SANDIA REPORT
SAND2012-7028
Unlimited Release
Printed August 2012

Numerical Manufacturing And Design Tool (NuMAD v2.0) for Wind Turbine Blades: User's Guide

Jonathan C. Berg and Brian R. Resor

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online order:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2012-7028
Unlimited Release
Printed August 2012

Numerical Manufacturing And Design Tool (NuMAD v2.0) for Wind Turbine Blades: User's Guide

Jonathan C. Berg
Brian R. Resor
Wind Energy Technologies Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1124

Abstract

Sandia National Laboratories has an on-going effort to reduce the cost of energy and improve reliability for wind systems through improved blade design and manufacture. As part of this effort, a software tool named NuMAD (Numerical Manufacturing And Design) has been developed to greatly simplify the process of creating a three-dimensional finite element model for a modern wind turbine blade. NuMAD manages all blade information including databases of airfoils, materials, and material placement to enable efficient creation of models. NuMAD is a stand-alone, user-friendly, graphical pre-processor for the ANSYS® commercial finite element package. The blade information contained in the NuMAD database is also used to manage capabilities such as output for CFD mesh, computation of blade cross section properties, and aeroelastic instability analysis of the blade. This user's manual describes the capabilities and usage of NuMAD.

ACKNOWLEDGMENTS

The authors, as well as the entire Wind Energy Technologies Department at Sandia would like to thank the Department of Energy Wind and Water Power Program for their continued support in our pursuit of state-of-the-art design tools and codes to support our ongoing research activities.

Additionally, the authors would like to thank the following individuals for their contributions relating to NuMAD and enabling access to advanced design tools for wind energy:

The NREL NWTC Design Codes Team, especially Jason Jonkman, Marshal Buhl, Gunjit Bir, Rick Damiani, and Bonnie Jonkman, for their work in developing and maintaining robust wind turbine aeroelastic simulation codes.

Daniel Laird, Sandia National Laboratories, for his significant contributions in the early development of the NuMAD software.

David Malcolm, Det Norske Veritas (DNV), for his efforts in developing the BPE algorithm for Sandia.

Brian Owens, PhD Student at Texas A&M, for his work in creation and verification of BLAST, a classical flutter tool for HAWTs.

Erick Johnson, Sandia National Laboratories, for his assistance in defining and creating the ability for NuMAD to output useful information for CFD analysis.

Joshua Paquette, Sandia National Laboratories, for his guidance in implementation of many of the design interactions represented in NuMAD.

CONTENTS

FIGURES

TABLES

NOMENCLATURE

APDL	Ansys Parametric Dynamic Language
BLAST	BLade Aeroelastic Stability Tool
BPE	Beam Property Extraction tool
CFD	Computational Fluid Dynamics
DOE	Department of Energy
DP	material Division Point
FAST	Fatigue, Aerodynamics, Structures and Turbulence
FE	Finite Element
FEA	Finite Element Analysis
GUI	Graphical User Interface
HAWT	Horizontal Axis Wind Turbine
LCS	Lift Curve Slope
MCR	Matlab Compiler Runtime library
NREL	National Renewable Energy Laboratory
NuMAD	NUmerical Manufacturing And Design tool
NWTC	National Wind Technology Center
SNL	Sandia National Laboratories

GLOSSARY OF TERMS

aerodynamic center This is an aerodynamic parameter that is an output from aerodynamic performance analysis of a two-dimensional airfoil section. The aerodynamic center is the point along the chord where the aerodynamic pitching moment does not vary with changes in angle of attack.

ANSYS ANSYS Mechanical APDL; Commercial finite element analysis software distributed by ANSYS, Inc., Canonsburg, PA.

blade pitch axis See 'blade reference axis.'

blade reference axis Line running the length of the blade and about which stations are positioned and rotated. May also be called the blade generator line or blade pitch axis.

blade generator line See 'blade reference axis.'

chord length Physical (non-normalized) distance from the nose to the tail of a given station.

chordwise Along the chord of the blade from leading edge to trailing edge. Direction perpendicular to spanwise.

high pressure (H.P.) surface The side of the airfoil with the greater aerodynamic pressure; the concave side of the camber line; the upwind side of the wind turbine blade

inboard Towards the root end of the blade, opposite of outboard.

keypoint (ANSYS terminology) A point in three dimensional space which serves as a reference for construction of the blade geometry; typically the keypoints generated by NuMAD correspond to material division points and airfoil coordinate pair values at each station.

low pressure (L.P.) surface The side of the airfoil with the least aerodynamic pressure; the convex side of the camber line; the downwind side of the wind turbine blade

material Generic term used to describe one of many types of material definitions in NuMAD: 1) isotropic material, 2) orthotropic/layer material, 3) composite material

material, composite (NuMAD terminology) A layer stack of one or more "isotropic" and/or "orthotropic/layer" materials; a fully defined material stack definition including thickness values and layer orientations for each layer.

material, isotropic (NuMAD terminology) A type of material property definition; to be used as a component in a composite material definition where thickness of the material is defined.

material, orthotropic (NuMAD terminology) A type of material property definition; to be used as a component in a composite material definition where thickness and orientation of the material is defined.

nose Leading edge of an airfoil or station.

outboard Towards the blade tip, opposite of inboard.

segment, BPE A discrete spanwise piece of the blade as analyzed by BPE.

spanwise Along the length of the blade from root to tip. Direction perpendicular to chordwise.

station Two dimensional cross-section of blade at a particular distance from the blade root. The shape of the station may be an airfoil, root section, or transitional shape.

tail Trailing edge of an airfoil or station.

twist The rotation of a station about the blade reference axis.

x-offset The distance from the “nose” of a station to the blade reference axis.

INTRODUCTION

Motivation for NuMAD

Blades for modern wind turbines have changed dramatically over the last couple of decades. The scale of the blades has increased dramatically as well as the corresponding cost. This has resulted in more attention being given to blade design and analysis. Though beam models are often adequate for preliminary design, a more detailed structural finite element model is necessary to verify final design.

Sandia takes part in an on-going effort to reduce the cost of energy and improve reliability for wind systems through improved blade design and manufacture. The design portion of this effort requires the level of detail possible with three-dimensional finite element analysis but not possible with two-dimensional beam models. A tool is needed to simplify and expedite the creation and analysis of finite element models for wind turbine blades, including parametric design studies. To address these issues, Sandia has created a software tool named NuMAD (Numerical Manufacturing And Design tool).

Commercial finite element analysis packages today are powerful and they are generally quite user-friendly. Still, even users well-versed in finite element model creation can spend significant amounts of time deciphering the correct format in which to input the extensive and complicated sets of data relating to wind turbine blade geometry and stacks of anisotropic materials (laminated composites). Experienced analysts may spend several weeks to develop a detailed finite element model of a turbine blade.

Depending on the methods used to develop the original model, subsequent modifications to the design may be relatively minor or quite time-consuming. Obviously, a process which maximizes model reusability would be beneficial, especially when parametric studies are required.

NuMAD has been designed to address all of these issues and provide an improved finite element model development environment for wind turbine blades. NuMAD is a stand-alone, user-friendly, Matlab[1]-based (graphical) pre-processor for the ANSYS® Mechanical APDL[2] commercial finite element engine. NuMAD may be run in a stand-alone mode to create ANSYS input files or in conjunction with ANSYS as a seamless interface. Experience with the ANSYS analysis package as well as with finite element methods is required for effective usage of the combined suite. No modification to the standard ANSYS package is necessary.

ANSYS and Other Finite Element Codes

ANSYS is just one of many commercial finite element packages that are adequate for analysis of wind turbine blades. ANSYS was chosen because of its common usage in the wind industry and because of its particular capabilities with respect to creation of the blade geometry. The architecture of the current release of NuMAD allows for relatively seamless integration of modules that output input scripts for other commercial FE codes, if the user thinks it is worthwhile. Users who wish to pursue this sort of development should contact the authors for guidance.

Capabilities

NuMAD provides an intuitive interface for defining the outer blade geometry, shear web locations, materials and stacks, and stack placement in the blade. The output from NuMAD is a sequence of ANSYS APDL commands used to create the FE model in ANSYS. Once the model is created, various analyses are used to understand the strength and response of the blade to given loads.

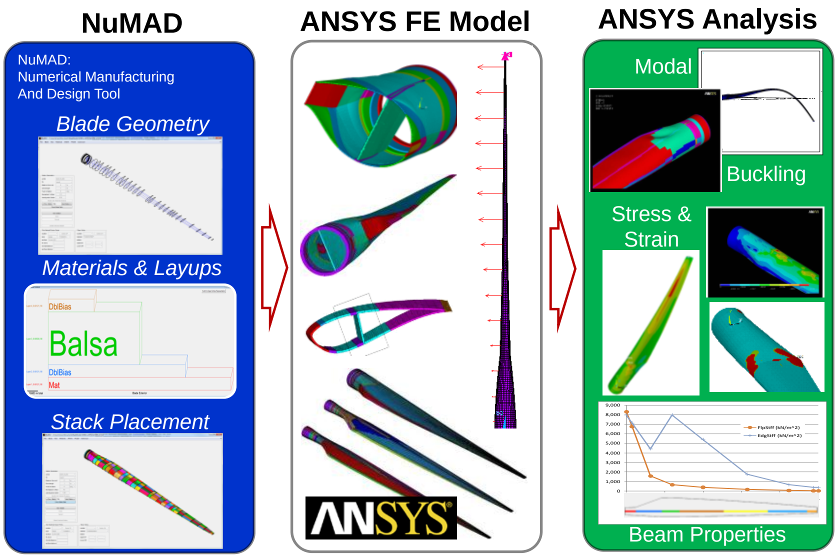


Figure 1 Graphical summary of NuMAD/ANSYS.

Motivation for Version 2

NuMAD first became available in the early 2000's. It was originally written in *Tcl*, chosen for its ability to easily manipulate graphics and port information in and out of the main code. NuMAD has always been designed to output files for ANSYS. The latest release of the previous NuMAD code is dated 31 March 2010.

Beginning with the last release of the previous NuMAD, its authors realized that blade design was quickly heading in a direction where a computationally intense, and platform independent programming environment was needed for NuMAD. Mathworks MATLAB® was chosen as the environment because of its widespread use, its computational power, its flexibility in operating systems, its graphical capabilities, its popularity with researchers and students, and its ability to compile as well as to run from raw source code. The current release of NuMAD also includes some advanced capabilities which go beyond the basic connection to creation of the finite element model: tabularized input format, swept blades, blades with pre-bend, output for CFD mesh creation and output of blade cross section properties.

The current version of NuMAD is available in two forms: 1) compiled MCR executable and 2) raw Matlab source files. Users are encouraged to study the raw source files in order to understand how NuMAD works. When there are capabilities that presently do not exist, the authors encourage users to work together with the authors to write modules that accomplish the tasks. It is the hope of the authors that the accessibility of the Matlab source code will, in the long run, enable an extremely capable tool that will continue to enable DOE's goals and benefit the entire wind industry.

QUICK-START GUIDE

This chapter is designed to quickly orient new users to the NuMAD installation procedure and interface. A usage example explores one of the blade models packaged with NuMAD and introduce the main program features. The subsequent chapters provide detailed instructions on building a new, albeit simple, blade model and performing basic analysis in PreComp and ANSYS.

File Package Contents

Table 1 NuMAD file package contents.

		32-bit MCR package, *arch*=PC WIN	64-bit MCR package, *arch*=PC WIN64	M-file package
NuMAD_Examples\				
	FlutterEx\	X	X	X
	pseudoSERI8_complete\	X	X	X
	SNLWindPACTv1.3\	X	X	X
NuMADexe_v2_*arch*.zip				
	Airfoils\	X	X	
	Macros\	X	X	
	MatDBsi.txt	X	X	
	MCRInstaller.exe	X	X	
	NuMAD.exe	X	X	
SNL_NuMAD_ToolBox_v2.zip				
	BLAST\			X
	BPE\			X
	Docs\			X
	Examples\			X
	MatlProps\			X
	NuMAD\			X
	Various m-files			X

Software and Hardware Requirements

The following are minimal requirements to run NuMAD

- Windows or Linux OS[i,ii]
- Matlab R2010 or newer (to utilize m-files) or the associated Matlab MCR (to run NuMAD in a compiled form)

[i] NuMAD is not currently set up to run from the compiled executable for Linux.

[ii] Support for the Mac OS is currently not available in NuMAD due to inaccessibility of Mac platforms at SNL for testing. However, Matlab is certainly compatible with Matlab. Users who wish to work with the NuMAD developers to implement compatibility with Mac OS are welcome to contact us for assistance.

- Download the script *txt2mat.m* (File ID#18430, 23 Jan 2008) from the *MatlabCentral* webpage: http://www.mathworks.com/matlabcentral/
- To create a FAST[3] blade file during blade property computation: download and install the BModes[4] code from the NREL NWTC Design Codes website: http://wind.nrel.gov/designcodes/preprocessors/bmodes/
- To perform section analysis using PreComp[5]: download and install the PreComp code from the NREL NWTC Design Codes website: http://wind.nrel.gov/designcodes/preprocessors/precomp/
- Optional, but highly recommended: ANSYS Mechanical APDL R12, or later.[iii]

Installing the MCR Version

The zip archive contains the compiled version of NuMAD as well as the Matlab MCR Installer. Using this version of NuMAD, users may execute the code without having access to an installation of Matlab. The following instructions assume the zip file has been extracted to *C:\DesignCodes\NuMADexe_v2_arch*

1. Move the example blade model *SNLWindPACTv1.3* to a convenient location in your computer. For example: *C:\Users\windowsuser\Documents\BladeModels*
2. Open the MCRInstaller by double-clicking *MCRInstaller.exe* and install the MCR package.
   Note: the MCR Installer that is distributed in this package is specific to the version of Matlab that was used to compile the NuMAD executable. It is typically required to install this package even if you already have another version of an MCR already installed on your system.
3. Open NuMAD by double-clicking on *NuMAD.exe*

Always monitor the output window that is generated when NuMAD is opened (Figure 2). The window is where errors appear and where occasional user input is required.

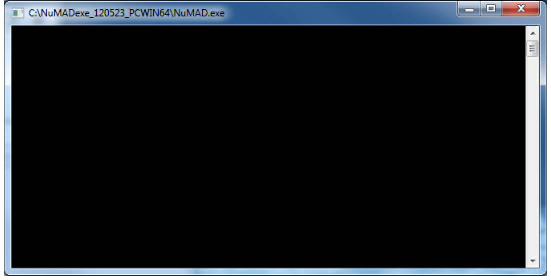


Figure 2 Output window as seen when running the compiled version of NuMAD.

Installing the M-file Version

The raw source scripts, or m-files, can be used directly be any installation of Matlab meeting the requirements described in the section of this manual titled, Software and Hardware Requirements. The NuMAD m-file version is distributed as a compressed folder containing the collection of Matlab functions. These functions may be placed anywhere in the file system. The

[iii] The current version of NuMAD is designed to work most effectively with ANSYS R14, which is the current release of ANSYS.

following instructions assume the zip file *SNL_NuMAD_ToolBox_v2.zip* has been extracted to *C:\DesignCodes\SNL_NuMAD_ToolBox\*

1. Move the example blade model *SNLWindPACTv1.3* to a convenient location in your computer. For example: *C:\Users\windowsuser\Documents\BladeModels*
2. Copy the file *snl.m* from ..*SNL_NuMAD_ToolBox\Docs* into your primary Matlab folder. This folder must be on the Matlab search path. Often, it is located (Windows OS) at *C:\Users\windowsuser\Documents\MATLAB*. You may see all the folders on your Matlab search path by typing *pathtool* at the Matlab command prompt.
3. Copy the file *txt2mat.m* to your Matlab search path. See the Section in this manual titled File Package Contents for file download information.
4. Open and edit the file *snl.m*. Set the toolbox path to match your actual path. For example,
   SNL_NuMAD_Toolbox_path = 'C:\SNL_NuMAD_ToolBox';

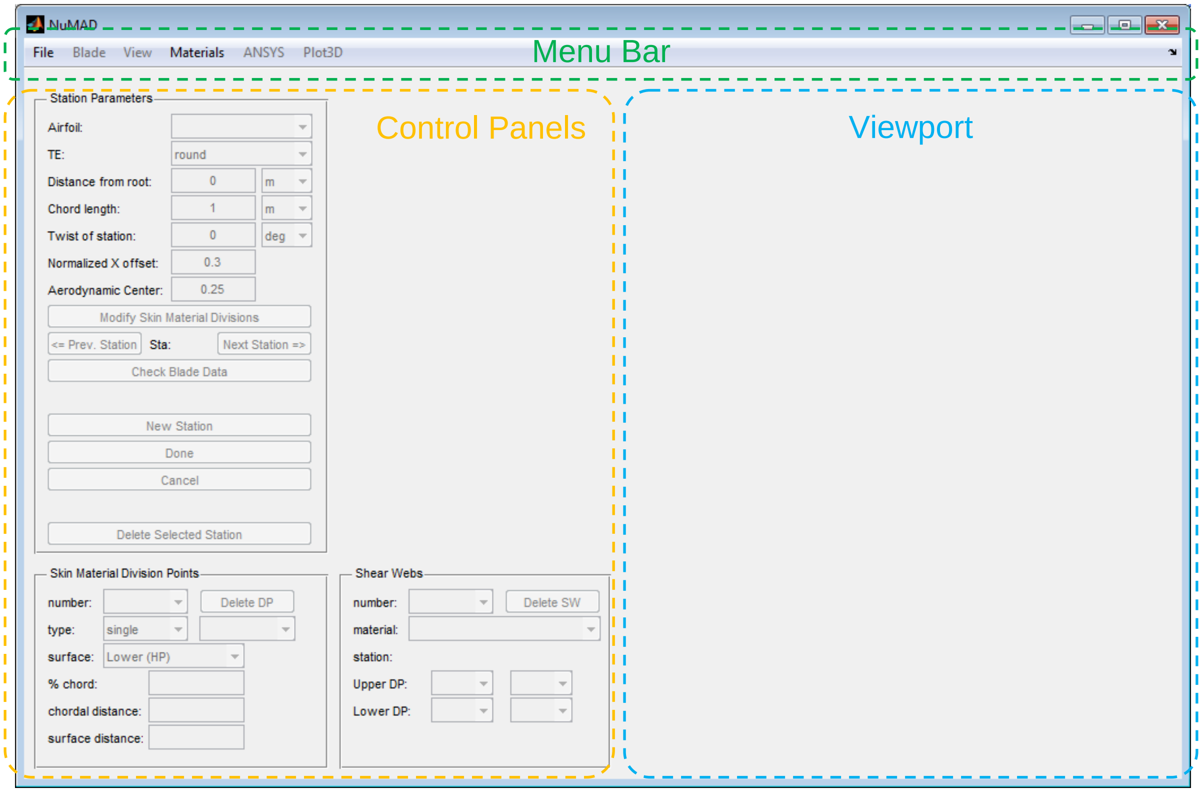
You may now automatically set the Matlab search path to include everything in the SNL_NuMAD_Toolbox simply by typing *snl* at the Matlab command prompt. As confirmation of successful path setup, the message, “SNL NuMAD Toolbox v*YY.MM.DD* path setup script complete” appears.

You may now open the NuMAD GUI by typing *numad* at the Matlab command prompt.

See Appendix A: NuMAD Toolbox Contents for a summary of all the contents of the m-file package.

NuMAD Interface Overview

The NuMAD interface, shown in Figure 2, has three main areas: (1) the viewport, (2) the control panels, and (3) the menu bar.



Airfoil:
TE: round
Distance from root: 0 m
Chord length: 1 m
Twist of station: 0 deg
Normalized X offset: 0.3
Aerodynamic Center: 0.25

number:
type: single
surface: Lower (HP)
% chord:
chordal distance:
surface distance:

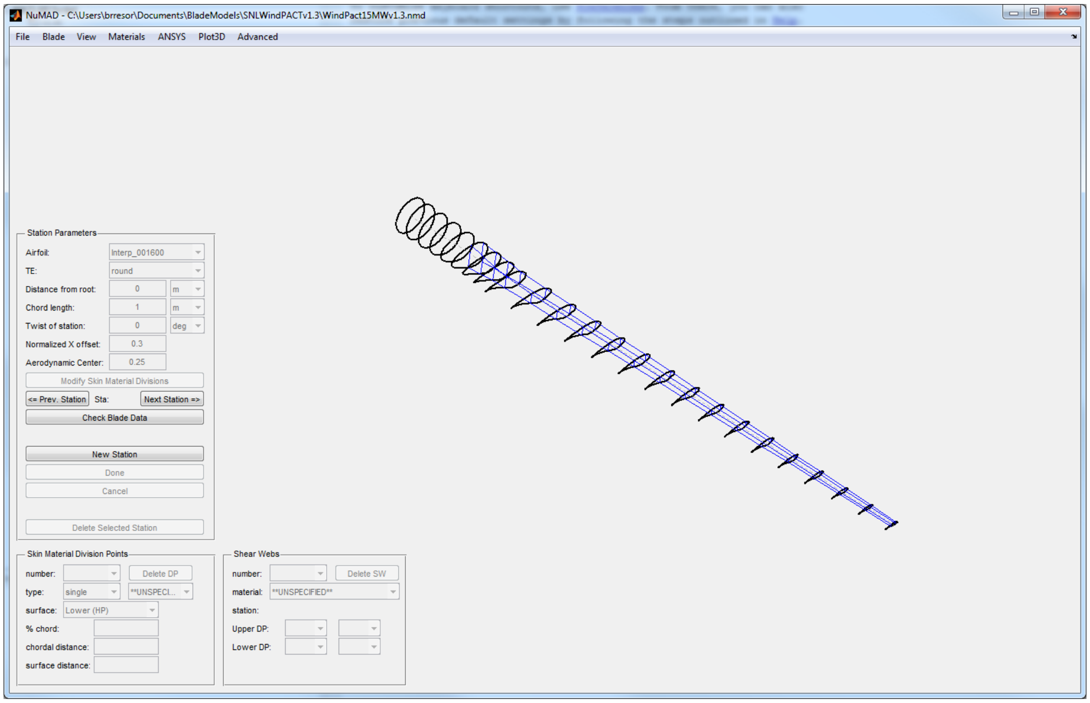
number:
material:
station:
Upper DP:
Lower DP:

Figure 3: Three parts of the NuMAD interface

Viewport

The viewport displays the blade geometry and provides an interactive graphical interface. We will now demonstrate the viewport's features with an example blade model. In NuMAD choose "File >> Open Model…", navigate to the blade model provided with NuMAD, and select *WindPACTv1.3.nmd*.

The newly opened blade model is displayed in the viewport as shown in Figure 4. The black curves are blade stations and the blue rectangles are shear webs.



Airfoil: Interp_001600
TE: round
Distance from root: 0 m
Chord length: 1 m
Twist of station: 0 deg
Normalized X offset: 0.3
Aerodynamic Center: 0.25

number:
type: single **UNSPECI...
surface: Lower (HP)
% chord:
chordal distance:
surface distance:

number:
material: **UNSPECIFIED**
station:
Upper DP:
Lower DP:

Figure 4: Blade model displayed in viewport

Viewport Tools

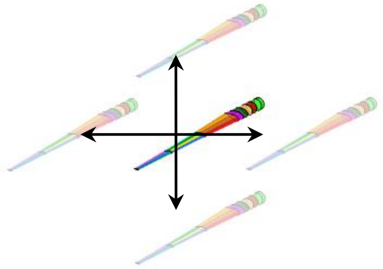
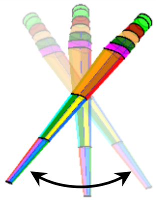
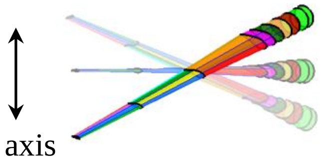
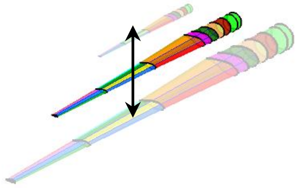
Three view tools allow the user to interact with the blade view. They are accessible from the View menu or by keyboard hotkeys.

The first view tool allows the user to relocate the geometry within the viewport by clicking and dragging with the left (first) mouse button. This tool is activated by either (a) selecting “Translate” tool from the “View” menu or (b) pressing the T key on the keyboard. The mouse pointer is an arrow (the standard mouse pointer) when the Translate Tool is active.

The second view tool allows the user to rotate the geometry within the viewport by clicking and dragging with the left mouse button. This tool is activated by either (a) selecting “Rotate 3D” tool from the “View” menu or (b) pressing the R key on the keyboard. The mouse pointer changes to a rotation icon when the “Rotate 3D” tool is active. Dragging the mouse left and right will rotate about an axis perpendicular to the blade. Dragging the mouse up and down will tilt the axis of rotation into and out of the viewport plane. Users with a three-button mouse can click and drag the middle mouse button to rotate (even when another view tool is active).

The third view tool allows the user to zoom in and out on the geometry by clicking and dragging with the left mouse button. This tool is activated either by (a) selecting “Zoom” tool from the “View” menu or (b) pressing the Z key on the keyboard. The mouse pointer changes to a magnifying glass when the Zoom tool is active. Dragging the mouse up and down will zoom out and zoom in, respectively. Dragging the mouse left and right has no effect. Users with a scroll-wheel mouse can zoom in and out with the scroll wheel (even when another view tool is active).

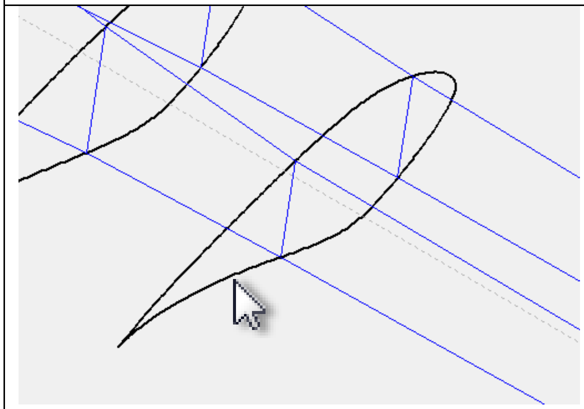
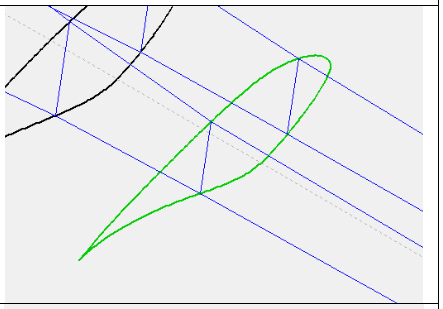
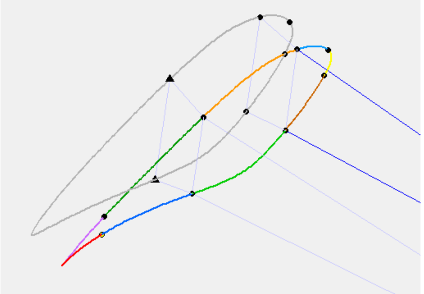
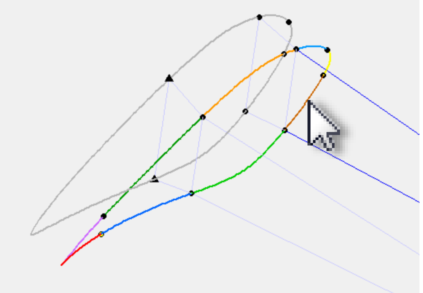
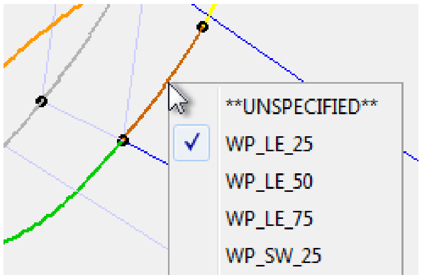
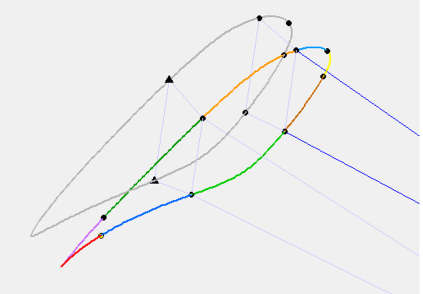
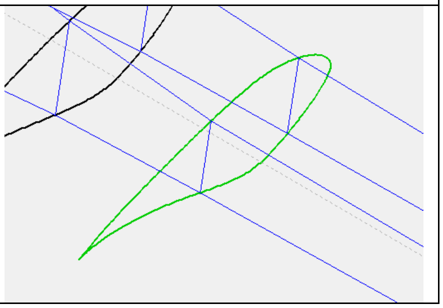
Table 2: Summary of translate, rotate, and zoom tools

			Click & Drag Motion	
View Tool	**Hotkey**	**Pointer**	**Left / Right**	**Up / Down**
Translate tool	T, G	↖		
Rotate 3D tool	R or 🖱	⟲	Rotation about blade perpendicular axis 	Rotation about screen horizontal 
Zoom tool	Z or 🖱 wheel	🔍	(no effect)	

Interactive Stations

The black curves representing the blade stations can be selected with a click of the left (first) mouse button. A station curve turns green when selected and its station parameters are displayed in the control panel area. Double-clicking a station curve opens the 3-station material placement view, which is described in the Station View section on page 50.

Table 3: Summary of station interactive capabilities in viewport

Initial State	Action	Result
	Click	
	Double-click	
	Right-click	 **UNSPECIFIED** ✓ WP_LE_25 WP_LE_50 WP_LE_75 WP_SW_25
	Done modifying	

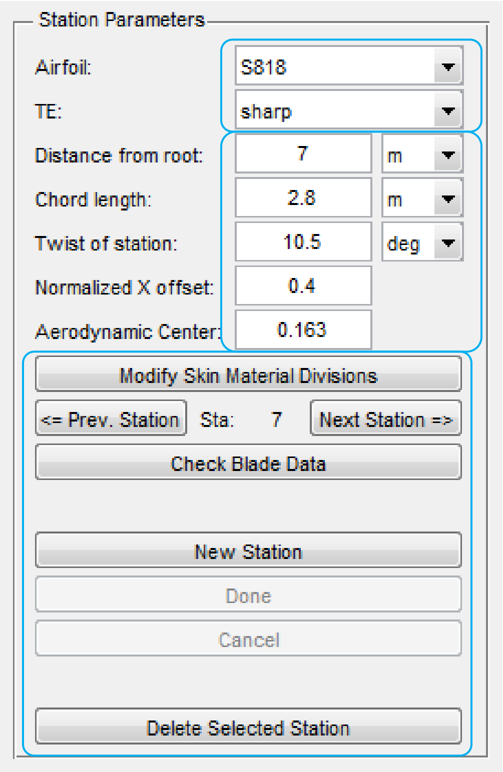
Control Panels

The control panels provide access to the currently selected station’s data. The buttons are grouped into three panels: Station Parameters, Skin Material Division Points, and Shear Webs. Some of the panels and controls may be inactive depending on the program state.

Figure 5 displays the various types of controls you will encounter in these panels. Drop-down menus allow one item to be selected from a list of options. Numeric data entry boxes accept only text recognized as numeric data. Some of the numeric data entry boxes have a drop-down menu of alternative measurement units which converts the numeric value automatically. Push buttons execute various program functions.

Within the Station Parameters panel, the “Next Station” and “Prev Station” buttons cycle through the stations sequentially, and the station number of the selected station is displayed between these two buttons.

See the Main Viewport and Control Panels section on page 49 for the full documentation on the control panels.



Station Parameters

Airfoil: S818

TE: sharp

Distance from root: 7 m

Chord length: 2.8 m

Twist of station: 10.5 deg

Normalized X offset: 0.4

Aerodynamic Center: 0.163

Sta: 7

Drop-down menu – select one option from list

Numeric data entry – some have measurement units drop-down menus

Push-buttons – execute various program functions

Figure 5: Types of controls in the control panels

Menu Bar

The main menu bar, identified in Figure 3, groups various program functions together into a few drop-down menus. The File menu creates, opens, and saves blade project files. The Blade, View, and Materials menus perform functions associated with blade model creation. The ANSYS and Plot3D menus perform output-related functions.

See the NuMAD Interface Reference on page 41 for the full documentation on the main menu bar.

BLADE MODELING INTRODUCTION

The following introduction to blade modeling utilizes the airfoil and material databases distributed with NuMAD to create a simple blade model. It is intended to demonstrate the overall blade modeling procedure without getting into the nuances and more advanced features of NuMAD. The blade modeled is nominally a SERI-8 (8 meter) blade[6] but is highly simplified to keep this example concise.

Open NuMAD and create a "New Model" via the File menu (Figure 6). Your screen should now look like Figure 7. The new blade model consists of two circular stations that are 10 meters apart.

Save the new model by selecting "Save As…" from the File menu. Choose a location to save the model and enter *pseudoSERI8* in the file name field (Figure 8). Then click the "Save" button. A new folder with the given name will be created and all data relating to this model will be saved in the folder.

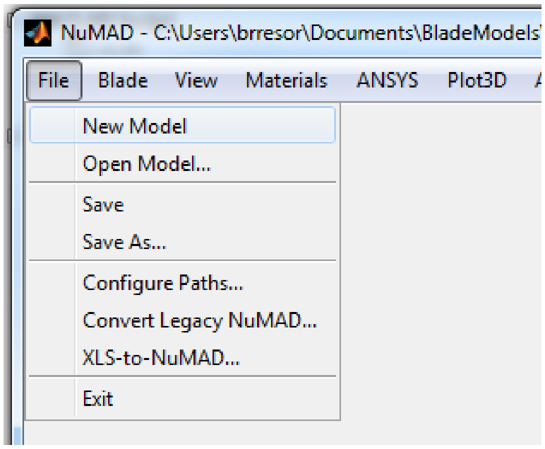
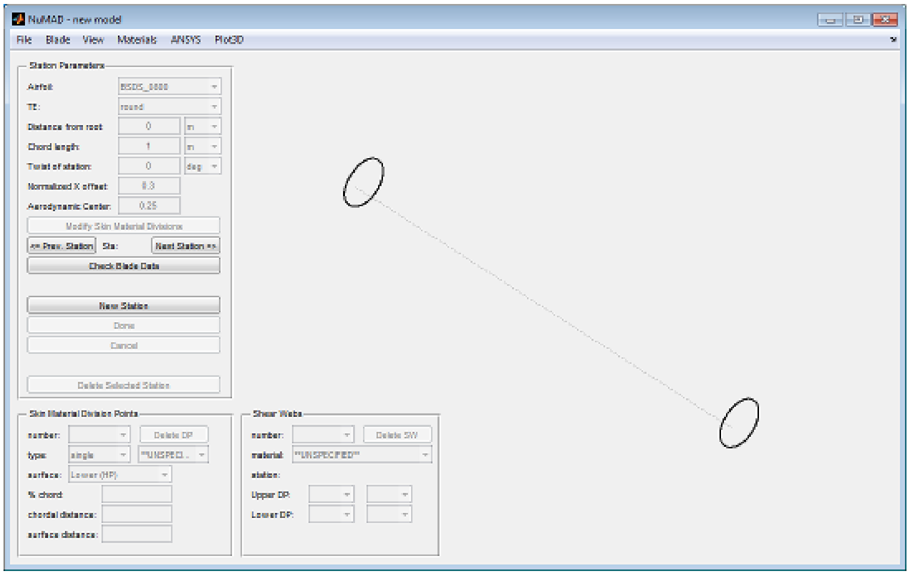


Figure 6. "New Model" entry selected from File menu.

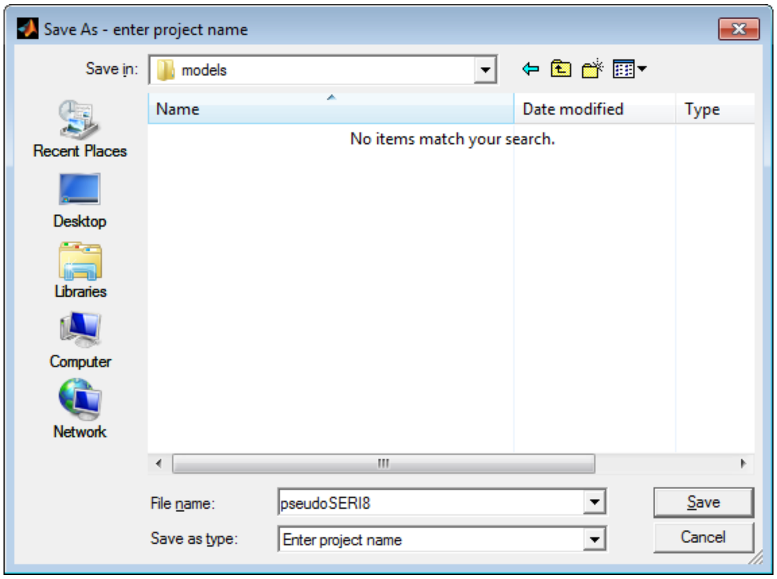


Airfoil: BSDS_0000
TE: round
Distance from root: 0 m
Chord length: 1 m
Twist of station: 0 deg
Normalized X offset: 0.3
Aerodynamic Center: 0.25

number:
type: single "UNSPECI..
surface: Lower (HP)
% chord:
chordal distance:
surface distance:

number:
material: "UNSPECIFIED"
station:
Upper DP:
Lower DP:

Figure 7: Start of a new blade model.



Save As - enter project name

Save in: models

No items match your search.

File name: pseudoSERI8

Save as type: Enter project name

Figure 8: Save As dialog box. In this example, the job name is "pseudoSERI8".

Modify the blade root and tip

Click on the left station line to select it; the line will turn green when selected (Figure 9). Alternatively, you may click the “Next Station” button until the left station is selected. Find the Station Parameters panel and locate the input fields for distance from root, chord length, station twist, and x-offset. This station will be the blade root and so the “distance from root” parameter should remain 0. Set the chord length to 0.419m, the twist to 20deg, and the normalized x-offset to 0.5 (see Figure 10). The trailing edge treatment (TE) and aerodynamic center can remain at their default settings.

Now select the right station (or click the “Next Station” button). Change the airfoil to S806A and its location to 7.925m. Next, set the chord length to 0.330m, the twist to 0deg, and the normalized x-offset to 0.3. These two stations define the root and tip of the blade. The rest of the geometry will be completed next.

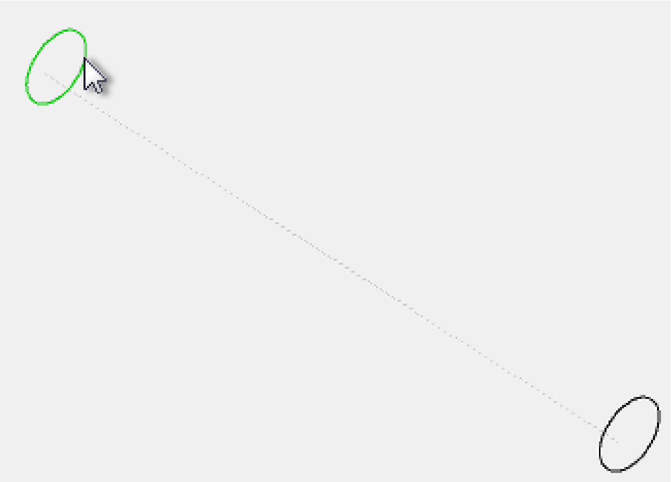
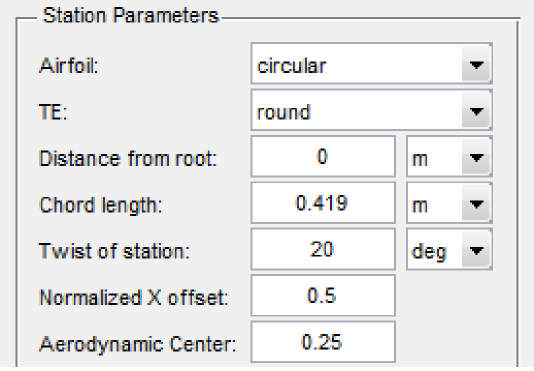


Figure 9: The left station line is green which indicates it is selected.



Station Parameters

Airfoil: circular

TE: round

Distance from root: 0 m

Chord length: 0.419 m

Twist of station: 20 deg

Normalized X offset: 0.5

Aerodynamic Center: 0.25

Figure 10: Station Parameters for the pseudoSERI8 root station.

Complete the blade geometry with new stations

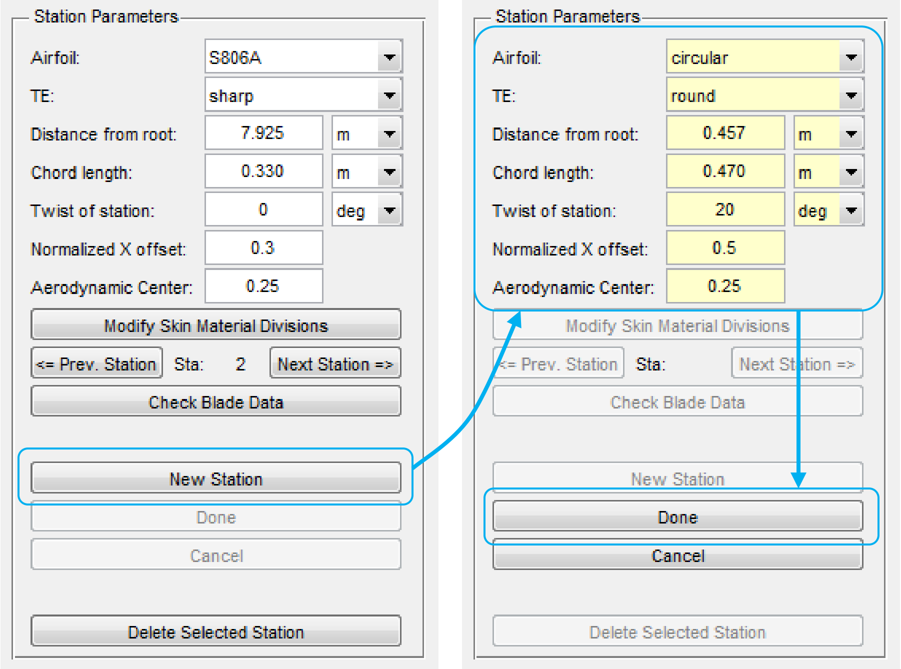
The next objective is to create the six additional stations which complete the blade geometry. Table 4 lists the parameters for all of the blade's stations. The first and last rows match the parameters already entered for the root and tip stations.

Complete the following steps for each of the remaining rows of the table: click the "New Station" button, change the appropriate station parameters, and then click "Done" to finish the station (see Figure 11). The model should look like Figure 12 after you have created all of the stations. Remember to save the model periodically as you create stations.

Table 4: Station parameters for the pseudoSERI8 model.

Station #	Airfoil[iv]	Location (m)	Chord (m)	Twist (deg)	X-offset
1 (root)	circular	0.000	0.419	20	0.5
2	circular	0.457	0.470	20	0.5
3	S808	1.524	1.118	20	0.3
4	S807	2.250	1.092	15.7	0.3
5	S805AS807	4.074	0.914	4.22	0.3
6	S805A	5.898	0.665	0.59	0.3
7	S806A	7.518	0.404	0.01	0.3
8 (tip)	S806A	7.925	0.330	0	0.3

[iv] Coordinates for S-series airfoils are publicly available through the NREL NWTC webpage. Certain airfoils are patented and may be licensed through NREL. More information on patents and commercial licensing can be found at the following URL: http://wind.nrel.gov/airfoils/AirfoilLicensing.html



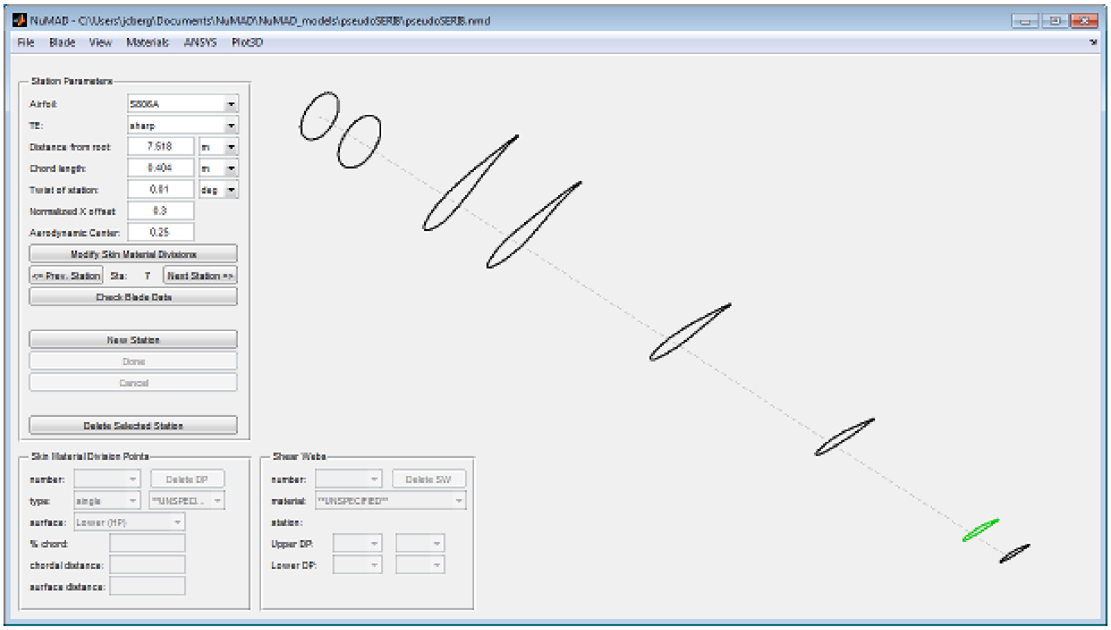
Station Parameters

Airfoil: S806A
TE: sharp
Distance from root: 7.925 m
Chord length: 0.330 m
Twist of station: 0 deg
Normalized X offset: 0.3
Aerodynamic Center: 0.25
Sta: 2

Station Parameters

Airfoil: circular
TE: round
Distance from root: 0.457 m
Chord length: 0.470 m
Twist of station: 20 deg
Normalized X offset: 0.5
Aerodynamic Center: 0.25
Sta:

Figure 11: Create additional stations by clicking "New Station", then editing parameters, and finally clicking "Done".



NuMAD - C:\Users\jcberg\Documents\NuMAD\NuMAD_models\pseudoSERI8\pseudoSERI8.nmd

Station Parameters

Airfoil: S806A
TE: sharp
Distance from root: 7.618 m
Chord length: 0.404 m
Twist of station: 0.01 deg
Normalized X offset: 0.3
Aerodynamic Center: 0.25
Sta: 7

Skin Material Division Points

number:
type: single "UNSPECI...
surface: Lower (HP)
% chord:
chordal distance:
surface distance:

Shear Webs

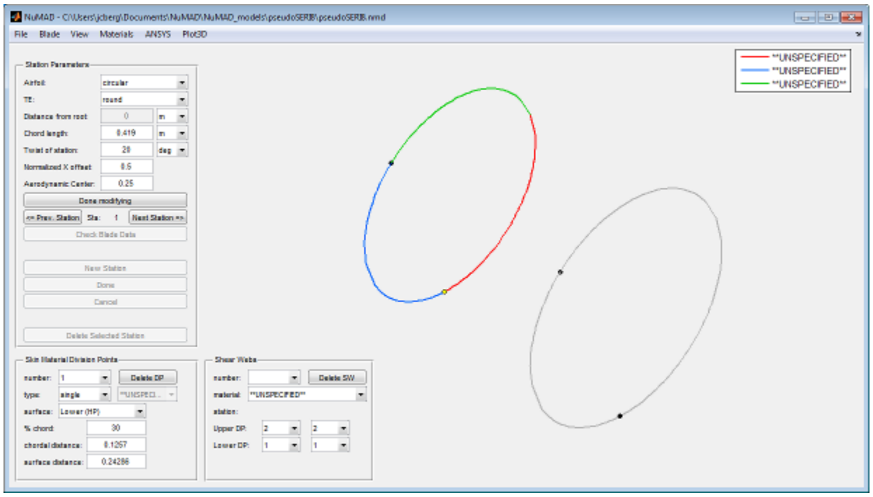
number:
material: "UNSPECIFIED"
station:
Upper DP:
Lower DP:

Figure 12: PseudoSERI8 stations defined.

Assign skin materials to the root (Station 1)

The outer blade geometry is now fully defined. Materials are next assigned to appropriate skin areas. You may press the "Check Blade Data" button to see the areas drawn between stations; note the model check reports that there are unspecified materials in the model.

Select the root station and click "Modify Skin Material Divisions" (alternatively, you may double-click the station). The root station along with the adjacent station will be displayed as shown in Figure 13. By default, the station has three segments with unspecified material. The extent of each segment is defined by Division Points (DPs).



NuMAD - C:\Users\jcberg\Documents\NuMAD\NuMAD_models\pseudoSERI8\pseudoSERI8.nmd

Station Parameters

Airfoil: circular

TE: round

Distance from root: 0 m

Chord length: 0.419 m

Twist of station: 28 deg

Normalized X offset: 0.5

Aerodynamic Center: 0.25

Sta: 1

Skin Material Division Points

number: 1

type: single **UNSPECI...

surface: Lower (HP)

% chord: 30

chordal distance: 0.1257

surface distance: 0.24286

Shear Webs

number:

material: **UNSPECIFIED**

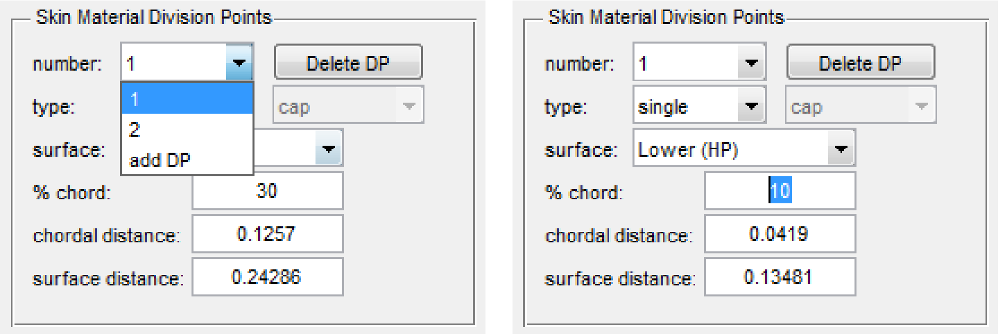
station:

Upper DP: 2 2

Lower DP: 1 1

Figure 13: Modifying the skin materials. There are three segments by default.

Modify the two existing DPs such that they are located at 10% chord on the lower and upper surfaces. First, choose DP number 1 from the dropdown menu and change the % chord from 30 to 10 (see Figure 14). Next, choose DP number 2 from the dropdown menu and change its % chord to 10.



Skin Material Division Points

number: 1

type:

surface:

% chord: 30

chordal distance: 0.1257

surface distance: 0.24286

Skin Material Division Points

number: 1

type: single cap

surface: Lower (HP)

% chord: 10

chordal distance: 0.0419

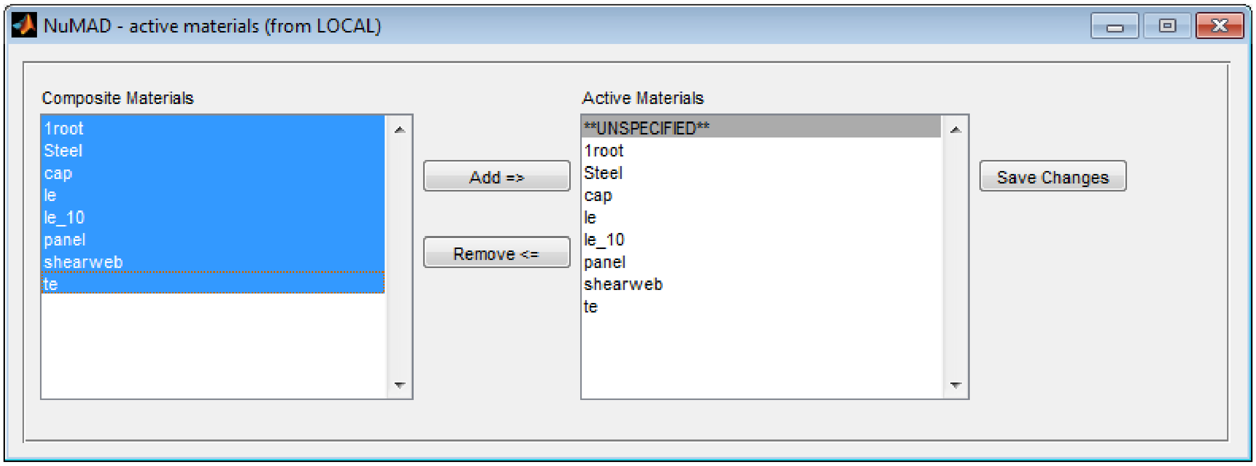
surface distance: 0.13481

Figure 14: Modifying the skin material Division Points.

Add four new DPs: two on the upper surface at 60 and 90% chord and two on the lower surface also at 60 and 90% chord. Add a DP by selecting “add DP” from the DP number dropdown menu. Be sure to click “Done” to finish creating the DP. The surface dropdown menu selects which side will receive the DP.

You should now have six DPs and seven segments with unspecified materials. A material can be assigned to a segment by right-clicking the segment. A material palette will be displayed. The list is currently empty.

Go to the “Materials >> Material Palette.” For this example, select all of the materials on the left and “Add” them to the list on the right as shown in Figure 15. Click the “Save Changes” button when you are finished.



Composite Materials

1root
Steel
cap
le
le_10
panel
shearweb
te

Active Materials

UNSPECIFIED
1root
Steel
cap
le
le_10
panel
shearweb
te

Figure 15: Modifying the list of active materials.

Now, when you right-click on a segment you will be able to select a material from the palette. Change the material of every segment in Station #1 to the "1root" material. The station should now look like Figure 16.

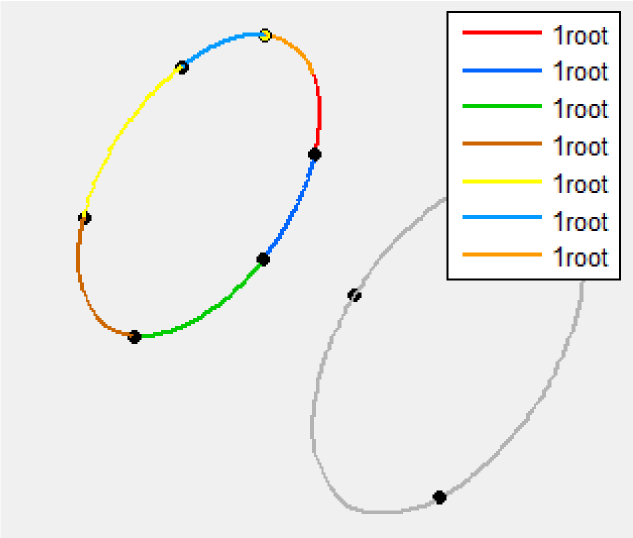


Figure 16: Material assignment for Station 1.

Assign skin materials to remaining stations

Repeat the same process to create DPs and assign materials for stations 2 through 8. Table 5 lists the DP locations for all of the blade stations. Modify existing DPs and create new DPs as needed. Remember to save the model periodically as you create DPs.

Assign materials according to the pattern given in
Table 6. The airfoil-shaped sections have four distinct areas. The leading edge area receives the "le" material. After the leading edge come the spar caps ("cap") on the upper and lower surfaces. Then, after the spar caps is "panel" material on the upper and lower surfaces. Finally, there is "te" material at the trailing edge of both surfaces. The tip station (#8) does not need any material

assignment because skin areas extend from one station to the next outboard station. Remember to save the model periodically as you assign materials.

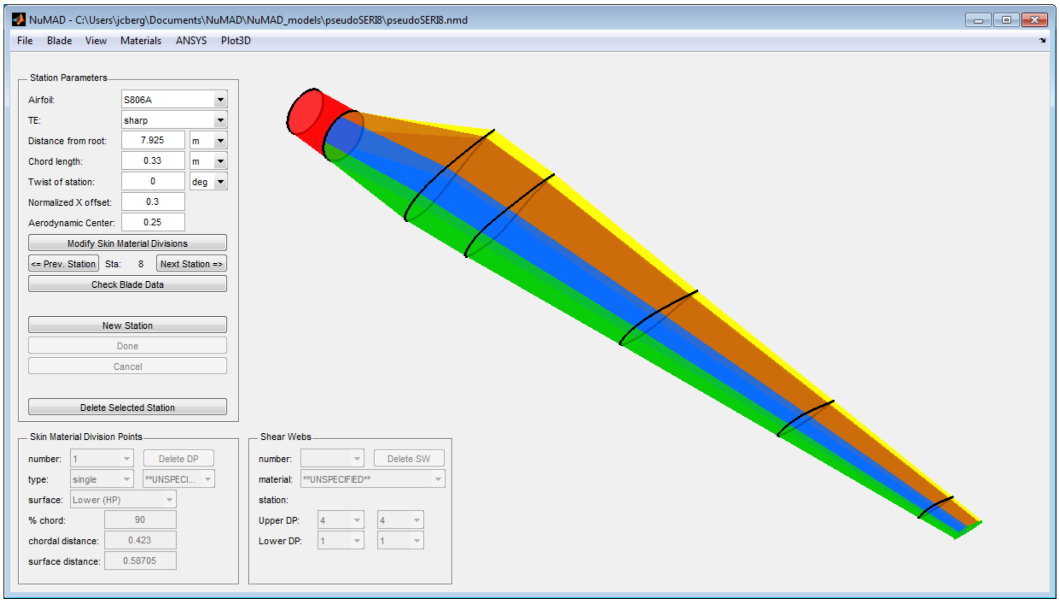
Table 5: DP locations for pseudoSERI8 model.

Stations	**Lower surface DPs - % chord**			**Upper surface DPs - % chord**		
1-2	90	60	10	10	60	90
3-8	90	50	20	20	50	90

Table 6: Material assignments for pseudoSERI8 model.

Segments are numbered from lower surface trailing edge, increasing toward leading edge, and wrapping around to upper surface trailing edge.							
Segments:	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Stations: 1	1root	1root	1root	1root	1root	1root	1root
2-7	te	panel	cap	le	cap	panel	te

The button labeled "Check Blade Data" performs a check on consistency between stations. In this example, we have thus far used only "single" DPs and therefore the blade check shouldn't find any problems unless one of the stations does not have 6 DPs or a material is unspecified. Perform a blade data check and fix any problems found. Figure 17 shows the skin areas drawn after a blade check.



NuMAD - C:\Users\jcberg\Documents\NuMAD\NuMAD_models\pseudoSERI8\pseudoSERI8.nmd

Station Parameters

- Airfoil: S806A
- TE: sharp
- Distance from root: 7.925 m
- Chord length: 0.33 m
- Twist of station: 0 deg
- Normalized X offset: 0.3
- Aerodynamic Center: 0.25
- Sta: 8

Skin Material Division Points

- number: 1
- type: single
- **UNSPECI...
- surface: Lower (HP)
- % chord: 90
- chordal distance: 0.423
- surface distance: 0.58705

Shear Webs

- number:
- material: **UNSPECIFIED**
- station:
- Upper DP: 4 4
- Lower DP: 1 1

Figure 17: PseudoSERI8 after a blade check.

Create a shear web

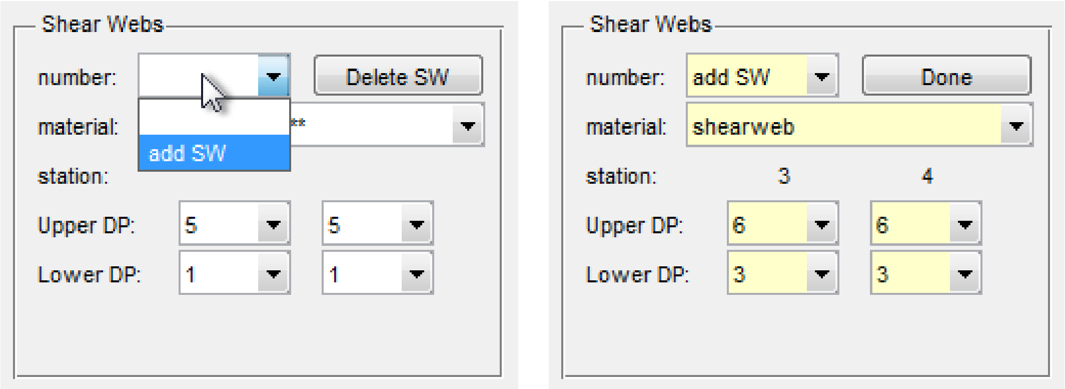
In NuMAD, shear webs are created by forming areas between stations using existing DPs. For the pseudoSERI8 blade, we will add new DPs within the spar cap segments and then create the shear web areas. Table 7 lists the new DPs which need to be added. Note that the new DPs will split existing line segments. The new line segments will each retain the same material as the line segment before it was split.

Table 7: New DPs for pseudoSERI8 shear web.

	% chord of new DPs	
Station	Lower surface	Upper surface
3	28	35
4	29	34
5	31	32.5
6	32	32

To create the shear web, select station 3 and go into station view (Modify Skin Material Divisions). In the Shear Webs panel, select "add SW" from the number dropdown menu (see Figure 18). A red outline will preview the location of the new shear web area. We want the shear web to extend between the new DPs that were recently created, so change the Upper DP to 6 for both station 3 and station 4 and change the Lower DP to 3 for both stations. Assign "shearweb" as the material and click "Done" to finish the shear web.

Repeat this process for Stations 4 and 5. The model should look like Figure 19 when you are finished.


Shear Webs
number: add SW
material: **
station:
Upper DP: 5 5
Lower DP: 1 1

Shear Webs
number: add SW
material: shearweb
station: 3 4
Upper DP: 6 6
Lower DP: 3 3

Figure 18: Creating a shear web area.

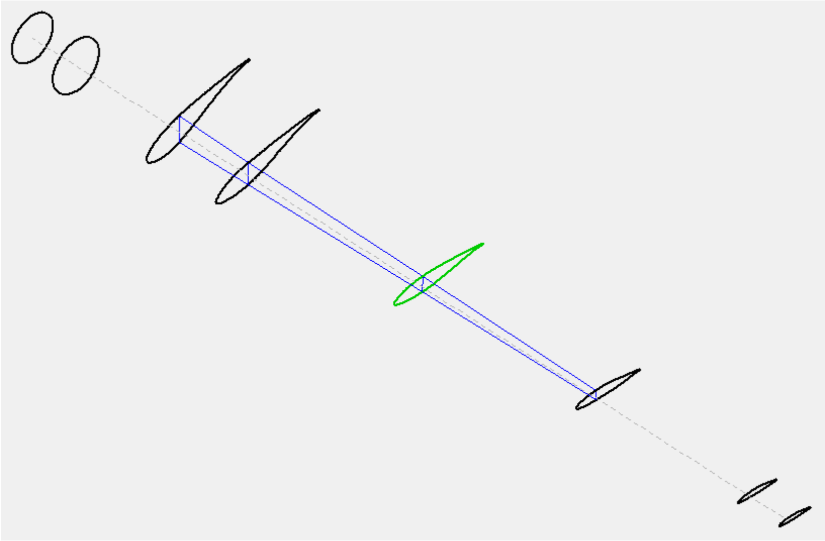
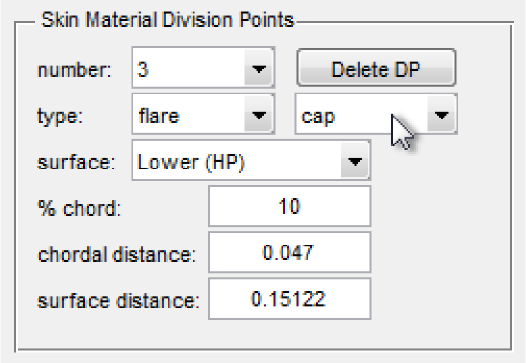


Figure 19: PseudoSERI8 with shear web.

A blade data check will show that the new DPs have created some mismatches between stations. There are two solutions to the problem: (1) create new DPs on the other stations also or (2) change the type of some of the DPs. We will use the second solution to demonstrate the “flare” and “double” types of DPs.

Select station 2 and prepare to modify its DPs. Select DP number 3 and change its type to “flare”. A flare allows the DP point to expand out into a new area and thus it requires a material assignment. Assign the “cap” material to this flare-DP (see Figure 20). Select DP number 4 and perform the same changes. These changes should resolve the mismatch between Stations 2 and 3.



Skin Material Division Points

number: 3

type: flare cap

surface: Lower (HP)

% chord: 10

chordal distance: 0.047

surface distance: 0.15122

Figure 20: Changing a DP's type to "flare" and assigning a material

To resolve the mismatch between stations 6 and 7, you will need to change the type of some DPs to “double”. A double allows a segment on the previous station to shrink down to a point at the double-DP. Select station 7 and change the type of the third and fourth DPs to double.

Run a blade data check to verify that all problems have been resolved.

Save the model – the pseudoSERI8 model will be used for subsequent analysis examples.

BLADE ANALYSIS EXAMPLES

This chapter explains the procedure for generating an ANSYS model from the NuMAD blade database and performing some basic analyses.

The primary function of NuMAD is to generate a text file of ANSYS APDL commands that are used to create the finite element in ANSYS. These commands are generated and stored in the file with default filename *shell7.src*. Effective users of the NuMAD and ANSYS combination must have experience with ANSYS Mechanical APDL[v] or must be able to learn about ANSYS Mechanical APDL such that they can interpret the commands found in *shell7.src*.

NuMAD only creates the APDL commands to create the blade model. It is the responsibility of the analyst to analyze the model in a manner that is consistent with their own design goals.

In this section we provide some basic examples of analyses that may be performed on the blade models. The starting point is the *pseudoSERI8* model created in the last chapter. If you have skipped the previous chapter, the *pseudoSERI8_complete* folder contains this model with everything completed up to the end of the blade modeling tutorial. Experienced users may want to use a different blade model and adapt the instructions as necessary.

ANSYS Output Options – shell7 Output

The first step before generating an ANSYS model is setting the ANSYS output options within NuMAD. This step generally includes:

- Specifying the blade boundary condition
- Choosing an element type and meshing options
- Selecting the output format(s)

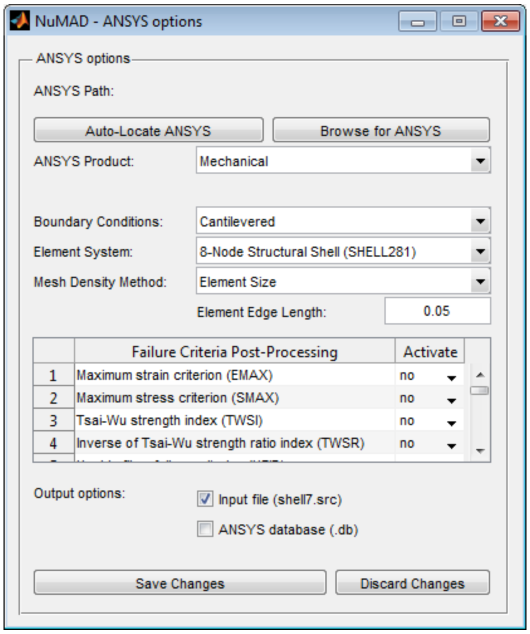
If the output format “ANSYS Database (.db)” is chosen, it may also be necessary to specify the path to the ANSYS executable and the name of the licensed ANSYS product.

Open the *pseudoSERI8* model within NuMAD and select Output Options from the ANSYS menu. The options window should look similar to Figure 21. For this tutorial, select the following options:

- Boundary Condition: Cantilevered
- Element System: SHELL281
- Meshing: Element Size, 0.05
- Failure Criteria Post-Processing: all deactivated by default
- Output format: shell7.src

Save the options changes.

[v] NuMAD is designed to create input scripts for use only in ANSYS Mechanical APDL. NuMAD is not directly compatible with the popular CAD-like modeling environment, ANSYS Workbench. It may possible, with care, to export the ANSYS Mechanical APDL models and import them into ANSYS Workbench. Be aware that past experience has shown that there can be problems associated with importing the composite layer information required for the definition of the Shell 181 and 281 elements in Workbench.


NuMAD - ANSYS options

ANSYS options

ANSYS Path:

ANSYS Product: Mechanical

Boundary Conditions: Cantilevered

Element System: 8-Node Structural Shell (SHELL281)

Mesh Density Method: Element Size

Element Edge Length: 0.05

	Failure Criteria Post-Processing	Activate
1	Maximum strain criterion (EMAX)	no
2	Maximum stress criterion (SMAX)	no
3	Tsai-Wu strength index (TWSI)	no
4	Inverse of Tsai-Wu strength ratio index (TWSR)	no

Output options:

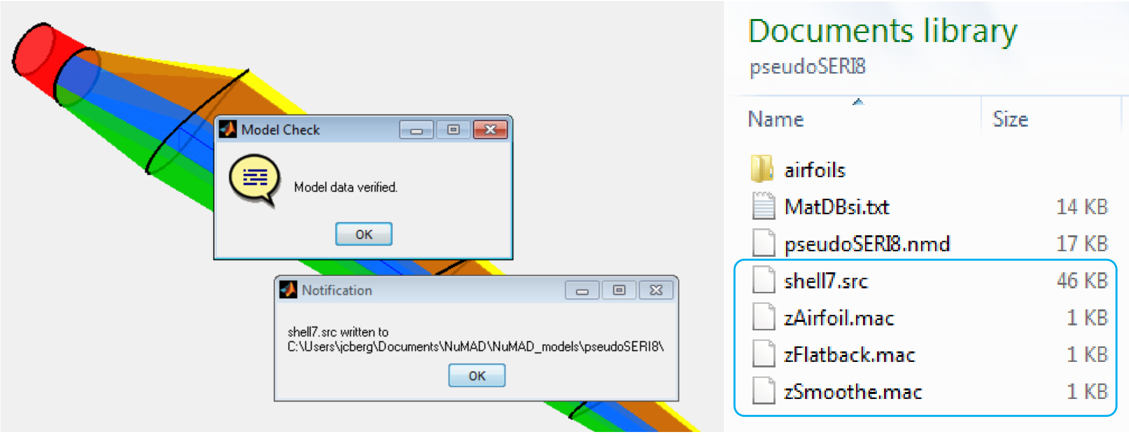
☑ Input file (shell7.src)

□ ANSYS database (.db)

Figure 21: ANSYS options window

Generate the ANSYS input file *shell7.src* by selecting “ANSYS >> Generate Now” from the menus. NuMAD will first perform a model check to ensure the number of DPs per station is consistent and all areas have been assigned a material. If the model passes these checks, NuMAD then generates the output files in the project folder. If your model did not pass the blade check, attempt to fix the problems or use the *pseudoSERI8_complete* model packaged with NuMAD.

As shown in Figure 22, a successful model generation will produce two user notifications and four new files. The three **.mac* files are macros used by the *shell7.src* ANSYS input file.


Model Check

Model data verified.

Notification

shell7.src written to
C:\Users\jcberg\Documents\NuMAD\NuMAD_models\pseudoSERI8\

Documents library
pseudoSERI8

Name	Size
airfoils	
MatDBsi.txt	14 KB
pseudoSERI8.nmd	17 KB
shell7.src	46 KB
zAirfoil.mac	1 KB
zFlatback.mac	1 KB
zSmoothe.mac	1 KB

Figure 22: Notifications and Files generated by the ANSYS "Generate Now" command.

Reading the shell7 ANSYS Input

We recommend using the ANSYS Mechanical APDL Product Launcher when starting or continuing an ANSYS session. The product launcher clearly shows which product license, working directory, and job name will be used during the session.

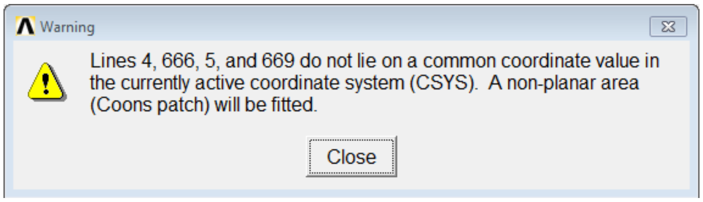
For this tutorial we are using the ANSYS Mechanical APDL environment with an ANSYS Mechanical license. You may need to select a different license option if "ANSYS Mechanical" is not available.

The working directory must be set to the NuMAD project directory so that the macros (*zAirfoil.mac*, *zFlatback.mac*, *zSmoothe.mac*) can be found while reading the shell7 file. For this tutorial, set the working directory to the *pseudoSERI8* project folder (use the "Browse" button to get the full path) and set the job name to *master*.

After ANSYS Mechanical APDL has launched, complete these steps to load the blade model from the *shell7.src* file.

1. Go to the ANSYS File menu and select "Read Input from …"
2. Select the *shell7.src* file within the *pseudoSERI8* working directory and click "OK"
3. Wait for the script to finish generating the blade geometry

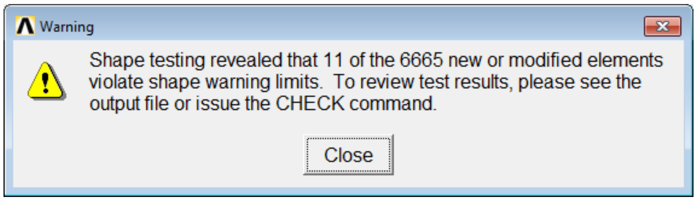
While the geometry is being generated, a number of warnings will appear when areas are created (see Figure 23 for example warning). The "Coons patch" warnings are a normal part of the geometry generation due to the nature of the surfaces used to define the surface of the blade.



Lines 4, 666, 5, and 669 do not lie on a common coordinate value in the currently active coordinate system (CSYS). A non-planar area (Coons patch) will be fitted.

Figure 23: Non-planar area warning

Another type of warning is the "shape test" warning shown in Figure 24 that may occur after the blade geometry has been created and the automatic meshing routine has commenced. The analyst needs to understand and address this warning as appropriate for their particular model, choice of finite element, etc. For this tutorial we can ignore this type of warning.



Shape testing revealed that 11 of the 6665 new or modified elements violate shape warning limits. To review test results, please see the output file or issue the CHECK command.

Figure 24: Element shape test warning

Inspecting the Blade Geometry

Now that the shell7 input script has completed, we will show some methods of inspecting the blade geometry. First, plot the skin areas by selecting "Areas" from the ANSYS "Plot" menu. The areas can be color-coded according to material section number by choosing

- "PlotCtrls >> Numbering"
- Set "Elem / Attrib numbering" to "Section numbers"
- Set "[/NUM] Numbering shown with" to "Colors only"

Next, plot the element mesh by selecting "Elements" from the ANSYS "Plot" menu. The plot reveals that the mesh near the leading edge needs to be refined to fully capture the geometry (Figure 25). One simple solution to this problem is placing a DP at the leading edge of each station.[vi]

Return to NuMAD and add leading edge DPs at every station. After generating a new shell7 script and clearing the ANSYS workspace (File >> Clear & Start New), read the input file once again and inspect the new mesh. The additional leading edge DPs should fix the meshing problems.

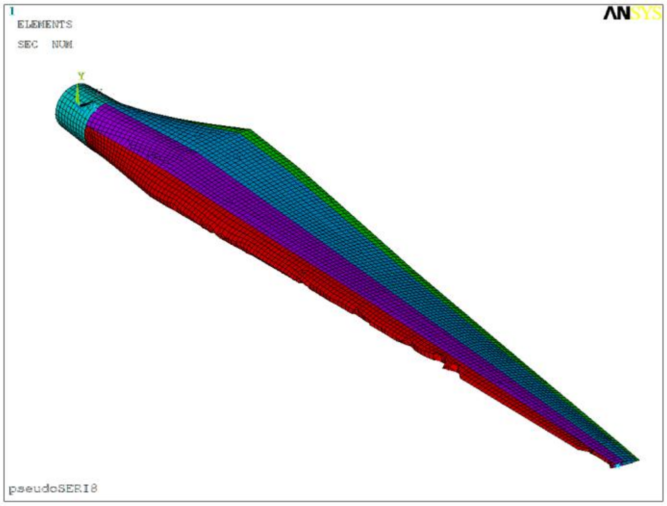


Figure 25: Meshing problems at leading edge can be resolved with additional DPs

Finally, inspect a slice of the blade. These steps show how to select a portion of the blade based on span location and visualize the shell thickness of the material definitions.

- From the ANSYS menu, choose "Select >> Entities"
- Choose "Nodes" and "By Location"
- Z coordinates 2, 2.2
- APPLY
- Choose "Elements" and "Attached to" Nodes

[vi] Placement of DP's at the leading edge of each section is highly recommended, and is required if PreComp analyses will be initiated from NuMAD.

- APPLY, REPLOT, OK
- From the ANSYS menu, choose “PlotCtrls >> Style >> Size and Shape”
- Turn “[/ESHAPE]” On
- OK

This concludes our demonstration of a few methods to inspect the blade geometry. Because the ESHAPE option can slow down visualization, we recommend turning that feature off before continuing. We next show how to perform some typical blade analyses.

Analysis Examples

Static Analysis

Static analysis can be used to find isolated locations in the blade that may experience high strain values and can be used to understand ply failure predicted by any defined material failure criteria.

The following example shows a static analysis with load applied at the tip of the blade in the flap-wise direction. A more realistic example of static loading would require a pressure distribution or lines of forces consistent with an actual blade loading.

Before beginning the analysis, ensure that all nodes and elements are selected by choosing “Select >> Everything” from the ANSYS menu.

The following commands assign the number of the highest numbered node (the blade tip node) to the variable *var1*. The force is then applied to that node and the problem is solved.

```
/SOLU
ANTYPE,STATIC
*GET,var1,node,0,num,max
F,var1,FY,1
SOLVE
FINISH
/POST1
RSYS,SOLU
LAYER,0
PLESOL,EPEL,X,0,50
```

Results for various layers in the model are viewed by using the LAYER,*N* command prior to plotting results, where *N* is the layer number.

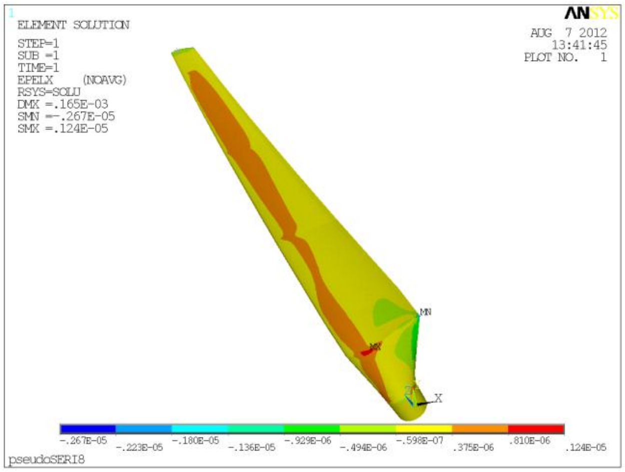


Figure 26 Example static analysis results. Material x-component of elastic strain in the elements

Buckling Analysis

Both eigenvalue (linear) buckling and nonlinear buckling loads can be computed in ANSYS using the NuMAD-generated model. Keep in mind that eigenvalue buckling predicts a buckling load that is non-conservative. The nonlinear buckling problem can produce more descriptive and perhaps more accurate results but the analysis is quite complex and time-consuming than the linear analysis.

The load applied in this linear buckling example is a simple point load at the tip of the blade in the flap-wise (downwind) direction in order to maintain simplicity in this example. A more realistic loading would represent a distributed force along the entire blade skin.

First, compute the loads using a static analysis. In this case, the load is a unity flap-wise point load at the tip of the blade.

```
/SOLU
ANTYPE,STATIC
PSTRES,ON
*GET,var1,node,1,num,max
F,var1,FY,1
SOLVE
FINISH
```

Solve the buckling problem. Use the same solver that was chosen above in the modal analysis (Block Lanczos). Extract 10 buckling modes and expand the mode shapes.

```
/SOLU
ANTYPE,BUCKLE
BUCOPT,LANB,10
SOLVE
FINISH
```

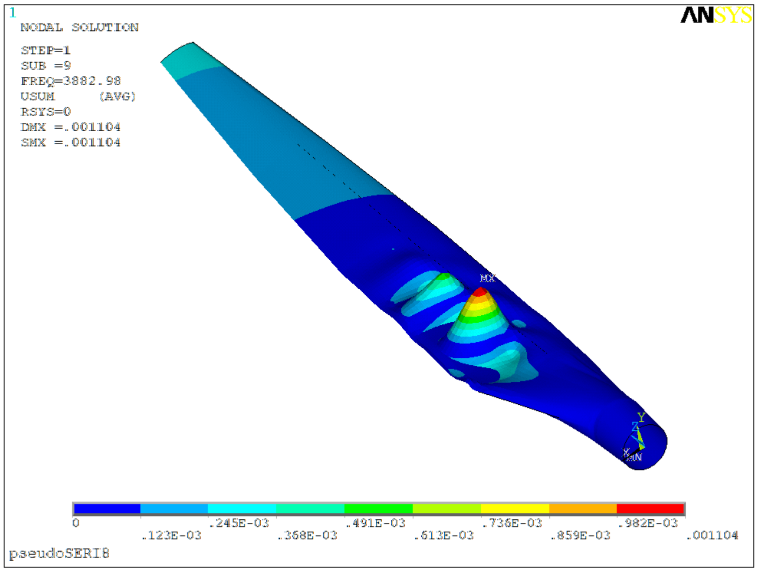


Figure 27: Eigenvalue buckling mode shape (Substep 9) in aft panel due to flap-wise loading.

The point load at the tip of the blade that induces the buckling mode shown in Figure 27 is found to be 3.88 kN in the positive Y-direction.

NUMAD INTERFACE REFERENCE

The following sections explain the functionality and capabilities of NuMAD in detail.

File Menu

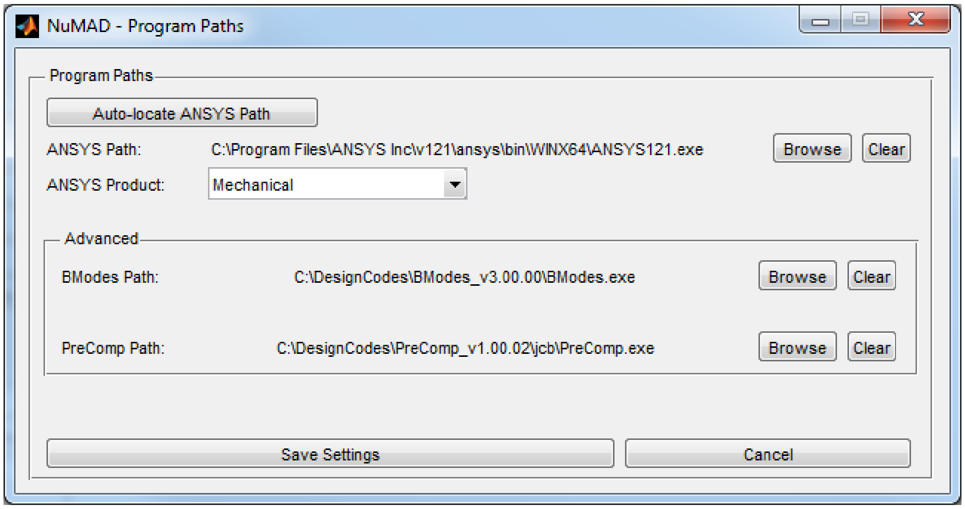
The “File” menu provides commands for creating a new model, opening an existing model, converting an older format NuMAD model, saving a model, and exiting NuMAD.

The “New Model” command resets the NuMAD interface and creates a new blade model with two circular sections which define the initial geometry. The user must save the model to create a new directory structure on disk. The model uses the global airfoil and material databases until the user saves the model, at which point local copies of these databases are made in the job directory.

The “Open Model…” command allows the user to retrieve a previously created model. A file browser window allows the user to navigate the file system and, by default, only files with the **.nmd* extension are displayed. The user loads a specific job by selecting and opening the *jobname.nmd* file.

The “Save” and “Save As” options allow the user to save model changes to disk. “Save As” allows the user to save changes under a new jobname, thereby creating a new job directory with new copies of local databases. The old job still exists but NuMAD begins using the new job directory as the current working directory. “Save” updates files in the current job directory.

The “Configure Paths…” option is where the user points NuMAD to important paths required for code interactions: 1) the ANSYS executable and product code, 2) the BModes executable and 3) the PreComp executable. BModes and PreComp are required for creation of section properties using the NuMAD database.



ANSYS Path: C:\Program Files\ANSYS Inc\v121\ansys\bin\WINX64\ANSYS121.exe

ANSYS Product: Mechanical

BModes Path: C:\DesignCodes\BModes_v3.00.00\BModes.exe

PreComp Path: C:\DesignCodes\PreComp_v1.00.02\jcb\PreComp.exe

Figure 28 Configure paths dialog box.

A NuMAD model saved before 2011 must be converted to the new format before the user can open it. A file menu option called “Convert Legacy NuMAD” performs this conversion.

The “XLS-to-NuMAD…” option is discussed in detail in the Sharing Models

During the course of creating a blade model, you may need to share blade models with colleagues, either for collaboration or for troubleshooting. Following is a list of the files that are needed to most effectively transfer your blade project to another individual:

If using the standard GUI input approach:

- *Airfoils* folder and all contents
- *MatDBsi.txt* file
- **.nmd* file
- *zAirfoil.mac*, *zFlatback.mac*, *zSmoothe.mac* files
- optional: *bpe_station_ids.txt*

If using the Excel input approach (described on page 62 in the section on Read Excel-Based Input File):

- *Airfoils* folder and all contents
- *NuMAD.xls(x)* file
- *zAirfoil.mac*, *zFlatback.mac*, *zSmoothe.mac* files
- optional: *bpe_station_ids.txt*

NuMAD Settings Files

NuMAD stores a few files in the Windows %APPDATA% location. This location is easily viewed by typing %APPDATA% in the Windows Explorer search bar, as shown in Figure 46. There are two files stored in this location in the NuMAD folder: *settings.txt* and *ansys-productvars.txt*

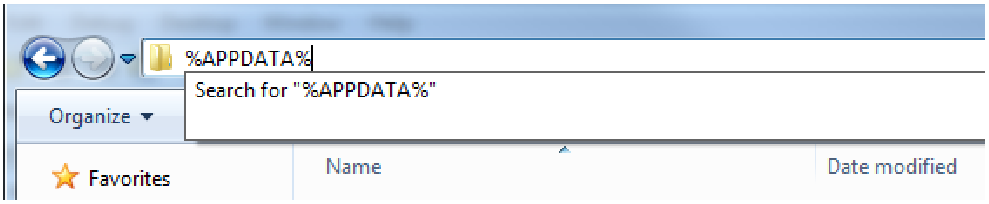


Figure 46 Viewing the APPDATA folder in Windows.

Settings File

This file contains various NuMAD system settings and default settings:

- ANSYS executable path
- ANSYS product code
- BModes executable path
- PreComp executable path
- GUI layout and size settings
- Path of last blade project

A simple way to reset all of these settings is to delete the file and then let NuMAD regenerate the file when it next opens.

ANSYS Product Variables

This file lists the ANSYS product variable names available in the NuMAD dropdown lists. You may manually edit this file to add/remove entries or change the description as you please. If you delete this file, the default content will regenerate when NuMAD restarts.

Advanced Capabilities section of this manual.

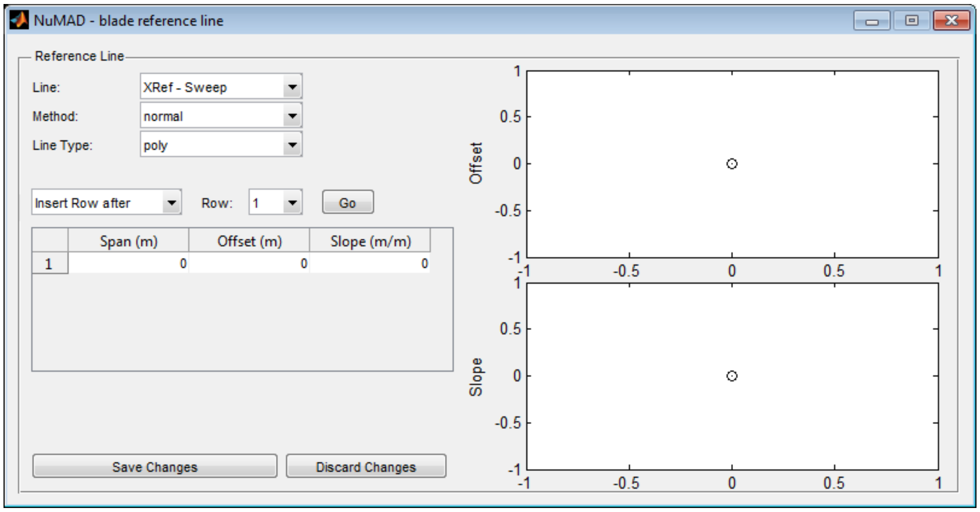
Blade Menu

The "Blade" menu allows the user to select the rotor's direction of rotation and edit the blade reference line.

Blade Reference Line

The Blade Reference Line is the reference or datum for all blade stations. A station's x-offset and twist are defined in relation to this line. By default, the reference line is straight, in which case it is equivalent to the blade pitch axis. NuMAD allows the blade designer to curve the blade reference line and thereby create a blade with sweep or pre-bend.

The Blade Reference Line interface, accessible from the NuMAD "Blade" menu, is shown in Figure 29 with the initial (straight-line) configuration. Two line definitions are available which correspond to the two directions a blade can curve. The x-reference line produces sweep (lead-lag curve) in a blade and the y-reference line produces pre-bend (flapwise curve) in a blade. The first menu dropdown selects which line to modify. The "method" dropdown refers to the method used to offset the blade stations. In the "shear" method, stations remain parallel to the root plane as they follow the reference line. In the "normal" method, each station is perpendicular to the local tangent of the reference line. The two methods are compared in Figure 30. Finally, the "line type" dropdown determines how the data table (discussed shortly) is interpreted.



NuMAD - blade reference line

Reference Line

Line: XRef - Sweep

Method: normal

Line Type: poly

Insert Row after   Row: 1   Go

	Span (m)	Offset (m)	Slope (m/m)
1	0	0	0

Figure 29: Interface for editing the Blade Reference Line.

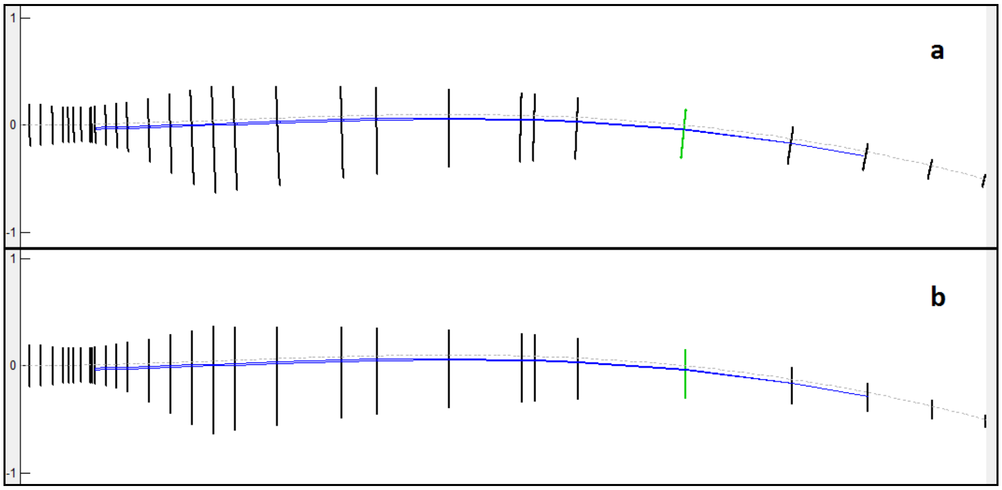
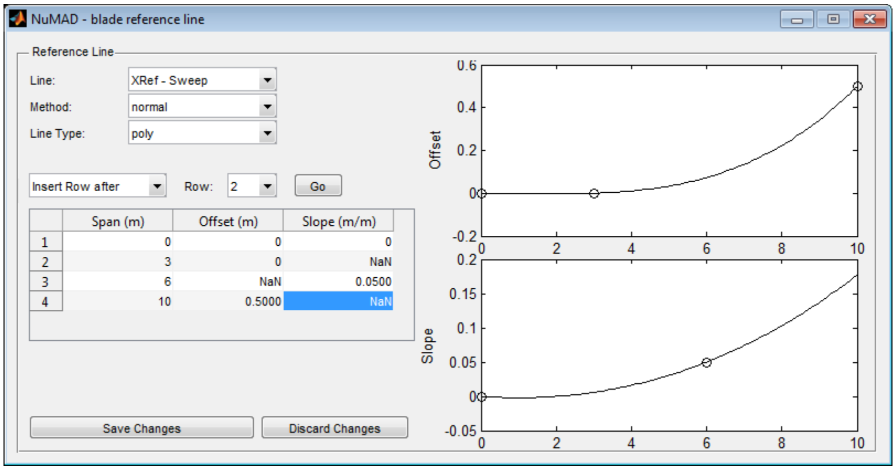


Figure 30: Blade Reference Line deformation methods include (a) "normal" and (b) "shear"

Each curve of the reference line is defined by a table of points the curve must pass through. The data table has three columns: span, offset, and slope. With line type “poly”, the offset and slope are constraints the curve must satisfy at the given spanwise locations. The user may specify offset, or slope, or both offset and slope at a given point. Enter “NaN” to leave a parameter unspecified. Type “poly” will attempt to fit a minimum order polynomial which satisfies all given constraints. With line type “spline” or “pchip”, the offset constraints are satisfied with a piecewise polynomial; however, the slope constraints are not satisfied.

Table rows can be added, moved, and deleted by using the set of controls directly above the table. For the following example, you must insert three rows after Row 1 so that there are four rows in the table.

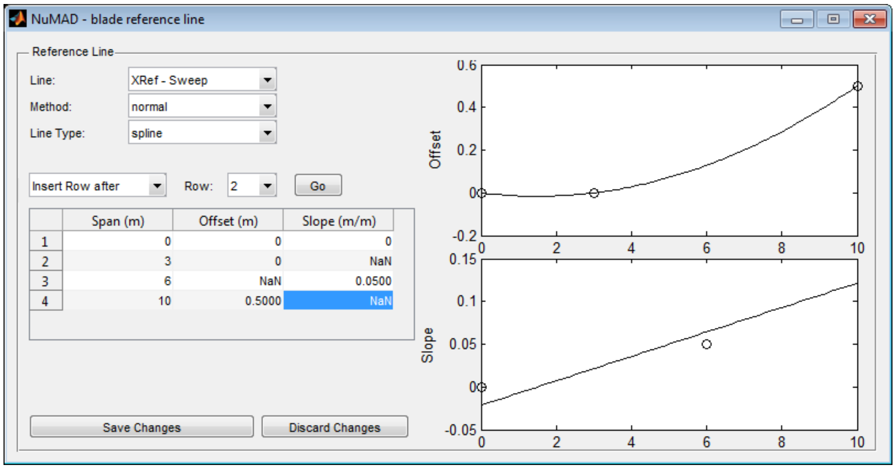
Edit the data table as shown in Figure 31 (this example is for illustration purposes only and does not represent a recommended blade shape). This table specifies five constraints: zero offset and slope at the root, zero offset at 3m span, 0.05 slope at 6m span, and 0.5m offset at 10m span. Note that circles are plotted at the constraints. Figure 32 and Figure 33 repeat the same example using the spline and pchip line types.



Line: XRef - Sweep
Method: normal
Line Type: poly
Row: 2

	Span (m)	Offset (m)	Slope (m/m)
1	0	0	0
2	3	0	NaN
3	6	NaN	0.0500
4	10	0.5000	NaN

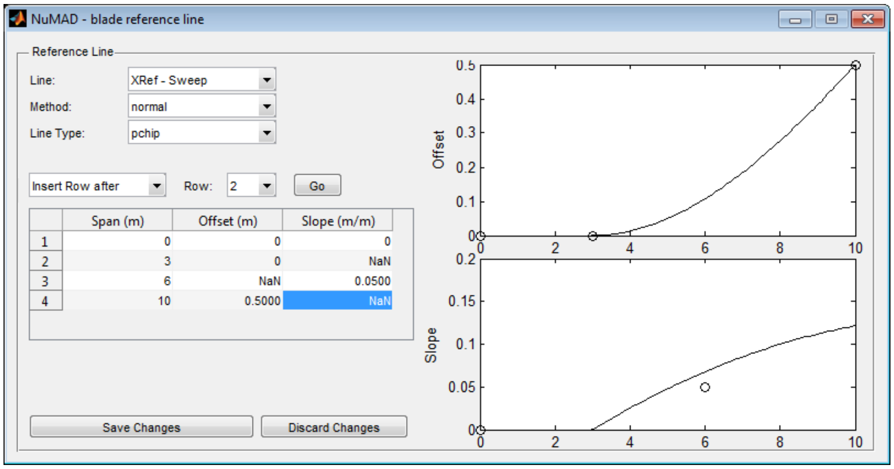
Figure 31: Blade Reference Line example using "poly" type.



Line: XRef - Sweep
Method: normal
Line Type: spline
Row: 2

	Span (m)	Offset (m)	Slope (m/m)
1	0	0	0
2	3	0	NaN
3	6	NaN	0.0500
4	10	0.5000	NaN

Figure 32: Blade Reference Line example using "spline" type.



Line: XRef - Sweep
Method: normal
Line Type: pchip
Row: 2

	Span (m)	Offset (m)	Slope (m/m)
1	0	0	0
2	3	0	NaN
3	6	NaN	0.0500
4	10	0.5000	NaN

Figure 33: Blade Reference Line example using "pchip" type.

Materials Menu

The “Materials” menu provides access to the material database. Because the collection of materials can grow quite large, a material palette provides a way to select a subset of materials shown while editing the blade model. Materials must be present in the palette for assignment to the blade.

Materials Database

NuMAD uses a database of material properties so that this information need not be redefined with each new model. The material definitions are simple text files in an XML format.

Master Materials Database

The master materials database is located in the root NuMAD directory. The database is comprised of a single file, *MatDBsi.txt*. Though this file could be modified with a text editor, facilities for modifying the materials database have been included in NuMAD to reduce the chance of file corruption.

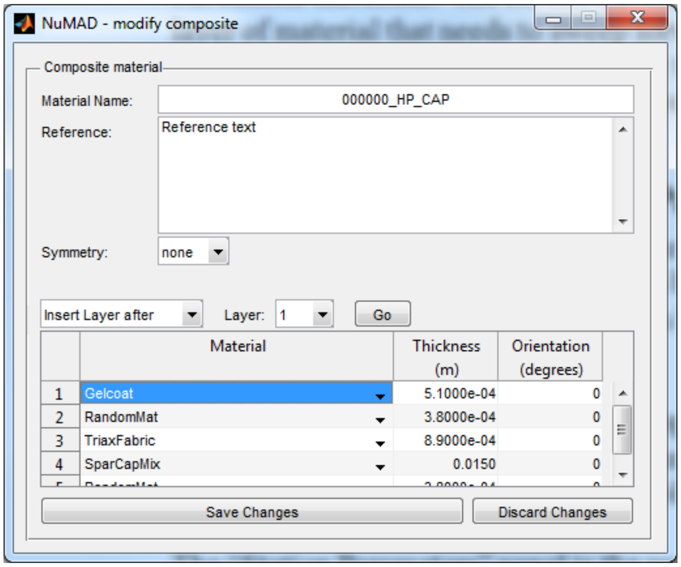
The master materials database file is copied from the root NuMAD directory to a blade project folder each time a new blade project is initiated and saved.

Local Materials Database

The master materials database is located in the blade project folder. The database is comprised of a single file, *MatDBsi.txt*. Once present in the blade project folder, this file supersedes the master materials database file.

Material Properties Input

Typical usage of materials in the current version of NuMAD first involves definition of the materials that comprise individual fabric layers. These materials can be isotropic or orthotropic. Then, one or more materials are combined into a composite where the layer thickness, orientation, and sequence are defined (Figure 34). Finally, composites are placed into the Material Palette so that they can be assigned to regions on the blade model.



NuMAD - modify composite

Composite material

Material Name: 000000_HP_CAP

Reference: Reference text

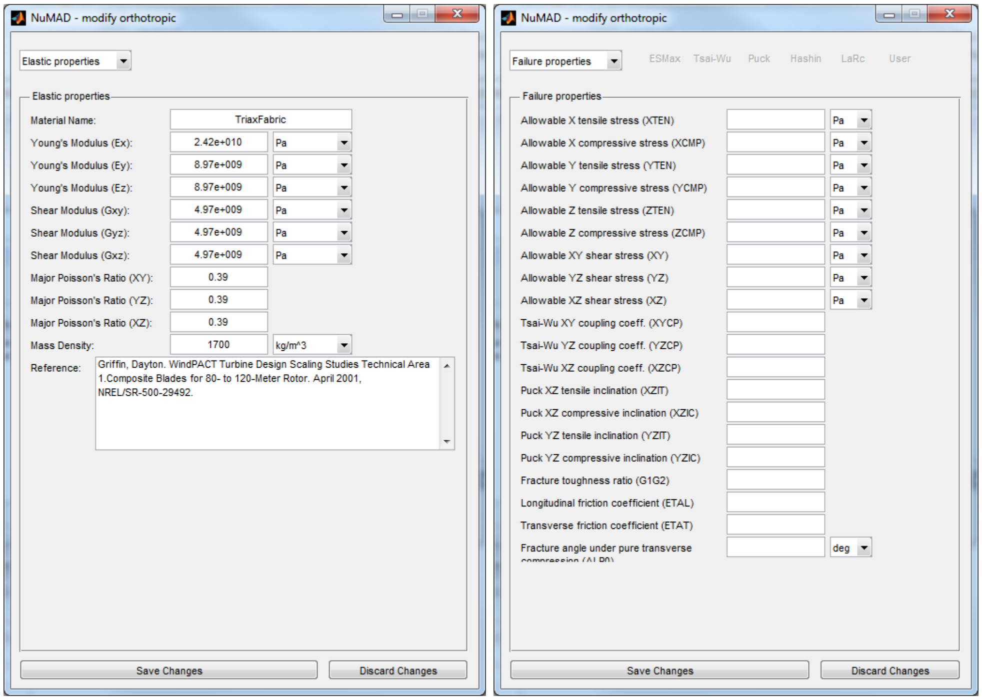
Symmetry: none

Insert Layer after Layer: 1

	Material	Thickness (m)	Orientation (degrees)
1	Gelcoat	5.1000e-04	0
2	RandomMat	3.8000e-04	0
3	TriaxFabric	8.9000e-04	0
4	SparCapMix	0.0150	0

Figure 34 Composite setup dialog window.

Materials and their properties are created, managed, and deleted in the materials database dialog window. The dialog is accessible through the “Materials >> Materials Database” menu item. Double-clicking on an orthotropic material shows you the properties of the material, as seen in Figure 35. Figure 35b shows a series of fields for entry of material failure criteria. If activated in the “ANSYS Options” prior to model generation, these criteria are used by ANSYS to compute material failure indices[vii]. As parameters are entered in sufficient detail, the various failure criteria names turn from grey to black.



Elastic properties

Material Name: TriaxFabric
Young's Modulus (Ex): 2.42e+010 Pa
Young's Modulus (Ey): 8.97e+009 Pa
Young's Modulus (Ez): 8.97e+009 Pa
Shear Modulus (Gxy): 4.97e+009 Pa
Shear Modulus (Gyz): 4.97e+009 Pa
Shear Modulus (Gxz): 4.97e+009 Pa
Major Poisson's Ratio (XY): 0.39
Major Poisson's Ratio (YZ): 0.39
Major Poisson's Ratio (XZ): 0.39
Mass Density: 1700 kg/m^3
Reference: Griffin, Dayton. WindPACT Turbine Design Scaling Studies Technical Area 1.Composite Blades for 80- to 120-Meter Rotor. April 2001, NREL/SR-500-29492.

Failure properties

Allowable X tensile stress (XTEN) Pa
Allowable X compressive stress (XCMP) Pa
Allowable Y tensile stress (YTEN) Pa
Allowable Y compressive stress (YCMP) Pa
Allowable Z tensile stress (ZTEN) Pa
Allowable Z compressive stress (ZCMP) Pa
Allowable XY shear stress (XY) Pa
Allowable YZ shear stress (YZ) Pa
Allowable XZ shear stress (XZ) Pa
Tsai-Wu XY coupling coeff. (XYCP)
Tsai-Wu YZ coupling coeff. (YZCP)
Tsai-Wu XZ coupling coeff. (XZCP)
Puck XZ tensile inclination (XZIT)
Puck XZ compressive inclination (XZIC)
Puck YZ tensile inclination (YZIT)
Puck YZ compressive inclination (YZIC)
Fracture toughness ratio (G1G2)
Longitudinal friction coefficient (ETAL)
Transverse friction coefficient (ETAT)
Fracture angle under pure transverse deg

(a) (b)

Figure 35 Properties of an orthotropic material: (a) elastic properties and (b) failure properties.

Layer Orientation

The scheme of fiber orientations is shown in Figure 36. Zero-degree fibers align with the blade reference axis. The rotation of the fibers' orientation is opposite in the global scheme of the entire blade on the HP surface compared to the LP surface of the model. This is due to the fact that the element coordinate systems all have the x-axes pointing along the blade span, toward the tip, and the element z-axes, normal to each element, points to the interior of the blade.

If layer orientations other than 0 or 90 degrees are used then the user must specify two different, but similar materials: one for the top surface and one for the bottom. For example, if there is a layer of material that needs to sweep forward 20 degrees on both top and bottom surfaces then the user should define a material for the top with layer orientation of +20 degrees as well as another material for the bottom with layer orientation of -20 degrees.

[vii] As of the writing of this manual, ANSYS has included several new failure criteria in the release of R14: Puck, Hashin, and LaRc. Implementation of plots for these failure criteria through the ANSYS GUI has not been set up. All failure criteria may be plotted on the model using text commands. See the ANSYS Help documentation for more information on plotting of failure criteria using PLESOL.

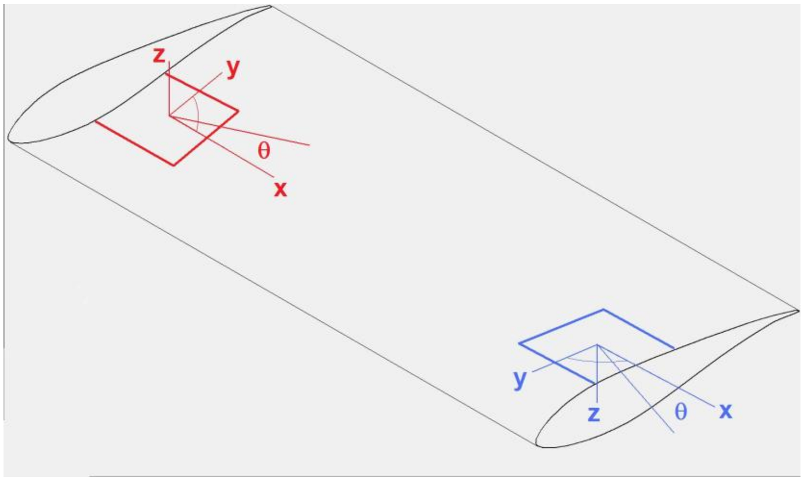


Figure 36 Schematic indicating the sense of the element coordinate systems and layer orientations on the top (blue) and bottom (red) surfaces of the blade.

Main Viewport and Control Panels

A large portion of the NuMAD interface is devoted to displaying and manipulating the blade geometry. The main viewport switches between two modes depending on the type of action being performed. These two modes can be described as blade view and station view.

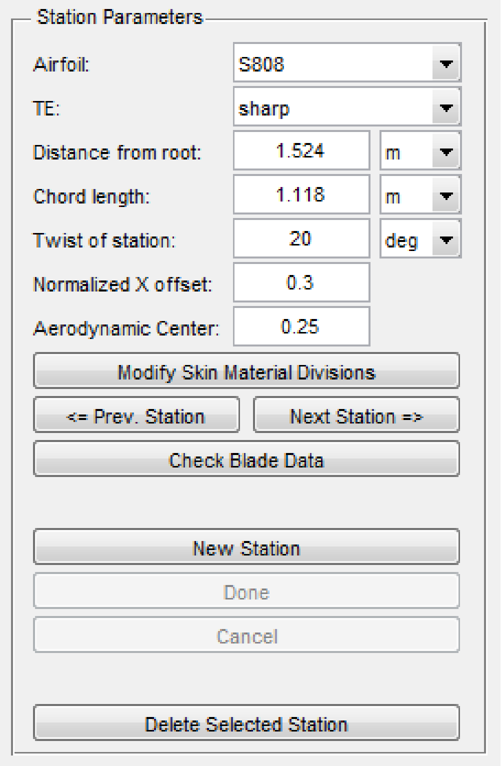
Blade View

In blade view, the entire geometry is displayed. Station outlines (airfoils) are drawn to scale in their actual orientation which includes any offsets and rotations due to sweep and prebend. This view is intended for defining the outer blade geometry.

The “Station Parameters” panel is the user interface for creating a station, selecting its airfoil, and defining its location, size, and orientation. Figure 37 shows this control panel.

The “Airfoil” dropdown lists the airfoils available in the database. The blade view will update immediately when the airfoil changes. Note that the chordal or surface distance of DPs may need to be redefined when the airfoil is changed.

The “TE” dropdown determines how ANSYS handles the construction of lines at the trailing edge. A sharp trailing edge is typical of most airfoils – the upper and lower surfaces form an acute angle. The “flat” option is used only for “flatback” airfoils. The “round” option is intended for circular sections or other cases in which the slope should be continuous around the trailing edge. See the section on Trailing Edge Types on page 55 for more information.



Station Parameters

Airfoil: S808

TE: sharp

Distance from root: 1.524 m

Chord length: 1.118 m

Twist of station: 20 deg

Normalized X offset: 0.3

Aerodynamic Center: 0.25

Figure 37: Station Parameters control panel.

Station View

In station view, the focus is on a single station and one or two adjacent stations are displayed for context. The distance between stations is scaled so that the group of stations can be viewed together. In addition, offsets and rotations due to blade sweep and prebend are not displayed. The station outline is divided into segments which show the distribution of materials. This view is intended for assigning materials to blade areas. Shear webs are also created while in station view.

Skin Material Division Points

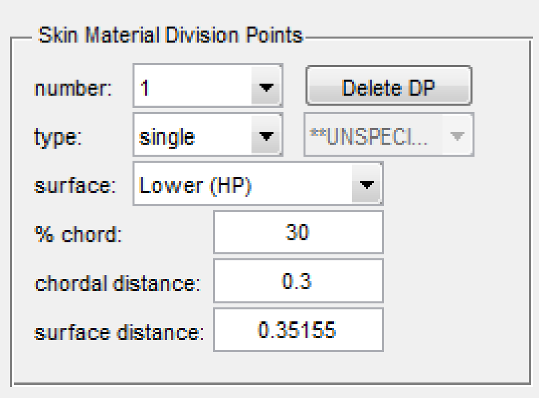
Material division points (DP's) are used to define the location of material boundaries around the surface of each station. DP's are defined in the "Skin Material Division Points" dialog (Figure 38) while in Blade Station View (after pressing the button for "Modify Skin Material Divisions").

DP numbering begins at the HP trailing edge with DP 1, proceeds over the HP surface to the leading edge and ends at the LP trailing edge. Add a new DP by choosing "add DP" from the "number" dropdown menu. By default, new DP's are added at the nose of the airfoil.

Most DP types are "single," meaning that the DP maps to both a single DP on the next inboard station as well as a single DP on the next outboard station. Figure 39 illustrations the other DP types. These DP types are used to form triangular areas on the blade.

- Double – maps two DP's on one station to a single DP on the next outboard station; material assignment in the triangular surface area is determined by the assignment of the material segment between the two DP's in the inboard station.
- Hourglass – maps two DP's on one station, to a single DP on the next outboard station, and then two DP's on the next outboard station after that; material assignment in the inboard triangular surface area is determined by the assignment of the material segment between the two DP's in the inboard station, while material assignment in the outboard triangular surface area is made in the "Skin Material Division Points, type" dialog.
- Flare - maps one DP on one station to two DP's on the next outboard station; material assignment in the triangular surface area is determined by the assignment made in the "Skin Material Division Points, type" dialog.

Note: It is not good practice to use DP types other than "single" at zero-percent chord (i.e. at the leading edge).



Skin Material Division Points

number: 1

type: single

**UNSPECI...

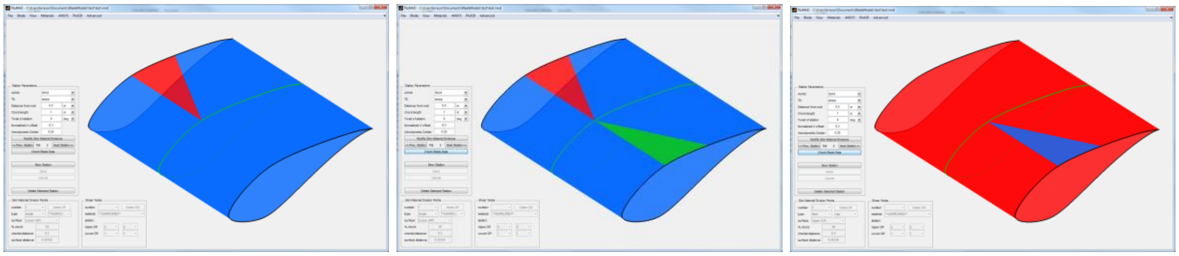
surface: Lower (HP)

% chord: 30

chordal distance: 0.3

surface distance: 0.35155

Figure 38 Skin material division points dialog.

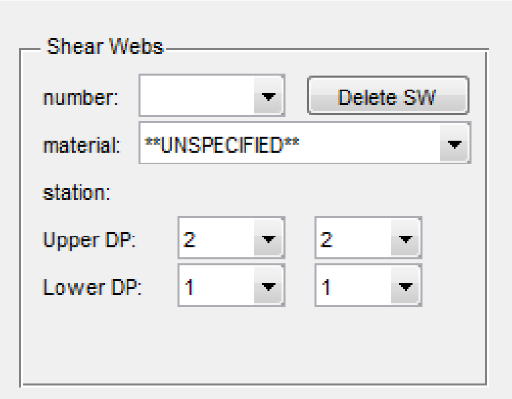


(a) (b) (c)

Figure 39 Illustration of DP Types: (a) double, (b) hourglass, (c) flare.

Shear Webs

With DP's created and located, shear webs can be defined. Figure 38 shows the "Shear Webs" dialog while in Blade Station View (after pressing the button for "Modify Skin Material Divisions"). Use the fields in this dialog to add shear webs, assign materials to shear webs, and locate the shear webs using the existing DP's.



number:

material: **UNSPECIFIED**

station:

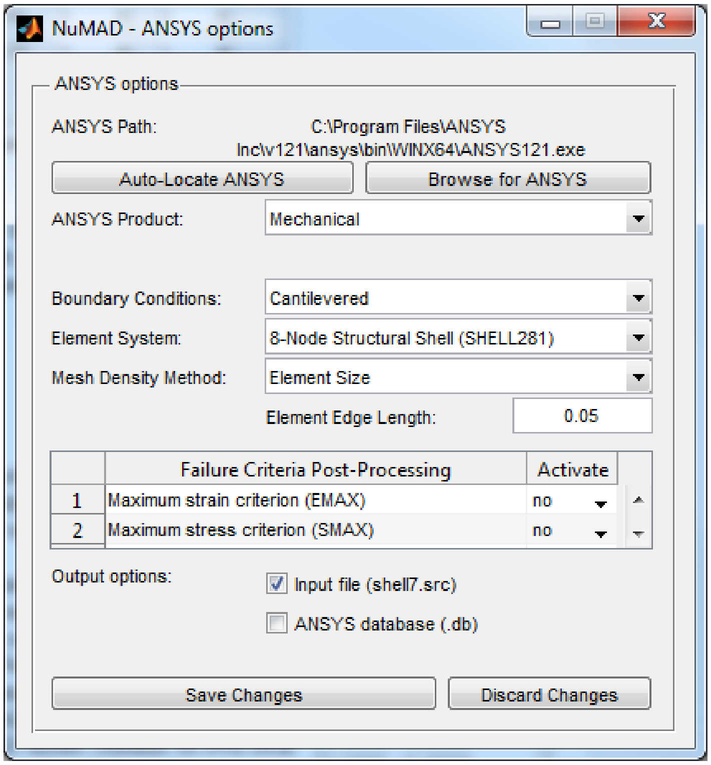
Upper DP: 2 2

Lower DP: 1 1

Figure 40 Shear webs dialog.

ANSYS Menu

The “ANSYS” menu allows the user to set model creation options including boundary conditions, element system, mesh size, and output type. The “Generate Now” menu item starts generating the shell7 file and/or the associated ANSYS blade model.



ANSYS Path: C:\Program Files\ANSYS Inc\v121\ansys\bin\WINX64\ANSYS121.exe

ANSYS Product: Mechanical

Boundary Conditions: Cantilevered

Element System: 8-Node Structural Shell (SHELL281)

Mesh Density Method: Element Size

Element Edge Length: 0.05

	Failure Criteria Post-Processing	Activate
1	Maximum strain criterion (EMAX)	no
2	Maximum stress criterion (SMAX)	no

Output options:

☑ Input file (shell7.src)

☐ ANSYS database (.db)

Figure 41 ANSYS options dialog.

Discussion of ANSYS Element Types

NuMAD is currently set up to utilize four different ANSYS layered shell elements:

Shell99, Offset layered shell – This element was undocumented[viii] in ANSYS R12 and was replaced by Shell281 with an improved formulation. The layers and material properties for this element type are assigned using ANSYS real constants. The element formulation for Shell99 was shown to be inaccurate in certain instances involving shear. See Laird et.al.[7] for more information on the inaccuracy. This element is recommended for users who only have access to older versions of ANSYS.

Shell91, Nonlinear offset layered shell – This element was undocumented in ANSYS R12 and was replaced by Shell281 with an improved formulation. The layers and material properties for this element type are assigned using ANSYS real constants. This element is recommended for users who only have access to older versions of ANSYS and want to perform nonlinear analyses.

Shell281, 8-node structural shell – This element first appeared in ANSYS R12 with an improved formulation. By the time ANSYS R13 was released, the improved formulation was fully verified and became the standard mode of use. The layers and material properties for this element type are assigned using ANSYS sections. As an 8-node element, it is able to capture the effects of curvature and is quite an accurate element. However, there are limitations on the ratio of thickness to curvature of the element such that many wind turbine blade nose elements can be in violation of the criteria. If these conditions exist, one approach for getting around the problem is to create additional DP's that bound the nose region and assign a material that is 10 times stiffer and denser, but 10 times thinner, as the actual material assignment.

Shell181, 4-node structural shell – This element first appeared in ANSYS R12 with an improved formulation. By the time ANSYS R13 was released, the improved formulation was fully verified and became the standard mode of use. The layers and material properties for this element type are assigned using ANSYS sections. Because this element is a linear, 4-node element, it does not capture curvature and it requires smaller elements. However, this element is not subject to the limitations on the ratio of curvature to thickness. Its drawback is that it may not be as accurate as the Shell281 element.

Please see the ANSYS documentation on these elements for even more detail information.

NuMAD currently is not set up to create blade models using solid elements.

Guidelines for Element Edge Length

The analyst is ultimately responsible for determining an appropriate element size for their work using a mesh convergence study that is specific to their blade and their output metrics.

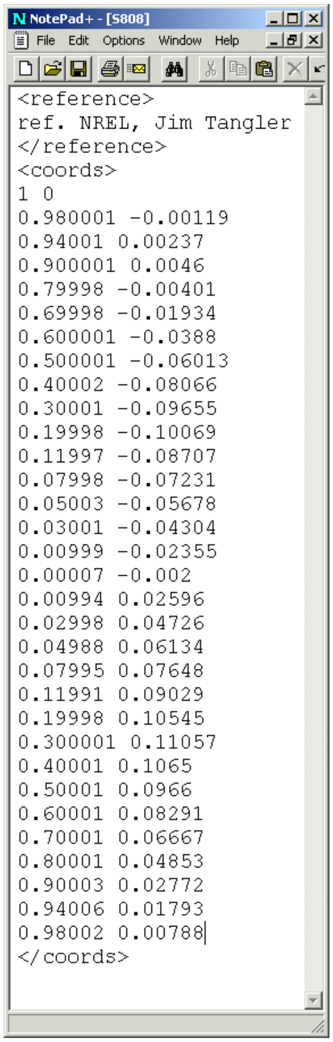
As a starting point for reasonably accurate results in most engineering analyses of the wind turbine blade, global element length can be set based on the max chord of the blade. When using Shell281 elements, an element length of approximately 0.050 times max chord is generally adequate. Using Shell181 elements, an element length of 0.025 times max chord of the blade is generally adequate. These criteria apply for computations involving panel buckling, modal

[viii] "Undocumented" means that the capability has been removed from the ANSYS documentation but it still exists in the software if the user knows how to call it. Once undocumented, ANSYS provides no further technical development or support for the capability.

analysis, static analysis, and static analysis in support of BPE. If stress concentrations are present in the blade, then of course a finer mesh is likely required for accurate analysis of the stresses in those regions. Again, it is ultimately up to the analyst to make the determination regarding the accuracy of their model.

Airfoils

NuMAD uses a database of airfoil profile descriptions so that this information need not be redefined with each new model. The airfoil profile descriptions are simple text files in an XML format.



```
<reference>
ref. NREL, Jim Tangler
</reference>
<coords>
1 0
0.980001 -0.00119
0.94001 0.00237
0.900001 0.0046
0.79998 -0.00401
0.69998 -0.01934
0.600001 -0.0388
0.500001 -0.06013
0.40002 -0.08066
0.30001 -0.09655
0.19998 -0.10069
0.11997 -0.08707
0.07998 -0.07231
0.05003 -0.05678
0.03001 -0.04304
0.00999 -0.02355
0.00007 -0.002
0.00994 0.02596
0.02998 0.04726
0.04988 0.06134
0.07995 0.07648
0.11991 0.09029
0.19998 0.10545
0.300001 0.11057
0.40001 0.1065
0.50001 0.0966
0.60001 0.08291
0.70001 0.06667
0.80001 0.04853
0.90003 0.02772
0.94006 0.01793
0.98002 0.00788
</coords>
```

Figure 42: Sample airfoil profile definition file.

Airfoil Files Locations

The definitions for the airfoils in the master airfoil profiles database are contained in files which reside in a directory named *airfoils* which is a subdirectory of the main NuMAD directory. This group of files is called the master airfoils database because the information is available for all NuMAD models. In addition to the master airfoils database, each NuMAD model contains its own local airfoils database. The location of the local airfoils database is within a directory named *airfoils* within the blade project directory.

Creating New Airfoil Profiles

The database of airfoil definitions is a collection of text files where each file contains a normalized set of x-y coordinate pairs and the file name (excluding the extension) is the name of the airfoil.

A typical airfoil profile definition is shown in Figure 42. At the beginning of the file is a reference block. The reference is delimited by the tags <reference> and </reference>. The user may place a reference or general note about the airfoil between the tags and this reference will be displayed in the window title bar when the airfoil is displayed. A reference is not required though the tags should be present, i.e., <reference></reference>.

The coordinate pairs are placed between the <coords> and </coords> tags. The coordinate pairs are listed one pair per line and one or more spaces separating the x-value from the y-value. All values are normalized such that the (x,y) coordinate of the nose is (0,0) and the (x,y) coordinate of the tail is (1,0). The first coordinate pair defines the tail and the subsequent coordinate pairs define the airfoil in a clockwise direction (Figure 43). The final coordinate pair, (0.98002, 0.00788) in this example, must not duplicate the first pair. It is assumed that the last coordinate pair connects to the first coordinate pair.

If a flat back airfoil, or an airfoil with a finite thickness trailing edge is being defined, then both the second coordinate pair and the final coordinate pair must be located at x=1, with y coordinates at nonzero values.

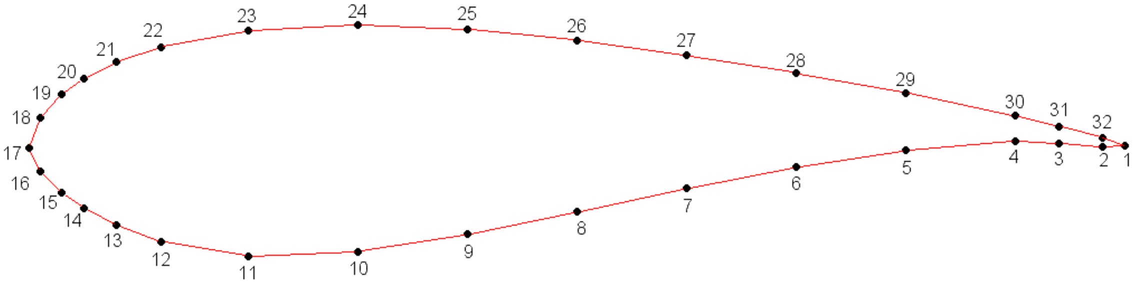


Figure 43: Sample airfoil profile with the clockwise coordinate numbering system.

The number of coordinate pairs in the file is not critical. One airfoil may be defined with 30 points while another is defined with 200. The more points used in the airfoil definition, the higher its fidelity in the final model.

NuMAD reads the Airfoils Database when it is first launched. For this reason, additions or modifications to the Airfoils Database will not take effect until NuMAD is next launched.

Trailing Edge Types

One of the parameters defined for each station is the trailing edge type. Options include sharp, round, and flat. This designation is important to the model creation process because it determines the manner in which the airfoil coordinate pairs are splined prior to creation of the blade surface areas. An illustrative example is shown in Figure 44. In the example, a flat back airfoil is splined using each of the three techniques. Each are described as follows:

Round

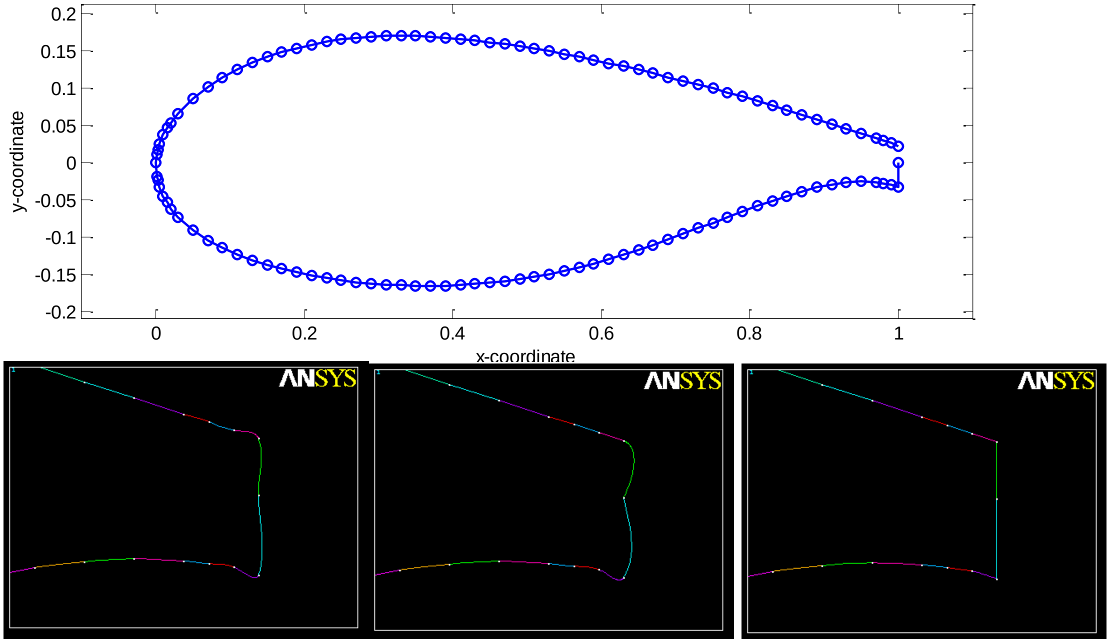
The first trailing edge type is “round” and refers to stations near the blade root which are usually circular or elliptical. Round stations will be generated with a smooth, continuous spline around the entire circumference of the airfoil, i.e. enforce a smooth tangent at the trailing edge and spline through all other airfoil coordinates.

Sharp

This trailing edge type allows a discontinuous tangent at the trailing edge and spline through all other airfoil coordinates, thus creating a smooth transition to a sharp trailing edge.

Flat

Flat back airfoil coordinate pairs 1, 2, and N are all located at x=1, with airfoil coordinate pair 1 located at (x,y) = (1,0). This trailing edge type allows a discontinuous tangent at the flat trailing edge corners, coordinate pairs 2 and N. See Figure 45. The remainder of airfoil coordinates are used to create a smooth spline, thus creating corners and a flat at the trailing edge. Note: this approach is also used to represent airfoils with finite thickness trailing edges.



(a) round (b) sharp (c) flat

Figure 44 Splining examples of three trailing edge types utilized on a flat trailing edge.

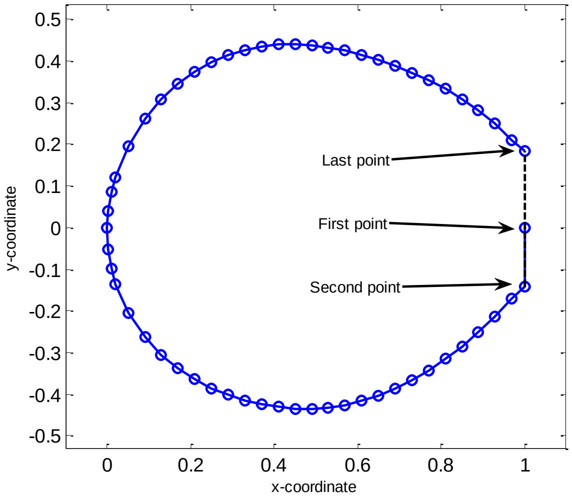


Figure 45 Flat trailing edge example.

Sharing Models

During the course of creating a blade model, you may need to share blade models with colleagues, either for collaboration or for troubleshooting. Following is a list of the files that are needed to most effectively transfer your blade project to another individual:

If using the standard GUI input approach:

- *Airfoils* folder and all contents
- *MatDBsi.txt* file
- **.nmd* file
- *zAirfoil.mac*, *zFlatback.mac*, *zSmoothe.mac* files
- optional: *bpe_station_ids.txt*

If using the Excel input approach (described on page 62 in the section on Read Excel-Based Input File):

- *Airfoils* folder and all contents
- *NuMAD.xls(x)* file
- *zAirfoil.mac*, *zFlatback.mac*, *zSmoothe.mac* files
- optional: *bpe_station_ids.txt*

NuMAD Settings Files

NuMAD stores a few files in the Windows %APPDATA% location. This location is easily viewed by typing %APPDATA% in the Windows Explorer search bar, as shown in Figure 46. There are two files stored in this location in the NuMAD folder: *settings.txt* and *ansys-productvars.txt*

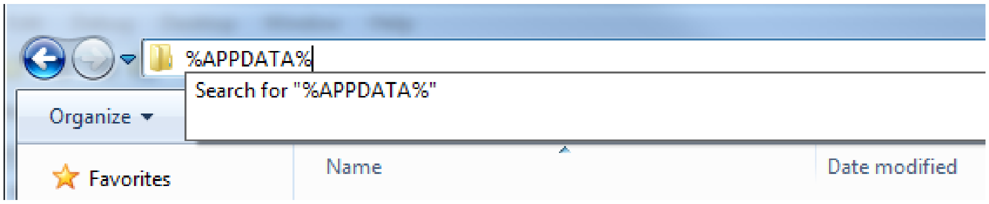


Figure 46 Viewing the APPDATA folder in Windows.

Settings File

This file contains various NuMAD system settings and default settings:

- ANSYS executable path
- ANSYS product code
- BModes executable path
- PreComp executable path
- GUI layout and size settings
- Path of last blade project

A simple way to reset all of these settings is to delete the file and then let NuMAD regenerate the file when it next opens.

ANSYS Product Variables

This file lists the ANSYS product variable names available in the NuMAD dropdown lists. You may manually edit this file to add/remove entries or change the description as you please. If you delete this file, the default content will regenerate when NuMAD restarts.

ADVANCED CAPABILITIES

Version 2 of NuMAD offers capabilities that go beyond the previous basics of entering and managing the blade geometry and materials information for output to an APDL script. These are capabilities that researchers at Sandia have created because they add quite a lot of efficiency to activities in wind turbine blade research. The capabilities in this section are relatively new and may contain unknown bugs and perhaps plenty of room for improvement. They are used on a regular basis and it is expected that they will develop and mature as feedback is gathered.

Output for CFD Geometry

Figure 47 shows a blade view of the Sandia CX-100 blade model in NuMAD. Figure 48 shows the menu options accessible from the "Plot3D >> Output Options…" menu item in NuMAD.

Plot3D is an open source program developed at NASA for viewing grid and results data associated with CFD. Plot3D file format is a standard file format used for storing grid and results data. It was created by NASA Ames Research Center in 1982 and remains an often used file format. Although it can only store a structured grid, it is the most compact way to store a structured grid and its results. It has the added benefit that by knowing the block number, i location, j location, and k location, any grid point value or result value may be extracted. Additionally, given the location of any value, the block number, i location, j location, and k location may be determined. This makes it simple to iterate through a series of volumetric elements and calculate all required properties for the purpose of a CFD analysis[8].

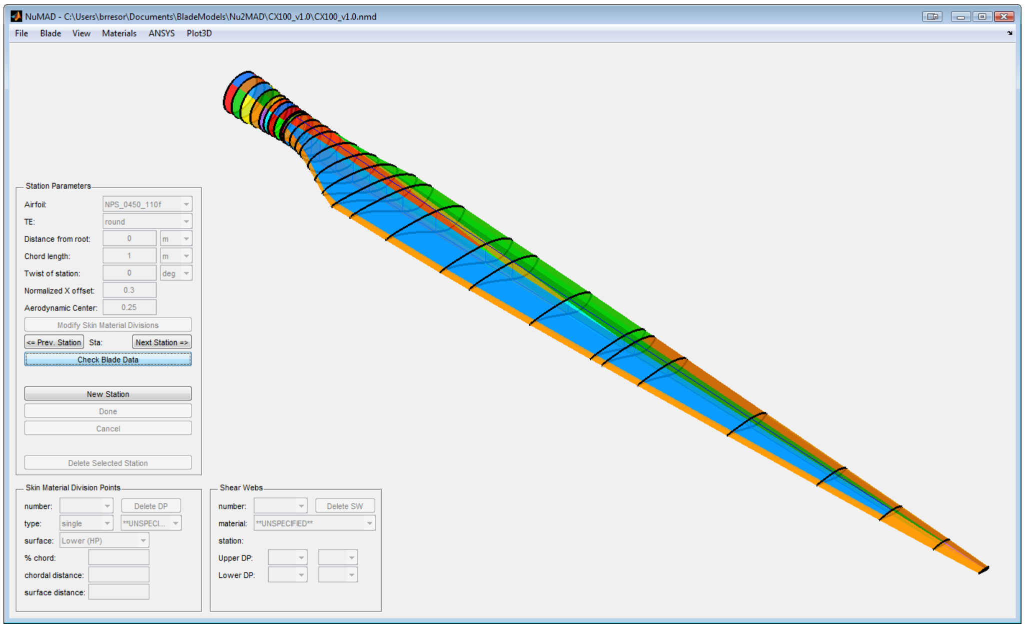
Following are the steps for usage of the Plot3D output capability in NuMAD:

- Step 1 - NuMAD is used to create the blade geometry (Figure 47). NuMAD outputs a Plot3D formatted data file (Figure 48 and Figure 49) which contains the geometry.
- Step 2 - The Plot3D formatted file is imported into a mesh generation tool (Figure 50) where the CFD analyst is able to create the appropriate mesh. Plot3D formatted data can be used as both a mesh and geometry source for mesh generation. Figure 50 shows the imported surface geometry in Pointwise®, a third-party CFD meshing program. Figure 51 shows examples of CFD meshes that were generated using Pointwise with the Plot3D formatted output from NuMAD.
- To enable better control of geometry seams away from high curvature regions of the blade, NuMAD allows the user to define surface splits at desired chord locations on the upper and lower surfaces. The recommended split location is near 10% chord. See Figure 50 and Figure 51 for examples of 10% chord splits.
- Step 3 - The mesh is used by a CFD solver in order to calculate dynamic fluid behavior.
- Step 4 - Results are plotted in a CFD visualization tool.

A variety of software can use Plot3D formatted data. An incomplete list of a few programs that support it would include

- Meshers:
    - Pointwise/Gridgen (a primary tool in use at Sandia)
- Solvers:
    - Nastran
    - Fluent
    - Star-CCM+/Star-CD
    - OpenFOAM
- Visualizers:
    - Tecplot
    - Paraview (open source)

A variety of mesh generation tools, CFD solvers and results visualization tools can be used. These tools are all separate from NuMAD and they are not distributed with NuMAD.



Airfoil: NPS_0450_110f
TE: round
Distance from root: 0 m
Chord length: 1 m
Twist of station: 0 deg
Normalized X offset: 0.3
Aerodynamic Center: 0.25

number:
type: single **UNSPECI...
surface: Lower (HP)
% chord:
chordal distance:
surface distance:

number:
material: **UNSPECIFIED**
station:
Upper DP:
Lower DP:

Figure 47 The Sandia CX-100 in NuMAD.

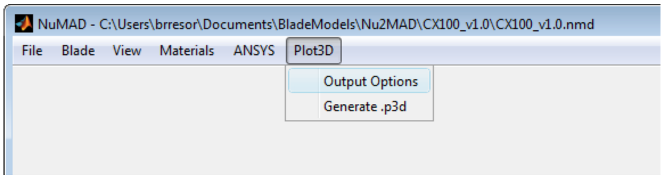

NuMAD - C:\Users\brresor\Documents\BladeModels\Nu2MAD\CX100_v1.0\CX100_v1.0.nmd
File
Blade
View
Materials
ANSYS
Plot3D
Output Options
Generate .p3d

Figure 48 NuMAD menu selections for Plot3D output for CFD meshing.

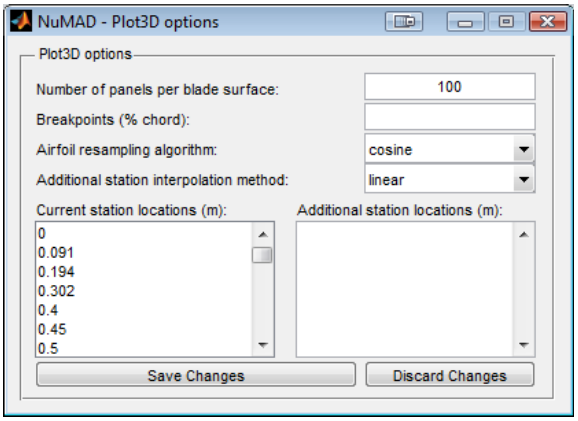
Figure 49 shows the output options for the Plot3D formatted file. Following is a description of each field:

- Number of panels – From CFD terminology. The number of discrete panels around the airfoil shape at each blade station.
- Breakpoints – These are points where the blade surface will be divided down the span of the geometry. They are helpful for defining areas, especially the leading edge of the

blade, where mesh refinement in later meshing processes will be needed. It is common to define a breakpoint at 10% chord.

- Airfoil resampling – The original NuMAD airfoil coordinates are splined and resampled based on the number of panels and the point spacing. This parameter defines the point spacing algorithm such that the points are evenly spaced around the airfoil (constant), bunched at the nose (half-cosine) or bunched at both the nose and the tail (cosine).
- Station interpolation method and Additional station locations – The user may specify locations in addition to given stations to generate airfoil coordinates. The new airfoil coordinates are created using an airfoil interpolation algorithm[ix] that first computes an interpolated camber line and then computes an interpolated thickness distribution. The airfoils are computed at every "Current station location" and each "Additional station location."

Important note: The Plot3D output capability is not currently set up to work with "flat" trailing edge types.



NuMAD - Plot3D options

Plot3D options

Number of panels per blade surface: 100

Breakpoints (% chord):

Airfoil resampling algorithm: cosine

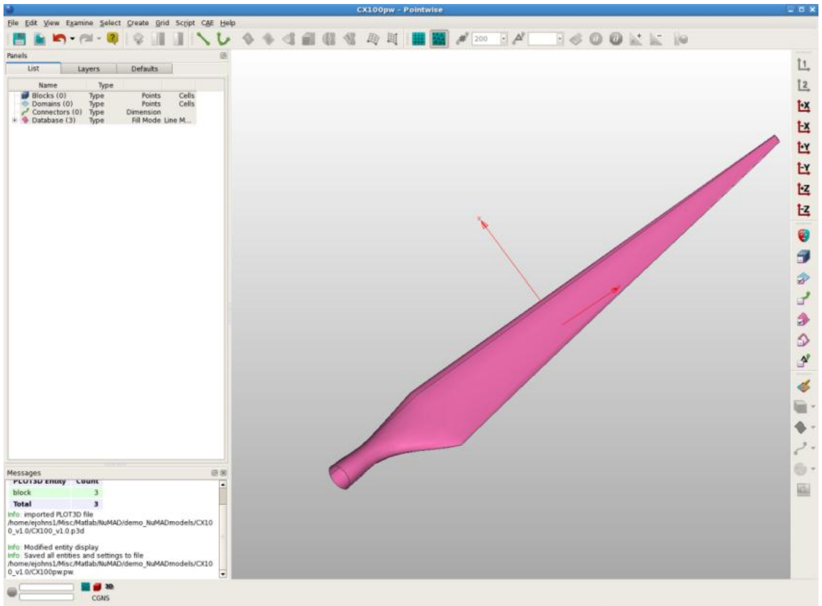
Additional station interpolation method: linear

Current station locations (m):
0
0.091
0.194
0.302
0.4
0.45
0.5

Additional station locations (m):

Figure 49 Plot3D "Output Options" dialog box in NuMAD.

[ix] The interpolation algorithm can be found in the *SNL_NuMAD_Toolbox*. The function is called *interpAirfoil()*.



Name	Type		
Blocks (0)	Type	Points	Cells
Domains (0)	Type	Points	Cells
Connectors (0)	Type	Dimension	
Database (3)	Type	Fill Mode	Line M...

PLOT3D Entity	Count
block	3
Total	3

```
Info  imported PLOT3D file
/home/ejohns1/Misc/Matlab/NuMAD/demo_NuMADmodels/CX10
0_v1.0/CX100_v1.0.p3d

Info  Modified entity display
Info  Saved all entities and settings to file
/home/ejohns1/Misc/Matlab/NuMAD/demo_NuMADmodels/CX10
0_v1.0/CX100pw.pw
```

Figure 50 Import CX100 surface geometry into Pointwise® using Plot3D formatted data.

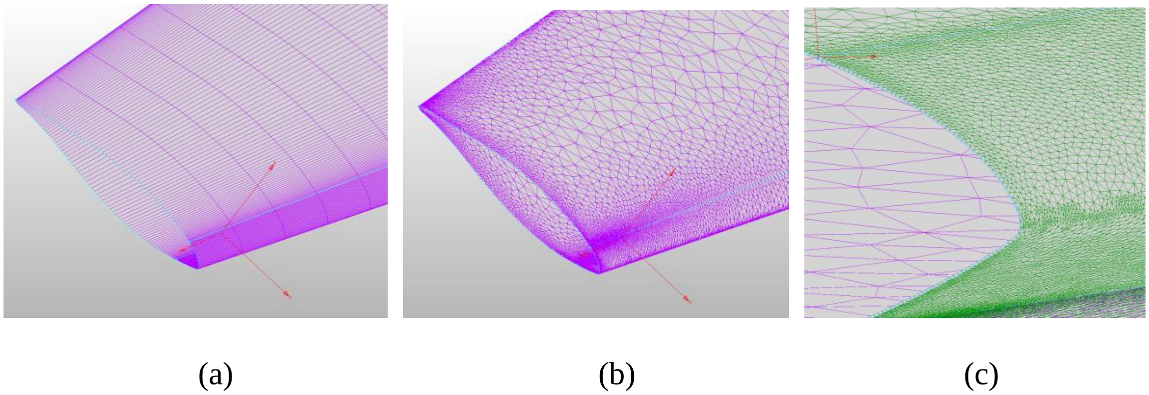

(a) (b) (c)

Figure 51 (a) Structured and (b) unstructured surface mesh; (c) Initial boundary layer extrusion in Pointwise.

Read Excel-Based Input File

While the graphical environment of NuMAD is great for setting up and managing a large amount of information effectively, many heavy users of NuMAD have a need for a tabular input of all the information required to create the blade model. Rather than implement a tabular input interface in NuMAD, we created an input format that takes advantage of Microsoft Excel's capabilities. Once the Excel workbook is created according to the required format, NuMAD can read the Excel workbook and create the blade model. This approach is very powerful because it allows parameterization of any number of blade inputs within the Excel file. Iterative design studies for blades can be accomplished quickly and easily. Additionally, the Excel workbook offers a potential means for sharing of blade model information between researchers.

The definition of the layout and information in the Excel workbook is evolving. The current format is effective for relatively simple blades. As the capability is exercised, we expect a more mature format to evolve. See Future Work on page 63 for a list of currently recognized needs in the next iteration.

Reading an Excel file from NuMAD

The example blade model *SNLWindPactv1.3* contains a blade model in Excel format. A NuMAD blade model is created from this file by choosing "File >> XLS-2-NuMAD…" and then choosing the appropriate Excel file. The file is converted into a **.nmd* file and a *MatDBsi.txt* file (overwriting the existing *MatDBsi.txt* file). Then, the model opens in NuMAD.

Getting started with a new blade model

It is highly recommended to start a new Excel blade model from an existing, working, Excel blade model file.

- Step 1, in NuMAD, create a new blade project and save it. As a result, a project folder and an airfoils folder are created.
- Step 2, populate the newly created airfoils folder with additional airfoil files required by your blade
- Step 3, copy a clean NuMAD.xls file example into your project folder. A great example file is located in *NuMAD_Examples/SNLWindPACTv1.3*
- Step 4, make modifications to the example file to reflect your blade model.

A detailed description of the contents of the *NuMAD.xls* file is found in Appendix A of this User's Manual.

The complexity of information contained in the Excel workbook can make it challenging to create a new model without experiencing errors in the Excel-to-NuMAD file conversion process. Error checking during the process is still crude. For this reason, it is recommended to always start with a working Excel workbook and to make changes gradually. In between changes, read the model into NuMAD to confirm that it is set up as expected. Use the graphical capabilities of NuMAD to inspect the model for correctness. Once your model is set up, incremental design iterations are achieved quite easily.

Future Work

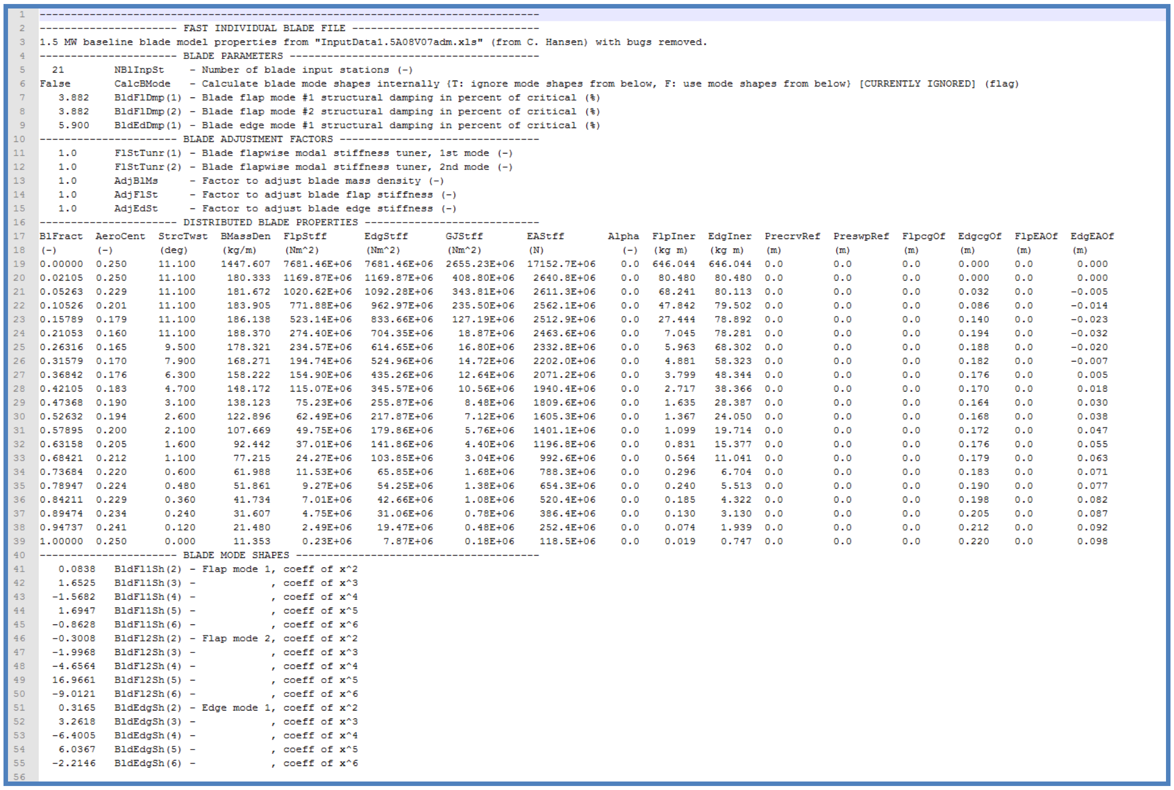
Following is a list of capabilities that are not currently included in the Excel representation of blade models. These are capabilities that will be developed in future versions.

- Material orientation angle is not represented. All angles are assumed to be zero-degrees.
- DP types "hourglass" and "flare" are not supported.
- Fields for material failure criteria parameters are not included
- Location of blade reference axis other than at the blade pitch axis are not included

Computation of Blade Cross Section Properties

Wind blade cross section properties are used in aeroelastic analyses associated with wind turbine system design. Examples of cross section properties include flapwise and edgewise stiffness, torsional stiffness, axial stiffness, mass density, inertias, location of elastic axis and location of center of gravity. Specifically, the cross section properties are used by the NREL FAST code to represent the wind turbine blade. NuMAD aids in the conversion of the detailed blade model into inputs that can be utilized by the FAST code as a blade input file, example shown in Figure 52. NuMAD connects to the FAST blade file through two different tools: PreComp and BPE.

Note: Both processes currently assume a straight blade, i.e. the blade reference axis is a straight line. Future releases of NuMAD will include implementation of BPE and PreComp for swept blades.



```
--------------------------------------------------------------------------------
---------------------- FAST INDIVIDUAL BLADE FILE ------------------------------
1.5 MW baseline blade model properties from "InputData1.5A08V07adm.xls" (from C. Hansen) with bugs removed.
---------------------- BLADE PARAMETERS ----------------------------------------
  21        NBlInpSt    - Number of blade input stations (-)
False       CalcBMode   - Calculate blade mode shapes internally (T: ignore mode shapes from below, F: use mode shapes from below) [CURRENTLY IGNORED] (flag)
   3.882    BldFlDmp(1) - Blade flap mode #1 structural damping in percent of critical (%)
   3.882    BldFlDmp(2) - Blade flap mode #2 structural damping in percent of critical (%)
   5.900    BldEdDmp(1) - Blade edge mode #1 structural damping in percent of critical (%)
---------------------- BLADE ADJUSTMENT FACTORS --------------------------------
   1.0      FlStTunr(1) - Blade flapwise modal stiffness tuner, 1st mode (-)
   1.0      FlStTunr(2) - Blade flapwise modal stiffness tuner, 2nd mode (-)
   1.0      AdjBlMs     - Factor to adjust blade mass density (-)
   1.0      AdjFlSt     - Factor to adjust blade flap stiffness (-)
   1.0      AdjEdSt     - Factor to adjust blade edge stiffness (-)
---------------------- DISTRIBUTED BLADE PROPERTIES ----------------------------
BlFract  AeroCent  StrcTwst  BMassDen  FlpStff      EdgStff      GJStff       EAStff        Alpha  FlpIner  EdgIner  PrecrvRef  PreswpRef  FlpcgOf  EdgcgOf  FlpEAOf  EdgEAOf
(-)      (-)       (deg)     (kg/m)    (Nm^2)       (Nm^2)       (Nm^2)       (N)             (-)  (kg m)   (kg m)   (m)        (m)        (m)      (m)      (m)      (m)
0.00000  0.250     11.100    1447.607  7681.46E+06  7681.46E+06  2655.23E+06  17152.7E+06    0.0  646.044  646.044  0.0        0.0        0.0      0.000    0.0        0.000
0.02105  0.250     11.100     180.333  1169.87E+06  1169.87E+06   408.80E+06   2640.8E+06    0.0   80.480   80.480  0.0        0.0        0.0      0.000    0.0        0.000
0.05263  0.229     11.100     181.672  1020.62E+06  1092.28E+06   343.81E+06   2611.3E+06    0.0   68.241   80.113  0.0        0.0        0.0      0.032    0.0       -0.005
0.10526  0.201     11.100     183.905   771.88E+06   962.97E+06   235.50E+06   2562.1E+06    0.0   47.842   79.502  0.0        0.0        0.0      0.086    0.0       -0.014
0.15789  0.179     11.100     186.138   523.14E+06   833.66E+06   127.19E+06   2512.9E+06    0.0   27.444   78.892  0.0        0.0        0.0      0.140    0.0       -0.023
0.21053  0.160     11.100     188.370   274.40E+06   704.35E+06    18.87E+06   2463.6E+06    0.0    7.045   78.281  0.0        0.0        0.0      0.194    0.0       -0.032
0.26316  0.165      9.500     178.321   234.57E+06   614.65E+06    16.80E+06   2332.8E+06    0.0    5.963   68.302  0.0        0.0        0.0      0.188    0.0       -0.020
0.31579  0.170      7.900     168.271   194.74E+06   524.96E+06    14.72E+06   2202.0E+06    0.0    4.881   58.323  0.0        0.0        0.0      0.182    0.0       -0.007
0.36842  0.176      6.300     158.222   154.90E+06   435.26E+06    12.64E+06   2071.2E+06    0.0    3.799   48.344  0.0        0.0        0.0      0.176    0.0        0.005
0.42105  0.183      4.700     148.172   115.07E+06   345.57E+06    10.56E+06   1940.4E+06    0.0    2.717   38.366  0.0        0.0        0.0      0.170    0.0        0.018
0.47368  0.190      3.100     138.123    75.23E+06   255.87E+06     8.48E+06   1809.6E+06    0.0    1.635   28.387  0.0        0.0        0.0      0.164    0.0        0.030
0.52632  0.194      2.600     122.896    62.49E+06   217.87E+06     7.12E+06   1605.3E+06    0.0    1.367   24.050  0.0        0.0        0.0      0.168    0.0        0.038
0.57895  0.200      2.100     107.669    49.75E+06   179.86E+06     5.76E+06   1401.1E+06    0.0    1.099   19.714  0.0        0.0        0.0      0.172    0.0        0.047
0.63158  0.205      1.600      92.442    37.01E+06   141.86E+06     4.40E+06   1196.8E+06    0.0    0.831   15.377  0.0        0.0        0.0      0.176    0.0        0.055
0.68421  0.212      1.100      77.215    24.27E+06   103.85E+06     3.04E+06    992.6E+06    0.0    0.564   11.041  0.0        0.0        0.0      0.179    0.0        0.063
0.73684  0.220      0.600      61.988    11.53E+06    65.85E+06     1.68E+06    788.3E+06    0.0    0.296    6.704  0.0        0.0        0.0      0.183    0.0        0.071
0.78947  0.224      0.480      51.861     9.27E+06    54.25E+06     1.38E+06    654.3E+06    0.0    0.240    5.513  0.0        0.0        0.0      0.190    0.0        0.077
0.84211  0.229      0.360      41.734     7.01E+06    42.66E+06     1.08E+06    520.4E+06    0.0    0.185    4.322  0.0        0.0        0.0      0.198    0.0        0.082
0.89474  0.234      0.240      31.607     4.75E+06    31.06E+06     0.78E+06    386.4E+06    0.0    0.130    3.130  0.0        0.0        0.0      0.205    0.0        0.087
0.94737  0.241      0.120      21.480     2.49E+06    19.47E+06     0.48E+06    252.4E+06    0.0    0.074    1.939  0.0        0.0        0.0      0.212    0.0        0.092
1.00000  0.250      0.000      11.353     0.23E+06     7.87E+06     0.18E+06    118.5E+06    0.0    0.019    0.747  0.0        0.0        0.0      0.220    0.0        0.098
---------------------- BLADE MODE SHAPES ---------------------------------------
   0.0838   BldFl1Sh(2) - Flap mode 1, coeff of x^2
   1.6525   BldFl1Sh(3) -            , coeff of x^3
  -1.5682   BldFl1Sh(4) -            , coeff of x^4
   1.6947   BldFl1Sh(5) -            , coeff of x^5
  -0.8628   BldFl1Sh(6) -            , coeff of x^6
  -0.3008   BldFl2Sh(2) - Flap mode 2, coeff of x^2
  -1.9968   BldFl2Sh(3) -            , coeff of x^3
  -4.6564   BldFl2Sh(4) -            , coeff of x^4
  16.9661   BldFl2Sh(5) -            , coeff of x^5
  -9.0121   BldFl2Sh(6) -            , coeff of x^6
   0.3165   BldEdgSh(2) - Edge mode 1, coeff of x^2
   3.2618   BldEdgSh(3) -            , coeff of x^3
  -6.4005   BldEdgSh(4) -            , coeff of x^4
   6.0367   BldEdgSh(5) -            , coeff of x^5
  -2.2146   BldEdgSh(6) -            , coeff of x^6
```

Figure 52 Example FAST Blade file.

Disclaimer

Accurate computation of wind turbine blade cross section properties is a critical step in the process of performing aeroelastic simulations to support wind turbine system design. Users of these capabilities are required to build confidence in these approaches by conducting their own series of verification exercises before proceeding to generate inputs for critical work.

PreComp Background

PreComp is a tool available as part of the NREL NWTC Design Codes suite for computation of equivalent section properties of wind turbine blades. PreComp is computationally efficient, and is publicly available. Even without access to ANSYS software, NuMAD users can essentially use the NuMAD GUI as a front end for PreComp for creation of FAST blade input files. NuMAD translates the information in the blade model database into PreComp input files, runs the PreComp analyses, and then converts the PreComp outputs into a FAST blade input file. More information can be found on PreComp theory and usage at the NWTC Design Codes website: http://wind.nrel.gov/designcodes/preprocessors/precomp/

Beam Property Extraction (BPE) Background

References by Malcolm describe the BPE process in detail.[9,10] BPE is a technique to determine equivalent beam properties for a number of discrete segments, with edges defined by NuMAD stations, along a three dimensional model of a structure, as seen in Figure 53.

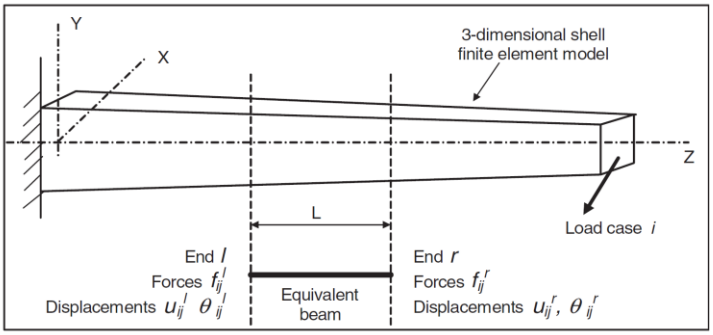

Y
X
3-dimensional shell finite element model
Z
L
Load case i
End l
Forces f_{ij}^{l}
Displacements u_{ij}^{l} θ_{ij}^{l}
Equivalent beam
End r
Forces f_{ij}^{r}
Displacements u_{ij}^{r}, θ_{ij}^{r}

Figure 53 Three-dimensional blade model and equivalent beam element[11]

The method is based on the following process:

- Use ANSYS to apply a unit load in each of the six degrees of freedom at the tip of the FE blade model and compute deflections
- Process the resulting nodal displacements at predefined NuMAD stations to compute the six bending and twist deformations of each segment
- Compute the 6×6 Timoshenko stiffness matrices and mass matrices for each segment
- Compute effective section properties for each segment based on the stiffness and mass matrices. See Table 8 for a list of FAST blade input parameters and their sources.

The method inherently includes three-dimensional effects such as shear and warping. It also captures effects arising from nearby boundary conditions and non-uniform blade geometry. Calculation of the section properties are demonstrated in a series of validation examples in Malcolm et.al.[12,13] and in Resor et.al.[14,15]

The BPE approach has limitations. Proper application of the force to the tip of the finite element blade model is an imperfect process[x]. Also, the sectional properties can be sensitive to size of the blade segments chosen. Beam segments that are too short can cause problems with singular matrices in the computation of the inverse problem. Beam segments that are too long will not adequately capture large gradients in spanwise properties.

[x] In the current process, the unity forces are applied to the blade tip at a point which is constrained to the entire tip station.

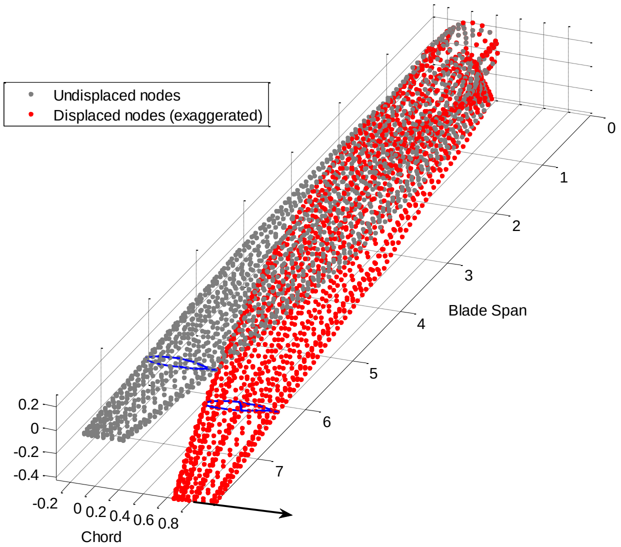


Figure 54 Example of nodes used by the BPE technique. A single section of nodes is highlighted in blue.

It is important to note the nature of the BPE output as it relates to spanwise location of computed properties. The BPE algorithm calculates properties for each segment in the blade. Segments are defined by choosing NuMAD stations from the full set of stations. When properties are computed for a single BPE segment, those properties are representative of an average value for the entire segment. The representation of the parameters used in this process, as a function of blade span, might be visualized as the stepped function as seen in Figure 55. The properties must be assigned to specific spanwise locations in order to report the property results in the form of a FAST blade file. The approach is to assign the computed average properties for each segment to the mid-point of each segment. Additionally, because values are assigned at the mid-point of the segments, properties at the blade root and tip are not known. The computed properties are used to extrapolate values at the root and tip.

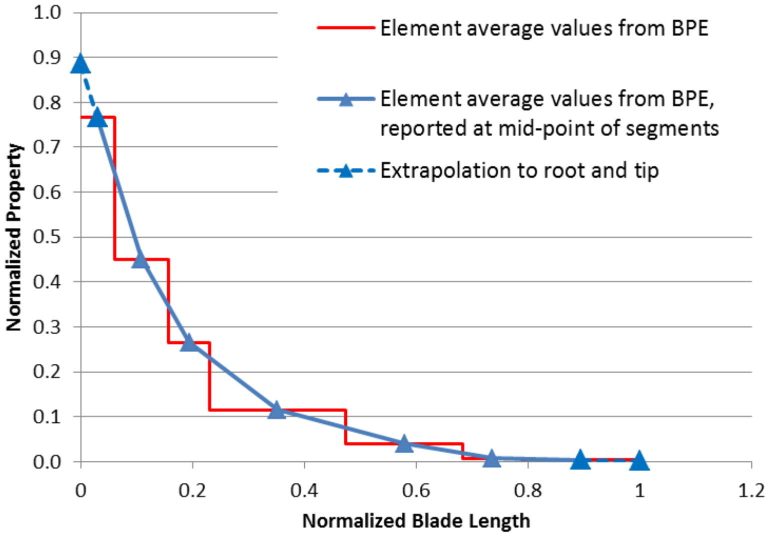


Figure 55 Blade property distributions.

Table 8 FAST blade file parameters output from NuMAD using either PreComp or BPE.

FAST Blade File Parameter	Comment
Bld**Dmp(1-3)	Assumed to be 1.5%; User should set to correct value for their blade
BLADE ADJUSTMENT FACTORS	All values set to 1.0
BlFract	Computed by NuMAD
AeroCent	Computed using X-Offset and Aerodynamic Center parameters from NuMAD model
StrcTwst	Assumed to be blade twist from NuMAD model
BMassDens	Computed by BPE or PreComp
FlpStff	Computed by BPE or PreComp
EdgStff	Computed by BPE or PreComp
GJStff	Computed by BPE or PreComp
EAStff	Computed by BPE or PreComp
Alpha	Not used
FlpIner	Computed by BPE or PreComp
EdgIner	Computed by BPE or PreComp
PrecrvRef	Not used
PreswpRef	Not used
FlpcgOf	Computed by BPE or PreComp
EdgcgOf	Computed by BPE or PreComp
FlpEAOf	Computed by BPE or PreComp
EdgEAOf	Computed by BPE or PreComp
Blade mode shape coefficients	Computed during the NuMAD process using BModes mode shapes

PreComp Usage Example

The PreComp analysis is ready to initiate once the NuMAD model has been fully defined and the PreComp path has been set under the “File >> Configure Paths…” menu. The user initiates the process by selecting “Advanced >> PreComp >> Run PreComp-to-FAST” from the “Advanced” menu dropdown.

1. Information in the NuMAD blade model is used to create PreComp inputs for each station in the blade model.
2. Cross section properties are computed using PreComp
3. The cross section properties are used as input to the BModes tool, which estimates mode shapes of the entire beam. The user is prompted in the Matlab console to identify the 1st and 2nd flapwise and 1st edgewise modes.
4. Polynomials are fit to the mode shapes. The polynomial coefficients are stored.
5. Cross section properties and mode shape polynomials are used to populate the FAST blade file.
6. The FAST blade file is saved as *FASTBlade_precomp.dat*

BPE Usage Example

Once the NuMAD blade model is complete and the ANSYS model can be generated, the user may set up and perform the BPE analysis. First, BPE segments are defined through the "Advanced" menu, as shown in Figure 56 and Figure 57. Figure 57 shows the blade planform as well as the lcoation of all defined stations in the NuMAD blade model. The user specifies the desired stations for BPE segment boundaries by either left-clicking on the stations in the graphic or by entering the station number in the test list. Stations in the graphic are highlighted green when they are designated as segment edges for subsequent analyses. Once the stations have been defined, the user clicks on "Save and Close" to save a text file to the blade project directory called *bpe_station_ids.txt*. This text file can be modified externally, if desired, and is an important input to the BPE computation process.

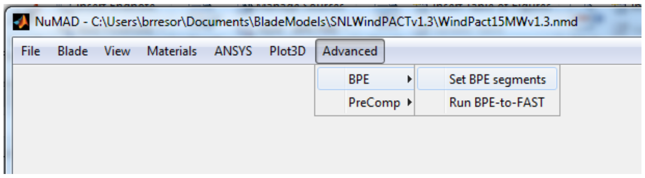
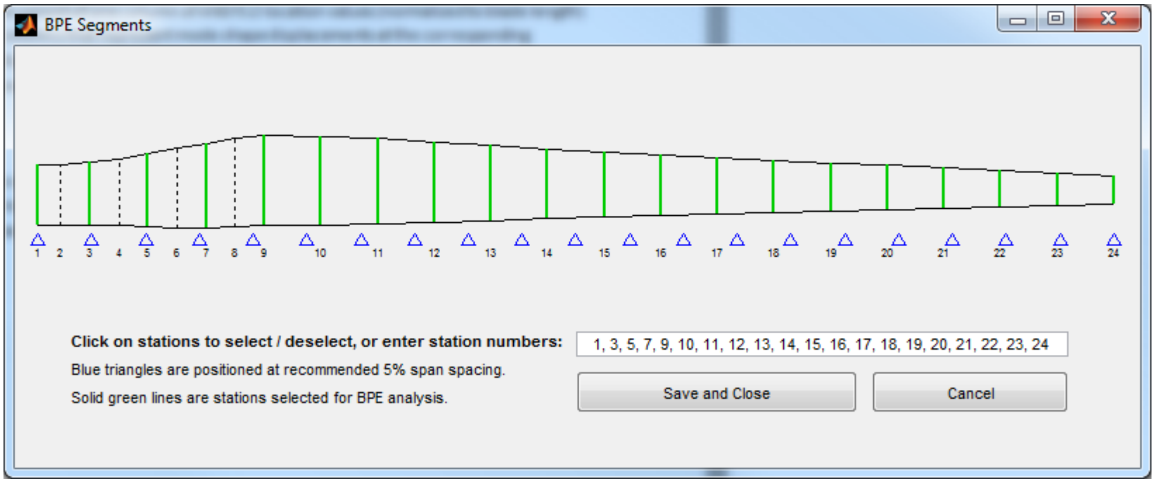

NuMAD - C:\Users\brresor\Documents\BladeModels\SNLWindPACTv1.3\WindPact15MWv1.3.nmd
File
Blade
View
Materials
ANSYS
Plot3D
Advanced
BPE
PreComp
Set BPE segments
Run BPE-to-FAST

Figure 56 Menu entry into BPE segment setup.

Segment Length Guidelines

Ideally, the user chooses a high number of NuMAD stations to define the edges of a large number of BPE segments. It is recommended to choose nodes closer where properties change fast and farther apart where they remain relatively constant. A good guideline is to set segments such that their lengths are each approximately 5% of blade length. Closer spacing, especially in the root region, is preferred if the shorter segment length does not cause singularity issues for BPE.



Click on stations to select / deselect, or enter station numbers: 1, 3, 5, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24

Blue triangles are positioned at recommended 5% span spacing.

Solid green lines are stations selected for BPE analysis.

Figure 57 Dialog box for specification of BPE segments.

The BPE analysis is ready to initiate when (1) the shell7.src file has been generated and (2) the BPE segment edges have been defined and saved to the file *bpe_station_ids.txt*. The user initiates the process by selecting "Advanced>>BPE>>Run BPE-to-FAST" from the menu. The following steps occur:

1. BPE-specific commands are appended to the *shell7.src* file in the project folder and saved as *shell7bpe.src*
2. The ANSYS model is generated and analyses are performed using the *shell7bpe.src* file
    - Six different loads are applied to the tip of the ANSYS blade model and nodal displacement results from the linear static analyses are stored
    - An inertial analysis is used to compute effective mass of all nodes and results are stored
    - All nodal results are compiled and saved in the text file *displacement.txt*
        - Note: Steps 1 and 2 are not performed if the *displacement.txt* file already exists in the blade file project folder
3. The BPE algorithm reads nodal results from the displacement.txt file. Plots of the nodes at each segment edge are shown for review by the user. BPE computes stiffness matrices and mass properties for each segment. Results are stored in a variety of text files, primarily *kout.txt*
4. Equivalent cross section properties are computed using the stiffness matrices and mass properties for each segment. Results are saved in the Matlab binary file called *BPE_SectionData.mat*
5. The cross section properties are used as input to the BModes tool, which estimates mode shapes of the entire beam. The user is prompted in the Matlab console to identify the 1st and 2nd flapwise and 1st edgewise modes.
6. Polynomials are fit to the mode shapes. The polynomial coefficients are stored.
7. Cross section properties and mode shape polynomials are used to populate the FAST blade file.
8. The FAST blade file is saved as *FASTBlade_bpe.dat*

BPE Intermediate Files

Several files are produced by BPE or by NuMAD during the process of creating the FAST blade input file.

bpe_station_ids.txt

List of NuMAD station ID's to include as BPE segment edges in subsequent BPE analysis.

bpe.txt

Input file utilized by BPE. Parameters are self-explanatory.

displacement.txt

This file is created at the end of the ANSYS analyses. It contains 23 columns of data: 1) node number, 2-4) coordinates of each node, 5-22) UX, UY, UZ displacements at each node for each of the 6 static load cases, and 23) mass of each node. This file is used by BPE to compute displacements of each segment boundary.

singularout.txt
Contents of this file display warnings from the BPE analysis regarding singular segments, if any. If a segment exhibits singular behavior, BPE attempts to resolve the issue by expanding the length of the segment and automatically choosing a nearest cross section of FE nodes for analysis. Sometimes, BPE is able to adequately deal with the problem segment. Most often, the user should use the warning of singular behavior as a sign that the choice of stations defining the segment edges should be reconsidered.

Discussion of BPE versus PreComp

References 14 and 15 compare and contrast the blade properties that are computed using both BPE and PreComp. Both approaches have been verified against analytical stiffness values and have been found to be accurate to within much less than 1% for simple uniform cross sections.

Discrepancies in computed properties from the two approaches become more evident when analyzing a real wind turbine blade geometry. Figure 58 shows a comparison of properties computed using BPE and PreComp for a blade model of a utility scale wind turbine. Discrepancies arise especially in the following regions:

- All stiffnesses: Inboard of maximum chord, especially at the root
- Torsional stiffness: throughout the blade
- Elastic axis offset: throughout the entire blade

A thorough test to adequately validate these predictions has not been done. There is no sure way to know which most representative of reality. This subject is an active research topic. It is likely that BPE has the potential to represent reality very well because it captures three dimensional effects that PreComp does not. However, discretization of the blade into segment for the BPE analysis likely takes away from its accuracy. PreComp uses a more traditional, time tested approach for computation of properties. The best approach may be a hybrid of the two. Ultimately, the blade designer makes the decision regarding the approach he decides to trust.

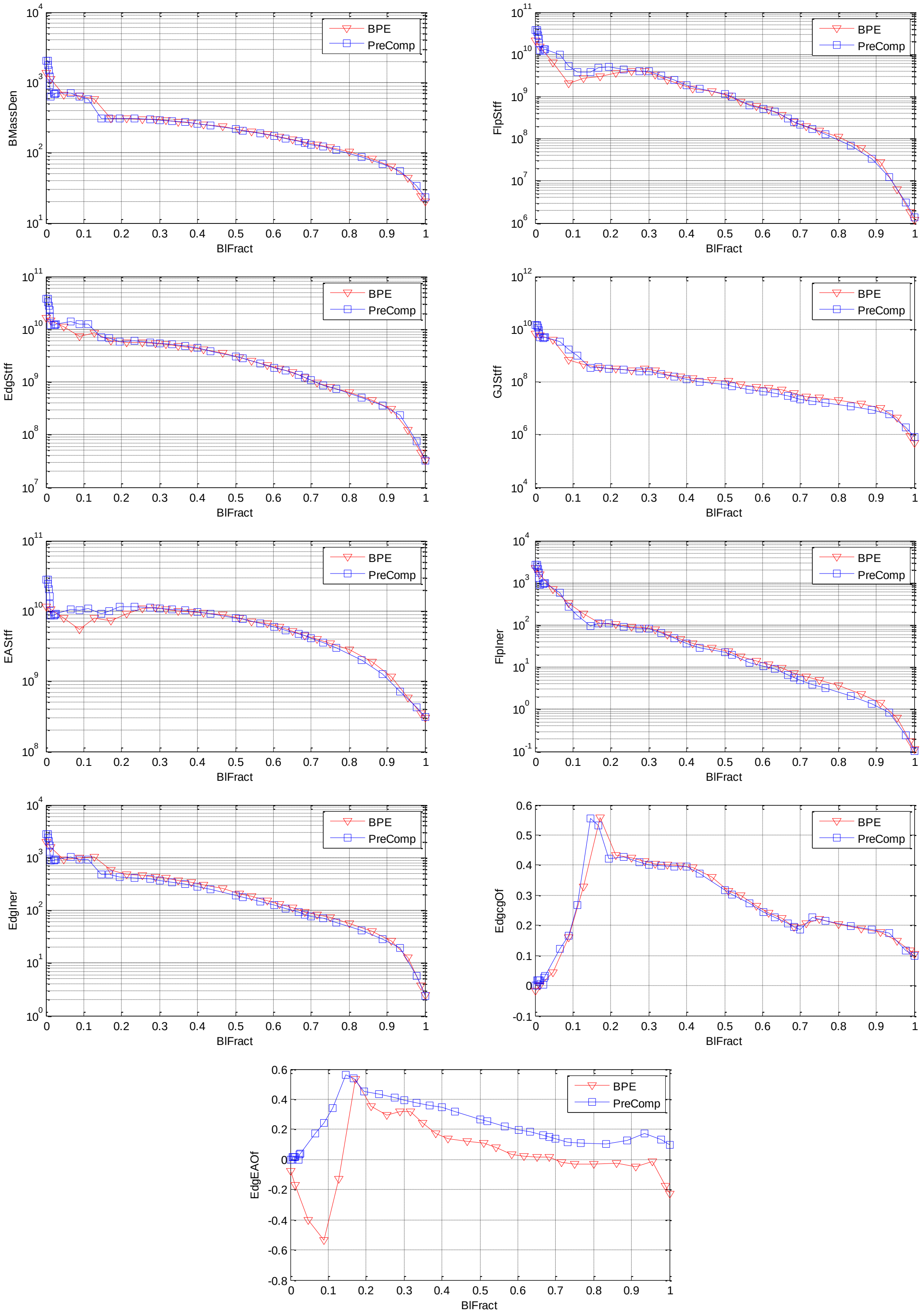


Figure 58 Comparison of computed section properties for a utility scale wind turbine blade model.

Classical Flutter Analysis

Through the "Advanced" menu item, the NuMAD GUI connects to an analysis module, called the BLade Aeroelastic Stability Tool (BLAST), that will perform a classical flutter analysis of an isolated rotating HAWT blade. This is a capability that was initially created to emulate the approach utilized by Lobitz[16]. The code has been verified against computations that were done with Lobitz's approach. BLAST was proven to meet, or exceed the fidelity and performance of the Lobitz tool. To date, no field test data are available for validation of a HAWT classical flutter prediction. Therefore, this tool should be used as only a guide in assessing a given wind turbine blade design's flutter margin with respect to operating speed. The tool is also valuable for flutter parameter studies on research blade concepts. If using this tool a blade concept is found to have a flutter speed near its operating speed, the blade designer should look seriously into spending a significant effort on validating a classical flutter computation for HAWTs in order to determine the confidence in these predictions.

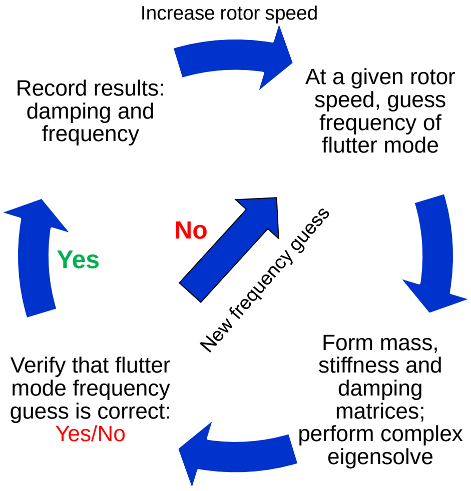
The core of BLAST is the formulation and implementation of a three-dimensional Euler-Bernoulli beam finite element. The formulation is three-dimensional in that it allows for three translational and three rotational motions at any point along the flexural axis of the beam. The underlying formulation accounts for rotational effects including Coriolis and spin softening phenomenon. Aeroelastic stability is examined by utilizing Theodorsen unsteady aerodynamic theory to modify system matrices through aerodynamic mass, damping, and stiffness terms. Nonlinear static analysis of centrifugal loadings on the rotating HAWT blade is used to modify the structural stiffness matrix of the blade for this equilibrium state. Modal analysis of the turbine including structural, rotational, and aerodynamic effects about this equilibrium state yields frequencies and damping ratios that give insight into the dynamic aeroelastic stability (flutter) of a HAWT blade design.

The finite element implementation allows for a robust modeling approach of complicated HAWT blade geometries. Non-uniform cross-sections with variable flexural axes locations can easily be accommodated. Aerodynamic and structural effects of swept and twisted bladed configurations are also accounted for in the BLAST formulation and implementation. BLAST incorporates structural couplings that exist due to offset mass axes and can account for couplings arising from cross-sectional geometry and composite materials. Rotational effects in the form of Coriolis and spin softening introduce further couplings in the deformation of the blade.

The analysis of classical flutter in wind turbines necessitates the use of unsteady aerodynamics. As pointed out by Leishman[17] for horizontal axis wind turbines there are two interconnected sources of unsteady aerodynamics. The first is a result of the trailing wake of the rotor and is addressed by investigating the interactions between the rotor motion and the inflow. The second (used here) is due to the shed wake of the individual blades and can be addressed using techniques developed for analyzing flutter in fixed wing aircraft.

To simplify the analysis for this latter source, the rotor is assumed to be turning in still air, and, thus, for no inflow, unsteady aerodynamics caused by the trailing wake can be neglected. Consequently, the aerodynamics for a single blade is similar to that of a fixed wing with a free-stream velocity that varies linearly from the root to the tip, assuming that the shed wake of the preceding blade dies out sufficiently fast so that the oncoming blade will encounter essentially still air.

Flutter analysis typically requires an iterative procedure to reach a converged and physically meaningful solution at a particular operating condition. The Theodorsen function introduces amplitude and phase modifications in aerodynamic forcing, but is dependent on the frequency of structural vibration. The frequency of vibration is not known until the aeroelastic problem is formulated and solved. Therefore, a "guess" of the expected excitation frequency is made and iteration occurs until the guess and the prediction have converged (Figure 59). This process can be quite time consuming if a number of modes are to be tracked manually across a variety of rotor rotating speeds. BLAST utilizes an automated procedure to track a subset of system modes without user interaction. The resulting frequency and damping curves can be inspected for potential flutter conditions.



Increase rotor speed
At a given rotor speed, guess frequency of flutter mode
Form mass, stiffness and damping matrices; perform complex eigensolve
Verify that flutter mode frequency guess is correct: Yes/No
Yes
No
New frequency guess
Record results: damping and frequency

Figure 59 The classical flutter analysis iterative process.

BLAST allows for a number of analysis methods to investigate the aeroelastic stability of a wind turbine blade. The conventional usage considers a blade rotating at a constant rotor speed and performs aeroelastic analysis considering gyroscopic and unsteady aerodynamic effects. A modified usage considers a parked blade exposed to a constant wind speed. For this analysis option, aeroelastic analysis is performed using unsteady aerodynamics without rotational effects.

In addition to analysis procedures, BLAST contains visualization options to characterize aeroelastic behavior of a HAWT blade. Simple mode shape plots present the "in-phase" and "out-of-phase" mode shapes for the fundamental deformation components of the blade flexural axis. A more advanced visualization procedure is also available that illustrates the mode shapes of the actual blade structure and can produce animations of the mode shapes with respect to time.

Disclaimer

Aeroelastic instability analysis is an advanced level topic that can be challenging to learn and understand. NuMAD users who wish to truly understand and interpret their analyses should become intimately acquainted with the theory and application behind this tool. Following are some useful references and courses that may be of use:

- Wright and Cooper, *Introduction to Aircraft Aeroelasticity and Loads,* John Wiley and Sons, Ltd. 2008.
- Bielawa, *Rotary Wing Structural Dynamics and Aeroelasticity, 2nd Ed,* AIAA Education Series, 2006.
- Hodges and Pierce, *Introduction to Structural Dynamics and Aeroelasticity, 2nd Ed,* Cambridge University Press, 2011.
- Bisplinghoff, et.al. *Aeroelasticity*, Dover Publications, 1955.
- AIAA Short Course, *Aeroelasticity: State-of-the-Art Practices.*
- KU Continuing Education Course, *Principles of Aeroelasticity.*

Additionally, the article by Hansen[18] is specific to flutter of HAWT blades and contain a wealth of information on aeroelastic stability analysis for wind energy.

Flutter Analysis Inputs

The flutter analysis requires several inputs in order to set up and perform the analysis. Inputs come from a variety of sources: AeroDyn,[19] FAST and the NuMAD model.

AeroDyn Inputs

The following parameters from the AeroDyn input file are utilized by BLAST:

- Rho, air density used to compute aerodynamic loads
- NumFoil, number of airfoil performance files
- FoilNm, paths to airfoil performance files are used to load the airfoil performance data and compute the LCS values (optional)
- BldNodes, number of aerodynamic blade nodes
- The table of RNodes, Chord and NFoil information is used to determine placement of the airfoil performance, represented by LCS values, and compute necessary parameters required for determination of unsteady aerodynamic forces.

FAST Inputs

The following parameters from the FAST input file are utilized by BLAST:

- TipRad, rotor tip radius
- HubRad, hub radius

FAST Blade Inputs

The following parameters from the FAST blade input file are utilized by BLAST:

- NBlInpSt, number of blade input stations in the blade properties table; used by BLAST to set the number of beam finite elements
- BlFract, blade span fraction; spanwise location of beam finite element nodes in BLAST; all spanwise parameter distributions are resampled based on these spanwise locations.
- AeroCent, along with pitch axis location (NuMAD x-offset) is used by BLAST to compute the location of the actual aerodynamic center

- StrcTwst, used to define the twist of the beam finite elements in BLAST
- BMassDen, used to compute mass matrix
- FlpStff, EdgStff , GJStff, EAStff, used to compute stiffness matrix
- Alpha, read and used by BLAST but not verified.
- FlpIner, EdgIner, used to compute mass matrix
- PrecrvRef, not used
- PreswpRef, used along with FlpEAOf to define the location of beam finite elements in the flapwise direction
- FlpcgOf, used to compute the mass matrix
- EdgcgOf, used to compute the mass matrix
- FlpEAOf, used along with PreswpRef to define the location of beam finite elements in the flapwise direction
- EdgEAOf, used to define the location of beam finite elements in the edgewise direction

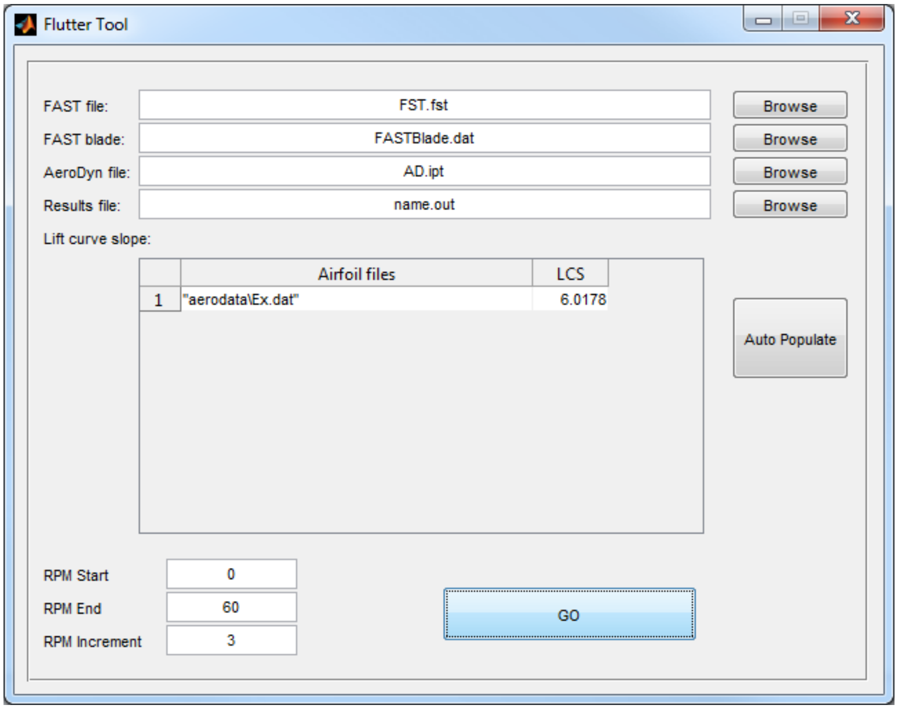
NuMAD Inputs
The following parameters directly from the NuMAD blade model are utilized by BLAST:

- X-offsets, used along with AeroCent from the FAST blade input file to compute the location of the actual aerodynamic centers
- Blade reference line, currently not used by BLAST

Flutter Analysis Example

Figure 60 shows the classical flutter analysis input dialog box that appears upon choosing the “Advanced >> Flutter…” menu item. The user must point to a FAST blade input file and an AeroDyn input file, each of which are generated independently prior to the flutter analysis. NuMAD may be used to manage the process for computing the FAST blade file. The AeroDyn input file must be created through other means. The Matlab scripts which are found in the SNL NuMAD Toolbox may be quite helpful in this process.

The user must determine the LCS values for each airfoil in the AeroDyn file. Values can be set manually, or NuMAD may be used to query the AeroDyn input files in order to calculate the slopes using the “Auto Populate” button. Finally, the user specifies the range of rotor speeds and increment of rotor speeds for the analysis.



FAST file: FST.fst

FAST blade: FASTBlade.dat

AeroDyn file: AD.ipt

Results file: name.out

Lift curve slope:

	Airfoil files	LCS
1	"aerodata\Ex.dat"	6.0178

RPM Start: 0

RPM End: 60

RPM Increment: 3

Figure 60 Flutter analysis inputs dialog box.

A set of simple example files is included with NuMAD in order to demonstrate the process. The files are found in the *FlutterEx* folder in the example files zip package. The setup for the flutter analysis is shown in Figure 60. Figure 61 shows the analysis results.

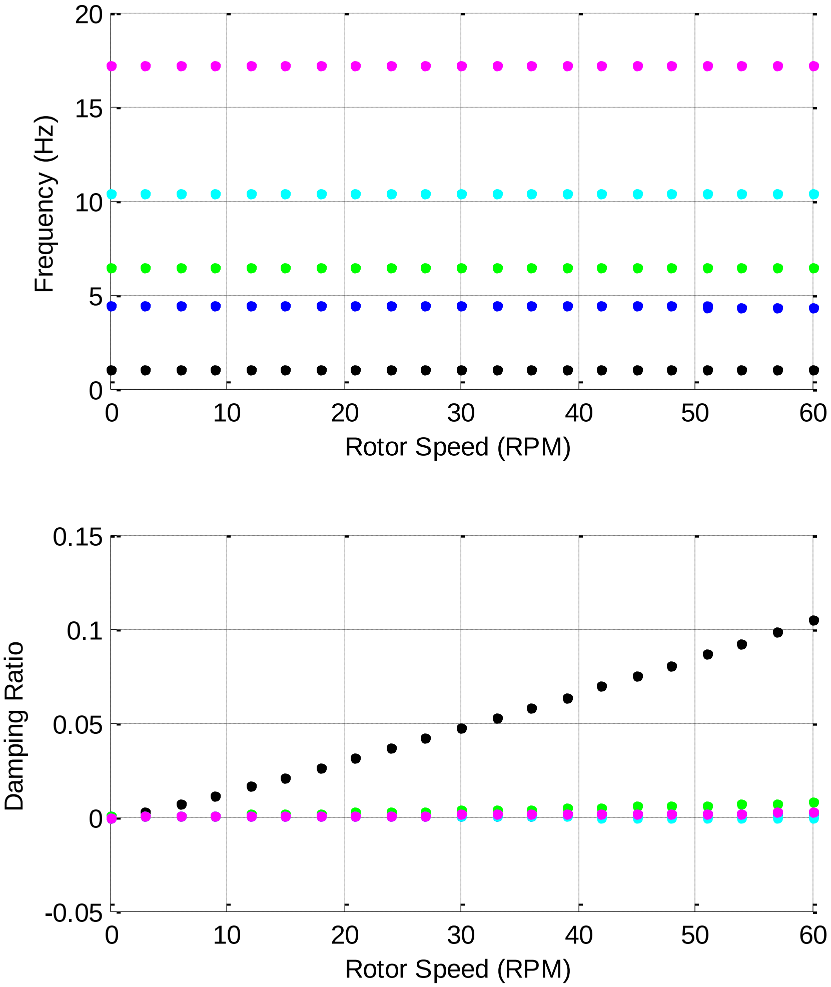


Figure 61 Classical flutter analysis results for a simple uniform blade example.

The algorithm is hard-coded to compute the eigenvalues and eigenmodes of the first 10 modes. Generally, the modes appear as complex conjugate pairs. The frequencies of this blade do not vary much with increasing rotor speed. On first glance, the damping values for the modes are increasing with increasing rotor speed. Upon closer inspection in Figure 62, there is a mode where with a damping trend that falls below zero, indicating flutter instability, at about 43 rpm rotor speed. The frequency of the mode at 43 rpm is about 10.3 Hz.

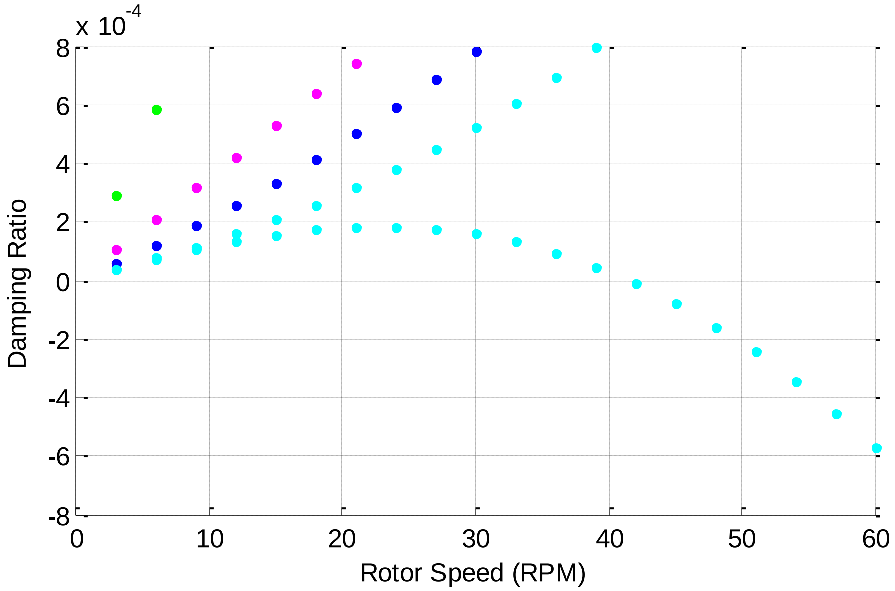


Figure 62 Zoom in on zero-crossing of damping for Mode 7.

BLAST saves a Matlab data file called *flutterInput.mat* during the flutter analysis. The data structure in that file can be used to run another analysis from the Matlab command line in order to visualize the mode shape for this mode at this operating speed. The commands are as follows:

```
omega=43;
freq=10.3;
fea(flutterInput,omega,freq,1,'F');
```

This function outputs the mode shapes for all modes at this operating speed. Note that the aerodynamic forcing frequency used in the input is set to 10.3 Hz, so only those modes responding at 10.3 Hz are valid mode shapes. Figure 63 shows a depiction of the shape of the 10.3 Hz mode at 43 rpm. The mode is dominated by torsional motion coupled with flapwise motion, which is typical of a classical flutter mode shape. Note that the example flutter files provided here represent a fictitious uniform blade. The amount of torsion illustrated is large compared to what is normally observed for a more realistic blade geometry. The typical flutter mode shape for a realistic blade typically exhibits a mix of motion in all degrees of freedom, but dominated by flapwise motion and torsion.

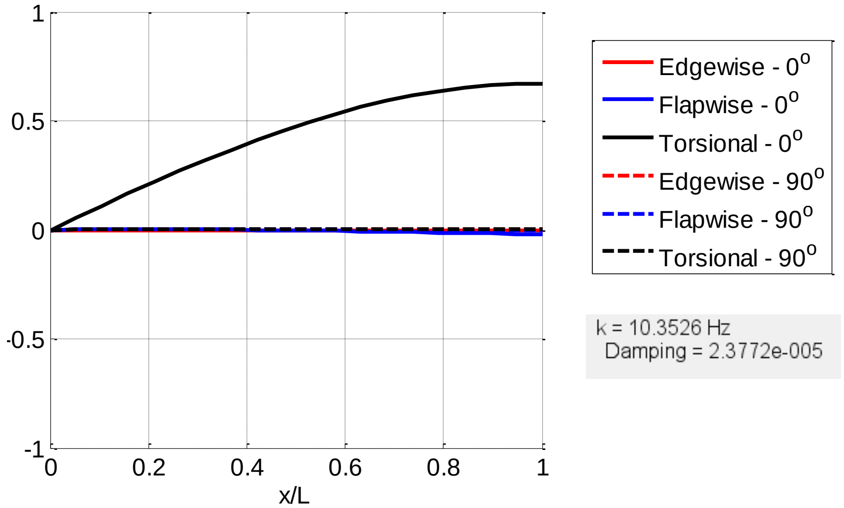


Figure 63 Flutter mode shape at 43 rpm and 10.3 Hz.

REFERENCES

[1] Mathworks Matlab. Natick, Massachusetts. https://www.mathworks.com/

[2] ANSYS Inc., Canonsburg, PA 15317, http://www.ansys.com

[3] NWTC Design Codes (FAST by Jason Jonkman, Ph.D.). http://wind.nrel.gov/designcodes/simulators/fast/. Last modified 08-March-2012; accessed 08-March-2012.

[4] NWTC Design Codes (BModes by Gunjit Bir). http://wind.nrel.gov/designcodes/preprocessors/bmodes/. Last modified 28-June-2012; accessed 28-June-2012.

[5] NWTC Design Codes (PreComp by Rick Damiani). http://wind.nrel.gov/designcodes/preprocessors/precomp/. Last modified 28-June-2012; accessed 28-June-2012.

[6] Tangler, J. L. and Somers, D. M., NREL Airfoil Families for HAWTs. *American Wind Energy Association Conference*, 1995.

[7] Laird, D. L., Montoya, F. C. & Malcolm, D. Finite Element Modeling of Wind Turbine Blades. *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005.

[8] Wikipedia Online. http://en.wikipedia.org/wiki/PLOT3D_file_format Accessed 12 April 2012.

[9] Malcolm, D. J. & Laird, D. L. Identification and Use of Blade Physical Properties. *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005.

[10] Malcolm, D. J. & Laird, D. L. Modeling of Blades as Equivalent Beams for Aeroelastic Analysis. *41st AIAA Aerospace Sciences Meeting and Exhibit*, 2003.

[11] Malcolm, D. J. & Laird, D. L. Extraction of Equivalent Beam Properties from Blade Models. *Wind Energy*, 2007, 10, 135-137.

[12] Malcolm, D. J. and Laird, D. L, Identification and Use of Blade Physical Properties. *43rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005.

[13] Malcolm, D. J. and Laird, D. L., Modeling of Blades as Equivalent Beams for Aeroelastic Analysis. *41st AIAA Aerospace Sciences Meeting and Exhibit*, 2003.

[14] Resor, B., Paquette, J. & Laird, D., An Evaluation of Wind Turbine Blade Cross Section Analysis Techniques, *51st AIAA Structures, Structural Dynamics, and Materials Conference*, 2010.

[15] Resor, B. & Paquette, J., Uncertainties in Prediction of Wind Turbine Blade Flutter. *52nd AIAA Structures, Structural Dynamics and Materials Conference*, 2011.

[16] Lobitz, D. W., Aeroelastic Stability Predictions for a MW-sized Blade. *Wind Energy*, 2004, 7, 211-224.

[17] Leishman, J. G., Challenges in Modeling the Unsteady Aerodynamics of Wind Turbines. *Wind Energy*, 2002, 5, 85-132.

[18] Hansen, M. H., Aeroelastic Instability Problems for Wind Turbines. *Wind Energy*, 2007, 10, 551-577.

[19] NWTC Design Codes (AeroDyn by Dr. David J. Laino). http://wind.nrel.gov/designcodes/simulators/aerodyn/. Last modified 28-June-2012; accessed 28-June-2012.

APPENDIX A: NUMAD TOOLBOX CONTENTS

The NuMAD source code zip-package contains many Matlab scripts which may be useful for researchers who are setting up and solving wind turbine aeroelastic problems and are fluent in the Matlab programming environment.

The files found in "SNL_NuMAD_Toolbox/Docs" contain some information about these scripts. Figure 64 show a flowchart of a wind turbine aeroelastic analysis process (also found as file named *SNLWindToolbox.pdf* in the *Docs* folder). The purpose of this diagram is to shown the pieces of the analysis framework and to show the names of Matlab scripts which can be helpful to an analysis process.

Many of these scripts are called as part of NuMAD. Many are not called by NuMAD, but are used individually on a regular basis to support research projects at Sandia. Most scripts are fairly mature, but some are still evolving. NuMAD users are encouraged to use them if they are helpful.

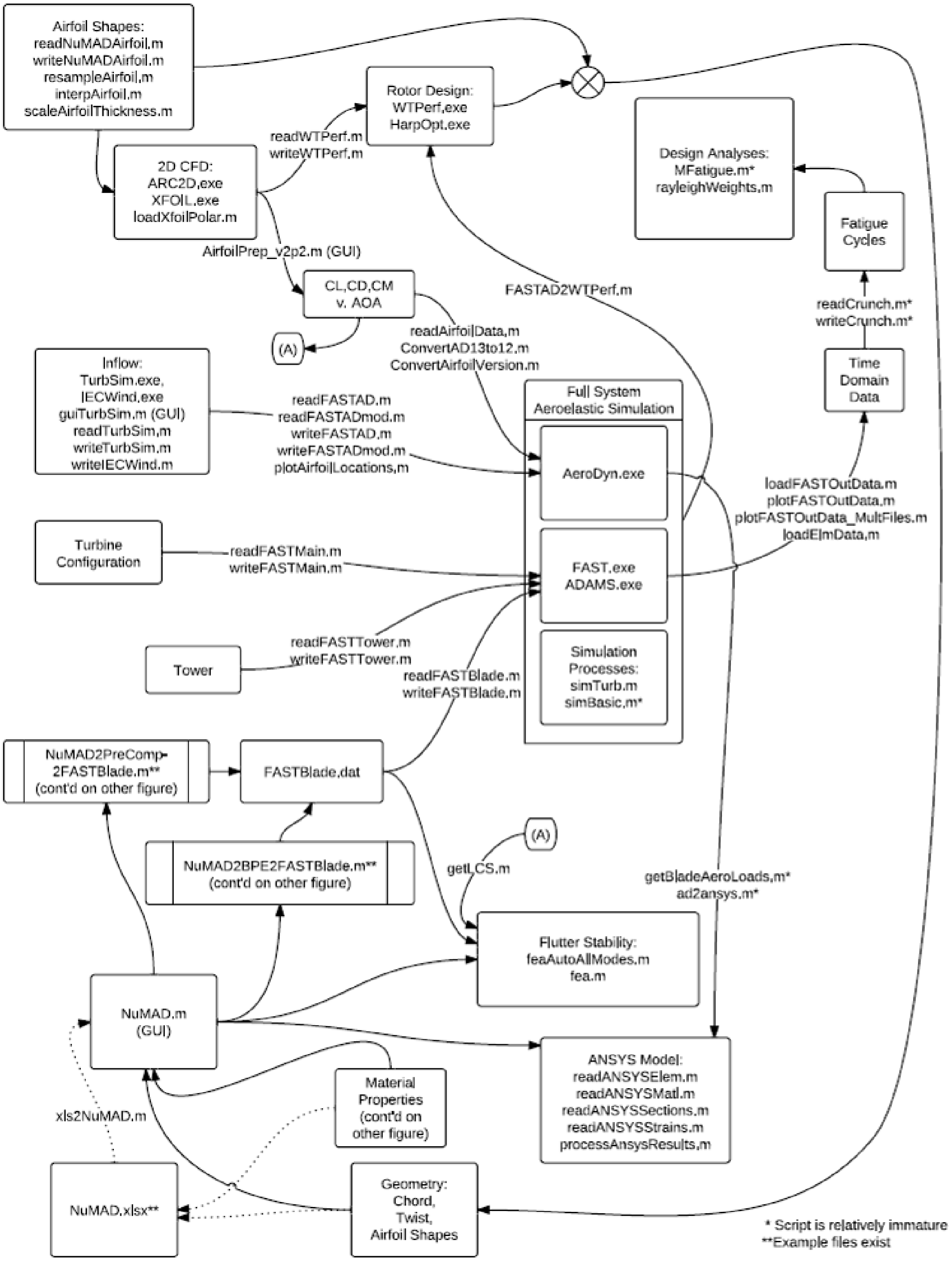


Figure 64 SNLWindToolbox.pdf: interconnections of wind turbine aeroelastic analyses processes.

APPENDIX B: NUMAD.XLS(X) FORMAT

The workbook contains three worksheets: Materials, Geometry and Parameters

'Materials' Format

This sheet has 3 header lines and 17 columns (columns A-Q). Entries in rows beyond the actual material information must be left blank.

- Column A – Material ID; integer numbers in increasing order from 1 to N, where N is the total number of materials in the model; entered manually; rows below the final material entry must remain blank
- Column B – Material Name; Text string with no spaces
- Column C – Type of material; There are two choices: isotropic and orthotropic
- Column D – Layer thickness; units of mm; thickness of a single layer of this material
- Columns E thru G – Young's Modulus Ex, Ey (Ez not used for orthotropic material; Ey & Ez not used for isotropic material); units of MPa
- Columns H thru J – Shear Modulus Gxy; used only for orthotropic material; units of MPa
- Columns K thru M – Major Poisson's ratios; unitless (pryz and prxz not used)
- Column N – material density; units of kg/m^3
- Column O – Ultimate tensile strength; units of MPa (currently not used)
- Column P – Ultimate compressive strength; units of MPa (currently not used)
- Column Q – Reference note; text string; spaces are allowed

'Geometry' Format

This sheet has many columns. The number of columns is related to the number of stacks, material segments and shear webs. Entries in rows beyond the actual station information must be left blank.

Fixed Entries

- Entry A1 – Project file name; text string with no spaces; do not include file extension
- Entry F1 – N_sw; Enter the number of shear webs in the blade
- Entry H1 – N_stk; Enter the number of stacks contained in the blade definition; a stack is defined as a homogenous combination of individual layers, example: 4 layers of carbon fabric is one stack of carbon.
- Entry H2 – N_seg; Enter the (maximum) number of segments around the perimeter of a blade station; it is required to include a segment edge at the leading edge of each station; the maximum number of segments allowed is 24.
- Column A – Station number; Integers 1 to N_sta; entered manually
- Column B –Span wise location of station along the blade; units of meters; first station must be at 0 meters
- Column C – Airfoil shape file name text string, filename; reference to NuMAD format airfoil shape text file located in the folder 'airfoils'; assumes filename convention filename.txt (do not include .txt extension in Column C text string). If airfoil shape is not known, then the shape can be determined using interpolation by entering the text string *interp* in place of a file name. The shape will be determined based on adjacent defined

shapes and will be saved into a NuMAD formatted airfoil shape file in the folder 'airfoils'.

- Column D – trailing edge type text string; options are limited to *round*, *flat* or *sharp*. Must fill in airfoil type entries at all stations, whether shape is interpolated or not. Whenever you are specifying the trailing edge type for interpolated shapes that are based off of a combination of round or sharp with a flatback airfoil, you must specify a flat TE type for the interpolated shape. Specification of a non-flat TE type in this case will cause the airfoil coordinates to be saved incorrectly.
- Column E – Twist; enter a value, in degrees, for the twist of the station. Leave field blank if the station shape is based on interpolation (i.e. Column C entry is interp) to calculate the value based on adjacent values.
- Column F – Chord length; enter a value, in meters, for the station chord length. Leave field blank if the station shape is based on interpolation (i.e. Column C entry is interp) to calculate the value based on adjacent values.
- Column G – X-offset of station; enter a value, in fraction of chord, for the x-offset. Leave field blank if the station shape is based on interpolation (i.e. Column C entry is interp) to calculate the value based on adjacent values.
- Column H – Aerodynamic center of station; enter a value, in fraction of chord from leading edge, for the aerodynamic center. Leave field blank if the station shape is based on interpolation (i.e. Column C entry is *interp*) to calculate the value based on adjacent values.
- Column I – Station label; unused by NuMAD; use for entering optional user notes about the blade layup

Variable entries
Beginning with Column J.

Stacks of layers
Subsequent N_Stk columns – There is a column for each stack of layers.

- Row 1 of each column is an integer representing a stack ID. Stack ID should begin at 1 and increase by one with each additional column. There should be a total of N_Stk (Entry H1) columns.
- Row 2 of each column is the Material ID (Materials sheet, Column A) associated with the material in the stack.
- Row 3 is a text string, with no spaces, used to name the stack (not used by NuMAD)
- The next N_sta rows, beginning at row 4, are integers representing the number of layers of the specified material to include at that station. Entries that are left blank will be determined with linear interpolation based on span location and defined adjacent values.

Note: Currently in creation of the NuMAD model files, each stack is represented in the ANSYS model as an equivalent single layer with thickness equal to number of plies times ply thickness.

Segment edge locations
Subsequent (N_seg+1) columns – Entries in this collection of columns are the fractional chord locations of segment edges at each station. There are additional columns for each segment edge

around the blade station. N_seg columns is equal to the maximum number of segments at any station in the blade.

- Row 1 - The first row of all columns in this group is an integer representing the segment ID, with exception of the first column of the first row in this group, which is arbitrary.
- Row 2 – Blank
- Row 3 – The third row of all columns in this group is a text string representing the segment name (no spaces). This name, along with span location, will be used in formation of a name for the composite material assigned to the segment.
- Rows 4 and beyond - The next N_sta rows in this group, beginning at row 4, are entries representing the location of the segment edges, in fraction of chord length, in relation to the blade station leading edge. Negative values are used for the HP skin segments and positive values are used for the LP skin segments. Numerical entries in the first column in this group must always be set to negative 1, indicating the trailing edge of the HP blade skin. The final column in the group must always be set to positive 1, indicating the trailing edge of the LP surface. It is best practice to include a segment edge at the leading edge of the station, i.e. zero. Leave entries blank if they are to be linearly interpolated based on span location and adjacent values. Enter 9999 for any segment edges that should be ignored due to the presence of NuMAD DP point types other than "single."

Segment edge DP types

Subsequent (N_seg+1) columns – Entries in this collection of columns are text strings indicating the NuMAD DP type for each segment edge around each blade station.

- Row 1 - The first row of all columns in this group is an integer representing the segment ID, with exception of the first column of the first row in this group, which is arbitrary.
- Row 2 – Blank.
- Row 3 – Blank.
- Rows 4 and beyond - The next N_sta rows in this group, beginning at row 4, are text string entries representing the DP type of the segment edges. Entries can be left blank for DP type *single*. Text string entries should be included for DP types as follows: *hourglass*, *flare* and *double.* (*hourglass* and *flare* are not currently supported)

Stack sequences

Subsequent N_seg columns – Entries in this collection of columns are comma separated value (CSV) lists of the stacks included in each segment.

- Row 1 - The first row of all columns in this group is an integer representing the segment ID.
- Row 2 – Blank
- Row 3 – The third row of all columns in this group is a text string representing the segment name (no spaces). This name is for the user's information only as it is not used by NuMAD. It is good practice to make these names match the names in the collection of columns associated with the segment edges.
- Rows 4 and beyond - The next N_sta rows in this group, beginning at row 4, are entries representing the sequence of stacks used to form an entire composite in each segment at

each station. Entries are comma separated values, where the values are integers corresponding to stack ID's defined in the collection of columns describing the layer stacks. The first entry in the CSV list is the exterior blade surface stack. Example: given stack ID / names as follows: 1/GelCoat, 2/DBGlass and 3/Balsa. A composite layup consisting of gel coat, glass, balsa then glass would be entered as follows: *1,2,3,2*. Entries for segments that are irrelevant due to presence of non-single DP's are left blank. If a segment has only one stack, then the comma should still be included in order to maintain the field's text properties. Example: a blade of monolithic 1/Aluminum should be entered as follows: *1,* (one-comma)

Shear web information

Subsequent 2xN_sw column collection – Entries in this collection of columns are CSV lists of the stacks included in each shear web and CSV lists of segment edge ID's for shear web connection points.

- Row 1 – Arbitrary; column labels
- Row 2 – Arbitrary; column labels
- Row 3 – Arbitrary; column labels
- Rows 4 and beyond – There will be N_sw pairs of columns in this collection of columns. Their content is described as follows:
    - First column in a single pair of columns: The next N_sta rows in this group, beginning at row 4, are entries representing the sequence of stacks used to form a composite in each shear web at each station. Entries are comma separated values, where the values are integers corresponding to stack ID's defined in the collection of columns describing the layer stacks. Example, use stack ID / names as follows: 2/DBGlass and 3/Balsa. A composite layup consisting of glass, balsa then glass would be entered as follows: *2,3,2*. Entries for stations that are irrelevant due to no presence of shear web are left blank.
    - Second column in pair of columns: The next N_sta rows in this group, beginning at row 4, are entries representing the sequence of NuMAD DP's, or segment edges, used to located a shear web at each station. Entries are comma separated values, where the values are integers corresponding to DP ID's. There are always four entries in the CSV list and the order goes as follows: inboard LP, inboard HP, outboard HP, outboard LP. Keep in mind that corresponding DP ID's change as DP's other than type "single" are utilized. Entries for stations that are irrelevant due to no presence of shear web are left blank. If the shear web goes all the way to the blade tip, do remember to leave the last row of stack and DP information blank (the most outboard shear web is defined by the information that is contained in the row associated with the next inboard station).

Other Notes on the 'Geometry' sheet

Xls2nmd requires blank entries beyond row N_sta+3. However, additional values may be placed in far right columns in the first N_sta+3 rows. For example, it can be useful to calculate the absolute distances of segment edges at all points for the purposes of generating an Excel chart showing boundaries of various layup regions.

In general, Excel charts can be placed in any location in the sheets of this workbook without affecting the file reading operations.

‘Parameters’ Format

The Parameters sheet follows no specific format because it is not utilized by NuMAD. The entries on this sheet are meant as inputs that may be utilized by formulas other parts of the workbook. For example, one may wish to include a parameter labeled ‘spar cap width’ with numerical field set to 500mm. This field can be used by formulas in the Geometry sheet in order to calculate the location of the segment edges. Thus, modifications to ‘spar cap width’ are made easy.

ANSYS Options

The element size setting in the NuMAD model file that is created in this process is set by default to 2.5% of maximum chord length for use with 4-node shell elements. The boundary condition flag is set for a fixed root.

APPENDIX C: COMMON APDL COMMANDS

- View layers in shell elements: /ESHAPE,1 then EPLOT
- Select elements based on real constant: ESEL,S (or A), REAL,,*real_constant_number.* Useful for selecting specific portions of the model as components (shear webs, panels, etc.)
- Adjust the number of errors until abort: /NERR,*nmerr*,*nmabt*
- Apply force to end node of blade: *GET,var1,node,0,num,max -- then -- F,var1,*comp,value* -- where comp is one of FX, FY, FZ, MX, MY, MZ.
- Show the "deck of cards" display of the layers in an element: LAYPLOT,*element_number*
- Selecting Elements with a Specific Layer Number: Use the ESEL,S,LAYER,*n* command to select elements that have a certain layer number. If an element has a zero thickness for the requested layer, the element is not selected.
- Specifying a Layer for Results Processing: Use the LAYER command (in the POST1 postprocessor) to specify the layer number for which results should be processed.
- The SHELL command specifies a TOP, MID, or BOT location within the layer.
- Check elements for warnings or errors. Select the elements that trigger: CHECK,ESEL,ERR (or WARN) Then, EPLOT to see the elements that have been selected.
- Verify shell normals. /NORMAL,ALL,*n* where *n* is either 0, 1, or -1. Value of 1 enables viewing of all elements with positive normals toward the view point.
- Specification to view results in element coordinate system: RSYS,SOLU and also SHELL,MIDDLE
- View mass of selected elements: PSOLVE,ELFORM

Simple Static Analysis

```
/SOLU
ANTYPE,STATIC
*GET,var1,node,0,num,max
F,var1,FY,1
SOLVE
FINISH
```

Modal Analysis (rotating)

```
/SOLU
ANTYPE,STATIC
PSTRES,ON
OMEGA,0,2.0944,0
SOLVE
FINISH
/SOLU
ANTYPE,MODAL
MODOPT,LANB,6
PSTRES,ON
SOLVE
FINISH
```

Buckling

```
!Eigen Buckling
/SOLU
ANTYPE,0
PSTRES,1
bcsoption,,incore
SOLVE
FINISH
/SOLU
ANTYPE,1
BUCOPT,LANB,10,0,0,
MXPAND,10,0,0,1,0.001,
SOLVE
```

Non-linear Buckling

```
/PREP7
DOFSEL,S,FX,FY
FSCALE,4,
DOFSEL,ALL
z_maxdisp=0.082602
z_inperf=0.05
UPGEOM,z_inperf/z_maxdisp,1,5,'Spar_EV_Buckle','rst',' '
/SOL
ANTYPE,0
NLGEOM,1
PSTRES,1
TIME,4
NSUBST,40,400,40
OUTRES,ERASE
OUTRES,ALL,ALL
SOLVE
FINISH
```

DISTRIBUTION

1	MS1124	Jonathon C. Berg	6121 (electronic copy)
1	MS1124	Brian R. Resor	6121 (electronic copy)
1	MS0899	Technical Library	9536 (electronic copy)

Sandia National Laboratories