

1 Overview

PROPID is a computer program for the design and analysis of horizontal axis wind turbines [1]. The unique strength of the current design method is its inverse design capability. For instance, the current method allows the designer to specify directly the peak power for a stall-regulated rotor. The iterative inverse solver is then used to adjust one of the user selected inputs so that the desired peak rotor power is achieved. More generally, the method permits the designer to specify several performance characteristics as long as an equal number of input parameters are allowed to be automatically adjusted by the method. The approach is based on similar design methodology for airfoils and cascades [2, 3].

PROPID not only allows for the specification of single variables, like peak power, but also distributions, like the blade lift distribution and/or axial interference factor distribution. Such distributions are achievable as long as another distribution is left to be determined, that being specifically the blade twist and/or chord distributions.

The method also has multipoint design capabilities. For example, the blade lift coefficient distribution can be prescribed for one condition while simultaneously the axial induction factor distribution can be prescribed for a different condition. In addition, the designer can simultaneously specify the peak rotor power constraint, which may correspond to yet another condition.

The widely used **PROPSH** blade-element/momentum code [4], which itself is a modified version of the **PROP** code [5], is used for analysis in the current version of **PROPID**. Desirable features of the code are that it allows for rapid analysis, accommodates different airfoil data for each blade element, and includes a 3-D post-stall airfoil performance synthesization method for better peak power prediction at high wind speed.

In what follows, Chapter 2 discusses typographical conventions used in this manual and also the input file structure. The input file is read automatically upon program execution and is assigned in the file `RUNFILES.TXT`, which is included on the distribution disk for PC. Chapter 3 presents several parameters required for a **PROPID** design or analysis run. Input for the wind turbine operating conditions is presented in Chapter 4. Once the operating conditions are specified as well as the various input parameters (some of which may change through iteration), an analysis of the rotor can be performed as discussed in Chapter 5. An example input file corresponding to the AeroStar 7.4 m blade for the Micon 65 is presented in Chapter 6. This file is named `AEROSTAR.IPT` on the distribution disk. Chapter 7 discusses how output files (e.g., power curve, blade chord distribution, etc.) are generated. Output files generated by the `AEROSTAR.IPT` file are included in the `DATFILES` directory on the distribution disk. The design mode is discussed in Chapter 8 and many additional input lines are briefly discussed in Chapter 9. In Chapter 10 an annotated input file is presented. Appendix A includes airfoil data for the AeroStar 7.4 m blade, and Appendix B discusses airfoil data input options consistent with the **PROP** code. A summary of commands is included in Appendix C for a quick reference.

2 Reading This User's Guide

PROPID is a keyword-based code. The input file (see Chapter 6) 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.

Examples:

```
MODE 1
!CLEAN
*
```

The line `MODE 1` 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 82 will not be read. The `!` character in the first column denotes that the remainder of the line is to be ignored. Most data lines can have trailing comments.

Example:

```
LTIP 1          ! use Prandtl tip loss model
```

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 following descriptions of the lines, the format

```
MODE <MODE>
```

will be used. The string `<MODE>` denotes that the value following the line will be set to the program variable `MODE`. For instance, a `1` for `<MODE>` indicates that `MODE = 1`. In some cases, all of the values on a line will not be used in which case `999` will be used as a dummy value. 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 `|` need not be entered if the defaults are acceptable. The actual input file does not contain the `|` character. As previously mentioned, lines with optional data cannot have trailing comments. Arrays and elements of arrays are denoted, for example, by `CH(.)` and `CH(1)` or `CH(JSEG)`, respectively. `JSEG` is used to denote the j th element of the blade. `ISEG` is the total number of blade elements. Finally, a `-` character in this user's guide 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 3) must precede lines that initiate design and analysis. These lines can be in any order. Second, some lines initiate the input of a sequence of data (e.g., the `AIRFOIL_MODE` line, Section 3.3). Comments are sometimes used in this manual to indicate the beginning and ending of a sequence of data.

3 Required Data for Design and Analysis Modes

The following lines are used to input data that is required for design and analysis. Only brief descriptions are given since further details can be found in Ref. [6].

3.1 Miscellaneous Parameters

The following lines are used to set single parameters and options.

```
MODE <MODE>
INCV <INCV>
LTIP <LTIP>
LHUB <LHUB>
IBR <IBR>
SH <SH>
ISTL <ISTL>
USEAP <USEAP>
WEXP <WEXP>
RHO <RHO>
RD <RD>
HUB <HUB>
HH <HH>
CONE <CONE>
BN <BN>
NS_NSEC <NS> <NSEC>
IS1 <IS1>
IS2 <IS2>
```

where

```
MODE = 1 → wind turbine mode
       2 → propeller mode (not implemented in the current version)
INCV = 0 → wind turbine model
       1 → propeller mode (not implemented in the current version)
LTIP = 0 → ignore tip loss effects
       1 → use Prandtl tip loss model
LHUB = 0 → ignore hub loss effects
       1 → use Prandtl hub loss model
IBR = 0 → use classical model
       1 → use advanced model
SH = 0 → ignore shaft tilt effects
     1 → include shaft tilt effects
ISTL = 0 → use flat plate post-stall model
       1 → use Viterna post-stall model
USEAP = 0 → ignore swirl effects
```

1 → include swirl effects
 WEXP = wind boundary layer profile exponent
 RHO = air density (lb sec²/ft⁴)
 RD = rotor radius (ft)
 HUB = normalized hub radius (i.e., the ratio of hub radius to rotor radius)
 HH = normalized hub height (i.e., the ratio of hub height to rotor radius)
 CONE = cone angle of the rotor (deg)
 BN = number of blades
 NS = number of blade segments (also ISEG)
 NSEC = number of circumferential segments (must be ≥ 4 when WEXP ≠ 0)
 IS1 = number of the first segment to be used in the analysis (usually 1)
 IS2 = number of the last segment to be used in the analysis (usually ISEG)

In the design mode, it is required that WEXP = 0 and NSEC = 1.

3.2 Blade Geometry

The CH_TW line is used to enter the chord and twist distributions from root to tip.

```

! begin chord and twist data
CH_TW
<CH(1)>    <TW(1)>          ! root chord and twist
...
<CH(JSEG)> <TW(JSEG)>
...
<CH(ISEG)> <TW(ISEG)>      ! tip chord and twist
! end
  
```

where CH(.) is the normalized blade chord (i.e., the ratio of the blade chord to the rotor radius), and TW(.) is the blade twist angle (deg).

3.3 Blade Airfoil Data

The AIRFOIL_MODE line initiates the input of the airfoil α - C_l and α - C_d data for each blade segment.

```

! begin airfoil data
AIRFOIL_MODE <1>
<1>    <MM(1)>    <NN(1)>          ! first segment
<AL(1)>    <CL(1)>
...
<AL(MM(1))> <CL(MM(1))>
<AD(1)>    <CD(1)>
...
  
```

```

<AD(MM(1))>    <CD(MM(1))>
...
<JSEG> <MM(JSEG)> <NN(JSEG)>    ! intermediate segment
<AL(JSEG)>    <CL(JSEG)>
...
<AL(MM(JSEG))> <CL(MM(JSEG))>
<AD(JSEG)>    <CD(JSEG)>
...
<AD(MM(JSEG))> <CD(MM(JSEG))>
...
<ISEG> <MM(ISEG)> <NN(ISEG)>    ! last segment
<AL(ISEG)>    <CL(ISEG)>
...
<AL(MM(ISEG))> <CL(MM(ISEG))>
<AD(ISEG)>    <CD(ISEG)>
...
<AD(MM(ISEG))> <CD(MM(ISEG))>
! end

```

AL(.) and AD(.) are the angles of attack (deg) corresponding to the lift and drag coefficients CL(.) and CD(.), respectively. MM(JSEG) and NN(JSEG) are the number of α - C_l and α - C_d pairs to follow for segment JSEG. No more than 20 α - C_l and α - C_d pairs can be used for a given segment. See Appendix B for options if two or more blade elements have the same data.

4 Design Points for Design and Analysis Modes

For both design and analysis modes, the DP line can be used to enter the conditions for which the wind turbine is to be analyzed. These conditions are entered with the line:

```
DP <IDP> <RPMDP(IDP)> <FLDP(IDP)> <XJDP(IDP)> <IXDIMDP(IDP)>
```

where

IDP = design point number
RPMDP(IDP) = rotor speed (rpm)
FLDP(IDP) = blade pitch (deg) at 75% of radius
XJDP(IDP) = speed as indicated by the units parameter IXDIMDP(IDP)
IXDIMDP(IDP) = 0 → XJDP(IDP) is wind speed (ft/sec)
1 → XJDP(IDP) is wind speed (m/sec)
2 → XJDP(IDP) is wind speed (mph)
3 → XJDP(IDP) is tip speed ratio

In the input file, these conditions are later referred to by the design point number. An alternative to entering the rotor speed, blade pitch, and wind speed for use in analysis is discussed in Chapter 5.

5 Analysis Mode

The analysis mode is used to determine the rotor performance via the 2D_SWEEP line and blade aerodynamic characteristics via 1D_SWEEP line, both of which must be preceded by lines that set the conditions. The data generated by the 2D_SWEEP and 1D_SWEEP lines is stored in memory until the WRITE_FILES line is issued to write out the data to an ascii file (see Chapter 7).

The 2D_SWEEP line is used to determine the rotor performance characteristics, such as, the power curve vs. wind speed, power coefficient vs. tip speed ratio, etc. Moreover, a family of curves can be generated for different values of pitch, rotor speed, or tip speed ratio. The blade pitch is set by using one of the following lines:

```
PITCH_FIXED <FL>                ! (deg)
PITCH_DP    <JDPFL>
PITCH_SWEEP <FLS> <FLF> <DFL>    ! (deg)
```

The PITCH_FIXED line sets the pitch to the value of FL, or the PITCH_DP line sets the pitch to the value given by one of the DP lines, in particular, the design point assigned to number JDPFL. Instead of using a single pitch, the pitch can be swept over a range to generate a family of curves through the PITCH_SWEEP line. FLS is the initial (start) value for the pitch, FLF is the final value, and DFL is the increment. Likewise, the rotor speed is set by one of the lines:

```
RPM_FIXED <RPM>                  ! (rpm)
RPM_DP    <JDPRPM>
RPM_SWEEP <RPMS> <RPMF> <DRPM>  ! (rpm)
```

If the pitch is swept over a range by the line PITCH_SWEEP then the rotor speed cannot be swept, and visa versa. That is, either PITCH_SWEEP or RPM_SWEEP can be used, but not both. Finally, the wind is set, and the analysis is performed by the lines:

```
WIND_SWEEP <XJS> <XJF> <DXJ> <IXDIM>
2D_SWEEP
```

where XJS, XJF, and DXJ are the initial, final, and incremental values for the wind speed according to the parameter IXDIM defined in Chapter 4.

Examples:

```
DP    1    65.0000    1.219 999.000    2
RPM_DP 1
PITCH_DP 1
WIND_SWEEP 7    50.0    1 2.0
2D_SWEEP
```

In this case, the rotor speed and pitch are both set by the values given in the first DP line. The value of 999.0000 in the DP line is not used and consequently could be anything. The wind is swept over the range from 7 to 50 mph in 1 mph increments.

```
DP 1 65.0000 1.219 999.000 2
RPM_DP 1
PITCH_SWEEP -2 4 1
WIND_SWEEP 7 50.0 1 2.0
2D_SWEEP
```

In addition to a sweep over the wind speed, the pitch is now swept over the range from -2 to 4 deg in 1 deg increments. The pitch value in the DP line is not used; however, the rotor speed from the DP line is used.

The 1D_SWEEP line is used to generate data along the blade span, such as the blade lift coefficient distribution. Preceding the 1D_SWEEP line must be lines to set the pitch, rotor speed, and wind speed, to be respectively selected from among the lines:

```
PITCH_FIXED <FL> ! (deg)
PITCH_DP <JDPFL>
PITCH_SWEEP <FLS> <FLF> <DFL> ! (deg)

RPM_FIXED <RPM> ! (rpm)
RPM_DP <JDPRPM>
RPM_SWEEP <RPMS> <RPMF> <DRPM> ! (rpm)

WIND_FIXED <XJ> <IXDIM>
WIND_DP <JDPWND>
WIND_SWEEP <XJS> <XJF> <DXJ> <IXDIM>
```

From each set, only one line is used to set the pitch, rotor speed and wind speed. At most only one sweep line can be used. Following these three lines, the 1D_SWEEP line generates the results that can be written out as discussed in Chapter 7.

Example:

```
DP 1 65.0000 1.219 999.000 2
DP 2 999.0000 999.000 19.160 2
DP 3 999.0000 999.000 15.000 2
RPM_DP 1
PITCH_DP 1
WIND_DP 3
1D_SWEEP
```

This sequence sets the rotor speed to 65 rpm, the pitch to 1.219 deg, and the wind speed to 15 mph for analysis. The second DP line is not used.

6 Input File

The input file is assigned in the RUNFILES.TXT which contains the following line

```
&ALFILES INFILE='AEROSTAR.IPT' &END
```

where in this case AEROSTAR.IPT is the input file. This input file is included on the **PROPID** distribution disk as file AEROSTAR.IPT. A typical input file is shown below. Only part the the airfoil data specified by the AIRFOIL_MODE line is shown below; Appendix A gives the complete airfoil input data for reference.

```
! File: AEROSTAR.IPT
! AeroStar 7.4 m blade used on the Micon 65 with
! NACA 44xx airfoils.

MODE 1.0           ! wind turbine
INCV 0.0           ! wind turbine
LTIP 1.0           ! tip loss model
LHUB 1.0           ! tip loss model
IBR 1.0            ! brake state model
ISTL 1.0           ! viterna stall model
USEAP 1.0          ! swiral
WEXP 0.0000        ! no ground boundary layer
HH 2.2400          ! tower height
NS_NSEC 10.0  1.0 ! number of blade elements/number of sectors
IS1 1.0            ! first segment used in analysis
IS2 10.0           ! last segment used in analysis
SH 0.0             ! shaft tilt effects on crossflow
BN 3.0             ! blade number
CONE 4.0           ! cone angle of rotor
RHO 0.002378       ! air density (slugs/ft^3)
HUB 0.15           ! nondimensional hub radius
RD 26.25           ! radius
CH_TW              ! chord and twist distributions
.1457 8.40
.1457 8.40
.1356 7.40
.1254 6.40
.1156 5.40
.1054 4.40
.0956 3.50
.0854 2.50
.0752 1.50
.0654 0.50
AIRFOIL_MODE 1     ! airfoil data
1.0 10.0 13.0
```

```

-4.0      0.039
-2.0      0.248
 0.0      0.459
 2.0      0.664
 4.0      0.865
 6.0      1.053
...
DP 1 50.3 -0.754733 999 2 ! design point

RPM_DP 1 ! use rpm from design point 1
PITCH_DP 1 ! use pitch from design point 1
WIND_SWEEP 7 50.0 1 2.0 ! wind speed range
2D_SWEEP ! perform analysis
WRITE_FILES 40 45 50 55 ! write output files

RPM_DP 1 ! use rpm from design point 1
PITCH_DP 1 ! use pitch from design point 1
WIND_FIXED 16 2 ! wind speed
1D_SWEEP ! perform analysis
WRITE_FILES 75 80 85 90 ! write output files

WRITE_FILES 95 100 ! write output files
*
```

The WRITE_FILES line will be discussed in Chapter 7.

7 Output Files

Results generated by the 2D_SWEEP and 1D_SWEEP lines can be written to ascii files by the line:

```
WRITE_FILES <IPRT(1)> <IPRT(2)> <IPRT(3)> ... | ... <IPRT(20)>
```

Example:

```
DP 1 65.0000 1.219 999.000 2
RPM_DP 1
PITCH_SWEEP -2 4 1
WIND_SWEEP 7 50.0 1 2.0
2D_SWEEP
WRITE_FILES 40 45
```

The lines above analyze a rotor, and then the WRITE_FILES line writes to ascii files the power vs. wind speed (FORT040.DAT) and power coefficient vs. TSR (FORT045.DAT) generated by the preceding 2D_SWEEP line.

Files generated by the 2D_SWEEP line that can be subsequently written out are listed below.

```
IPRT(.) Data written out to logical unit IPRT(.)
40      rotor power vs. wind speed
45      rotor power coefficient vs. TSR
50      rotor power coefficient vs. wind speed
```

Files generated by the 1D_SWEEP line are:

```
IPRT(.) Data written out to logical unit IPRT(.)
60      blade power distribution
65      blade power coefficient distribution
70      blade local windfarm figure of merit distribution
75      blade airfoil lift-to-drag ratio distribution
80      blade angle of attack distribution
85      blade lift coefficient distribution
90      blade axial induction factor distribution
```

At anytime following the CH_TW line, the blade chord and twist distributions can be written by using 95 and 100, respectively.

8 Design Mode

The design mode can be used to specify a desired output that is achieved by automatically adjusting one of the inputs. If a single value is specified (e.g., peak rotor power), then a NEWT1-type line is used. If a function is prescribed (e.g., lift coefficient distribution), then instead a NEWT2-type line is used. Any number of NEWT1- and NEWT2-type lines can be used as long as no two lines specify the same desired output or input for adjustment. If no solution is found, it usually indicates that either (1) the above rule is violated or (2) the desired output is not physically possible.

8.1 NEWT1 Lines

The NEWT1ISWP can be used to specify a desired peak power (or some other variable that is determined by analyzing the rotor over a given wind speed range). The general form of the NEWT1ISWP line is given by

```
NEWT1ISWP <IFTP1(.)> <FNEWT1(.)> <XJSNT1(.)> <XJFNT1(.)> <DXJNT1(.)> -  
          <KDPRPM1(.)> <KDPFL1(.)> <KDPXJ1(.)> -  
          <ITP1(.)> <ITP2(.)> <ITP3(.)> | <CLAMP1(.)> <TOL1(.)>
```

where

IFTP1(.) = 300 → peak rotor power (kW)
 301 → wind speed (mph) at peak power
 302 → peak power coefficient
FNEWT1(.) = value for specified parameter
XJSNT1(.) = lowest value for wind speed (mph) for range
XJFNT1(.) = highest value for wind speed (mph) for range
DXJNT1(.) = increment in wind speed (mph)
KDPRPM1(.) = design point # for rotor speed
KDPFL1(.) = design point # for blade pitch
KDPXJ1(.) = 999 (value is ignored)
ITP1(.) = used to identify input variable for iteration (see Table 1)
ITP2(.) = used to identify input variable for iteration (see Table 1)
ITP3(.) = used to identify input variable for iteration (see Table 1)
CLAMP1(.) = positive step limit used during iteration (optional)
TOL1(.) (optional if CLAMP1(.) is specified)
 > 0 and specified → convergence tolerance for auto iteration mode
 unspecified → interactive iteration mode in force

Input parameters for iteration are specified according to the table:

ITP1(.)	ITP2(.)	ITP3(.)
1	1 → Scale Rotor 2 → Rotor Speed 3 → Pitch 4 → Wind Speed 5 → Cone Angle 6 → Air Density 7 → Rotor Radius	999 # → Design Point # # → Design Point # # → Design Point # 999 999 999
2	# → Blade Chord # 999	999 100 → Offset Chord
3	# → Blade Twist #	999

The # in the above table is used to indicate which segment of the blade is used for iteration. “Offset Chord” means that the blade chord at each segment is increased or decreased by an equal amount, which effectively changes the rotor solidity.

The `CLAMP1(.)` sets the step limit for each input variable used for the iteration. Sometimes the predicted change in the input variable is too large and can cause the solution to diverge. In this situation, specifying a step limit can usually improve convergence. If no step limit is desired, then the value for `CLAMP1(.)` can be left unspecified in the `NEWT1` line. In this case, the value `TOL1(.)` must be unspecified as well.

The parameter `TOL1(.)` is used in the convergence test and is the desired difference between the current value and the specified value for the output parameter. If iteration is to proceed automatically until convergence without user input, then `TOL1(.)` must be specified for use in the convergence test. In this case, all `NEWT1` and `NEWT2` lines must contain values for `TOL1(.)` and as later discussed `TOL2(.)` if a `NEWT2` line is used. If `TOL1(.)` is not specified, then the iteration steps are performed interactively. In most cases, the interactive iteration mode is most useful since the design of the rotor can be monitored for potential convergence difficulties.

After any number of `NEWT1` and `NEWT2` lines, the iteration is initiated by the `IDES` line. Anytime the `IDES` line is issued, the iteration scheme will attempt to achieve the desired specifications for all the `NEWT1` and `NEWT2` lines that preceded the current `IDES` line. Once convergence is achieved, more `NEWT1` and `NEWT2` lines can be used, and the `IDES` can be issued again to converge the solution to all the preceding specifications.

Example:

```
DP 1 50.3 2.5 999 2
NEWT1ISWP 300 70 8 40 1. 1 1 1 2 999 100 .05 .1
IDES
```

For this example, the peak rotor power is specified to be 70 kW over the wind speed range from 8 mph to 40 mph in increments of 1 mph. The corresponding rotor speed and pitch are 50.3 rpm and 2.5 deg, respectively. To achieve the desired peak power,

the chord is offset. A clamp is used on the chord offset, and iteration will be performed automatically until the actual peak power is within 0.1 kW of the desired peak power of 70 kW. For the AeroStar 7.4 m blade given in Section 6 and Appendix A, the following output is echoed to the screen for the example lines given above.

NEWT1-line

```
1: Prescribed peak power (kW) = 70. at design point rpm( 1) pitch( 1)
   Adjust chord uniformly with step limit = 0.05
```

```
Initial wind turbine design for stage: 1
```

```
Residues for NEWT1* equations:
```

```
FNT1_0( 1) = 19.76599
```

```
Iteration 1
```

```
Calculating sensitivities for NEWT1 design parameter: 1
```

```
Residues for NEWT1* equations:
```

```
FNT1_1( 1) = 1.17902 DELTAS1() = -0.01705 CLAMP1() = 0.050
```

```
Finished iteration = 1
```

```
Iteration 2
```

```
Calculating sensitivities for NEWT1 design parameter: 1
```

```
Residues for NEWT1* equations:
```

```
FNT1_1( 1) = 0.00012 DELTAS1() = -0.00105 CLAMP1() = 0.050
```

```
Finished iteration = 2
```

```
Converged solution
```

As the output indicates, when a NEWT1 (or NEWT2) line is given, the specifications are echoed to the screen for verification. The IDES line initiates the iteration, which in this case will be performed automatically since TOL1(.) is specified. Stage 1 indicates that this is the first time an iteration sequence has been initiated. If a second IDES line were to follow later, then that iteration sequence would be referred to as stage 2, and so on. After the IDES line, **PROPID** determines the peak power (maximum power) over the specified wind speed range and determines the difference between the actual peak power and the specified peak power—the *residue*, which in this case is nearly 20 kW. In other words, the actual peak power is nearly 90 kW rather than 70 kW. Since the difference is not less than 0.1 kW, iteration begins. The sensitivity of the peak power to the blade chord offset is determined, and a step of -0.01705 is computed, that is, the chord is everywhere reduced to this amount. The clamp is not enforced since the magnitude of the step size is smaller than the clamp of 0.05. Once the blade chord is offset by -0.01705 , the difference between the actual and

desired peak power is only 1.17902. The iteration is performed again and convergence is achieved.

Example:

```
DP 1 50.3 2.5 999 2
NEWT1ISWP 300 70 8 40 1. 1 1 1 2 999 100 .015
IDES
```

This example, which is similar to the last, illustrates the interactive iteration mode (TOL1(.) is left unspecified). The interactive iteration mode is most useful when starting a new design problem, since convergence can sometimes be difficult in which case clues can be gleaned from the convergence history. In this example, a rather small clamp is used to illustrate what happens when a step size is greater than the respective clamp. The following output is echoed to the screen.

NEWT1-line

```
1: Prescribed peak power (kW) = 70. at design point rpm( 1) pitch( 1)
Adjust chord uniformly with step limit = 0.02
```

Initial wind turbine design for stage: 1

Residues for NEWT1* equations:

```
FNT1_0( 1) = 19.76599
```

Select option:

```
0 Stop iteration for current stage
# Number of consecutive iterations
999 To stop execution
99 More options
```

>>3

In consecutive iteration mode...

Iteration 1

```
Calculating sensitivities for NEWT1 design parameter: 1
```

Residues for NEWT1* equations:

```
FNT1_1( 1) = 3.49121 DELTAS1() = -0.01500 CLAMP1() = 0.015
```

Iteration 2

```
Calculating sensitivities for NEWT1 design parameter: 1
```

Residues for NEWT1* equations:

```
FNT1_1( 1) =    0.01712  DELTAS1() = -0.00309  CLAMP1() = 0.015
```

```
Iteration    3
```

```
Calculating sensitivities for NEWT1 design parameter:    1
```

```
Residues for NEWT1* equations:
```

```
FNT1_1( 1) =    0.00003  DELTAS1() = -0.00002  CLAMP1() = 0.015
```

```
Select option:
```

```
0  Stop iteration for current stage
#  Number of consecutive iterations
999 To stop execution
99  More options
```

```
>>1
```

```
Iteration    4
```

```
Calculating sensitivities for NEWT1 design parameter:    1
```

```
Residues for NEWT1* equations:
```

```
FNT1_1( 1) =    0.00000  DELTAS1() =    0.00000  CLAMP1() = 0.015
```

```
Select option:
```

```
0  Stop iteration for current stage
#  Number of consecutive iterations
999 To stop execution
99  More options
```

```
>>0
```

In this case, from the options table a 3 is entered so that 3 consecutive iterations are performed. The >> notation is used to denote user input. For the first iteration, the clamp is reached, and the magnitude of the step is reduced to the clamp size. After the first three iterations, one more iteration is then performed to reduce the residue nearly to machine zero. At anytime, 999 could have been entered to stop the program. 99 is used to enter another options menu to interactively double or halve the clamp sizes in case convergence is slow (→ increase the clamp sizes) or in case the solution begins to diverge (→ decrease the clamp sizes).

Once a blade has converged, the converged input data can be written out to file FORT021.DAT with the DUMP_PROPID line given by

```
DUMP_PROPID
```

The contents of this new output file can then be substituted back into the original input file as the design process continues. As an alternative, a PROP93/PROPSH data file can be written out to file FORT022.DAT by using

DUMP_PROP93 <JDP>

where JDP is the design point.

Additional NEWT1 lines are listed as follows

```
NEWT1IDP <IFTP1(.)> <FNEWT1(.)> -  
          <KDPRPM1(.)> <KDPFL1(.)> <KDPXJ1(.)> -  
          <ITP1(.)> <ITP2(.)> <ITP3(.)> | <CLAMP1(.)> <TOL1(.)>
```

where

IFTP1(.) = 200 → power (kW)
FNEWT1(.) = value for specified parameter
KDPRPM1(.) = design point # for rotor speed
KDPFL1(.) = design point # for blade pitch
KDPXJ1(.) = design point # for wind speed

The parameters ITP1(.), etc. are the same as previously described.

Local blade characteristics such as the lift coefficient can be prescribed by the line

```
NEWT1LDP <IFTP1(.)> <JSEGIX1(.)> <FNEWT1(.)> -  
          <KDPRPM1(.)> <KDPFL1(.)> <KDPXJ1(.)> -  
          <ITP1(.)> <ITP2(.)> <ITP3(.)> | <CLAMP1(.)> <TOL1(.)>
```

where

IFTP1(.) = 500 → local lift coefficient
 501 → local axial induction factor
 502 → local airfoil angle of attack
 504 → local power coefficient
 505 → local power
JSEGIX1(.) = blade segment for specified parameter
FNEWT1(.) = value for specified parameter
KDPRPM1(.) = design point # for rotor speed
KDPFL1(.) = design point # for blade pitch
KDPXJ1(.) = design point # for wind speed

Again, the parameters ITP1(.), etc. are the same as previously described.

8.2 NEWT2 Lines

NEWT2 lines are used to specify, for example, the lift coefficient distribution *relative to a specified location*. Such a distribution is referred to as a *relative lift coefficient distribution* or \tilde{C}_l . This new notation is best introduced through Figure 1, which

Figure 1: Parameterization of blade chord and twist distributions.

shows a convenient parameterization of the blade geometry. The chord c is composed of the sum of a constant level at the blade root and a chord distribution relative to this constant level, that is, $c_0 + \tilde{c}$. Likewise, the blade pitch is the sum of the pitch at 75% radius ($\beta_{75\%}$) and the twist relative to this point (θ), both measured positive in the direction toward feather from the rotor plane. In general, the chord, twist, or lift coefficient distributions can be referenced relative to any location as specified by the NEWT2 line.

To illustrate the approach, a fairly sophisticated example is considered. A three-bladed, 9.25 m radius rotor operates at a constant rotor speed of 50 rpm with a fixed pitch of 3 deg at 75% of radius. The NREL S814 (root), S809 (primary) and S810 (tip) advanced wind turbine airfoils are used along the blade span. (Airfoil data for the S814/809/810 series can be obtained through the National Renewable Energy Laboratory, Golden, CO.) The blade is defined by 10 segments. The design goals are to achieve (1) a specified peak power of 75 kW and (2) a desired lift coefficient distribution at a wind speed yet to be determined. As previously discussed, one means of achieving the desired peak power is to adjust the solidity via the blade chord offset c_0 (as shown in Figure 2) by the lines

```

DP   1   50.0000   2.500 999.000  2
DP   2  999.0000 999.000  14.444  2
NEWT1ISWP 300 75 20 35 1.   1 1 1   2 999 100
IDES
```

Continuing with the current example, the desired lift coefficient distribution is shown in Figure 3. As with the blade twist distribution, it is convenient to divide the desired C_l distribution into two components: $C_{l,75\%} + \tilde{C}_l$. This C_l distribution

Figure 2: Adjustment of blade chord offset to achieve desired peak power.

Figure 3: Desired C_l distribution corresponding to a wind speed yet to be determined.

Figure 4: Adjustment of wind speed to achieve desired C_l at 75% radius.

corresponds roughly to the maximum L/D condition of the airfoils along the blade span. The speed corresponding to this C_l distribution is defined in part by the peak power level for the following reason. As shown in Figure 2, the peak rotor power is reached at a wind speed near 13.41 m/s (30 mph). At this speed, the rotor C_l distribution is nearly centered about the $C_{l,max}$ of the airfoils along the span. Likewise, near the cut-in speed of 4.47 m/s (10 mph), the net C_l distribution is slightly above zero lift. Thus, it would be inconsistent to specify that the desired C_l distribution should occur at a wind speed near 13.41 m/s (30 mph) or near the cut-in speed. In fact, the speed corresponding to the desired C_l cannot be specified; rather, this speed must be determined since it is predefined somewhat by the cut-in and peak power speeds.

This wind speed shown in Figure 4 (see point 2) is determined by the additional lines

```
NEWT1LDP 500 8 .65 1 1 2 1 4 2 1
IDES
```

which in the input file follows the previous lines for the peak power prescription. As indicated by this line, the C_l is specified to be 0.65 at segment 8 (75% of radius) for a rotor speed of 50 rpm and blade pitch of 2.5 deg. The wind speed for the second design point is adjusted to achieve the desired lift coefficient on segment 8. The IDES line begins the second iteration stage for both the design point wind speed as well as the blade chord offset so that the desired peak power and local lift coefficient are achieved. Thus, a two-dimensional iteration is performed.

In the next stage, the twist θ is adjusted to achieve the desired \tilde{C}_l as shown in Figure 5 (see solid line). In particular, the twist inboard of 75% radius is adjusted to achieve the desired inboard \tilde{C}_l . Also, the outboard twist is iterated to achieve the desired outboard \tilde{C}_l .

Figure 5: Adjustment of twist distribution to achieve desired \tilde{C}_l distribution.

In general, the NEWT2SDDP line is used to adjust either the relative chord or twist distribution to achieve a desired relative output distribution. The NEWT2SDDP line is given by

```

NEWT2SDDP <IFTP2(.)> <JSEGIX2(.)> <JSEGIX3(.)> <JSEGREL(.)> <KADJSBS(.)>
<SSS(1)> <SSF(1)>
...
<SSS(KADJSBS(.))> <SSF(KADJSBS(.))>
<KDPRPM2(.)> <KDPFL2(.)> <KDPXJ2(.)> <ISDTP(.)> <ISCHED2(.)> | -
          <CLAMP2(.)  TOL2(.)

```

where

IFTP1(.)	= 100 → relative lift coefficient distribution
	101 → relative axial induction factor distribution
	102 → relative airfoil angle of attack distribution
	104 → relative power coefficient distribution
	105 → relative power distribution
JSEGIX2(.)	= inboard segment for relative distribution
JSEGIX3(.)	= outboard segment for relative distribution
JSEGREL(.)	= relative segment, i.e., where the relative distribution is 0
KADJSBS(.)	= number of segments [JSEGIX3(.) - JSEGIX2(.)]
SSS(1)	= first segment prescribed (always = 1)
SSF(1)	= relative value for first segment
SSS(KADJSBS(.))	= last segment prescribed (always = JSEGREL(.))
SSF(KADJSBS(.))	= relative value for last segment
KDPRPM2(.)	= design point # for rotor speed
KDPFL2(.)	= design point # for blade pitch
KDPXJ2(.)	= design point # for wind speed
ISDTP(.)	= used to identify input variable for iteration (see Table 2)
ISCHED2(.)	= used to identify input variable for iteration (see Table 2)

CLAMP2(.) = positive step limit used during iteration (optional)
TOL2(.) (optional if CLAMP2(.) is specified)
> 0 and specified → convergence tolerance for auto iteration mode
unspecified → interactive iteration mode in force

The input parameters for iteration are specified according to the table:

Table 2: Input Variable Specification for NEWT2 Iteration	
ISDTP(.)	ISCHED2(.)
1	100 → Move Corresponding Chord Independently
2	100 → Move Corresponding Twist Independently

The blade twist adjustment shown in Figure 5 is performed according to the lines

```

NEWT2SDDP 100 2 7 8 6
1 .308
2 .302
3 .276
4 .218
5 .118
6 .042
1 1 2 2 100 2
NEWT2SDDP 100 9 10 8 2
1 -.030
2 -.053
1 1 2 2 100 2
IDES
```

The first NEWT2SDDP line prescribes the \tilde{C}_l from segments 2 through 7 relative to 8. Six values for \tilde{C}_l then follow. At segment 2 the \tilde{C}_l is prescribed to be 0.308, 0.302 for segment 3, and so on. The specification corresponds to a rotor speed of 50 rpm and pitch of 2.5 deg. The wind speed corresponds to that of the second design point (which is changing according to the previous NEWT1LDP line). To achieve the desired \tilde{C}_l , the corresponding twist is adjusted for segments 2 through 7. The second NEWT2SDDP line prescribes the \tilde{C}_l from segments 9 through 10 relative to 8. Again, the twist is adjusted for the corresponding segments (9 and 10). Note that for this example, the twist for segment 8 is left unchanged at 0 deg. Therefore, the pitch of the blade is also unchanged by the present iteration schedule. It should be noted that the combined NEWT1 and NEWT2 lines lead to a iteration on 10 variables (blade chord offset, design point wind speed, and eight θ values) for 10 desired output values (peak power, C_l at 75% radius, and 8 \tilde{C}_l values).

The current example can be extended to include iteration on the blade chord so that a desired *relative* axial induction factor distribution can be achieved. In particular, the lines

```

NEWT2SDDP 101 3 10 2 8
1 0.00
```

```

2  0.00
3  0.00
4  0.00
5  0.00
6  0.00
7  0.00
8  0.00
1  1  2  1  100  .5
IDES

```

produce a constant axial induction factor from segments 2 through 10. The axial induction factor for a specific blade element can be prescribed with the `NEWT1LDP` line. Doing so together with the relative distribution prescribed (as above) would then fix the entire axial induction factor distribution for the specified condition.

8.3 General Hints

- *Selecting suitable input variables for iteration.* When selecting input variables for iteration, convergence is most rapidly achieved when the specified output variable depends strongly on the selected input variable. For instance, peak power is a strong function of solidity, blade pitch, and rotor speed, but peak power is a weak function of cone angle for all practical purposes. Some specific suggestions are in order. If the relative lift is specified, then convergence is best achieved through iteration on the blade twist. If the axial induction factor is prescribed, convergence is best achieved through iteration on the blade chord.

- *Under-specification of variables for iteration.* Care should be taken to ensure that an input variable is not selected for iteration more than once. There are no special checks in the code should this inadvertently occur. Specifying the same variable twice for iteration is equivalent to not specifying enough variables for iteration.

- *Specified lift coefficient distribution.* If the lift coefficient is specified along the span, the lift coefficient should not (of course) exceed the local maximum lift coefficient of the airfoil (since such specification would not be achievable). Moreover, a problem can occur when the specified C_l distribution is too close to $C_{l,max}$. For instance, if $C_{l,max}$ is 1.2 and the specified local lift coefficient is 1.1, two different solutions exist—one before stall and one after. The user may assume that the specified C_l will be achieved at an angle of attack below $C_{l,max}$; however, it could also be achieved for an angle of attack above $C_{l,max}$. The wind turbine must be analyzed to determine which case exists after iteration. To avoid this potential difficulty, it is suggested that the specified C_l be at least 0.2 below $C_{l,max}$.

- *Hub radius/cut-out.* No specifications should be applied to those segments that are within the hub radius (or cut-out) or those segments that are not included in the analysis by the `IS1` and `IS2` lines.

- *Errors and Warnings.* There are several checks in the code for errors and potential errors. For instance, if the peak power is specified for a rotor and iteration

is performed on the blade chord offset, the chord can become negative at which point an error will be issued to the user. It is easy to envision a case for which this could occur. If the actually peak power is greater than the prescribed peak power, the chord will be reduced everywhere by an equal amount. The iteration process will be repeated as long as the actually peak power is greater than the prescribed peak power. At some point, the local chord could be reduce to a negative value (most likely at the tip). Since the data is always checked between iterations, an error to this effect will be subsequently issued. Warnings can be innocuous, but usually they are hints of problems either with the initial input or with the current solution.

- *Residues do not go to zero.* This is an indication that what is specified is not physically possible through an adjustment in the selected input parameters. It is suggested that the iteration be done in several stages in order to help deduce the source of the difficulty.

- *Physical interpretation of DELTAS(.) and suggested clamp sizes.* There is no rule of thumb for determining the appropriate clamp size; for some problems a clamp may not even be required. Generally, if between successive iterations the residue is reduced and does not switch sign, then the clamp can most likely be increased for each variable. But if the residue grows for a particular output variable or if the magnitude remains the same but switches sign, then the clamp size should be reduced. DELTAS(.) is the step size for a given variable for an iteration (see output given in Section 8.1). The clamp is applied to this variable and hence it is necessary to know the physical interpretation of DELTAS(.). Table 3 lists (1) the physical interpretation of DELTAS(.) corresponding to the parameters listed Tables 1 and 2 and (2) suggested clamp sizes as a starting point.

Keyword	DELTAS(.)	Clamp Size
Scale Rotor†	% Growth	0.20
Rotor Speed†	Δ Rotor Speed	5
Pitch†	Δ Pitch	1
Wind Speed†	Δ Wind Speed/TSR	1/()*
Cone Angle†	Δ Cone Angle	()*
Air Density†	Δ Air Density	()*
Rotor Radius†	Δ Rotor Radius	2
Blade Chord†	Δ Nondimensional Blade Chord	0.05
Offset Chord†	Δ Nondimensional Offset Chord	0.05
Blade Twist†	Δ Blade Twist	2
All Chords‡	Δ Chord	0.05
All Twists‡	Δ Twist	2

† NEWT1 case

‡ NEWT2 case

* no suggested value (rarely used)

9 Additional Input Lines

9.1 PAUSE Line

To pause program execution, the input file should contain the line:

```
PAUSE
```

Program execution will then continue when a `return` is entered interactively.

9.2 DRY Line

Sometimes it is useful to have **PROPID** echo to the screen the desired **NEWT1** and **NEWT2** line prescriptions without performing the iteration usually prompted by the **IDES** lines. To only show the prescriptions as a check on the input file, the line

```
DRY
```

is used to toggle on and off any following **IDES** lines. For example, if two **DRY** lines are used, the first will turn off all of the following **IDES** lines until the second **DRY** line is reached.

9.3 ITERMAX Line

When tolerances are specified for the automatic convergence mode, the maximum number of iterations can be specified via

```
ITERMAX <ITERMAX>
```

where **ITERMAX** is the maximum number of iterations. Iteration will then be performed until either convergence is reached or the number of iterations exceeds the maximum specified.

9.4 ZERO_TWIST Line

In some cases, it may be desirable to iterate on the twist along the entire blade span. In such instance, the twist at the 75% radial station will be adjusted from zero. Thus, the true blade pitch is the specified blade pitch plus the twist at the 75% station. If the converged input data were written out with the **DUMP_PROPID** line, the twist at the 75% station would not be zero. Prior to the **DUMP_PROPID** line, the **ZERO_TWIST** line can be used to zero the twist at the desired station by the line

```
ZERO_TWIST <RADLOC>
```

where `RADLOC` is the radius location for which the twist is to be zero. For example, to zero the twist at the 75% station `RADLOC` should be set to 0.75. The `ZERO_TWIST` line also properly adjusts the pitch in the DP lines to reflect the true blade pitch with zero twist at the prescribed location.

9.5 WT_NAME Line

Wind turbine design files can be given names in the input file by the line

```
WT_NAME <NAME>
```

where `NAME` is the name of the turbine.

9.6 PRINT_INPUT Line

As with the original `PROPSH` code, the input data can be written out for inspection. The line

```
PRINT_INPUT
```

is used to write the input data to logical unit 11.

9.7 BE_DATA Line

Blade element performance data like that of the `PROPSH` code can be written to logical unit 11 with the line

```
BE_DATA <IS>
```

where if `IS` is 1 data is printed — 0 and data is not printed (default). If data is printed in the design mode, the file can become thousands of lines long. Also, during the design mode the data is not particularly useful since the blade geometry and other parameters may be changing. The `BE_DATA` line is most useful during the analysis performed after the design process. When this data is desired during the analysis mode, the `BE_DATA` line should precede the `2D_SWEEP` and `1D_SWEEP` lines.

9.8 BEEP Line

Beeps to the screen can be sent with the line

```
BEEP
```

9.9 RNEWT Line

Sometimes it is useful to re-initialize the iteration process. The line

`RNEWT`

resets the iteration process anew. For example, suppose that after a series of `NEWT1` and `NEWT2` lines the number of variables used for iteration is large. `RNEWT` can be used to reset the iteration so that the next `NEWT/IDES` sequence only iterates on the variables following the `RNEWT` line. In this case, the previous specifications will no longer be satisfied.

10 Annotated Examples

The following example is from an input file used in generating data presented in Ref. [1]. In particular, the file below was used to design wind turbine case 5-5 as referred to in the paper. The solution is essentially converged already, but one iteration is performed to illustrate the output to the screen. The lengthy NREL airfoil data has been removed.

```
! Radius = 44.75 ft
! NREL airfoils
! Hub radius 35 in  -> HUB = (35/12)/44.75 = 0.0652

MODE 1.0           ! wind turbine
INCV 0.0           ! wind turbine
LTIP 1.0           ! tip loss model
LHUB 1.0           ! tip loss model
IBR 1.0            ! brake state model
ISTL 1.0           ! viterna stall model
USEAP 1.0          ! swiral
WEXP 0.0000        ! no boundary layer
HH 2.2400          ! tower height
NS_NSEC 10.0  1.0  ! number of blade elements/number of sectors
IS1 1.0            ! first segment used in analysis
IS2 10.0           ! last segment used in analysis
BE_DATA 0          ! don't printout blade element data
SH 0.0             ! shaft tilt effects on crossflow
BN 2.0             ! 2 blades
CONE 6.0           ! cone angle of rotor

RHO 0.002378
HUB 0.065200
RD 44.750000
CH_TW
  0.163175  29.2989
  0.123109  15.0263
  0.080087  8.5007
  0.058116  5.1895
  0.047366  3.2868
  0.043629  2.2542
  0.040730  1.1725
  0.037054  0.0000
  0.033025  -0.2953
  0.025503  -0.5866      ! 10 segments used
AIRFOIL_MODE 1
... ! All NREL airfoil data here
DP 1 65.0000 0.687 999.000 2
```

```
DP 2 999.0000 999.000 18.356 2
DP 3 999.0000 999.000 15.000 2
```

```
DRY ! All IDES lines will be ignored until DRY is encounter again below
```

```
! Set peak rotor power constraint
```

```
NEWT1ISWP 300 365 30 50 1. 1 1 1 2 999 100 .05
```

```
! Set peak power coefficient
```

```
NEWT1ISWP 302 0.45635067 13 30 1. 1 1 1 1 3 1 2
```

```
IDES
```

```
! Specify the lift coefficient on segment 8
```

```
NEWT1LDP 500 8 .65 1 1 2 1 4 2 1
```

```
! Specify the inboard relative Cl distribution
```

```
NEWT2SDDP 100 2 7 8 6
```

```
1 .308
```

```
2 .302
```

```
3 .276
```

```
4 .218
```

```
5 .118
```

```
6 .042
```

```
1 1 2 2 100 2
```

```
! Specify the outboard relative Cl distribution
```

```
NEWT2SDDP 100 9 10 8 2
```

```
1 -.030
```

```
2 -.053
```

```
1 1 2 2 100 2
```

```
! Specify a relative axial induction factor distribution (constant)
```

```
NEWT2SDDP 101 3 10 2 8
```

```
1 0.0
```

```
2 0.0
```

```
3 0.0
```

```
4 0.0
```

```
5 0.0
```

```
6 0.0
```

```
7 0.0
```

```
8 0.0
```

```
1 1 3 1 100 .5
```

```
DRY ! turn off dry run so that the follow IDES line will start the iteration
```

```
IDES
```

```
! Analyze the rotor
```

```
RPM_DP 1
```

```
PITCH_DP 1
```

```
WIND_SWEEP 7 50.0 1 2.0
```

2D_SWEEP

WRITE_FILES 40 45 55 100 50 95

! Reset the iteration scheme

RNEWT

! The following lines are used to find the values for

! (1) peak power coefficient

! (2) peak windfarm figure of merit

! (3) windfarm figure of merit for 15 mph case

! (4) axial induction factor for segment 5 (which is constant 2 -> 10)

! No iteration will be performed (see following output) since only the

! actual values are of interest here.

! The value specified is 0 in each case so the residue written to the

! screen will be the actual value. The variable for iteration is the

! same for each line...which would be incorrect if iteration were

! performed.

NEWT1ISWP 302 0.00 13 30 1. 1 1 1 1 3 1

NEWT1ISWP 303 0.00 13 40 1. 1 1 1 1 3 1

NEWT1IDP 201 0.00 1 1 3 1 3 1

NEWT1LDP 501 5 0.00 1 1 3 1 3 1

IDES

DUMP_PROPID

BEEP

*

The screen output and user interactive input follows.

... dry run turned on

NEWT1-line

1: Prescribed peak power (kW) = 365. at design point rpm(1) pitch(1)

Adjust chord uniformly with step limit = 0.050

NEWT1-line

2: Prescribed max Cp = 0.4564 at design point rpm(1) pitch(1)

Adjust pitch(1) with step limit = 2.000 (deg)

NEWT1-line

3: Prescribed Cl(8) = 0.65 at design point rpm(1) pitch(1) xj(2)

Adjust xj(2) with step limit = 1.000

NEWT2-line

1: Prescribed Cl dist from 2 to 7 relative to 8

at design point rpm(1) pitch(1) xj(2)

Adjust twist (each independently) from 2 to 7 with step limit = 2.00

NEWT2-line

2: Prescribed Cl dist from 9 to 10 relative to 8

at design point rpm(1) pitch(1) xj(2)

Adjust twist (each independently) from 9 to 10 with step limit = 2.00

NEWT2-line

3: Prescribed a dist from 3 to 10 relative to 2

at design point rpm(1) pitch(1) xj(3)

Adjust chord (each independently) from 3 to 10 with step limit = 0.50

... dry run turned off

Initial wind turbine design for stage: 2

Residues for NEWT1* equations:

FNT1_0(1) = -0.02527

FNT1_0(2) = -0.00035

FNT1_0(3) = 0.00003

Residues for NEWT2* equations:

FNT2_0(1) = -0.00059

FNT2_0(2) = 0.00024

FNT2_0(3) = -0.00002

FNT2_0(4) = 0.00022

FNT2_0(5) = -0.00020

FNT2_0(6) = 0.00000

FNT2_0(7) = 0.00000

FNT2_0(8) = 0.00049

FNT2_0(9) = -0.00000

FNT2_0(10) = 0.00000

FNT2_0(11) = 0.00000

FNT2_0(12) = 0.00001

FNT2_0(13) = 0.00001

FNT2_0(14) = 0.00001

FNT2_0(15) = 0.00000

FNT2_0(16) = 0.00000

Select option:

0 Stop iteration for current stage

Number of consecutive iterations

999 To stop execution

99 More options

>>1

```
Iteration    1
Calculating sensitivities for NEWT1 design parameter:    1
Calculating sensitivities for NEWT1 design parameter:    2
Calculating sensitivities for NEWT1 design parameter:    3
Calculating sensitivities for NEWT2 design parameter:    1
Calculating sensitivities for NEWT2 design parameter:    2
Calculating sensitivities for NEWT2 design parameter:    3
Calculating sensitivities for NEWT2 design parameter:    4
Calculating sensitivities for NEWT2 design parameter:    5
Calculating sensitivities for NEWT2 design parameter:    6
Calculating sensitivities for NEWT2 design parameter:    7
Calculating sensitivities for NEWT2 design parameter:    8
Calculating sensitivities for NEWT2 design parameter:    9
Calculating sensitivities for NEWT2 design parameter:   10
Calculating sensitivities for NEWT2 design parameter:   11
Calculating sensitivities for NEWT2 design parameter:   12
Calculating sensitivities for NEWT2 design parameter:   13
Calculating sensitivities for NEWT2 design parameter:   14
Calculating sensitivities for NEWT2 design parameter:   15
Calculating sensitivities for NEWT2 design parameter:   16
```

Residues for NEWT1* equations:

```
FNT1_1( 1) =  -0.00940  DELTAS1() = -0.00006  CLAMP1() = 0.050
FNT1_1( 2) =   0.00005  DELTAS1() = -0.00948  CLAMP1() = 2.000
FNT1_1( 3) =   0.00000  DELTAS1() = -0.01422  CLAMP1() = 1.000
```

Residues for NEWT2* equations:

```
FNT2_1( 1) =  -0.00008  DELTAS2() = -0.03057  CLAMP2() = 2.000
FNT2_1( 2) =  -0.00000  DELTAS2() = -0.01495  CLAMP2() = 2.000
FNT2_1( 3) =  -0.00000  DELTAS2() = -0.01061  CLAMP2() = 2.000
FNT2_1( 4) =  -0.00001  DELTAS2() = -0.00441  CLAMP2() = 2.000
FNT2_1( 5) =  -0.00000  DELTAS2() = -0.00520  CLAMP2() = 2.000
FNT2_1( 6) =   0.00000  DELTAS2() = -0.00148  CLAMP2() = 2.000
FNT2_1( 7) =  -0.00000  DELTAS2() =  0.00120  CLAMP2() = 2.000
FNT2_1( 8) =  -0.00001  DELTAS2() =  0.00633  CLAMP2() = 2.000
FNT2_1( 9) =   0.00003  DELTAS2() =  0.00011  CLAMP2() = 0.500
FNT2_1(10) =   0.00003  DELTAS2() =  0.00010  CLAMP2() = 0.500
FNT2_1(11) =   0.00003  DELTAS2() =  0.00011  CLAMP2() = 0.500
FNT2_1(12) =   0.00003  DELTAS2() =  0.00009  CLAMP2() = 0.500
FNT2_1(13) =   0.00003  DELTAS2() =  0.00010  CLAMP2() = 0.500
FNT2_1(14) =   0.00002  DELTAS2() =  0.00010  CLAMP2() = 0.500
FNT2_1(15) =   0.00002  DELTAS2() =  0.00010  CLAMP2() = 0.500
FNT2_1(16) =   0.00004  DELTAS2() =  0.00014  CLAMP2() = 0.500
```

Select option:

- 0 Stop iteration for current stage
- # Number of consecutive iterations
- 999 To stop execution
- 99 More options

>>0

Performing 2D sweep analysis...
...done performing 2D sweep analysis.

*****OUTPUT*****

- * rotor P vs V --> fort.40
- * rotor Cp vs X --> fort.45
- * rotor Cp vs wind speed --> fort.50
- * rotor figure of merit M --> fort.55
- * blade chord --> fort.95
- * blade twist --> fort.100

*****OUTPUT*****

Reseting Newton iteration...

NEWT1-line

- 1: Prescribed max Cp = 0.0000 at design point rpm(1) pitch(1)
Adjust pitch(1) with step limit = 0.000 (deg)

NEWT1-line

- 2: Prescribed max FOM = 0.0000 at design point rpm(1) pitch(1)
Adjust pitch(1) with step limit = 0.000 (deg)

NEWT1-line

- 3: Prescribed FOM = 0.0000 at design point rpm(1) pitch(1) xj(3)
Adjust pitch(1) with step limit = 0.000 (deg)

NEWT1-line

- 4: Prescribed a(5) = 0.000 at design point rpm(1) pitch(1) xj(3)
Adjust pitch(1) with step limit = 0.000 (deg)

Initial wind turbine design for stage: 1

Residues for NEWT1* equations:

- FNT1_0(1) = 0.45640
- FNT1_0(2) = 0.90372
- FNT1_0(3) = 0.75735
- FNT1_0(4) = 0.31688

Select option:

```
0 Stop iteration for current stage
# Number of consecutive iterations
999 To stop execution
99 More options
```

>>0

```
*****
* Writing PROPID dump file (21) *
*****
```

This last example is that taken from Section [8.2] in complete form. It is quite similar to the preceding example.

```
...
DP 1 50.0000 2.500 999.000 2
DP 2 999.0000 999.000 14.444 2
! prescribed peak power
NEWT1ISWP 300 75 20 35 1. 1 1 1 2 999 100
IDES
! prescribed lift coefficient for segment 8
NEWT1LDP 500 8 .65 1 1 2 1 4 2 1
IDES
! prescribed relative lift coefficient distribution
NEWT2SDDP 100 2 7 8 6
1 .308
2 .302
3 .276
4 .218
5 .118
6 .042
1 1 2 2 100 2
NEWT2SDDP 100 9 10 8 2
1 -.030
2 -.053
1 1 2 2 100 2
IDES
! prescribed relative axial induction factor distribution (constant)
NEWT2SDDP 101 3 10 2 8
1 0.00
2 0.00
3 0.00
4 0.00
5 0.00
```

6 0.00
7 0.00
8 0.00
1 1 2 1 100 .5
IDES
*

References

- [1] Selig, M.S. & Tangler, J.L., “A Multipoint Inverse Design Method for Horizontal Axis Wind Turbines,” presented at the AWEA WINDPOWER’94 Conference, Minneapolis, Minnesota, May 9–12, 1994.
- [2] 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.
- [3] Selig, M.S., “Multipoint Inverse Design of an Infinite Cascade of Airfoils,” *AIAA J.*, Vol. 32, No. 4, April 1994, pp. 774–782..
- [4] Tangler, J., Smith, B., Kelley, N. & Jager, D., “Measured and Predicted Rotor Performance for the SERI Advanced Wind Turbine Blades,” NREL/TP-257-4594, Feb., 1992.
- [5] Hibbs, B. & Radkey, R.L., “Small Wind Energy Conversion Systems (SWECS) Rotor Performance Model Comparison Study,” Aerovironment, Inc., prepared for Rockwell International Corporation, Nov., 1981.
- [6] Tangler, J.L., “A Horizontal Axis Wind Turbine Performance Prediction Code for Personal Computers (User’s Guide),” Solar Energy Research Institute, Jan., 1987.

A Appendix: AeroStar 7.4 m Airfoil Data

What follows is the complete input airfoil data for the Aerostar 7.4 m blade.

```
AIRFOIL_MODE 1
1.0      10.0      13.0
-4.0     0.039
-2.0     0.248
 0.0     0.459
 2.0     0.664
 4.0     0.865
 6.0     1.053
 8.0     1.225
 8.78    1.286
12.0     1.286
15.0     1.286
-4.0     0.0104
-2.0     0.0104
 0.0     0.0104
 2.0     0.0104
 4.0     0.0113
 6.0     0.0130
 8.0     0.0156
 8.78    0.0169
12.0     0.0247
15.0     0.0700
 20.0000 .1750
 25.0000 .2750
 27.5000 .3630
2.0      0.0      1.0
3.0      0.0      1.0
4.0      10.0     13.0
-4.0     0.038
-2.0     0.254
 0.0     0.470
 2.0     0.682
 4.0     0.892
 6.0     1.091
 8.0     1.275
10.03    1.426
12.0     1.426
15.0     1.426
-4.0     0.0091
-2.0     0.0091
 0.0     0.0090
 2.0     0.0088
```

4.0	0.0095
6.0	0.0112
8.0	0.0140
10.03	0.0187
12.0	0.0251
15.0	0.0700
20.0000	.1750
25.0000	.2750
27.5000	.3630

5 10 13

-4.0	0.039
-2.0	0.255
0.0	0.471
2.0	0.684
4.0	0.893
6.0	1.094
8.0	1.278
10.1	1.433
12.0	1.433
15.0	1.433
-4.0	0.0089
-2.0	0.0088
0.0	0.0087
2.0	0.0086
4.0	0.0093
6.0	0.0109
8.0	0.0137
10.1	0.0186
12.0	0.0243
15.0	0.0700
20.0000	.1750
25.0000	.2750
27.5000	.3630

6 10 13

-4.0	0.034
-2.0	0.253
0.0	0.473
2.0	0.690
4.0	0.905
6.0	1.115
8.0	1.306
10.7	1.505
12.0	1.505
15.0	1.505
-4.0	0.0080
-2.0	0.0078

0.0	0.0078	
2.0	0.0071	
4.0	0.0080	
6.0	0.0098	
8.0	0.0137	
10.7	0.0207	
12.0	0.0235	
15.0	0.0700	
20.0000	.1750	
25.0000	.2750	
27.5000	.3630	
7	10	13
-4.0	0.033	
-2.0	0.252	
0.0	0.472	
2.0	0.691	
4.0	0.907	
6.0	1.119	
8.0	1.302	
11.04	1.543	
12.0	1.543	
15.0	1.543	
-4.0	0.0077	
-2.0	0.0077	
0.0	0.0077	
2.0	0.0068	
4.0	0.0077	
6.0	0.0095	
8.0	0.0153	
11.04	0.0210	
12.0	0.0234	
15.0	0.0700	
20.0000	.1750	
25.0000	.2750	
27.5000	.3630	
8	10	13
-4.0	0.033	
-2.0	0.252	
0.0	0.472	
2.0	0.691	
4.0	0.907	
6.0	1.120	
8.0	1.302	
11.09	1.548	
12.0	1.548	
15.0	1.548	

-4.0	0.0077
-2.0	0.0077
0.0	0.0076
2.0	0.0067
4.0	0.0077
6.0	0.0095
8.0	0.0152
11.09	0.0210
12.0	0.0231
15.0	0.0700
20.0000	.1750
25.0000	.2750
27.5000	.3630

9 10 13

-4.0	0.033
-2.0	0.252
0.0	0.472
2.0	0.691
4.0	0.907
6.0	1.120
8.0	1.302
11.08	1.547
12.0	1.547
15.0	1.547
-4.0	0.0077
-2.0	0.0077
0.0	0.0076
2.0	0.0067
4.0	0.0077
6.0	0.0095
8.0	0.0152
11.08	0.0210
12.0	0.0231
15.0	0.0700
20.0000	.1750
25.0000	.2750
27.5000	.3630

10 10 13

-4.0	0.033
-2.0	0.252
0.0	0.472
2.0	0.691
4.0	0.907
6.0	1.120
8.0	1.302
11.06	1.545

12.0	1.545
15.0	1.545
-4.0	0.0077
-2.0	0.0077
0.0	0.0077
2.0	0.0067
4.0	0.0077
6.0	0.0095
8.0	0.0152
11.06	0.0210
12.0	0.0232
15.0	0.0700
20.0000	.1750
25.0000	.2750
27.5000	.3630

B Appendix: Airfoil Data Options

If two or more blade elements have the same airfoil data as a preceding element, then the line

```
...  
<JSEG> <MM(JSEG)> <NN(JSEG)>  
...
```

can be used for the current segment during a sequence of `AIRFOIL_MODE` data. `JSEG` is the current segment number. `MM(JSEG)` must be zero to indicate that no new data is required. `NN(JSEG)` is the preceding segment from which the data is to be taken. It is not necessary for current segment to immediately follow segment `NN(JSEG)`. Examples can be found in the AeroStar 7.4 m data presented in Appendix A.

C Appendix: Quick Reference

=====
= Files on distribution disk
=====

AEROSTAR.IPT ! AeroStar 7.4 m blade input file
EXAMPLE1.IPT ! Example input file
EXAMPLE2.IPT ! Example input file
PROPID.FOR ! Lahey F77L EM/32 version of source code
PROPID.INC ! Include file
README.TXT ! This file
RUNFILES.TXT ! Assigns INFILE to the input file (AEROSTAR.IPT currently)
DATFILES <DIR> ! Directory with output files generated by PROPID with the
! AEROSTAR.IPT file as input with no iteration.

=====
= Required data for design and analysis modes
=====

MODE 1.0 ! wind turbine
INCV 0.0 ! wind turbine
LTIP 1.0 ! include tip loss model
LHUB 1.0 ! include tip loss model
IBR 1.0 ! use classical brake state model
ISTL 1.0 ! use viterna stall model
USEAP 1.0 ! include swirl
WEXP 0.0 ! no boundary layer
HH 2.24 ! normalized hub height (height/rotor radius)
NS_NSEC 10.0 1.0 ! number of blade elements/number of sectors
IS1 1.0 ! first segment used in analysis
IS2 10.0 ! last segment used in analysis
BE_DATA 0 ! do not printout blade element data
SH 0.0 ! ignore shaft tilt effects on crossflow
BN 2.0 ! 2 blades
CONE 6.0 ! cone angle of rotor (deg)
RHO 0.002378 ! air density (lb sec²/ft⁴)
HUB 0.0652 ! normalized hub radius (hub radius/rotor radius)
RD 44.75 ! rotor radius (ft)

=====
= Chord (chord/rotor radius) and twist (deg) distributions
= Twist is measure positive toward feather
=====

```

CH_TW
<CH(1)>    <TW(1)>           ! root chord and twist
...
<CH(JSEG)> <TW(JSEG)>
...
<CH(ISEG)> <TW(ISEG)>       ! tip  chord and twist

```

```

=====
= Airfoil data
=====

```

```

AIRFOIL_MODE    1
<1>    <MM(1)>    <NN(1)>      ! first segment
<AL(1)>    <CL(1)>
...
<AL(MM(1))> <CL(MM(1))>
<AD(1)>    <CD(1)>
...
<AD(MM(1))> <CD(MM(1))>
...
<JSEG> <MM(JSEG)> <NN(JSEG)>  ! intermediate segment
<AL(JSEG)>    <CL(JSEG)>
...
<AL(MM(JSEG))> <CL(MM(JSEG))>
<AD(JSEG)>    <CD(JSEG)>
...
<AD(MM(JSEG))> <CD(MM(JSEG))>
...
<ISEG> <MM(ISEG)> <NN(ISEG)>  ! last segment
<AL(ISEG)>    <CL(ISEG)>
...
<AL(MM(ISEG))> <CL(MM(ISEG))>
<AD(ISEG)>    <CD(ISEG)>
...
<AD(MM(ISEG))> <CD(MM(ISEG))>

```

```

=====
= Design points
=====

```

```

DP <IDP> <RPMDP(IDP)> <FLDP(IDP)> <XJDP(IDP)> <IXDIMDP(IDP)>
    ! IDP          design point number
    ! RPMDP(IDP)  rotor speed (rpm)

```

```

! FLDP(IDP) blade pitch (deg) at 75\% of radius
! XJDP(IDP) speed as indicated by the units parameter IXDIMDP(IDP)
! IXDIMDP(IDP)} = 0 wind speed in (ft/sec)
!           1 wind speed in (m/sec)
!           2 wind speed in (mph)
!           3 tip speed ratio

! example with rpm = 65, pitch @ 75% = 1.4 deg, wind speed = 15 mph
DP  1  65.0000  0.687  15.  2

```

```

=====
= Analysis lines
=====

```

```

! ***** 2D_SWEEP for power curves, etc.
! Use one pitch line and one rpm line then WIND_SWEEP and 2D_SWEEP.
! But do not pick both PITCH_SWEEP and RPM_SWEEP.
! Use at most 2 sweep lines.

```

```

! Blade pitch
PITCH_FIXED <FL>           ! fix pitch to <FL> deg
PITCH_DP    <JDPFL>        ! pitch referenced to DP line <JDPFL>
PITCH_SWEEP <FLS> <FLF> <DFL> ! sweep pitch
! FLS initial (start) value for the pitch
! FLF final value, and
! DFL increment

```

```

! Rotor speed
RPM_FIXED <RPM>           ! fix rotor speed to <RPM> (revs/min)
RPM_DP    <JDPRPM>        ! pitch referenced to DP line <JDPFL>
RPM_SWEEP <RPMS> <RPMF> <DRPM> ! sweep rpm

```

```

! Wind speed
WIND_SWEEP <XJS> <XJF> <DXJ> <IXDIM> ! sweep wind
! XJS initial (start) value
! XJF final value
! DXJ increment
! IXDIM units (see DP line for convention)

```

```

! perform analysis
2D_SWEEP

```

```

! ***** 1D_SWEEP for cl distribution, etc.
! Use at most one sweep line.

```

```

PITCH_FIXED <FL>                ! fix pitch to <FL> deg
PITCH_DP    <JDPFL>              ! pitch referenced to DP line <JDPFL>
PITCH_SWEEP <FLS> <FLF> <DFL>   ! sweep pitch

RPM_FIXED <RPM>                  ! fix rotor speed to <RPM> (revs/min)
RPM_DP     <JDPRPM>              ! pitch referenced to DP line <JDPFL>
RPM_SWEEP <RPMS> <RPMF> <DRPM>  ! sweep rpm

WIND_FIXED <XJ> <IXDIM>          ! fix wind speed
WIND_DP    <JDPWND>              ! wind speed referenced to DP line <JDPFL>
WIND_SWEEP <XJS> <XJF> <DXJ> <IXDIM> ! sweep wind

```

```
1D_SWEEP
```

```

=====
= Output files
=====

```

```

WRITE_FILES <IPRT(1)> <IPRT(2)> <IPRT(3)> ... | ... <IPRT(20)>
! Files generated by the 2D_SWEEP line
! IPRT(.) Data                               File name FORT*.DAT
! 40     rotor power vs. wind speed          FORT040.DAT
! 45     rotor power coefficient vs. TSR     FORT045.DAT
! 50     rotor power coefficient vs. wind speed FORT055.DAT
! Files generated by the 1D_SWEEP line
! 60     blade power distribution            FORT060.DAT
! 65     blade power coefficient distribution FORT065.DAT
! 75     blade airfoil lift-to-drag ratio distribution FORT075.DAT
! 80     blade angle of attack distribution  FORT080.DAT
! 85     blade lift coefficient distribution FORT085.DAT
! 90     blade axial induction factor distribution FORT090.DAT
! Others
! 95     blade chord distribution            FORT095.DAT
! 100    blade twist distribution           FORT100.DAT

```

```

=====
= NEWT1* lines
=====

```

```

NEWT1ISWP <IFTP1(.)> <FNEWT1(.)> <XJSNT1(.)> <XJFNT1(.)> <DXJNT1(.)> -
          <KDPRPM1(.)> <KDPFL1(.)> <KDPXJ1(.)> -
          <ITP1(.)> <ITP2(.)> <ITP3(.)> | <CLAMP1(.)> <TOL1(.)>

! IFTP1(.) 300 peak rotor power (kW)

```

```

!           301  wind speed (mph) at peak power
!           302  peak power coefficient
! FNEWT1(.)  value for specified parameter
! XJSNT1(.)  lowest value for wind speed (mph) for range
! XJFNT1(.)  highest value for wind speed (mph) for range
! DXJNT1(.)  increment in wind speed (mph)
! KDPRPM1(.) design point # for rotor speed
! KDPFL1(.)  design point # for blade pitch
! KDPXJ1(.)  999 (value is ignored)
! ITP1(.)    used to identify variable for iteration (see table)
! ITP2(.)    used to identify variable for iteration (see table)
! ITP3(.)    used to identify variable for iteration (see table)

```

```

NEWT1IDP <IFTP1(.)> <FNEWT1(.)> -
<KDPRPM1(.)> <KDPFL1(.)> <KDPXJ1(.)> -
<ITP1(.)> <ITP2(.)> <ITP3(.)> | <CLAMP1(.)> <TOL1(.)>

```

```

! IFTP1(.)  200 power (kW)
! FNEWT1(.)  value for specified parameter
! KDPRPM1(.) design point # for rotor speed
! KDPFL1(.)  design point # for blade pitch
! KDPXJ1(.)  design point # for wind speed
! ITP1(.)    used to identify variable for iteration (see table)
! ITP2(.)    used to identify variable for iteration (see table)
! ITP3(.)    used to identify variable for iteration (see table)

```

```

NEWT1LDP <IFTP1(.)> <JSEGIX1(.)> <FNEWT1(.)> -
<KDPRPM1(.)> <KDPFL1(.)> <KDPXJ1(.)> -
<ITP1(.)> <ITP2(.)> <ITP3(.)> | <CLAMP1(.)> <TOL1(.)>

```

```

! IFTP1(.)  500 local lift coefficient
!           501 local axial induction factor
!           502 local airfoil angle of attack
!           504 local power coefficient
!           505 local power
! JSEGIX1(.) blade segment for specified parameter
! FNEWT1(.)  value for specified parameter
! KDPRPM1(.) design point # for rotor speed
! KDPFL1(.)  design point # for blade pitch
! KDPXJ1(.)  design point # for wind speed
! ITP1(.)    used to identify variable for iteration (see table)
! ITP2(.)    used to identify variable for iteration (see table)
! ITP3(.)    used to identify variable for iteration (see table)

```

```
-----
```

Input Variable Specification for NEWT1* Iteration			
ITP1(.)	ITP2(.)	ITP3(.)	CLAMP1(.) - suggested
1	1 - Scale Rotor	999	(see table 3)
	2 - Rotor Speed	Design Point #	(see table 3)
	3 - Pitch	Design Point #	2
	4 - Wind Speed	Design Point #	(see table 3)
	5 - Cone Angle	999	(see table 3)
	6 - Air Density	999	(see table 3)
	7 - Rotor Radius	999	(see table 3)
2	Local Blade Chord #	999	0.05
	999	100 - Offset Chord	0.05
3	Local Blade Twist #	999	2

```
-----
```

```
=====
= NEWT2* lines
=====
```

```
NEWT2SDDP <IFTP2(.)> <JSEGIX2(.)> <JSEGIX3(.)> <JSEGREL(.)> <KADJSBS(.)>
<SSS(1)> <SSF(1)>
```

```
...
<SSS(KADJSBS(.))> <SSF(KADJSBS(.))>
<KDP RPM2(.)> <KDPFL2(.)> <KDPXJ2(.)> <ISDTP(.)> <ISCHED2(.)> | -
      <CLAMP2(.) TOL2(.)
```

```
! IFTP1(.) 100 relative lift coefficient distribution
!           101 relative axial induction factor distribution
!           102 relative airfoil angle of attack distribution
!           104 relative power coefficient distribution
!           105 relative power distribution
! JSEGIX2(.) inboard segment for relative distribution
! JSEGIX3(.) outboard segment for relative distribution
! JSEGREL(.) relative segment, i.e., where the relative distribution is 0
! KADJSBS(.) number of segments [JSEGIX3(.) - JSEGIX2(.)]
! SSS(1) first segment prescribed (always = 1)
! SSF(1) relative value for first segment
! SSS(KADJSBS(.)) last segment prescribed (always = JSEGREL(.))
! SSF(KADJSBS(.)) relative value for last segment
! KDP RPM2(.) design point # for rotor speed
! KDPFL2(.) design point # for blade pitch
! KDPXJ2(.) design point # for wind speed
! ISDTP(.) used to identify input variable for iteration (see table)
! ISCHED2(.) used to identify input variable for iteration (see table)
```

```

-----
Input Variable Specification for NEWT2* Iteration
ISDTP(.)  ISCHED2(.)          ITP3(.)          CLAMP2(.) - suggested
1         100 - Move corresponding chord independently  0.05
2         100 - Move corresponding twist independently   2
-----

```

```

=====
= Misc lines
=====

```

```

DUMP_PROPID ! write converged data to the file FORT021.DAT
DUMP_PROP93 <JDP> ! write the file FORT022.DAT for a PROP93 run
PAUSE      ! pause program execution
DRY        ! perform a 'dry run' to check NEWT lines
ITERMAX <ITERMAX> ! max number of iterations to perform
ZERO_TWIST <RADLOC> ! zero the pitch at the location <RADLOC>
              ! in the range [0:1]
WT_NAME <NAME> ! NAME = wind turbine name
PRINT_INPUT ! print input for checking data
BE_DATA <IS> ! IS = 0/1 do not print/print blade element data from analysis run
BEEP      ! send a beep
RNEWT     ! reset iteration scheme

```

```

=====
= Comments, blank lines, and program termination
=====

```

```

! A line beginning with an '!' is a comment line
! blank lines are ignored
* ! used to stop execution

```

Notes