

XFOIL 6.9 User Guide

Mark Drela, MIT Aero & Astro
Harold Youngren, Aerocraft, Inc.
11 Jan 2001

Contents

1	General Description	3
1.1	History	4
1.2	Theory References	4
1.3	Inviscid Formulation	5
1.4	Inverse Formulation	5
1.5	Viscous Formulation	5
2	Data Structure	6
2.1	Stored airfoils and polars	6
2.2	Current and buffer airfoils	6
3	Program Execution	7
3.1	Airfoil file formats	9
3.1.1	Plain coordinate file	9
3.1.2	Labeled coordinate file	9
3.1.3	ISES coordinate file	10
3.1.4	MSES coordinate file	10
3.2	Buffer airfoil normalization	10
3.3	Buffer airfoil generation via interpolation	10
3.4	Further buffer airfoil manipulation	11
3.5	Generation of current airfoil	11
3.6	Saving airfoil coordinates	11
4	Units	12
5	Analysis Routine (OPER)	12
5.1	Force calculation	16
5.2	Transition criterion	17
5.3	Numerical accuracy	18
5.3.1	Panel density requirements	18
5.3.2	Differencing order of accuracy	18

5.4	Viscous solution acceleration	18
5.5	Polar calculations and plotting	19
5.6	Off-line polar plotting	19
5.7	Re, Mach dependence on C_L	20
5.7.1	Type 1	20
5.7.2	Type 2	21
5.7.3	Type 3	21
6	Output	21
6.1	Plot Hardcopy	22
7	Full-Inverse Surface Speed Design Routine (MDES)	22
7.1	Creation of seed surface speed distribution	22
7.2	Modification of surface speed distributions	24
7.2.1	Cursor input of modifications	24
7.2.2	Modification endpoint blending	24
7.2.3	Smoothing	24
7.2.4	Symmetry forcing	24
7.2.5	New geometry computation	25
7.2.6	Multipoint surface speed display	25
7.3	Generation of new geometry	25
8	Mixed-Inverse Surface Speed Design Routine (QDES)	25
8.1	Creation of seed surface speed distribution	26
8.2	Modification of surface speed distribution	26
8.3	Generation of new airfoil geometry	26
9	Geometry Design Routine	27
9.1	Creating seed buffer airfoil	28
9.1.1	Point addition (typ. to Eppler and Selig airfoils)	28
9.2	Modifying buffer airfoil	29
9.3	Saving buffer airfoil into current airfoil	31
10	Start-up Defaults	31
11	Caveats	32

1 General Description

XFOIL is an interactive program for the design and analysis of subsonic isolated airfoils. It consists of a collection of menu-driven routines which perform various useful functions such as:

- Viscous (or inviscid) analysis of an existing airfoil, allowing
 - forced or free transition
 - transitional separation bubble(s)
 - limited trailing edge separation
 - lift and drag predictions just beyond $C_{L\max}$
 - Karman-Tsien compressibility correction
- Airfoil design and redesign by interactive specification of a surface speed distribution via screen cursor or mouse. Two such facilities are implemented.
 - Full-Inverse, based on a complex-mapping formulation
 - Mixed-Inverse, an extension of XFOIL's basic panel method

Full-inverse allows multi-point design, while Mixed-inverse allows relatively strict geometry control over parts of the airfoil.

- Airfoil redesign by interactive specification of new geometric parameters such as
 - new max thickness and/or camber
 - new LE radius
 - new TE thickness
 - new camber line via geometry specification
 - new camber line via loading change specification
 - flap deflection
 - explicit contour geometry (via screen cursor)
- Blending of airfoils
- Drag polar calculation with fixed or varying Reynolds and/or Mach numbers.
- Writing and reading of airfoil geometry and polar save files
- Plotting of geometry, pressure distributions, and polars (Versaplot-derivative plot package used)

XFOIL is best suited for use on a good workstation. A high-end PC is also effective, but must run Unix to support the X-Windows graphics. The source code of XFOIL is Fortran 77. The plot library also uses a few C routines for the X-Windows interface.

1.1 History

XFOIL 1.0 was written by Mark Drela in 1986. The main goal was to combine the speed and accuracy of high-order panel methods with the new fully-coupled viscous/inviscid interaction method used in the ISES code developed by Drela and Giles. A fully interactive interface was employed from the beginning to make it much easier to use than the traditional batch-type CFD codes. Several inverse modes and a geometry manipulator were also incorporated early in XFOIL's development, making it a fairly general airfoil development system.

Since version 1.0, XFOIL has undergone numerous revisions, upgrades, hacks, and enhancements. These changes mainly originated from perceived shortcomings during actual design use, so XFOIL is now strongly geared to practical airfoil development. Harold Youngren provided the Xplot11 plot package which is a vast improvement over the grim Versaplot-type package used initially. Enhancements and suggestions from Youngren and other people were also incorporated into XFOIL itself along the way.

Over the past few years, bug reports and enhancement suggestions have slowed to practically nil, and so after a final few enhancements from version 6.8, XFOIL 6.9 is officially "frozen" and has been made public. Although any bugs will likely be fixed, no further development is planned at this point. Method extensions are being planned, but these will be incorporated in a completely new next-generation code.

Note to code developers and code enhancers . . .

XFOIL does not exactly have the cleanest implementation, but it isn't too bad considering its vast modification history. Feel free to muck with the code as you like, provided everything is done under the GPL agreement. Drela and Youngren will not be inclined to assist with any code modifications at this point, however, since we each have a dozen other projects waiting. So proceed at your own risk.

1.2 Theory References

The general XFOIL methodology is described in

Drela, M.,
XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils,
Conference on Low Reynolds Number Airfoil Aerodynamics,
University of Notre Dame, June 1989.

which also appears as a chapter in:

Low Reynolds Number Aerodynamics. T.J. Mueller (Editor).
Lecture Notes in Engineering #54. Springer Verlag. 1989.
ISBN 3-540-51884-3
ISBN 0-387-51884-3

The boundary layer formulation used by XFOIL is described in:

Drela, M. and Giles, M.B.
Viscous-Inviscid Analysis of Transonic and Low Reynolds Number Airfoils
AIAA Journal, 25(10), pp.1347-1355, October 1987.

The blunt trailing edge treatment is described in:

Drela, M.,
Integral Boundary Layer Formulation for Blunt Trailing Edges,
Paper AIAA-89-2200, August 1989.

Other related literature:

Drela, M.,
Elements of Airfoil Design Methodology,
Applied Computational Aerodynamics, (P. Henne, editor),
AIAA Progress in Aeronautics and Astronautics, Volume 125, 1990.

Drela, M.,
Low-Reynolds Number Airfoil Design for the MIT Daedalus Prototype: A Case Study,
Journal of Aircraft, 25(8), pp.724-732, August 1988.

Drela, M.,
Pros and Cons of Airfoil Optimization,
Chapter in *Frontiers of Computational Fluid Dynamics, 1998*
D.A. Caughey, M.M. Hafez, Eds., World Scientific, ISBN 981-02-3707-3

1.3 Inviscid Formulation

The inviscid formulation of XFOIL is a simple linear-vorticity stream function panel method. A finite trailing edge base thickness is modeled with a source panel. The equations are closed with an explicit Kutta condition. A high-resolution inviscid calculation with the default 160 panels executes in less than one second on a workstation. Subsequent operating points for the same airfoil but different angles of attack are obtained nearly instantly.

A Karman-Tsien compressibility correction is incorporated, allowing good compressible predictions all the way to sonic conditions. The theoretical foundation of the Karman-Tsien correction breaks down in supersonic flow, and as a result accuracy rapidly degrades as the transonic regime is entered. Of course, shocked flows cannot be predicted with any certainty.

1.4 Inverse Formulation

There are two types of inverse methods incorporated in XFOIL: Full-Inverse and Mixed-Inverse. The Full-Inverse formulation is essentially Lighthill's and van Ingen's complex mapping method, which is also used in the Eppler code and Selig's PROFOIL code. It calculates the entire airfoil geometry from the entire surface speed distribution. The Mixed-Inverse formulation is simply the inviscid panel formulation (the discrete governing equations are identical) except that instead of the panel vortex strengths being the unknowns, the panel node coordinates are treated as unknowns wherever the surface speed is prescribed. Only a part of the airfoil is altered at any one time, as will be described later. Allowing the panel geometry to be a variable results in a non-linear problem, but this is solved in a straightforward manner with a full-Newton method.

1.5 Viscous Formulation

The boundary layers and wake are described with a two-equation lagged dissipation integral BL formulation and an envelope e^n transition criterion, both taken from the transonic analysis/design ISES code. The entire viscous solution (boundary layers and wake) is strongly interacted with the incompressible potential flow via the surface transpiration model (the alternative displacement body model is used in ISES). This permits proper calculation of limited separation regions. The drag is determined from the wake momentum thickness far downstream. A special treatment is used for a blunt trailing edge which fairly accurately accounts for base drag.

The total velocity at each point on the airfoil surface and wake, with contributions from the freestream, the airfoil surface vorticity, and the equivalent viscous source distribution, is obtained from the panel solution with the Karman-Tsien correction added. This is incorporated into the viscous equations, yielding a nonlinear elliptic system which is readily solved by a full-Newton method as in the ISES code. Execution times are quite rapid, requiring a few seconds on a fast workstation for a high-resolution calculation with 160 panels. For a sequence of closely spaced angles of attack (as in a polar), the calculation time per point can be substantially smaller.

If lift is specified, then the wake trajectory for a viscous calculation is taken from an inviscid solution at the specified lift. If α is specified, then the wake trajectory is taken from an inviscid solution at that α . This is not strictly correct, since viscous effects will in general decrease lift and change the trajectory. This secondary correction is not performed, since a new source influence matrix would have to be calculated each time the wake trajectory is changed. This would result in unreasonably long calculation times. The effect of this approximation on the overall accuracy is small, and will be felt mainly near or past stall, where accuracy tends to degrade anyway. In attached cases, the effect of the incorrect wake trajectory is imperceptible.

2 Data Structure

XFOIL stores all its data in RAM during execution. Saving of the data to files is *not* normally performed automatically, so the user must be careful to save work results before exiting XFOIL. The exception to this is optional automatic saving to disk of polar data as it's being computed in OPER (described later).

2.1 Stored airfoils and polars

XFOIL 6.9 stores multiple polars and associated airfoils and parameters during one interactive session. Each such data set is designated by its "stored polar" index:

```
polar 1:  x,y, CL(a), CD(a)...  Re, Ma, Ncrit...
polar 2:  x,y, CL(a), CD(a)...  Re, Ma, Ncrit...
.
.
```

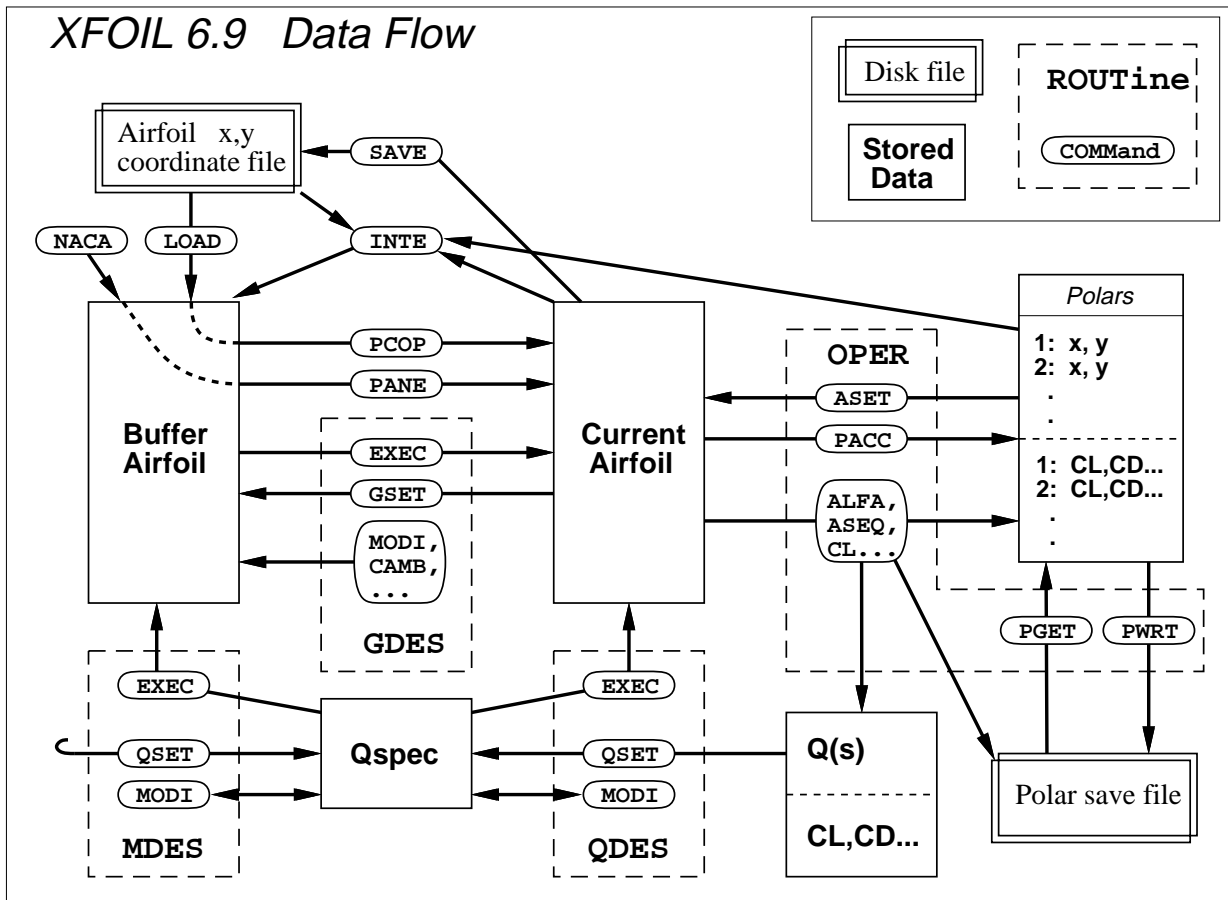
Not all of the data need to be present for each stored polar. For example, x, y would be absent if the CL,CD polar was read in from an external file rather than computed online.

Earlier XFOIL versions in effect only allowed one stored airfoil and stored polar at a time. The new multiple storage feature makes iterative redesign considerably more convenient, since the cases can contain multiple design versions which can be easily overlaid on plots.

2.2 Current and buffer airfoils

XFOIL 6.9 retains the concept of a "current airfoil" and "buffer airfoil" used in previous versions. These are the airfoils on which the various calculations are performed, and they are distinct from the "polar" x, y coordinates described above. The polar x, y are simply archived data, and do not directly participate in computations. The polar x, y must first be transferred into the current airfoil if they are to be used for computation.

The figure below indicates the data flows resulting from user commands described in the remainder of this manual.



3 Program Execution

XFOIL is simply executed with:

```
% xfoil
```

When the program starts, the following top level menu and prompt appear:

```

QUIT   Exit program

.OPER   Direct operating point(s)
.MDES   Complex mapping design routine
.QDES   Surface speed design routine
.GDES   Geometry design routine

SAVE f Write airfoil to labeled coordinate file
```

```

PSAV f Write airfoil to plain coordinate file
ISAV f Write airfoil to ISES coordinate file
MSAV f Write airfoil to MSES coordinate file
REVE   Reverse written-airfoil node ordering

LOAD f Read buffer airfoil from coordinate file
NACA i Set NACA 4,5-digit airfoil and buffer airfoil
INTE   Set buffer airfoil by interpolating two airfoils
NORM   Buffer airfoil normalization toggle

BEND   Display structural properties of current airfoil

PCOP   Set current-airfoil panel nodes directly from buffer airfoil points
PANE   Set current-airfoil panel nodes ( 140 ) based on curvature
.PPAR  Show/change paneling

.PLOP  Plotting options

WDEF f Write   current-settings file
RDEF f Reread  current-settings file
NAME s Specify new airfoil name
NINC   Increment name version number

Z      Zoom    | (available in all menus)
U      Unzoom  |

XFOIL  c>

```

The commands preceded by a period place the user in another lower-level menu. The other commands are executed immediately and the user is prompted for another top level command. The lowercase letters *i,r,f,s* following some commands indicate the type of argument(s) expected by the command:

```

i  integer
r  real
f  filename
s  character string

```

Commands will be shown here in uppercase, although they are not case sensitive.

Typically, either the **LOAD** or the **NACA** command is issued first to create an airfoil for analysis or redesign. The **NACA** command expects an integer argument designating the airfoil:

```
XFOIL  c>  NACA 4415
```

As with all commands, omitting the argument will produce a prompt:

```
XFOIL  c>  NACA
```

```
Enter NACA 4 or 5-digit airfoil designation  i>  4415
```


The `LOAD` command reads and processes a formatted airfoil coordinate file defining an arbitrary airfoil. It expects a filename argument:

```
XFOIL  c>  LOAD e387.dat
```

The `NACA` or `LOAD` commands can be skipped if `XFOIL` is executed with a filename as an argument, as for example

```
% xfoil e387.dat
```

which then executes the `LOAD` procedure before the first menu prompt is given.

3.1 Airfoil file formats

`LOAD` recognizes four file formats: Plain, Labeled, ISES, MSES. All data lines are significant with the exception of lines beginning with “#”, which are ignored.

3.1.1 Plain coordinate file

This contains only the x, y coordinates, which run from the trailing edge, round the leading edge, back to the trailing edge in either direction:

```
X(1)  Y(1)
X(2)  Y(2)
.      .
.      .
X(N)  Y(N)
```

3.1.2 Labeled coordinate file

This is the same as the plain file, except that it also has an airfoil name string on the first line:

```
NACA 0012
X(1)  Y(1)
X(2)  Y(2)
.      .
```

This is deemed the most convenient format to use. The presence of the name string is automatically recognized if it does not begin with a Fortran-readable pair of numbers. Hence, “00 12 NACA Airfoil” cannot be used as a name, since the “00 12” will be interpreted as the first pair of coordinates. “0012 NACA” is OK, however.

Some Fortran implementations will also choke on airfoil names that begin with `T` or `F`. These will be interpreted as logical variables, defeating the name-detection logic. Beginning the name with `_T` or `_F` is a workable solution to this “feature”.

3.1.3 ISES coordinate file

This has four or five ISES grid domain parameters in addition to the name:

```
NACA 0012
-2.0 3.0 -2.5 3.0
X(1) Y(1)
X(2) Y(2)
. .
```

If the second line has four or more numbers, then these are interpreted as the grid domain parameters.

3.1.4 MSES coordinate file

This is the same as the ISES coordinate file, except that it can contain multiple elements, each one separated by the line

```
999.0 999.0
```

The user is asked which of these elements is to be read in.

3.2 Buffer airfoil normalization

XFOIL will normally perform all operations on an airfoil with the same shape and location in cartesian space as the input airfoil. However, if the normalization flag is set (toggled with the **NORM** command), the buffer airfoil coordinates will be immediately normalized to unit chord and the leading edge will be placed at the origin. A message is printed to remind the user.

3.3 Buffer airfoil generation via interpolation

The **INTE** command is new in XFOIL 6.9, and allows interpolating or "blending" of airfoils in various proportions. The polar shape of an interpolated airfoil will often be quite close to the interpolated polars of its two parent airfoils. Extrapolation can also be done by specifying a blending fraction outside the 0..1 range, although the resulting airfoil may be quite weird if the extrapolation is excessive.

A good way to use **INTE** is to "augment" or "tone down" the modifications to an airfoil performed in **MDES** or **GDES**. For example, say airfoil B is obtained by modifying airfoil A:

A → MDES → B

Suppose the modification changed A's polar in the right direction, but not quite far enough. The additional needed change can be done by extrapolating past airfoil B in **INTE**:

```
Airfoil "0": A
Airfoil "1": B
Interpolating fraction 0..1 : 1.4
Output airfoil: C
```

Plotted along the "modification axis", the airfoils are:

A	B	C	
0.0	1.0	1.4	...

So airfoil C has 40% more of the change received by B in the redesign. Airfoil C's polar will also be changed about 40% more as intended.

3.4 Further buffer airfoil manipulation

The GDES facility allows very extensive manipulation of the buffer airfoil. This will be described in much more detail in a later section. If only analysis is performed, the GDES facility would not normally be used.

3.5 Generation of current airfoil

When the buffer airfoil coordinates are read from a file during startup, or read in via the **LOAD** command, they are by default also copied directly into the "current", or working airfoil. Hence, no special action is needed to start analysis operations. However, if the starting buffer airfoil has a poor point distribution (too many, too few, poorly spaced, etc), one can use **PANE** to create a better panel node distribution for the current airfoil on the splined buffer airfoil shape. The paneling routine increases the point density in areas of high curvature (i.e. the leading edge) and at the trailing edge to a degree specified by the user. The user can also increase panel density over one additional interval on each airfoil side, perhaps near transition. The current-airfoil paneling can be displayed and/or modified with **PPAR**.

In some cases it is desirable to explicitly re-copy the buffer airfoil into the current airfoil via **PCOP**. In previous XFOIL versions this had to be done with the equivalent command sequence

```
LOAD
GDES
EXEC
```

With XFOIL 6.9, the **GDES**, **EXEC** commands after **LOAD** are now superfluous.

The **NACA** command automatically invokes the paneling routine to create a current airfoil with a suitable paneling.

3.6 Saving airfoil coordinates

A coordinate file in any one of these four formats can be written with the **PSAV**, **SAVE**, **ISAV**, or **MSAV** command, respectively. When issuing the **MSAV** command, the user is also asked which element in the file is to be overwritten. XFOIL can thus be used to easily "edit" individual elements in MSES multielement configurations. Of course, normalization should not be performed on an element if it is to be written back to the same multielement file. Only the current-airfoil coordinates can be saved to a file. If the buffer or polar x, y coordinates need to be saved, they must first be copied into the current airfoil.

4 Units

Most XFOIL operations are performed on the airfoil's cartesian coordinates x, y , which do not necessarily have a unit chord c . Since the chord is ambiguous for odd shapes, the XFOIL force coefficients C_L , C_D , C_M are obtained by normalizing the forces and moment with only the freestream dynamic pressure (the reference chord is assumed to be unity). Likewise, the XFOIL Reynolds number RE is defined with the freestream velocity and viscosity, and an implied unit chord:

$$\begin{array}{ll} C_L = L/q & V = \text{freestream speed} \\ C_D = D/q & \nu = \text{freestream kinematic viscosity} \\ C_M = M/q & \rho = \text{freestream density} \\ RE = V/\nu & q = 0.5 \rho V^2 \end{array}$$

The conventional definitions are

$$\begin{array}{l} C_l = L/qc \\ C_d = D/qc \\ C_m = M/qc^2 \\ Re_c = Vc/\nu \end{array}$$

so that the conventional and XFOIL definitions differ only by the chord factor c or c^2 .

For example, a NACA 4412 airfoil is operated in the OPER menu at

$$\begin{array}{l} RE = 500000 \\ \alpha = 3^\circ \end{array}$$

first with $c = 1.0$, and then with $c = 0.5$ (changed with `SCAL` command in the GDES menu, say).

The results produced by XFOIL are:

$$\begin{array}{ll} c = 1.0 & : \quad C_L = 0.80 \quad C_D = 0.0082 \quad (RE = 500000, Re_c = 500000) \\ c = 0.5 & : \quad C_L = 0.40 \quad C_D = 0.0053 \quad (RE = 500000, Re_c = 250000) \end{array}$$

Since C_L is not normalized with the chord, it is nearly proportional to the airfoil size. It is not exactly proportional, since the true chord Reynolds number Re_c is different, and there is always a weak Reynolds number effect on lift. In contrast, the C_D for the smaller airfoil is significantly greater than 1/2 times the larger-airfoil C_D , since chord Reynolds number has a significant impact on profile drag. Repeating the $c = 0.5$ case at $RE = 1000000$, produces the expected result that C_L and C_D are exactly 1/2 times their $c = 1.0$ values.

$$c = 0.5 \quad : \quad C_L = 0.40 \quad C_D = 0.0041 \quad (RE = 1000000, Re_c = 500000)$$

Although XFOIL performs its operations with no regard to the size of the airfoil, some quantities are nevertheless defined in terms of the chord length. Examples are the camber line shape and BL trip locations, which are specified in terms of the relative $x/c, y/c$ along and normal to the airfoil chord line. This is done only for the user's convenience. In the input and output labeling, " \mathbf{x}, \mathbf{y} " always refer to the cartesian coordinates, while " $\mathbf{x}/c, \mathbf{y}/c$ " refer to the chord- based coordinates which are shifted, rotated, and scaled so that the airfoil's leading edge is at $(x/c, y/c) = (0, 0)$, and the airfoil's trailing edge is at $(x/c, y/c) = (1, 0)$. The two systems coincide only if the airfoil is normalized.

5 Analysis Routine (OPER)

Issuing the `OPER` command will produce the prompt

.OPERi c>

Typing a "?" will result in the full OPER analysis menu being displayed:

```
<cr>      Return to Top Level
!          Redo last ALFA,CLI,CL,ASEQ,CSEQ,VELS

VISC r     Toggle Inviscid/Viscous mode
.VPAR      Change BL parameter(s)
RE  r      Change Reynolds number
MACH r     Change Mach number
TYPE i     Change type of Mach,Re variation with CL
ITER       Change viscous-solution iteration limit
INIT       Toggle BL initialization flag

ALFA r     Prescribe alpha
CLI  r     Prescribe inviscid CL
CL   r     Prescribe CL
ASEQ rrr   Prescribe a sequence of alphas
CSEQ rrr   Prescribe a sequence of CLs

SEQP       Toggle polar/Cp(x) sequence plot display
CINC       Toggle minimum Cp inclusion in polar
HINC       Toggle hinge moment inclusion in polar
PACC i     Toggle auto point accumulation to active polar
PGET f     Read new polar from save file
PWRT i     Write polar to save file
PSUM       Show summary of stored polars
PLIS i     List stored polar(s)
PDEL i     Delete stored polar
PSOR i     Sort stored polar
PPL0 ii.   Plot stored polar(s)
APL0 ii.   Plot stored airfoil(s) for each polar
ASET i     Copy stored airfoil into current airfoil
PREM ir    Remove one point from stored polar
PPAX       Change polar plot axis limits

RGET f     Read new reference polar from file
RDEL i     Delete stored reference polar

GRID       Toggle Cp vs x grid overlay
CREF       Toggle reference Cp data overlay
FREF       Toggle reference CL,CD.. data display

CPX        Plot Cp vs x
CPV        Plot airfoil with pressure vectors (gee wiz)
.VPLO      BL variable plots
.ANNO      Annotate current plot
HARD       Hardcopy current plot
```

```

SIZE r   Change plot-object size
CPMI r   Change minimum Cp axis annotation

BL  i    Plot boundary layer velocity profiles
BLC      Plot boundary layer velocity profiles at cursor
BLWT r   Change velocity profile scale weight

FMOM      Calculate flap hinge moment and forces
FNEW rr   Set new flap hinge point
VELS rr   Calculate velocity components at a point
DUMP f    Output Ue,Dstar,Theta,Cf vs s,x,y to file
CPWR f    Output x vs Cp to file
CPMN      Report minimum surface Cp
NAME s    Specify new airfoil name
NINC      Increment name version number

```

```
.OPERi  c>
```

Note that some commands expect multiple arguments.

Hopefully, most of the commands are self-explanatory. For inviscid cases, the **CLI** and **CL** commands are identical. For viscous cases, **CLI** is equivalent to specifying α , this being determined a priori from the specified lift coefficient via an inviscid solution. **CL** will return a viscous solution with the specified true viscous lift coefficient at an α which is determined as part of the solution (prescribing a **CL** above $C_{L\max}$ will cause serious problems, however!). The user is always prompted for any required input. When in doubt, typing a "?" will always produce a menu.

After an **ALFA**, **CL**, or **CLI** command is executed, the C_p vs x distribution is displayed, and can be displayed again at any time with **CPX**. If the viscous mode is active, the true viscous C_p is shown as a solid line, and the inviscid C_p at that same α is shown as a dashed line. Each dash covers one panel, so the local dashed line density is also a useful visual indicator of panel resolution quality. If the inviscid mode is active, only the inviscid C_p is shown as a solid line.

The difference between the true viscous C_p distribution (solid line) and the inviscid C_p distribution (dashed line) is due to the modification of the effective airfoil shape by the boundary layers. This effective airfoil shape is shown superimposed on the actual current airfoil shape under the C_p vs x plot. The gap between these effective and actual shapes is equal to the local displacement thickness δ^* , which can also be plotted in the **VPAR** menu. This is only about 1/3 to 1/2 as large as the overall boundary layer thickness, which can be visualized via the **BL** or **BLC** commands which display velocity profiles through the boundary layer. **BL** displays a number of profiles equally spaced around the airfoil's perimeter, while **BLC** displays profiles at cursor-selected locations. The zooming commands **Z**, **U**, may be necessary to better see these small profiles in most cases.

If the C_p reference data overlay option is enabled with **CREF**, initiating a C_p vs x plot will first result in the user being prompted for a formatted data file with the following format:

```

x(1)  Cp(1)
x(2)  Cp(2)
.      .
.      .

```

The C_p vs x plot is then displayed as usual but with the data overlaid. If FREF has been issued previously, then numerical reference values for C_L , C_D , etc. will be requested and added to the plot next to the computed values.

Boundary-layer quantities are plotted from the VPLO menu:

```

H      Plot kinematic shape parameter
DT     Plot top      side Dstar and Theta
DB     Plot bottom side Dstar and Theta
UE     Plot edge velocity
CF     Plot skin friction coefficient
CD     Plot dissipation coefficient
N      Plot amplification ratio
CT     Plot max shear coefficient
RT     Plot Re_theta
RTL    Plot log(Re_theta)

X rrr Change x-axis limits
Y rrr Change y-axis limits on current plot

BLOW   Cursor blowup of current plot
RESE   Reset to default x,y-axis limits
SIZE r Change absolute plot-object size
.ANNO  Annotate plot
HARD   Hardcopy current plot

GRID   Toggle grid plotting
SYMB   Toggle node-symbol plotting
LABE   Toggle label plotting
CLIP   Toggle line-plot clipping

..VPLO c>
```

This menu is largely self-explanatory. The skin friction coefficient plotted with the CF command is defined as

$$C_f = \tau / 0.5 \rho V_\infty^2$$

This differs from the standard boundary layer theory definition which uses the local U_e rather than V_∞ for the normalization. Using the constant freestream reference makes $C_f(x)$ have the same shape as the physical shear stress $\tau(x)$.

The dissipation coefficient C'_D (this is NOT the drag coefficient!!!) is plotted with the CD command. $C'_D(x)$ is proportional to the local energy dissipation rate due to viscous shear and turbulent mixing. Hence, it indicates where on the airfoil drag is being created. It is in fact a much better indicator of drag production than $C_f(x)$, since C_f does not account for pressure drag. C'_D , on the other hand, accounts for everything. Its relationship to the total profile drag coefficient is simply

$$C_D = \int_0^\infty 2 C'_D ds$$

with the integration performed over both boundary layers and also the wake. It will be seen that if the flow is separated at the trailing edge, much of the drag contribution (energy dissipation) of C'_D occurs in the wake.

As mentioned earlier, all forces are normalized with freestream dynamic pressure only. C_L , C_D , C_M are the usual chord-based definitions only if the airfoil has a unit chord – in general, they will scale with the airfoil's chord. Also, C_M is defined about the cartesian point $(x_{\text{ref}}, y_{\text{ref}}) = (0.25, 0.0)$, which is not necessarily the airfoil's 1/4 chord point.

5.1 Force calculation

The lift and moment coefficients C_L , C_M , are calculated by direct surface pressure integration:

$$C_L = L/q = \oint C_p d\bar{x} \quad C_M = M/q = \oint -C_p [(x - x_{\text{ref}}) dx + (y - y_{\text{ref}}) dy]$$

where $\bar{x} = x \cos(\alpha) + y \sin(\alpha)$
 $\bar{y} = y \cos(\alpha) - x \sin(\alpha)$

The integrals performed in the counterclockwise direction around the airfoil contour. The pressure coefficient C_p is calculated using the Karman-Tsien compressibility correction.

The drag coefficient C_D is obtained by applying the Squire-Young formula at the last point in the wake — *not* at the trailing edge.

$$C_D = D/q = 2\theta_\infty = 2\theta (u/V)^{(H+5)/2}$$

where $\theta = \text{momentum thickness}$
 $u = \text{edge velocity}$
 $H = \text{shape parameter}$ } at end of computed wake

and $\theta_\infty = \text{momentum thickness}$
 $V = \text{freestream velocity}$ } very far downstream

The Squire-Young formula in effect extrapolates the momentum thickness to downstream infinity. It assumes that the wake behaves in an asymptotic manner downstream of the point of application. This assumption is strongly violated in the near-wake behind an airfoil with trailing edge separation, but is always reasonable some distance behind the airfoil. Hence, the usual application of Squire-Young at the trailing edge is questionable with separation present, but its application at the last wake point (typically 1 chord downstream) is always reasonable. Also, application at the last wake point also results in the formula having a smaller effect in any case, since there $u \simeq V$, and hence $\theta_\infty \simeq \theta$.

In most 2-D airfoil experiments, drag is measured indirectly by measuring $2\theta/c$ in the wake, often within one chord of the airfoil's trailing edge. For consistency, this should be compared to the Theta value predicted by XFOIL at the same wake location, rather than the "true" $C_d = 2\theta_\infty/c$ value which is effectively at downstream infinity. In general, θ_∞ will be smaller than θ . In most airfoil drag measurement experiments, this difference may amount to the drag measurement being several percent too large, unless some correction is performed.

In addition to calculating the total viscous C_D from the wake momentum thickness, XFOIL also determines the friction and pressure drag components C_{D_f} , C_{D_p} of this total C_D . These are calculated by

$$C_{D_f} = \oint C_f d\bar{x} \quad C_{D_p} = C_D - C_{D_f}$$

Here, C_f is the skin friction coefficient defined with the freestream dynamic pressure, not the BL edge dynamic pressure commonly used in BL theory. Note that C_{D_p} is deduced from C_D and C_{D_f} instead of being calculated via surface pressure integration. This conventional definition

$$C_{D_p} = \oint C_p d\bar{y}$$

is *not* used, since it is typically swamped by numerical noise.

5.2 Transition criterion

Transition in an XFOIL solution is triggered by one of two ways:

- free transition: e^n criterion is met
- forced transition: a trip or the trailing edge is encountered

The e^n method is always active, and free transition can occur upstream of the trip. The e^n method has the user-specified parameter n_{crit} , which is the log of the amplification factor of the most-amplified frequency which triggers transition. A suitable value of this parameter depends on the ambient disturbance level in which the airfoil operates, and mimics the effect of such disturbances on transition. Below are typical values of n_{crit} for various situations.

Situation	n_{crit}
sailplane	12–14
motorglider	11–13
clean wind tunnel	10–12
average wind tunnel	9
dirty wind tunnel	4–8

The choice $n_{\text{crit}} = 9$ corresponds to the “standard” e^9 method, and is the most common choice. It must be pointed out that the e^n method in XFOIL is actually the simplified envelope version, which is the same as the full e^n method only for flows with constant $H(x)$. If H is not constant, the two methods differ somewhat, but this difference is typically within the uncertainty in choosing n_{crit} .

The e^n method is only appropriate for predicting transition in situations where the growth of 2-D Tollmien-Schlichting waves via linear instability is the dominant transition-initiating mechanism. Fortunately, this happens to be the case in a vast majority of airfoil applications. Other possible mechanisms are:

- Crossflow instabilities. These occur on swept wings with significant favorable chordwise pressure gradients.
- Attachment-line transition. This requires large sweep, large LE radius, and a large Reynolds number. Occurs primarily on big jets.
- Bypass transition. This occurs in cases with sufficient wall roughness and/or large freestream turbulence or vibration levels. The linear-instability phase predicted by the e^n method is “bypassed”, giving relatively early transition. Usually occurs in favorable pressure gradients, while the linear-instability mechanism usually dominates in adverse pressure gradients.

If any of these alternative transition mechanisms are present, the trips must be set to mimick their effect. The bypass transition mechanism can be mimicked to some extent by the e^n method by

setting n_{crit} to a small value — $n_{\text{crit}} = 1$ or less. This will cause transition just after linear instability begins. For very large freestream turbulence or roughness in favorable pressure gradients, bypass transition can occur before the linear instability threshold, and in this case trips will have to be set as well.

5.3 Numerical accuracy

5.3.1 Panel density requirements

If strong separation bubbles are present in a viscous solution, then it is very important to have good panel resolution in the region of the bubble(s). The large gradients at a bubble tend to cause significant numerical errors even if a large number of panels is used. If a separation bubble appears to be poorly resolved, it is a good idea to re-panel the airfoil with more points, and/or with points bunched around the bubble region. The paneling is controlled from the PPAR menu. A good rule of thumb is that the shape parameter H_k just after transition in the bubble should not decrease by more than 1.0 per point. Likewise, the surface velocity U_e/V_∞ should not change by more than 0.05 per point past transition, otherwise there may be significant numerical errors in the drag. The point values can be observed by issuing SYMB from the VPLO menu. Moderate chord Reynolds numbers (1–3 million, say) usually require the finest paneling, since the bubbles are still important, but very small. On many airfoils, especially those with small leading edge radii, the development of the small bubble which forms just behind the leading edge can have a significant effect on $C_{L\text{max}}$. For such cases, the default paneling density at the bubble may not be adequate. In all cases, inadequate bubble resolution results in a "ragged" or "scalloped" C_L vs C_D drag polar curve, so fortunately this is easy to spot.

5.3.2 Differencing order of accuracy

The BL equations are normally discretized with two-point central differencing (i.e. the Trapezoidal Scheme), which is second-order accurate, but only marginally stable. In particular, it has problems with the relatively stiff shape parameter and lag equations at transition, where at high Reynolds number the shape parameter must change very rapidly. Oscillations and overshoots in the shape parameter will occur with the Trapezoidal Scheme if the grid cannot resolve this rapid change. To avoid this nasty behavior, upwinding must be introduced, resulting in the Backward Euler Scheme, which is very stable, but has only first-order accuracy. Previous versions of XFOIL allowed a specific constant amount of upwinding to be user-specified. Currently, XFOIL automatically introduces upwinding into the equations only in regions of rapid change (typically transition). This ensures that the overall scheme is stable and as accurate as possible.

Since only a minimal amount of upwinding is introduced in the interest of numerical accuracy, small oscillations in the shape parameter H will sometimes appear near the stagnation point if relatively coarse paneling is used there. These oscillations are primarily a cosmetic defect, and do not significantly affect the downstream development of the boundary layer. Eliminating them by increasing upwinding would in fact produce much greater errors in the overall viscous solution.

5.4 Viscous solution acceleration

The execution of a viscous case requires the solution of a large linear system every Newton iteration. The coefficient matrix of this system is 1/3 full, although most of its entries are very small.

Substantial savings in CPU time (factor of 4 or more) result when these small entries are neglected. SUBROUTINE BLSOLV which solves the large Newton system ignores any off-diagonal element whose magnitude is smaller than the variable VACCEL, which is initialized in SUBROUTINE INIT, and which can be changed at runtime from the VPAR menu with the VACC command.

A nonzero VACCEL parameter should in principle degrade the convergence rate of the viscous solution and thus result in more Newton iterations, although the effect is usually too small to notice. For very low Reynolds number cases (less than 100000), it *may* adversely affect the convergence rate or stability, and one should try reducing VACCEL or even setting it to zero if all other efforts at convergence are unsuccessful. The value of VACCEL has absolutely no effect on the final converged viscous solution (if attained).

5.5 Polar calculations and plotting

The polar calculation facility driven from the OPER menu deserves a detailed description. It has been considerably upgraded from previous XFOIL versions.

The simplest way to create a polar is to issue the PACC command which sets the auto polar accumulation toggle and asks for the optional save and dump filenames. If either filename is given, each computed operating point will be stored internally and also written to the specified file. If no filename is given, the automatic writing is not performed.

The polar's operating points can be computed individually with ALFA, or more conveniently en masse with ASEQ. One can also use CL or CSEQ, although these will not work close to C_{Lmax} .

The polar can be plotted anytime with PPLO. If previous polars have been computed or read in with PGET, they can be plotted as well. If a polar is deemed incomplete, additional points can be computed as needed.

If automatic writing of a polar was not chosen (no filename was given for PACC), the polar can be written later all at once with the PWRT command. The only drawback to this approach is that if the program crashes during a polar calculation sweep for whatever reason, the computed polar and all other stored information will be lost.

If existing filenames are given to PACC, the subsequent computed points will be appended to these files, but only if the airfoil name and flow parameters in the file match the current parameters. This is to prevent clobbering of the polar file with "wrong" additional points. Messages are always produced informing the user of what's going on.

5.6 Off-line polar plotting

Polar save file(s) can also be plotted off-line with the separate program PPLOT. This is entirely menu driven, and is simply executed:

```
% pplot
```

The file pplot.def contains plotting parameters, and is read automatically if available. If it's not available, then internal defaults are used.

Like the RGET, FREF commands in OPER, PPLOT permits reference data to be overlaid. A reference polar data file has the following form:

```

CD(1)  CL(1)
CD(2)  CL(2)
.      .
.      .
999.0  999.0
alpha(1) CL(1)
alpha(2) CL(2)
.      .
.      .
999.0  999.0
alpha(1) Cm(1)
alpha(2) Cm(2)
.      .
.      .
999.0  999.0
Xtr/c(1) CL(1)
Xtr/c(2) CL(2)
.      .
.      .
999.0  999.0

```

The number of points in each set (CD-CL, alpha-CL, etc.) is arbitrary, and can be zero.

The contents of a polar dump file can be selectively plotted with the separate menu-driven program PXPLOT. It is executed with:

```
% pxplot <dump filename>
```

This allows surface plots of C_p vs x , H vs x , etc. for any or all of the saved operating points. Of course, these plots can be generated in XFOIL for any individual operating point, so PXPLOT and the dump file itself are somewhat redundant in this respect.

5.7 Re, Mach dependence on C_L

A few comments are in order on the **TYPE** command, which allows the user to set the dependence of the Mach and Reynolds numbers on C_L . Any C_L - C_D polar can be of the following three types:

Type	parameters held constant	varying	fixed
1	M , RE	lift	chord, velocity
2	$M\sqrt{C_L}$, $RE\sqrt{C_L}$	velocity	chord, lift
3	M , $RE C_L$	chord	lift , velocity

5.7.1 Type 1

This corresponds to a given wing at a fixed velocity going over an angle of attack range, as in a wind tunnel test alpha sweep or a sudden aircraft pullup. This is also the common form for an airfoil polar.

5.7.2 Type 2

This corresponds to an aircraft in level flight at a given altitude undergoing trim speed changes. This is the most useful airfoil polar form for determining a drag polar for an aircraft at 1-g. For this case, The "Mach number" input with the **MACH** command is actually interpreted as the product $M\sqrt{C_L}$, and the "Reynolds number" input with the **VISC** or **RE** commands is actually interpreted as $RE\sqrt{C_L}$. For a wing in level flight, these products can be computed from the following exact relations, with RE based on the mean chord:

$$M\sqrt{C_L} = \left[\frac{2W/S}{\gamma p} \right]^{1/2} \qquad RE\sqrt{C_L} = \frac{1}{\nu} \left[\frac{2W}{\rho AR} \right]^{1/2}$$

W = weight	p = ambient pressure
S = wing area	ν = kinematic viscosity
AR = aspect ratio	ρ = ambient air density
b = span	$\gamma = 1.4$

5.7.3 Type 3

This corresponds to a wing of "rubber chord" with a given lift at a given speed. This is best used for selecting an optimum C_L for an airfoil while taking Reynolds number changes into account. The product $RE C_L$ can be computed from the following:

$$RE C_L = \frac{2W}{\nu \rho V b}$$

Caution must be used with Types 2 and 3 so as to not allow the C_L to go negative. In addition, with non-zero Mach and Type 2, the C_L must not fall below that value which makes Mach exceed unity. Warning messages are printed when these problems occur.

6 Output

All output goes directly to the terminal screen. H. Youngren's plot package Xplot11 (**libPlt.a**) used by XFOIL drives monochrome and color X-Windows graphics, and generates B&W or color PostScript files for hardcopy. The default setup assumes color X-Windows graphics (if available), and B&W PostScript. These defaults are controlled by the **IDEV** and **IDEVRP** flags in SUBROUTINE INIT (in **xfoil.f**).

The Xplot11 library should work on all Unix systems. The Makefile in the **./plotlib/** directory requires some modifications for some machines.

The default X-graphics window is in Landscape mode, with a black (reverse-video) background. A white (normal-video) background can be selected by setting the Unix shell variable

```
% setenv XPLOT11_BACKGROUND white
```

before Xfoil is started. The nicer reverse-video is restored with:

```
% unsetenv XPLOT11_BACKGROUND
```

See the `plotlib/Doc` file for more info on the plot library.

Xplot11 provides a built-in Zoom/Unzoom capability which can be applied to whatever is on the screen. Zooming/Unzooming can be performed with the `Z` and `U` commands from nearly all the menus — these commands are not listed to reduce clutter.

Some of the menus also have their own Blowup/Reset commands. The distinction is that XFOIL's plots don't try to adjust themselves to Zoom parameters, so a highly-"Zoomed" plot may show nothing at all. In contrast, Blowup/Reset instructs XFOIL to change its own plot scales, so a highly-"Blown-up" plot will at least show the axes.

6.1 Plot Hardcopy

For hardcopy, the current screen plot can be echoed to a PostScript file `plot.ps` with the `HARD` command. The size of the plot objects on the screen and on hardcopy can be changed with the `SIZE` command from most menus. The number requested is the width of the plot in inches.

If the `plot.ps` file is to be previewed with some X-Windows PostScript viewer, or imported into word-processing systems, XFOIL must be exited with `QUIT` in order for the `plot.ps` file to be properly terminated. For just printing, this may or may not be necessary.

For the geometry plot in GDES, and the $Q_{\text{spec}}(s)$ plots in QDES and MDES (described below), the hardcopy plot size will also be affected if the graphics window is resized with the cursor at the window manager level. This is because the plot is always scaled so that it fills up as much of the window as possible. If the window size is left at its start-up size, the hardcopy plot widths will come out with the specified size in inches. If any window dimension is increased from its default value, then a subsequent hardcopy plot will probably not fit on a standard 8.5" x 11.0" sheet.

7 Full-Inverse Surface Speed Design Routine (MDES)

XFOIL's Full-Inverse complex-mapping facility MDES takes as input a speed distribution " Q_{spec} " specified over the entire airfoil surface, modifies it somewhat to satisfy the Lighthill constraints, and generates a new overall geometry. First a bit of the underlying theory...

The geometry and the surface velocities can both be computed from a set of complex mapping coefficients C_n in the form

$$\begin{aligned} x + iy &= z(\omega; C_n) \\ u - iv &= w(\omega; C_n, \alpha) \end{aligned}$$

where $\omega = 0..2\pi$ is the independent parameter going around the airfoil. The z and w functions are rather complicated but this is not important here. The key to the full-inverse method is that the mapping coefficients C_n can be computed from a known contour angle $\theta(\omega) = \arctan(dy/dz)$ OR from a surface speed $Q(\omega) = |u - iv| = |w|$. The other quantity then follows. In summary, the operations and their commands are...

- a) Direct problem: $\theta \rightarrow C_n \rightarrow u - iv, Q$ (**QSET**)
- b) Inverse problem: $Q_{\text{spec}} \rightarrow C_n \rightarrow x + iy, \theta$ (**EXEC**)

7.1 Creation of seed surface speed distribution

MDES performs **QSET** and sets $Q_{\text{spec}} = Q$ automatically upon entry if Q_{spec} does not exist.

This default initialization in effect makes MDES a redesign method in which the surface speed distribution of an existing airfoil is used as a starting point to generate a new speed distribution. A “pure” design code which requests the entire surface speed distribution every time is often less natural to use, since airfoil design is invariably an iterative process involving repeated analyze/fix cycles. The MDES menu is shown below.

```

<cr>   Return to Top Level
!       Redo previous command

INIT    Re-initialize mapping
QSET    Reset Q_spec <== Q
AQ r..  Show/select alpha(s) for Qspec
CQ r..  Show/select CL(s)   for Qspec

SYMM    Toggle symmetry flag
TGAP r  Set new TE gap
TANG r  Set new TE angle

MODI    Modify Qspec
MARK    Mark off target segment for smoothing
SMOO    Smooth Qspec inside target segment
FILT    Apply Hanning filter to entire Qspec
SLOP    Toggle modified-Qspec slope matching flag

EXEC    Execute full-inverse calculation

VISC    Qvis overlay toggle
REFL    Reflected Qspec overlay toggle
PLOT    Plot Qspec (line) and Q (symbols)
SPEC    Plot mapping coefficient spectrum

BLOW    Blowup plot region
RESE    Reset plot scale and origin
SIZE r  Change absolute plot-object size
.ANNO    Annotate plot
HARD    Hardcopy current plot

PERT    Perturb one Cn and generate perturbed geometry

.MDES  c>

```

As described above, the initial Q_{spec} distribution is taken from Q , the speed distribution corresponding to the current geometry at the last angle of attack employed in OPER. Q_{spec} can be set back to this Q with QSET anytime.

7.2 Modification of surface speed distributions

7.2.1 Cursor input of modifications

Q_{spec} can be modified to whatever is desired with the **MODI** command by specifying points with the screen cursor which are then splined. The points can be entered in any order. The previously-input points can be erased one by one by clicking on the "Erase" button or simply typing **e** in the graphics window. The input sequence is terminated by clicking on the "Done" button or by typing **d** in the graphics window. The "Abort" button or typing **a** aborts the **MODI** command and returns to the **MDES** menu. The **BLOW** command can be used to enlarge regions of interest at any time by specifying opposite corners of the blowup region.

7.2.2 Modification endpoint blending

Normally, the modified piece of $Q_{\text{spec}}(s)$ is blended into the current $Q_{\text{spec}}(s)$ with matching slopes at the piece endpoints. This slope matching can be turned on/off with the **SLOP** toggle command. If slope matching is turned off, the modified piece will match only the existing value, but a slope discontinuity will be allowed.

7.2.3 Smoothing

Q_{spec} can be smoothed with the **SMOO** command, which normally operates on the entire distribution, but can be confined to a target segment whose endpoints are selected with the **MARK** command. The smoothing acts to alleviate second derivatives in $Q_{\text{spec}}(s)$, so that with many consecutive **SMOO** commands $Q_{\text{spec}}(s)$ will approach a straight line over the target segment. If the slope-matching flag is set, the endpoint slopes are preserved.

The **FILT** command is an alternative smoothing procedure which acts on the Fourier coefficients of Q_{spec} directly, and is global in its effect. It is useful for "cleaning up" the entire $Q_{\text{spec}}(s)$ distribution if noise is present from some geometric glitch on the airfoil surface. Also, unintended noise might be introduced into Q_{spec} from a poor modification via the cursor.

FILT acts by multiplying the Fourier coefficients by a Hanning window filter function raised to the power of a filter parameter F . This tapers off the high frequencies of Q_{spec} to varying degrees. A value of $F = 0.0$ gives no filtering, $F = 1.0$ gives the standard Hanning filter, $F = 2.0$ applies the Hanning filter twice, etc. The standard Hanning filter appears to be a bit too drastic, so a filter parameter of $F = 0.2$ is currently used. Hence, issuing **FILT** five times corresponds to the standard Hanning filter. The **SPEC** command displays the mapping coefficient spectrum at any time.

7.2.4 Symmetry forcing

The symmetry-forcing option (**SYMM** toggle) is useful when a symmetric airfoil is being designed. If active, this option zeroes out all antisymmetric (camber) Q_{spec} changes, and doubles all symmetric (thickness) changes. This unfortunately has the annoying side effect of also doubling the numerical roundoff noise in Q_{spec} every time a **MODI** operation is performed. This noise sooner or later becomes visible as high-frequency wiggles which double with each **MODI** command. Occasionally issuing **FILT** keeps this parasitic noise growth under control.

7.2.5 New geometry computation

The **MODI**, **BLOW**, **MARK**, **SMOO**, **SLOP**, **FILT** commands can be issued repeatedly in any order until Q_{spec} is modified to have the desired distribution. In general, the speed distributions actually plotted will not exactly match what was input with the cursor, since corrections are automatically added to maintain the specified trailing edge gap and to enforce consistency with the freestream speed. These are known as the Lighthill constraints on the surface speed.

The trailing edge gap is initialized from the initial airfoil and can be changed with **TGAP**. To reduce the "corrupting" effect of the constraint-driven corrections, a good rule of thumb is that the Q_{spec} distribution should be modified so as to preserve the total C_L . The C_L is simply twice the area under the $Q_{\text{spec}}(s)$ curve ($= 2 \times \text{circulation}$), so that this area should be preserved.

7.2.6 Multipoint surface speed display

A very useful feature of the MDES facility is the ability to display and modify a number of Q_{spec} distributions corresponding to different alpha or inviscid C_L values. These values are displayed and/or selected via the **AQ** or **CQ** commands. When any one Q_{spec} distribution is modified, the result of modification is also displayed on all the other distributions. This allows rapid design at multiple operating points. When the Q_{spec} curves correspond to specified C_L values, the alpha for each curve will be adjusted after each Q_{spec} modification so as to preserve that curve's C_L . The resulting Q_{spec} will therefore not match the input cursor points exactly because of this alpha correction.

7.3 Generation of new geometry

The **EXEC** command generates a new buffer airfoil corresponding to the current Q_{spec} distribution. If subsequent operations on this airfoil are to be performed (**SAVE**, **OPER**, etc.), it is necessary to first generate a current airfoil from this buffer airfoil using **PANE** at the top level menu. This seemingly complicated sequence is necessary because the airfoil points generated by **EXEC** are uniformly spaced in the circle plane, which gives a rather poor point (panel node) spacing distribution on the physical airfoil. This sequence also prevents the current airfoil from being overwritten immediately when **EXEC** is issued. Once the new current airfoil is generated with **PANE**, it can then be analyzed in **OPER**, modified in **GDES**, or whatever.

The **PERT** command allows manual input of the complex mapping coefficients C_n which determine the geometry. These coefficients are normally determined from $Q_{\text{spec}}(s)$ (this is the essence of the inverse method). The **PERT** command is provided simply as a means of allowing generation of geometric perturbation modes, possibly for external optimization or whatever.

The manually-changed C_n values result in changes in geometry as well as the current $Q_{\text{spec}}(s)$ distributions. The **QSET** command will restore everything to its unperturbed state.

The Full-Inverse facility is very fast, after an initialization calculation of several seconds (on a RISC workstation), it requires only a fraction of a second to generate the new buffer airfoil.

8 Mixed-Inverse Surface Speed Design Routine (QDES)

XFOIL's Mixed-Inverse facility QDES is useful in certain redesign problems where parts of the airfoil cannot be altered under any circumstances. The Mixed-Inverse menu is shown below.

```

<cr>   Return to Top Level

QSET    Reset Qspec <== Q

MARK    Mark off target segment
MODI    Modify Qspec
SMOO    Smooth Qspec inside target segment
SLOP    Toggle modified-Qspec slope matching flag

EXEC i  Execute mixed-inverse calculation
REST    Restore geometry from buffer airfoil
CPXX    CPxx endpoint constraint toggle

VISC    Qvis overlay toggle
REFL    Reflected Qspec overlay toggle

PLOT    Plot Qspec (line) and Q (symbols)
BLOW    Blowup plot region
RESE    Reset plot scale and origin
SIZE r  Change absolute plot-object size
.ANNO    Annotate plot
HARD    Hardcopy current plot

.QDES   c>

```

8.1 Creation of seed surface speed distribution

The QDES menu above is intentionally geared for the redesign of a segment of an existing airfoil (with its surface speed distribution calculated previously in OPER) rather than the generation of a totally new airfoil. When QDES is entered, the specified speed distribution Q_{spec} is initialized to the current speed distribution Q last set in OPER. If a direct solution for the current airfoil hasn't been calculated yet, QDES goes ahead and calculates it, using the last-set angle of attack. If this isn't the desired angle, it can be set in OPER using ALFA. QSET can then be used to set Q_{spec} from the current Q distribution.

8.2 Modification of surface speed distribution

Q_{spec} can be repeatedly modified with the screen cursor and the MODI command, exactly as in MDES. It is also necessary to mark off the target segment where the geometry is to be modified with the MARK command.

8.3 Generation of new airfoil geometry

EXEC modifies the airfoil over the target segment to match Q_{spec} there as closely as possible. The remainder of the airfoil geometry is left unaltered. EXEC requests the number of Newton iterations to be performed in the inverse calculation. Although as many as six iterations may be required for convergence to machine zero, it is *not* necessary to fully converge a Mixed-Inverse case. Two

iterations are usually sufficient to get very close to the new geometry. In any case, the new surface speed distribution Q which actually results from the inverse calculation will typically differ somewhat from the specified distribution Q_{spec} by function modes which are added to Q_{spec} . At least two modes are added, with their magnitudes determined by geometric closure requirements at the inverse segment endpoints. As with the MDES complex-mapping routine, the necessary modifications to Q_{spec} will be smallest if Q_{spec} is modified so that C_L (the area under the $Q_{\text{spec}}(s)$ curve) is roughly preserved.

Issuing **PLOT** after the **EXEC** command finishes will compare the specified (Q_{spec}) and resulting (Q) speed distributions. If extra smoothness in the surface speed is required, the **CPXX** command just before **EXEC** will enable the addition of two additional modes which allow the second derivative in the pressure at the endpoints to be unchanged from the starting airfoil. The disadvantage of this option is that the resulting surface speed Q will now deviate more from the specified speed Q_{spec} . It is allowable to repeatedly modify Q_{spec} , set or reset the **CPXX** option, and issue the **EXEC** command in any order.

The Mixed-Inverse modification is performed on the current airfoil directly, in contrast to Full-Inverse which generates the buffer airfoil as its output. In fact, it is important *not* to issue the **PANE** or **PCOP** commands at top level after doing work in the QDES menu, as the new current airfoil will be overwritten with the old buffer airfoil.

9 Geometry Design Routine

Executing the **GDES** command from the top level menu will put the user into the GDES routine. It has a rather extensive menu:

```

<cr>      Return to Top Level
!          Redo previous command

GSET       Set buffer  airfoil <== current airfoil
EXEC       Set current airfoil <== buffer  airfoil
SYMM       Toggle y-symmetry flag

ADEG r     Rotate about origin (degrees)
ARAD r     Rotate about origin (radians)
TRAN rr    Translate
SCAL r     Scale about origin
DERO       Derotate (set chord line level)
LINS r..   Linearly-varying y scale

TGAP rr    Change trailing edge gap
LERA rr    Change leading edge radius

TCPL       Toggle thickness and camber plotting
TFAC rr    Scale existing thickness and camber
TSET rr    Set new thickness and camber
HIGH rr    Move camber and thickness highpoints
.CAMB      Modify camber shape directly or via loading

```

```

FLAP rrr Deflect trailing edge flap

MODI      Modify contour via cursor
SLOP      Toggle modified-contour slope matching flag

CORN      Double point with cursor (set corner)
ADDP      Add    point with cursor
DELP      Delete point with cursor
MOVP      Move   point with cursor

UNIT      Normalize buffer airfoil to unit chord
DIST      Determine distance between 2 cursor points
CLIS      List curvatures
CANG      List panel corner angles
CADD ri.  Add points at corners exceeding angle threshold

PLOT      Replot buffer airfoil
BLOW      Blowup plot region
RESE      Reset plot scale and origin
TICK      Toggle node tick-mark plotting
GRID      Toggle grid plotting
GPAR      Toggle geometric parameter plotting
OVER f    Overlay disk file airfoil

SIZE r    Change absolute plot-object size
.ANNO      Annotate plot
HARD      Hardcopy current plot

NAME s    Specify new airfoil name
NINC      Increment name version number

.GDES  c>

```

9.1 Creating seed buffer airfoil

The first command typically executed is **GSET**, which sets the temporary buffer airfoil from the current airfoil. Sometimes it might be desired to operate directly on the coordinates of an already existing buffer airfoil. It typically contains coordinates read in from a disk file by **LOAD** at Top Level, or coordinates generated by **EXEC** from the MDES menu, depending on what was done last. In either of these cases, **GSET** is skipped.

9.1.1 Point addition (typ. to Eppler and Selig airfoils)

If the buffer airfoil has an excessively coarse point spacing, additional points can be added with the **CADD** command. Using the **PANE** command at top level also does this, but **CADD** allows the point addition to be restricted to locations with excessive corner angles (displayed with **CANG**), and also to locations which fall within a specified x-range. Different spline parameters can also be used to determine the inserted spline points. For example, the command

```
.GDES  c>  CADD 10.0  2  -0.1 0.2
```

will add spline points adjacent to each existing point whose panel angle exceeds ± 10 degrees, and only if the added point will fall within the interval $-0.1 < x < 0.2$. The "2" indicates that an arclength spline parameter is to be used. The **PANE** command will always use the arclength spline.

Some archived airfoils, notably the Eppler airfoils and some of the Selig airfoils have an excessively coarse point spacing around the leading edge. The spacing has apparently been tailored for a uniform-parameter spline, and often produces a badly shaped leading edge with the arclength-parameter spline used in Xfoil. The following command will insert additional points giving a much smoother shape for subsequent analysis.

```
.GDES  c>  CADD 10.0  1  -0.1 1.1
```

The 10.0 degree angle tolerance can be varied as needed (1/2 of the max angle is the default). The "1" argument (also a default) specifies a uniform-parameter spline for the interpolation since this works best for Eppler airfoils, and the default x range indicates that the entire airfoil is to be treated. The **CADD** command can be repeated to keep reducing the max panel angle, but this may or may not improve the smoothness of the resulting splined airfoil.

9.2 Modifying buffer airfoil

Once the buffer airfoil is suitably initialized, most of the **GDES** commands can then be used to modify it. The resulting new shape will usually be replotted immediately in a highlighted color. The plot can be refreshed anytime with the **PLOT** command.

Sometimes a sequence of commands is necessary to achieve the desired effect. For instance, suppose an airfoil with the current thickness envelope is to be given an entirely new camber line. Issuing **TSET** and inputting a new thickness of 999 (keep same thickness) and a new camber of 0 will result in the current thickness envelope unchanged and the current camber eliminated, so that a symmetrical airfoil remains. The new camber line can then be added in the **CAMB** sub-menu:

```
<cr>    Return to GDES
INPC    Input  added camber  x/c,y/c from keyboard
MODC    Modify added camber  x/c,y/c with cursor
INPP    Input  added loading x/c,DCp from keyboard
MODP    Modify added loading x/c,DCp with cursor
SLOP    Toggle modified-camber,dCp slope matching flag
SCAL r  Scale the added camber
CLR      Clear the added camber
ADD      Add the camber to the existing camberline

BLOW     Blowup plot region
RESE     Reset plot scale and origin
CPLI rr Change DCp axis plot limits
```

```
Added camber line incidence at LE =   0.00  deg.
Added camber line incidence at TE =   0.00  deg.
Max added camber y/c =   0.0000  at x/c =   0.000
```

```
..CAMB    c>
```

INPC takes the new camber line as a sequence of $x/c, y/c$ coordinate pairs which are splined. INPP takes a sequence of $x/c, \text{delta}(C_p)$ pairs instead. This $\text{delta}(C_p)$ (i.e. loading) distribution, defined as

$$\text{delta}(C_p) = (C_p)_{\text{lower}} - (C_p)_{\text{upper}}$$

is then used in Glauert's thin-airfoil relations to define the $x/c, y/c$ camber line.

With INPC and INPP, a slope discontinuity in $y(x)$ or $C_p(x)$ can be specified with two identical consecutive x/c values, which prevents splining across this point. INPP can thus easily generate a camber line with a piecewise-linear $\text{delta}(C_p)$ loading distribution, as for example an $a = 0.8$ NACA 6-digit airfoil:

```
x/c delta(Cp)
0.0    0.5
0.8    0.5
0.8    0.5
1.0    0.0
```

This results in a constant $\text{delta}(C_p) = 0.5$ for $0.0 < x/c < 0.8$, then decreasing linearly to $\text{delta}(C_p) = 0.0$ for $0.8 < x/c < 1.0$

Once a suitable added camber is input, it is added to the existing buffer airfoil camber via the **ADD** command.

The various GDES commands for modifying camber-line, thickness, leading edge radius, etc. should suffice for most geometry modification tasks. If truly frustrated, the user can "draw" the new contour with the **MODI** command, which accepts cursor inputs in the same manner as the **MDES** and **QDES** procedures. Slope matching at the modified-piece endpoints can likewise be enabled/disabled with the **SLOP** toggle command. The only important difference is that here the points must be entered in consecutive order along the new contour. One can "erase" a just-entered point by typing **e** in the graphics window.

A point can be doubled with the **CORN** command. A doubled point is useful wherever a sharp corner is required, such as at a flap break. Normally, the spline routine enforces slope continuity at all points, effectively preventing sharp corners. A doubled point, marked by a small diamond symbol on the plot, causes separate splines to be generated on each side of the corner, thus allowing the slope break. The doubled point is eliminated by clicking on it after issuing the **DELP** command. Using **DELP** on a normal single point will delete that point entirely.

The **TGAP** command sets the thickness (or gap) of the blunt trailing edge. The gap Δs is defined as the distance between the upper and lower coordinate endpoints: $\Delta s^2 = \Delta x^2 + \Delta y^2$. If the gap is already nonzero, then the new TE base vector $(\Delta x, \Delta y)$ will have the same orientation as the old one, i.e.

$$\left. \frac{\Delta y}{\Delta x} \right|_{\text{new}} = \left. \frac{\Delta y}{\Delta x} \right|_{\text{old}}$$

If the gap is zero to begin with, then the new base vector will be perpendicular to the trailing edge bisector. If the base orientation comes out in an unexpected way, it probably means that the gap was not *exactly* zero. The fix is to first set the gap to zero, and then set it again to the desired value.

After a new gap size is input, a "blending distance/ c " will also be requested. This controls how rapidly the new TE blends into the original airfoil, and is essentially the length scale for the blending function, which is exponential in x/c . The limiting values are:

- distance/ $c = 0$: Only the upper- and lower-surface TE points are changed
- $= 1$: A linear "wedge" is added or subtracted from the airfoil

9.3 Saving buffer airfoil into current airfoil

Once the desired buffer airfoil is created, a new current airfoil is set directly from the buffer airfoil with the **EXEC** command (equivalent to **PCOP** at top level). Alternatively, the new current airfoil can be re-paneled from the buffer airfoil with the **PANE** command at top level. The new current airfoil can then be analyzed in **OPER**. If the buffer airfoil has any doubled corner points, the doubled points will be eliminated, but a current-airfoil node will fall exactly on each buffer-airfoil corner.

10 Start-up Defaults

XFOIL has hardwired parameters (in SUBROUTINE INIT) controlling the paneling, plotting, and viscous execution. Most of these can be changed at runtime in the various menus. To avoid the need to change the parameters everytime XFOIL is executed, they can be saved to the default file **xfoil.def** with the **WDEF** command at TOP LEVEL. This file has the format:

140	1.0000	0.1500	0.2000	Npan	PPanel	TErat	REFrat
1.0000	1.0000	1.0000	1.0000	XrefS1	XrefS2	XrefP1	XrefP2
10.0000	0.5500	0.0150	0.8500	Size	plotAR	CHsize	ScrnFr
11.0000	8.5000	0.0000	0.0000	Xpage	Ypage	Xmargn	Ymargn
F	T			Lcolor	Lcursor		
1.0000	-2.0000	-0.5000		CPmax	CPmin	CPdel	
0.0900	0.7000			XoffAir	ScalAir	BLUwt	
0.0000	1.5000	0.5000		CLmin	CLmax	CLdel	
0.0000	0.0200	0.0100		CDmin	CDmax	CDdel	
-4.0000	10.0000	2.0000		ALmin	ALmax	ALdel	
0.0000	0.3000	0.1000		CMmin	CMmax	CMdel	
1	0.0000	0.0100		MAtype	Mach	Vaccel	
1	0.0000	9.0000		REtype	Re/10 ⁶	Ncrit	
1.0000	1.0000			XtripT	XtripB		

Line 1: Paneling parameters from the PPAR menu
 Line 2: Paneling refinement locations
 Line 3: Specifies the absolute plot size, the plot aspect ratio,
 and scales the character, number, and symbol heights
 Line 4: Defines page size and page margins, in inches
 Line 5: Flags for color PostScript output, cursor screen input
 Line 6: Defines the Cp axis annotations
 Line 7: x-offset and scale for airfoil on Cp vs x plot, BL u profile scale
 Line 8: CL axis annotations on polar plot
 Line 9: CD axis annotations on polar plot
 Line 10: alpha axis annotations on polar plot
 Line 11: CM axis annotations on polar plot
 Line 12: Mach-CL dependence type, Mach number, solution acceleration parameter
 Line 13: Re-CL dependence type, Reynolds number, transition parameter
 Line 14: Forced transition x/c locations on top,bottom sides

This file will be read at any time with the `RDEF` command, thus avoiding the manual entry of all the information.

11 Caveats

The XFOIL code is not foolproof, and requires some level of aerodynamic expertise and common sense on the part of the user. Although the inviscid analysis (OPERi), geometry design (GDES), and Full-Inverse (MDES) routines are nearly invulnerable to failure, the Mixed-Inverse (QDES) design routines and especially the viscous analysis (OPERv) routines will fail if a "reasonable" problem is not specified. Typical failure scenarios are:

- Viscous Analysis (OPERv)
 - Massive separation from excessive airfoil thickness, flap deflection, or angle of attack
 - Inherently unsteady flow (von Karman vortex street, etc.)
 - Poor resolution of leading edge pressure spike
 - Poor resolution of small viscous features (e.g. separation bubbles)
 - Reynolds number too low
- Mixed-Inverse Surface Speed Design (QDES)
 - Re-entrant airfoil shape (negative thickness)

A possible consequence of these occurrences is an arithmetic fault causing program failure. This is unlikely, but it does happen occasionally. It is therefore a good idea to save any previous work before an ambitious calculation is attempted.

The following situations may give problems strictly due to numerical roundoff:

- Excessively small panel(s) somewhere on the airfoil
- Airfoil located too far from origin
- Airfoil too thin

These situations will rarely result in an arithmetic failure, but will typically result in a "ragged" C_p distribution. Examine the paneling in the GDES menu, making the GSET command if necessary to set the current paneling. Eliminate excessively small panels by deleting one or more panel nodes with the DELP command.

When performing viscous analysis calculations, it is always a good idea to sequence runs so that alpha does not change too drastically from one case to another. The Newton solution method always uses the last available solution as a starting guess for a new solution, and works best if the change from the old to the new solutions is reasonably small. For this reason, it is best to perform difficult calculations (such as past C_{Lmax}) by gradually increasing alpha. The ASEQ command in OPER is convenient for this. If the user insists on a large change from one point to another, it is best to force a re-initialization of the boundary layers with the INIT command from the VPAR menu in OPER before the radical calculation is performed. INIT should always be executed whenever the viscous solution blows up but the program doesn't crash.

The viscous analysis will execute no more Newton iterations than set by the current iteration limit each time an ALFA, CL, etc. command is issued. If convergence is not achieved within this limit, ALFA or CL can be issued as often as needed (most easily with "!"), with another set of Newton iterations being performed each time. This iteration limit can be changed from its default value of 10 with the ITER command in OPER.

One should always be wary of trusting solutions which show regions of supersonic flow. Such flows can be reliably predicted only with a truly nonlinear field method (such as the MSES code). As a rule of thumb, if the maximum Mach number doesn't exceed 1.05 anywhere, shock losses will be very small, the C_p distributions will be reasonably accurate, and the drag predicted by XFOIL is likely to be accurate.