ze^{-i\alpha_0}}{R}. \quad (2)$$

The condition (1) preserves uniformity of the stream at infinity, so the complex

velocity potential for the flow in the plane of the airfoil has the form

$$f(\zeta) = w(z(\zeta)) \equiv w(z(\zeta); U_\infty, \alpha_0, R, \Gamma), \quad (3)$$

To determine appropriate values of the parameters  $U_\infty$ ,  $\alpha_0$  and  $\Gamma$  in (2), consider the complex velocity

$$f'(\zeta) = \frac{w'(z(\zeta))}{\zeta'(z)} = U_\infty e^{-i(\alpha_1 + \alpha_0)} + \frac{i\Gamma}{2\pi} \zeta^{-1} + O(\zeta^{-2}). \quad (4)$$

If  $\alpha$  is the uniform flow direction angle in the  $\zeta$ -plane, then we must choose  $\alpha_0 = \alpha - \alpha_1$ . Parameter  $U_\infty$  must be set to the magnitude of the uniform flow at infinity. Similarly, formula (4) shows that the circulation  $\Gamma$  of the flow in the  $z$ -plane coincides with the circulation of the flow in the physical  $\zeta$ -plane. Choice of this parameter's value is decided by the Kutta-Zhukovskii condition, and will be discussed later.

The required mapping  $z(\zeta)$  can be obtained in three steps.

- 1) First the Zhukovskii transformation

$$\zeta = \tilde{z} + \frac{a^2}{\tilde{z}} \quad (5)$$

maps the flow domain onto a domain in an auxiliary  $\tilde{z}$ -plane. As a result of this transformation, a cusped part of the boundary is mapped onto a smooth segment, and the trailing edge singularity is removed. Mapping (5) also turns an oblong shape that the airfoil commonly has onto a rounded domain which is referred to as a *near-circle* [1].

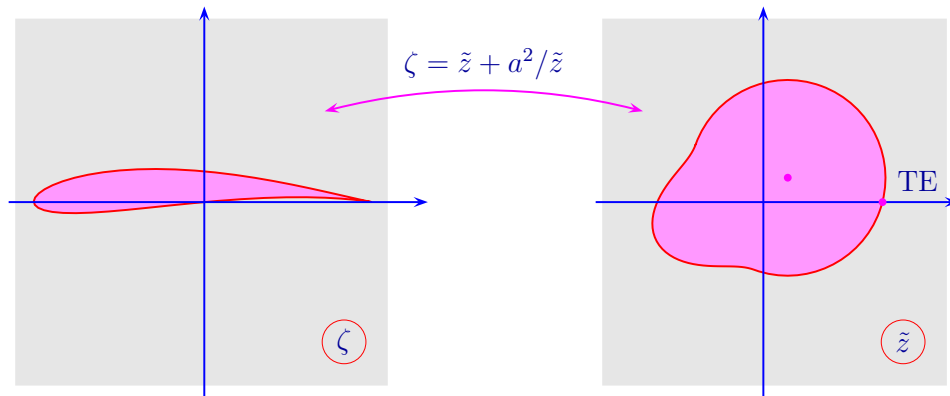


Figure 1: Zhukovskii mapping

By changing to a new variable

$$\bar{z} = (\tilde{z} - \varepsilon)e^{-i\alpha_1}, \quad (6)$$

or, equivalently, translating and rotating the coordinate system, we make its origin coincide with the “center” of the near-circle and its positive real axis intersect the image of the trailing edge. This transformation is needed for computational convenience.

2) The second step is to apply the mapping

$$\bar{z} = z \exp\{\omega(z)\}, \quad (7)$$

where  $\omega(z) = \omega(z; R)$  is analytic if  $|z| > R$  and  $\omega(z) \rightarrow 0$  as  $|z| \rightarrow \infty$ . This form of relation between  $z$  and  $\bar{z}$  and properties of mapping (5) guarantee that the condition (1) is met.

3) Combining the first two steps we obtain the composite mapping  $\zeta = \zeta(\bar{z}(z))$  and the ideal flow past the airfoil.

There is a certain degree of freedom in the choice of auxiliary parameters  $a$  and  $\varepsilon$  and we use this freedom to improve convergence of the scheme. These parameters cannot be chosen arbitrarily however.

Suppose the chord length of the airfoil is  $L$  and let  $a$  be a little less than  $L/4$ . Position the airfoil so that the two conditions are satisfied:

- a) the trailing edge is at  $\zeta = 2a$ ,
- b) the point  $\zeta = -2a$  is inside the airfoil.

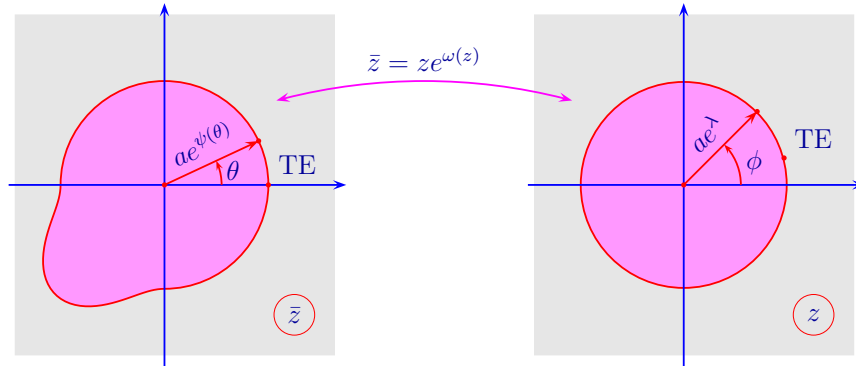


Figure 2: Correspondence between the boundary points

Let  $\theta$  and  $\phi$  be polar angles in the  $\bar{z}$ - and  $z$ -planes respectively. The boundaries of the near-circle in the  $\bar{z}$  plane and its image in the  $z$ -plane can be represented as

$$\bar{z} = ae^{\psi(\theta)+i\theta} \quad \text{and} \quad z = ae^{\lambda+i\phi}. \quad (8)$$

Here  $\psi(\theta)$  is a known function, which we determine from (5) and (6). If  $\zeta = \zeta(t)$  is a given parametric form of the airfoil, then  $\bar{z}(t)$  found from (5) and (6) is a parametrization of the near-circle. Determine  $t(\theta)$  from the equation  $\arg[\bar{z}(t)] = \theta$  and write

$$ae^{\psi(\theta)} = |\bar{z}(t(\theta))| \Rightarrow \psi(\theta) = \log(|\bar{z}(t(\theta))|/a). \quad (9)$$

Note that  $\lambda$  is a constant yet to be chosen, and  $R = ae^\lambda$ .

Correspondence under the mapping (7) between the boundary points of the  $\bar{z}$  and  $z$  domains may be expressed in terms of the dependence  $\theta = \theta(\phi)$ , for which we must have

$$ae^{\psi(\theta(\phi))+i\theta(\phi)} = ae^{\lambda+i\phi+\omega(ae^{\lambda+i\phi})}. \quad (10)$$

It turns out that knowledge of  $\theta(\phi)$  allows us to determine  $\omega(z)$ . Here we reproduce the Laurent series demonstration [9, 10, 1] of this fact. An alternative proof uses the Schwarz formula.

Since  $\omega(z)$  is analytic in the domain external to the circle  $z = Re^{i\phi} = ae^{\lambda+i\phi}$  and vanishes at infinity we can write

$$\omega(z) = \sum_{m=1}^{\infty} (a_m + ib_m)(R/z)^m. \quad (11)$$

It follows from (10) that

$$\psi - \lambda + i(\theta - \phi) = \sum_{m=1}^{\infty} (a_m + ib_m)(\cos m\phi - i \sin m\phi),$$

and so

$$\psi - \lambda = \sum_{m=1}^{\infty} (a_m \cos m\phi + b_m \sin m\phi), \quad (12)$$

$$\theta - \phi = \sum_{m=1}^{\infty} (b_m \cos m\phi - a_m \sin m\phi). \quad (13)$$

It follows from (12) that

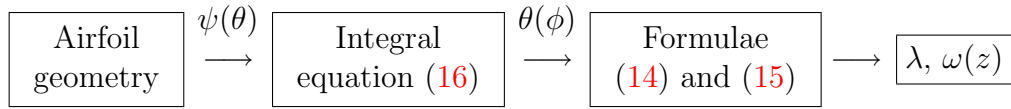
$$\lambda = a_0 \equiv \frac{1}{2\pi} \int_0^{2\pi} \psi(\theta(\phi)) d\phi, \quad (14)$$

$$\begin{Bmatrix} a_m \\ b_m \end{Bmatrix} = \frac{1}{\pi} \int_0^{2\pi} \psi(\theta(\varphi)) \begin{Bmatrix} \cos m\varphi \\ \sin m\varphi \end{Bmatrix} d\varphi. \quad (15)$$

Substitute (15) into (13), and use Dirichlet's kernel and the Riemann–Lebesgue lemma to obtain the integral equation for  $\theta(\phi)$

$$\theta(\phi) - \phi = \frac{1}{2\pi} \text{p. v.} \int_0^{2\pi} \cot \left( \frac{\varphi - \phi}{2} \right) \psi(\theta(\varphi)) d\varphi. \quad (16)$$

The strategy for constructing  $\omega(z)$  can therefore be represented as



There is a semi-inverse approach when either  $\psi$  or  $\theta$  are specified in terms of  $\phi$ . Expansion into the Fourier series (12) or (13) gives the coefficients  $\{a_m, b_m\}$  and therefore  $\lambda$  and  $\omega(z)$ . However, when such an approach is used, the shape of the airfoil cannot be considered as the problem input, but must be recovered once  $\omega(z)$  is found.

We start the next section by reviewing Naiman's method for approximate solution of (16). This method is based on the properties of finite trigonometric expansions and uses the series derivation of (16) as a prototype.

### 3 Naiman's method and implementation remarks

#### Naiman's method

Consider a function  $\Psi(\phi)$  that represents a finite trigonometric series of the form:

$$\Psi(\phi) = A_0 + \sum_{m=1}^n (A_m \cos m\phi + B_m \sin m\phi), \quad (17)$$

where  $B_n$  is taken to be zero.

If  $\Psi(\phi)$  is given by (17), then conditions

$$\Psi(k\pi/n) = \psi_k \quad k = 0, \dots, 2n-1 \quad (18)$$

are equivalent to

$$\begin{aligned} A_0 &= \frac{1}{2n} \sum_{k=0}^{2n-1} \psi_k , \\ A_m &= \frac{1}{n} \sum_{k=0}^{2n-1} \psi_k \cos m \frac{k\pi}{n} \quad (m = 1, \dots, n) , \\ B_m &= \frac{1}{n} \sum_{k=0}^{2n-1} \psi_k \sin m \frac{k\pi}{n} \quad (m = 1, \dots, n) . \end{aligned} \quad (19)$$

Now introduce

$$\epsilon(\phi) \equiv \sum_{m=1}^n (A_m \sin m\phi - B_m \cos m\phi)$$

and consider this function at  $\phi = k'\pi/n$  ( $k' = 0, \dots, 2n-1$ ). It can be shown [8, 1] that

$$\epsilon(\phi = k'\pi/n) = -\frac{1}{n} \sum_{k=0}^{2n-1} \psi_k \sigma(k - k') , \quad (20)$$

$$\text{where } \sigma(K) \equiv \begin{cases} \cot \frac{K\pi}{2n} , & K \text{ odd,} \\ 0 , & K \text{ even.} \end{cases}$$

Now let  $\phi_j \equiv j\pi/n$ ,  $j = 0, \dots, 2n-1$ , and consider a finite sequence of numbers  $\{\theta_k\}$  ( $k = 0, \dots, 2n-1$ ) from  $[0, 2\pi)$  that satisfies

$$\theta_j - \phi_j = \frac{1}{n} \sum_{k=0}^{2n-1} \psi(\theta_k) \sigma(k - j), \quad j = 0, \dots, 2n-1. \quad (21)$$

Let  $\psi_j = \psi(\theta_j)$  and use (19) to define  $A_m$  and  $B_m$  and hence

$$\Omega(z) = \sum_{m=1}^n (A_m + iB_m)(ae^{A_0}/z)^m .$$

The image of the circle  $z = ae^{A_0+i\phi}$  ( $\phi \in [0, 2\pi)$ ) under such a transformation can be parametrized as

$$\bar{z} = ae^{\psi^*(\theta)+i\theta} . \quad (22)$$

The superscript  $\star$  was used here to indicate that in general  $\psi^*(\theta)$  and  $\psi(\theta)$  are different functions. Parameters  $\phi$  and  $\theta$  corresponding to the same point on the circle are connected as

$$\theta = \theta^*(\phi) \equiv \arg [\bar{z} (z = ae^{A_0+i\phi})]$$

Rewrite (22) as

$$\bar{z} = ze^{\psi^*(\theta) - A_0 + i(\theta - \phi)}.$$

Using  $\bar{z} = z \exp\{\Omega(z)\}$  with  $z = ae^{A_0 + i\phi}$  we have

$$\psi^*(\theta^*(\phi)) = A_0 + \sum_{m=1}^n (A_m \cos m\phi + B_m \sin m\phi), \quad (23)$$

$$\theta^*(\phi) - \phi = \sum_{m=1}^n (B_m \cos m\phi - A_m \sin m\phi). \quad (24)$$

It follows from (23) and the way the coefficients of this finite trigonometric series have been formed that

$$\psi^*(\theta^*(\phi_j)) = \psi_j \equiv \psi(\theta_j). \quad (25)$$

Furthermore, it follows from (20) that

$$\theta^*(\phi_j) - \phi_j \equiv \sum_{m=1}^n (B_m \cos m\phi - A_m \sin m\phi) = \frac{1}{n} \sum_{k=0}^{2n-1} \psi_k \sigma(k - j).$$

Since  $\theta_j$  satisfy

$$\theta_j - \phi_j = \frac{1}{n} \sum_{k=0}^{2n-1} \psi(\theta_k) \sigma(k - j) = \frac{1}{n} \sum_{k=0}^{2n-1} \psi_k \sigma(k - j),$$

we obtain

$$\theta^*(\phi_j) - \phi_j = \theta_j - \phi_j \quad \Rightarrow \quad \theta^*(\phi_j) = \theta_j.$$

This along with (25) gives

$$\psi^*(\theta_j) = \psi(\theta_j) \quad (j = 0, \dots, 2n - 1).$$

Hence the shape  $\bar{z} = a \exp(\psi^*(\theta) + i\theta)$  coincides with the required near-circle  $\bar{z} = a \exp(\psi(\theta) + i\theta)$  at  $\theta = \theta_j$  ( $j = 0, \dots, 2n - 1$ ) thus approximating it. As a result the approximate airfoil will coincide with the given airfoil at points  $\zeta(a \exp(\psi(\theta_j) + i\theta_j))$ , where  $\zeta(\bar{z})$  is defined by (5) and (6).

If the flow is uniform at infinity with magnitude  $U_\infty$  and direction angle  $\alpha$ , then its complex velocity potential is

$$f(\zeta) = w(z(\zeta); U_\infty, \alpha_0 \equiv \alpha - \alpha_1, ae^{A_0}, \Gamma).$$

Circulation  $\Gamma$  is chosen so as to satisfy the Kutta-Zhukovskii condition (also referred to as the Chaplygin condition [4, 5]):



for  $f'(\zeta)$  to be finite at the trailing edge, where  $\zeta'(z) = 0$ , the rear stagnation point of the  $z$ -flow must correspond to the trailing edge of the airfoil.

Suppose that  $z_T = ae^{A_0+i\phi_T}$  is a point corresponding to the trailing edge  $\zeta = 2a$ . Then we must have:

$$U_\infty \left( e^{-i\alpha_0} - \frac{a^2 e^{2A_0+i\alpha_0}}{a^2 e^{2A_0+i2\phi_T}} \right) + \frac{i\Gamma}{2\pi} \frac{1}{ae^{A_0+i\phi_T}} = 0,$$

or

$$\Gamma = 4\pi ae^{A_0} U_\infty \sin(\alpha_0 - \phi_T) = 4\pi ae^{A_0} U_\infty \sin(\alpha - \alpha_1 - \phi_T). \quad (26)$$

The angle  $\phi_T$  in (26) is found from the equation:

$$\phi + \sum_{m=1}^n (B_m \cos m\phi - A_m \sin m\phi) = 0. \quad (27)$$

## Implementation remarks

The key step is to determine  $\{\theta_k\}$  by solving the non-linear system (21) with  $\phi_j = j\pi/n$  and  $\psi(\theta)$  assumed to be known. Formula (9) shows how  $\psi(\theta)$  is determined in the case when the dependence  $t = t(\theta)$  connecting the airfoil's boundary parameter with the polar angle in the  $\bar{z}$  plane can be easily established. If the boundary of the airfoil is specified as a sequence of coordinates we define  $\psi(\theta)$  using cubic interpolation.

We solve (21) using the iterative procedure:

$$\begin{cases} \theta_j^{[\ell+1]} = \phi_j + \frac{1}{n} \sum_{k=0}^{2n-1} \psi(\theta_k^{[\ell]}) \sigma(k-j), \\ \theta_j^{[0]} = \phi_j \end{cases} \quad (28)$$

where,  $j = 1, \dots, 2n-1$  for each  $\ell = 0, \dots$

Our numerical procedure requires an explicit form for the transformation  $\tilde{z} = \tilde{z}(\zeta)$  defined by (5). This form is

$$\tilde{z} = \frac{1}{2}(\zeta + \sqrt{\zeta^2 - 4a^2}), \quad (29)$$

where the used branch of  $\sqrt{\zeta^2 - 4a^2}$  satisfies the following conditions:

- (a)  $\sqrt{\zeta^2 - 4a^2} = \zeta + O(\zeta^{-1})$  as  $|\zeta| \rightarrow \infty$ ;
- (b) its Riemann surface is cut along a line which connects the branching points  $\zeta = \pm 2a$  and lies entirely inside the airfoil.

Transformation (6) has a twofold purpose. Firstly, as we noted earlier, it enables us to center the origin of the new coordinate system on the image of the airfoil, which helps to achieve a better convergence of the iterative procedure. Secondly, this simple transformation can in some cases facilitate the choice of the coordinate system for which the  $\theta$ -parametrization of the boundary in (8) becomes possible. If the image of the airfoil under (29) is not a star-shaped region, another mapping must be used instead of (6).

Mapping (5) applied to a slender airfoil may result in a shape with a protuberant segment corresponding to the nose if the parameter  $a$  is taken from the permissible interval indiscriminately. This can worsen convergence or even make it impossible no matter what  $\varepsilon$  is chosen in (6). The same difficulty arises when the airfoil is not properly oriented with respect to the axes of  $\zeta$ . (Note however that the orientation of the airfoil with respect to the flow is given and so may not be altered.)

Convergence of the process (28) thus strongly depends on the choice of the characteristics: the value of  $a$  in (5), parameter  $\varepsilon$  and the orientation of the airfoil in the  $\zeta$ -plane. Search for a suitable combination of these parameters can be carried out with a tool that uses them as an input and instantly provides a visual feedback by displaying the resulting image of the airfoil in the  $\tilde{z}$ -plane. We organize this search as follows.

Suppose that the chord of the airfoil is  $L$ . Let  $\varepsilon_{\text{norm}}$  be a complex parameter. The mapping

$$\zeta_{\text{norm}} = \tilde{z}_{\text{norm}} + 1/\tilde{z}_{\text{norm}}$$

transforms the circle  $\tilde{z}_{\text{norm}} = \varepsilon_{\text{norm}} + |1 - \varepsilon_{\text{norm}}|e^{it}$  ( $t \in [0, 2\pi)$ ), into a Zhukovskii airfoil. We denote its chord by  $l(\varepsilon_{\text{norm}})$ . Now taking

$$a = a(\varepsilon_{\text{norm}}) = L/l(\varepsilon_{\text{norm}}), \quad \varepsilon = a\varepsilon_{\text{norm}}$$

and considering the image of the circle  $\tilde{z} = \varepsilon + |a - \varepsilon|e^{i\theta}$  ( $\theta \in [0, 2\pi)$ ) under the transformation (5) we obtain a Zhukovskii airfoil whose trailing edge is at  $\zeta = 2a$  and whose chord is  $al(\varepsilon_{\text{norm}}) = L$ . Displaying this and the given airfoil in the same axes with their trailing edges juxtaposed at  $\zeta = 2a(\varepsilon_{\text{norm}})$  we can observe the discrepancy between them and control it by changing  $\varepsilon_{\text{norm}}$ . The latter can be most conveniently done by making  $\varepsilon_{\text{norm}}$  depend on the position of a movable object on

the same screen (see Figure 3). The smaller the discrepancy between the noses of the airfoils is achieved the better becomes the chance that the image of the given airfoil under mapping (5) is an acceptable near-circle shape. A separate screen containing images of both airfoils in the  $\bar{z}$ -plane provides an additional control over the described manipulations. Note that in such a procedure the required value for  $\varepsilon$  is obtained automatically.

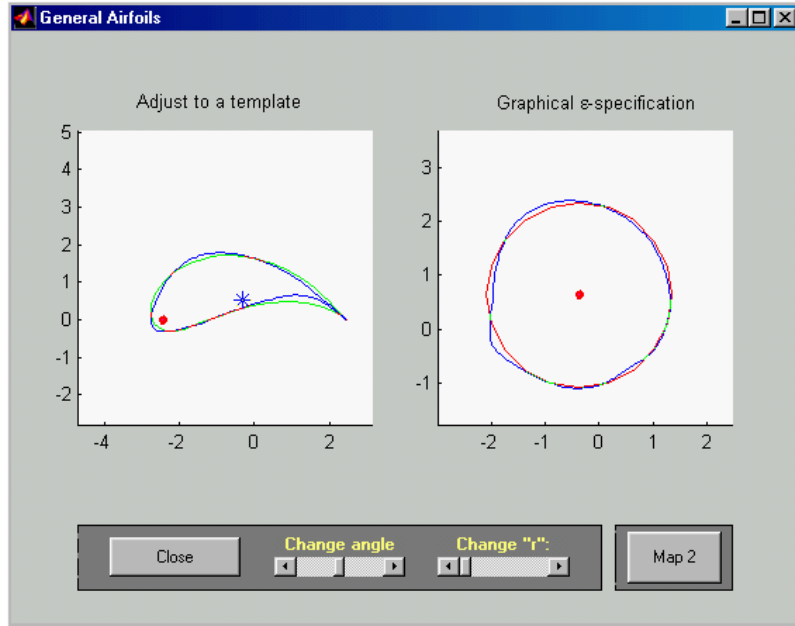


Figure 3: Choice of optimal mapping parameters using Zhukovskii's templates

When the system (21) has been solved we use (19) to define  $A_m$  and  $B_m$ . The transformation  $\zeta(z)$  now can be written as

$$\zeta = \zeta(z) = \varepsilon + \bar{z}(z)e^{i\alpha_1} + \frac{a^2}{\varepsilon + \bar{z}(z)e^{i\alpha_1}}, \quad (30)$$

where

$$\bar{z}(z) = z \exp \left\{ \sum_{m=1}^n (A_m + iB_m) \left( \frac{z}{ae^{A_0}} \right)^{-m} \right\}.$$

The flow past the airfoil is given by formula (3) in which  $\alpha_0 = \alpha - \alpha_1$  and  $\Gamma$  is defined by (26).

We have already noted that the method is applicable if the boundary of the near-circle admits parametrization in terms of the polar angle  $\theta$ . It is also essential that the function

$$\theta^*(\phi) = \phi + \sum_{m=1}^n (B_m \cos m\phi - A_m \sin m\phi)$$

is one-to-one. This property does not simply follow from  $\theta(\phi)$  being one-to-one but also depends on the behaviour of  $\psi(\theta)$ . If the near-circle has *salient* points, then  $\theta^*(\phi)$  is likely to be oscillatory in their neighbourhood no matter how large  $n$  is taken, a behavior akin to the Gibbs phenomenon. Normally these considerations apply only to the point corresponding to the trailing edge of the airfoil. If mapping (5) is used, then, in order for the boundary of the near-circle to be smooth, the trailing edge must be a cusp of the first kind. If the trailing edge represents a finite angle, the Zhukovskii transformation results in a near-circle with a corner point, which leads to oscillations of  $\theta^*(\phi)$ , as shown on Figure 4 and self-intersections of the boundary of the approximating airfoil, on the computational level the procedure failing at the stage of solving equation (27) needed to satisfy the Kutta-Zhukovskii condition.

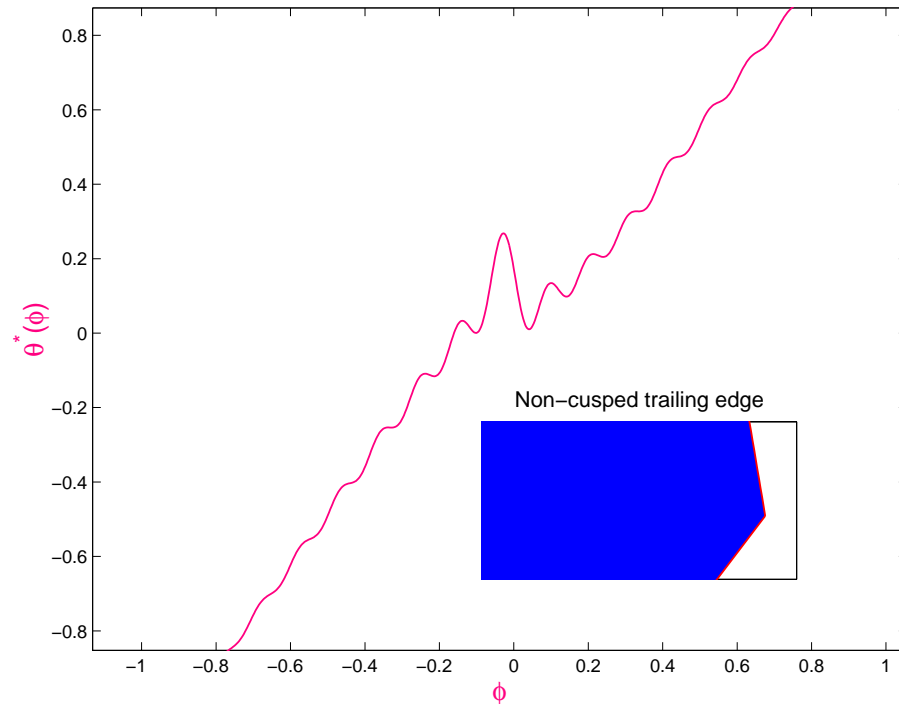


Figure 4: Typical behaviour of  $\theta^*(\phi)$  for non-cusped geometry

The Karman-Trefftz transformation is commonly used (as, for example in [3, 2]) instead of the Zhukovskii mapping to generate a near-circle without a corner point if the trailing edge is arbitrary.

## 4 Main states of the GUI and concluding remarks

We implement the algorithm reviewed in Section 3 in the form of a single m-function. In addition to the segment performing the iterative procedure (28), the tool incorporates interfaces for on-screen design of airfoils and choice of optimal mapping parameters needed for initial approximation. Post-processing options are provided including basic visualisation and storing data in a specific format.

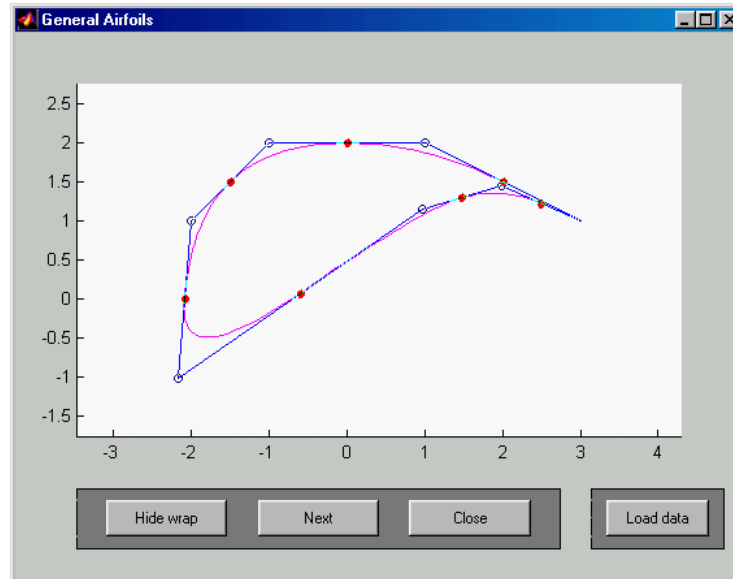


Figure 5: On-screen shape generation

Computer-aided airfoil shape generation uses second-order Bezier curves. Figure 5, representing the initial state of the GUI, shows such curves combined in an airfoil. Their control points are active and can be moved on the screen using the pointer. For the airfoil outline to remain smooth, the joint control points of the adjacent Bezier curves are only allowed to move along the segments between the respective middle control points, which as a result form vertices of a circumscribed polygon. The joint control points are shown as solid disks. Clicking on any of them with

the right button refines the partition of the outline by adding to it a Bezier curve segment in the neighbourhood of the disk.

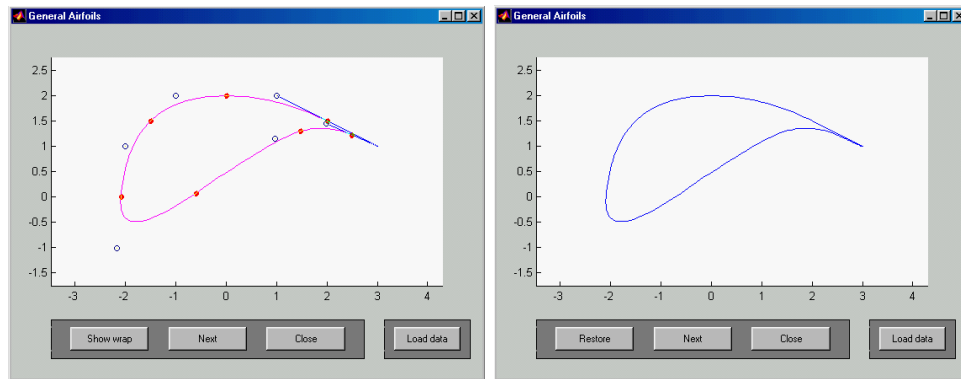


Figure 6: States of the shape generation mode

The sides of the polygon, as well as the control points of the Bezier curves, can be removed at any stage of the shape generation mode as shown in Figure 6.

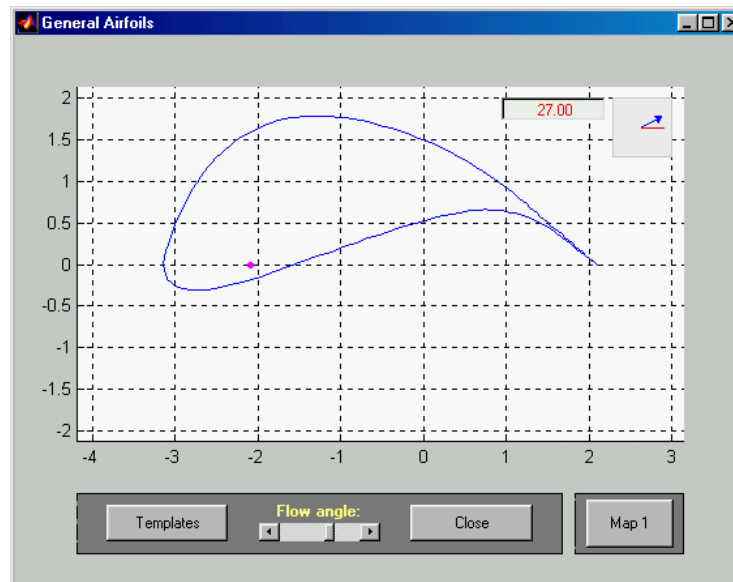


Figure 7: Default orientation of the airfoil

Figure 7 exhibits the state that occurs after the airfoil outline has been generated. The button “Load data” of the previous mode allows to access this state directly if

the wing geometry is provided as an m-function `getwing`. Below we give an example of `getwing` for a non-symmetric Zhukovskii airfoil.

```
function Zeta=getwing
epsilon=-0.1+0.1i;
a=1;
N=100;
Z=epsilon+(a-epsilon)*exp(i*[0:N-1]*2*pi/N);
Zeta=Z+a^2./Z;
```

The airfoil is oriented so that its longest chord is horizontal and it is this orientation with respect to which we shall measure the direction of the uniform flow.

Now we have to specify the mapping parameters  $a$ ,  $\epsilon$  and  $\alpha_1$ . Buttons “Templates” and “Map 1” allow to proceed to one of the two possible states shown in Figures 3 and 8. The method of Figure 3 uses Zhukovskii’s airfoils and is described in Section 3. The state in Figure 8 uses a more straightforward but less effective method, in which the values of  $a$  and  $\epsilon$  are decided on the basis of comparison of the image of the airfoil under the Zhukovskii mapping with circles.

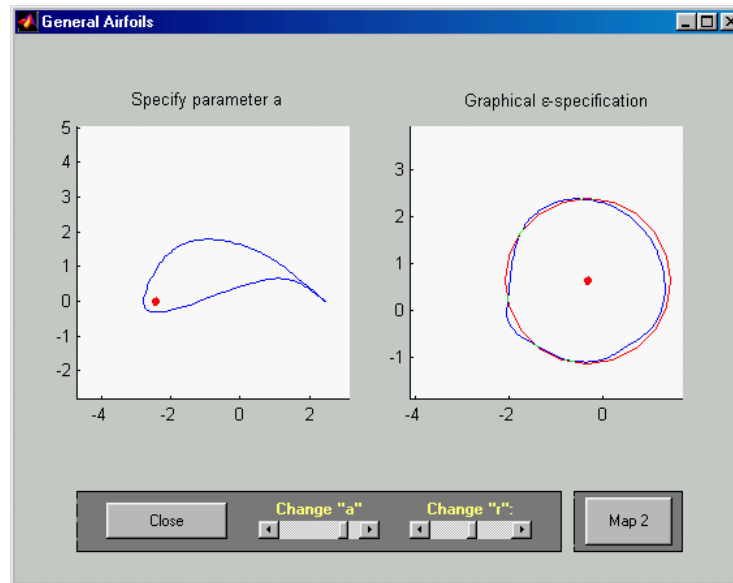


Figure 8: Determination of mapping parameters

Pressing on the button “Map 2” brings the user to the control panel shown in Figure 9.

**Parameters of computation and visualisation**

N	$U_\infty$	$\alpha_{\text{chord}}$
60	1.0000	00.0000
$N_{\text{iter}}$	$\chi$	$\psi\text{-step}$
15	2.5000	0.1000
$N_{\text{TE}}$	$\sigma$	$N_{\text{lines}}$
30	0.0010	2
Save V-data?	$N_{\text{up}}$	$N_{\text{down}}$
Don't save	250	200
Save BLC-data?	$N_{\text{stagn.}}$	$\phi_{\text{step}}$
Don't save	200	0.0050

Proceed

Figure 9: Setting up computation and post-processing parameters

Here one can set up such parameters as the number of iterations, or the number of points the approximating airfoil is to share with the exact airfoil. Visualisation and post-processing parameters can also be entered at this stage.

Pressing the button “Proceed” starts the iterative procedure whose aim is to determine  $\{A_m, B_m\}$ . When these coefficients have been found, the program creates two standard figures as in Figure 10. One of them displays the original and the approximating airfoils to allow their comparison.

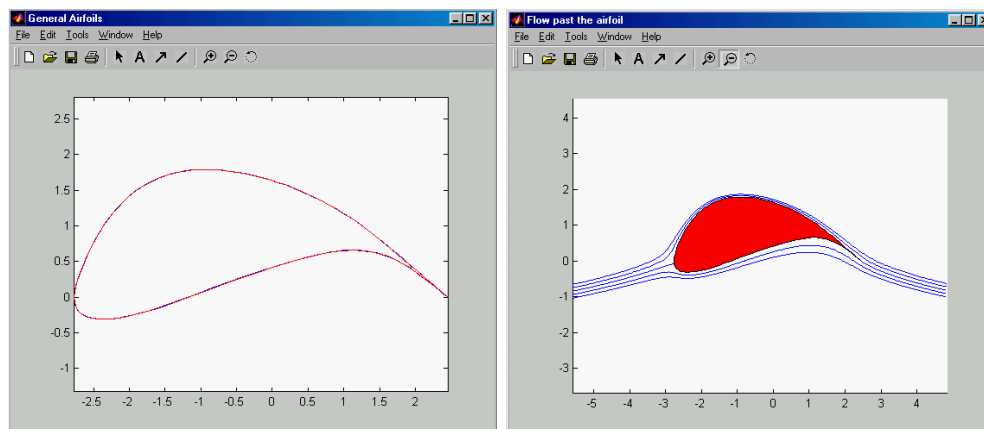


Figure 10: Basic visualisation



Another figure shows a few streamlines of the computed flow that pass near the airfoil. Visualisation provided within the tool is basic and was primarily intended to facilitate detection of possible errors in the code or calculations by examining local behaviour of the flow near the body, in particular near the stagnation point and the trailing edge.

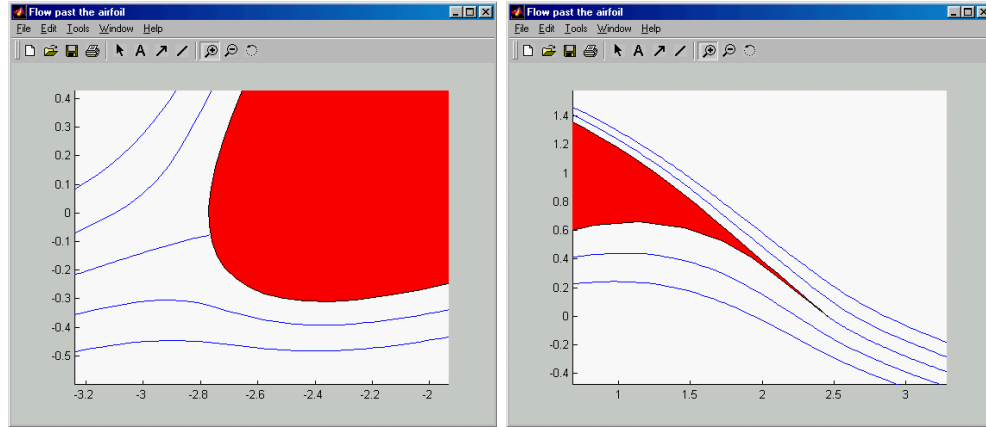


Figure 11: Flow near the stagnation point and the trailing edge

The resulting coefficients  $\{A_m, B_m\}$  can be saved (using the pop-up menu “Save V-data?”) in a mat-file for further interpretation, which could be some form of a 3-D flow as shown in Figure 12 or various types of pressure-loading curves.

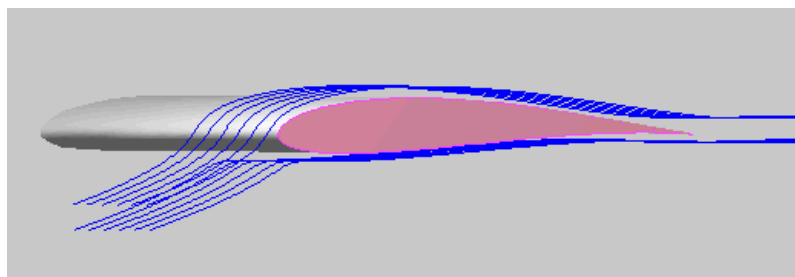


Figure 12: A variant of 3-D visualisation

Surface pressure distribution output can also be saved for boundary-layer computation (BLC) in the format described in [12]. The corresponding BLC module is based on a program written for [11] and currently exists in the form of a C++ code.

## References

- [1] I. H. Abbott and A. E. von Doenhoff. *Theory of wing sections*. Dover, New York, 1959.
- [2] T. Cebeci, R. W. Clark, K. C. Chang, N. D. Halsey and K. Lee. Airfoils with separation and the resulting wakes. *J. Fluid Mech.*, 163:323–347, 1986
- [3] N. D. Halsey. Potential flow analysis of multielement airfoils using conformal mapping. *AIAA J.*, 17:1281–1288, 1979.
- [4] L. D. Landau and E. M. Lifshitz. *Theoretical physics. Vol. VI. Hydrodynamics*. Nauka, Moscow, 1988.
- [5] M. A. Lavrentjev and B. V. Shabat. *Hydrodynamical problems and their mathematical models*. Nauka, Moscow, 1977.
- [6] J. A. Moriarty and E. O. Tuck. Thin airfoils with high-incidence flaps or blunt trailing edges. *Aeronautical J.*, 93:93–99, 1989.
- [7] I. Naiman. Numerical evaluation of the  $\epsilon$ -integral occurring in the Theodorsen arbitrary airfoil potential theory. NACA ARR L4D27a, 1944.
- [8] I. Naiman. Numerical evaluation by harmonic analysis of the  $\epsilon$ -function of the Theodorsen arbitrary potential theory. NACA ARR L5H18, 1945.
- [9] T. Theodorsen. Theory of wing sections of arbitrary shape. NACA Rept. 411, 1931.
- [10] T. Theodorsen and I. E. Garrick. General potential theory of arbitrary wing sections. NACA Rept. 452, 1933.
- [11] E. O. Tuck and A. S. Dostovalova. Airfoil nose shapes delaying leading-edge separation. *Aeronautical J.*, to appear.
- [12] M. J. Werle and R. T. Davis. Incompressible laminar boundary layers on a parabola at angle of attack: A study of the separation point. *Trans. ASME: J. Appl. Mech.*, pages 7–12, March 1972.