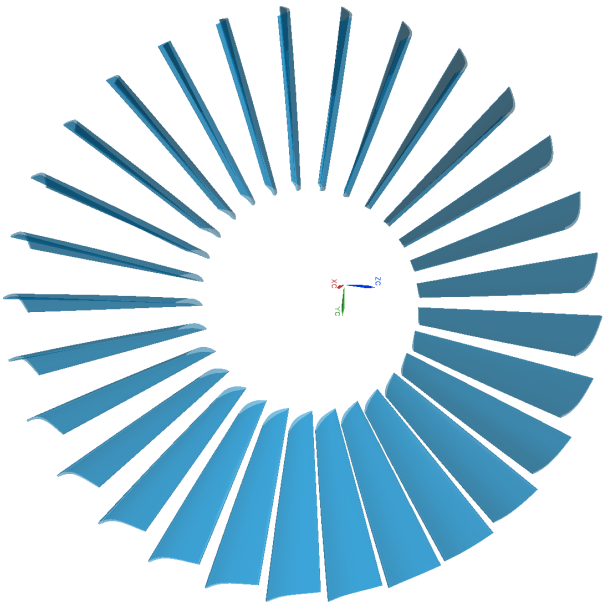
NAX v0.3

THE NON-AXISYMMETRIC TURBOMACHINERY DESIGN SYSTEM



User Guide v0.3

Sandeep Kumar
Gas Turbine Simulation Laboratory
University of Cincinnati, OH,USA

November 2018

Contents

List of Figures

1 Overview

The NAX is an open–source turbo-machinery design system which can be used to produce axisymmetric and/or non–axisymmetric types of blade row(s). The system is based on harmonics approach and uses Fourier coefficients to produce the non–axisymmetric blade shapes. By design, NAX is highly modular and open–structured which helps in its customisation and easy integration of existing turbo-machinery design systems at multiple process levels. This system comprises of three primary modules:

1. Generation of non-axisymmetric design inputs
2. Span controlled geometry generation (3D CAD)
3. Input creation for 3D CFD grid generator

In the present version of NAX , the circumferential non–axisymmetry can be achieved by span-wise variations for the blade properties like blade leading and trailing edge metal angles, true lean or sweep, chord , thickness etc. This system can also be used to produce and place the blades at various angular positions around the annulus. Fourier coefficients/ phase values of generic distortion profile(s) are used to produce perturbations in the blade properties around the annulus to cater any distorted flow conditions. A detailed discussion on all modules in NAX and their working is further present in subsequent sections on this document.

2 System Requirements

The NAX is meant for designers/ engineers with basic understanding of turbo-machinery. No explicit knowledge of any programming language is required to run and use NAX system. The system comprises of a combination of bash/shell and python scripts. Hence, the following package/environments are must for NAX -

1. Shell script - This runs natively on MacOSX/ Linux OS environment (Unix based). The users with Windows OS based machines can download Cygwin package from the internet. Cygwin is a collection of GNU and open source tools which provide functional similarity to a Linux distributions on a Windows OS based machine. The installation package can be downloaded directly from Cygwin official website `https://www.cygwin.com`
2. Python scripts - Different modules of NAX are Python scripts (version 3.5 from v0.3, previously version Python2.7 was used) which also requires users to import standard libraries. These scripts are operating system (OS) independent, provided the Python libraries are pre-installed on user machine. Users can download Python directly from their official website `https://www.python.org/downloads/` based on their operating systems.
   Since, NAX requires multiple libraries , it is advisable here that users can download Python package/distribution manager – Anaconda Python from `https://www.python.org/downloads/` , which is a GUI and navigator to various Python packages. A non–GUI version of Anaconda is : 'conda' package manager. Detailed user–guide can be referred from `https://conda.io/docs/user-guide/install/`

download.html. It is important to note that the users with pre-installed python IDLE versions other that version 3.5 (required here) , must change their Python environment before running NAX. Also, users who are using pre-installed Anaconda/ Conda for Python versions other than version 3.5, can create a separate environment to use NAX, and don't need to uninstall their current versions. The detailed instructions for this process can be referred from page https://conda.io/docs/user-guide/install/index.html

3 Theory Guide

The NAX produces non-axisymmetric blade shapes by introducing perturbations in various blade properties. These properties can be traced back to the native capabilities of T-Blade3. NAX system generates the non-axisymmetric blade shapes by inducing circumferential perturbations in user defined blade properties such as blade angles, lean, sweep, chord, thickness etc. A generic representation of the circumferential spatial variation of any blade property $f(\theta)$, can be specified as

$$f(\theta) = a_0 + a_1 sin(\theta_1 + \phi_1) + a_2 sin(\theta_2 + \phi_2) + ... + a_\infty sin(\theta_\infty + \phi_\infty) \tag{1}$$

$$f(\theta) = a_0 + \sum_{i=1}^{\infty} (a_i sin(\theta + \phi)) \tag{2}$$

$$f(\theta) = Average + Perturbations \tag{3}$$

The NAX is designed to accept and use the results obtained from T-AXI, which contain the blade geometry information calculated based on axisymmetric assumptions. These T–AXI outputs serve as the average value (a_0) and the perturbations are calculated from the Fourier coefficients and phase values obtained using input distortion profile(s). The axisymmetric and non-axisymmetric information file is generated for *each* respective blade row. A detailed discussion on using T-AXI and T-Blade3 suite of codes can be obtained from GTSL, UC website http://gtsl.ase.uc.edu/GTSL/codes.html. The resources on implementation of DFT for numerical analysis using python code can also be referred from Scipy website https://docs.scipy.org/doc/numpy-1.13.0/reference/routines.fft.html.

4 Design using NAX

4.1 Description

As discussed previously, NAX comprises of 3 different modules.The process flow chart for NAX is shown in Figure 1 .

4.1.1 Module-1

The first module of NAX is designed to accept and use the results obtained from T-AXI. It is dedicated to Pre–Processing the data for generation of non-axisymmetric blade inputs. It is further divided into two parts -

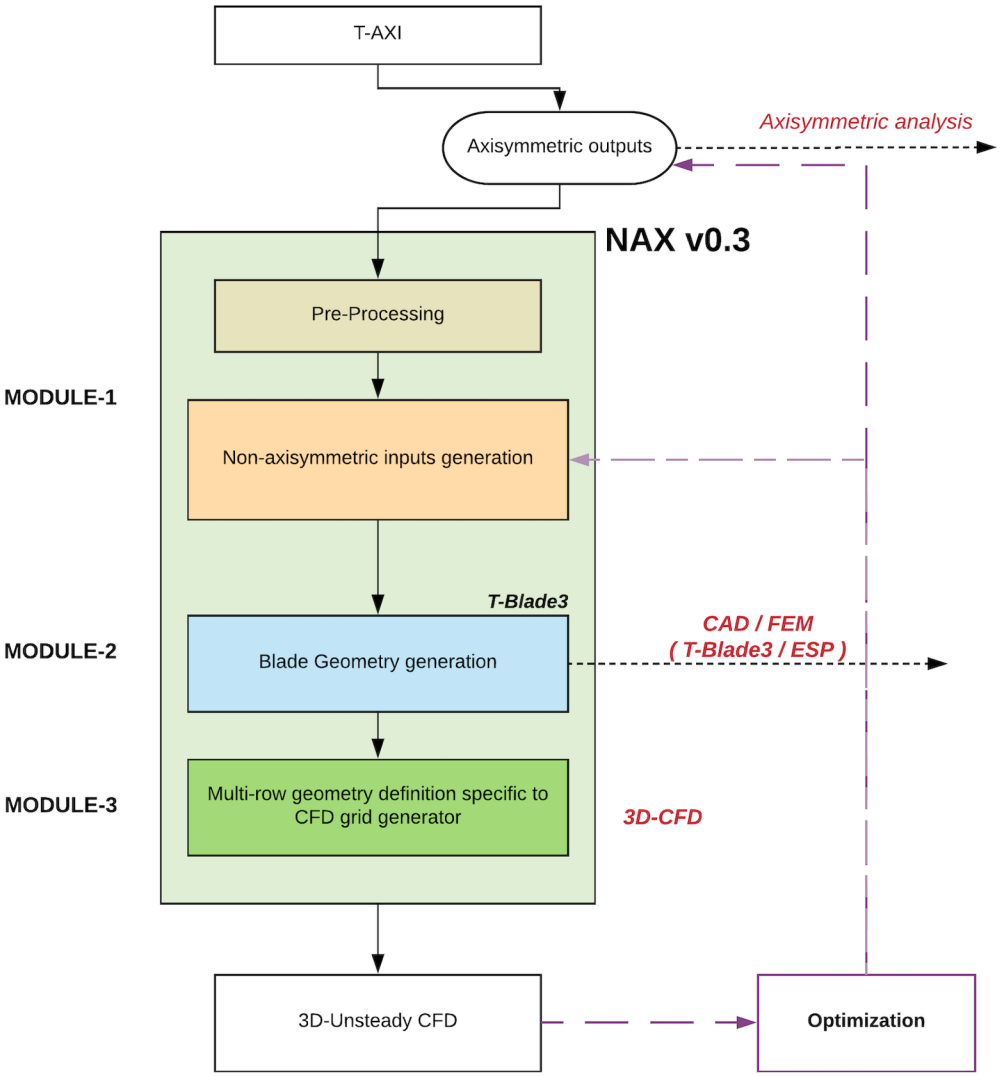


Figure 1: Process flow chart of NAX

Part A : Generation of blade row specific **naxinput** files using input distortion profile(s)
Part B: Generation of non-axisymmetric inputs for 3D geometry using **naxinput** generated in Part A

Please note that the purpose of Part A (Module-1) is to produce **naxinput** files using information in **naxinfo** file. Users can skip the Part A , if they want to define and use their own **naxinput** file. In that case the NAX system should be started from Part B onwards. For customisation of existing NAX system , please see Section 5.
Unlike the axisymmetric design produced by T–AXI, that produces inputs for each bladerow, NAX produces *individual* blade geometry information file for corresponding non-axisymmetric blade row(s). This enables the user to identify any individual blade in physical space.

4.1.2 Module-2

In module-2, the output (non-axisymmetric inputs) from Module-1 (Part-B) is further used for 3D geometry generation/ CAD modelling. For this purpose T-Blade3 is used, which is also an open source–code developed at GTSL, UC (`http://gtsl.ase.uc.edu`). It uses smooth span-wise control of properties to produce versatile blade shapes. T-Blade3 mandatorily uses **spancontrolinputs** files for this smooth span-wise variations. This attribute of T-Blade3 is used in NAX to produce perturbations in blade shapes at individual blade span locations and control design features.

4.1.3 Module-3

The third module of NAX is used to create the geometry definition in native form to grid generators for CFD analysis. In present distribution, this module is capable of producing ***.geomturbo** files which are native format to AutoGrid, used by NUMECA FINE Turbo. Module-3 finally produces a *single* file containing a multi–row information, which can be directly imported into AutoGrid for meshing.

NAX is designed with capability to selectively target a specific blade row to induce non-axisymmetry based on user-defined inputs. NAX can be adjusted to give a complete axisymmetric /non-axisymmetric or a combination of both types of blade rows. Thus, the functionality of overall system is maintained and it can be integrated with other analysis/optimisation tools also. A modular code by design, NAX outputs range from geometry generation and structural analysis (CAD/FEM), 3D CFD analysis, Optimization of design or combinations of all of system design tools.

4.2 Input files

This section discusses the input files and their attributes required to successfully setup and run NAX system.

4.2.1 naxinfo.dat

This is the one of the most important file that contains basic information about the design setup. A preview of the file is available in Figure 2.

```
10  ANALYSIS_TYPE          NON-AXI
11  IGV                    NONE
12  BLADE_ROWS             3
13  OGV                    NONE
14
15  AXI_ROW#               1,3
16  NON-AXI_ROW#           2
```

(a)

```
10  ANALYSIS_TYPE          NON-AXI
11  IGV                    @ROW1
12  BLADE_ROWS             1
13  OGV                    NONE
14
15  AXI_ROW#                  0
16  NON-AXI_ROW#              1
```

(b)

Figure 2: Preview of valid naxinfo.dat files

In this file at Figure 2 (a) we can see that the design is setup for total of 3 blade rows (line -12) and then we can also specify each blade row for its type. In this example Row 1,3(line -15) are declared as axisymmetric type and Row 2 is declared as non-axisymmetric type. The row type *must* be present at line 15 and line 16 as shown. Also,

user can declare the Row-type in any order i.e., in this case user can declare Axisymetric or Non-Axisymmetric rows in sequence of (3, 1) rather than (1,3). Another use of this file is to hold general purpose information about the process and future development of the system. This file thus can be used to store and parse information, for e.g.,when NAX is integrated to an Optimization process. Although, the information in line 10-14 is currently *not* part of any calculations, but it is indended to understand the configuration and for other future usage. Also, please note that in NAX the blade row numbering starts from 1 which *includes* IGV. In present example , the IGV can be considered as row1 and OGV as row3. Another example of valid input is given in Figure 2 (b) where users can define setup for only 1 blade row.

4.2.2 distortion.row#.csv

The distortion profiles are named after the blade rows in which user wants to create non-axisymmetry. For example for given **naxinfo** file as in Figure 2, we will need atleast one 'distortion' profile for each non-axisymmetric type blade rows i.e., row 2. The files in present case should be named as - **distortion.row2.csv** and likewise if any other non-axisymmetric blade row is defined. A snippet preview of the file is available in Figure 3

```
1  -180,42.9751968
2  -170,43.2294438
3  -160,43.3082592
4  -150,43.355625
5  -140,43.4671392
6  -130,43.6981038
7  -120,44.0710368
8  -110,44.5826082
9  -100,45.21
```

Figure 3: Snippet of distortion file

The file must be in ***.csv** format with no header information (notice file starts at line-1). It is important to note that the preview only shows a part of the file, the complete file length contains information from [-180,170] degrees with a grid interval of 10 degrees. First column contains the circumferential coordinates of blade property [-180,170] and second column contains its value. In this case the second column denotes the blade leading edge inlet angles. User can define their own distortion profiles and use them. Also, there is no minimum grid interval (as shown 10 degrees in this example). Based on information in this file, the Fourier coefficients/phase values are calculated to introduce perturbations in the blade rows.

4.2.3 naxinput.#.dat

This file is generated using Part A of Module-1 in NAX. However, user can produce or customise the generated files for their own design results. For customisation of existing NAX system , please see Section 5. Based on the information in distortion profiles (given in file **distortion.row#.csv**) and **naxinfo** file, a Figure 4 is generated with information on amplitude and phase data for *each non-axisymmetric* blade row. This information is generated based on each of the distortion profile(s) provided by user. This helps user to visualise and decide the number of Fourier modes to use for further design. For e.g., in Figure 4 we can observe that first 3 modes play significant role as compare to

rest of modes.Thus, user can choose to include variable number of modes, say (n) for the design. Using Part A of Module-1 in NAX the designer is prompted to enter the number of modes to use for different blade properties like Leading edge blade angles, circumferential position of blades, axisymmetric /true lean and sweep. The essence of this system lies in providing the flexibility to user in deciding arbitrary number of Fourier modes for different blade properties. A set of valid input files are shown in Figure 5.

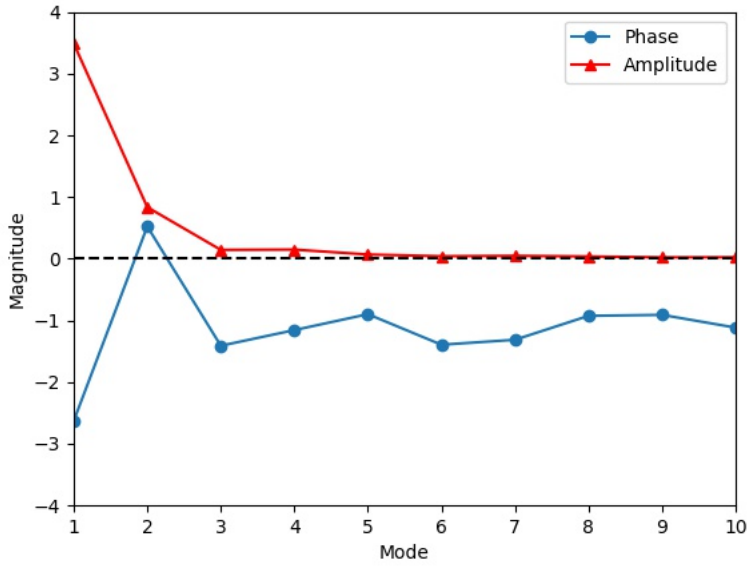


Figure 4: Fourier amplitude-phase data for a row with given distortion profile

These files are created using run-time user inputs and are different for individual blade rows. Here, we can see that "zero" modes means "no–perturbations". Users can choose to use different modes for blade properties (marked in file). Also user can choose to select different number of modes for different blade properties at multiple span-wise locations. If user wishes to use different modes even at different span-wise locations, it can be done by running Module-1 (Part A) and then rest of NAX system scripts. For details please refer section 5. Hence, NAX provides a complete flexibility in defining the non-axisymmetry circumferentially and along span-wise direction.

```
NAX v0.3 - input file
Casename:
nax_case
Bladerow:
2


Sweep Span Control Points     Fourier Modes (Magnitude , Phase)
 4                                    0
0.0000    0.0, 0.0, 0.0, 0.0, 0.0, 0.0
0.3333    0.0, 0.0, 0.0, 0.0, 0.0, 0.0
0.6667    0.0, 0.0, 0.0, 0.0, 0.0, 0.0
1.0000    0.0, 0.0, 0.0, 0.0, 0.0, 0.0

Lean Span Control Points     Fourier Modes (Magnitude , Phase)
 4                                    0
0.0000    0.0, 0.0, 0.0, 0.0, 0.0, 0.0
0.3333    0.0, 0.0, 0.0, 0.0, 0.0, 0.0
0.6667    0.0, 0.0, 0.0, 0.0, 0.0, 0.0
1.0000    0.0, 0.0, 0.0, 0.0, 0.0, 0.0

In_beta Span Control Points     Fourier Modes (Magnitude , Phase)
 4                                    2
0.0000    3.487463375651948, -2.628995440287781, 0.8355054830799211, 0.5227224880665621
0.3333    3.487463375651948, -2.628995440287781, 0.8355054830799211, 0.5227224880665621
0.6667    3.487463375651948, -2.628995440287781, 0.0000000000000000, 0.0000000000000000
1.0000    3.487463375651948, -2.628995440287781, 0.0000000000000000, 0.0000000000000000

Out_beta Span Control Points     Fourier Modes (Magnitude , Phase)
 4                                    1
0.0000    3.487463375651948, -2.628995440287781
0.3333    3.487463375651948, -2.628995440287781
0.6667    3.487463375651948, -2.628995440287781
1.0000    3.487463375651948, -2.628995440287781

Chord_multiplier Span Control Points     Fourier Modes (Magnitude , Phase)
 4                                    0
0.0000    0.0, 0.0, 0.0, 0.0, 0.0, 0.0
0.3333    0.0, 0.0, 0.0, 0.0, 0.0, 0.0
0.6667    0.0, 0.0, 0.0, 0.0, 0.0, 0.0
1.0000    0.0, 0.0, 0.0, 0.0, 0.0, 0.0

tm/c Span Control Points     Fourier Modes (Magnitude , Phase)
 4                                    1
0.0000    3.487463375651948, -2.628995440287781
0.3333    3.487463375651948, -2.628995440287781
0.6667    3.487463375651948, -2.628995440287781
1.0000    3.487463375651948, -2.628995440287781
```

Figure 5: naxinput file with different Fourier modes for blade properties in row 2

4.2.4 3dbgbinput.#.dat and spancontrolinputs.#.dat

The **3dbgbinput** and **spancontrolinputs** files belong natively to T-Blade3 geometry generation system. For NAX, they should be used in corresponding pairs for any specific blade row.The **3dbgbinput** files for each blade row is generated using T-AXI suite of codes.The **spancontrolinputs** files are primarily used as templates, but can be modified and used to control the span-wise property distribution in a blade. Users can download any **spancontrolinputs** file from GTSL/T-blade3 website and modify to use it. These

files contain blade geometry information and act as inputs for *tblade3* executable which generates the 3D CAD geometry files for blade rows. NAX system uses these input files (**3dbgbinput**) along-with the **spancontrolinputs** file to produce the non-axisymmetric inputs for each individual blade in a non-axisymmetric blade row. Thus module-1 of NAX provides inputs for 3D blade shape generation of axi-symmetric or non-axisymmetric blade rows. A much detailed discussion on both of these files can be referred from `http://gtsl.ase.uc.edu/t-blade3/`

4.3 Setting up new design case

This section contains instructions for the user to set up a new design. Download the NAX system and extract the package contents by unzipping it. The downloaded setup should contain the following items as shown in Figure 6. Please note that the executables as shown in Figure 7b can be downloaded directly from T-Blade3 website `http://gtsl.ase.uc.edu/t-blade3/`.The contents of each of the native directories are also shown in Figure 7 .

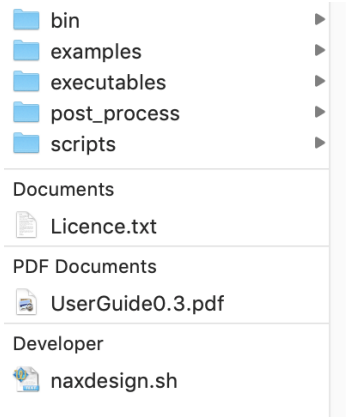
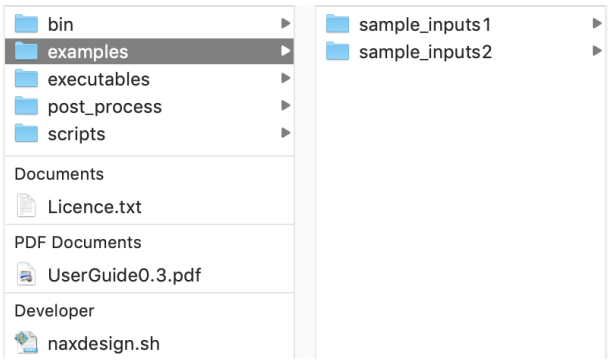


Figure 6: Contents of NAX system

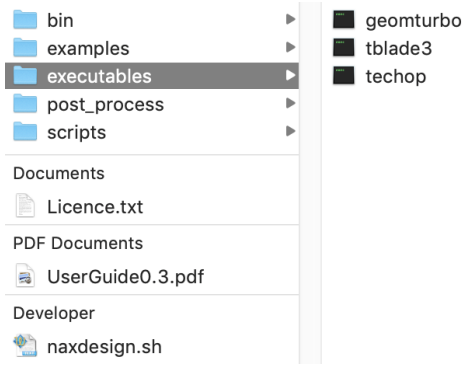
Now, we are ready to set up a new design case. Manually copy the executables (as per your Operating System) from *bin* directory and paste them into *executables* directory. Create a new directory in the current working directory as shown in Figure8.Users can choose any arbitrary name for this directory.The new case directory shown here is named as *new_design*. Now, inside the new case directory *new_design* , create another directory named as ***inputs***. It is to note that this directory *must* be named as *inputs*.

User can set their own case or use files from *examples* directory. A new case must contain **naxinfo.dat** file with blade row information - as axisymmetric (blade row1,3) and non-axisymmetric (blade row 2) as shown in Figure 2. Thus we will require the **distortion.row2.csv** file along-with the respective **3dbgbinput** and **spancontrolinputs** files for each blade rows. Figure9 shows here the contents of this new design case.

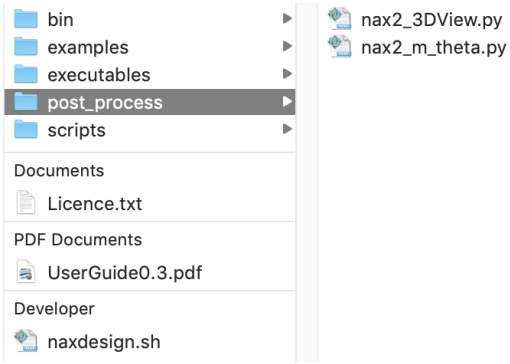
Next step is to run the shell script (*naxdesign.sh*) present in the parent directory of NAX system. For running NAX we must open the terminal window and make the directory *new_design* as your current directory. For Linux OS, we can use Shift +F4 in *new_design* to open terminal window. For Windows OS type 'cmd' in the address bar of *new_design* directory and press RETURN (Enter).While using Cygwin, please launch Cygwin terminal and type 'pwd' for finding your present working directory. Now make the *new_design* as your current directory by typing $cd\ \langle full - path \rangle$ and press return.



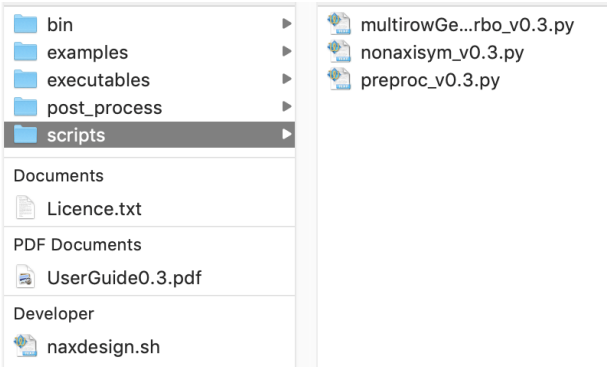
(a) examples directory



(b) executables directory



(c) post_process directory



(d) scripts directory

Figure 7: Contents of different directories in NAX

Mac OSX users can use spotlight and search 'terminal' window and give full path (hint: use 'pwd' command in terminal to know your current working directory)to *new_design* directory. Now, locate the *naxdesign.sh* and run it from *new_design* directory.To do this type following in your terminal window and press return

../*naxdesign.sh* (PRESS RETURN)

For this case we have set-up our design with 29 blades in blade-row2 (non-axi). These information on number of blade are present in **3dbgbinput** files for each blade–row. After pressing RETURN the script *naxdesign.sh* produces non-axisymmetric design based on user inputs in **naxinfo.dat** files. As discussed Module-1 prompts users to provide input for producing '**naxinfo.dat** file for row2 . The amplitude-Phase data is saved in the *inputs* directory for both the non-axisymmetric rows in form of images. The Part A of module-1 in NAX also produces the **naxinput.dat** files for rows 2 .The NAX system then produces separate directories for each blade row(s). Since for axisymmetric blade rows (row1 and row3), all blade are similar in a row, only one directory is produced with blade geometry files.However, since blade-row 2 is non-axisymmetric in nature (as defined) all blades are different from each other. Thus NAX generates separate directory for each blade in blade–rows as shown in Figure 10.

Now , it is to note that the output of Module-2 , the *tblade3* run which produces 3D blade geometry are present in each of these blades directories (shown for row-2 in Figure 10). These geometry files contains a lot of useful information and users are highly

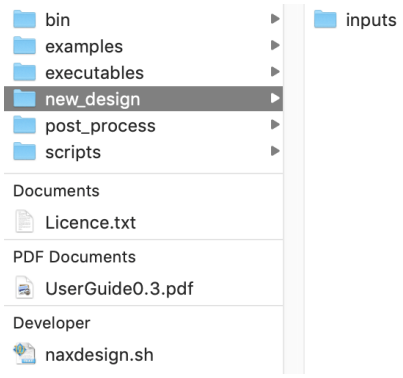


Figure 8: Setup for a new design case

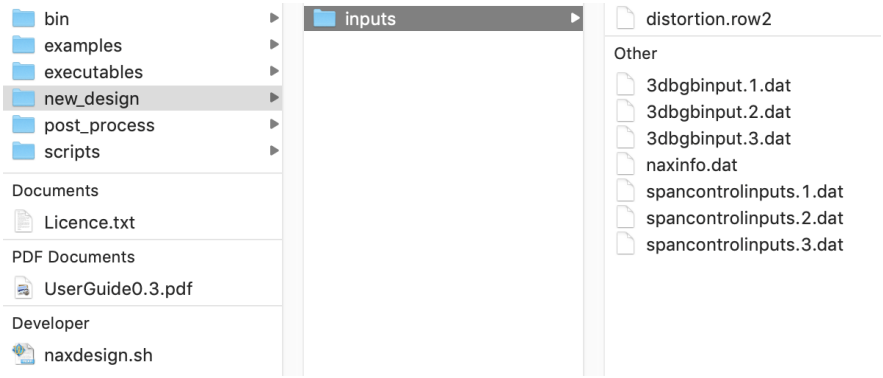


Figure 9: Contents of inputs directory for new design

encouraged to explore them. Also , we can see in *new_design* directory a new multi–row **geomTurbo** file is generated by Module-3 of NAX. This file can directly be imported to AutoGrid for analysis purpose.The directory also contains the *caserun.log* file which logs each step in NAX run. Additionally the complete terminal window run details are present in Figure 11. This completes the design process using NAX. Users can modify the existing inputs to produce geometries as per their design requirements. Users can produce any number of design cases , by following the instructions as discussed in this section.

4.4 post processing

The directory *post_process* comes with 2 visualisation scripts. It is important to note here that this section is not a part of native NAX design system. Users can plot m'-θ plot for full annulus view of user specific section of any specified blade-row. The results for section 10 is shown in Figure 12 for blade rows–2 and row–3. Similarly the other python script can be used to plot a 3D view of generated blade–rows. To visualise m'-θ, make sure that current directory is *new_design*. The results of there scripts are images which are by default saved in the current directory.This script also marks the individual blade so that it can be traced in physical space.To execute the script, specify the row number and blade span-wise section number in the script and run using following command at terminal window

../*nax2_m_theta.py* (PRESS RETURN)

The procedure can be repeated for a 3D plot by running nax2_3DView.py script (see Figure 12). However, please set the required parameters in scripts to get desired results. A 3D view of the resultant blade shape can be seen in Figure 14. This is generated by exporting the section details from t-blade3 run.

5 Advanced use of NAX

This section includes information about different scripts and user guidelines to customise them and produce different functionality. Using information given in this section, user can exploit the benefits of modular and open structure of NAX design system. Specific comments are also provided inside the scripts to help users.

1. Shell script *naxdesign.sh*
   This shell script acts as a 'wrapper' to all of the NAX functions and controls the scripts and executables run. If user wishes to run only a part of the NAX , specific parts of this script can be commented. The shell scripts can be commented using (#) character placed at beginning of the code line. It can used to design a different user flow than the default. Also, it can be used to retain all the blade row or specific blade **geomTurbo** file(s) and still produce 'multi-row' geomTurbo file.The script is itself highly commented with user instructions and is written in same structure as seen in Figure 11. By default the post_processing scripts are not included in this script. However, user can include those commands in this script. Also, if user wants to use Part A of Module-1 separately, please specify that portion of code in this script.

2. Python scripts for NAX
   The NAX system contains primarily 3 python scripts in *scripts* directory.

   (a) *preproc_v0.3.py*
   User can skip the use of this script, if they want to define their own **naxinput** file(s). By default this script runs on the inputs as previously discussed and saves the image for Fourier amplitude and phase. It then prompts the user to choose different modes for specific blade row property. Users can modify this script to set default for the number of modes for individual properties , if they don't want a user prompt.

   (b) *nonaxisym_v0.3.py*
   It is responsible of producing inputs for *tblade3* execution. It reads the data for producing perturbations in blades shapes using the information from **naxinput** files .This script reads the **naxinput** and **3dbgbinput** files against the **line number** of the files. So, please notice that if a new format of T-Blade3 inputs is being used, they need to modify this script. The script code would be similar to the code block for other properties too. Users can refer the existing code snippet as draft.

   (c) *multirowGeomturbo_v0.3.py*
   This script produced a single file for multiple rows of turbo-machinery. During the process, the it produces temporary files for each blade rows. Users can comments the part of script it they wish to generate these geomTurbo files.

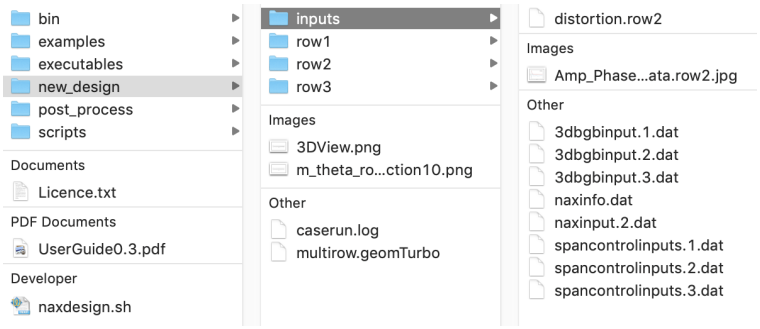
They can also change the default name of final multi–row geomTurbo file, which is by default set as **multirow.geomTurbo**.

3. Post processing python scripts
   Users can uncomment them for a prompt to enter the information like row# , section# , the angle of view/rotation etc. Default values of the inputs are already provided in the script for automation purpose. Also, users can change the images resolution and directory to save the images.

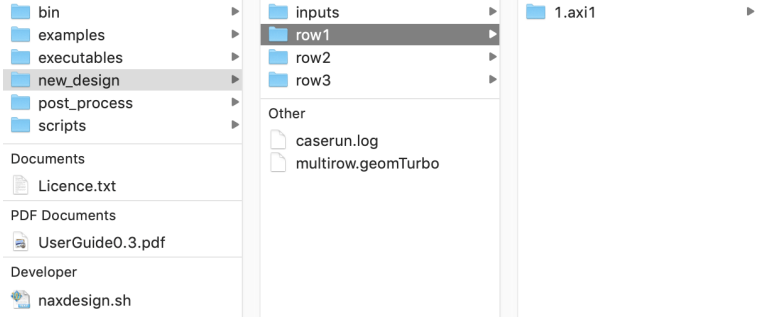
6 Revision History

1. v0.3 - Oct 2018
   This version has 4 major changes . (1) NAX v0.3 is compatible with latest T-Blade3 v1.12 which has extended capability of thickness control. (2) NAX native scripts written in Python 2.7 are now re-written in Python 3.5. This is done to get continual support from Python's future evoluton. (3) NAX can now be used only for 1 blade row case. For previous versions of NAX, users were mandatorily required to specify 2 or more blade rows in naxinfo file. (4) naxinput file generated using Module1 (Part A) is also re-structured to include more property definitions.

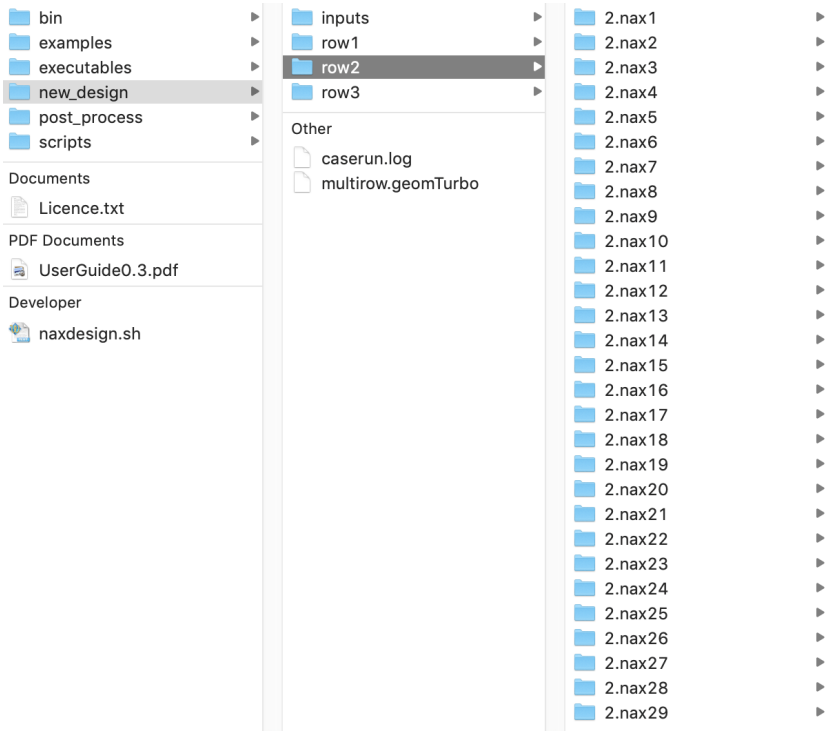
2. v0.2 - Feb 2018
   The format of naxinput and modular structure of code was changed to enhance the capability of NAX. Before this versions , users could only specify inlet/exit angles (alpha) and calculate their circumferential variation. NAX v0.2 comes with capability to include angles, lean, sweep , chord as the blade properties. Also , a pre-processor step was added to Module-1 , which helps writing naxinput file and provides flexibility to the user to choose the number of modes at run time also.



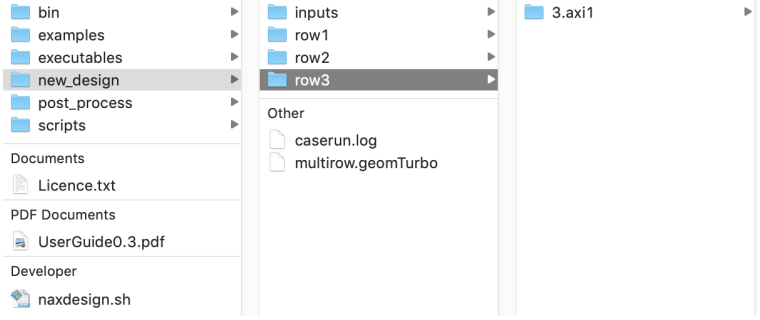
(a) inputs directory after NAX run



(b) axisymmetric blade row1 directory



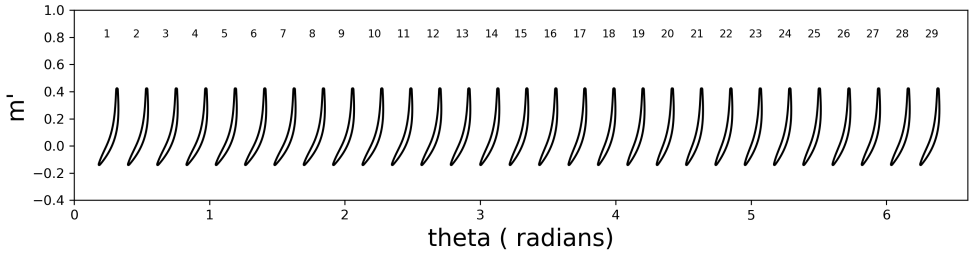
(c) nonaxisymmetric blade row2 directory



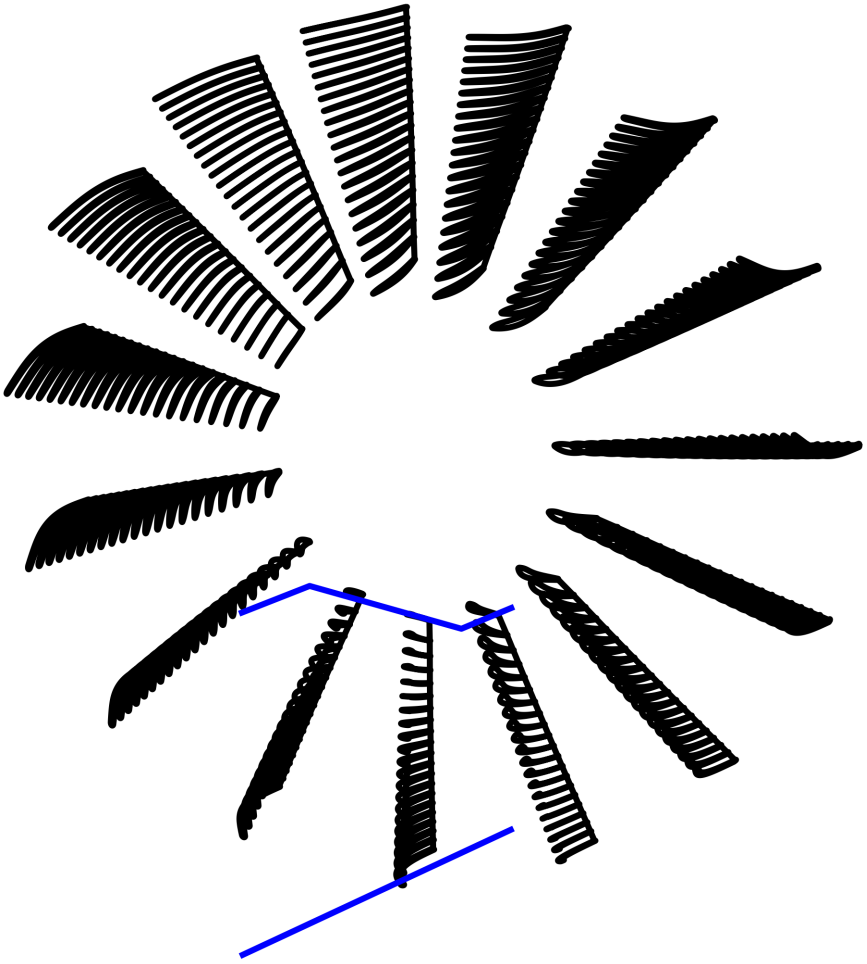
(d) nonaxisymmetric blade row3 directory

Figure 10: Contents of different directories in NAX after running the case

Figure 11: NAX run history on terminal window screen



(a) sample m'-theta plots at section10 - 29 blades



(b) sample 3D Wireframe plot with 15 blades

Figure 12: Sample examples of m'-theta plots using post-processing script

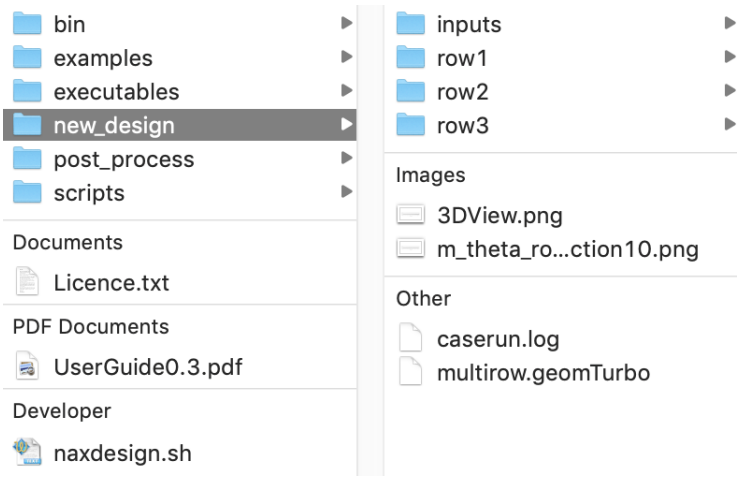


Figure 13: Contents of new design case directory after post processing

Figure 14: Sample Non-Axisymmetric 3D CAD with 29 blades