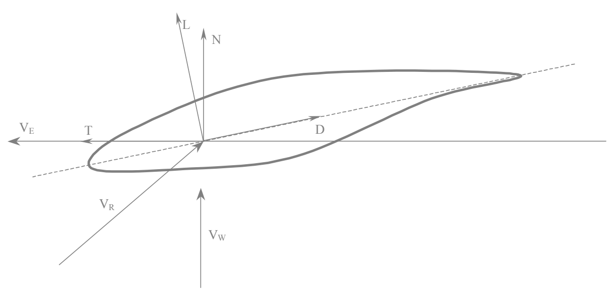
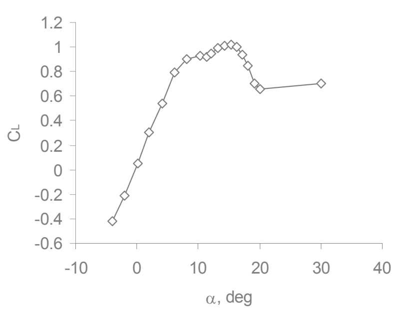
# USER'S GUIDE

to the Wind Turbine Aerodynamics Computer Software

# AeroDyn





David J. Laino
A. Craig Hansen
Windward Engineering, LC
Salt Lake City, UT 84117
www.windwardengineering.com

Phone: 801-278-7852
Fax: 801-272-4132
email: dlaino@ windwardengineering.com
chansen@windwardengineering.com


Software date and version
AeroDyn 12.50, 24-Dec-2002


Prepared for the
National Renewable Energy Laboratory
under Subcontract No.
TCX-9-29209-01

## Notice

**Table of Contents**

## List of Figures



## List of Tables

# USER'S GUIDE

to the Wind Turbine Aerodynamics Computer Software

# AeroDyn

## About this Guide

This Users' Guide is written to assist engineers with the preparation and use of the computer software AeroDyn for wind turbine aerodynamics. This software is a plug-in type code for interfacing with a number of dynamics programs. It has been interfaced with the YawDyn, SymDyn, and FAST wind turbine dynamics codes as well as the ADAMS® commercial dynamics analysis package. This offers the engineer a vast selection of modeling capabilities. User's guides for YawDyn and the ADAMS interface are also available.

This Guide outlines, in general terms, how AeroDyn interfaces with the dynamics codes. When the proper conventions are used and the necessary information is passed to AeroDyn, AeroDyn provides aerodynamic loads on a wind turbine blade element at the desired location and time. AeroDyn provides a straightforward interface to dynamics codes. This guide only addresses AeroDyn. Please refer to guides provided with dynamics programs for information about specifics regarding those codes.

Note that AeroDyn is NOT a stand-alone program, but software designed to interface with dynamics programs. For programs such as YawDyn, which are written specifically to work with AeroDyn, only the dynamics program and AeroDyn are required to run the program (i.e., the interface to AeroDyn is "built in"). Running ADAMS, however, requires not only ADAMS and AeroDyn, but a set of subroutines to interface the two codes. This interface, the AeroDyn Interface to ADAMS, is available with a User's Guide of its own for those who wish to use AeroDyn with ADAMS.

<u>Other Guides in this Series</u>

This guide is one in a set of three (as of the publication date on the cover), which include:

1. The AeroDyn 12.3 Users' Guide (this guide)
2. The YawDyn 12.0 Users' Guide, and
3. The AeroDyn Interface for ADAMS 12.02 User's Guide.

Depending on which programs you intend to run, you may need to refer to other User's Guides in the series. If you plan to use ADAMS with AeroDyn, you will need guides 1 (this guide) and 3. If you plan to use YawDyn with AeroDyn, you will need guides 1 (this guide) and 2.

# 1.0 Major changes since the last version of AeroDyn

This chapter provides assistance to experienced users of AeroDyn in updating models for use with the latest version of the code. New users can skip to the Introduction.

There are two major sections to this chapter. Users new to version 12 should read section 1.1 before continuing on to section 1.2. Users who are familiar with the version 12 of AeroDyn, and are upgrading to version 12.4 or later, can skip directly to section 1.2

## 1.1 Overview of Changes from Versions 11 to 12

AeroDyn has undergone a major overhaul in terms of functionality from version 11 to 12. The changes that will most affect users of the code are: 1) Old input files are not compatible with AeroDyn 12.3; 2) The YawDynVB Windows interface is no longer compatible or supported; and 3) AeroDyn is no longer bundled with any dynamics software package. Details on all of these statements are provided below. Upgrading AeroDyn (or any of the dynamics packages it interfaces with) is not a trivial matter. When you plan to update your software, it will be useful to maintain the earlier version while you convert over to version 12.3.

Since AeroDyn is now used with 4 different dynamics codes (YawDyn, SymDyn, FAST_AD, and ADAMS), it is no longer bundled with YawDyn and the ADAMS interface routines. By separating the codes and making AeroDyn thoroughly independent, the interface to the codes is more universal, and maintaining the independent codes is easier. This also means you will not be forced to update other codes when AeroDyn is updated.

Because of the interplay between AeroDyn and so many other codes, we have found a directory structure that places all the codes under one parent directory to work best for purposes of updating codes and minimizing confusion as to which version of what code is being used. We use a parent directory named "AeroDyn Programs", and then execute the self-extracting archive for each code in that directory. This places the code in a subdirectory for that program. An "AeroDyn" subdirectory has the AeroDyn code in it, and a "YawDyn" subdirectory the YawDyn code. Maintaining a directory structure such as this helps to keep track of current versions of each code, and simplifies the update of executables dependent on different codes.

AeroDyn 12.3 uses free-format Fortran 90 conventions. The "include" files of previous versions have been eliminated and replaced with modules to transport variables between subroutines. The result is that AeroDyn now consists of 3 source code files:

1. AeroSubs.F90 –subroutines to perform the aerodynamics calculations.
2. GenSubs.F90 – supporting routines to perform non-aerodynamic calculation functions.
3. AeroMods.F90 – the modules containing variables required by the AeroDyn code.

AeroDyn cannot be used as a stand-alone program. To create an executable code, it must be compiled and linked with another dynamics analysis code. Performing this task is addressed in the literature provided with those codes.

### 1.1.2 Changes to Inputs

AeroDyn 12.3 uses a new input file. Previously, AeroDyn inputs were carried in the yawdyn.ipt file. There is still a yawdyn.ipt file for use with YawDyn, but that file now contains only inputs used in YawDyn. The AeroDyn inputs are now in the file aerodyn.ipt. This file is used with AeroDyn regardless of the dynamics code being used (though some inputs may not be used or may have different meanings depending on the dynamics code used). A conversion program is available to convert version 11.0 yawdyn.ipt files to version 12.0 yawdyn.ipt files and version 12.3 aerodyn.ipt files. Refer to Table 6.2 for a description of each line of the aerodyn.ipt file. No other input file formats have been changed. Wind and airfoil data files do not need to be modified to work with AeroDyn 12.3.

### 1.1.3 Changes to Outputs

Output files created running simulations that use AeroDyn will depend upon the dynamics program used. However, AeroDyn will affect at least two of these files. The first is the OPT file (e.g., yawdyn.opt in YawDyn, or gfosub.opt in ADAMS) which will include a nearly line-by-line review of the inputs in aerodyn.ipt. The second file is the element output file (e.g., element.plt in YawDyn, or reqelem.plt in ADAMS – formerly aelement.plt), which is an option in the aerodyn.ipt file. Other time series output files (e.g., yawdyn.plt in YawDyn, and reqsub1.plt in ADAMS) are opened by AeroDyn to provide consistent program identifier headings.

AeroDyn now keeps track of errors and warnings experienced during simulations. These are logged in a file named error.log. This file is cumulative, meaning it is not overwritten, making it possible to keep track of errors over batches of simulations. As a result it is possible for this file to grow large over time, so it may need to be manually deleted after a period of time.

An attempt has been made to reduce screen outputs from AeroDyn. Some information is still provided, including any errors or warnings encountered. These messages are recorded in the error.log file.

### 1.1.4 New Dynamic Inflow Model

A major change has been made in the functionality of the dynamic inflow calculations of AeroDyn, which are conducted when the DYNIN option is selected in aerodyn.ipt. The new dynamic inflow model is based on the Generalized Dynamic Wake (GDW) model, which is a completely different theory than blade element/momentum (BEM) used for the EQUIL option. This is a significant change from the version 11.0 DYNIN model, which used a Pitt & Peters model. Refer to Appendix E for more details on the GDW model.

When using DYNIN in YawDyn, YawDyn seeks a trim solution for the mean (over the rotor area) induced velocity using the induction factor tolerance (ATOLER). If a trim solution is not found after 50 iterations, the simulation will terminate. The user should attempt a larger tolerance, or use the EQUIL option instead of DYNIN to run the simulation. When using other codes that do not conduct a trim solution, such as ADAMS, start up transients may occur. Discarding the first 5 – 30 seconds of a simulation will avoid any effects these may have on results.

There are three points to stress regarding the new GDW model:

1. The GDW model reduces simulation times compared to the BEM model. This is because the BEM model (EQUIL option) iterates for the induction factor, whereas the GDW model (DYNIN option), does not.
2. The dynamic inflow effect is insignificant except when rapid changes in blade angle-of-attack occur (e.g., large yaw error or shear). Outside of those cases, results should be comparable to those from the BEM model.
3. As with any new software feature, we recommend you use it cautiously, and review your results carefully for obvious errors. This model has not been extensively tested. One known problem occurs when the rotor model is in the "brake state" (very low wind speeds where the rotor does not produce power). In such cases where the wake does not progress downstream quickly, the GDW model can be erratic. Be very careful when attempting to model such situations with the DYNIN option.

We recommend the use of the DYNIN option, with careful consideration to the caveats above.

### 1.1.5 Other Programming Changes

Most arrays in AeroDyn are now dynamically allocated which permits any number of elements, blades, airfoil data points, etc. to be used in a simulation without the need to recompile the code for unusually large values. There is one notable exception: if multiple airfoil tables are used in one airfoil data file, the limit is currently set to 10. This is of little consequence to most users.

All references to the multiple airfoil data table option have been generalized throughout the source code. Where references to aileron angle were used, the more general table location term is used. This is done to avoid confusion and highlight the generality of this option. It can be used to handle ailerons and other aerodynamic devices as well as Reynold's number or any other parameter that may affect the element aerodynamics. This change will only be of concern to those who choose to alter the source code of Aero-Dyn.

## 1.2 Overview of Changes in Version 12.4

Version 12.4 has seen several changes, including changes to the input file for AeroDyn. These changes are:

- Addition of a correction to the Prandlt tip loss model, as presented in Xu and Sankar (2002) of the Georgia Institute of Technology. The tip loss model can now be selected on line 9 the AeroDyn input file as either PRAND or GTECH, or it can be turned off using NONE. These selections are only effective when using the equilibrium wake model (EQUIL) option in AeroDyn.

- Kinematic air viscosity has been added to line 15 of the AeroDyn input file. It is used to calculate element Reynold's number, which in turn can be used to move between multiple airfoil data tables when simulating Reynold's sensitive airfoil data.

- The SINGLE/MULTIple airfoil table option has been expanded. The choices are now SINGLE (or blank), ReNum for moving between multiple tables based on Reynold's number, and USER for moving between tables based on user defined parameters (such as aileron angle). See section 9.1 for more details on this option.

- File names (hub-height wind or full-field turbulence, and airfoil data) must now be put in quoted strings. This permits the use of comments on the same line as the file names, which can be useful for some users.

Other changes to the code are not as obvious to the user. Most important of these is a fix to a bug introduced during the change from version 11 to 12. The average inflow across the rotor was incorrectly calculated resulting in errors in the skewed wake calculations. This error will only manifest itself when large yaw errors are present in a simulation.

Another minor bug was found in the average inflow calculation. The average inflow across the rotor must be determined from induction factors for the previous time step so they can be averaged across the rotor. However, AeroDyn was using the average inflow from two time steps prior to the current. The difference should be nearly impossible to notice, unless particularly large aerodynamic time steps are used in a simulation.

Other changes of importance are updates to some internal utilities. Unfortunately, these updates affect other codes linked to AeroDyn. For compatibility, YawDyn, SymDyn and ADAMS2AD must be updated for use with AeroDyn 12.4 or later. This result is unfortunate, and can be considered a "growing pain" of the new interface. We don't foresee this problem as a common one in future code updates.

1.2.1 FAST and AeroDyn 12

Some of the changes in AeroDyn 12.4 are the result of bringing the FAST code up to date to use AeroDyn 12. There are features in FAST that are not yet available to the other codes linked with AeroDyn. These include:

- The AeroDyn input file name is declared in the FAST input file. The name "aerodyn.ipt" is not required to be used with FAST, though it can be used if specified as such in the FAST input file.

- If an element data output file is created when running FAST, it will be named [rootname].elm, where [rootname] is the prefix of the FAST input file name.

- AeroDyn inputs and other information regarding the simulation are written to an opt file named [rootname].opt. The information written here is similar to that found in other opt files (e.g., yawdyn.opt or gfosub.opt)

For more information on FAST, see the FAST Users' Guide.

1.3 Overview of Changes in Version 12.5

Version 1.5 sees two additional features added to the code. These are the addition of the ability to read 4D wind inflow files in the format used by NCAR's Large Eddy Simulation of a K-H wave, and a Prandtl hub-loss model. Version 12.5 includes only one change to the format of the input file: the addition of the switch to turn on the hub loss model. See the section on the input file format for details.

## 2.0 Introduction

This document is intended to provide information necessary to use the computer software AeroDyn with a dynamics simulation package. This guide will outline how the interface to AeroDyn is designed and what AeroDyn expects of and provides to those codes. This guide is one of a current set of three covering AeroDyn, YawDyn, and The AeroDyn Interface for ADAMS software. Depending on how you intend to use AeroDyn, you should acquire the appropriate software and User's Guides for your task.

AeroDyn was developed and is maintained with the support of the National Renewable Energy Laboratory (NREL) National Wind Technology Center (NWTC). AeroDyn calculates the aerodynamic loads on wind turbine blade elements based on velocities and positions provided by dynamics analysis routines and simulated wind inputs. Sample input and output files used by AeroDyn are provided with the distribution code. When interfaced properly with a dynamics analysis code such as YawDyn, these files can be used to run AeroDyn. This Guide contains no discussion of the underlying theories used in AeroDyn, or the limitations of the models. That discussion is available in technical reports and journal articles [see list of references].

In 1992, the aerodynamics analysis subroutines from YawDyn were modified for use with the ADAMS® program, which is available from Mechanical Dynamics, Inc. (Ann Arbor, MI). At this point, the separate AeroDyn subroutines were born. These subroutines were bundled with YawDyn through version 11.0 (1998). There was much redundant code carried in AeroDyn, YawDyn and the ADAMS interface. This redundant code has now been eliminated in version 12.3 making updating all of the codes simpler. The ability to use AeroDyn with (currently 4) different dynamics codes allows the engineer to select just the amount of complexity required for a model. This creates the most versatile and powerful wind turbine dynamics modeling capability known to the authors.

AeroDyn is an element-level wind-turbine aerodynamics analysis routine. It requires information on the status of a wind turbine from the dynamics analysis routine and returns the aerodynamic loads for each blade element to the dynamics routines. To do this properly, conventions outlined in this guide must be followed. A flow-chart outlining how AeroDyn calculates aerodynamic forces for dynamics simulations is presented in Appendix B.

This version of the User's Guide is current as of the date and version shown on the cover page. It is applicable only to the specified version of the code. Since the software development is continuing, and significant changes are continuously being made to the codes, the reader should be certain the guide is appropriate to the version being used. Research is ongoing regarding the strengths and limitations of the YawDyn and AeroDyn codes. Users may wish to consult recent wind energy literature to improve their understanding of the code and its accuracy. The change.log file also maintains a running list of changes and improvements to the code.

### 2.1 The YawDynVB Program

With version 10.0 of YawDyn we introduced a new Windows interface named YawDynVB. This "point-and-click" interface was updated to version 2.0 for compatibility with YawDyn/AeroDyn 11.0. However, YawDynVB is no longer maintained and is not compatible with AeroDyn version 12.3.

## 3.0 AeroDyn Files

Three files contain the AeroDyn source code. They are: the main subroutines of the software, AeroSubs.F90; a file containing subroutines that perform peripheral tasks, GenSubs.F90; and a file containing parameters used throughout the AeroDyn and dynamics codes, AeroMods.F90.

The primary data input file is called AERODYN.IPT. At least 2 other input files are required. The first is a wind file, containing either hub-height (steady or time varying) wind or full-field turbulence. The second file an airfoil data file (with a minimum of one for the entire blade, and a maximum equal to the number of elements in the AeroDyn model). The contents of each of these files are described in this guide.

The dynamics analysis code used with AeroDyn will have control over most of the outputs files created. However, the OPT file (e.g., yawdyn.opt or gfosub.opt) will include a nearly line-by-line review of aerodyn.ipt. If element output is selected in aerodyn.ipt, the element output file (e.g., element.plt or reqelem.plt – formerly aelement.plt) will be created. Other time series output files (e.g., yawdyn.plt and reqsub1.plt) are opened by AeroDyn to provide consistent program identifier headings. Errors and warnings encountered by AeroDyn will be logged in the error.log file.

## 4.0 Memory Requirements

Most arrays in AeroDyn are dynamically allocated to the model parameters (number of blades, elements, airfoil data points, etc.). By far the largest of these arrays are the full-field wind arrays, which are allocated to the time length of the simulation. If you encounter any trouble with RAM shortage, reduce the length of the simulation. If your computer is properly suited to run the dynamics code you are using, AeroDyn should have no trouble running as well.

## 5.0 Nomenclature and Sign Conventions

While AeroDyn is written to interact with many different dynamics routines, all of which have their own conventions and nomenclature, AeroDyn has its own conventions that must be followed. AeroDyn has to handle several parameters provided by the dynamics code. Most of these parameters are listed in the aerodyn.ipt file, and are discussed in Section 6. Other parameters provided by the wind files are discussed in sections 7 and 8. AeroDyn also handles parameters provided internally by the AeroDyn interface codes. Conventions for these are only of interest to those who work directly with the programming of the codes. These parameters are discussed in detail in Appendix C.

## 6.0 Input Data File Description

A sample aerodyn.ipt input data file is given in Table 6.1. Aerodyn.ipt is a text data file that must be present in the directory or folder from which the model is to be run. The formatting of this file is list-directed (or free). There are no restrictions on the spacing of the values other than the order of the variables on a line, the order of the lines, and the presence (absence) of a decimal point in a floating point (integer) value. If multiple values appear on one line, one or more spaces or tabs should separate them. Each line (except the first) can be terminated with a text string to identify that line. For lines that contain wind or airfoil file names, the file names must be contained in quotes. Each line must terminate with a return character. Each line must contain all of the variables specified for that line. Omission of a value that is not used in a particular simulation may not result in a runtime error, but the line on which that parameter should be located must be present.

A line-by-line description of the input data file is given in Table 6.2. Note that line 2 of this file allows the user to choose English or SI (metric) units for the model. Care must be taken to ensure consistency of units.

Table 6.1 - Sample AeroDyn Input Data File for the NREL Combined Experiment Wind Turbine

```
Combined Experiment Baseline for AeroDyn version 12.5
ENGLISH    Units for input and output [SI or ENGlish]
BEDDOES    Dynamic stall model [BEDDOES or STEADY]
USE_CM     Aerodynamic pitching moment model [USE_CM or NO_CM]
EQUIL      Inflow model [DYNIN or EQUIL]
SWIRL      Induction factor model [NONE or WAKE or SWIRL]
5.0000E-03 Convergence tolerance for induction factor
PRAND      Tip-loss model (for EQUIL only) [PRANdtl, GTECH, or NONE]
PRAND      Hub-loss model (for EQUIL only) [PRANdtl, or NONE]
"yawdyn.wnd"          Hub-height wind file name (quoted string)
55.0       Wind reference (hub) height.
0.1        Tower shadow centerline velocity deficit.
3.0        Tower shadow half width.
4.0        Tower shadow reference point.
2.0000E-03 Air density.
1.63e-4    Kinematic air viscosity
1.0000E-03 Time interval for aerodynamic calculations.
1          Number of airfoil files used. Files listed below:
"S809_Cln.dat"        Airfoil data file name (quoted string)
10         Number of blade elements per blade
RELM       Twist      DR         Chord      File ID    Elem Data
 0.7400       0.0000   1.4800     1.5000     1
 2.2200       0.0000   1.4800     1.5000     1         PRINT
 3.7000       0.0000   1.4800     1.5000     1
 5.1800       0.0000   1.4800     1.5000     1         PRINT
 6.6600       0.0000   1.4800     1.5000     1
 8.1400       0.0000   1.4800     1.5000     1         PRINT
 9.6200       0.0000   1.4800     1.5000     1
11.1000       0.0000   1.4800     1.5000     1         PRINT
12.5800       0.0000   1.4800     1.5000     1
14.0600       0.0000   1.4800     1.5000     1         PRINT
```

Table 6.2 - Descriptions of AeroDyn Input File Parameters

| ID Number[1] | Units[2] | Description |
|---|---|---|
| 1<br>TITLE | -- | Any character string (up to 80 characters) to identify the system being analyzed. This also serves as an aid to identifying the contents of the data file. |
| 2<br>SI or ENGLISH | -- | This input designates the system of units you are using for the input and output variables. If you enter SI then AeroDyn works in the SI system (Newtons, kilograms, meters, seconds and their combinations). For example, air density will be $kg/m^3$. If you enter ENGLISH then AeroDyn works in the English system (pound force, slugs, ft, seconds). In this case, air density will be in $slug/ft^3$. All angles, such as pitch, twist and angle of attack, are input and output in degrees regardless of the units system selected. Users must be certain that the units selected here are consistent with the units employed in their dynamics model. No units conversions are performed by AeroDyn. |
| 3<br>STEADY or BEDDOES | -- | This value determines whether the Beddoes-Leishman dynamic stall model will be used. Enter BEDDOES for this dynamic stall model or STEADY for quasi-steady airfoil characteristics. We recommend using the BEDDOES model in most situations. |
| 4<br>USE_CM or NO_CM | -- | This value controls the option of calculating aerodynamic pitching moment. Enter USE_CM if you want to calculate the pitching moment. Enter NO_CM if you wish to ignore the pitching moment calculation. If you enter USE_CM, then you must provide a column of pitching moment coefficients (CMs) in all of your airfoil data tables (see Table 9.1 for an example). If you enter NO_CM, then the CM values need not be present in your airfoil tables (but they can be present *provided you have only one airfoil data table in the file*). |
| 5<br>DYNIN or EQUIL | -- | This input controls the dynamic inflow option. When the value is DYNIN, a generalized dynamic wake (GDW) inflow model is used to calculate the induction factor. The direct calculation method of DYNIN is considerably faster than the iterative method of the EQUIL option. For more information on the dynamic inflow method, see Appendix E. A value of EQUIL assumes that the wake is always in equilibrium with the forces on a blade element (the "quasi-steady" or equilibrium wake assumption). For EQUIL, the BEM theory with skewed wake and tip loss corrections is used. |

[1] This column contains a sequential number, the ID number, and a name. The name represents the variable name in most cases. However, in the case of inputs that control a program option, the allowable inputs are listed.

[2] Units are specified for English system in the first line and SI units in the second line (if different). If SI units are desired the unit identifier (ID #2) must be SI.

| | | |
|---|---|---|
| 6<br><br>WAKE,<br>SWIRL or<br>NONE | -- | This value controls the wake or induced velocity calculation. There are three options, WAKE, SWIRL and NONE. This value should normally be SWIRL so that the axial and tangential induction will be analyzed. If WAKE is used, then only the axial induction will be calculated. If the value is NONE, then the induced velocity calculation will be completely bypassed and all induction factors will be zero. This option is available primarily to assist the debugging of new models. We suggest that the first tests of a new model for some dynamics routines (like ADAMS) ignore the wake to accelerate the calculations and eliminate the possibility of convergence problems in the induction factor iteration. A warning is generated when this value is NONE to remind the user this is a highly unusual situation. |
| 7<br><br>ATOLER | -- | The tolerance used for convergence testing in the iterative solution to find the induction factor, $a$. In earlier versions of the software, this value was always 0.005. This is a good default value that should be used unless there are compelling reasons to do otherwise. Some users may find it desirable to change this value to avoid convergence problems (with some loss of accuracy) or to speed the calculations. The value represents the maximum allowed difference between two successive estimates of A. That is, if the new estimate of A differs from the estimate from the previous iteration by an amount less than ATOLER, the solution has converged, and the last value of A is used. ATOLER is used for all induction factor calculations when using the EQUIL option (ID #5), but only for the trim solution in YawDyn when using the DYNIN option. |
| 8<br><br>PRAND,<br>GTECH or<br>NONE | -- | This value controls the tip loss model. There are three options, PRAND, GTECH and NONE. This value is only used with the EQUIL option (ID #5). PRAND is the Prandtl tip loss model, GTECH is the Georgia Tech correction to the Prandtl tip loss model. NONE turns off the tip loss correction altogether. The GTECH model is a new addition to version 12.4 of AeroDyn. It is intended to better model the effects of the relatively large inflow velocities (compared to hovering rotors, for which the Prandtl model was developed) experienced by wind turbine rotors spacing the tip vortex rings farther apart in the wake. |
| 9<br><br>PRAND, or<br>NONE | -- | This value controls the hub loss model. There are two options, PRAND, and NONE. This value is only used with the EQUIL option (ID #5). PRAND invokes the Prandtl tip loss model, to be used to determine hub losses. NONE turns off the hub loss correction altogether. This option is a new addition to version 12.50 of AeroDyn. It is intended to model losses experienced by the rotor blade by elements close to the rotor hub. Generally, these effects are of little consequence. |

| | | |
|---|---|---|
| 10<br><br>HHWindFile or FFWindFile or FDPFile | -- | A quoted string that represents the hub-height wind file name or the prefix for the full-field wind files, or the prefix for the 4-D wind parameter file. The filename can be up to 100 characters long (96 columns for the full-field prefix) and can include the full path. Any character in quotes will be considered part of the file name. Comments can be placed on the same line after and outside the quoted string.<br><br>AeroDyn automatically detects the type of wind file being used based on its name. It first assumes the string is the file name and searches for the file. If found, it assumes it is a hub-height file. If it fails to find it, it treats the string as if it were the full-field or 4-D file prefix, and searches for those files. Failing to find those as well aborts the simulation. AeroDyn requires a wind file to run. Hub-height wind files are discussed in section 7.0, and full-field turbulence files in section 8.0.<br><br>If full-field turbulence files are used, the string entered on this line of the aerodyn.ipt file is the leader or prefix that is common to two file names. It can be up to 96 characters long and *must be in quotes*. The filenames will end in `.wnd` and `.sum`. These files contain the full-field wind data in binary format, and a summary of the simulated turbulence, respectively. For example, consider the case of turbulence files named `9ms.wnd` and `9ms.sum`. If these files are in the 'myturb' directory on the C: drive, then the characters "`C:\myturb\9ms`" would be entered on this line and could remind the user that the mean wind speed for these files is 9 m/s. Even though two data files will be read, only one line is used to enter the file name prefix. See documentation from NREL concerning the turbulence data files.<br><br>For most users, a 4-D wind paraemeter file will never be used, but if it is used, the string entered on this line the prefix of a file that ends in .LEP. |
| 11<br><br>HH | ft<br><br>m | Wind reference (hub) height above the ground. This is the height of the center of the undeflected hub. When using hub-height wind files, this is used as the reference height to determine wind shear. When using full-field turbulence, it defines the height of the center of the turbulence grid. It is not equal to the tower height if the nacelle is tilted, nor does this value change with deflections of the dynamics model. When using a dynamics model with a ground level global coordinate system (such as ADAMS), this HH value is subtracted from the ground level coordinate height to determine the vertical location relative to hub-height used by AeroDyn. |
| 12<br><br>TWRSHAD | -- | A measure of the strength of the velocity deficit in the wake of the tower (tower shadow). The value is the magnitude of the fractional decrease in local wind speed at the center of the tower shadow. The deficit and the width (next line) are specified at the reference length T_Shad_Refpt (see below), which is essentially at the hub distance from the tower centerline. Typical values are 0.0 to 0.2. The value can be zero for an upwind rotor, unless downwind operation is anticipated. See Appendix A for more details. |
| 13<br><br>SHADHWID | ft<br><br>m | The half-width, b, of the tower shadow, measured at a distance T_Shad_Refpt (see ID #14) from the tower. The tower wake width increases as the square root of the distance from the tower, and the wake strength decays inversely proportional to the root of the distance. See Appendix A for more details. |

| | | |
|---|---|---|
| 14<br><br>T_Shad_Refpt | ft<br><br>m | A distance that indicates the point downstream of the tower centerline where the indicated tower shadow deficit strength (ID #12) is encountered. It is usually the distance from the yaw axis to the center of the hub. The value is always positive, and does not change with deflections of the dynamics model. In previous versions of AeroDyn, this also served as the rotor overhang (distance from yaw axis to hub) in YawDyn, but this is no longer the case. |
| 15<br><br>RHO | slug/ft$^3$<br><br>kg/m$^3$ | Ambient air density. |
| 16<br><br>KinVisc | ft$^2$/sec<br><br>m$^2$/sec | Kinematic air viscosity. This value is used in AeroDyn to calculate the local element Reynold's number, which can be used to move between multiple tables in the airfoil files (see ID # 22). |
| 17<br><br>DTAERO | sec | Time interval for aerodynamics calculations in some dynamics routines (such as ADAMS). Other programs (like YawDyn) ignore this value, but the line must be present in the data set. In some simulation programs, like ADAMS, the typical integration time step is quite small. The simulation can run faster and be more immune to numerical stability problems if a value for DTAERO is entered that is greater than the integration time step, but less than the time scale for the changes in aerodynamic forces. Typically, the aerodynamic forces should not be expected to change faster than the time it takes the blade to rotate 2-4°. For example, if a rotor runs at 30 rpm, it will take 0.02 sec for the blade to move 3.6°. If DTAERO is 0.02 sec, the aerodynamic calculations will repeat often enough to catch the true variations in loads, but several times slower than the integration time step.<br><br>As with other time-step controls in numerical integrators, it is best to experiment with a range of values to determine the maximum value that will give the same results as the smaller values. Use of a very sharp tower shadow will normally require use of a small value for DTAERO.<br><br>Since the integration time stepping is under the control of the dynamics program, the aerodynamics calculations will be repeated after a time interval that is *at least* DTAERO (but less than DTAERO+ the integration time step). That is, the calculations are generally not repeated exactly at the interval DTAERO. |
| 18<br><br>NUMFOIL | -- | The number of different airfoil files that will be used to describe the blade elements. There is no inherent limit to the number allowed in AeroDyn, as arrays are allocated at run-time as needed. |
| 19<br><br>FOILNM | -- | The name of the first data file (number 1) that contains the airfoil data. See the airfoil data file section below for a description of the contents of this file. *The filenames are limited to 80 characters and must be placed in quotes ("[FOILNM]").* Comments can be placed on the same line after and outside the quotes.<br><br>If multiple airfoil files are used, successive lines will provide the filename for each of the other airfoil data tables in the same manner. Only one filename is entered on each line. |

| | | |
|---|---|---|
| 20<br><br>NELM | -- | The number of blade elements per blade. There is no inherent limit to the number allowed in AeroDyn as arrays are allocated at run-time. |
| 21<br><br>RELM | ft<br><br>m | Following a header line, the element data table is listed on the subsequent NELM lines of the aerodyn.ipt file. Each line describes the details of each blade element for the aerodynamic analysis. Note that the blade elements <u>need not</u> be equally spaced, and they <u>must</u> be entered in order proceeding from inboard sections to outboard sections.<br><br>The first entry in each line specifies the location of the center of the blade element. RELM is measured from the blade root (flapping or teeter hinge axis) to the center of the element in the direction of the blade span. See Figure 6.1 for a sketch of the element geometry and nomenclature. It is used to convey the positions of the elements to AeroDyn, and specifies where the aerodynamic forces are applied in the dynamics model. <u>This value is ignored when running ADAMS</u> (or other program that defines element location inherently in its data set). The correct value is obtained from markers in the ADAMS data set. |
| 21<br><br>TWIST | deg | The second entry on each line is the twist of the blade element. The twist is measured relative to the plane of rotation. When combined with the pitch angle of the blade, it defines the angle the element's chord line makes with the plane of rotation. The twist affects only one blade element while the pitch changes the angles of all blade elements, just as a full-span pitch control system would. The sign convention for TWIST is the same as for PITCH (see Figure C2). TWIST is used to convey the orientations of the elements to AeroDyn, and specifies the orientations of the aerodynamic forces applied in the dynamics model. Like RELM, <u>the TWIST value is ignored when running ADAMS</u> (or other program that defines element orientation inherently in its data set). |
| 21<br><br>DR | ft<br><br>m | The third entry on each line is the span-wise width of the blade element, measured along the span of the blade (see Figure 6.1). Use care selecting DR and RELM if you are not using equally spaced elements. The two values must be compatible, but AeroDyn does not test fully for compatibility or contiguity of the elements. |
| 21<br><br>CHORD | ft<br><br>m | The fourth entry on each line is the chord of the blade element. The planform area of the blade element equals CHORD*DR. |
| 21<br><br>NFOIL | | The fifth entry on each line is an integer between one and NUMFOIL (ID #18) that determines which airfoil data file is to be used for each blade element. The first airfoil file listed in the aerodyn.ipt is number 1, the second number 2, etc. If a value of NFOIL=1, then the blade element will use data from the first airfoil file. If NFOIL=3, the airfoil data will be read from the third filename, etc. |

| | | |
|---|---|---|
| 21<br><br>PRINT or NOPRINT | -- | The sixth and final entry on each line is an optional string flag to control whether output is written for that element to the element file (element.plt for YawDyn, reqelem.plt for ADAMS, [rootname].elm for FAST). If the string PRINT is found, output is written for that element. If PRINT is not found (or NOPRINT is found), element data is not written. AeroDyn responds to whichever string occurs first on each line. If no string is found on a line, no data is written for that element. The element file is not created if no element data is requested (i.e., PRINT is not found on any line). This file is useful when debugging the aerodynamics of a model. It contains data such as angle of attack, aerodynamic coefficients and forces, and induction factors for the selected elements. The output frequency to the element file will coincide with the frequency requested for other time series data in the dynamics simulation. |
| 22<br><br>SINGLE or ReNum or USER | -- | This value is optional, and will likely be missing for most models (as it is in the example of Table 6.1). It determines whether you will use a SINGLE airfoil table (or single interpolation point between two airfoil tables), or multiple airfoil tables (or multiple interpolation points between tables) during the simulation. If your airfoil files have only one table in them, this line is ignored. You should use SINGLE if you want to use a single table from a file containing multiple tables (or if you want to use a single interpolation point between two tables) for the duration of the simulation. Use ReNum if you want to move between multiple airfoil tables in your files based on Reynold's Number. AeroDyn is set up to handle those calculations, but be sure to provide an accurate viscosity (ID #16). Use USER only if you supply your own algorithm in the dynamics code to move between airfoil tables during a simulation. Most AeroDyn users will not need this option. It is useful for those who wish to implement control algorithms, or investigate roughness or Reynolds number effects in AeroDyn. |

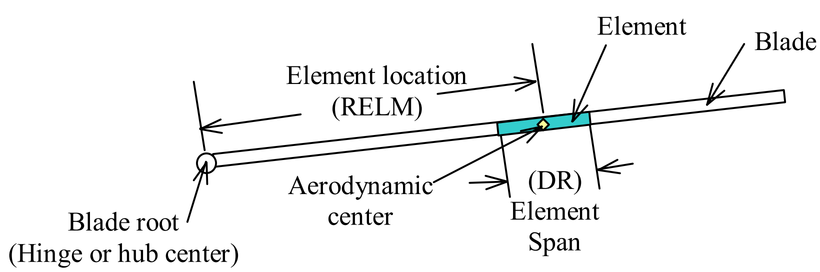



Figure 6.1 Sketch of the blade element geometry and nomenclature.

## 7.0 The Hub-Height Wind Data File

In order to run a simulation using AeroDyn, a wind input file is required. There are 2 types of wind input files. This section will discuss hub-height (HH) wind files. Section 8.0 addresses full-field (FF) turbulence files. The wind file name appears on line 8 of the aerodyn.ipt file. The name can be up to 100 characters long.

The HH wind file option in AeroDyn is flexible in that it allows several different parameters to be updated throughout the simulation, though re-programming is required to exercise some of these. How to reprogram for other parameters is described in Section 7.1. The file can also be used to simulate steady wind conditions by simply limiting the data to one line.

In the release version of AeroDyn, the values of horizontal wind speed at the hub (V), wind direction (DELTA), vertical wind speed (VZ), horizontal wind shear (HSHR), power law vertical wind shear (VSHR), linear vertical wind shear (VLinShr) and gust velocity (VG) are entered in tabular form as a function of time (TDAT). A sample HH wind file is presented in Table 7.1. A description of each column in the hub-height wind file is provided in Table 7.2. (More or fewer parameters can be programmed by the user – see section 7.3).

The first lines of the file can be "comment" lines. A comment line must appear before any data in the file and it must contain the ! character somewhere in the line (usually at the beginning, but this is not required). Any number of comment lines can be used at the beginning of the file, but no comments can be embedded within the wind data lines.

One or more spaces or tabs must separate the data on each line of the hub-height wind file. All values for each time in the file (TDAT) must be on the same line, and each line must end with a return character.

The time step between lines need <u>not</u> be constant. Linear interpolation is used for simulated time that is between two values of TDAT in the wind data file. Thus, a simulation with linearly changing wind direction, for example, would need only two lines regardless of the length of the simulation.

If a simulation runs longer than the time length of the wind file, the simulation continues using the last line of the wind file as a steady wind condition for the remainder of the simulation. AeroDyn generates a warning indicating when this has occurred.

## 7.1 Hub-height wind file conventions

Of the 8 wind parameters listed in Table 7.2, the direction and shears are the only ones whose conventions are of concern. Wind direction ($\delta$) is positive clockwise looking down, as shown in Figure 7.1. This is the same convention used for yaw angle ($\gamma$). This is consistent with compass directions generally used in reporting wind direction. Using this convention, the yaw error (or difference between the compass rotor direction and wind direction) is $\gamma - \delta$.

Vertical and horizontal wind shears can be specified in the hub-height wind file. Figure 7.1 shows a sketch of the wind shear conventions. A positive vertical wind shear (whether linear or power-law – both are available for use in the wind file) causes an increase in wind speed with height above ground. A positive horizontal wind shear causes an increase in wind speed to the left looking downwind.

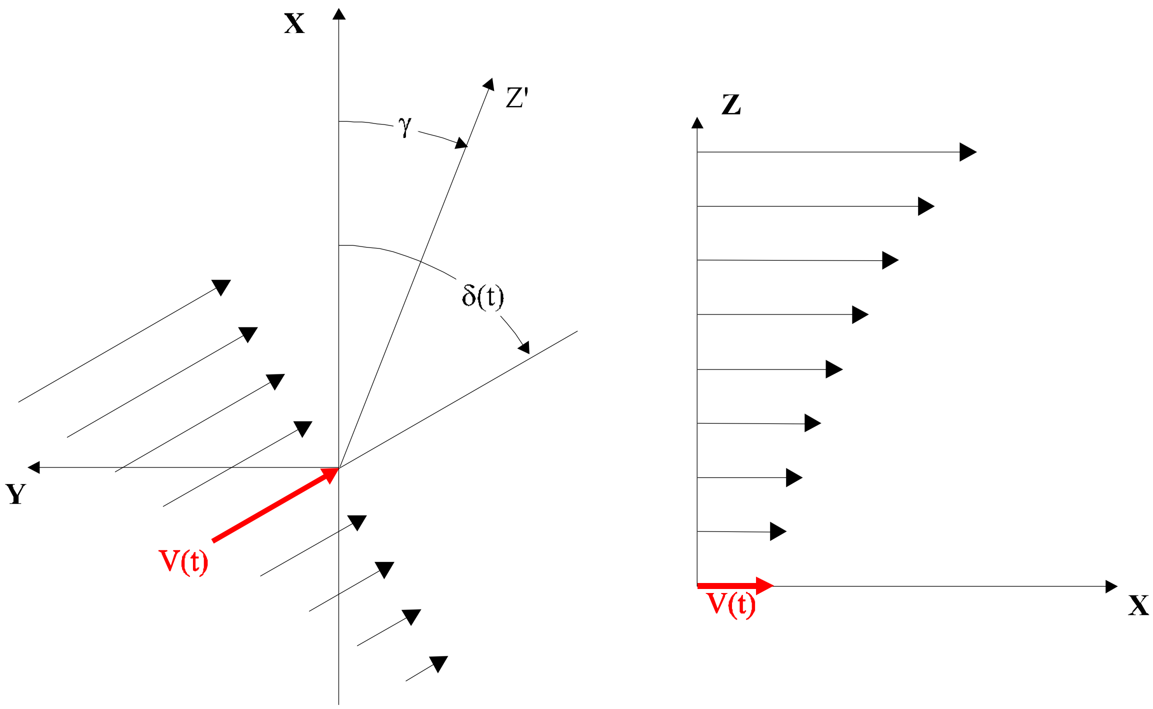



Figure 7.1 Wind shear and direction models: Overhead view of horizontal shear in left sketch; Cross-section view of vertical shear in right sketch. Note the wind direction (δ) and yaw angle (γ) are both defined with respect to the **X** axis (global direction indicating zero yaw and wind direction angles). All directions and shears are shown in their positive sense.

## 7.2 Simulating IEC wind conditions

A utility program named IECWind is available to generate hub-height wind files for AeroDyn that model the discrete gust conditions specified by the 2nd edition IEC wind turbine design standard. (Note that some of the IEC extreme wind conditions have changed from the 1st to the 2nd edition standard.) Be sure to use the latest version of IECWind (for YawDyn 11.0 and later), as earlier versions generate formats that are no longer compatible with the latest AeroDyn. There is also a Windows version of the IECWind code called WindMaker. Again, be sure to get the latest version (for YawDyn 11.0 and later). IECWind or WindMaker can be quite helpful when analyzing the variety of gust conditions that must be considered during turbine design.

## 7.3 Reprogramming AeroDyn to read other wind file parameters

By making relatively simple changes to the source code, users can modify the list of parameters that will be read from the hub-height wind file. Fortran READ statements which access this file are found in the subroutine GetHHWind. All statements are of the form:

```
 READ(91,*) TDATD (2), VD (2), DELTAD (2), VZD(2), HSHRD(2), VSHRD(2), VLinShrD (2), VGD
(2)
```

If desired, the list of variables can be shortened or extended to meet particular requirements. All that is required is 5 simple changes:

1. Create a dummy argument array of size 2 for the desired parameter.
2. Add the parameter to all Read statements in the proper position.

3. Replace the old values for new at the appropriate time.
4. Add an interpolation on the dummy arguments to find the correct value of the updated parameter.
5. Update the variable if the wind file runs out before the end of the simulation.

For example to add pitch angle to the file these steps would be:

1. Define the array:

```
REAL, SAVE :: PitchD(2,NB)
```

   The SAVE statement is to retain values read for subsequent interpolations on it, and the array itself is allocated space for the two dummy values to interpolate on as well as space for each blade (NB).

2. Rewrite the READ statements as:

```
READ(91,*) TDATD(2), VD(2), DELTAD(2), …, (PitchD(2,I), I=1,NB)
```

   This change must be made in three locations in GetHHWind. Once for PitchD(1, I) and twice for PitchD(2, I).

3. Before reading in subsequent values of PitchD(2, I), PitchD(1,I) must be updated by adding the loop:

```
DO I=1,NB
   PitchD(1,I) = PitchD(2,I)
END DO
```

4. Then, the interpolation on PitchD is added as:

```
DO I=1,NB
   Pitch(I) = PitchD(1,I) + P * (PitchD(2,I) - PitchD(1,I))
END DO
```

   The value of P is already set as the interpolation value.

5. Finally, if the wind file ends before the simulation, the final value in the wind file is updated as:

```
DO I=1,NB
   Pitch(I) = PitchD(2,I)
END DO
```

If the pitch angles are entered in the wind file in degrees, they should be converted to radians with the statement:

```
PitchD = PitchD * DtoR
```

added after each read statement. DtoR is the conversion factor to convert degrees to radians in AeroDyn.

Table 7.1 – Sample hub-height Wind File

```
! Sample hub-height wind file for AeroDyn
! Time  Wind     Wind     Vert.    Horiz.   Vert.    LinV     Gust
!       Speed    Dir      Speed    Shear    Shear    Shear    Speed
0       15.000   5.000    -1       0.020    0.14     0        0
0.1     16.545   4.755    -0.9     0.022    0.14     0        0
0.2     17.939   4.045    -0.8     0.024    0.14     0        0
0.3     19.045   2.939    -0.7     0.027    0.14     0        0
0.4     19.755   1.545    -0.6     0.030    0.14     0        0
0.5     20.000   0.000    -0.5     0.033    0.14     0        0
0.6     19.755   -1.545   -0.4     0.036    0.14     0        0
0.7     19.045   -2.939   -0.3     0.040    0.14     0        0
0.8     17.939   -4.045   -0.2     0.045    0.14     0        0
0.9     16.545   -4.755   -0.1     0.049    0.14     0        0
1       15.000   -5.000   0        0.054    0.14     0        0
1.1     13.455   -4.755   0.1      0.060    0.14     0        0
1.2     12.061   -4.045   0.2      0.066    0.14     0        0
1.3     10.955   -2.939   0.3      0.073    0.14     0        0
1.4     10.245   -1.545   0.4      0.081    0.14     0        0
1.5     10.000   0.000    0.5      0.090    0.14     0        0
1.6     10.245   1.545    0.6      0.099    0.14     0        0
1.7     10.955   2.939    0.7      0.109    0.14     0        0
1.8     12.061   4.045    0.8      0.121    0.14     0        0
1.9     13.455   4.755    0.9      0.134    0.14     0        0
2       15.000   5.000    1        0.148    0.14     0        0
```

Table 7.2 – Hub-height File Column Descriptions

| Column | Parameter | Units | Description |
|---|---|---|---|
| 1 | TDAT | sec | Time at which the conditions on the current line are specified to occur. The first value should be zero, with subsequent values increasing monotonically. Intervals between time values need not be constant. Wind conditions between specified TDAT values are linearly interpolated. If the simulation time duration exceeds the last value of TDAT, the final value of each parameter is held constant for the remainder of the simulation. |
| 2 | V | m/s<br>ft/s | This hub-height wind speed represents the total horizontal wind component. Units must be consistent with the selection made in aerodyn.ipt on line 2. |
| 3 | DELTA | deg | The wind direction of the horizontal component specified above, with zero aligned with the zero yaw angle (see Figure 7.1). |
| 4 | VZ | m/s<br>ft/s | The vertical wind speed component is specified with the convention positive up. This value is assumed uniform over the rotor disc (i.e. it is unaffected by any specified shear values). |
| 5 | HSHR | -- | The horizontal wind shear parameter represents the linear variation of wind speed across the rotor disc. Typical values are –1. < HSHR < +1, and represent the wind speed at the blade tip on one side of the rotor, minus the wind speed at the blade tip on the opposite side of the rotor, divided by the hub-height wind speed (V). The shear is measured in the direction perpendicular to the hub-height wind vector specified by DELTA above. See Figure 7.1 for sign convention. |

| | | | |
|---|---|---|---|
| 6 | VSHR | -- | The vertical power law shear is the exponent of a power-law shear profile. It is used to determine the wind speed, $V_z$, at any height, z, based on the hub-height, $z_{hub}$ (see line 9 of aerodyn.ipt), and hub-height wind speed, $V_{hub}$, using the equation: $V_z = V_{hub} (z/z_{hub})^{VSHR}$ A typical value is 0.14 representing a 1/7[th] power-law profile. Normally you should use either VSHR or VLinSHR (see next), not both (set the parameter you do not wish to use to zero). |
| 7 | VLinSHR | -- | The linear vertical shear parameter works in the same way as HSHR but in the vertical plane across the rotor disc. It represents the wind speed at the blade tip at the top of the rotor, minus the wind speed at the blade tip at the bottom of the rotor, divided by the hub-height wind speed (V). Normally you should use either VLinSHR or VSHR (above), not both (set the parameter you do not wish to use to zero). |
| 8 | VG | m/s<br>ft/s | The gust velocity is a parameter used to add a constant horizontal wind speed component across the entire rotor disc. <u>This parameter is not influenced by any shear values.</u> This parameter is seldom used and can be set to zero for most cases. Changes in wind speed are normally specified using the hub-height wind speed parameter, V. |

## 8.0 The Full-Field Turbulence Wind Data Files

The other wind input option in AeroDyn is simulated full-field wind data that represent all three components of the wind vector varying in space and time. This permits a detailed simulation of a wind field with the appropriate scales and correlation of atmospheric turbulence. Two files, one binary wind data file and one summary file, must be in the specific form generated by the NREL program SNLWIND-3D or SNWind. These programs are available from the NREL web site.

The components of the wind vector are expressed in the inertial coordinate system that has its origin on the undeflected yaw axis, at the undeflected hub height of the rotor. See Figure C1 to see the location of this **XYZ** coordinate system. A grid of fixed points (much like a vertical plane array of anemometers) is located in the **YZ** plane and centered at the hub height. The three velocity components are available at each grid point as a function of time. The subroutines interpolate in all three spatial dimensions (using a convection velocity to get a time shift for the **X** dimension) to obtain the wind velocity vector at each blade element at each time step.

AeroDyn reads and stores the turbulence files into memory at the start of execution. The total number of samples that can be stored, and hence the time duration that can be simulated, is determined by the length simulated in the full-field files. See section 4.0 for information on memory requirements for AeroDyn if you experience trouble running long turbulence simulations.

In earlier versions of the code a bicubic interpolation routine slowed the simulations considerably. In version 9.1 and later the interpolation is linear, yielding simulations with turbulence nearly as fast as those with steady wind and giving the same results as the much slower bicubic interpolation method.

## 9.0 The Airfoil Data Files

For each element in the AeroDyn model, a different airfoil data file can be used to represent the aerodynamic properties. Each airfoil file can contain as many angle-of-attack entries for each of up to 10 (or MAXTABLE) tables. A sample data file is shown in Table 9.1. A line by line description of this file is presented in Table 9.2. Each different airfoil section is described in a separate data file as identified in the aerodyn.ipt file. The following table illustrates the format of the input data. All data are free-format. Comments can be included at the end of any line to serve as a reminder of the contents of that line. Any line that is for use by the dynamic stall model must be present even if dynamic stall is not considered in the simulation.

Appendix D describes a utility program called FoilCheck that may help with the preparation of airfoil data files.

### 9.1 Multiple Airfoil Tables

AeroDyn allows the user to use multiple airfoil tables for any given element. AeroDyn moves between these tables based on the value of the parameter chosen, and linearly interpolates for points between the tables. AeroDyn is set up to handle the case for Reynold's number, but the user may select any parameter (such as aileron angle), as long as the value is provided to AeroDyn for each aerodynamic calculation. These values will be passed into AeroDyn via the dynamics code through the GetElemParams routine arguments.

If ReNum is used to move between airfoil tables, your multiple tables in the airfoil files should each be labeled as *Reynold's number x$10^6$*. AeroDyn works internally with Reynold's number x$10^6$.

If AeroDyn encounters a value that lies outside the table limits, the table limits are used. For example, suppose you provide AeroDyn with tables based on Reynold's numbers from 0.5x$10^6$ to 1.0x$10^6$. If AeroDyn calculates Reynold's numbers below 0.5x$10^6$, it will use the table values for 0.5x$10^6$. Likewise if Reynold's numbers above 1.0x$10^6$ are encountered, the table values for 1.0x$10^6$ are used. This prevents both the need to bracket all possible values that may be encountered, and the possibility of large errors due to extrapolation far beyond the tabulated limits.

Table 9.1 - Sample Airfoil Data File for the NREL Combined Experiment Wind Turbine

```
S809 Airfoil, OSU data at Re=.75 Million, Clean roughness
NREL/TP-442-7817 Appendix B, Viterna used aspect ratio=11
   1           Number of airfoil tables in this file
      .00      Table ID parameter
      .00      No longer used, enter zero
      .00      No longer used, enter zero
      .00      No longer used, enter zero
      .00      No longer used, enter zero
     -.38      Zero lift angle of attack (deg)
    7.12499    Cn slope for zero lift (dimensionless)
     1.9408    Cn at stall value for positive angle of attack
     -.8000    Cn at stall value for negative angle of attack
     2.0000    Angle of attack for minimum CD (deg)
      .0116    Minimum CD value
  -180.00   .000  .1748  .0000
  -170.00   .230  .2116  .4000
  -160.00   .460  .3172  .1018
  -150.00   .494  .4784  .1333
  -140.00   .510  .6743  .1727
  -130.00   .486  .8799  .2132
  -120.00   .415 1.0684  .2498

   portion deleted for brevity, see the sample file on the distribution disk

    -8.20  -.560  .0233 -.0051
    -6.10  -.640  .0131  .0018
    -4.10  -.420  .0134 -.0216
    -2.10  -.210  .0119 -.0282
      .10   .050  .0122 -.0346
     2.00   .300  .0116 -.0405
     4.10   .540  .0144 -.0455
     6.20   .790  .0146 -.0507
     8.10   .900  .0162 -.0404
    10.20   .930  .0274 -.0321
    11.30   .920  .0303 -.0281
    12.10   .950  .0369 -.0284
    13.20   .990  .0509 -.0322
    14.20  1.010  .0648 -.0361
    15.30  1.020  .0776 -.0363
    16.30  1.000  .0917 -.0393
    17.10   .940  .0994 -.0398
    18.10   .850  .2306 -.0983
    19.10   .700  .3142 -.1242
    20.10   .660  .3186 -.1155
    30.00   .705  .4784 -.2459
    40.00   .729  .6743 -.2813
    50.00   .694  .8799 -.3134
    60.00   .593 1.0684 -.3388
    70.00   .432 1.2148 -.3557
    80.00   .227 1.2989 -.3630
    90.00   .000 1.3080 -.3604
   100.00  -.159 1.2989 -.3600
   110.00  -.302 1.2148 -.3446
   120.00  -.415 1.0684 -.3166
   130.00  -.486  .8799 -.2800
   140.00  -.510  .6743 -.2394
   150.00  -.494  .4784 -.2001
   160.00  -.460  .3172 -.1685
   170.00  -.230  .2116 -.5000
   180.00   .000  .1748  .0000
```

Table 9.2 - Descriptions of Airfoil Data File Parameters

| Line | Position | Name | Units | Description |
|---|---|---|---|---|
| 1 | 1 | TITLE(1) | -- | Up to 40 characters of text to identify this data file. This title will be written to the screen when AeroDyn executes to remind the operator which airfoil tables are being used. |
| 2 | 1 | TITLE(2) | -- | Up to 40 characters of any text to identify this data file. |
| 3 | 1 | NTables | -- | The number of different airfoil tables contained in this file. If you wish to simulate ailerons or Reynold's number dependency, for example, this value represents the number of aileron angle settings or Reynold's numbers for which aerodynamic coefficient data are provided. |
| 4 | 1... NTables | MulTabMet | -- | The parameter that identifies each airfoil table (the multiple airfoil table metric). Examples include aileron angle, Reynolds number, etc. Linear interpolation is done between tables based upon the value of the table ID that is desired. If using the ReNum option in AeroDyn for Reynold's number, the values here need to be Reynold's number x$10^6$ (i.e., a Reynold's number of 1 million would be listed as 1.0) |
| 5 | -- | | -- | Reserved for future use and backward compatibility. AeroDyn does not use this value, but the line must be present in the file. |
| 6 | -- | | -- | Reserved for future use and backward compatibility. AeroDyn does not use this value, but the line must be present in the file. |
| 7 | -- | | -- | Reserved for future use and backward compatibility. AeroDyn does not use this value, but the line must be present in the file. |
| 8 | -- | | -- | Reserved for future use and backward compatibility. AeroDyn does not use this value, but the line must be present in the file. |

The next few lines of input pertain to the dynamic and static stall characteristics of the airfoil for use in the dynamic stall models. They must be present, though the values will be ignored, when the dynamic stall option is not selected.

| Line | Position | Name | Units | Description |
|---|---|---|---|---|
| 9 | 1... NTables | ALPHAL | deg | The zero-lift angle-of-attack of the airfoil. This parameter is used only in the Beddoes dynamic stall model. |

| | | | | |
|---|---|---|---|---|
| 10 | 1... NTables | CNA | -- | The static $C_N$ (approximately equal to $C_L$) curve slope near zero lift. This dimensionless value is critical to the success of the Beddoes model and must be consistent with the tabular data that follow. We recommend using a least-squares fit to the linear portion of the $C_N$ data to determine this value. An "eyeball" fit is not accurate enough in most situations. This value is used only by the Beddoes dynamic stall model. This value is dimensionless ($\Delta C_N$/radian). |
| 11 | 1... NTables | CNS | -- | The value of $C_N$ at positive static stall. This is the nominal stall for positive and increasing angles-of-attack. This value is typically 1.0-3.0 and occurs at angle-of-attack near 15° to 30° (the higher values may be observed inboard on a rotating blade). This value is used only by the Beddoes dynamic stall model. We have found better correlation with test results when we use the value of CN extrapolated from the linear portion of the CN curve to the stall angle ALPHAS. This gives better results than using the actual CN value at the stall angle. |
| 12 | 1... NTables | CNSL | -- | The value of $C_N$ at the negative static stall angle of attack. This is "stall" for negative and decreasing angles-of-attack. This value is typically -1.0 and occurs at angles-of-attack near -10° or -20°. This value is used only by the Beddoes dynamic stall model. |
| 13 | 1... NTables | AOD | deg | The angle of attack for the minimum drag coefficient ($C_{Dmin}$). This value is used only by the Beddoes dynamic stall model. |
| 14 | 1... NTables | CDO | -- | The minimum drag coefficient of the airfoil. This value is used only by the Beddoes dynamic stall model. |
| 15+ | 1 | AL | deg | The angle of attack for the first point in the lift & drag coefficient table. The table must be written in order of increasing angle of attack. It must cover the entire range of angles of attack that might be encountered by any blade element. It is preferable to use a table for angles between -180° and +180°. If AeroDyn attempts to find a value outside the table range, program execution will stop with an error message. Note that inboard blade elements can easily encounter angles of attack approaching ±180° if the rotor is operating at a large yaw angle. Care should also be taken to ensure that the values of the coefficients are the same at +180° and -180° to avoid a discontinuity.<br><br>AeroDyn obtains all airfoil data by linear interpolation from the tables provided. The user must be certain that adequate resolution is available in the table to make linear interpolation accurate. The points need not be equally spaced, so it is advisable to enter many points near stall and fewer points at very large (positive or negative) angles. |

| | | | | |
|---|---|---|---|---|
| 15+ | 2, 4, 6, .. or 2, 5, 8, .. | CL | -- | The static lift coefficient corresponding to the angle of attack entered on this line. Many lines such as this are entered to completely specify the lift coefficient vs. angle-of-attack curve. |
| 15+ | 3, 5, 7, .. or 3, 6, 9 … | CD | -- | The drag coefficient corresponding to the angle of attack entered on this line. Note the CL, CD and CM (if used) must be specified for the same angles of attack. |
| 15+ | 4, 7, 10, .. or not present | CM | -- | The pitching moment coefficient corresponding to the angle of attack entered on this line. Note the CL, CD and CM (if used) must be specified for the same angles of attack. This value must be present if the USE_CM option is selected. It can, but need not, be present and will be ignored if the NO_CM option is selected for a single-table airfoil file.<br><br>*If you are using multiple tables in your airfoil file, then you must not have CM values in the table if you have selected the NO_CM option.* |

## 10.0 AeroDyn Outputs

AeroDyn is written so that no user input is required while it is running. However, since AeroDyn is invoked by a dynamics analysis, user interaction during the simulation is dependent upon the dynamics analysis program used.

AeroDyn does log errors and warnings in the error.log file, and print these to the screen as well. Among the other outputs printed to the screen directly by AeroDyn are:

- The heading of the aerodyn.ipt file
- Whether a hub-height or full-field wind file was found

Other screen outputs will depend upon the dynamics code used.

The aerodyn.ipt file inputs are output as part of an OPT file that is written by the dynamics code. The name of this file will depend on the dynamics code used (e.g., yawdyn.opt for YawDyn, gfosub.opt for ADAMS).

### 10.1 The element output file

AeroDyn creates only one output file, the element file, which is optional. Other output files are controlled by the dynamics routines. The element file is a tab-delimited time-series file containing wind and aerodynamic data for the elements selected in the aerodyn.ipt file. It has 3 header lines: 1) the program identification and creation time stamp, 2) the column headings, and 3) the column units. The data follows beginning on line 4.

The data output in the element file, along with descriptions, are listed in Table 10.1. These vary slightly whether you choose USE_CM in the aerodyn.ipt file or not. The column numbers if USE_CM is selected are presented in parentheses next to the column number if NO_CM is chosen. The first 4 columns are always present in the element file. If you choose USE_CM, then there are 13 columns for each element for which you request data, and 11 columns for each element if you choose NO_CM.

The "#" symbol is used where the element number would be in the column heading. The "(NELM)" used in the first 3 column headings here would be replaced by the actual most outboard element number in the element file.

Table 10.1 - Descriptions of Element Output File Columns

| Column | Heading | Units | Description |
|---|---|---|---|
| 1 | TIME | sec | The time during the simulation that the data on this line of the file represents. |
| 2 | VX(NELM) | m/sec<br>ft/sec | The global X velocity of the wind at the most outboard element location. This includes all shear and tower shadow effects, but <u>no induced velocity effects</u>. |
| 3 | VY(NELM) | m/sec<br>ft/sec | The global Y velocity of the wind at the most outboard element location. This includes all shear and tower shadow effects, but <u>no induced velocity effects</u>. |
| 4 | VZ(NELM) | m/sec<br>ft/sec | The global Z velocity of the wind at the most outboard element location. This includes all shear and tower shadow effects, but <u>no induced velocity effects</u>. |
| 5 | Alpha# | deg | The element angle of attack. |
| 6 | DynPres# | $N/m^2$<br>$lbf/ft^2$ | The dynamic pressure on the element, equal to $\frac{1}{2}\rho V^2$, where $\rho$ is the air density defined in aerodyn.ipt (RHO), and V is the total velocity at the element (including effects of rotor induced velocity). |
| 7 | CLift# | -- | The element coefficient of lift. |
| 8 | CDrag# | -- | The element coefficient of drag. |
| 9 | CNorm# | -- | The element normal (to the chord line) coefficient = CLift * cos(Alpha) + CDrag * sin(Alpha). |
| 10 | CTang# | -- | The element tangential (to the chord line) coefficient = CLift * sin(Alpha) - CDrag * cos(Alpha). |
| - (11) | CMom# | -- | The element coefficient of pitching moment. |
| 11 (12) | Pitch# | deg | The element pitch angle. A combination of the element twist angle and the blade pitch angle. It is the angle the element chord to the plane of rotation. The convention is positive toward feather as shown in Figure C2. |
| 12 (13) | AxInd# | -- | The induction factor on the element normal to the rotor plane. Always a value between 0 and 1. It represents a fractional drop in inflow to the rotor of the wind velocity component normal to the plane of rotation (or the blade span to be precise). |
| 13 (14) | TanInd# | -- | The induction factor on the element tangential to the rotor plane. Always a value between 0 and 1. It represents a fractional increase in inflow to the rotor of the wind velocity component tangential to the plane of rotation. |

| | | | |
|---|---|---|---|
| 14 (15) | ForcN# | N<br><br>lbf | The aerodynamic force on the element normal to the plane of rotation (or blade span), not the chord line (see Figure C2). Note this is the aerodynamic force only (no inertial forces), and is the force passed back to the dynamics routine by AeroDyn. |
| 15 (16) | ForcT# | N<br><br>lbf | The aerodynamic force on the element tangential to the plane of rotation (or blade span), not the chord line (see Figure C2). Note this is the aerodynamic force only (no inertial forces), and is the force passed back to the dynamics routine by AeroDyn. |
| - (17) | Pmomt# | N-m<br><br>ft-lbf | The aerodynamic pitching moment on the element. A positive pitching moment causes a negative pitching moment (see Figure C2). Note this is the aerodynamic moment only (no inertial forces), and is the value passed back to the dynamics routine by AeroDyn. |
| 16 (18) | ReNum# | ($x10^6$) | The element Reynold's number based on the chord and relative absolute local velocity. Note the numbers output are in millions ($x10^6$). These values are only used in AeroDyn when the ReNum option is selected for moving between multiple airfoil data tables based on Reynold's number. |

## References


Hansen, A. C., 1992, "Yaw Dynamics of Horizontal Axis Wind Turbines: Final Report." National Renewable Energy Laboratory, NREL Technical Report, TP 442-4822.

Hansen, A. C., C. P. Butterfield and X. Cui, 1990, "Yaw Loads and Motions of a Horizontal Axis Wind Turbine." Journal of Solar Energy Engineering. Vol. 112, No. 4. pp. 310-314.

Hansen, A. C. and X. Cui, 1989, "Analyses and Observations of Wind Turbine Yaw Dynamics." ASME Journal of Solar Energy Engineering. Vol. 111, No. 4. pp. 367-371.

Hansen, A. C. and A. D. Wright, 1991, "A Comparison of Combined Experiment Flap Load Predictions by the FLAP and YawDyn Codes." Tenth ASME Wind Energy Symposium, Houston, TX.

Hansen, A. C. 1993. "Structural Response to Unsteady and Stall Aerodynamics." 1993 DOE Wind Energy Program Review Meeting, Golden, CO.

Hansen, A. C., Butterfield, C. P., 1993. Aerodynamics of Horizontal-Axis Wind Turbines. Annual Review of Fluid Mechanics, 1993. Palo Alto, Annual Reviews, Inc.

Laino, D., Butterfield, C. P. 1994. "Using YawDyn to Model Turbines with Aerodynamic Control Systems." ASME Wind Energy Conference, New Orleans, LA.

Leishman, J. G., Beddoes, T. S., 1989. A Semi-Empirical Model for Dynamic Stall. Journal of the American Helicopter Society. 34(3): 3-17.

Malcolm, D. J., Wright, A. D. 1994. "The Use of ADAMS to Model the AWT-26 Prototype." ASME Wind Energy Symposium, New Orleans.

Peters, D. A., HaQuang, N., 1988. Dynamic Inflow for Practical Applications. Journal of the American Helicopter Society. 33(4): 64-68.

Xu, G. and Sankar, L.N., 2002. "Application of a Viscous Flow Methodology to the NREL Phase VI Rotor." AIAA-2002-0030, Proceedings of the 21 ASME Wind Energy Conference, Reno, NV, pp 83 – 93.

# Appendix A. Tower Shadow Model

The tower shadow model has been improved from the model described in the references at the end of this guide. A schematic of the tower wake is shown in Figure A1. The wake is symmetric about its centerline. It is assumed to align with the instantaneous horizontal wind vector. (This assumption will be improved in a future revision by aligning the wind with a short-term average wind direction.) The velocity deficit in the wake is of the form

$$\text{deficit} = \begin{cases} u_1 \cos^2(\dfrac{\pi d}{2b}) & |d| \le b \\ 0 & |d| > b \end{cases}$$

and the horizontal components of wind speed are reduced by the deficit fraction:

$$VX = (1 - \text{deficit})VX_\infty$$
$$VY = (1 - \text{deficit})VY_\infty$$

Here b is the wake half-width, $u_1$ is the centerline deficit, and d is the perpendicular distance from the wake centerline to the point in question. The deficit is applied to the ambient horizontal wind rather than to the wind after it has been modified by the induced velocity of the rotor. Note the tower shadow has no effect on the vertical component of wind speed.

The width of the wake increases as the square root of the distance from the tower:

$$b = b_{ref}\sqrt{\frac{\iota}{\iota_{ref}}}$$

Where $\iota$ is the streamwise distance from the tower. The centerline deficit decays with streamwise distance according to the function

$$u_1 = u_{1\,ref}\sqrt{\frac{\iota_{ref}}{\iota}}$$

The reference length, $\iota_{ref}$, is typically the distance from the yaw axis to the hub. In the AeroDyn code it is represented by the absolute value of the FORTRAN variable T_Shad_Refpt.

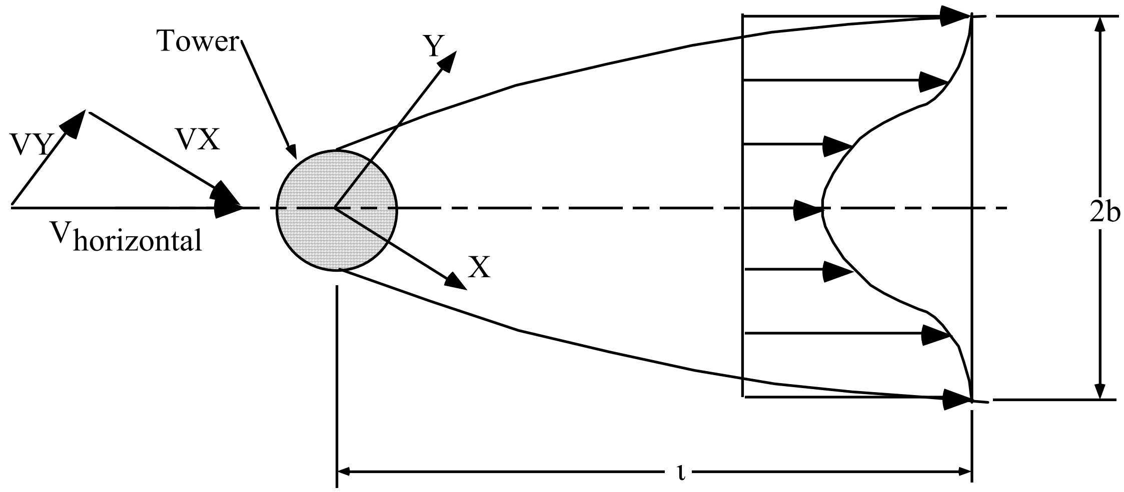


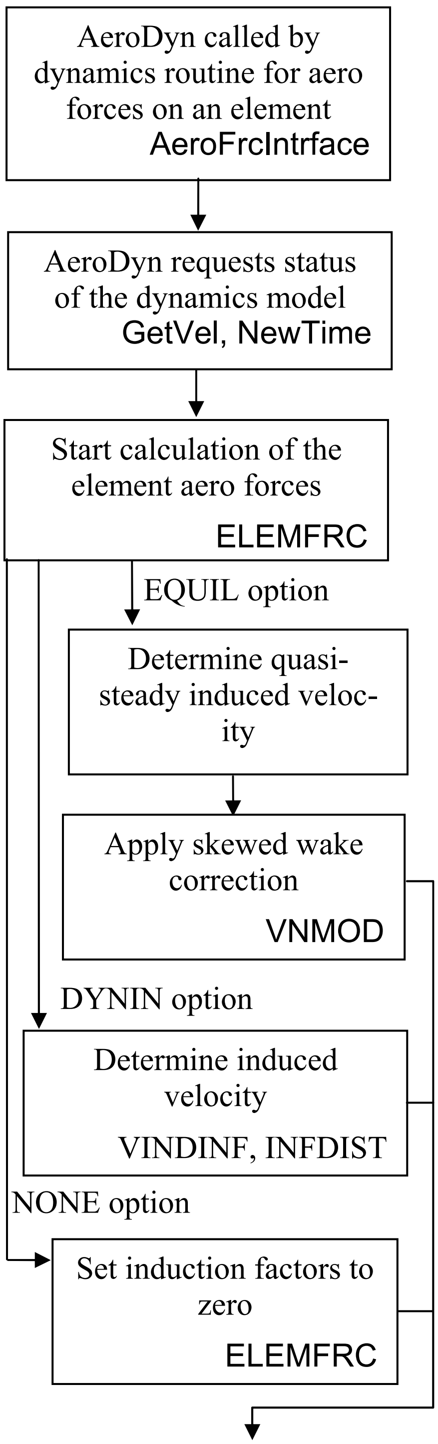

Figure A1. Schematic of the tower shadow model with a cross flow (VY). The tower wake decays in strength and grows in width as the distance from the tower, ι, increases. The strength and half-width are specified at a reference position, a distance $L_S$ from the tower center.

# Appendix B. Top-Level Flow Chart of the Aerodynamics Calculations

This appendix provides a simplified flow chart for the aerodynamics calculations. The purpose of the chart is to familiarize the user with the method that is used and some of the key assumptions. The chart does not map the flow of the entire code, nor does it use the format of traditional software flowcharts. The chart focuses on the operations rather than the code or subroutine structure. Each box of the chart does, however, indicate the name of the subroutine(s) in which the procedure is performed in the lower, right-hand corner of the box (in Arial font). This is intended to assist users who wish to examine the details within the subroutines.

The procedure that is shown is completed once for each blade element at each time step. Other subroutines, not shown in this chart, handle input and output of data and integration of the equations of motion.

Flow Chart of the Aerodynamics Calculations
and Major Assumptions in AeroDyn
May, 2001




- Dynamics routine calls AeroDyn to request aerodynamic forces

- AeroDyn calls specific subroutines in the dynamics program to request model status, including element pitch and velocities.
- All velocities expressed as components normal and tangential to the rotor plane

- Call appropriate routines to calculate induction factors and aero coefficients

EQUIL Option

- Iterative procedure to determine induction factor
- Drag force excluded in momentum balance
- Prandtl tip-loss model included
- Static $C_L$ is used in momentum balance

- Skewed wake correction based on dynamic inflow method of Pitt and Peters
- Applied in quasi-steady manner to the local induced velocity
- Uses average induced velocity to determine inflow angle to rotor

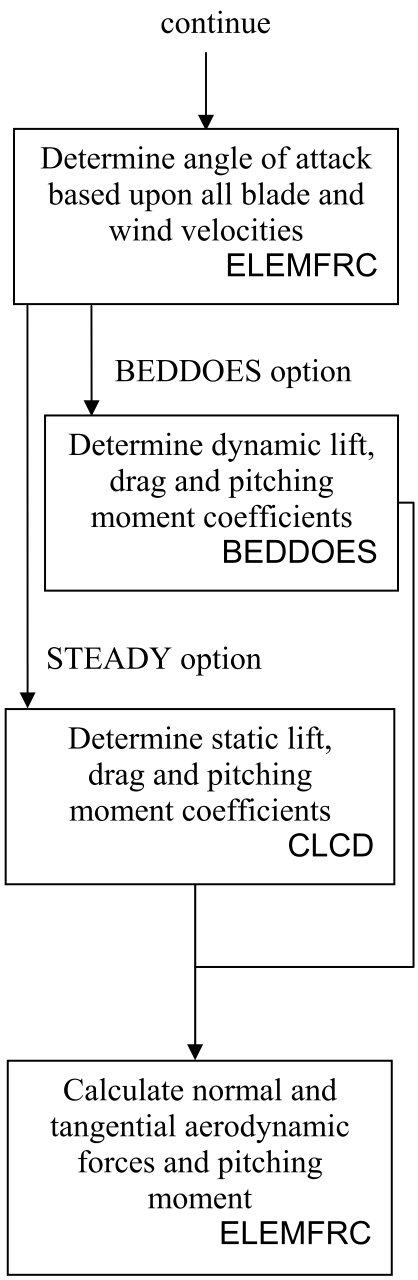
DYNIN Option

- Induced velocity calculated directly based on modified method of Pitt and Peters
- Skewed wake correction is inherent in the model
- Prandtl tip-loss model included

NONE Option

- Induced velocity calculations bypassed
- Induction factors set to zero.

Aerodynamics Flow Chart (continued)




- All motions or wind components are considered in calculation of angle of attack
- Method of Beddoes and Leishman
- Look-up coefficients in the static data tables.
- Based upon all blade and wind velocities
- Applied in the local coordinates of the element of the deformed blade

# Appendix C. AeroDyn Interface Parameters and Subroutines

Dynamic program parameters in AeroDyn

AeroDyn requires certain parameters to be supplied by the dynamics program and then returns aerodynamic forces for each blade element. The proper conventions must be used to obtain valid results. These parameters and their conventions are listed here. Figures C1 and C2 may help in understanding these conventions.

The model parameters that must be provided to AeroDyn by the dynamics code can be divide into two sets. The first is model definitions provided at initialization of the AeroDyn routines. These values must be known before AeroDyn is first accessed by the dynamics code. There is only one required value:

- Number of blades – an integer number that defines the number blades on the rotor.

The second set of parameters are those provided to AeroDyn for the determination of aerodynamic forces throughout the simulation. These are:

- Rotor speed ($\Omega$)– measured in radians per second, is always positive regardless of clockwise or counter-clockwise rotation.
- Yaw angle ($\gamma$) – measured in radians, is positive clockwise when viewed from above the nacelle. Zero reference is also the zero reference for wind direction ($\delta$) so that yaw error (or difference between the compass rotor direction and wind direction) is $\gamma - \delta$. Note that $\delta$ is positive for a *negative* $V_Y$ (see Figure 7.1).
- Hub velocity due to yaw rate ($V_{HUB}$)– measured in distance units (meters or feet) per second, is the tangential velocity of the rotor hub due solely to the yaw rate of the nacelle. Positive yaw rate (clockwise from above) leads to positive $V_{HUB}$.
- Tilt ($\tau$)– measured in radians, is the tilt angle of the rotor shaft to the horizontal plane. Positive tilt lowers the upwind end of the nacelle.
- Azimuth angle ($\psi$) – measured in radians, is the azimuth angle of a <u>blade</u> (not the entire rotor). The convention is $\psi = 0$ for a blade straight down (the 6 o'clock position), increasing with clockwise rotation of the rotor when looking downwind. AeroDyn should handle any value, (whether between 0 and $2\pi$, $-\pi$ and $\pi$, or cumulative) as long as it follows this convention. The rotor does not need to rotate clockwise, but the azimuth measurement must follow this convention.
- Multiple airfoil table location (MulTabLoc)– this is a user specified parameter for use when aerodynamics are determined from multiple airfoil tables dependent on an additional parameter such as aileron angle or Reynold's number. Can be ignored by most users.
- Pitch angle ($\theta$)– measured in radians, is the pitch angle of the local element relative to the plane of rotation. Positive pitch angle rotates the element leading edge (from zero pitch reference) into the wind, or nose down in fixed wing terminology (see Figure C2).
- Local element radius ($R_{local}$) – measured in distance units (meters or feet), is the perpendicular distance from the rotor axis of rotation to the element aerodynamic reference point (point where the aerodynamic loads are applied, usually the ¼ chord). Figure C1 shows an example.
- Element location ($X_{GRND}$, $Y_{GRND}$, $Z_{GRND}$)– measured in distance units (meters or feet), this is the 3-coordinate location in space of the element aerodynamic center. This location is relative to the undeflected tower centerline and hub-height. X is positive downwind (in the 0 yaw angle, 0 wind angle) direction, and Z is positive up. Figure C1 diagrams this convention.
- Total tangential velocity ($V_{T\text{-}total}$)– measured in distance units (meters or feet) per second, this is the velocity of the wind relative to the blade element tangential to the plane of rotation (not the chord line). AeroDyn provides the wind velocity. Positive tangential velocity results from posi-

tive rotor rotation and is nominally in the direction from leading to trailing edge of the blade under normal operating conditions (see Figure C2)

- Normal wind velocity ($V_{N\text{-}w}$)- measured in distance units (meters or feet) per second, this is the wind velocity normal to the local blade span (not the chord line) at the element location. The dynamics code must calculate this based on the wind velocities ($V_X$, $V_Y$, $V_Z$) provided by AeroDyn and the current element location. Positive is nominally in the direction from the pressure to the suction side of the blade under normal operating conditions (see Figure C2).

- Normal element velocity ($V_{N\text{-}el}$) - measured in distance units (meters or feet) per second, this is the element velocity normal to the local blade span and nominal plane of rotation (not the chord line). Positive $V_{N\text{-}el}$ is nominally upwind so that the relative wind velocity it creates adds to the positive normal wind velocity ($V_{N\text{-}w}$).

<u>Interface Subroutines</u>

The dynamics program enters the AeroDyn code in two distinct gateways. These are for: 1) retrieving inputs for and initializing AeroDyn parameters, and 2) calculating element aerodynamic forces. AeroDyn can also be accessed for writing element outputs, but this is optional and easily implemented through the input file, so it will not be detailed in this guide. Details on the other two gateways into AeroDyn are given below.

AeroDyn Input Gateway

The first interaction with AeroDyn is when the dynamics program calls subroutine ADInputGate. There are no arguments passed in the call statement. However, arguments are passed through variables in the AeroDyn modules. This call should be made early in the simulation, preferably during the first time step of the simulation. Before this call, however, the number of blades must be known. This means that the call to ADInputGate should occur after other model parameters are read from the dynamics program model or input file.

During this call to the ADInputGate, AeroDyn sets the constants dependent on $\pi$, reads the aerodyn.ipt file (and the wind and airfoil data files), allocates arrays and initializes other variables, and writes the OPT file (e.g., yawydn.opt or gfosub.opt).

Aerodynamic Force Interface

When the dynamics program needs the element aerodynamic forces, a call to the subroutine AeroFrcIntrface is made. There are five arguments in the call statement. The first two are passed to AeroDyn. These are:

- FirstLoop –a logical argument (true or false) that tells AeroDyn whether or not to initialize AeroDyn variables on the first call for each element. Programs such as ADAMS that do not conduct a trim solution need to set FirstLoop true for the first call for each element. After looping through all the elements, it should be set to false.
- JElem –the integer number of the current element. The elements must be numbered starting at one and increasing from most inboard (at the blade root) to most outboard (at the blade tip). The blade number (IBlade) is passed via the module Blade in AeroMods.F90.

Three aerodynamic loads are returned in the arguments of the call to AeroFrcIntrface. Note that the forces returned are in the blade coordinate system (not the element), as shown in Figure C2. These 3 arguments follow the two listed above in the following order:

- Normal force ($F_N$) – measured in force units (Newtons or pounds-force), this is the aerodynamic force on the element in the direction normal to the blade, perpendicular to the local span, in the direction downwind (nominally toward the suction side of the airfoil).

- Tangential force ($F_T$) – measured in force units (Newtons or pounds-force), this is the aerodynamic force on the element in the direction tangential to the blade, also perpendicular to the local span, but in the direction of positive rotation of the rotor (nominally toward the leading edge).
- Pitching moment ($M_P$) – measured in force units (Newtons or pounds-force) per length unit (meter or foot), this is the aerodynamic pitching moment on the element. Positive pitching moment causes negative pitch. That is, positive pitching moment tends to pitch the element to stall, or nose up in fixed wing terminology.

During the call to AeroFrcIntrface, AeroDyn in turn calls certain subroutines that must be part of the dynamics analysis code, in order to get the model status. Through these subroutines, the dynamics code provides AeroDyn with the parameters listed above in section 5.0.

AeroDyn calls four sets of subroutines to get information on the physical status of the dynamics model. The first three return information on the position of the turbine. The last deals with element and wind velocity. These subroutines and there arguments (which were detailed in Section 5.2) are as follows:

- Subroutine GetRotorParams (Rotor speed ($\Omega$), Yaw angle ($\gamma$), Hub velocity due to yaw rate ($V_{HUB}$), Tilt ($\tau$)) - called once per time step.

- Subroutine GetBladeParams (Azimuth angle ($\psi$)) - called once for each blade at each time step.

- Subroutine GetElemParams (Multiple airfoil table location (MulTabLoc), Pitch angle ($\phi$), Local element radius ($R_{ELM}$), X element location ($X_{GRND}$), Y element location ($Y_{GRND}$), Z element location ($Z_{GRND}$)) - called once for each element at each time step.

- Subroutine GetVNVT (X wind velocity ($V_X$), Y wind velocity ($V_Y$), Z wind velocity ($V_Z$), Total tangential velocity ($V_{T\text{-}total}$), Normal wind velocity ($V_{N\text{-}w}$), Normal element velocity ($V_{N\text{-}el}$)) – called once for each element. The first three arguments are provided by AeroDyn for the dynamics program to calculate $V_{T\text{-}total}$ and $V_{N\text{-}w}$. These are shown in Figure C1 and are:
    - X wind velocity ($V_X$) – measured in distance units (meters or feet) per second, the horizontal wind velocity in the global X direction at the element location.
    - Y wind velocity ($V_Y$) – measured in distance units (meters or feet) per second, the horizontal wind velocity in the global Y direction at the element location.
    - Z wind velocity ($V_Z$) – measured in distance units (meters or feet) per second, the vertical wind velocity (in the global Z direction) at the element location.

Once all this information is collected through these 4 calls, AeroDyn calculates the 3 aerodynamic loads on the element and returns them in the arguments of the AeroFrcIntrface call as listed above. These forces can then be applied to the dynamics model in the appropriate manner.

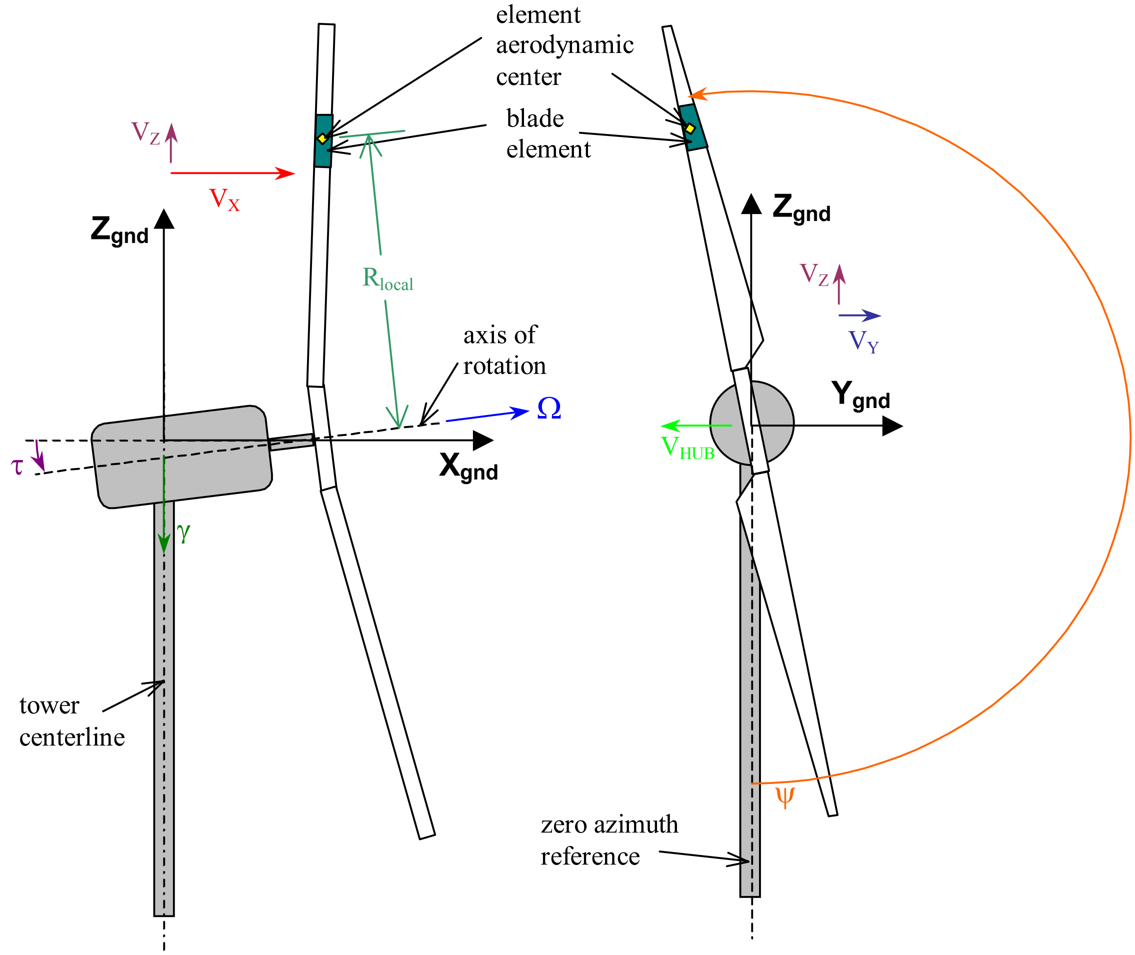



Figure C1 – AeroDyn ground coordinate system and sign conventions. All angles shown are positive. Rotor shown in zero yaw angle condition. Ground coordinate system, X, Y, Z does not move or rotate with any part of the turbine.

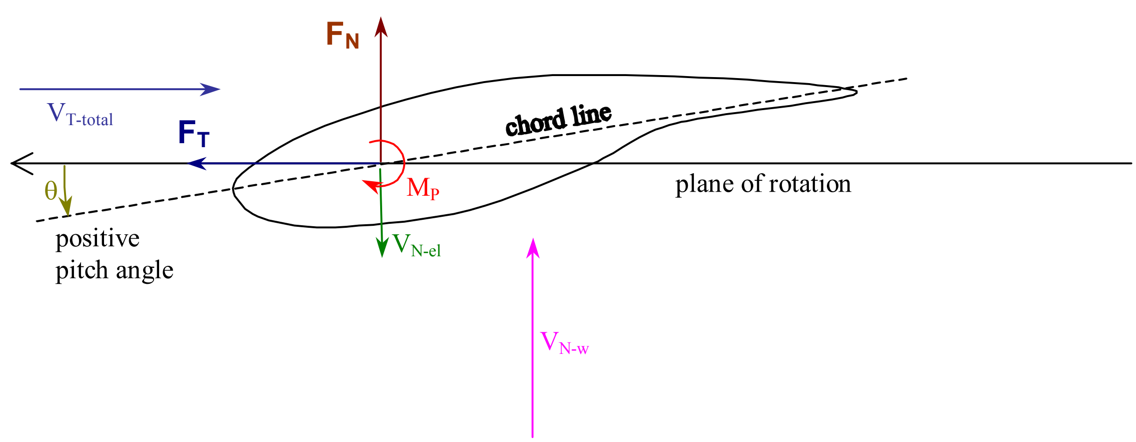



Figure C2 – Cross-section view of a blade element indicating a positive pitch angle, ϕ.

# Appendix D.  User's Guide to the FoilCheck Program

## Introduction

Airfoil data are rarely available for angles of attack over the entire range of ±180°.  This is unfortunate for the wind turbine designer, because wind turbine airfoils do operate over this entire range.  The AeroDyn software requires the user to provide airfoil data tables over the entire range so that it will be able to analyze any combination of wind speed, rotor speed, wind direction and yaw angle.  If a table is not provided over the complete range, and an unusual angle of attack is encountered during the calculations, AeroDyn will terminate the simulation with an error message.

Fortunately, the aerodynamic characteristics of an airfoil generally become independent of the airfoil section shape for very high positive or negative angles of attack.  This makes it possible to extrapolate from wind tunnel data (for the particular airfoil) to flat-plate characteristics for angles of attack near ±90°.  The flat-plate lift and drag characteristics depend only upon the aspect ratio of the plate.

The FoilCheck program is a simple utility program that helps the user create an airfoil data table for AeroDyn.  It performs the following major functions:

1) Extrapolates airfoil data from a limited range of angles to the entire range of angles using flat-plate characteristics.

2) Calculates the parameters of the Beddoes dynamic stall model, based upon the static characteristics of the airfoil.

3) Writes an airfoil data file in the format required by AeroDyn.

4) Writes an auxiliary file that can be examined to evaluate the "goodness" of the airfoil data file.

FoilCheck requires the user to start with a data file in the AeroDyn format.  All static airfoil characteristics of the airfoil that are known to the user must be contained in the input file.  The static data need not cover the angles between ±180°.  Dummy values for the dynamic stall characteristics must also be in the file as place holders.

**CAUTION**:  The program assists the user in creating an airfoil data file.  The process still requires accurate input data and judgment by the user.  It does not completely automate the process of creating accurate data files.  The user is prompted for inputs that require engineering judgment and, sometimes, a bit of guesswork.  It is very important that the user check the resulting data file to be certain it is credible.  This is one of the most important and difficult steps in creating an accurate AeroDyn model of a turbine blade.  We hope that FoilCheck eases the burden of creating the data files in the necessary format, but we know it cannot ease the burden of ensuring the data are accurate.  The importance of accurate airfoil data cannot be overstressed.  We encourage all users to devote considerable energy to locating airfoil data that is appropriate for the Reynolds number and surface roughness that will be seen on the turbine.  The list of references at the end of this Appendix contains several sources of data for wind turbine airfoils over an extended range of angle of attack.  Furthermore, extrapolation cannot be as accurate as test data.  The accuracy of your turbine simulation is highly dependent upon accurate airfoil characteristics.

## Method

FoilCheck uses a combination of wind-tunnel data, the Viterna equations for deep stall, and user experience to generate airfoil data for all angles from a limited set of measurements.  The method is not proven, but it has been helpful to some users of AeroDyn. So we decided to include the program in the AeroDyn distribution.

Figure D1 shows lift and drag coefficients for an example airfoil. Letters A-G across the top of the plot show different regions of angle of attack. Region A is the location of the wind tunnel data for this airfoil. It is quite common to only have reliable data for angles between approximately 0° and 20°. All of the remaining regions are constructed from this data set using FoilCheck. So, it is clear that there is some hazard involved in the extrapolations.

Region B, from a point just beyond stall to 90°, is the region that the Viterna equations are applied in their original form. The equations are taken from a report by Viterna and Janetzke. (Note there is a typographical error in the equations in the report. The correct equations are given below.) Additional references are given at the end of this Appendix.

$$C_{D_{\max}} = 1.11 + 0.018AR \tag{1}$$

$$C_D = C_{D_{\max}} \sin^2 \alpha + B_2 \cos \alpha \tag{2}$$

where

$$B_2 = \frac{C_{D_s} - C_{D_{\max}} \sin^2 \alpha_s}{\cos \alpha_s} \tag{3}$$

and subscript s denotes the value at the stall angle (called the matching point in this User's Guide because it need not be exactly at stall). *AR* is the blade aspect ratio. The lift is given by

$$C_L = \frac{C_{D_{\max}}}{2} \sin 2\alpha + A_2 \frac{\cos^2 \alpha}{\sin \alpha} \tag{4}$$

where

$$A_2 = \left(C_{L_s} - C_{D_{\max}} \sin \alpha_s \cos \alpha_s\right) \frac{\sin \alpha_s}{\cos^2 \alpha_s} \tag{5}$$

These equations yield $C_L=0$ and $C_D=C_{D\max}$ at $\alpha$=90°, and the stall (or matching point values) at $\alpha_s$. Thus it is important to select $\alpha_s$ carefully.

Regions C, D and E of Figure D1 values are obtained by scaling and reflecting the values from Region B. The reflections are evident from the figure. FoilCheck applies a scaling factor to $C_L$ to account for the asymmetry of the airfoil. The scaling factor is 0.7. That is, all lift values are reduced by 30% from the values shown in Region B. Drag values are not changed, just reflected. In regions F and G, linear interpolation is used to connect the various regions. $C_L$ is forced to zero at $\alpha=\pm180°$.

Pitching moment coefficients can also be extrapolated from tabular data. The extrapolation method is based upon the following assumptions:

1) The center of pressure moves to the midchord at $\alpha$=90°. This implies CM at 90° is $-C_{D\max}/4$.

2) The location of the center of pressure can be estimated using a Tan($\alpha$) function. The curve is fit between the value at 90° at the matching point, $\alpha_S$. This assumption is not proven, but gives reasonable results in the few cases where data are available over the entire range of angles of attack. Better results will be obtained in this extrapolation if you have tabulated values for angles greater than 20°.

3) The CM curve is reflected to positive values for negative angles of attack.

4) Fixed values are used at angles near 180°. The pitching moment can be large for reversed flow over the airfoil. Some wind turbines have experienced structural failures in their pitch control system as a result of this type of flow in very high winds. We therefore elected to use large and constant values as follows: (-170°, 0.40), (180°, 0.0), (170°, -0.50). These are the largest (absolute) values seen in limited data that are available from the Ohio State University reports listed at the end of this appendix. Of course, if you have data for your airfoil, we suggest using that data instead of FoilCheck.

The program is written in modular form that we hope will be easy for others to understand. If other users can improve upon any of these assumptions, we hope they will modify FoilCheck accordingly, and inform us of what they find.

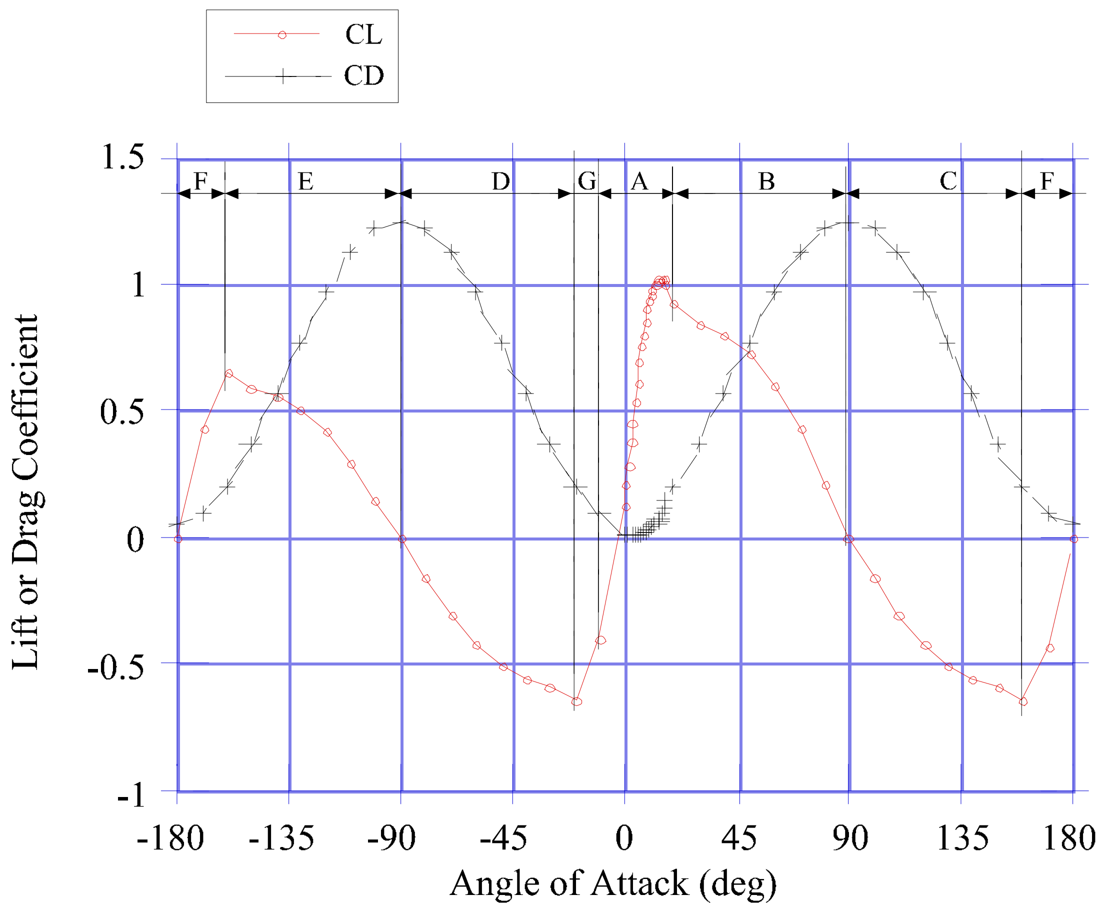



Figure D1. Lift and drag coefficients for a typical airfoil.

Installation

The source code for FoilCheck is held in the file Foilchk.F90, which contains all the source code main program and subroutines. To compile FoilCheck requires 2 other files from AeroDyn: AeroMods.F90 and GenSubs.F90. The source code must be compiled and linked using a Fortran 90 compiler.

Input data

FoilCheck requires two types of input. The data file read by the program must be in the same format as the airfoil data file used by AeroDyn. This format and file are described in the AeroDyn User's Guide. Any errors in the file will be caught by FoilCheck as it would by AeroDyn. The other type of input is interactive input from the keyboard during program execution. This is detailed later in this guide.

The format of the airfoil data file must match that of the AeroDyn airfoil data file exactly. However all numerical values related to the dynamic stall characteristics need not be accurate. These values are found in lines 5 through 12 in the data file. (Lines 3 and 4 - the number of airfoil tables and the ID parameter – must be accurate.) Also, the static airfoil lift and drag table should only cover the range for which values are accurately known. It is not necessary to provide a table starting at -180° and ending at +180°. FoilCheck will read the static table provided and build the new table based upon the input values. A sample input data file is shown in Table D1. Note the use of zeroes for all of the dynamic stall inputs and the relatively short range of angles of attack for the static lift and drag coefficient table.

Table D1 - Sample input airfoil file

```
S809 Airfoil, OSU data at Re=.75 Million, Clean roughness
NREL/TP-442-7817 Appendix B
   1        Number of airfoil tables in this file
      .00   Table ID parameter
     0.00   Not used
     0.00   Not used
     0.00   Not used
     0.00   Not used
     0.00   Zero lift angle of attack (deg)
     0.00   Cn slope for zero lift (dimensionless)
     0.00   Cn at stall value for positive angle of attack
     0.00   Cn at stall value for negative angle of attack
     0.00   Angle of attack for minimum CD (deg)
     0.00   Minimum CD value
   -20.10  -.560  .3027   .0612    Alpha, CL, CD, CM
   -18.10  -.670  .3069   .0904
   -16.10  -.790  .1928   .0293
   -14.20  -.840  .0898   -.0090
   -12.20  -.700  .0553   -.0045
   -10.10  -.630  .0390   -.0044
    -8.20  -.560  .0233   -.0051
    -6.10  -.640  .0131   0.0018
    -4.10  -.420  .0134   -.0216
    -2.10  -.210  .0119   -.0282
      .10   .050  .0122   -.0346
     2.00   .300  .0116   -.0405
     4.10   .540  .0144   -.0455
     6.20   .790  .0146   -.0507
     8.10   .900  .0162   -.0404
    10.20   .930  .0274   -.0321
    11.30   .920  .0303   -.0281
    12.10   .950  .0369   -.0284
    13.20   .990  .0509   -.0322
    14.20  1.010  .0648   -.0361
    15.30  1.020  .0776   -.0363
    16.30  1.000  .0917   -.0393
    17.10   .940  .0994   -.0398
    18.10   .850  .2306   -.0983
    19.10   .700  .3142   -.1242
    20.10   .660  .3186   -.1155
```

The interactive input is best described using an example. The left column of Table D2 below is a copy of the prompts and user input to the screen during program execution. (The Screen display is shown in `Courier` font, user inputs are shown **`bold`**. Spacing and fonts have been altered, and borders have been drawn for ease of reading, otherwise the left column is a copy of the screen display.) The right column provides some additional description of the options.

As with earlier versions of FoilCheck, default values are offered for many of the dynamic stall parameters, as well as for all the "y/n" (yes/no) prompts. The choice in square brackets, "[ ]", is the default that is used if you simply hit the Enter key. Some prompts also offer a "?" option which will provide more help on that topic if selected.

Table D2 - Sample FoilCheck Session

| Text appearing on computer display (both prompts and responses) | Explanation |
|---|---|
| WELCOME TO<br><br>FoilCheck 2.0 for AeroDyn 12.3 (04-Jun-2001)<br><br>============================================<br>(Respond to the prompts to generate<br>airfoil data tables for AeroDyn)<br><br>Enter the name of the airfoil file that you wish<br>to examine (<80 characters)<br>**s809_cln.dat** | You are first prompted for the name of the file containing the airfoil data. Any name, including path, up to 80 characters is accepted. |
| File "s809_cln.dat" found.<br><br>Does this file contain aerodynamic pitching<br>moment coefficient data?<br>Y/[N] or ? >**y** | You must tell FoilCheck if your input data file contains pitching moment coefficients. If you wish to have your output file contain $C_M$ values, then your input file must contain them. |
| FoilCheck determined there are 2<br>airfoil tables in this file.<br>Enter the number of the airfoil table<br>that you wish to examine.<br>**1** | This prompt appears only if you have more than one airfoil table in your data file. FoilCheck can only output results for one table, but it can be any table from your input file. |
| Do you want to calculate lift and drag<br>coefficients using the Viterna method?<br><br>(You should answer yes unless you are starting<br>with a table that covers -180 < alpha < 180)<br>Y/[N] >**y** | If your airfoil table does not include angles over the entire range between ±180°, you can use the Viterna equations to generate static $C_L$ and $C_D$ values. 'Y' or 'y' responses will invoke the Viterna calculations. |
| The blade aspect ratio (AR) is used to estimate<br>the maximum Cd for the airfoil (Cdmax)<br>using the equation Cdmax = 1.11 + 0.018*AR<br><br>If you wish to enter a different Cdmax then<br>you should enter an aspect ratio of zero<br><br>Enter the aspect ratio of the blade<br><br>**11** | The Viterna method requires input of $C_{Dmax}$. You can either enter the aspect ratio, from which the maximum drag is calculated, or you can enter 0. You will then be prompted for a value of $C_{Dmax}$. |

```
The lift and drag coefficients will be matched at
 the angle that you enter below and calculated
 for angles greater than the value you enter.

The angle should be at or above the stall angle.
 Normally, the largest angle in your table is best.

It should also be one of the values appearing
 in your original airfoil table.
     [Press ENTER to continue]

 Here is a list of tabulated values near stall
    ALPHA      CL        CD
     10.20     0.930    0.0274
     11.30     0.920    0.0303
     12.10     0.950    0.0369
     13.20     0.990    0.0509
     14.20     1.010    0.0648
     15.30     1.020    0.0776
     16.30     1.000    0.0917
     17.10     0.940    0.0994
     18.10     0.850    0.2306
     19.10     0.700    0.3142
     20.10     0.660    0.3186


 Enter the angle of attack you wish to use for
  the matching point (in degrees)
20.1
```

This is the most critical step in using the Viterna method. The equations fit a smooth curve between the matching point you enter here and finite-length flat plate characteristics at higher angles. The angle should be above stall, but generally not above approximately 30°. Angles between 15°-20° are most common. It is important to examine the resulting airfoil file after the program runs to be certain the curve is reasonable. But remember, FoilCheck is correcting for aspect ratio, so the results will not match 2-D data.

The program will list a table of values from which to choose.

You should enter a value from the input table. If you do not, you will be prompted for $C_L$, and $C_D$ values for the angle you entered.

```
To calculate the new airfoil table, FoilCheck needs
   the matching point for the lower bound.

The lift and drag coefficients will be matched at
the angle that you enter below and calculated
for angles less than the value you enter.

A typical value will be near zero degrees.
(You may choose the smallest angle in your table.)

It should be one of the values appearing
in your original airfoil table.
     [Press ENTER to continue]


Here is a list of tabulated values near zero degrees
   ALPHA      CL        CD
   -18.10    -0.670    0.3069
   -16.10    -0.790    0.1928
   -14.20    -0.840    0.0898
   -12.20    -0.700    0.0553
   -10.10    -0.630    0.0390
    -8.20    -0.560    0.0233
    -6.10    -0.640    0.0131
    -4.10    -0.420    0.0134
    -2.10    -0.210    0.0119
     0.10     0.050    0.0122
     2.00     0.300    0.0116
     4.10     0.540    0.0144

Enter the lower-bound angle of attack (in deg.)
  The value must not be less than the lowest
  angle in the original airfoil table.
-18.1
```

A range is established by your last entry and this next entry (20.1° and -18.1° in this example). Airfoil values for angles outside this range are created by calculating, scaling and reflecting the values within the range as described in the text of this User's Guide.

A table of values near zero degrees will be listed. You should select one of these values.

Again, it is important to look at the final airfoil file to see if good choices were made while running the program. This judgment is one of the most difficult parts of the process. FoilCheck cannot assist your selection other than by listing candidate values from your input table.

```
The zero-lift pitching moment coeff. = -0.0334

  A new data table has been created.
  The number of points in the three intervals:
         AOA below the table range:            17
         AOA within the table range:           25
         AOA above the table range:            16
  Number of entries in the table =           58

     [Press ENTER to continue]
```

The program creates a new data table that contains your original values, plus values at 10° intervals over the entire range outside of your original table.

```
A first iteration for the CN slope was performed
  with the following results obtained:
  [Press ENTER to continue]


 Minimum angle of attack for the interval =  -2.100
 Maximum angle of attack for the interval =   6.200
 Number of points in the interval         =  5

 CN Slope from linear least squares fit =  6.9071
 CN Slope read from the airfoil table   =  0.0000
 CN intercept from least squares fit    =  0.0450

  ALPHA (DEG)    CN-TABLE    CN-CALCULATED     ERR
   -2.100         -0.210        -0.208       0.0021
    0.100          0.050         0.057       0.0070
    2.000          0.300         0.286      -0.0141
    4.100          0.540         0.539      -0.0004
    6.200          0.787         0.792       0.0054

  Root-Mean-Square CN error from curve fit
    calculated over the specified interval
      RMS error =  0.0075


 Do you want to do another CN slope? (Y/[N])
Y
```

We have finished creation of the static airfoil table for all angles of attack. Now we turn to the dynamic stall parameters. First is the normal force slope. The program fits a least-squares line through the $C_N$ values for a range of angles you specify. The slope required by the theory is the slope at $C_N$=0.

You must reach a balance between having enough points in the curve fit to give confidence in the slope, and not increasing the range to large angles (large $C_N$). The program makes an initial attempt to find a slope spanning $C_N$ = 0.0, with an RMS error < 0.01. The result is written to the screen.

You can evaluate the fit and decide whether to try a different range or accept this range. While this first attempt usually provides useful results, in this example let's try a different range.

```
The program will calculate the CN slope
for the series consisting of all the data
points in the angle-of-attack range that you
specify in the next two lines of input

Enter the minimum angle of attack (deg)
-3
Enter the maximum angle of attack (deg)
3

Minimum angle of attack for the interval =   -3.000
Maximum angle of attack for the interval =    3.000
Number of points in the interval         =   3

CN Slope from linear least squares fit =   7.1250
CN Slope read from the airfoil table   =   0.0000
CN intercept from least squares fit    =   0.0466

 ALPHA (DEG)      CN-TABLE     CN-CALCULATED   ERR
  -2.100          -.210          -.214       -.0042
    .100           .050           .059        .0091
   2.000           .300           .295       -.0049

 Root-Mean-Square CN error from curve fit
   calculated over the specified interval
RMS error =   .0064

Do you want to do another CN slope? (Y/N)
n


  A CN slope value of    7.124995     will be
   written to the new airfoil data file.
        [Press ENTER to continue]
```

By trying a smaller range we get a smaller RMS error.

The 'n' response tells the program to accept this last value of $C_N$ slope and move on to the next set of questions.

You can answer 'y' as many times as you like, until you get a result that is satisfactory. If you want to use one of the results you saw earlier, run that range of angles again.

The CN-TABLE value is calculated from the $C_L$ and $C_D$ in your input table. CN-CALCULATED is the value calculated from the linear regression through the CN-TABLE values. ERR is the difference between the two.

The zero-$C_N$ angle of attack is also determined from the results of the linear regression.

| | |
|---|---|
| <pre>Next you must enter the stall angle of attack.<br> so FoilCheck can determine the CN value for that angle.<br><br>  The following points are your table values<br>     at angles bracketing stall:<br><br>      ALPHA        CL<br>     10.20     0.930<br>     11.30     0.920<br>     12.10     0.950<br>     13.20     0.990<br>     14.20     1.010<br>     15.30     1.020<br>     16.30     1.000<br>     17.10     0.940<br>     18.10     0.850<br>     19.10     0.700<br>     20.10     0.660<br>  A stall angle of attack of    15.30000<br>   has been found by FoilCheck.<br><br>  Do you want to accept this value ([Y]/N)?<br>**n**<br><br>  Enter the angle of attack at stall (deg)<br>**15.7**</pre> | FoilCheck is requesting the angle of attack for stall. To assist you, FoilCheck echoes your input table for angles near stall. FoilCheck also provides a default value for this angle of attack based on the angle of attack in the table with the maximum $C_L$. To accept this value enter “y” or simply press the Enter key.<br><br>If you wish to enter a different value than the default, enter “n”. You are then prompted to enter the stall angle. The value is entered in degrees. It does not have to equal one of the points from the input table.<br><br>FoilCheck calculates the $C_N$ at this stall angle. The value is extrapolated from the linear $C_N$-curve slope found above. This has been found to yield better results than the static value from the table. |
| <pre> Now you must enter a value for CN at the<br>  stall point for negative angles of attack.<br><br> The program will use a default value<br>  Equal to -0.8 if you would like.<br><br> Do you want to accept this default value ([y]/n)?<br>**Y**</pre> | Now we look at “negative stall”. If you know the value of $C_N$ at stall for negative angles, reject the default and enter the actual value. Otherwise, accept the default value. |
| <pre>Next you must enter the angle of attack for Cdmin.<br><br> The following points are your table values<br>    at angles bracketing minimum CD:<br><br>     ALPHA        CD<br>    -8.20    0.0233<br>    -6.10    0.0131<br>    -4.10    0.0134<br>    -2.10    0.0119<br>     0.10    0.0122<br>     2.00    0.0116<br>     4.10    0.0144<br>     6.20    0.0146<br>     8.10    0.0162<br>    10.20    0.0274<br>    11.30    0.0303<br><br> A minimum CD value of  0.0116<br>  At an angle of attack of    2.00 deg<br>  Has been found by FoilCheck.<br><br> Do you want to accept this value ([Y]/N)?<br>**N**<br><br>  Enter the angle for minimum CD (deg)<br>**1.5**</pre> | The final entry is the angle of attack at which $C_D$ is a minimum (near zero degrees, not 180°). FoilCheck presents a default from this table. Enter “y” (or press the Enter key) to accept this default, or “n” to input your own choice.<br>If you reject the default, you are prompted to enter an angle. It need not be from the table that is input, but the table is echoed to the screen to help you select a value. |

| | |
|---|---|
| `Finished. Three data files have been written`<br>`    foilnew.plt is the new airfoil data file`<br>`    foilchk.plt contains diagnostic data`<br>`    foilchk.opt is a copy of some screen output`<br><br>`  Remember: CHECK THE RESULTS BEFORE USING THE NEW DATA FILES!`<br><br>`Press Enter to exit program` | FoilCheck finishes by informing you of the files generated during the session. It also reminds you to check the new data before using it.<br><br>You can now press the Enter key to exit FoilCheck. |

FoilCheck creates three new data files. FOILNEW.PLT is the new airfoil data file in the AeroDyn format. You can edit the first two lines of the file to replace the generic TITLEs with meaningful notes regarding the file contents. This file contains only one airfoil table—for the table that was used as input. If you are using airfoil files with multiple data tables (such as aileron tables) you must cut and paste the multiple table file together from the many FOILNEW.PLT files that you will generate by running FoilCheck many times. FOILCHK.OPT contains the results of all of the CN slope calculations for review if desired. FOILCHK.PLT contains a number of calculated values for checking your results. It is a tab-delimited ASCII file suitable for import to a variety of graphics or spreadsheet programs for plotting. The column identification and description are shown in Table C3 below. We strongly suggest you examine this file closely to verify the accuracy of your airfoil tables as much as possible.

Table D3 - Column headings in the FOILCHK.PLT file.

| Column Heading | Description |
|---|---|
| Alpha | Angle of attack in degrees, from -180 to +180. |
| CL | Static lift coefficient |
| CD | Drag coefficient |
| CM | Pitching moment coefficient |
| CN | Normal coefficient (static) |
| CT | Tangential (chordwise) coefficient (static) |
| LiftDragRatio | The Lift/Drag ratio. This can be used to check the results. Very high or very low maximum L/D, or sharp discontinuities in the values should be examined closely. |
| CLBeddoes | This is the value of CL calculated from the Beddoes dynamic stall parameters *for static conditions*. It should match the CL values of column 2 very closely. If it does not, there is an error in one of the Beddoes parameters. |
| CDBeddoes | The drag coefficient calculated from the Beddoes dynamic stall parameters. This should also match the CD values of column 3 very closely |
| FtbBeddoes | A parameter used in internal dynamic stall calculations. It represents the fraction of attached flow on the airfoil. Values range from zero to one. This is available for users who are familiar with the details of the Beddoes dynamic stall calculations. |

## References for Appendix D

Butterfield, C. P. 1989. "Aerodynamic Pressure and Flow-Visualization Measurement from a Rotating Wind Turbine Blade." Eighth ASME Wind Energy Symposium, Houston, TX.

Butterfield, C. P. 1989. "Three-Dimensional Airfoil Performance Measurements on a Rotating Wing." European Wind Energy Conference, Glasgow, Scotland.

Butterfield, C. P., Simms, D., Scott, G., Hansen, A. C. 1991. "Dynamic Stall on Wind Turbine Blades." Windpower '91, Palm Springs, CA.

Eggers, A. J., Digumarthi, R. V. 1992. "Approximate Scaling of Rotational Effects of Mean Aerodynamic Moments and Power Generated by the Combined Experiment Rotor Blades Operating in Deep-Stalled Flow." Eleventh ASME Wind Energy Symposium, Houston, TX.

Gormont, R. E., May, 1973. "A Mathematical Model of Unsteady Aerodynamics and Radial Flow for Application to Helicopter Rotors." U.S. Army Air Mobility Research and Development Laboratory, USAAMRDL Technical Report, 76-67.

Leishman, J. G., Beddoes, T. S., 1989. A Semi-Empirical Model for Dynamic Stall. Journal of the American Helicopter Society. 34(3): 3-17.

Miley, S. J., February, 1982. "A Catalog of Low Reynolds Number Airfoil Data for Wind Turbine Applications." Rockwell International Rocky Flats Plant, RFP-3387.

Roensten, G., Dahlberg, J. A., Meijer, S. 1989. "Pressure Measurements on a 5.35m HAWT in CARDC 12x16m Wind Tunnel Compared to Theoretical Pressure Distributions." European Wind Energy Conference, Glasgow, Scotland.

Snel, H., Houwind, R., 1993. "Section Prediction of 3-D Effects for Stalled Flow on rotating Blades and Comparison with Measurements." European Community Wind Energy Conference, Luebeck_Travemuende, Germany.

Tangler, J. L., Ostowari, C., June, 1991. "Horizontal Axis Wind Turbine Post Stall Airfoil Characteristics Synthesization." NREL, SERI/TP-257-4400.

Viterna, L. A., Corrigan, R. D. 1981. "Fixed-Pitch Rotor Performance of Large HAWTs." DOE/NASA Workshop on Large HAWTs, Cleveland, OH.

Viterna, L. A. and Janetzke, D. C., Theoretical and Experimental Power From Large Horizontal-Axis Wind Turbines. NASA TM-82944, September, 1982.

Sources of airfoil data over an extended range of angle of attack:

Janiszewska, J. M., R. Reuss Ramsay, et al. (1996), Effects of Grit Roughness and Pitch Oscillations on the LS(1)-0417MOD Airfoil. National Renewable Energy Laboratory, NREL/TP-442-7819.

Pope, A. (1949). “The Forces and Moments Over an NACA 0015 Airfoil Through 180° Angle of Attack.” Aero Digest: p. 76.

Reuss Ramsay, R., M. J. Hoffman, et al. (1995), Effects of Grit Roughness and Pitch Oscillations on the S809 Airfoil. National Renewable Energy Laboratory, NREL/TP-442-7817.

Reuss Ramsay, R., M. J. Hoffman, et al. (1995), Effects of Surface Roughness and Vortex Generators on the LS(1)-0417MOD Airfoil. National Renewable Energy Laboratory, NREL/TP-442-6474.

Reuss Ramsay, R., M. J. Hoffman, et al. (1995), Effects of Surface Roughness and Vortex Generators on the NACA 4415 Airfoil. National Renewable Energy Laboratory, NREL/TP-442-6472.

Reuss Ramsay, R., M. J. Hoffman, et al. (1996), Effects of Grit Roughness and Pitch Oscillations on the LS(1)-0421MOD Airfoil. National Renewable Energy Laboratory, NREL/TP-442-6473.

Reuss Ramsay, R., M. J. Hoffman, et al. (1996), Effects of Grit Roughness and Pitch Oscillations on the S801 Airfoil. National Renewable Energy Laboratory, NREL/TP-442-7818.

Reuss Ramsay, R., M. J. Hoffman, et al. (1996), Effects of Grit Roughness and Pitch Oscillations on the S810 Airfoil. National Renewable Energy Laboratory, NREL/TP-442-7816.

## Appendix E  Description of Dynamic Inflow Model

With version 12.3 of AeroDyn, we have implemented a dynamic inflow model based on the Generalized Dynamic Wake (GDW) theory.  This replaces the modified Pitt and Peters model included since release 11.0.  This model runs significantly faster than the blade element/momentum (BEM) model (EQUIL option) because it does not require iteration at each time step (it must still iterate for a trim solution, however).  As this is the first release of this model, we recommend it with the usual caveats that accompany any new software.

The GDW model was developed as an expanded version of the Pitt and Peters model.  It uses a series solution to describe the induced velocity field in the rotor plane, which includes Legendre functions in the radial direction and trigonometric functions in the tangential direction.  The current GDW model in AeroDyn employs 0P, 1P, 2P and 3P terms.  The AeroDyn GDW model is based on the work of Suzuki.

The dynamic inflow effect is often insignificant, so results in most cases should not differ from the BEM results.  The exception is cases with rapid changes in blade angle-of-attack, where the dynamic inflow effect can be significant.  Curious results have resulted for cases of low wind speed, where the rotor operates in the "brake state."  Yaw oscillation resulting from the GDW model has also been observed while modeling a small, upwind, tail-vane turbine, though the reasons for this are not yet known.

References for Appendix E

Suzuki, A. and Hansen, A.C., April 1998, "Dynamic Inflow Model for YawDyn," Proceedings, Windpower 1998, Bakersfield, CA, U.S.A., pp. 777-783, American Wind Energy Association.

Suzuki, A. and Hansen, A.C., January 1999, "Generalized Dynamic Wake Model for YawDyn," Proceedings, 18th ASME Wind Energy Symposium (37th Aerospace Science Meeting and Exhibit), Reno, NV, U.S.A., pp. 186-191, ASME and AIAA.

Suzuki, A. and Hansen, A.C., June 1999, "Dynamic Inflow Models with Nonlinear Induced Velocity Distribution for YawDyn/AeroDyn Codes," Proceedings in CD-ROM, Windpower 1999, Burlington, VT, U.S.A., American Wind Energy Association.

Suzuki, A., August 2000, "Application of Dynamic Inflow Theory to Wind Turbine Rotors," Doctoral Dissertation, The University of Utah Dept. of Mechanical Engineering, Salt Lake City, UT.

# INDEX

## A



## B



## C

## D



## E



## F

## U



## V



## W

## Y