PROFOIL

A Multipoint Inverse Airfoil Design Method

User Guide

Version 4
November 22, 2022

Michael S. Selig, Ph.D.
Professor Emeritus
m-selig@illinois.edu

University of Illinois at Urbana-Champaign
Department of Aerospace Engineering
https://m-selig.ae.illinois.edu

1 Introduction

PROFOIL is a computer program for the design of isolated airfoils [1, 2]. *PROFOIL* belongs to the category of so-called “inverse” design methods. This approach is best explained by first introducing the conceptually simpler “direct” method. As the name implies, direct methods (see Fig. 1) are based on the design-by-analysis approach in which a known airfoil shape is analyzed to determine, say, its velocity distribution. The airfoil shape is then manually adjusted until the desired velocity distribution or performance is obtained.

A difficulty with direct methods is that although the designer is often quite interested in the velocity distribution, there is no direct means to control it. Instead the geometry must be tediously manipulated to achieve the desired aerodynamic performance. This process can be quite time consuming and often leads to designer fatigue rather than an optimal design. It is, therefore, more advantageous to specify the velocity distribution from the outset and from that determine the airfoil shape. Such methods are called inverse methods as shown conceptually in Fig. 2.

The current method has it roots in a branch of mathematics called conformal mapping, knowledge of which is not required to use the design method effectively. It is only necessary to know that the airfoil is generated from a circle that is “mapped” to an airfoil as depicted in Fig. 3. In the method, such an airfoil is termed a four-segment airfoil. With *PROFOIL* the velocity distribution can be specified, often by a limited number of parameters. In fact, at a minimum only eight key parameters are needed to define the entire velocity distribution about a four-segment airfoil. Of course, for more complex airfoils (more practical designs) more parameters are needed.

The Joukowski airfoil problem is the simplest application of conformal mapping to airfoil design. The mathematical transformation is fixed and the airfoil is moved to generate

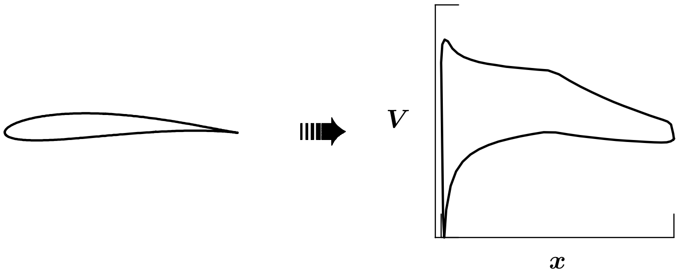

Figure 1: The direct approach to airfoil design.

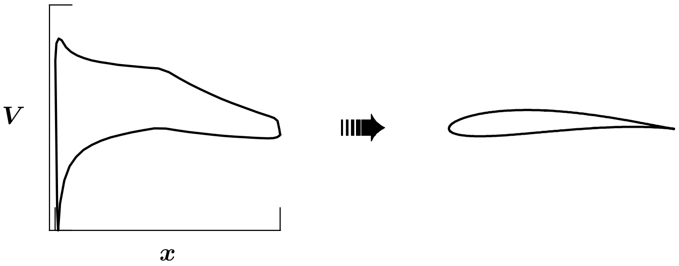

Figure 2: The inverse approach to airfoil design.

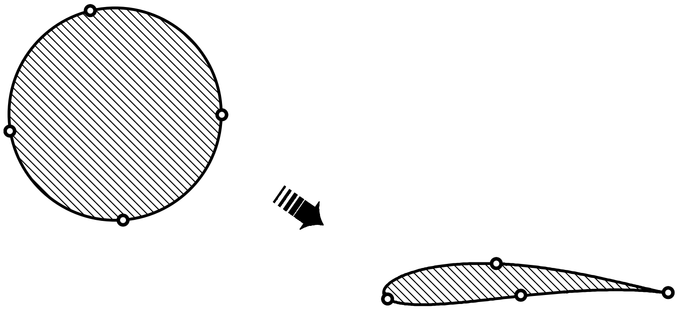

Figure 3: Mapping of a circle to a four-segment airfoil.

different airfoil geometries. An alternative approach is to fix the geometry and change the transformation to obtain different airfoils. This latter strategy is used in the current approach. The transformation, however, is not changed explicitly by the designer, but instead it is determined from the parameters that define the velocity distribution. Thus, conformal mapping is merely used as a bridge to allow the designer to specify the velocity distribution. The details of the mathematics are for the most part transparent to the user.

Another aspect of the method is that "multipoint" design can be performed. The velocity distributions previously shown were for a single angle of attack. When the velocity distribution is specified for a single condition, it is referred to as a "single point" airfoil design problem. Typically, however, good performance is required over a range of angles of attack, say, two different angles of attack. For example, high-lift (high angle of attack) performance may be required as well as low-lift (low angle of attack). The process of prescribing the velocity at two (or more conditions) is referred to as multipoint design. For instance, the velocity distribution can be prescribed for the upper surface at a high angle of attack while simultaneously the velocity distribution can be prescribed for the lower surface at a low angle of attack. This multipoint design feature will be discussed in more detail in the examples.

In what follows, Chapter 2 presents an summary of the capabilities in *PROFOIL*. Chapter 3 gives the typographical conventions used in this User's Guide and also the input file structure. A brief description of the system requirements is given in Chapter **??**, but a separate more detailed instruction sheet is required for the complete installation. Chapter 4 presents several input parameters required for a *PROFOIL* design session and the subsequent generation of the airfoil shape and velocity distributions. The use of Newton iteration during the design process is introduced in Chapters 5 and 6, and the many features are summarized in Chapter 7. Finally, Chapter 8 discusses the input lines for non-constant velocity along a segment.

Finally, it is recommended that the reader obtain a copy of the book *Airfoil Design and Data* by R. Eppler [3]. The current method draws on the foundation developed by Eppler. Since the current method is based on a similar procedure (although the current method has entirely different code with more design features), much can be gleaned from the book's series of airfoils presented along with their corresponding predicted performance characteristics.

2 Features

At the most fundamental level the current method is closely related to the Eppler airfoil design method [3, 4, 5]. No knowledge of the Eppler code is required to use *PROFOIL*, but for those with that background it is helpful to begin by contrasting the two codes. Features common to both the Eppler code and *PROFOIL*

- *Multipoint design capability* allows for matching performance a various conditions
- *Segment descriptions* of the airfoil with associated design angles of attack and arc limits
- *Constant velocity segment velocity distributions*
- *Trailing edge thickness parameter K_S specification* through iteration on various design parameters
- *Cusped trailing-edge airfoils*

Features unique to *PROFOIL*

- *Non-constant segment velocity distributions* can be adjusted to achieve a specified boundary-layer development
- *Specified anchor velocity for a given segment* — Useful when dealing with hydrofoil design when there are limitations on the velocity for cavitation avoidance.
- *Finite trailing-edge angle airfoils*
- *Single parameter prescriptions:*
  - *Maximum thickness*
  - *Maximum camber*
  - *Other geometric constraints*
  - *Pitching moment*
  - *Trailing edge velocity differential* at closure arc limits — Useful when trying to control the amount of aft loading without necessarily having a constraint on the pitching moment.
  - *Zero-lift angle of attack*
  - *Trailing-edge velocity*
- *Segment prescriptions (not discussed in this manual):*
  - *Velocity distribution along the airfoil arc length*
  - *Boundary-layer developments*, such as the shape parameter H_{12}, the Eppler transition criterion curve, skin friction, and more

It must be mentioned that *PROFOIL* is not an all inclusive airfoil-design/analysis package. Inverse design, as discussed, is its strength; analysis is not — it has none. The airfoil performance characteristics (viscous results) are determined with post-processing analysis tools such as the Eppler code, *XFOIL*, and others.

3 Conventions

PROFOIL is a keyword-based program. The input file is a script-type file that contains a journal of commands for either batch mode execution or interactive use. Three basic data-line types are used in the input file as follows.

```
! A 10 segment airfoil
# A 10 segment airfoil
COORD 60
*
```

The line `COORD 60` is a line type that prompts action, either that of storing data for future calculations or initiating either design or analysis calculations. Lines that do not start in the first column are ignored. If a keyword is not recognized, then the line will be echoed to the screen, and user will be given the option of proceeding or stopping. Note that the values following the line are read in unformatted mode; however, values beyond column 50 will not be read. The characters `!` and `#` in the first column denote that the lines are comment lines and should be ignored. The `#` is used more commonly for comment lines. Some lines can have trailing comments, for example

```
COORD 60        !  number of airfoil coordinates
```

Lines that cannot have trailing comments are those with optional data on the line as discussed later. The `*` character in the last line is the stop line, which denotes the end of the input and stops execution. Anything following the stop line is ignored.

In the descriptions of the lines, the format

```
COORD <IARGP>
```

will be used. The string `<IARGP>` denotes that the value following the line will be set to the program variable `IARGP`, in this case the number of airfoil coordinates. Dots . . . indicate that there are one or several intervening lines. A `|` character that separates a string of input parameters (e.g., `<P1> <P2> | <P3>`) means that the values past the `|` are optional and need not be entered. The actual input file does not contain the `|` character. As previously mentioned, lines with optional data cannot have trailing comments unless all of the optional data is entered. Arrays and elements of arrays are denoted, for example, by `AM(.)` and `ALFAS(1)` or `ALFAS(JSEG)`, respectively. `JSEG` is used to denote the jth segment of the airfoil. `ISEG` is the total number of airfoil segments. Finally, a `-` character in this User's Manual denotes the continuation of an input line. The `-`, however, is not used in the actual input data file.

Some rules apply to the sequence of lines. First, required input data (Chapter 4) must precede lines that initiate the design. These required lines can be in any order, with the exception of the `FOIL` lines. Second, some lines initiate the input of a sequence of data. Comment lines are sometimes used in this manual to indicate the beginning and ending of a sequence of data.

Several detailed examples are given in the text. These are denoted by the hand-writer symbol ✍. The examples are numbered, and the corresponding input files are included on the distribution disks. The sealed-envelope symbol ✉ is used to denoted material that is advanced. For beginners, the details will not matter, but for more experienced users the information is sometimes useful. Finally, the snow-flake symbol ❄ is used to denote material that may be of interest to those with experience using the Eppler code.

4 Design Data Lines and Output

In this chapter, the basic design parameters and associated input lines are discussed as well as the coordinate and velocity distribution output files.

4.1 Some Definitions

The variables described here are used to define the velocity distribution from which the airfoil shape is derived. As depicted in Fig. 4, the airfoil is divided into a given number of segments, which are numbered in counter-clockwise order. The arc limits ϕ_1, ϕ_2, ϕ_3 and ϕ_4 on the circle are mapped to the points s_1, s_2, s_3, and s_4 on the airfoil. The arc limit ϕ is measured from the point at ϕ_4. In the method, the minimum number of segments is four — two on the upper surface of the airfoil and two on the lower surface as shown. The circle arc limit ϕ begins at $\phi = 0$ and ends at $\phi = 60$ (rather than 360 deg); thus, the circle is divided into 6 deg sectors. The option exists to use smaller sectors if desired.

Associated with each segment is a corresponding "design angle of attack" α^*, which is best described by way of example. Suppose that the design angle of attack for the second segment α_2^* is set at 10 deg. When the resulting airfoil is then operated at 10 deg, the velocity distribution along the second segment will be *constant*. For the third segment, the design angle of attack can be set to 0 deg, which would give a constant velocity along that segment of 0 deg.

It should be noted that the design angle of attack is referenced to the airfoil zero-lift line (rather than the airfoil chord line). Thus, specifying the design angle of attack allows one to prescribed a constant velocity distribution on a segment for a given lift coefficient, since the lift coefficient of the airfoil is approximately $2\pi\alpha$.

For the first and last segments (segments 1 and 4 in the example), termed the recovery regions, the velocity distributions are not constant when the airfoil operates at the corresponding design angles of attack. Moreover, the design α^*-values in these regions have little effect on the velocity distributions. In part, the recovery velocity distributions are determined by the method so that the airfoil closes at the trailing edge. This is a requirement of the method. (In fact, any inverse method must allow for some degrees of freedom in the velocity distribution so that (1) the airfoil trailing edge closes and (2) the airfoil velocity distribution is consistent with the freestream.)

4.2 Required Design Data Lines

The required input lines for design are as follows

```
COORD <IARGP>
FOIL <AM(1)> <ALFAS(1)>       | <JSEG>
...
FOIL <AM(JSEG)> <ALFAS(JSEG)> | <JSEG>
...
FOIL <AM(ISEG)> <ALFAS(ISEG)> | <JSEG>
PHIS <AS(1)> <AS(2)>
```

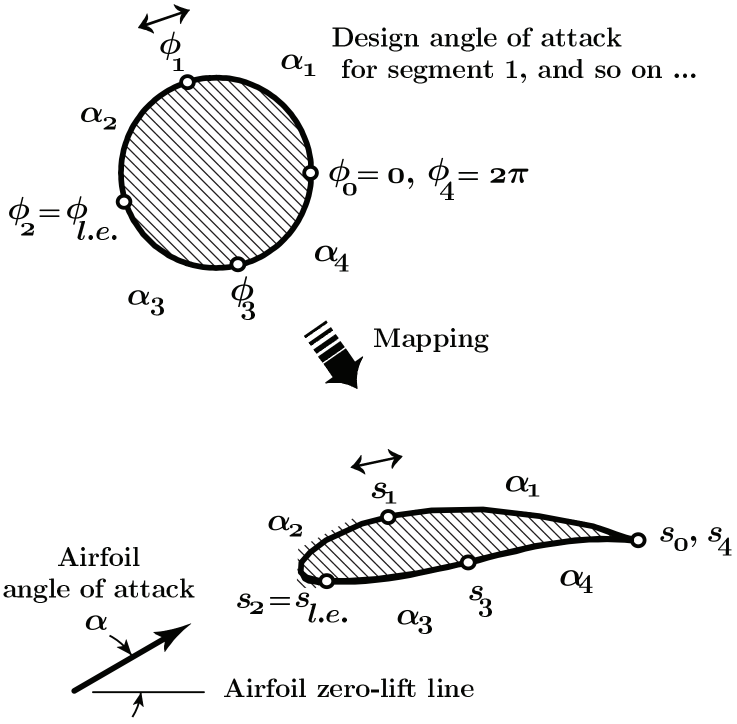

Figure 4: Parameterization of the circle as mapped to the airfoil.

```
REC <AKA(1)> <AKA(2)>
VLEV <JSEG> <VS(JSEG)>
ILE <ILE>
IDES
```

where

`IARGP` = number of airfoil coordinates
`AM(.)` = dimensionless arc limits, ϕ-values
`ALFAS(.)` = design angles of attack, α^*-values
`AS(1)` = trailing-edge upper surface closure arc limit, ϕ_s
`AS(2)` = trailing-edge lower surface closure arc limit, $\overline{\phi}_s$
`JSEG` = segment index for book-keeping
`AKA(1)` = upper surface recovery parameter, K
`AKA(2)` = lower surface recovery parameter, $\overline{K}$
`VS(JSEG)` = design velocity level v^* for specified segment `JSEG`
`ILE` = leading edge arc limit index for ϕ_{ILE}

The number of `FOIL` lines defines the number of segments. Thus, for a four-segment airfoil, four `FOIL` lines are used for the four ϕ–α^* pairs. The optional segment numbers following the ϕ–α^* pairs are used for book-keeping purposes. The arc limit for the last `FOIL` line always defines the number of sectors used around the airfoil. For instance, if the last `FOIL` line has $\phi_{ISEG} = 60$, then the circle is divided into 60 sectors. Throughout this User's

Guide, 60 sectors will be used. In this case, $\phi = 0$ corresponds to the trailing edge, 15 is near the middle of the upper surface, 30 is near the leading edge, 45 is near the middle of the lower surface, and finally 60 is the trailing edge again. The parameters `AS(.)`, `AKA(.)`, `VS(JSEG)` and `ILE` will be discussed in the first example.

Note that the trailing edge upper and lower surface closure arc limits ϕ_s and $\overline{\phi}_s$ fall inside the first and last segments defined by $\phi_W = \phi_1$ and $\overline{\phi}_W = \phi_{ISEG-1}$. Thus, the closure arc limits do not define new segments, but instead sectors of the first and last segments. These closure are limits are not shown in Fig. 4.

After all the input parameters are defined, the `IDES`-line is used to solve the inverse equations, which yield additional parameters that are used in the generation of the airfoil coordinates and velocity distribution. In a practical design, there are usually intervening lines between the input parameters and `IDES`-line.

✻ In the current method, the Eppler variable ϕ_W that defines the beginning of the main pressure recovery is automatically set equal to ϕ_1, and $\overline{\phi}_W$ is set to ϕ_{ISEG-1}. Thus, inputs are not required for ϕ_W and $\overline{\phi}_W$. In the Eppler method, ϕ_S and $\overline{\phi}_S$ can be set to any values; however, these trailing-edge closure arc limits are only useful in the trailing-edge closure region. For this reason, the current method requires that ϕ_S and $\overline{\phi}_S$ be located in the recovery regions, that is, $\phi_S < \phi_W = \phi_1$ and $\overline{\phi}_S > \overline{\phi}_W = \phi_{ISEG-1}$. (Again, the snow-flake symbol paragraphs like this one refer to material that relates the current method to the Eppler code.)

4.3 Coordinates and Velocity Distributions

To generate airfoil coordinates and the velocity distribution, the following lines are required.

```
FINISH 100
ALFASP 2
4
10
VELDIST 50 60
```

The numbers 100, 50 and 60 above cause the output (coordinates and velocity distributions) to be written to files `fort.100` and `fort.50/fort.60`, respectively. The file `fort.50` contains the velocity distribution $v(x/c)$ while `fort.60` contains $v(\phi)$. Depending on the system and compiler, the `fort*` file names may differ slightly. The `ALFASP`-line specifies that the velocity distribution be plotted for two angles of attack: 4 and 10 deg relative to the airfoil zero-lift line. The `VELDIST`-line is then used to compute the velocity distribution. If the `ALFASP`-line were omitted, then the user would be prompted to enter the desired angles of attack for which the velocity distribution is to be determined. For the Matlab GUI to *PROFOIL*, only `VELDIST 60` is required.

✍ EXAMPLE 5.1 *040*

In this example (file `man040.in`), an airfoil, named Airfoil 040, with four segments is designed. The coordinates are then generated and written to file `fort.100`, and the velocity distribution is written to files `fort.50/fort.60`. The `VELDIST`-line generates the velocity distributions for the selected angles of attack. The top part of the input file follows.

```
AIRFOIL MAN040
COORD 60
FOIL    17.50000    10.00000    1
FOIL    32.16061    10.00000    2
FOIL    45.50000     4.00000    3
FOIL    60.00000     4.00000    4
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           1.52728
ILE      2
IDES
FINISH 100
ALFASP 2
4
10
VELDIST 50 60
```

The resulting airfoil and velocity distributions for $\alpha = 4$ and 10 deg are shown together in Fig. 5. Several comments should be made regarding the airfoil and its input data.

- The design angle of attack for the second segment α_2^* is 10 deg. For this angle of attack the velocity distribution on the corresponding segment is constant. Likewise, for the third segment, the velocity is constant for 4 deg.

- The arc limits map to the airfoil as described in the Chapter 1. Since the airfoil is cambered, ϕ_2 is greater than 30.

- The trailing-edge closure arc limits, ϕ_S and $\overline{\phi}_S$ on the PHIS-line, define the region over which the airfoil closes at the trailing edge. Typically these are limits are placed close to the trailing edge, but they can be as large as the segment endpoints, ϕ_1 and ϕ_{ISEG-1}.

- The upper and lower surface recovery parameters, K and $\overline{K}$ given in the REC-line, control the initial slope of the recovery region—the larger the values, the steeper the slope.

- By prescribing the design velocity level v_2^* to be 1.527 for the second segment, that value for the velocity on segment 2 is obtained when the airfoil is operated at the corresponding design angle of attack. All velocity levels for the other segments are determined by the method. Thus, only one velocity level (one VLEV-line) can be prescribed.

  ✉The prescribed velocity level is the value at the beginning of the segment, that is, at the location of ϕ_2 in the example. This distinction can be important when later the velocity is not constant over the segment at the design angle of attack. (As a reminder "sealed-letter" paragraphs, such as this, contain material that should probably be skipped on first reading. Sometimes concepts in these paragraphs are not fully explain until later chapters.)

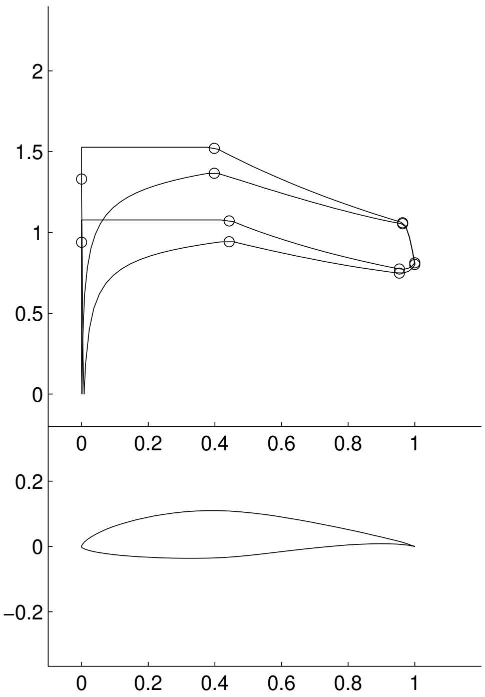

Figure 5: Airfoil 040 and velocity distributions for angles of attack of 4 and 10 deg relative to the zero-lift line.

- *ILE* gives the index of leading-edge arc limit, in this case, 2.

  ✉ `ILE` only becomes important when Newton iteration is used to adjust the upper and lower surface α^*-values in an opposed fashion, that is, adding an increment to the upper surface values and subtracting an equal increment from the lower surface values.

The effect of increasing ϕ_{ILE} ($\phi_2 = 31.160$) by 0.02 is shown in Fig. 5.3. A change of 0.05 in the velocity level v_1^* produces a equally dramatic effect. Clearly, it would be difficult to select appropriate values for ϕ_{ILE} and v_1^* that yield typical airfoils, which are not crossed or bulbous. As will be shown in the following chapter, however, it is possible to bypass the difficult task of selecting appropriate values for ϕ_{ILE} and the velocity level v_1^* for a given segment. Through Newton iteration, these values can be determined so that a desired trailing-edge thickness K_S and pitching-moment coefficient c_{m_0} are achieved.

Clearly, crossed airfoils are not excluded from the solution in any closed-form mathematical way. Even though the airfoil shown in Fig. 6 satisfies all the fundamental conditions, the airfoil is crossed. The problem stems from the high, trailing-edge velocity ratio. By empirical observation, the trailing-edge velocity ratio of finite-thickness, uncrossed airfoils is always less than one. Many inverse methods make use of this fact and allow for the adjustment of an inverse design parameter in order to match a specified trailing-edge velocity ratio. One shortcoming of this approach is that thicker airfoils generally have lower trailing-edge velocities than thinner ones, but the airfoil thickness is not known a priori, thereby making it difficult to preassign the proper trailing-edge velocity ratio. Also, the specification of the trailing-edge velocity ratio is not a viable option for the design of airfoils that have a finite trailing-edge angle for which the trailing-edge velocity is always zero.

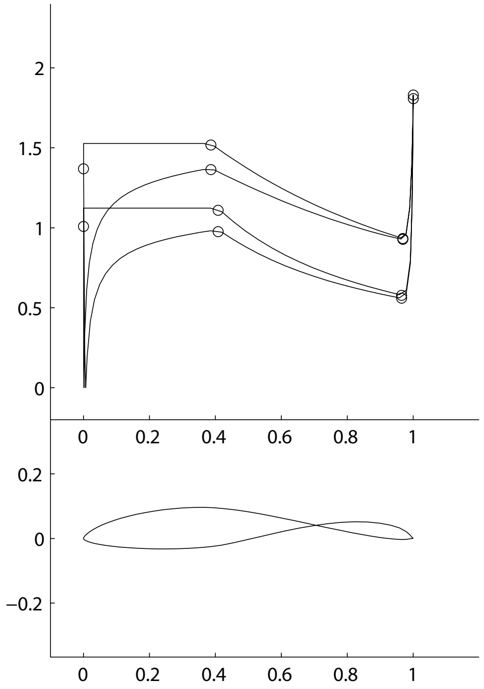

Figure 6: The effect of changing ϕ_2 by 0.02 for Airfoil 040 for angles of attack of 4 and 10 deg relative to the zero-lift line.

Besides the value of the trailing-edge velocity ratio, much can be deduced from the character of the velocity distribution in the vicinity of the trailing edge. Figure 7 shows the trailing-edge velocity distribution and the trailing-edge shape for three, symmetric, 8% thick, cusped airfoils at 5 deg angle of attack. Only the last 25% of chord is shown, and the vertical y/c-scale has been expanded to five times that of the x/c-scale. The trend is that the larger the drop in velocity (i.e., pressure recovery) at the trailing edge, the thicker the airfoil in the vicinity of the trailing edge (e.g, Fig. 7a). If there is no drop in the velocity, the trailing edge is very thin (e.g., Fig. 7b). If the velocity shows an increase, the airfoil is usually crossed (e.g., Fig. 7c). While these comments are specific to symmetric airfoils such as those shown in Fig. 7, the same trends are observed for cambered airfoils as long as the *net* velocity drop is considered. For example, if the velocity decreases on the upper surface by the same amount that it increases on the lower surface, there is a zero net velocity drop, and the airfoil will in such instance be thin at the trailing edge.

The trend just identified must be translated into an equation if crossed airfoils are to be avoided in the design process. The high, trailing-edge velocity ratio for the airfoil shown in Fig. 8 is produced by the large negative values of K_H ($K_H = -12.62$) and $\overline{K}_H$ ($\overline{K}_H = -16.64$) which control the closure recovery functions $w_S^{K_H}(\phi)$ and $\overline{w}_S^{\overline{K}_H}(\phi)$ [1]. If K_H and $\overline{K}_H$ are small positive quantities (for example, 0.2), then the trailing-edge velocity distribution will decrease slightly as shown in Fig. 7a. If they are both zero, there will be no decrease or increase in the velocity as shown in Fig. 7b. For negative values, the velocity will increase—Fig. 7c being a mild case and Fig. 8 extreme.

Practical experience with the method has shown that normal trailing-edge velocity distributions are determined not so much by the individual values of K_H and $\overline{K}_H$ but by their

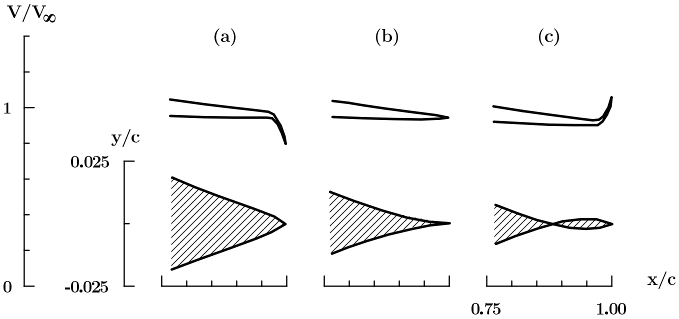

Figure 7: Impact of the trailing-edge velocity distribution on the shape of the trailing edge.

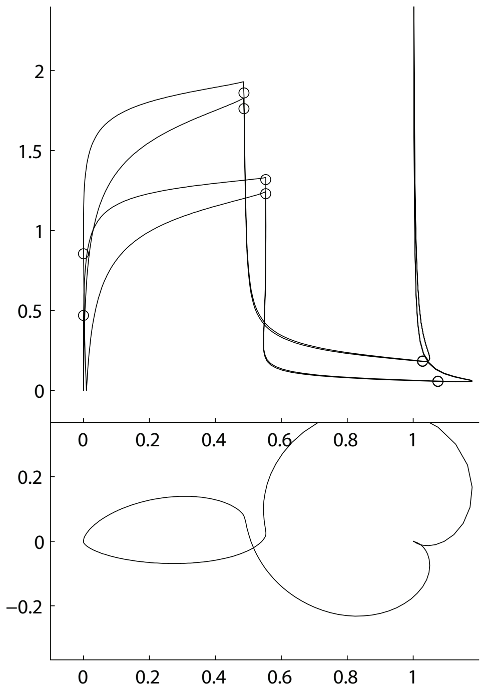

Figure 8: Example of a crossed airfoil with high trailing-edge velocity ratio for $\alpha = 15$ deg.

sum $K_H + \overline{K}_H$. If the sum is in the range 0.0 to 0.8, normal trailing-edge shapes are usually produced—the smaller the sum in this range, the thinner the airfoil in the vicinity of the trailing edge. Thus, as in the Eppler method [4, 3], $K_S = K_H + \overline{K}_H$ serves as a useful equation to control the trailing-edge velocity distribution. Thus, by specifying K_S as illustrated in the several examples in the following chapter, crossed airfoils can be avoided and a desired trailing-edge thickness can be achieved.

5 Four-Segment Airfoil

Several example airfoils will be presented in this chapter to illustrate the design method. All airfoils in this series will have four segments and be presented for angles of attack of 4, 6, 8, 10, 12 and 14 deg. Although four segments is the minimum number possible in the method, a surprisingly wide variety of airfoils can be designed for this simplest case. While reading this chapter, it is suggested that *PROFOIL* be used interactively.

In all of the examples presented in this chapter, the lines

```
...
# Converge to a tolerance of 0.00001
TOLSPEC 0.00001
# Iterate for a maximum of 15 iterations
ITERMAX 15
...
FINISH 100
ALFASP 7
2
4
6
8
10
12
14
VELDIST 50 60
DUMP
```

have been added to the input files. In this text, however, these last grouping of lines common to all files is not shown. The `NEWT*`-lines (e.g., the `NEWT1G0`-lines) allow for the specification of the (1) airfoil trailing-edge thickness parameter K_S, which as described controls the relative thickness of the trailing edge, and (2) airfoil pitching-moment coefficient c_{m_0}, which in turn determines the amount of aft camber or "aft loading." A detailed discussion of the `NEWT`-lines will be given in Chapter 7. At this point is it only necessary to know that they are used to assign ϕ_{ILE} and v_1^* for iteration in order to achieve specified values for K_S and c_{m_0}, respectively. For the lines shown, the first `NEWT`-line sets $K_S = 0.350$, and the second sets $c_{m_0} = -0.15$. The `TOLSPEC/ITERMAX`-lines are used to have convergence achieved automatically.

✉ For most new airfoil templates, it is best to begin Newton iteration without using the automatic convergence. In this case, manual iteration can be used to debug problems should they arise during the iteration. Once a template successfully converges, the `TOLSPEC/ITERMAX`-lines thereafter are quite useful. For the airfoils discussed in this chapter, the maximum number of iterations is set to 15 by the `TOLSPEC/ITERMAX`-lines. If the specified design parameters do not converge within the given number of iterations, the airfoil will not have the desired K_S and c_{m_0}.

✍ EXAMPLE 6.1 *050*

The first airfoil in the series to follow is given below. This airfoil will serve as point of departure for subsequent designs.

```
AIRFOIL MAN050
COORD 60
FOIL    15.50000    10.00000    1
FOIL    32.20000    10.00000    2
FOIL    45.50000     4.00000    3
FOIL    60.00000     4.00000    4
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           1.52728
ILE      2
TOLSPEC 0.00001
ITERMAX 15
# Specify the trailing-edge thickness parameter to be 0.350
NEWT1G0  100 0.350 1 2
IDES
# Specify the pitching moment to be -0.15
NEWT1G0  101 -0.15  4 1
IDES
```

After the airfoil is designed, the velocity distributions (`fort.50/fort.60`), airfoil coordinates `fort.100`) and converged input file (`fort.21`) are generated. The `fort.21` file is produced by the `DUMP`-line given in the input file `man050.in`. As seen in Fig. 9, the velocity is constant over segments 2 and 3 for angles of attack corresponding to the respective α^*-values for the segments.

As a reminder, the upper curves in the velocity distribution correspond to the upper (suction) surface, while the lower curves are for the lower (pressure) surface. As the angle of attack increases, the difference between the upper and lower surface velocity increases—the greater the difference between the curves, the higher the lift coefficient. Thus, the highest and lowest curves correspond to the upper and lower surfaces for the highest angle of attack, which is 14 deg.

✍ EXAMPLE 6.2 *055*

To move the recovery point s_2 (see Fig. 10) further back on the upper surface, the arc limit ϕ_1 is moved from 15.5 toward the trailing edge to 10.5.

```
AIRFOIL MAN055
COORD 60
FOIL    10.50000    10.00000    1
FOIL    32.20000    10.00000    2
FOIL    45.50000     4.00000    3
FOIL    60.00000     4.00000    4
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           1.52728
```

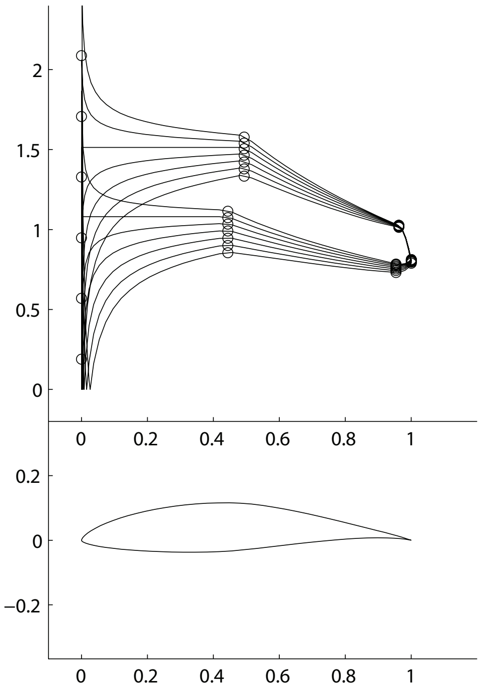

Figure 9: Airfoil 050 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

```
ILE      2
TOLSPEC 0.00001
ITERMAX 15
NEWT1G0  100 0.350 1 2
IDES
NEWT1G0  101 -0.15  4 1
IDES
```

✍ EXAMPLE 6.3 *060*

Starting with Airfoil 055, the pitching moment is increased from −0.15 to −0.25 to provide more aft loading/aft camber (see Fig. 11). Note that still the velocity is constant along the desired segments at the corresponding design angles of attack.

```
AIRFOIL MAN060
COORD 60
FOIL    10.50000    10.00000    1
FOIL    32.20000    10.00000    2
FOIL    45.50000     4.00000    3
FOIL    60.00000     4.00000    4
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           1.52728
```

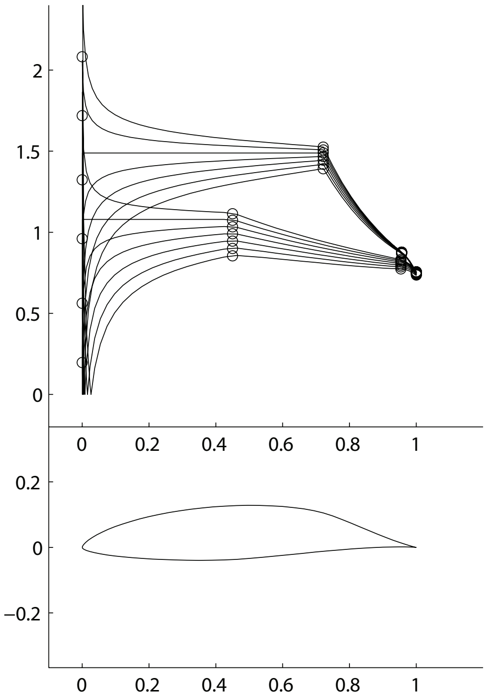

Figure 10: Airfoil 055 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

```
ILE      2
TOLSPEC 0.00001
ITERMAX 15
NEWT1G0  100 0.350 1 2
IDES
NEWT1G0  101 -0.25  4 1
IDES
```

✍ EXAMPLE 6.4 *065*

As mentioned in the definitions section, K_S is the trailing-edge thickness parameter. A small value of makes the trailing-edge thickness K_S essentially zero, while a relatively large value of 2 produces a thick trailing edge (see Fig. 12).

```
AIRFOIL MAN065
COORD 60
FOIL    10.50000    10.00000    1
FOIL    32.20000    10.00000    2
FOIL    45.50000     4.00000    3
FOIL    60.00000     4.00000    4
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           1.52728
ILE      2
```

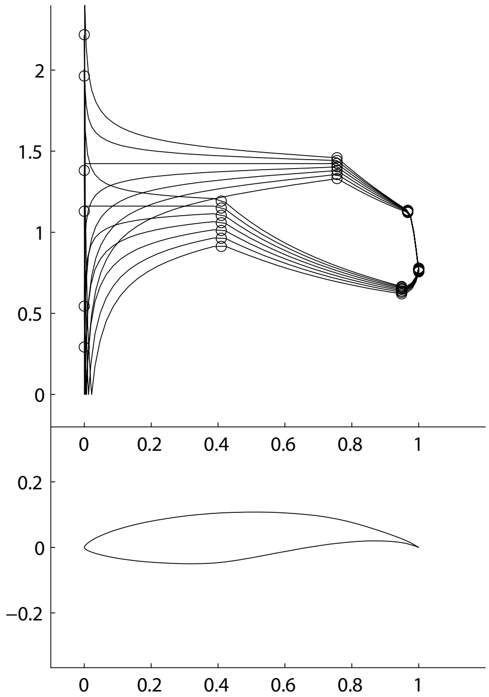

Figure 11: Airfoil 060 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

```
TOLSPEC 0.00001
ITERMAX 15
NEWT1G0  100 2.000 1 2
IDES
NEWT1G0  101 -0.25  4 1
IDES
```

✍ EXAMPLE 6.5 *070*

Starting from Airfoil 050, this next airfoil (see Fig. 13) illustrates the effects of changing the design angles of attack. For a design angle of attack of 10 deg for the second segment, the velocity distribution is constant over that segment at that angle of attack. When the angle of attack increases (see results for 12 and 14 deg in Fig. 9), the velocity distribution becomes adverse; that is, a high velocity is reached near the nose of the airfoil followed by a rapid deceleration. This feature of the velocity distribution eventually is so extreme that the flow will separate and cause stall.

In a direct design method, the approach usually taken to prevent the airfoil from stalling too early is to increase the airfoil camber or thickness at the nose. One advantage of the current inverse method is that the velocity distribution (in particular, the gradient of the velocity distribution) can be specified to avoid an "adverse pressure gradient" (a rapid deceleration in velocity) until the desired angle of attack is reached.

For the baseline airfoil (Airfoil 050), the design angle of attack for the second segment is 10 deg, which corresponds to a lift coefficient near 1. Stall can be expected to occur shortly after this angle of attack (or lift coefficient) is reached. In the current example, the angle

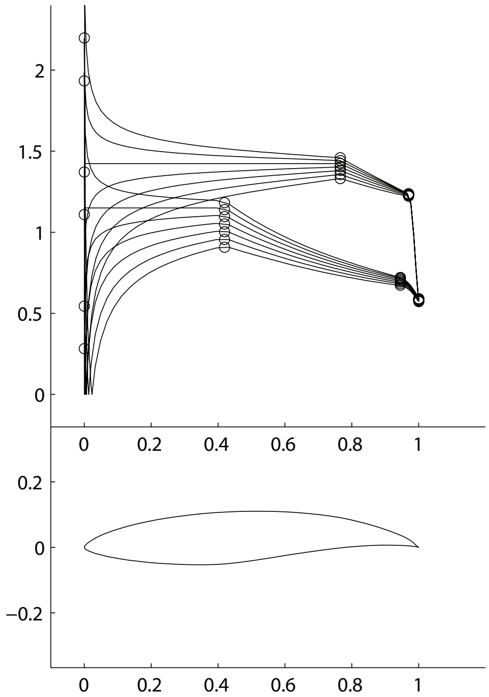

Figure 12: Airfoil 065 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

of attack on the upper-surface second segment α_2^* is changed to 12 deg. Now the velocity distributions are favorable up to 12 deg and stall will occur shortly thereafter. In this case, the lift coefficient is near 1.2.

```
AIRFOIL MAN070
COORD 60
FOIL    15.50000    10.00000    1
FOIL    32.20000    12.00000    2
FOIL    45.50000     4.00000    3
FOIL    60.00000     4.00000    4
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1             1.52728
ILE      2
TOLSPEC 0.00001
ITERMAX 15
NEWT1G0  100 0.350 1 2
IDES
NEWT1G0  101 -0.25  4 1
IDES
```

As seen the airfoil is somewhat thicker. This has happened because the lower surface was not changed; it must still operate with a constant velocity on the lower surface down

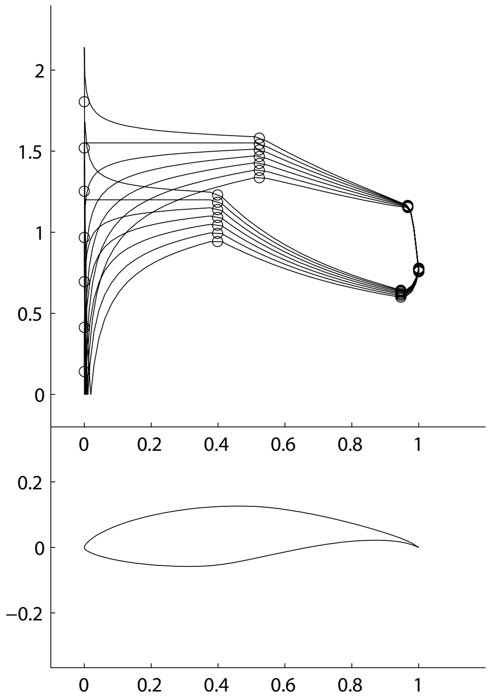

Figure 13: Airfoil 070 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

to an angle of attack of 4 deg. Below this angle of attack the velocity distributions become unfavorable, and the airfoil can be expected to separate off the nose of the lower surface.

✍ EXAMPLE 6.6 *075*

Airfoil 070 is the starting point for this last example (see Fig. 14). The design angle of attack of the lower surface is changed from 4 to 8 deg. Airfoil 070 behaved well down to 4 deg — the lower surface design angle of attack. The new airfoil only operates well down to 8 deg — the lower surface design angle of attack. The airfoil, therefore, has a narrower operating range (8 to 12 deg) as compared with Airfoil 070 (4 to 12 deg). Likewise, the airfoil is also thinner, which could have been anticipated because of the narrower operating range.

```
AIRFOIL MAN075
COORD 60
FOIL    15.50000    10.00000    1
FOIL    32.20000    12.00000    2
FOIL    45.50000     8.00000    3
FOIL    60.00000     4.00000    4
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           1.52728
ILE      2
TOLSPEC 0.00001
ITERMAX 15
```

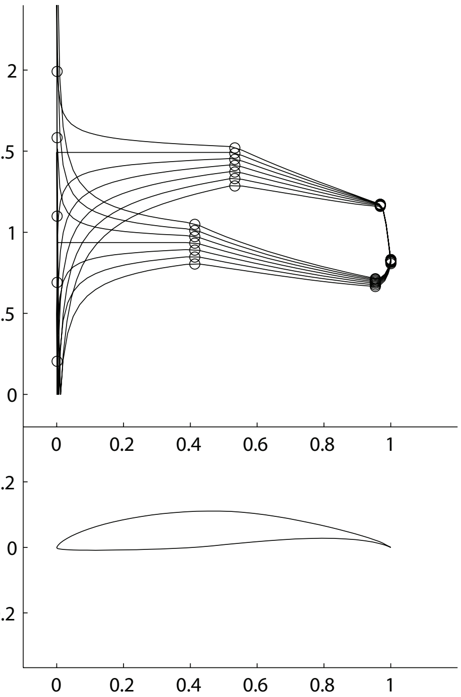

Figure 14: Airfoil 075 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

```
NEWT1G0  100 0.350 1 2
IDES
NEWT1G0  101 -0.25  4 1
IDES
```

✻ For those with some experience with the Eppler code, the pitching moment coefficient c_{m_0} takes the place of the ω-values. The μ-values are now determined by the method and cannot be explicitly prescribed unless further iteration is performed; that is, in addition to prescribing c_{m_0} and K_S, the μ-values can also be specified, provided that additional design variables are allowed to be free for iteration.

6 Airfoils with More Than Four Segments

In the previous chapter, the velocity distribution on the upper and lower surfaces was limited since only four segments were used. In this chapter, more segments are used to refine the velocity distributions and more iteration options are introduced. In the first series (five airfoils), iteration on the α^*-values is illustrated as a means of controlling the airfoil thickness and camber. Also, iteration on the arc limits is used to control the segment juncture chordwise position. In the second series (three airfoils), additional segments are used to illustrate a technique for smoothing the velocity distributions. In the last example, an alternative (more powerful) approach to prescribing the velocity distribution is presented.

✍ EXAMPLE 7.1 *080*

In this example shown in Fig. 15, four segments are used on the upper surface and two on the lower.

```
AIRFOIL MAN080
COORD 60
FOIL    15.50000    10.00000    1
FOIL    19.50000     6.00000    2
FOIL    24.50000     8.00000    3
FOIL    32.20000    10.00000    4
FOIL    45.50000     4.00000    5
FOIL    60.00000     4.00000    6
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           1.52728
ILE      4
TOLSPEC 0.00001
ITERMAX 15
NEWT1G0  100 0.350 1 4
IDES
NEWT1G0  101 -0.15  4 1
IDES
```

In this case, ILE is now 4. As before, iteration is used to achieve desired values for K_S and c_{m_0}. As shown in Fig. 15, the velocity distribution is constant over segments 2, 3 and 4 at 6, 8 and 10 deg—the respective α^*-values.

✉ The locations of the 60 coordinates coincide with the 60 sectors around the circle. The junctions between segments (ϕ-values), however, fall midway between coordinate pairs. Thus, the velocity distribution is somewhat rounded about these points. If the ϕ-values on the upper surface were changed to integer values, they would coincide with coordinate points and the velocity distributions would have sharp kinks at the junctions between segments. Since most often smooth velocity distributions are desired, the segment junctions in this User's Guide will be located to fall between coordinate pairs.

✍ EXAMPLE 7.2 *085*

As discussed in the previous chapter (see Airfoil 070), increasing the α^*-values on the upper

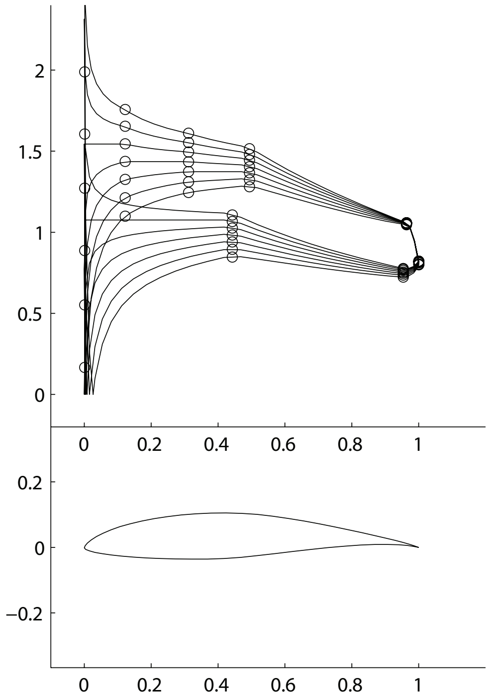

Figure 15: Airfoil 080 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

surface will increase the thickness. Likewise, decreasing the values on the lower surface will also increase the thickness. The desired thickness can be achieved automatically by adding a new NEWT1-line as follows

```
AIRFOIL MAN085
COORD 60
FOIL    15.50000    10.00000    1
FOIL    19.50000     6.00000    2
FOIL    25.50000     8.00000    3
FOIL    32.20000    10.00000    4
FOIL    45.50000     4.00000    5
FOIL    60.00000     4.00000    6
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           1.52728
ILE      4
TOLSPEC 0.00001
ITERMAX 15
NEWT1G0  100 0.350 1 4
IDES
NEWT1G0  101 -0.15  4 1
IDES
# Specify the max thickness to be 15%
```

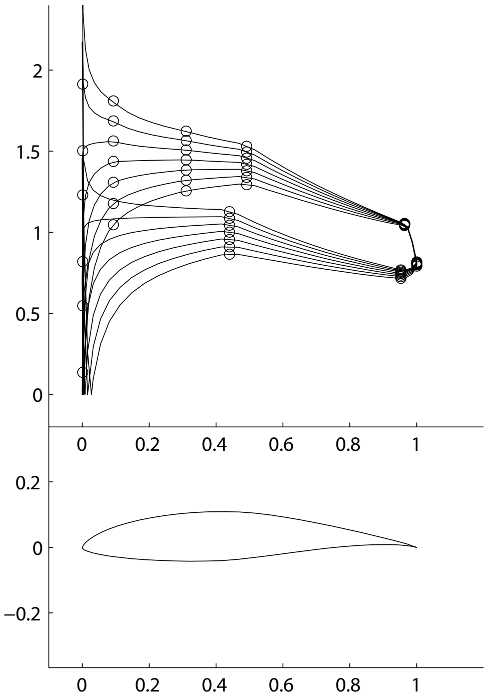

Figure 16: Airfoil 085 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

```
NEWT1G0  102 .1500 6 300 .5
IDES
```

In this case, the airfoil thickness is prescribed to be 15%, instead of 13.96% for the baseline Airfoil 080. The iteration mode (via the `NEWT1G0 102`-line) used here increases the upper-surface α^*-values by an increment (0.37 deg) and decreases the lower surface ones by the same amount (see Fig. 16).

Note that since the design angles of attack were changed from those shown above, the velocity is not constant, for example, on the second segment for an angle of attack of 6 deg. Instead for that segment, the velocity is constant at an angle of attack of 6.37 deg due to the added increment needed to achieve the desired thickness of 15%. Again, these converged values are included in the output file `fort.21` generated by the `DUMP`-line.

✉ For most practical airfoil designs, either the upper or lower limit of the low-drag range is set by the airfoil design requirements. This fixed corner of the polar can be maintained by fixing the α^*-distribution for the corresponding airfoil surface. In such a case, a thickness constraint can be achieved by iterating on the α^*-values on the other "free" surface.

✍ EXAMPLE 7.3 *090*

The α^*-values when both increased or decreased can be respectively used to increase or decrease the camber. The previous Airfoil 085, has a camber of 3.76%, which can be changed to 5.5% by adding another `NEWT1`-line as follows.

```
AIRFOIL MAN090
```

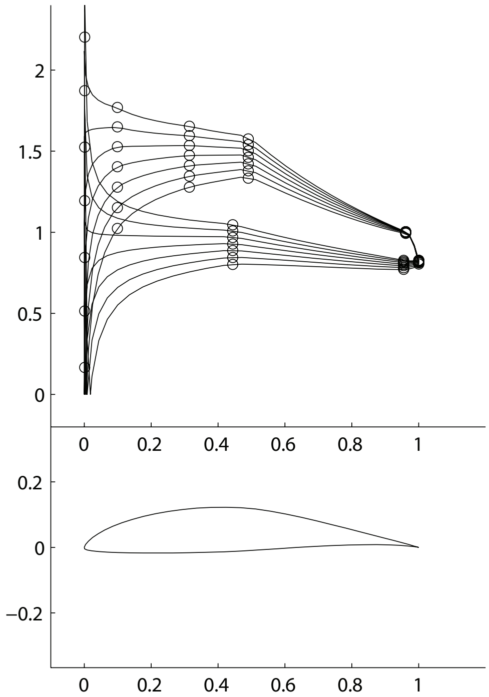

Figure 17: Airfoil 090 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

```
COORD 60
FOIL    15.50000    10.00000    1
FOIL    19.50000     6.00000    2
FOIL    25.50000     8.00000    3
FOIL    32.20000    10.00000    4
FOIL    45.50000     4.00000    5
FOIL    60.00000     4.00000    6
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           1.52728
ILE      4
TOLSPEC 0.00001
ITERMAX 15
NEWT1G0  100 0.350 1 4
IDES
NEWT1G0  101 -0.15  4 1
IDES
# Specify the camber to be 5.5%
NEWT1G0  105 .0550 6 400 .5
IDES
```

After the NEWT1-lines for K_S and c_{m_0}, the next NEWT1-line adds in equal increment to both the upper- and lower-surface α^*-values to achieve the desired camber of 5.5% (see Fig. 17).

✍ EXAMPLE 7.4 *095*

It is possible to achieve both a desired thickness and camber by iterating on the upper- and lower-surface α^*-values independently.

```
AIRFOIL MAN095
COORD 60
FOIL    15.50000    10.00000    1
FOIL    19.50000     6.00000    2
FOIL    25.50000     8.00000    3
FOIL    32.20000    10.00000    4
FOIL    45.50000     4.00000    5
FOIL    60.00000     4.00000    6
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           1.52728
ILE      4
TOLSPEC 0.00001
ITERMAX 15
NEWT1G0  100 0.350 1 4
IDES
NEWT1G0  101 -0.15  4 1
IDES
NEWT1G0  102 .1200 6 200 .5
IDES
NEWT1G0  105 .0550 6 400 .5
IDES
```

Following the NEWT1-lines for K_S and c_{m_0}, the NEWT1-line achieves a thickness of 12% by iterating on the lower-surface α^*-values. After this is achieved, the next NEWT1-line iterates on the upper-surface α^*-values to achieve the desired camber of 5.5% (see Fig. 18).

✉ In the iteration process, desired parameters, such as the thickness and camber, are set by iterating on an appropriate set of basic design parameters. It should be understood, however, that the resulting system of equations (that grows larger with each stage in the iteration) are coupled—each parameter affects another. Thus, although it may be stated that one basic input parameter, e.g., ϕ_{ILE}, is used for iteration to achieve a desired parameter, e.g., K_S, all of the input parameters used for iteration have to some degree an effect on each of the desired parameters.

✉ As each stage in the iteration process converges, subsequent stages will "unconverge" the already converged stages. Thus, it may seem that there is no advantage to approaching the iteration process in stages. However, the staged approach ensures that as each new prescription is given, the current solution is feasible, thereby, making it readily apparent if one of the subsequent prescriptions is not physically realizable.

✍ EXAMPLE 7.5 *105*

In this last example of the series, the lower-surface arc limit ϕ_5 is used for iteration so that the segment juncture occurs at 70% chord. The new NEWT1-line is placed after the pitching moment NEWT-line.

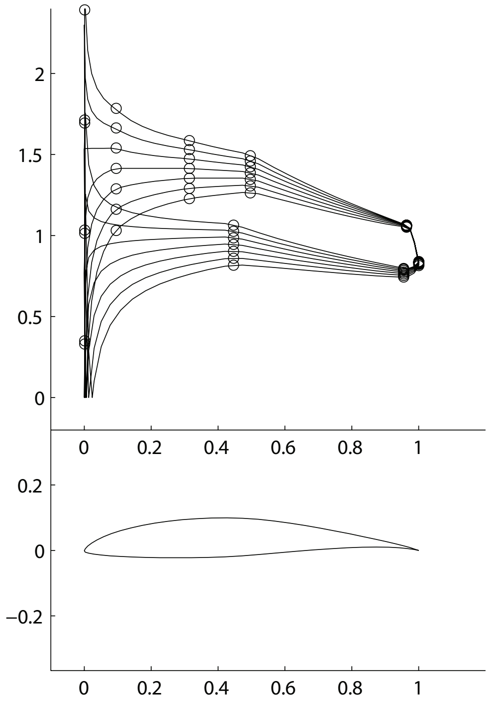

Figure 18: Airfoil 095 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

```
AIRFOIL MAN105
COORD 60
FOIL    15.50000    10.00000    1
FOIL    19.50000     6.00000    2
FOIL    25.50000     8.00000    3
FOIL    32.20000    10.00000    4
FOIL    45.50000     4.00000    5
FOIL    60.00000     4.00000    6
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           1.52728
ILE      4
TOLSPEC 0.00001
ITERMAX 15
NEWT1G0  100 0.350 1 4
IDES
NEWT1G0  101 -0.15  4 1
IDES
NEWT1G0  102 .1200 6 200 .5
NEWT1G0  105 .0550 6 400 .5
# Specify the arc limit 5 to correspond to 70% of chord
NEWT1S0  400  5  0  .700 1    5  5
IDES
```

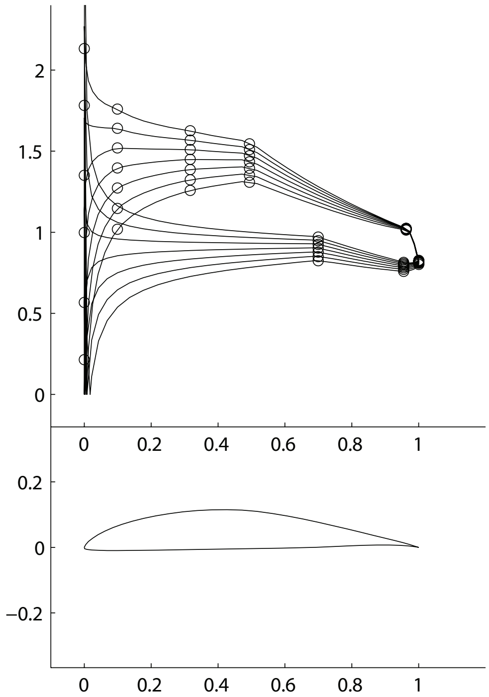

Figure 19: Airfoil 105 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

Note that as this example illustrates, the `IDES`-line does not have to be placed after each `NEWT`-line. Thus, in this example, t/c, camber and x/c_5 are obtained simultaneously, rather than one after another.

✍ EXAMPLE 7.6 *110*

The relatively sharp corners on the velocity distribution shown in Airfoil 080 can be smoothed by adding more segments on upper surface with α^*-values that change gradually. Typically, smooth velocity distributions can be achieved by using a 12 deg spacing about the circle, which for 60 sectors around the circle leads to a segment arc length of 2 as follows.

```
AIRFOIL MAN110
COORD 60
FOIL    15.50000    10.00000    1
FOIL    17.50000     3.00000    2
FOIL    19.50000     4.00000    3
FOIL    21.50000     5.00000    4
FOIL    23.50000     6.00000    5
FOIL    25.50000     7.00000    6
FOIL    27.50000     8.00000    7
FOIL    29.50000     9.00000    8
FOIL    32.20000    10.00000    9
FOIL    45.50000     4.00000   10
FOIL    60.00000     4.00000   11
```

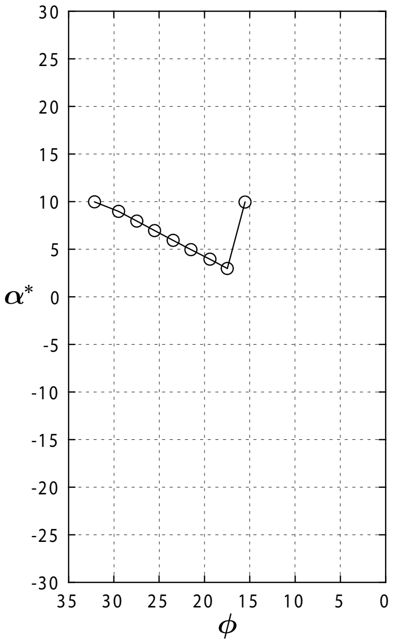

Figure 20: Upper surface α^*-distribution for Airfoil 110.

```
PHIS     3.50000    56.50000
REC        .02000      .02000
VLEV     1             1.52728
ILE      9
TOLSPEC 0.00001
ITERMAX 15
NEWT1G0  100 0.350 1 9
IDES
NEWT1G0  101 -0.15  4 1
IDES
```

Figure 20 shows the upper-surface distribution of α^*-values versus the arc limit ϕ. The ϕ-scale is plotted so that it coincides approximately with the chordwise position. Note that when the upper-surface ϕ–α^* curve is plotted in this way its shape is reflected in the velocity distributions, which are shown in Fig. 21.

✍ EXAMPLE 7.7 *115*

To further illustrate the effects of the ϕ–α^* curve on the velocity distribution, a second upper-surface ϕ–α^* curve is shown in Fig. 22. Again the general character is reflected in the velocity distributions in Fig. 23.

```
AIRFOIL MAN115
COORD 60
FOIL    15.50000   -12.35017    1
FOIL    17.50000    -9.45999    2
FOIL    19.50000    -6.56982    3
```

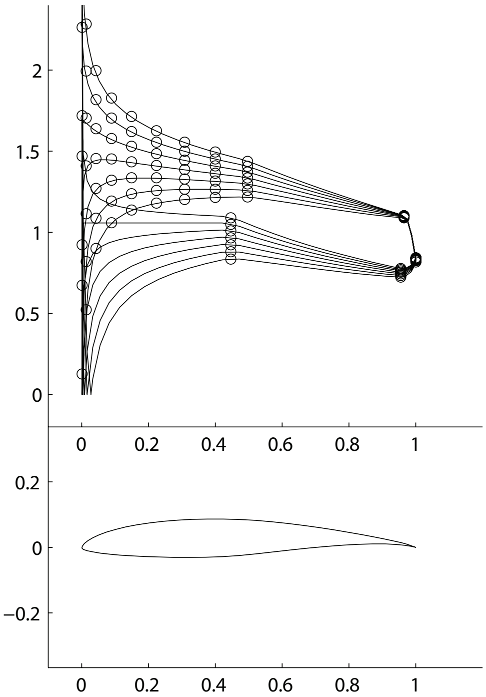

Figure 21: Airfoil 110 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

```
FOIL    21.50000    -3.67965    4
FOIL    23.50000    -0.78947    5
FOIL    25.50000     2.10070    6
FOIL    27.50000     4.99087    7
FOIL    29.50000     7.88105    8
FOIL    32.20000    11.78278    9
FOIL    45.50000     4.00000   10
FOIL    60.00000     4.00000   11
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           .8
ILE      9
TOLSPEC 0.00001
ITERMAX 15
NEWT1G0  100 0.350 1 9
IDES
NEWT1G0  101 -0.15  4 1
IDES
```

Clearly, this airfoil is not typical since there is a great deal of aft loading on an airfoil with very little camber (2.09%). One approach to add camber is through the NEWT1-line as previously discussed. Another approach is simply to add an increment to all the α^*-values. This can be done manually, or though the BUMPALPHAS-line given by

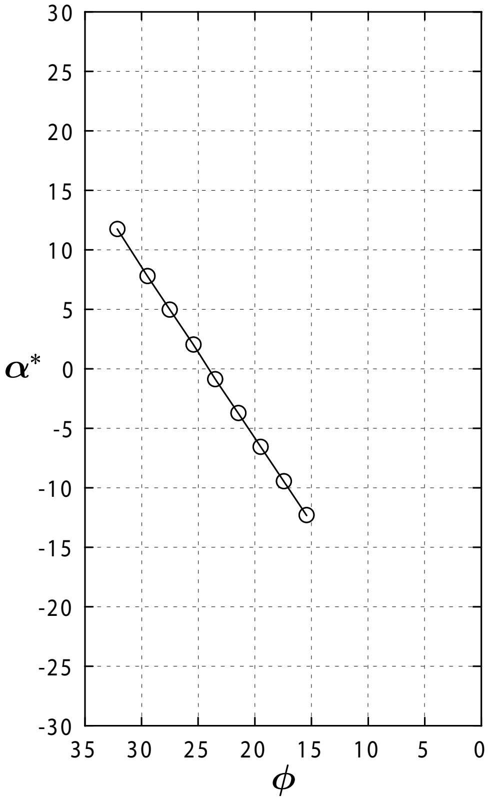

Figure 22: Upper surface α^*-distribution for Airfoil 115.

```
BUMPALFAS <ICASE> <DELTA>
```

where

`ICASE` = 100 → add an increment to all upper α^*-values
200 → add an increment to all lower α^*-values
300 → add an increment to all upper α^*-values
and subtract an increment from all upper α^*-values
400 → add an increment to all upper α^*-values
and add an increment to all upper α^*-values

`DELTA` = the increment in α^*

✉ For some airfoils, especially when the ϕ–α^* curve distribution is pushed close to the airfoil trailing edge, the shape of the resulting velocity distribution deviates from the ϕ–α^* curve. Nevertheless, tailoring the ϕ–α^* curve distribution is still an effective means of obtaining a desired airfoil and velocity distribution.

✍ EXAMPLE 7.8 *120*

For the last example in the series, the lower-surface velocity distribution in prescribed in a manner similar to that for the upper surface. The ϕ–α^* curve shown in Fig. 24, however, is oriented differently to properly reflect the character of the velocity distribution (see Fig. 25.

```
AIRFOIL MAN120
```

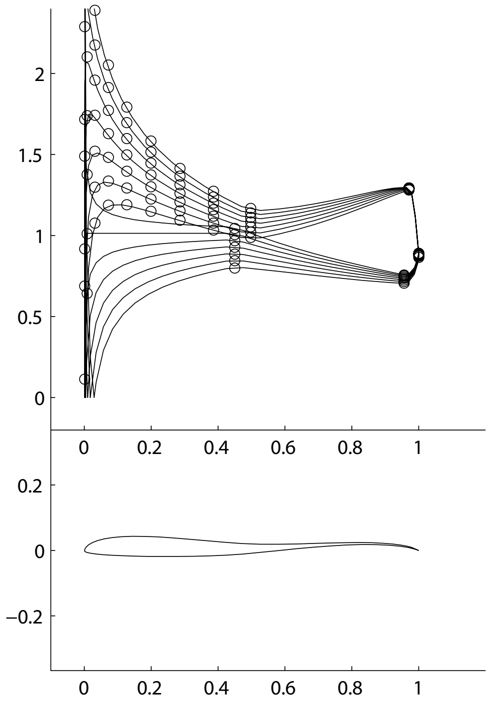

Figure 23: Airfoil 115 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

```
COORD 60
FOIL    15.50000    10.00000    1
FOIL    17.50000     3.00000    2
FOIL    19.50000     4.00000    3
FOIL    21.50000     5.00000    4
FOIL    23.50000     6.00000    5
FOIL    25.50000     7.00000    6
FOIL    27.50000     8.00000    7
FOIL    29.50000     9.00000    8
FOIL    32.20000    10.00000    9
FOIL    34.50000     2.00000   10
FOIL    36.50000     3.00000   11
FOIL    38.50000     4.00000   12
FOIL    40.50000     5.00000   13
FOIL    42.50000     6.00000   14
FOIL    44.50000     7.00000   15
FOIL    46.50000     8.00000   16
FOIL    60.00000     9.00000   17
PHIS     3.50000    56.50000
REC       .02000      .02000
VLEV     1           1.52728
ILE      9
TOLSPEC 0.00001
```

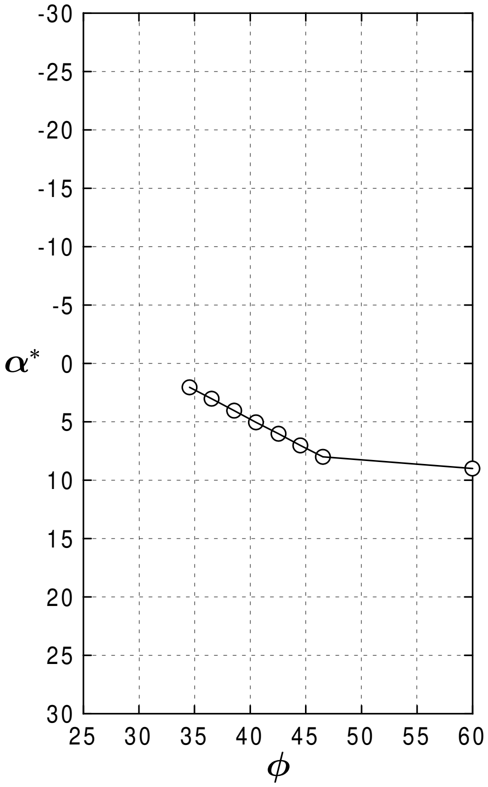

Figure 24: Lower-surface α^*-distribution for Airfoil 120.

```
ITERMAX 15
NEWT1G0  100 0.350 1 9
IDES
NEWT1G0  101 -0.15  4 1
IDES
```

✍ EXAMPLE 7.9 *125*

This next airfoil presents an alternative means of prescribing the velocity for a segment. In all the preceding examples, the velocity distribution for each segment was constant when the angle of attack corresponded to the segment α^*. In some cases, there is an advantage to prescribing a non-constant velocity distribution for a segment. The example below is quite simple, but it will illustrate some of the various options, which will be summarized in Chapter 8.

(Note for code in github repository as of Nov-2022: The code does not support the `DELV`-16 line below. The spline code in *PROFOIL* was removed and could be replaced with libraries from: LINPACK.)

```
AIRFOIL MAN125
COORD 60
FOIL    18.33000     5.00000    1
FOIL    31.74561    15.00000    2
```

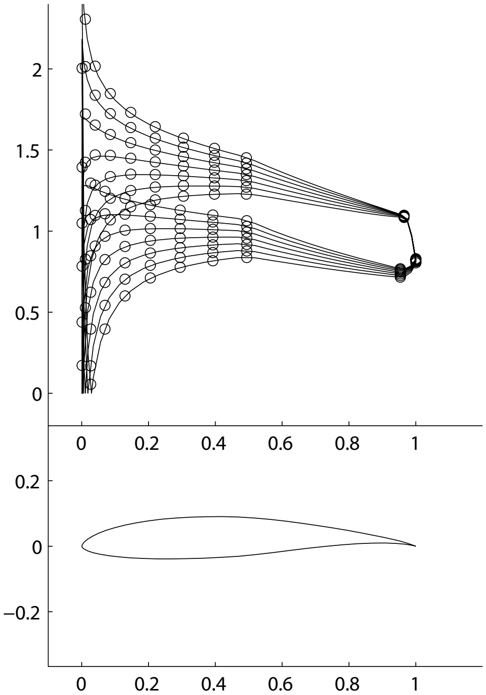

Figure 25: Airfoil 120 and velocity distributions for angles of attack of 2, 4, 6, 8, 10, 12, and 14 deg relative to the zero-lift line.

```
FOIL    46.66600      .00000    3
FOIL    60.00000    10.00000    4
DELV 16    2      4
   .300     .200000
   .700     .600000
   .800     .450000
  1.000     .350000
DELV  1    3      .20000
PHIS     3.33000    56.66600
REC      2.00000     2.00000
VLEV     1           1.50711
ILE      2
TOLSPEC 0.00001
ITERMAX 15
NEWT1G0 100  .3  1 2
IDES
NEWT1G0 101 -.05  4 1
IDES
```

The resulting airfoil is shown in Figs. 26 and 27. As required by the method, segments 1 and 4 are dominated by the pitching moment and trailing edge thickness prescriptions as well as the velocity recovery functions. For segment 2, the velocity distribution is prescribed to be the $\tilde{v}_2$ function shown in Fig. 27. The special notation $(\tilde{\ })$ is used to denote the function

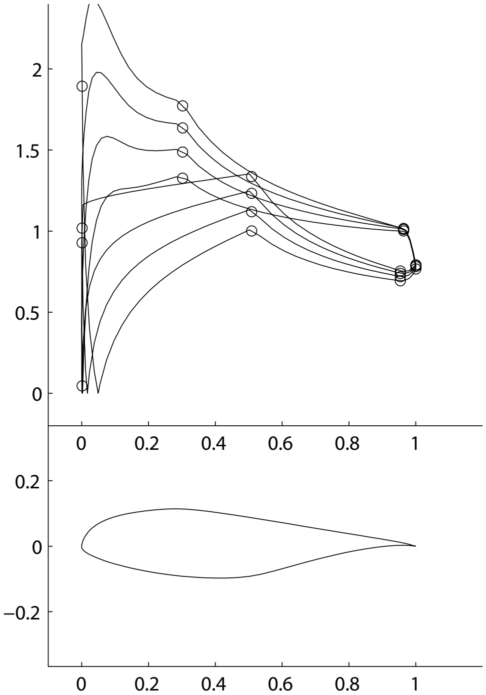

Figure 26: Fig. 7.12 Airfoil 125 and velocity distributions for angles of attack of 0, 5, 10, and 15 deg relative to the zero-lift line.

relative to the beginning of the segment. In the input file, the velocity for segment 2 is given by the first DELV-line, which prescribes the "relative velocity" from the beginning of the segment to the end of the segment, that is, from ϕ_1 to ϕ_2. This relative velocity distribution is defined by the coordinate system depicted in Fig. 28, albeit the segment in the figure is for the lower surface. In the input file, the velocity distribution is prescribed for the entire segment, which is normalized to unit length. The first point, which is by definition (0,0), is not included in the input file.

The next DELV-line prescribes the relative velocity for segment 3 on the lower surface. In this case, the relative velocity is linear in terms of ϕ, and therefore, only the endpoint is needed.

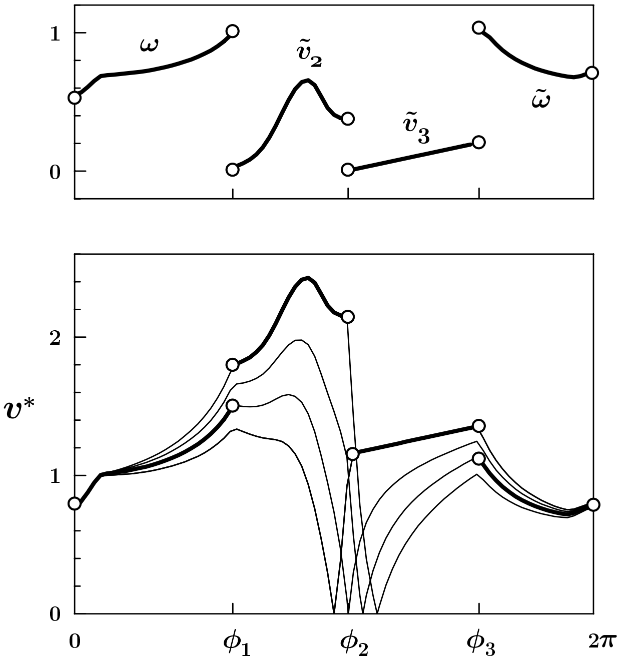

Figure 27: Relative velocity distributions for the recovery regions (segments 1 and 4) and main segments 2 and 3 and overall velocity distribution $v(\phi)$ for Airfoil 125.

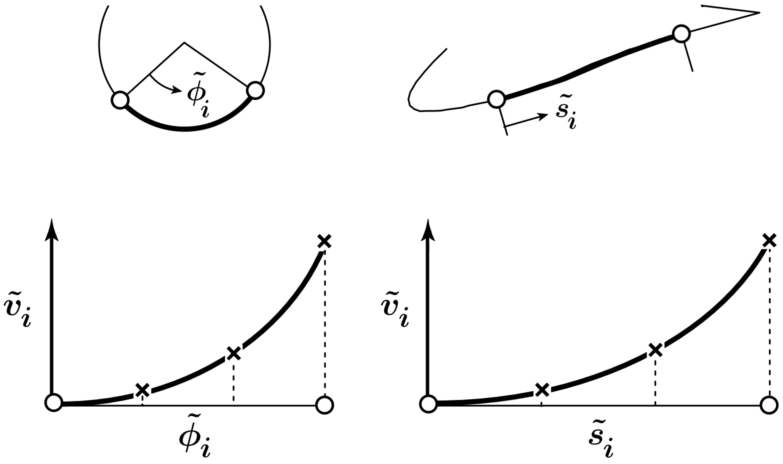

Figure 28: Definition of the splined relative velocity distribution in terms of the circle coordinate $\tilde{\phi}_i$ and $\tilde{s}_i$.

7 Newton Iteration Lines for Desired Single Parameters

In Chapters 5 and 6, various parameters (e.g., airfoil thickness, pitching moment, etc.) were achieved through Newton iteration as determined by the NEWT1*-lines. In this chapter, the various parameters that can be specified will be discussed as well as the input parameters available for iteration.

As an overview, IFTP1(.) is used to identify which desired airfoil parameter is being specified. The actual value of the desired airfoil parameter is set by FNEWT1(.). In order to achieve this single desired airfoil parameter one of the basic input design parameters must be selected for iteration. ITP1(.) and ITP2(.) are used to identify which design parameter is being selected for iteration. If only one NEWT1*-line is used this sets up a one-dimensional Newton iteration. If N NEWT1*-lines are used this sets up a N-dimensional Newton iteration.

During the Newton iteration, it is possible that the step size could be too large and lead to divergence. To prevent this from taking place, the step size can be limited. For example, if an arc limit ϕ is used for iteration and if it is known that taking a step in ϕ of greater than 10 deg leads to divergence, then the step size can be limited to be no greater than 10 deg. The limits on the design variables used for iteration is specified through CLAMP1(.), which is optional.

Several general NEWT1*-line types are implemented. The most basic involve such things as the trailing-edge thickness parameter K_S, the airfoil maximum thickness t/c_{max}, zero-lift pitching moment c_{m_0} and more. Others allow for the specification of desired airfoil parameters that depend on, say, the angle of attack or Reynolds number. If this is the case, the angle of attack or Reynolds number is input by COND*(.).

The first NEWT1-line, which was used frequently in the examples in the previous chapters, is given by

```
NEWT1G0 <IFTP1(.)>  <FNEWT1(.)> <ITP1(.)> <ITP2(.)> | <CLAMP1(.)>
```

where

IFTP1(.) = Newton function type
100 → K_S
101 → c_{m_0} at zero-lift
102 → t/c_{max}
103 → zero-lift angle of attack α_0
104 → x/c location of maximum thickness
105 → maximum camber
106 → leading-edge radius
107 → t/c_{max} via Eppler method
184 → ω
185 → $\overline{\omega}$
190 → μ
191 → $\overline{\mu}$
192 → K_H

193 → $\overline{K}_H$
194 → ω, faster convergence than 184
195 → $\overline{\omega}$, faster convergence than 185
196 → ω'
197 → $\overline{\omega}'$
198 → ω_T (incompatible with IFTP1 = 200)
199 → $\overline{\omega}_T$ (incompatible with IFTP1 = 200)

FNEWT1(.) = desired value for specified parameter
ITP1(.) = used to identify input variable for iteration (see Table 1)
ITP2(.) = used to identify input variable for iteration (see Table 1)
CLAMP1(.) = positive step limit used during iteration (optional)

✻ Iteration mode types 184, 185, 190, 191, 194, 195, 196, 197, 198 and 199 are available to allow for the specification of parameters that are familiar to those with experience using the Eppler code. It should be noted that although, for example, ω and $\overline{\omega}$ may be specified on the Eppler-code TRA2 input line, these values in the Eppler method may not be achieved after iteration for the desired K_S. Therefore, to match an airfoil designed by the Eppler code, it is necessary to specify in *PROFOIL* those converged values from the Eppler code. The appropriate data can be found in the Eppler output file.

The remaining NEWT1*-lines are given as follows:

```
NEWT1G1 <IFTP1(.)> <COND1(.)> <FNEWT1(.)> <ITP1(.)> <ITP2(.)> -
       | <CLAMP1(.)>
```

where

IFTP1(.) = Newton function type
200 → trailing-edge velocity v_{TE} for given angle of attack
201 → trailing-edge recovery velocity differential, $v(\phi_S) - v(\overline{\phi}_S)$, for given angle of attack
202 → trailing-edge recovery velocity $v(\phi_S)$
203 → trailing-edge recovery velocity $v(\overline{\phi}_S)$
205 → t/c for given x/c location

COND1(.) = angle of attack for specified value
FNEWT1(.) = desired value for specified parameter
ITP1(.) = input variable for iteration (see Table 1)
ITP2(.) = input variable for iteration (see Table 1)
CLAMP1(.) = positive step limit used during iteration (optional)

Table 1: Input Variable Specification for `NEWT1*`-Lines

`ITP1(.)`	`ITP2(.)`
1	$i \rightarrow \phi_i + \Delta\phi$ 100 → Upper-surface ϕ-values + $\Delta\phi$, except ϕ_{ILE} 200 → Lower-surface ϕ-values + $\Delta\phi$, except ϕ_{ILE} 300 → Upper-surface ϕ-values + $\Delta\phi$ and lower-surface ϕ-values - $\Delta\phi$, except ϕ_{ILE} 400 → Upper-surface and lower-surface ϕ-values + $\Delta\phi$, except ϕ_{ILE}
2	$100 \rightarrow \phi_W (= \phi_1) + \Delta\phi$ $200 \rightarrow \overline{\phi}_W (= \phi_{ISEG-1}) + \Delta\phi$ $300 \rightarrow \phi_W + \Delta\phi$ and $\overline{\phi}_W - \Delta\phi$ $400 \rightarrow \phi_W + \Delta\phi$ and $\overline{\phi}_W + \Delta\phi$
3	$100 \rightarrow \phi_S (= \phi_1) + \Delta\phi$ $200 \rightarrow \overline{\phi}_S (= \phi_{ISEG-1}) + \Delta\phi$ $300 \rightarrow \phi_S + \Delta\phi$ and $\overline{\phi}_S - \Delta\phi$ $400 \rightarrow \phi_S + \Delta\phi$ and $\overline{\phi}_S + \Delta\phi$
4	$i \rightarrow v_i$ (`ITP1(.)` of 4 is allowed only once)
5	$i \rightarrow \tilde{v}_i + \Delta\tilde{v}$
6	$i \rightarrow \alpha^*_i + \Delta\alpha^*$ 100 → Upper-surface α^*_i-values + $\Delta\alpha^*$ 100 → Lower-surface α^*_i-values + $\Delta\alpha^*$ 300 → Upper-surface α^*_i-values + $\Delta\alpha^*$ and lower-surface α^*_i-values - $\Delta\alpha^*$ 400 → Upper-surface α^*_i-values + $\Delta\alpha^*$ and lower-surface α^*_i-values + $\Delta\alpha^*$
7	$100 \rightarrow K + \Delta K$ $200 \rightarrow \overline{K} + \Delta\overline{K}$ $300 \rightarrow K + \Delta K$ and $\overline{K} - \Delta\overline{K}$ $400 \rightarrow K + \Delta K$ and $\overline{K} + \Delta\overline{K}$

The following three `NEWT1*`-lines are used to prescribed a desired parameter at the beginning or end of a segment. For reference, EOS will be used to refer to the end of a segment with respect to its position about the circle. Likewise, BOS will be used to refer to the beginning of the segment. For example, for a segment that extends from 30 deg to 80 deg has the EOS at 30 deg and the BOS at 80 deg.

```
NEWT1SO <IFTP1(.)> <JSEGIX1(.)> <LLBOS(.)> <FNEWT1(.)> <ITP1(.)> -
        <ITP2(.)> | <CLAMP1(.)>
```

where

`IFTP1(.)` = Newton function type
400 → x/c
401 → s/c
402 → y/c
`JSEGIX1(.)` = segment for which parameter is specified
`LLBOS(.)` = 1 → indicates value for BOS
0 → indicates value for EOS

`FNEWT1(.)` = desired value for specified parameter
`ITP1(.)` = input variable for iteration (see Table 1)
`ITP2(.)` = input variable for iteration (see Table 1)
`CLAMP1(.)` = positive step limit used during iteration (optional)

```
NEWT1S1 <IFTP1(.)> <JSEGIX1(.)> <LLBOS(.)> <COND1(.)> <FNEWT1(.)> -
        <ITP1(.)> <ITP2(.)> | <CLAMP1(.)>
```

where

`IFTP1(.)` = Newton function type
500 → H_{12} for segment at the design angle of attack α^* and specified Reynolds number
505 → H_{32} for segment at the design angle of attack α^* and specified Reynolds number
`JSEGIX1(.)` = segment for which parameter is specified
`LLBOS(.)` = 1 → indicates value for beginning of segment
0 → indicates value for end of segment
`COND1(.)` = Reynolds number for specified value
`FNEWT1(.)` = desired value for specified parameter
`ITP1(.)` = input variable for iteration (see Table 1)
`ITP2(.)` = input variable for iteration (see Table 1)
`CLAMP1(.)` = positive step limit used during iteration (optional)

```
NEWT1S2 <IFTP1(.)> <JSEGIX1(.)> <LLBOS(.)> <COND1(.)> <COND2(.)> -
        <FNEWT1(.)> <ITP1(.)> <ITP2(.)> | <CLAMP1(.)>
```

where

`IFTP1(.)` = Newton function type
600 → H_{12} for specified angle of attack and Reynolds number
601 → H_{32} for specified angle of attack and Reynolds number
`JSEGIX1(.)` = segment for which parameter is specified
`LLBOS(.)` = 1 → indicates value for beginning of segment
0 → indicates value for end of segment
`COND1(.)` = angle of attack for specified value
`COND2(.)` = Reynolds number for specified value
`FNEWT1(.)` = desired value for specified parameter
`ITP1(.)` = used to identify input variable for iteration (see Table 1)
`ITP2(.)` = used to identify input variable for iteration (see Table 1)
`CLAMP1(.)` = positive step limit used during iteration (optional)

8 Non-Constant Velocity Segment Prescriptions

As shown in the last example in Chapter 7, the `DELV`-line can be used to prescribe a non-constant velocity for a segment at the design angle of attack. When only the relative velocity at the end of the segment is specified, a single-line `DELV`-line is used. This first general form of the `DELV`-line is given by

```
DELV <IDVTP(JSEG)> <JSEG> <VTILDE(JSEG)>
```

where

`IDVTP(JSEG)` = $\tilde{v}$ type
  1 → linear velocity distribution in ϕ
  2 → parabolic velocity distribution in ϕ with slope matched at BOS
  3 → approximately linear in x/c
`JSEG` = $\tilde{v}$ prescribed for segment `JSEG`
`VTILDE(JSEG)` = relative velocity at the end of the segment (EOS)

When a distribution is specified by a series of collocation points (termed here subsegments), the following `DELV`-line is used.

```
DELV <IDVTP(JSEG)> <JSEG> <NPTS>
<WGHTPHI(1)> <SSDELV(1)>
...
<WGHTPHI(NPTS)> <SSDELV(NPTS)>
```

where

`IDVTP(JSEG)`= $\tilde{v}$ type
  11 → linear velocity distribution between subsegments
  13 → cubic velocity distribution with slope matched at BOS and zero second derivative at EOS
  14 → cubic velocity distribution with zero second derivative at BOS and slope matched at EOS
  15 → cubic velocity distribution with slopes matched at BOS and EOS
  16 → cubic velocity distribution with zero second derivative at BOS and EOS
`JSEG` = $\tilde{v}$ prescribed for segment `JSEG`
`NPTS` = number of $\tilde{\phi}$–$\tilde{v}$ pairs to follow
`WGHTPHI(.)` = nondimensional relative distance $\tilde{\phi}$ from BOS
`SSDELV(.)` = corresponding relative velocity $\tilde{v}(\tilde{\phi})$

References

[1] Selig, M.S. & Maughmer, M.D., "Multipoint Inverse Airfoil Design Method Based on Conformal Mapping," *AIAA J.*, Vol. 30, No 5, May 1992, pp. 1162–1170.

[2] Selig, M.S. & Maughmer, M.D., "Generalized Multipoint Inverse Airfoil Design," *AIAA J.*, Vol. 30, No. 11, November 1992, pp. 2618–2625.

[3] Eppler, R., *Airfoil Design and Data*, Springer-Verlag, New York, 1990.

[4] Eppler, R., "Direkte Berechnungvon Tragflügelprofilen aus der Druckverteilung," *Ingenieur-Archive*, Vol. 25, No. 1, 1957, pp. 32–57. (Translated as "Direct Calculation of Airfoils from Pressure Distribution," NASA TT F-15, 417, 1974.)

[5] Eppler, R. & Somers, D. M., "A Computer Program for the Design and Analysis of Low-Speed Airfoils," NASA TM-80210, August 1980.