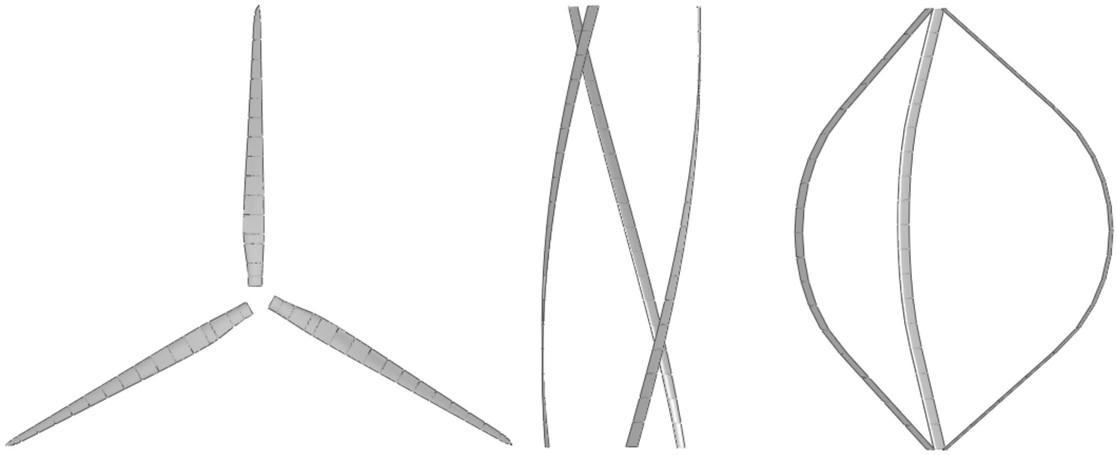
QBlade Guidelines



v0.6

David Marten
Juliane Wendler

January 18, 2013

Contact: david.marten(at)tu-berlin.de

Contents

1 Introduction **5**
1.1 Blade design and simulation in the wind turbine industry 5
1.2 The software project 7

2 Software implementation **9**
2.1 Code limitations 9
2.2 Code structure 9
2.3 Plotting results / Graph controls 11

3 TUTORIAL: How to create simulations in QBlade **13**

4 XFOIL and XFLR/QFLR **29**

5 The QBlade 360° extrapolation module **30**
5.0.1 Basics 30
5.0.2 Montgomery extrapolation 31
5.0.3 Viterna-Corrigan post stall model 32

6 The QBlade HAWT module **33**
6.1 Basics 33
6.1.1 The Blade Element Momentum Method 33
6.1.2 Iteration procedure 33
6.2 The blade design and optimization submodule 34
6.2.1 Blade optimization 36
6.2.2 Blade scaling 37
6.2.3 Advanced design 38
6.3 The rotor simulation submodule 39
6.4 The multi parameter simulation submodule 40
6.5 The turbine definition and simulation submodule 41
6.6 Simulation settings 43
6.6.1 Simulation Parameters 43
6.6.2 Corrections 47
6.7 Simulation results 52
6.7.1 Data storage and visualization 52
6.7.2 Variable listings 53

7 The QBlade VAWT Module 56

7.1 Basics . . . 56

7.1.1 Method of operation . . . 56

7.1.2 The Double-Multiple Streamtube Model . . . 57

7.1.3 Velocities . . . 59

7.1.4 Iteration procedure . . . 59

7.1.5 Limitations . . . 60

7.2 The blade design and optimization submodule . . . 61

7.2.1 Blade optimization . . . 61

7.2.2 Blade scaling . . . 63

7.3 The rotor simulation submodule . . . 64

7.4 The turbine definition and simulation submodule . . . 65

7.5 Simulation settings . . . 66

7.5.1 Simulation Parameters . . . 66

7.5.2 Wind profile . . . 67

7.5.3 Corrections . . . 68

7.6 Simulation results . . . 69

7.6.1 Data storage and visualization . . . 69

7.6.2 Variable listings . . . 70

Bibliography 74

1 Introduction

Solving the worlds energy problems is one of the great topics of our time. Recently, an increase in global awareness has led to a boom in the market for renewable energy technology. To reduce the dependence on fossil fuels, efficient power generation methods in the fields of solar, wind, wave, biomass and thermal energy are in demand. In Germany wind energy plays a central role to reach the goal of an almost carbon dioxide free energy production till 2050 [1]. Because there is a limit to the number of adequate sites for turbines, that do not interfere with natural protection laws or residents, a part of the strategy to keep the development of wind energy on a high level is to replace older turbines at good wind sites with newer, more efficient ones.

A condition for an efficient conversion of the wind energy into mechanical energy with wind turbines is the optimal design of the rotor blades. Methods for rapid development, reliable and robust predictions of the aerodynamic characteristics and simulation of the flow conditions around a rotor blade are essential for this design task.

1.1 Blade design and simulation in the wind turbine industry

The blade design methods for wind turbines originate from the aircraft design industry and apply the same techniques. But the flow conditions that a turbine blade experiences are quite different from those affecting a plane. Hence, a lot of the assumptions made in flight aerodynamics cant't be applied to the complex flow field around a wind turbine. The latter is unsteady, three dimensional, turbulent, roughly incompressible and often separated from the flow contour. The aerodynamics of a wind turbine are influenced by far field conditions far up- and downstream from the rotor. At the same time, they depend on small scale turbulent flow conditions around the blades. This implies the need for a large simulated domain as well as a fine spatial resolution. A full CFD analysis that fulfills these requirements and accounts for all the named effects, is very time consuming and expensive. An alternative to CFD simulations are vortex methods, with the limitation that they cannot model viscous behavior since they are based on potential flow theory.

That is why only [9] design and evaluation tools that are based on the *Blade Element Momentum (BEM) method* are used to predict the efficiency of *Horizontal Axis Wind Turbines (HAWT)* in the industry business. The use of other methods such as CFD, RANS and vortex models is therefore narrowed to research environments. The main advantage of the BEM model compared to CFD is that it is very cost efficient and the computational time is significantly less. The prediction of a wind turbines performance, operating in a fluctuating wind field complicates the application of the BEM method, that assumes a steady state wind field. The BEM model, which is in fact a two dimensional method extrapolated into the third dimension applies semi-empirical corrections, derived from correlations with measurements or full CFD computations, to account for three dimensional effects. TANGLER states that in general the BEM underestimates the overall performance of a Turbine and overestimates the peak power [14]. But nevertheless, the blade element momentum method is widely applied in the wind turbine industry because the use of analysis techniques of such lower order-accuracy greatly simplifies the turbine design. Thanks to the BEM model, it is possible to rapidly develop and test different rotor designs against one another, commit small changes and test again, and in this way evolve a preliminary design that can be studied in greater detail with other techniques, like CFD, later. The power of this iterative approach and the verification of BEM simulations with wind tunnel and field measurements justify the use of BEM computational methods to analyze the blades from a two dimensional point of view. The BEM method's ability for robust analysis and low computational costs make up for most shortcomings and inaccuracies. Virtually, all modern HAWT rotors were designed using this model.

Recently, a new interest in *Vertical Axis Wind Turbines (VAWT)* has can be observed. One reason is their ability to capture wind from all directions, which is utile in urban areas where no main wind direction can be found because of numerous barriers (buildings, trees etc.) and flow channelling effects. However, the installation of small-scale VAWT's on the top of large building is a new idea to support the production of clean energy in urbanized areas. Another approach is the development of huge vertical offshore applications. Though, the rotation of a VAWT blade that passes its own wake during a revolution poses some new problems for the simulation and power analysis of such wind turbines. Extensive research including field tests and software development lead to a model introduced by Ion Paraschivoiu, the *Double-Multiple Streamtube (DMS) model*. It considers the circular path of the blade and the energy extraction in 2 steps (up- and downstream of the rotational axis) and will be the subject of a subsequent chapter.

1.2 The software project

This software project is realized being a part of the wind energy group at the Berlin Technical University Department of Experimental Fluid Mechanics, led by Prof. Dr. Christian Oliver Paschereit. The aim of this project is to provide an open source turbine calculation software that is seamlessly integrated into *XFOIL*, an airfoil design and analysis tool. The motivation for this is to create a one solution software for the design and aerodynamical computation of wind turbine blades. The integration in XFOIL enables the user to rapidly design custom airfoils and compute their polars, extrapolate the polar data to a range of 360°, and directly integrate them into a wind turbine simulation. This step of exporting and importing foil and geometry data between different programs is skipped as well as the associated trouble. At the same time, the integration of the BEM and DMS code into XFOIL's sophisticated GUI will make this software accessible to a huge number of interested people without the usual command line interface software tools. The software is especially adequate for teaching, as it provides a 'hands on' feeling for HAWT and VAWT rotor design and shows all the fundamental relationships and concepts between twist, chord, foils, turbine control and type and the power curve in an easy and intuitive way. The GUI serves as a post processor to conducted rotor simulations as well and gives deep insight into all relevant blade and rotor variables for verification, to compare different rotor configurations, or even to study the numerical algorithm and the dependencies among the aerodynamic variables. In addition to that, the software at hand is a very flexible and user-friendly platform for wind turbine blade design. Hence, it can also act as a modular system for future implementations that can exploit the possibilities that a combination of manual and parametric airfoil design and analysis coupled with a blade design and simulation tool offers.

The functionality of the BEM software includes the following features:

- extrapolation of XFOIL generated or imported polar data to 360° AoA
- advanced blade design and optimization, including 3D visualization, using XFOIL generated or imported profiles
- wind turbine definition (rotor blade, turbine control, generator type, losses)
- computation of rotor performance over λ range
- computation of wind turbine performance over windspeed range
- annual yield computation with WEIBULL distribution

- manual selection of BEM and DMS correction algorithms
- manual selection of all relevant simulation parameters
- data browsing and visualization as post processing
- export functionality for all created simulation data
- blade geometry export in .stl format
- storing of projects, rotors, turbines and simulations in a runtime database

2 Software implementation

2.1 Code limitations

Like the original XFOIL written by MARK DRELA and XFLR, written by ANDRE DEPERROIS, QBlade has been developed and released according to the principles of the *General Public License*. One important point about the GPL is that this program is distributed without any warranty (neither the warranty of merchantability nor the warranty of fitness for a particular purpose). The resulting software is not intended as a professional product and does not offer any guarantee of robustness or accuracy. It is distributed as a personal use application only. This software may not be faultless and there will certainly be more bugs discovered after the distribution. However, a validation against other BEM software permits some trust in the provided results.

2.2 Code structure

The QBlade 360° extrapolation module allows an extrapolation of the AoA to a range of 360°. There are two different methods available: the Montgomery and Viterna extrapolation. The QBlade HAWT and VAWT modules both consist of three submodules:

- blade design and optimization
- rotor simulation
- turbine definition and simulation

The HAWT module additionally provides a characteristic graph submodule. All modules will be discussed in greater detail on the following pages.

An overview of the data objects storing the blades, turbines, polars and simulation data as well as their relation to XFOIL can be found in Fig. 2.1

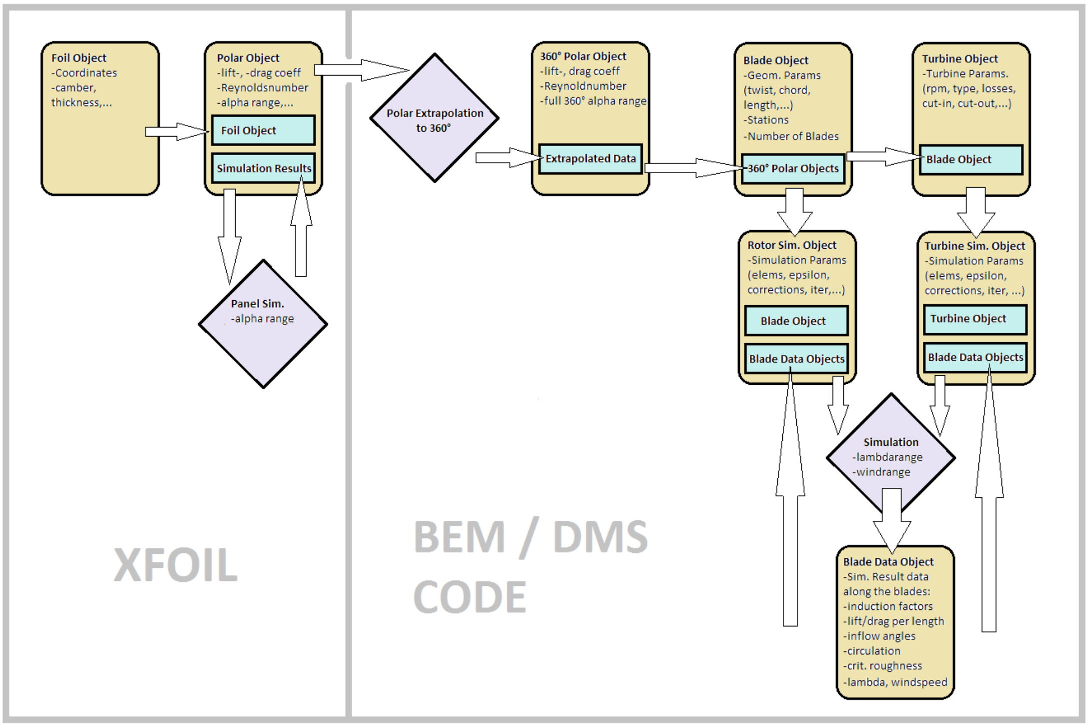


Figure 2.1: Data objects and data flow in QBlade

360° **polar Object**

A 360° polar object is created in the 360° polar extrapolation submodule. It is defined by a name and a parent polar (the original XFOIL polar that was extrapolated). The lift and drag coefficients over the whole 360° range of the AoA are stored as data. If the parent polar is deleted the extrapolated polar will be deleted as well to ensure consistency. If a 360° polar object is deleted, all blades, turbines and simulations utilizing it are deleted, too.

Blade object

A blade object is created in the blade design and optimization submodule. It stores the geometric blade data (chord, radial position, twist, offset etc.) of the specified blade sections and the associated foils and 360° polars. It is defined by a name. If a blade object is deleted, all associated simulations and turbines are deleted as well.

Turbine object

A turbine object is created in the turbine definition submodule. It stores all turbine parameters (pitch, stall, single/variable/2-step transmission etc.) and the associated rotor-blade. If a turbine object is deleted, all associated simulations are deleted as well.

Simulation object

A rotor or turbine simulation object is created when the respective simulation is defined. It memorizes the simulation parameters (max. iterations, elements, epsilon, ...) and corrections (tiploss etc.) and, during a simulation, stores the global results characterizing the whole rotor (λ, C_P, C_T etc.). If a simulation object is deleted, all associated blade data objects are deleted, tool.

Blade data object

Blade data objects are created automatically during a simulation. They store the data that is computed for the elements along the blade.What's more, one blade data object is created for every λ or windspeed increment of the simulation. The blade data objects require the most memory space.

2.3 Plotting results / Graph controls

QBlade inherits the Graph.cpp and QGraph.cpp classes from XFLR5 which are very convenient to display simulation results in various ways. Each Graph can be manipulated with the mouse. Using the mouse wheel zooms in and out of a graph. Zooming in and out while pressing X or Y on the keyboard only zooms the corresponding axis. By double clicking on a graph (Fig. 2.2) the user can specify variables for the X- and Y-Axis, such as plotting the C_P value over the tip speed ratio. By right clicking on a graph the type of graph can be specified

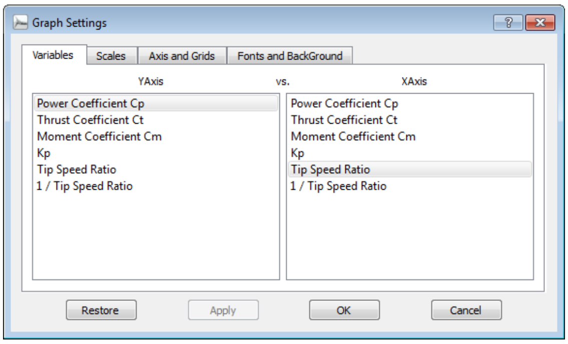


Figure 2.2: Defining Variables in Graph Settings

(see Data storage and visalization). Also in the graph context menu (Fig. 2.3), instead of displaying the whole set of blade variable curves, the currently selected blade curve can be isolated and copared to the curve of other simulations (if simulations with the same tip speed ratio or windspeed exist). Each conducted

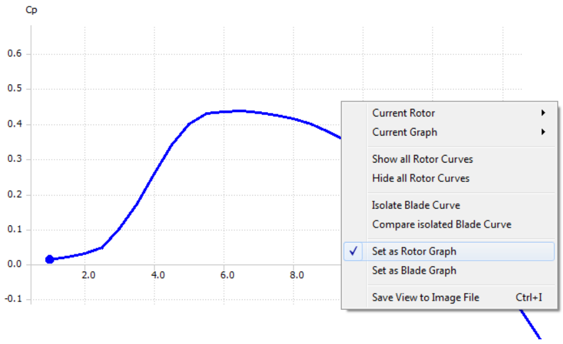
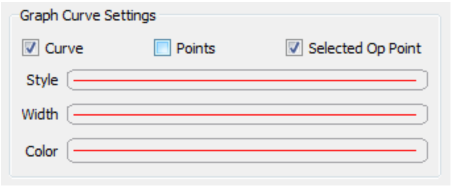


Figure 2.3: Graph and Graph Context Menu

simulation is displayed as a single curve in the rotor graph (the blade graphs show always a set of curves). Each curve can be manipulated individually (Fig. 2.4). The color, thickness and line style of a curve can be changed, computed points can be displayed and a curve can also be hidden from view.


Graph Curve Settings

☑ Curve □ Points ☑ Selected Op Point

Figure 2.4: Curve Settings

3 TUTORIAL: How to create simulations in QBlade

This section will give a short introduction on how to design and simulate a rotor or turbine with QBlade. Only the very basic functionalities are mentioned here and this is merely an overview in which succession the modules and submodules are to be run through to create a blade and simulate it. All the functions not mentioned in this part are more or less self-explanatory and the user may experiment freely with them.

Figure 3.1 shows the start screen of QBlade. From the main toolbar or the `File` menu, the user can navigate through QBlade's functionalities. The main toolbar contains the following buttons from left to right: The black ones provide simple file loading and saving functions. The red buttons belong to XFOIL and XFLR functionalities. The 360° polar extrapolation module can be opened by clicking on the 360 button. HAWT and VAWT related submenus are displayed by selecting the blue and green buttons respectively. The 360° polar extrapolation as well as the HAWT and VAWT menu points distinguish QBlade from XFLR. More information about XFOIL and XFLR5 in general can be found in [3] and [2].

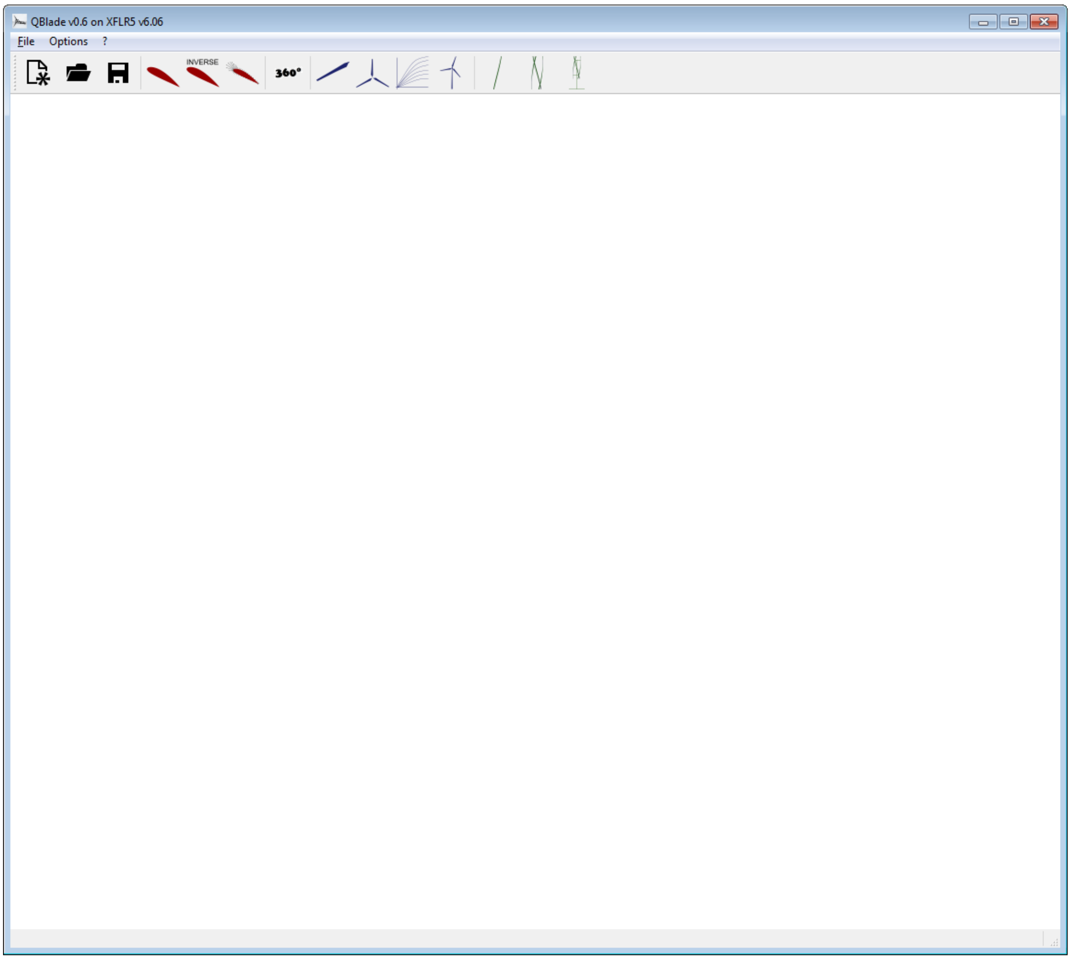
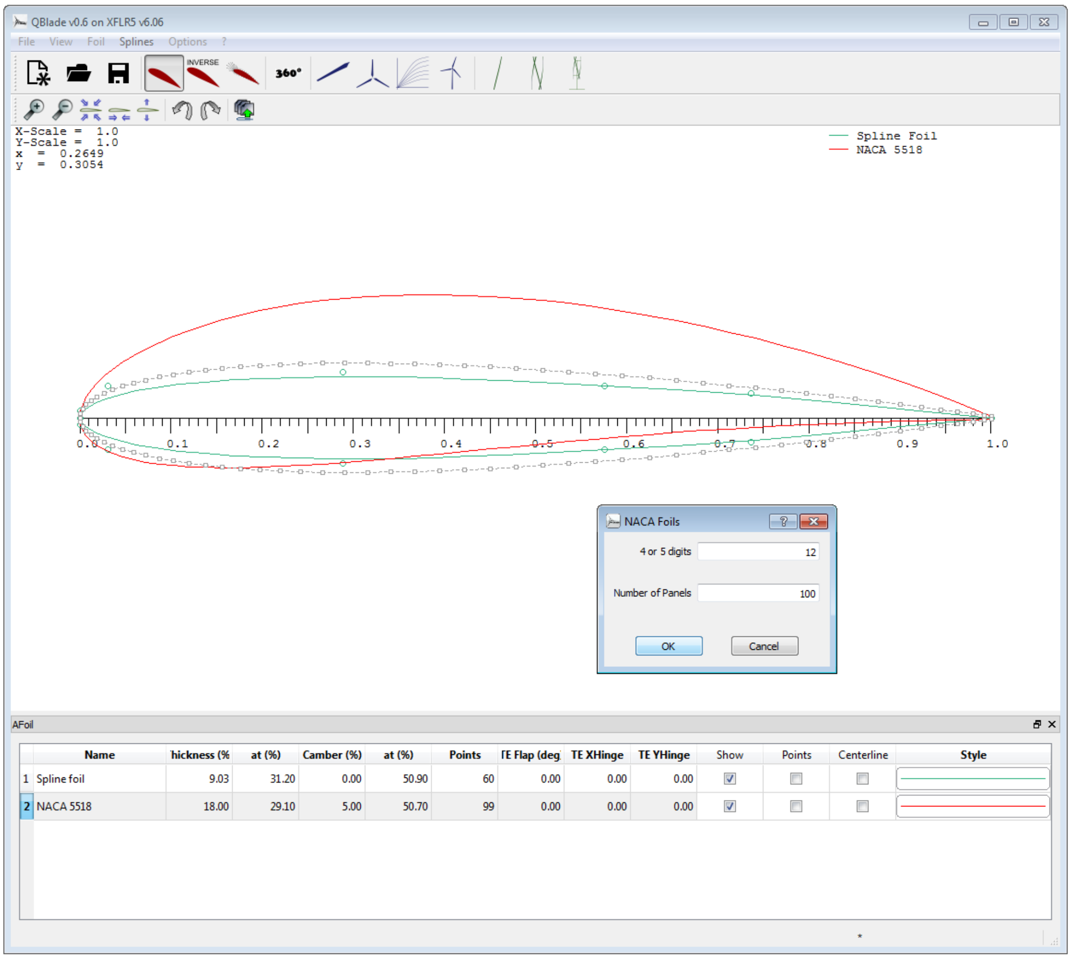


Figure 3.1: QBlade startup screen

The tutorial starts in the *Airfoil Design* module. In the run-up to creating a rotor, all its airfoils and the corresponding polar data need to be defined. Airfoils can be created using splines, a NACA airfoil generator or via an import function in XFLR5. In this case, the NACA 5518 is loaded.



```
X-Scale =   1.0
Y-Scale =   1.0
x  =  0.2649
y  =  0.3054
```

4 or 5 digits: 12

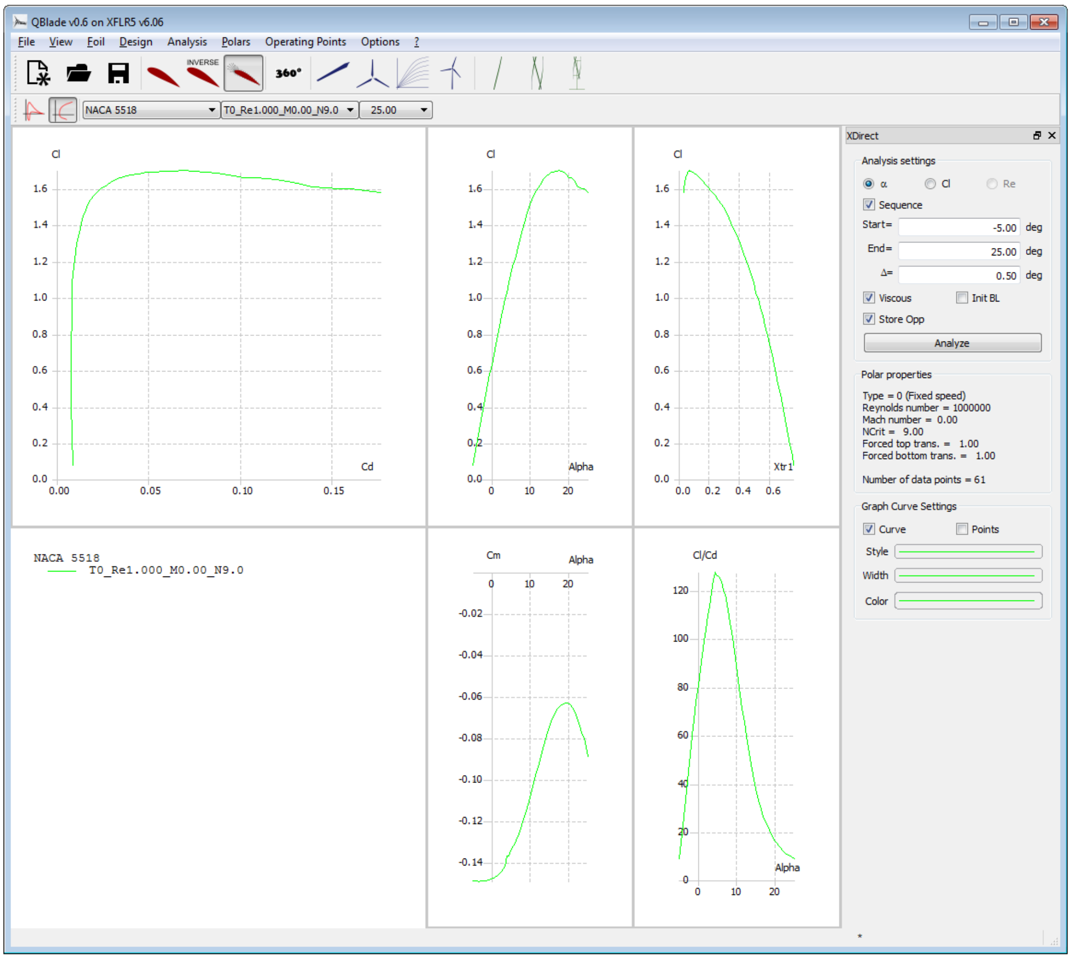
Number of Panels: 100

	Name	hickness (%	at (%)	Camber (%)	at (%)	Points	TE Flap (deg	TE XHinge	TE YHinge	Show	Points	Centerline	Style
1	Spline foil	9.03	31.20	0.00	50.90	60	0.00	0.00	0.00	☑	□	□	🟩
2	NACA 5518	18.00	29.10	5.00	50.70	99	0.00	0.00	0.00	☑	□	□	🟥

Figure 3.2: Airfoil Design module

1. press the `Airfoil Design` button (first red button)
2. click `Foil > Naca Foils`
3. enter 5518 in the `4 or 5 digits` line edit
4. press `OK`

In the *XFOIL Direct Analysis* module, the flow around the airfoils is simulated to create a polar. An analysis can be defined under the menu point *Polars*, then the simulation can be started. The analysis will only converge for a limited range of AoA values, typically from about -5° AoA to +25° AoA. It is also possible to import polar data in this module, for instance to utilize experimental data. Please note that, whenever polar data is imported, an arbitrary airfoil with exactly the same name as the imported polar data needs to be created to connect the polar data with this airfoil.



Analysis settings

● α ○ Cl ○ Re

☑ Sequence

Start= -5.00 deg

End= 25.00 deg

Δ= 0.50 deg

☑ Viscous □ Init BL

☑ Store Opp

Analyze

Polar properties

Type = 0 (Fixed speed)
Reynolds number = 1000000
Mach number = 0.00
NCrit = 9.00
Forced top trans. = 1.00
Forced bottom trans. = 1.00

Number of data points = 61

Graph Curve Settings

☑ Curve □ Points

Style

Width

Color

Figure 3.3: XFOIL Direct Analysis module

1. press the `XFOIL Direct Design` button (third red button)
2. click `Analysis > Define an Analysis`
3. enter 1000000 in the `Reynolds =` line edit and press `OK`
4. check the `Sequence` checkbox in the `Analysis Settings` on the right
5. enter an AoA range from -5 to +25° and an increment of 1° and press `Analyze`

To simulate a wind turbine, the AoA range of the polar needs to be extrapolated to 360°, this is done in the *polar extrapolation* module. Only extrapolated polar data can be used to simulate a turbine or rotor. The sliders on the left side of Fig. 3.4 can be used to better match the extrapolation with the previously computed polar data. Below the sliders, a value for $C_{D,90}$ can be specified. When the extrapolated polar is saved, it is stored in the runtime database. Additionally, it is possible to load a cylindrical foil and its 360° polar. This foil has zero lift and a drag value that can be specified by the user. Such cylindrical foils are used in the root region of a HAWT blade for structural reasons.

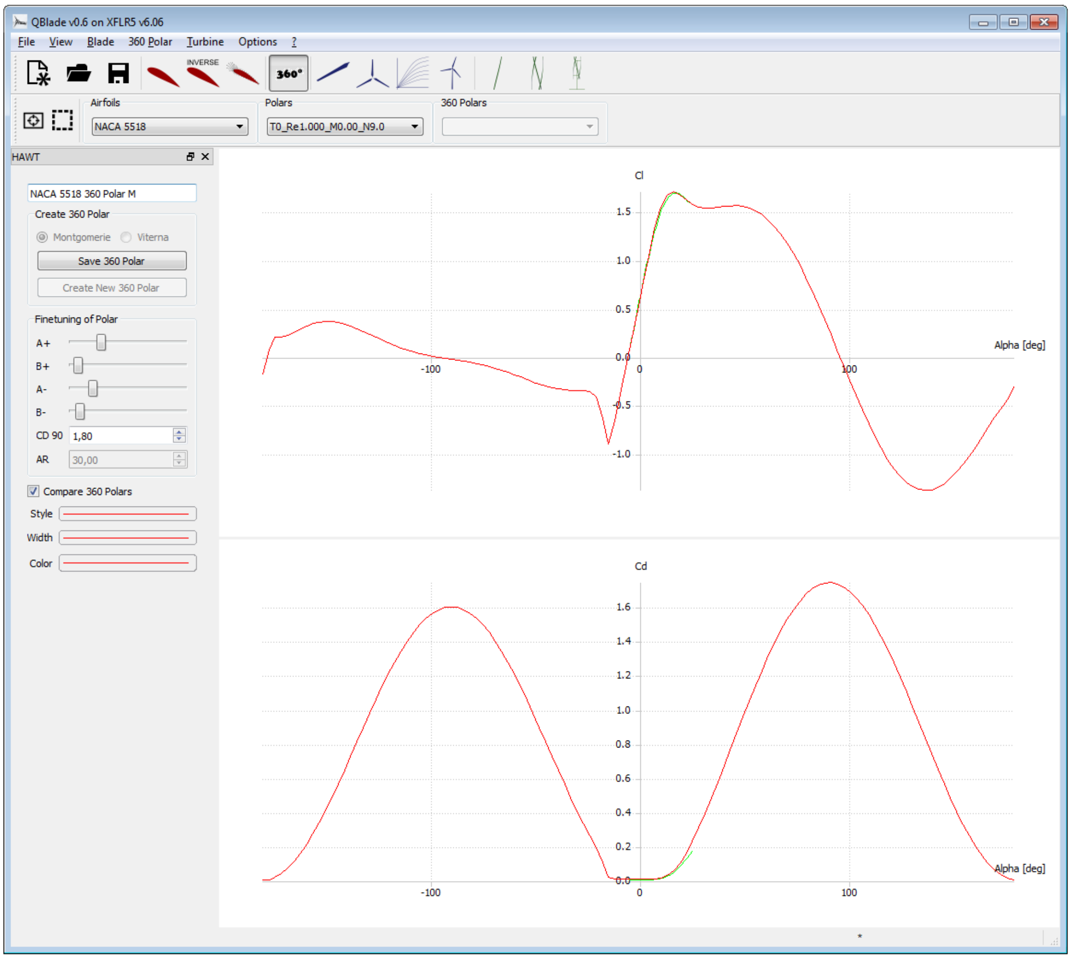
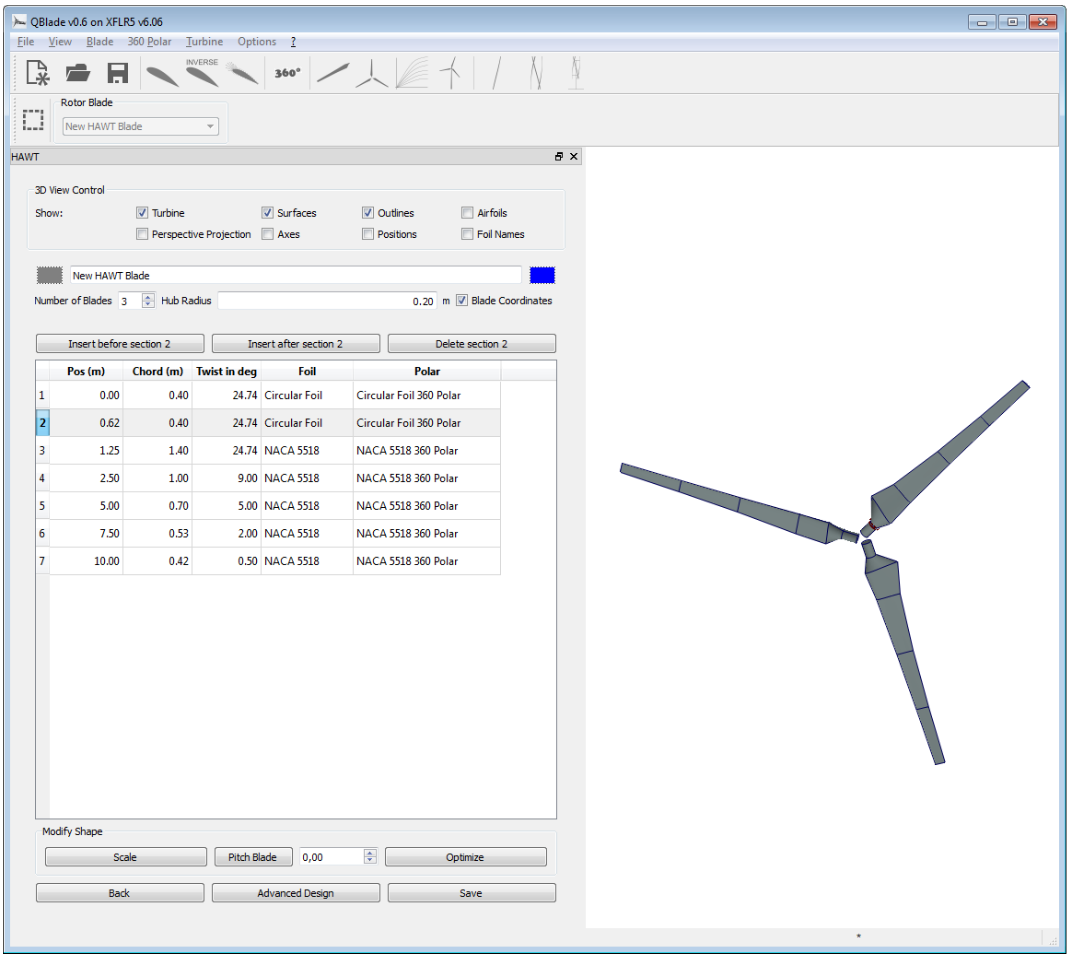

Airfoils: NACA 5518
Polars: T0_Re1.000_M0.00_N9.0
NACA 5518 360 Polar M
CD 90: 1,80
AR: 30,00

Figure 3.4: 360° polar extrapolation module

1. press the `Polar Extrpolation to 360` button 360° (360 button)
2. select the `Montgomery` radio button and press `Create New 360 Polar`
3. configure the initial polar via the C_{D90} value, the A± and B± sliders and press `Save 360 Polar`
4. click `360 Polar > Generate a Circular Foil`

If one or more 360° polars have been created, a blade can be designed in the *HAWT and VAWT blade design and optimization* submodules. The `Optimize` and `Scale` buttons open the corresponding dialog windows. Stations can be added or removed, one airfoil and one 360 polar has to be specified at every section. Additionally, every blade has a "Number of Blades" that predefines of how many blades the future rotor consists. When a blade is fully defined, it can be saved. Instead of creating a new blade, it is possible to edit stored blades. Whenever this is done, only a copy of the previous blade is edited.



Show: ☑ Turbine ☑ Surfaces ☑ Outlines □ Airfoils
□ Perspective Projection □ Axes □ Positions □ Foil Names

New HAWT Blade

Number of Blades 3 Hub Radius 0.20 m ☑ Blade Coordinates

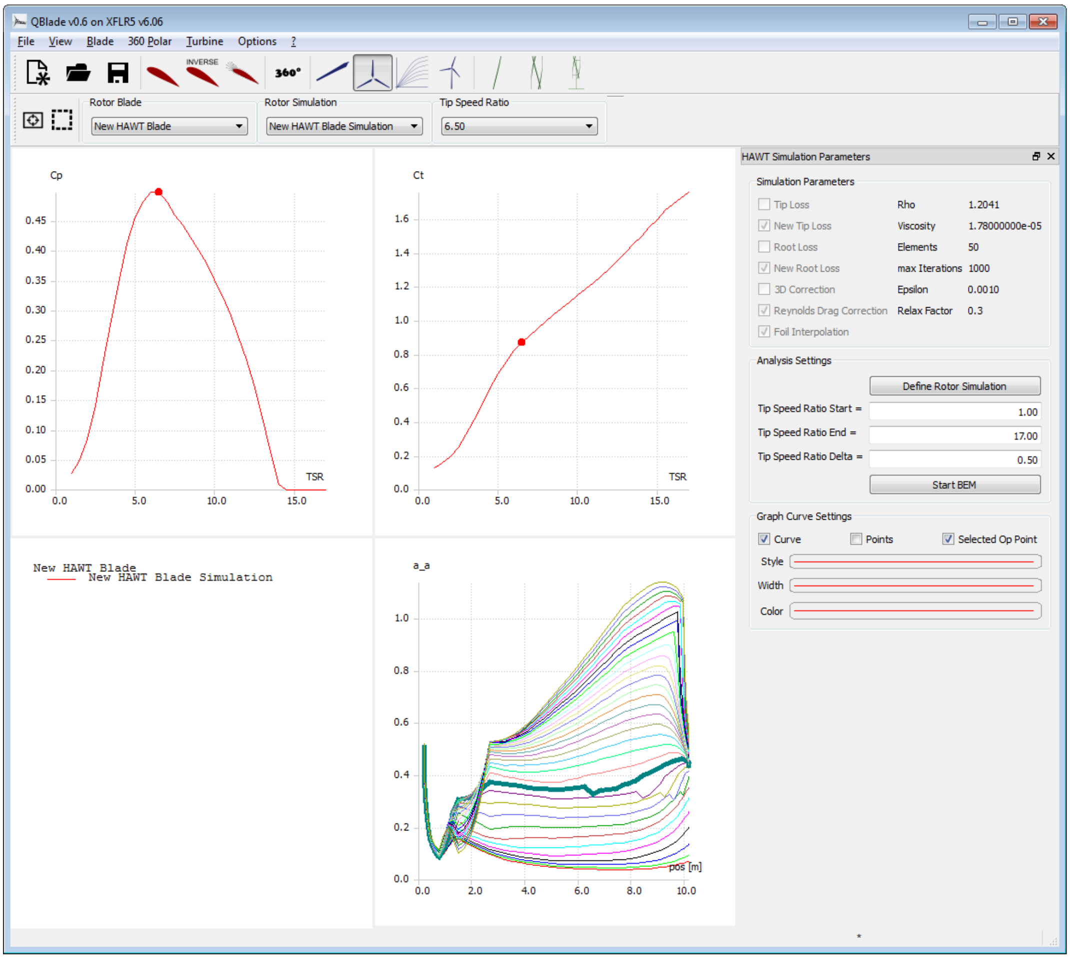
	Pos (m)	Chord (m)	Twist in deg	Foil	Polar
1	0.00	0.40	24.74	Circular Foil	Circular Foil 360 Polar
2	0.62	0.40	24.74	Circular Foil	Circular Foil 360 Polar
3	1.25	1.40	24.74	NACA 5518	NACA 5518 360 Polar
4	2.50	1.00	9.00	NACA 5518	NACA 5518 360 Polar
5	5.00	0.70	5.00	NACA 5518	NACA 5518 360 Polar
6	7.50	0.53	2.00	NACA 5518	NACA 5518 360 Polar
7	10.00	0.42	0.50	NACA 5518	NACA 5518 360 Polar

Pitch Blade 0,00

Figure 3.5: HAWT blade design and optimization submodule

1. press the `HAWT Rotorblade Design` button (first blue button)
2. press `New Blade`
3. enter the blade data via the tabular (see Fig. 3.5) or configure your own blade using the `Scale`, `Optimize` and `Advanced Design` options
4. press `Save`

All rotors in the database can be simulated in the *HAWT rotor simulation* submodule. Whenever a rotor simulation is defined, all simulation parameters need to be specified. Then, the dimensionless simulation is conducted over the desired range of tipspeed ratios. The three graphs show the simulation results. By double-clicking on a graph, the user may change the plotted variables. By right-clicking on a graph, the graph type to display local or global variables can be set: local variables are displayed in the same `blade graph` for all computed lambda values. The curve belonging to the currently selected lambda value (in the right drop-down menu) is highlighted. In the graph's context menu, the user may isolate the highlighted curve and then compare it to the local curves of other rotor simulations at the same tipspeed ratio. If multiple rotor simulations are stored in the database, the curves of global variables are always shown simultaneously in a `rotor graph`.



Rotor Blade: New HAWT Blade
Rotor Simulation: New HAWT Blade Simulation
Tip Speed Ratio: 6.50

Simulation Parameters

□ Tip Loss	Rho	1.2041
☑ New Tip Loss	Viscosity	1.78000000e-05
□ Root Loss	Elements	50
☑ New Root Loss	max Iterations	1000
□ 3D Correction	Epsilon	0.0010
☑ Reynolds Drag Correction	Relax Factor	0.3
☑ Foil Interpolation		

Analysis Settings

Tip Speed Ratio Start = 1.00
Tip Speed Ratio End = 17.00
Tip Speed Ratio Delta = 0.50

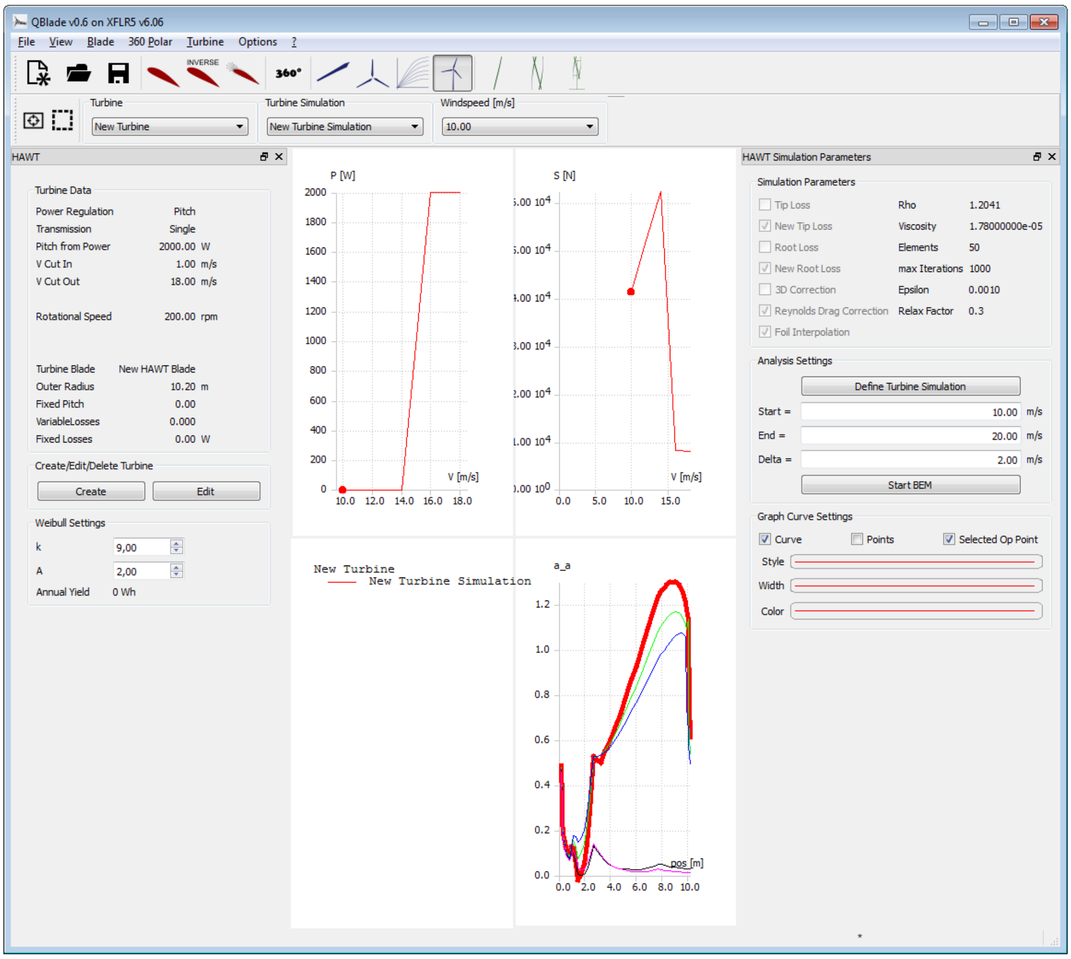
Graph Curve Settings

☑ Curve □ Points ☑ Selected Op Point
Style
Width
Color

Figure 3.6: HAWT rotor simulation submodule

1. press the `Rotor BEM Simulation` button (second blue button)
2. press `Define Rotor Simulation`
3. enter the simulation parameters and select corrections (see the groupbox in the top right corner of Fig. 3.6)
4. enter a tipspeed ratio range from 1 to 17 and an increment of 0.5
5. press `Start BEM`
6. explore the created simulation data by changing plot variables and graph types
7. create another rotor simulation for the same rotor but with different simulation parameters and compare the results, for example by isolating and comparing local curves
8. create another rotor, simulate it and compare the results
9. press the `Multi Parameter BEM Simulation` button (third blue button) to explore the `HAWT multi parameter simulation` submodule for parameter studies

In the *HAWT turbine definition and simulation* submodule, a turbine can only be defined if a blade is already stored in the database. The user specifies all turbine parameters and saves the turbine. Afterwards, a turbine simulation over a range of windspeeds can be conducted. When the user sets the k and A values for the Weibull distribution, the annual yield is computed automatically.



Turbine: New Turbine

Turbine Simulation: New Turbine Simulation

Windspeed [m/s]: 10.00

HAWT

Turbine Data

Power Regulation	Pitch
Transmission	Single
Pitch from Power	2000.00 W
V Cut In	1.00 m/s
V Cut Out	18.00 m/s
Rotational Speed	200.00 rpm
Turbine Blade	New HAWT Blade
Outer Radius	10.20 m
Fixed Pitch	0.00
VariableLosses	0.000
Fixed Losses	0.00 W

Create/Edit/Delete Turbine: Create Edit

Weibull Settings

k 9,00

A 2,00

Annual Yield 0 Wh

HAWT Simulation Parameters

Simulation Parameters

□ Tip Loss	Rho	1.2041
☑ New Tip Loss	Viscosity	1.78000000e-05
□ Root Loss	Elements	50
☑ New Root Loss	max Iterations	1000
□ 3D Correction	Epsilon	0.0010
☑ Reynolds Drag Correction	Relax Factor	0.3
☑ Foil Interpolation		

Analysis Settings

Define Turbine Simulation

Start = 10.00 m/s

End = 20.00 m/s

Delta = 2.00 m/s

Start BEM

Graph Curve Settings

☑ Curve □ Points ☑ Selected Op Point

Style

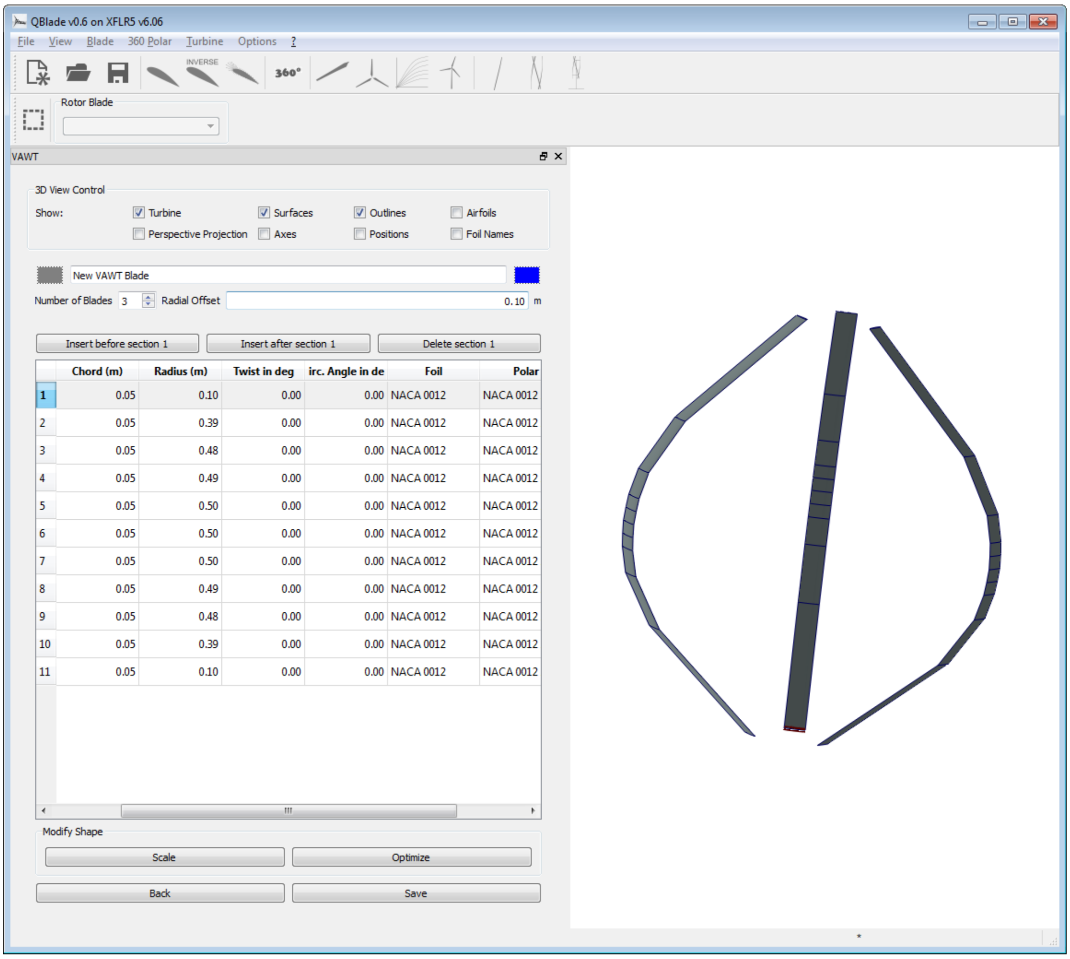
Width

Color

Figure 3.7: HAWT turbine definition and simulation submodule

1. press the `Turbine BEM Simulation` button (fourth blue button)
2. press `Create` to create a turbine
3. enter the turbine data (see the groupbox on the left side of Fig. 3.7)
4. press `Save`
5. press `Define Turbine Simulation`
6. enter the simulation parameters and select corrections (see the groupbox in the top right corner of Fig. 3.7)
7. enter a windspeed range from 1 to 18$^{\mathrm{m}}/_{\mathrm{s}}$ and an increment of 0.5$^{\mathrm{m}}/_{\mathrm{s}}$
8. press `Start BEM`
9. explore the created simulation data by changing plot variables and graph types
10. change the k and A parameters of the Weibull distribution to investigate the annual yield
11. create another turbine simulation for the same turbine but with different simulation parameters and compare the results, for example by isolating and comparing local curves
12. create another turbine, simulate it and compare the results

With the created foils and 360 polars, VAWT's can be created and simulated using the same procedural method as for HAWT's.



3D View Control

Show: ☑ Turbine ☑ Surfaces ☑ Outlines ☐ Airfoils ☐ Perspective Projection ☐ Axes ☐ Positions ☐ Foil Names

New VAWT Blade

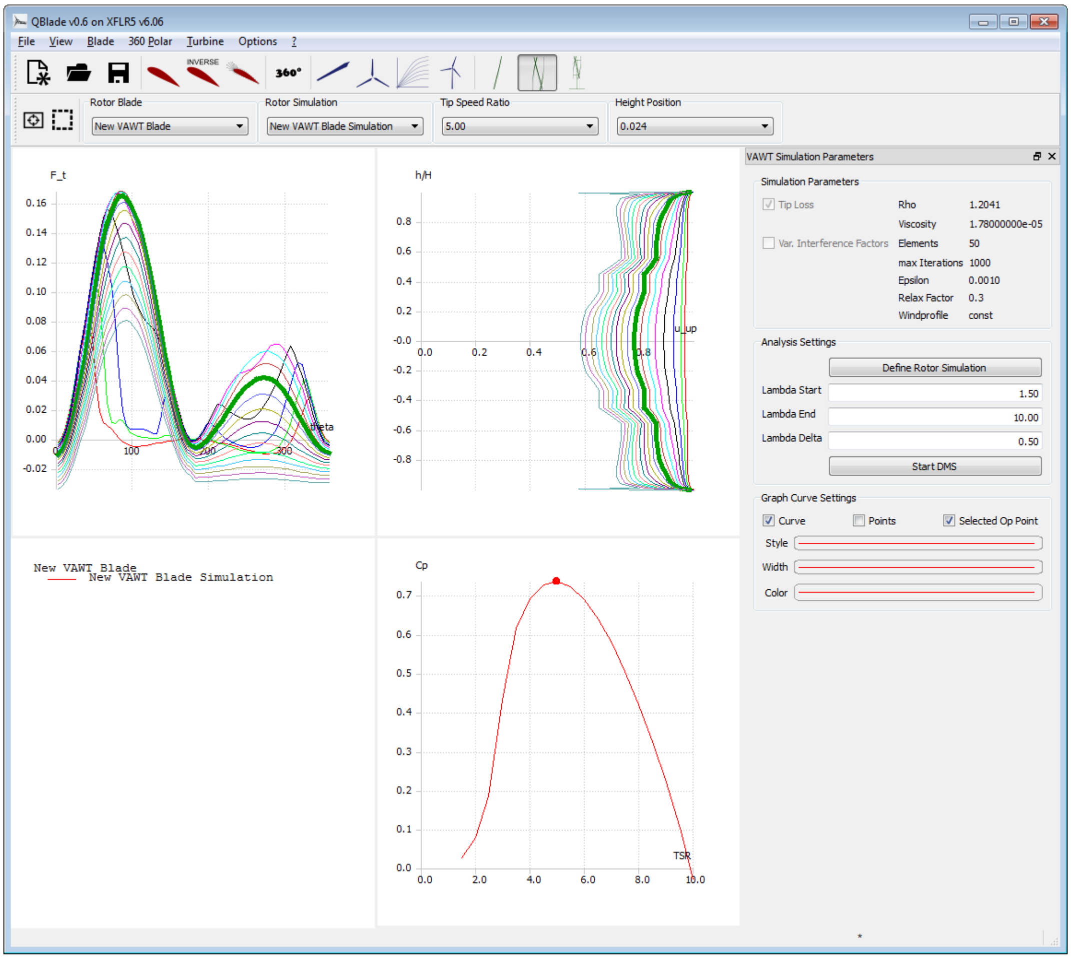
Number of Blades 3 Radial Offset 0.10 m

	Chord (m)	Radius (m)	Twist in deg	irc. Angle in de	Foil	Polar
1	0.05	0.10	0.00	0.00	NACA 0012	NACA 0012
2	0.05	0.39	0.00	0.00	NACA 0012	NACA 0012
3	0.05	0.48	0.00	0.00	NACA 0012	NACA 0012
4	0.05	0.49	0.00	0.00	NACA 0012	NACA 0012
5	0.05	0.50	0.00	0.00	NACA 0012	NACA 0012
6	0.05	0.50	0.00	0.00	NACA 0012	NACA 0012
7	0.05	0.50	0.00	0.00	NACA 0012	NACA 0012
8	0.05	0.49	0.00	0.00	NACA 0012	NACA 0012
9	0.05	0.48	0.00	0.00	NACA 0012	NACA 0012
10	0.05	0.39	0.00	0.00	NACA 0012	NACA 0012
11	0.05	0.10	0.00	0.00	NACA 0012	NACA 0012

Figure 3.8: VAWT blade design and optimization submodule

1. press the `VAWT Rotorblade Design` button (first green button)
2. press `New Blade`
3. configure the Darrieus rotor using the `Scale` and `Optimize` options (see Fig. 3.8) or configure your own blade
4. press `Save`

In a DMS simulation, an additional graph type is available, the `azimuthal graph`. As the flight path of a VAWT blade is a circle around its rotational axis, the relative speed at the blade is not only dependent on the blade element height position but also on the blade position on this flight path.



Rotor Blade: New VAWT Blade

Rotor Simulation: New VAWT Blade Simulation

Tip Speed Ratio: 5.00

Height Position: 0.024

New VAWT Blade
New VAWT Blade Simulation

VAWT Simulation Parameters

Simulation Parameters

☑ Tip Loss

☐ Var. Interference Factors

Rho	1.2041
Viscosity	1.78000000e-05
Elements	50
max Iterations	1000
Epsilon	0.0010
Relax Factor	0.3
Windprofile	const

Analysis Settings

Lambda Start 1.50

Lambda End 10.00

Lambda Delta 0.50

Graph Curve Settings

☑ Curve ☐ Points ☑ Selected Op Point

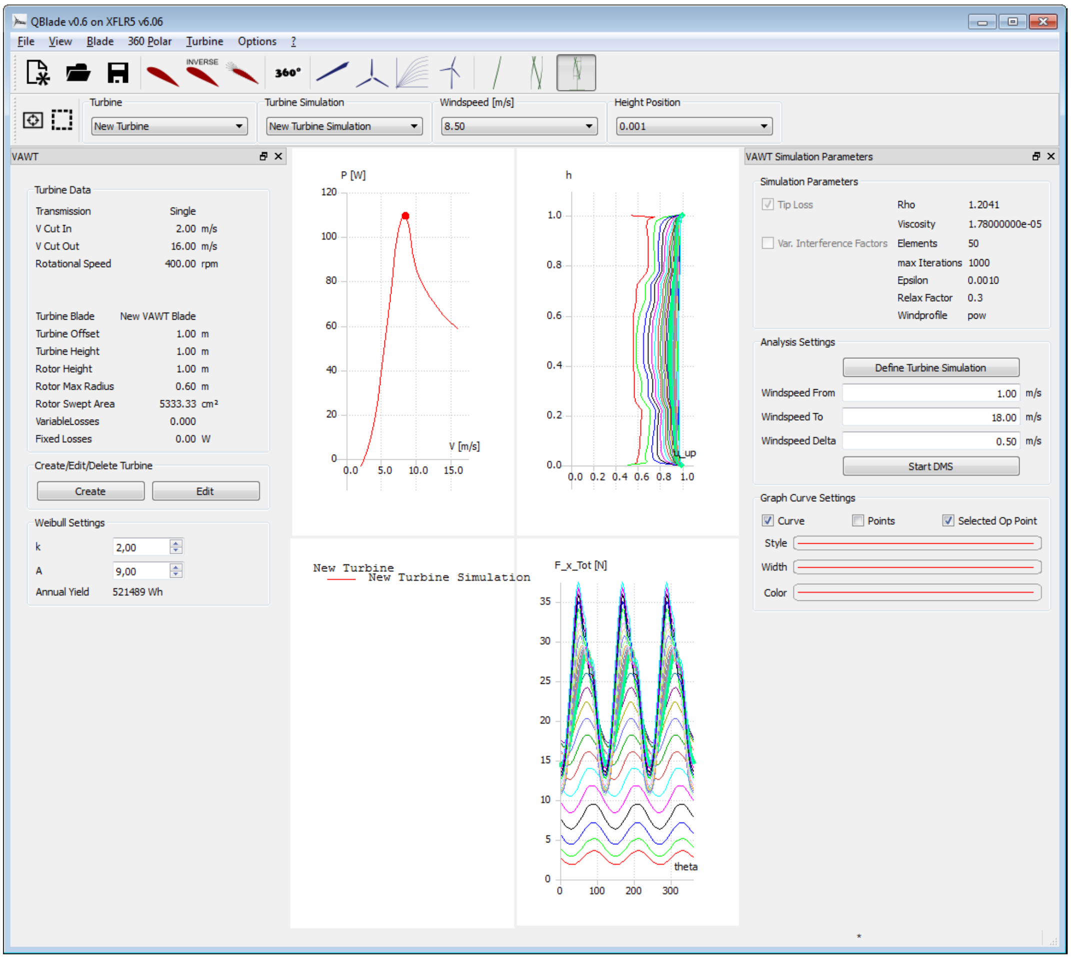
Style

Width

Color

Figure 3.9: VAWT rotor simulation submodule

1. press the `Rotor DMS Simulation` button (second green button)
2. press `Define Rotor Simulation`
3. enter the simulation parameters and select corrections (see the groupbox in the top right corner of Fig. 3.9)
4. enter a tipspeed ratio range from 1 to 10 and an increment of 0.5
5. press `Start DMS`
6. explore the created simulation data by changing plot variables and graph types
7. change the tipspeed ratio and height position in the upper dropdown menus
8. create another rotor simulation for the same rotor but with different simulation parameters and compare the results, for example by isolating and comparing local curves
9. create another rotor, simulate it and compare the results



Turbine: New Turbine

Turbine Simulation: New Turbine Simulation

Windspeed [m/s]: 8.50

Height Position: 0.001

Turbine Data

Transmission	Single	
V Cut In	2.00	m/s
V Cut Out	16.00	m/s
Rotational Speed	400.00	rpm
Turbine Blade	New VAWT Blade	
Turbine Offset	1.00	m
Turbine Height	1.00	m
Rotor Height	1.00	m
Rotor Max Radius	0.60	m
Rotor Swept Area	5333.33	cm²
VariableLosses	0.000	
Fixed Losses	0.00	W

Create/Edit/Delete Turbine

Weibull Settings

k	2,00
A	9,00
Annual Yield	521489 Wh

Simulation Parameters

☑ Tip Loss

□ Var. Interference Factors

Rho	1.2041
Viscosity	1.78000000e-05
Elements	50
max Iterations	1000
Epsilon	0.0010
Relax Factor	0.3
Windprofile	pow

Analysis Settings

Windspeed From	1.00	m/s
Windspeed To	18.00	m/s
Windspeed Delta	0.50	m/s

Graph Curve Settings

☑ Curve □ Points ☑ Selected Op Point

Style

Width

Color

Figure 3.10: VAWT turbine definition and simulation submodule

1. press the `Turbine DMS Simulation` button (third green button)
2. press `Create` to create a turbine
3. enter the turbine data (see the groupbox on the left side of Fig. 3.10)
4. press `Save`
5. press `Define Turbine Simulation`
6. enter the simulation parameters and select corrections (see the groupbox in the top right corner of Fig. 3.10)
7. enter a windspeed range from 1 to 18m/s and an increment of 0.5m/s
8. press `Start DMS`
9. explore the created simulation data by changing plot variables and graph types
10. change the tipspeed ratio and height position in the upper dropdown menus
11. change the k and A parameters of the Weibull distribution to investigate the annual yield
12. create another turbine simulation for the same turbine but with different simulation parameters and compare the results, for example by isolating and comparing local curves
13. create another turbine, simulate it and compare the results

Via the `File` menu or the main toolbar, the user can save the whole project, including foils, polars, blades and simulations as a `.wpa` file.

1. click `File > Save Project As ...` or press the `Save` button 💾 (third black button)
2. select a fitting project name and file location
3. press `Save`

Later on, this project file can be reloaded via the `File` menu as well:

1. recently saved or opened project files are stored directly in the `File` menu and can be loaded by clicking on the desired file path
2. otherwise, click `File > Insert Project` or `File > Open` or press the `Open` button 📂 (second black button)
3. select the desired project file
4. press `Open`

A project can be closed via the `File` menu by clicking `File > Close the Project`. To create a new project, the user can either click `File > New Project` or press the `New Project` button 🗋 (first black button)

4 XFOIL and XFLR/QFLR

The software *XFOIL* is a program to analyze and compute the flow around subsonic isolated airfoils. It is a standard software for the calculation of profile polars. *XFOIL* has been validated against other numerical methods and against experimental data (for low angles of attack [20], for high angles of attack [19]). The source code is released under the *GNU General Public License*. The first version was written in 1986 by MARK DRELA. The goal was to combine a high-order panel method with the new fully-coupled viscous/inviscid interaction method developed by DRELA and GILES at the MIT [3]. Since then, the software has undergone numerous revisions and has developed from a simple command line tool to a cross-platform compatible C++/Qt4 tool (XFLR [2] /QFLR) with a sophisticated GUI. While the software was ported and some additional modules like calculations with the vortex-lattice-method were added, the algorithms for the foil analysis are exactly the same as those in the original *XFOIL* code. Therefore, the *XFOIL 6.94 User Guide* [3] is still valid for all the different versions and can be consulted for further details.

XFOIL's features , that are relevant for QBlade are:

- viscous or inviscid analysis of an airfoil, considering

  forced or free transition

  transitional separation bubbles

  limited trailing edge separation

  lift and drag predictions just beyond stall

- airfoil design and redesign by specification of a surface speed distribution
- airfoil redesign by interactive specification of new geometric parameters
- blending of Airfoils
- import of airfoil geometry

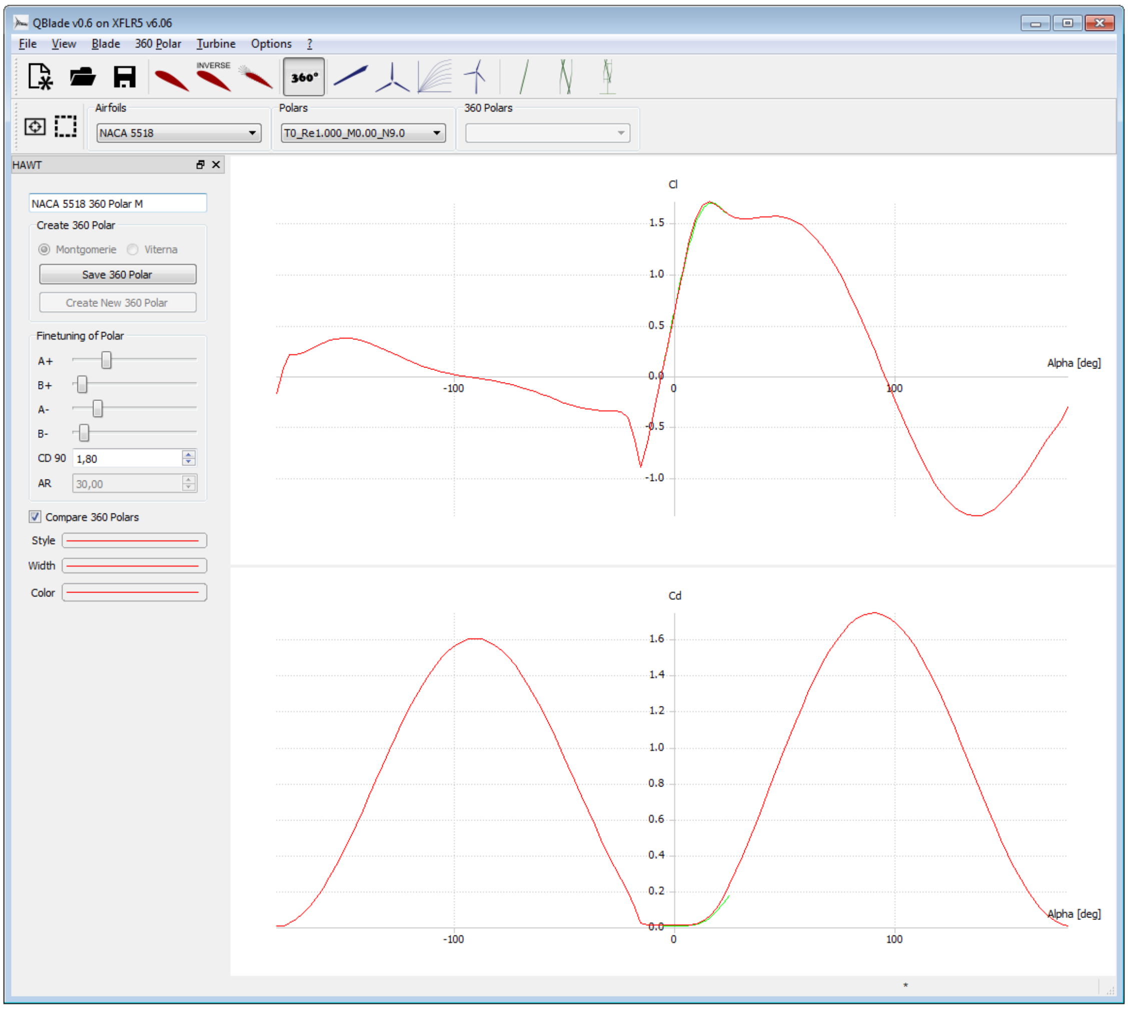
5 The QBlade 360° extrapolation module

5.0.1 Basics

During a revoultion, a wind turbine blade experiences much higher angles of attack than the wing of an airplane. For stall-regulated wind turbines, the flow phenomenon is even used as a means to limit the produced power. As stall is generally avoided in aeronautical engineering, the coefficients corresponding to high AoA are not of interest and therefore not available. Additionally, AoA in the post stall range may occur temporarily during the QBlade iteration procedures. Thus, polar data for all possible 360° AoA has to be available to ensure the continuation of the BEM and DMS algorithms.

The lift and drag coefficients consequently have to be extrapolated for the whole 360° range of the AoA. In general, experimental pre-stall data ($\alpha \approx \pm 15^\circ$) is available for common airfoils and *XFOIL* is suitable to generate such data as well. For post stall data on the other hand, some further considerations are needed. With increasing AoA, the frontal area facing the airflow increases, too. Stall occurs, the foil dramatically stops producing lift and the drag coefficient increases. Around 180°, the trailing edge of the streamlined airfoil faces the flow resulting in decreasing drag and higher lift again. Hence, the flow characteristics evolve from those of a thin, streamlined airfoil to those of a blunt body and back during a 180° revolution of the blade.

Polars, that are either imported or a result of an XFOIL analysis, can be extrapolated to the full 360° AoA range in the 360° polar extrapolation submodule. A polar can be selected from the drop-down menu in the toolbar. When creating an extrapolation, the user can choose between two extrapolation algorithms: firstly, the sophisticated Montgomery extrapolation can be chosen. Secondly, the Viterna-Corrigan post stall model which is favored in industrial environments is disponible, too.



Airfoils: NACA 5518

Polars: T0_Re1.000_M0.00_N9.0

360 Polars:

NACA 5518 360 Polar M

● Montgomerie ○ Viterna

CD 90: 1,80

AR: 30,00

☑ Compare 360 Polars

Figure 5.1: The 360° polar extrapolation submodule

5.0.2 Montgomery extrapolation

For the Montgomery extrapolation procedure, basic lift and drag curves are needed. It is then assumed, that the flow around the airfoil can be treated as potential flow near 0° and 180° AoA. At other AoA's, the flow approximately behaves like for a stalled, thin plate. A blending function is used in the transition region between the potential flow straight line and the flat plate curve.

The user can take influence on the apexes of the resulting coefficient curves via different sliders. The two points C_{L1} and C_{L2}, from which the blending function f for the positive extrapolation is constructed, can be manipulated using the sliders `A+` (for $CL1$) and `B+` (for C_{L2}). The sliders `A-` and `B-` manipulate the corresponding points for the negative extrapolation. Additionally, a favored 2D drag coefficient $C_{D,90}$ at 90° AoA can be selected manually as well. Whenever a slider is moved or the value for $C_{D,90}$ is changed, the whole extrapolation is computed again. With the option "edit current polar", the currently selected 360° polar can be edited manually after it has been extrapolated.

The Montgomery extrapolation is carried out as described in *Methods for Root Effects, Tip Effects and Extending the Angle of Attack Range to* +-100°, *with Application to Aerodynamics for Blades on Wind Turbines and Propellers* [10].

5.0.3 Viterna-Corrigan post stall model

The Viterna approach to extending the polar data to all possible angles of attack consists of empirical equations. It is an idealized model which results in an approximately constant power output after stall, provided that the blade experiences high windspeeds. It is also based on the aspect ratio of the future wing, which can be modified by the user. As the extrapolation depends on the AR value, blades with different aspect ratios require separate 360° polars. Whenever the aspect ratio is changed, the whole extrapolation is computed again.

The Viterna extrapolation is carried out as described in *Fixed Pitch Rotor Performance of Large Horizontal Axis Wind Turbines* [11].

6 The QBlade HAWT module

6.1 Basics

6.1.1 The Blade Element Momentum Method

The classical Blade Element Momentum (BEM) theory couples the momentum theory or disk actuator theory, a mathematical model of an ideal actuator disc, with the blade element theory, which describes the local events taking place at the actual blade. The blade is discretized into a finite number of blade elements. Two sections bound an element that sweeps the rotor plane on a circular path. The blade cross sections are defined by their radial position, profile, chord, twist and length. With the momentum theory, the relative windspeed for every section can be computed. This allows the calculation of the angle of attack and the derivation of the lift and drag coefficients of the respective profile out of a table. With these coefficients and the area of an element the normal and tangential force components, thus the thrust and torque of an element are computed. The elements' contributions can then be added up to yield the final thrust and torque of the whole rotor. For different ratios of windspeed and angular speed, characteristic curves and quantities of the rotor can be computed.

6.1.2 Iteration procedure

The iteration variables of the BEM method are the axial and radial induction factors which are defined as

$$a = \left(\frac{4\sin^2\phi}{\sigma C_n} + 1\right)^{-1} \tag{6.1}$$

$$a' = \left(\frac{4\sin\phi\cos\phi}{\sigma C_t} - 1\right)^{-1} \tag{6.2}$$

with the inflow angle ϕ, the tangential and normal force coefficients C_t and C_n and the rotor solidity σ, which is the part of an annular element that is covered by blades:

$$\sigma = \frac{cB}{2\pi r} \tag{6.3}$$

In this equation, c is the chordline and B the number of blades.

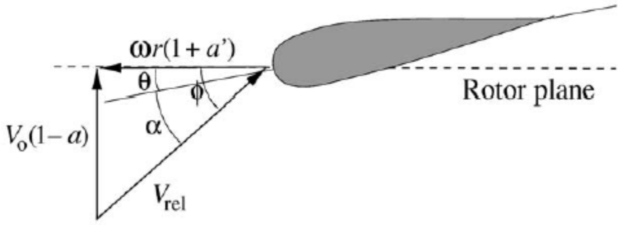


Figure 6.1: Velocities at the rotor plane [5]

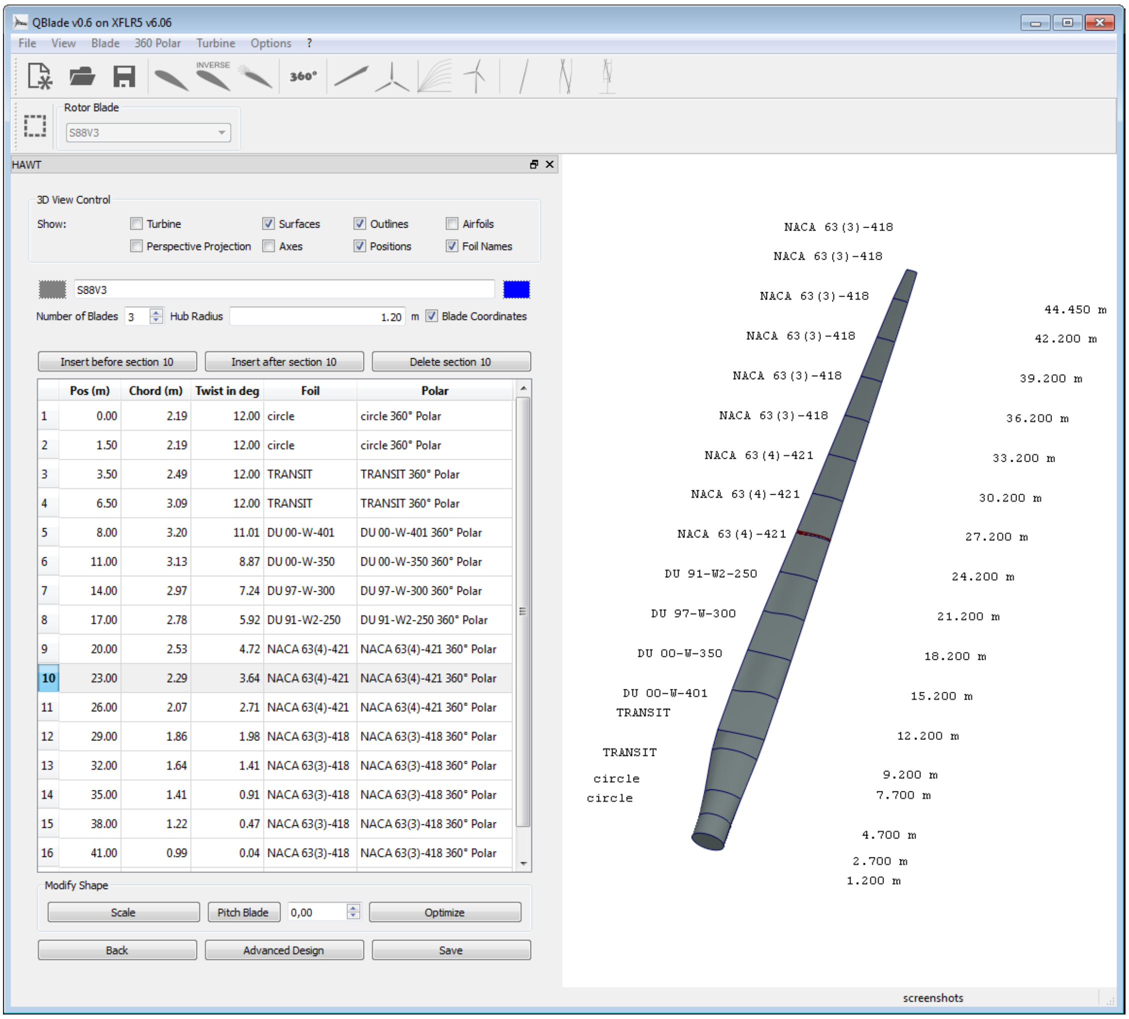
Figure 6.1 shows the velocities at the rotor plane. If the induction factors are known, it it possible to compute the inflow angle ϕ. ϕ gives the angle of attack α, the angle between the airfoil's chordline and the relative windspeed experienced by the rotating blade:

$$\alpha = \phi - \theta\ , \tag{6.4}$$

where $\theta = \theta_p + \beta$ is a combination of the twist angle β and the pitch angle θ_p. Now the lift and drag coefficients of the airfoil can be obtained from a table and the lift and drag caused by the airfoil can be calculated. From these forces, new induction factors can be computed and compared to the initial induction factors. If $\texttt{max}(\Delta a, \Delta a')$ is below the convergence criterion ϵ, the iteration has converged and the next annular element can be computed.

6.2 The blade design and optimization submodule

In the blade design and optimization sub-module the user can create a rotorblade. A blade consists of an arbitrary number of sections, see Fig. 6.3. Every section is defined by *position*, *chord*, *twist*, *airfoil* and the associated 360° *polar*. A new blade can only be created when there is at least one 360° polar object present in the runtime database and it can only be saved when a foil and a polar is selected at every station. The *number of blades* of the future rotor has to be



Rotor Blade: S88V3

3D View Control

Show: □ Turbine ☑ Surfaces ☑ Outlines □ Airfoils
□ Perspective Projection □ Axes ☑ Positions ☑ Foil Names

S88V3

Number of Blades 3 Hub Radius 1.20 m ☑ Blade Coordinates

	Pos (m)	Chord (m)	Twist in deg	Foil	Polar
1	0.00	2.19	12.00	circle	circle 360° Polar
2	1.50	2.19	12.00	circle	circle 360° Polar
3	3.50	2.49	12.00	TRANSIT	TRANSIT 360° Polar
4	6.50	3.09	12.00	TRANSIT	TRANSIT 360° Polar
5	8.00	3.20	11.01	DU 00-W-401	DU 00-W-401 360° Polar
6	11.00	3.13	8.87	DU 00-W-350	DU 00-W-350 360° Polar
7	14.00	2.97	7.24	DU 97-W-300	DU 97-W-300 360° Polar
8	17.00	2.78	5.92	DU 91-W2-250	DU 91-W2-250 360° Polar
9	20.00	2.53	4.72	NACA 63(4)-421	NACA 63(4)-421 360° Polar
10	23.00	2.29	3.64	NACA 63(4)-421	NACA 63(4)-421 360° Polar
11	26.00	2.07	2.71	NACA 63(4)-421	NACA 63(4)-421 360° Polar
12	29.00	1.86	1.98	NACA 63(3)-418	NACA 63(3)-418 360° Polar
13	32.00	1.64	1.41	NACA 63(3)-418	NACA 63(3)-418 360° Polar
14	35.00	1.41	0.91	NACA 63(3)-418	NACA 63(3)-418 360° Polar
15	38.00	1.22	0.47	NACA 63(3)-418	NACA 63(3)-418 360° Polar
16	41.00	0.99	0.04	NACA 63(3)-418	NACA 63(3)-418 360° Polar

Pitch Blade 0,00

Figure 6.2: Blade design submodule

defined for every rotor blade. The *hub radius* must be specified, which is the position where the blade root is connected to the hub of the turbine. The radial positions of a blade can either be defined in *blade coordinates* where the positions are the distance from the blade root, or in *absolute coordinates* where the positions are the total distance from the turbine's hub center. The option *pitch blade* adds an offset to the rotorblades' twist at every section of the blade. When a blade design is finished the user can export the blade geometry either as a cloud of points or in the .stl file format. To export a blade geometry select in the top toolbar:
Blade –> Export Rotorblade Geometry

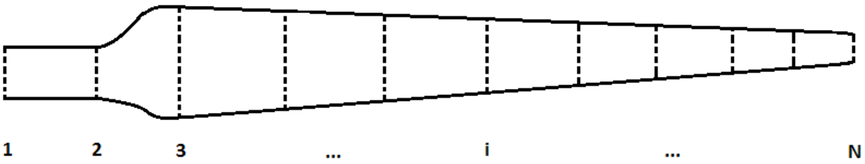
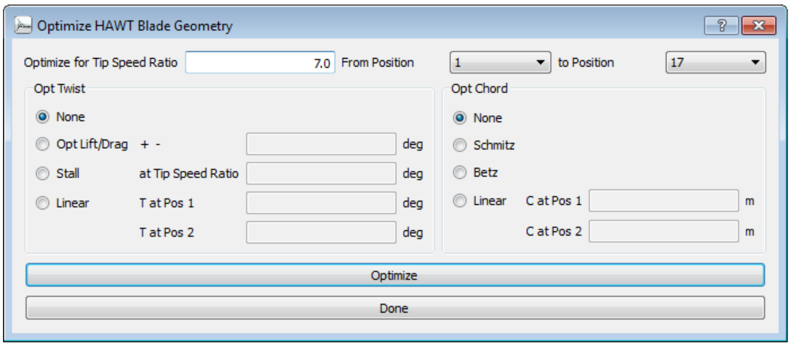


Figure 6.3: Sections along a blade

6.2.1 Blade optimization

While a blade is edited or created and every section is fully defined, the user can optimize the blade geometry in the blade optimization dialog, see Fig. 6.4.



Optimize HAWT Blade Geometry

Optimize for Tip Speed Ratio 7.0 From Position 1 to Position 17

Opt Twist

● None
○ Opt Lift/Drag + - deg
○ Stall at Tip Speed Ratio deg
○ Linear T at Pos 1 deg
T at Pos 2 deg

Opt Chord

● None
○ Schmitz
○ Betz
○ Linear C at Pos 1 m
C at Pos 2 m

Figure 6.4: Blade optimization dialog

The user has to choose a tipspeed ratio λ_0 to optimize for, and the sections (positions) that are to be optimized. From this tipspeed ratio, an assumed inflow angle is computed for every section:

$$\alpha_{loc} = \tan^{-1}\left(\frac{1}{\lambda_{0,loc}}\frac{2}{3}\right) \tag{6.5}$$

Optimize for lift/drag sets the twist, at the specified $\lambda_{0,loc}$ at which the blade section operates, to an AoA that yields the highest glide ratio. The option to decrease or increase this angle exists for the case that the AoA yielding the highest glide-ratio is close to the stall point. If the user optimizes for stall, the twist is set in such a way that all the stations, at the same time, experience stall at the selected λ_0 value. The third option allows to set a linear twist.

The chord distribution can be optimized according to BETZ [4, p.202]:

$$c(r) = \frac{16}{9}\frac{\pi R}{BC_L\lambda_0}\frac{1}{\sqrt{(\lambda_0\frac{r}{R})^2+\frac{4}{9}}}, \tag{6.6}$$

or SCHMITZ [4, p.202]:

$$c(r) = \frac{16\pi r}{BC_L} \sin^2\left(\frac{1}{3}\tan^{-1}\left(\frac{R}{\lambda_0 r}\right)\right) . \tag{6.7}$$

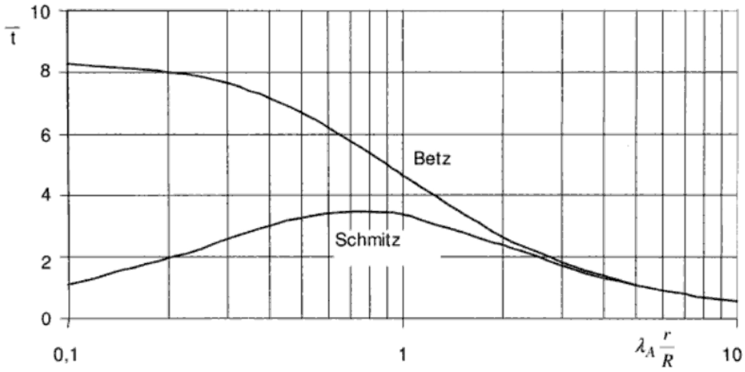
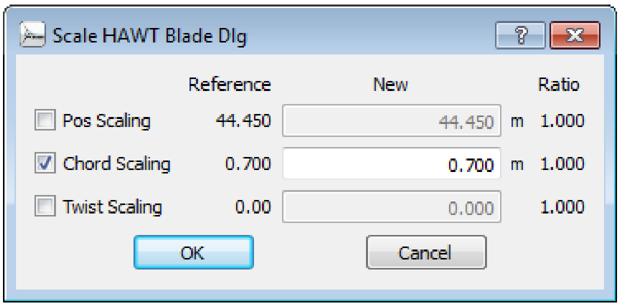


Figure 6.5: Comparison of dimensionless chord distribution after BETZ and Schmitz [4]

It is important to note that C_L is computed from the expected angle (λ_0) and the twist angle for every station individually. It may happen, especially for stall blades, that a low C_L value at λ_0 causes a very high value for the chord. This is not a failure of the optimization equations. One has to be careful with large chord values. The value for solidity, $\sigma = \frac{cB}{2\pi r}$ (the section of an annulus that is covered by blades), must be between zero and unity. For a section whose solidity is greater than unity the BEM algorithm does not converge. Attention has to be paid especially in the root region because both the BETZ and the SCHMITZ equation lead to large chord values here.

6.2.2 Blade scaling

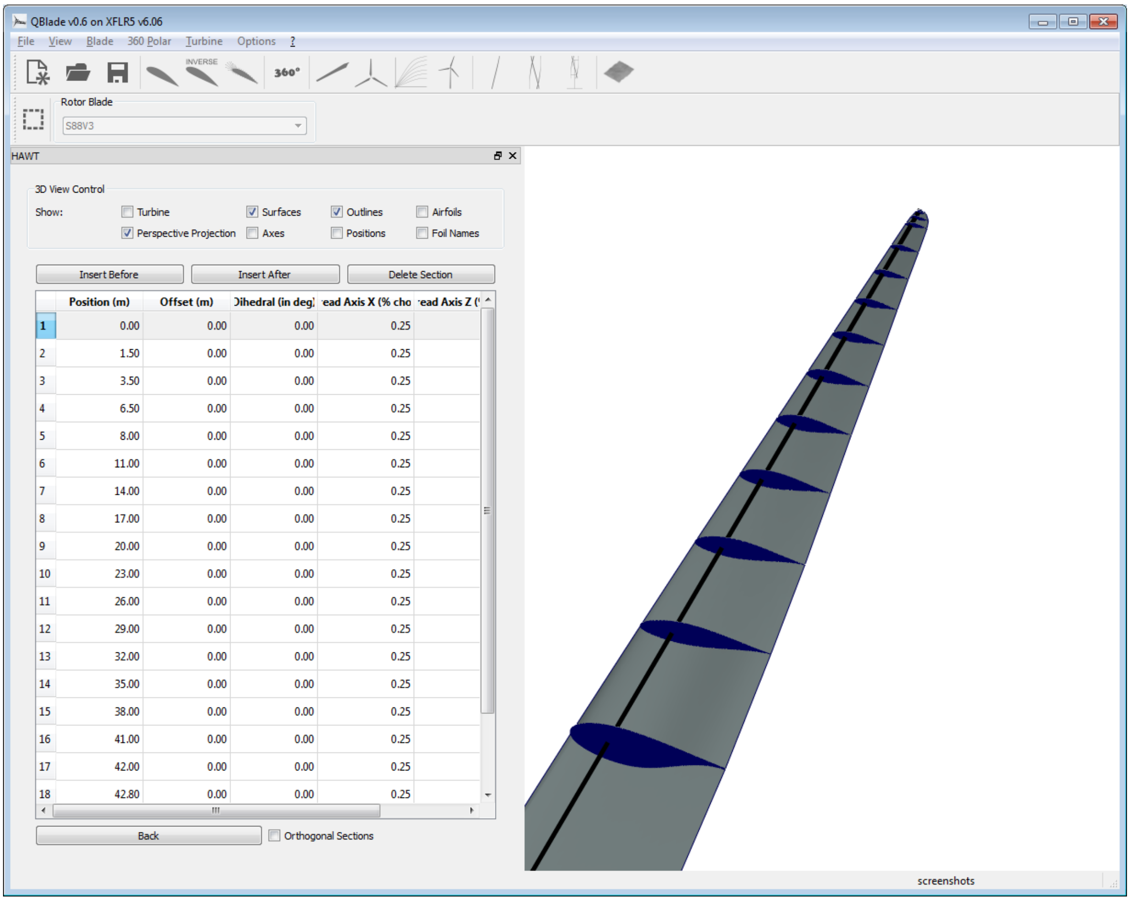

Scale HAWT Blade Dlg

	Reference	New		Ratio
□ Pos Scaling	44.450	44.450	m	1.000
☑ Chord Scaling	0.700	0.700	m	1.000
□ Twist Scaling	0.00	0.000		1.000

Figure 6.6: Scale blade dialog

The scale option allows to scale the blade's twist, chord and/or position. This can be done by simply giving a new value for the variable that is scaled. The scaling ratio is then computed automatically.

6.2.3 Advanced design

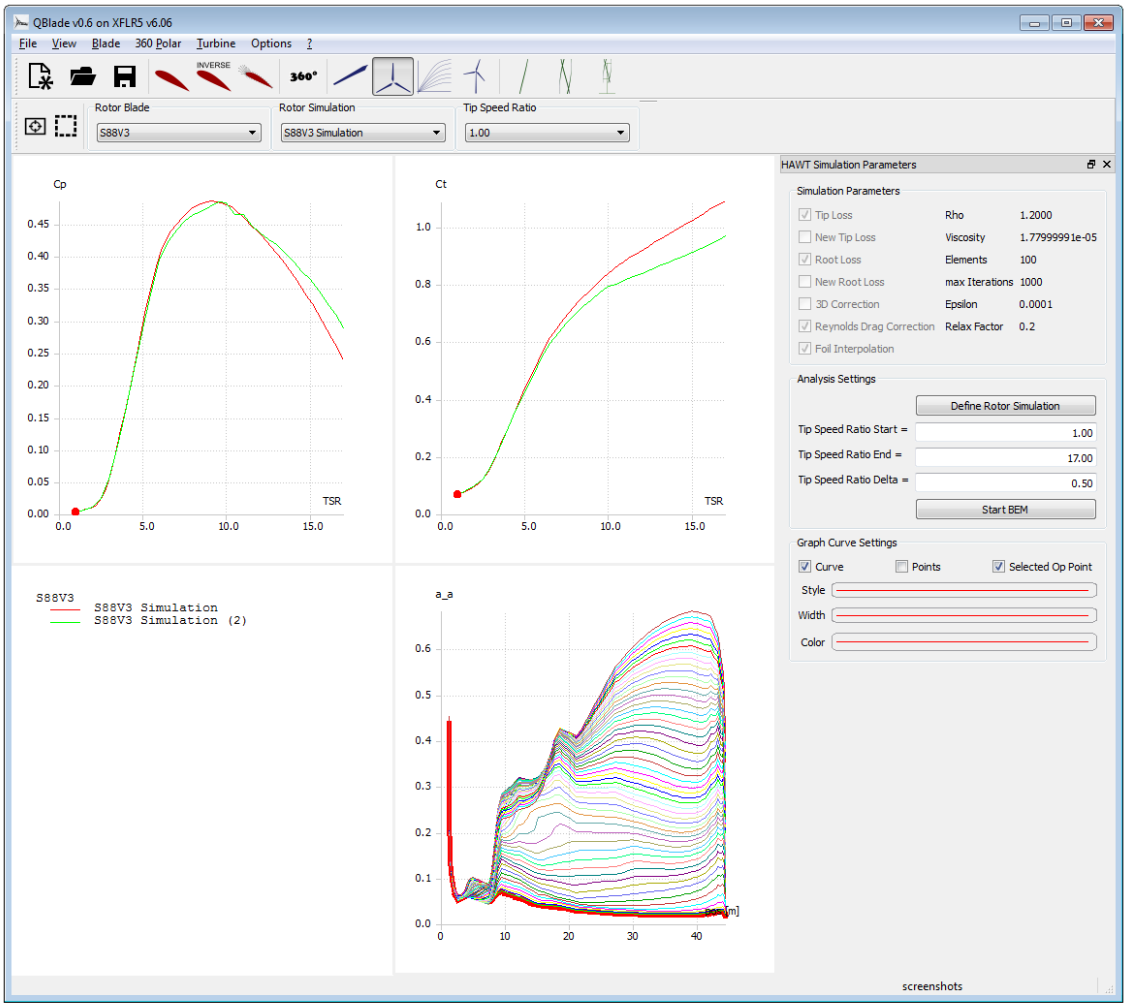


	Position (m)	Offset (m)	Dihedral (in deg)	ead Axis X (% cho	ead Axis Z ('
1	0.00	0.00	0.00	0.25	
2	1.50	0.00	0.00	0.25	
3	3.50	0.00	0.00	0.25	
4	6.50	0.00	0.00	0.25	
5	8.00	0.00	0.00	0.25	
6	11.00	0.00	0.00	0.25	
7	14.00	0.00	0.00	0.25	
8	17.00	0.00	0.00	0.25	
9	20.00	0.00	0.00	0.25	
10	23.00	0.00	0.00	0.25	
11	26.00	0.00	0.00	0.25	
12	29.00	0.00	0.00	0.25	
13	32.00	0.00	0.00	0.25	
14	35.00	0.00	0.00	0.25	
15	38.00	0.00	0.00	0.25	
16	41.00	0.00	0.00	0.25	
17	42.00	0.00	0.00	0.25	
18	42.80	0.00	0.00	0.25	

Figure 6.7: Advanced design window

The *advanced design* module enables the user to define the blade shape in more detail. *Offset* specifies an offset of the airfoil section in *x*-direction. Dihedral defines the angle between the blade chord and the *y*-axis. *Thread axis X* and *Z* specify the point on the selected airfoils surface around which it is twisted. However the changes applied on a blade in the *advanced design* module dont affect the blades performance in a BEM simulation. This is because the BEM doesnt account for the full 3D shape of a blade. However the geometry of the 3D blade can be exported as an *.stl* file. Future implementations planned for *QBlade* will make use of the full 3D description of the blade.

6.3 The rotor simulation submodule



Rotor Blade: S88V3
Rotor Simulation: S88V3 Simulation
Tip Speed Ratio: 1.00

HAWT Simulation Parameters

Simulation Parameters

☑ Tip Loss | Rho 1.2000
☐ New Tip Loss | Viscosity 1.77999991e-05
☑ Root Loss | Elements 100
☐ New Root Loss | max Iterations 1000
☐ 3D Correction | Epsilon 0.0001
☑ Reynolds Drag Correction | Relax Factor 0.2
☑ Foil Interpolation

Analysis Settings

Define Rotor Simulation
Tip Speed Ratio Start = 1.00
Tip Speed Ratio End = 17.00
Tip Speed Ratio Delta = 0.50
Start BEM

Graph Curve Settings

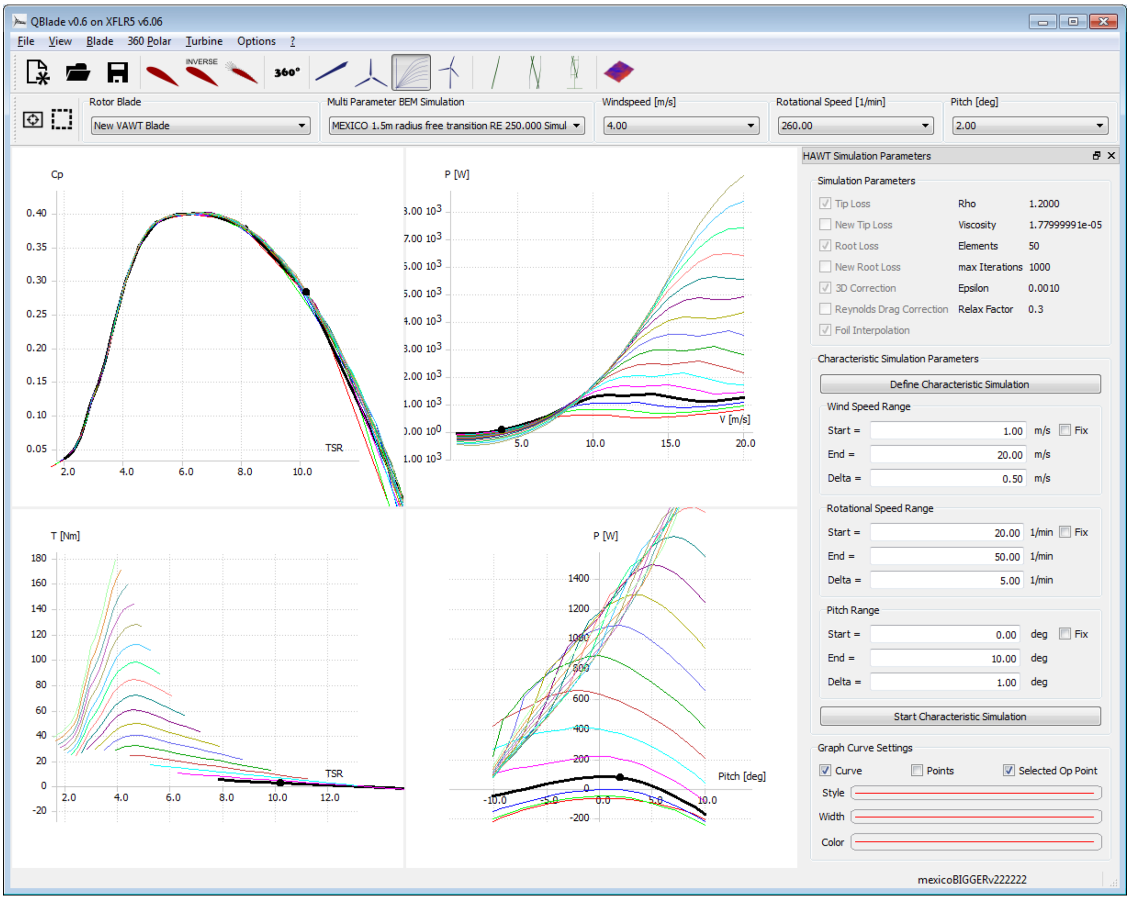
☑ Curve ☐ Points ☑ Selected Op Point
Style
Width
Color

Figure 6.8: The rotor simulation submodule

In the rotor simulation submodule, the user can commit rotor blade simulations over a range of tipspeed ratios. A rotor simulation can only be defined when at least one rotor blade is present in the runtime database. When defining a rotor simulation, the user has to select the desired corrections to the BEM algorithm and the simulation parameters. Once a simulation is defined, the user can select a range of λ-values (tipspeed ratios), and the incremental step for the simulation. A rotor simulation is always carried out dimensionless. The freestream velocity is assumed to be unity and the rotor radius is normalized for the computation. This implies, that no power curve or load curves, like the bending moment, can be computed during a rotor simulation.

6.4 The multi parameter simulation submodule

In the multi parameter simulation submodule simulations can be carried out over a specified range of windspeeds, rotational speeds and pitch angles. To limit excessive usage of memory only the rotor graph curves are stored as a simulation result. By right clicking on a graph the user can specify the main variable and a parameter for each individual graph. For the currently selected main variable and parameter, the resulting series of curves is displayed in each graph. When the selected windspeed, rotational speed or pitch angle is changed in the top toolbar, the series of curves is changed accordingly. This submodule is of great help when designing custom control strategies for variable rotational speed and/or pitch controlled wind turbine rotors.


QBlade v0.6 on XFLR5 v6.06
File View Blade 360 Polar Turbine Options ?
INVERSE
360°
Cp
TSR
P [W]
V [m/s]
T [Nm]
Pitch [deg]
HAWT Simulation Parameters

Rotor Blade: New VAWT Blade

Multi Parameter BEM Simulation: MEXICO 1.5m radius free transition RE 250.000 Simul

Windspeed [m/s]: 4.00

Rotational Speed [1/min]: 260.00

Pitch [deg]: 2.00

Simulation Parameters

☑	Tip Loss	Rho	1.2000
□	New Tip Loss	Viscosity	1.77999991e-05
☑	Root Loss	Elements	50
□	New Root Loss	max Iterations	1000
☑	3D Correction	Epsilon	0.0010
□	Reynolds Drag Correction	Relax Factor	0.3
☑	Foil Interpolation		

Characteristic Simulation Parameters

Define Characteristic Simulation

Wind Speed Range

Start = 1.00 m/s □ Fix
End = 20.00 m/s
Delta = 0.50 m/s

Rotational Speed Range

Start = 20.00 1/min □ Fix
End = 50.00 1/min
Delta = 5.00 1/min

Pitch Range

Start = 0.00 deg □ Fix
End = 10.00 deg
Delta = 1.00 deg

Start Characteristic Simulation

Graph Curve Settings

☑ Curve □ Points ☑ Selected Op Point

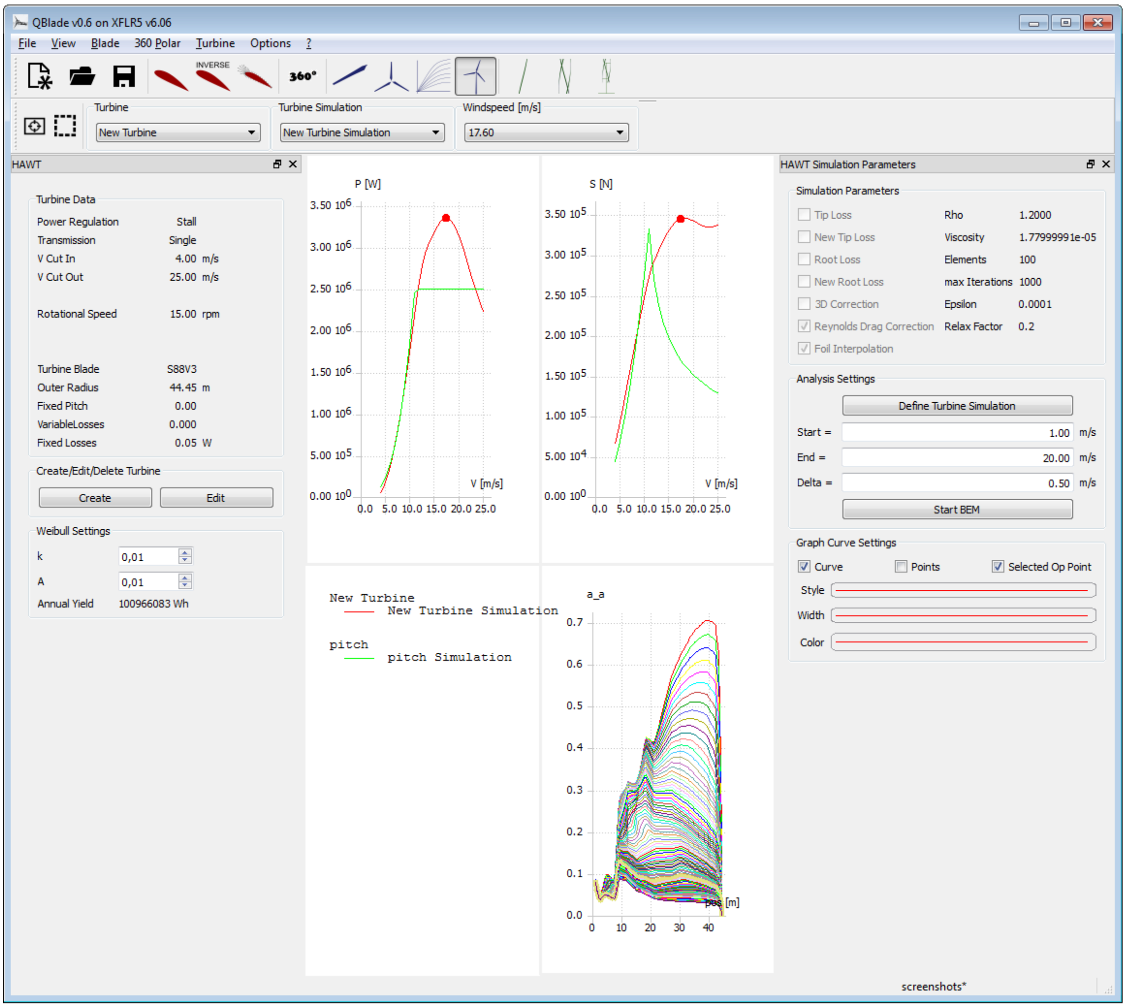
Style

Width

Color

mexicoBIGGERv222222

Figure 6.9: The multi parameter simulation submodule



Turbine: New Turbine
Turbine Simulation: New Turbine Simulation
Windspeed [m/s]: 17.60

Turbine Data

Power Regulation	Stall
Transmission	Single
V Cut In	4.00 m/s
V Cut Out	25.00 m/s
Rotational Speed	15.00 rpm
Turbine Blade	S88V3
Outer Radius	44.45 m
Fixed Pitch	0.00
VariableLosses	0.000
Fixed Losses	0.05 W

Create/Edit/Delete Turbine

Create Edit

Weibull Settings

k	0,01
A	0,01
Annual Yield	100966083 Wh

New Turbine
New Turbine Simulation
pitch
pitch Simulation

HAWT Simulation Parameters

Simulation Parameters

□ Tip Loss	Rho	1.2000
□ New Tip Loss	Viscosity	1.77999991e-05
□ Root Loss	Elements	100
□ New Root Loss	max Iterations	1000
□ 3D Correction	Epsilon	0.0001
☑ Reynolds Drag Correction	Relax Factor	0.2
☑ Foil Interpolation		

Analysis Settings

Define Turbine Simulation

Start =	1.00	m/s
End =	20.00	m/s
Delta =	0.50	m/s

Start BEM

Graph Curve Settings

☑ Curve □ Points ☑ Selected Op Point

Style

Width

Color

Figure 6.10: The turbine definition and simulation submodule

6.5 The turbine definition and simulation submodule

In the turbine definition and simulation submodule, the user can define a wind turbine. To define a wind turbine, a rotor blade must be present in the runtime database. To create a turbine, the turbine type and the turbine parameters have to be specified. The turbine type is defined by:

- Regulation: pitch or stall
- Transmission: single, 2-step or variable

If a pitch regulated turbine is defined, the user has to specify a nominal power output. When the windspeed that yields the nominal power output is reached, the blades are pitched to reduce the power for higher windspeeds to the nominal output. A stall regulated turbine has no pitch control and the power output is limited solely when stall occurs at the rotor. Designing a stall turbine that limits its power to the desired output and at the desired windspeed requires an iterative approach.

For a single speed transmission, the user has to select only one rotational speed, in which the turbine operates over the whole range of windspeeds. For 2-step transmission, two rotational speeds and a windspeed at which the transmission switches have to be selected. A variable transmission turbine has a minimum and a maximum value for the rotational speed. Additionally, the user selects a desired tipspeed ratio, λ_0. From this ratio, a rotational speed is computed for every given wind speed during the simulation. If the computed rotational speeds are lower or larger than the bounding minimum or maximum values, these values give the rotational speed.

The turbine parameters that define a turbine are:

- rotor blade
- cut-in windspeed
- cut-out windspeed
- fixed losses
- variable losses

For every turbine, the rotor blade needs to be defined. This can be any blade that is stored in the runtime database. At the cut-in windspeed, the turbine starts and at the cut-out windspeed the turbine stops operation. To account for power losses that are not of aerodynamical nature but are caused by the efficiency of the generator and the gearbox, a value for fixed losses and a value for variable losses can be selected. The equation in which these losses are implemented is

$$P_{out} = (1 - k_v)P_0 - P_{fixed} \tag{6.8}$$

in which k_v is the variable loss factor and P_{fixed} the fixed loss factor.

If a turbine simulation has been conducted, the user may calculate the annual yield of the turbine by specifying an annual windspeed distribution via the two parameters k and A of the WEIBULL distribution. The probability for a wind-

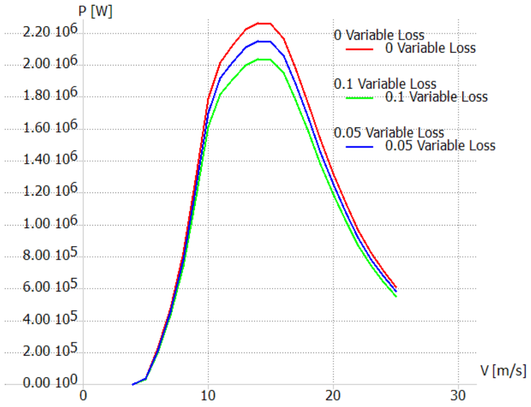


Figure 6.11: Different variable loss factors and their effect on power output

speed to occur is

$$h_W(V_0) = \frac{k}{A}\left(\frac{V_0}{A}\right)^{k-1} \exp\left(-\left(\frac{V_0}{A}\right)^k\right) . \tag{6.9}$$

The probability $f(V_i < V_0 < V_{i+1})$ that a windspeed lies between V_i and V_{i+1} is

$$f(V_i < V_0 < V_{i+1}) = \exp\left(-\left(\frac{V_i}{A}\right)^k\right) - \exp\left(-\left(\frac{V_{i+1}}{A}\right)^k\right) . \tag{6.10}$$

Thus, the annual energy production is calculated as

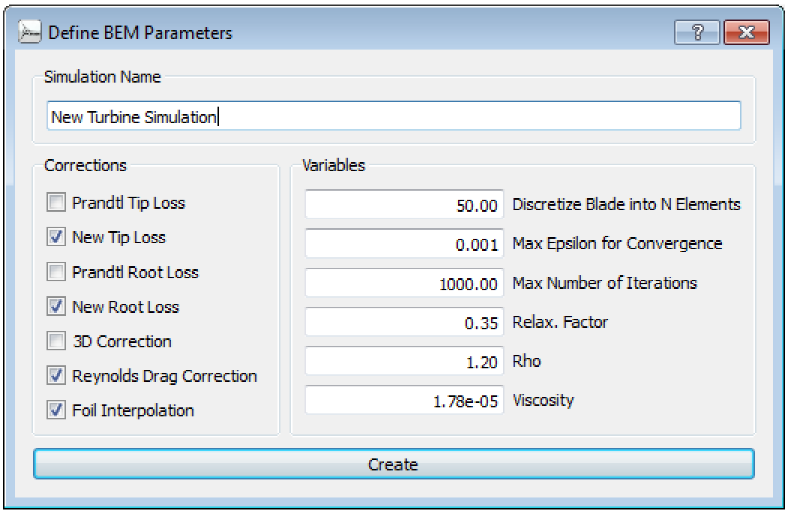
$$AEP = \sum_{i=1}^{N-1} \frac{1}{2}\left(P(V_{i+1}) + P(V_i)\right) \cdot f(V_i < V_0 < V_{i+1}) \cdot 8760 . \tag{6.11}$$

A turbine simulation is carried out over a range of windspeeds with the chosen incremental step size. Depending on the specified rotational speed of the turbine, a tipspeed ratio is computed for every windspeed. Then a BEM simulation over the computed tipspeeds, that is equivalent to a rotor simulation, is carried out.

6.6 Simulation settings

6.6.1 Simulation Parameters

When defining a simulation, the following parameters have to be set:



Define BEM Parameters

Simulation Name

New Turbine Simulation

Corrections

□ Prandtl Tip Loss
☑ New Tip Loss
□ Prandtl Root Loss
☑ New Root Loss
□ 3D Correction
☑ Reynolds Drag Correction
☑ Foil Interpolation

Variables

50.00 Discretize Blade into N Elements
0.001 Max Epsilon for Convergence
1000.00 Max Number of Iterations
0.35 Relax. Factor
1.20 Rho
1.78e-05 Viscosity

Figure 6.12: Simulation definition dialog

- fluid density ρ
- fluid viscosity ν
- number of blade elements N
- maximum number of iterations
- maximum ϵ for convergence
- relaxation factor ω_{relax}

Density

The density of the fluid around the wind turbine is needed to calculate the power output:

$$P = \frac{1}{2}\rho A V_0^3 C_P \; . \tag{6.12}$$

The density is only used to compute the power output for a turbine simulation. During a rotor simulation, all variables are dimensionless and only depend on the tipspeed ratio.

Dynamic viscosity

The dynamic viscosity is needed to compute the local REYNOLDS number along the blade:

$$Re(r) = \frac{V_{rel}(r)c(r)\rho}{\mu} \,. \tag{6.13}$$

The dynamic viscosity is only used to compute the REYNOLDS number during a turbine simulation. During a rotor simulation, all variables are dimensionless and only depend on the tipspeed ratio.

Number of elements

The number of elements specifies into how many elements the blade is divided. This number is independent from the number of blade sections. The BEM algorithm is executed once for every element. The input values, like chord and twist, are interpolated between the blade stations, where they are defined, and computed for the centers of the elements. The elements are distributed using sinusoidal spacing. This allows for more elements to be placed in the tip and root region where largest gradients usually are to be expected. By using sinusoidal spacing, the overall number of elements that are required is less and the computational time is reduced.

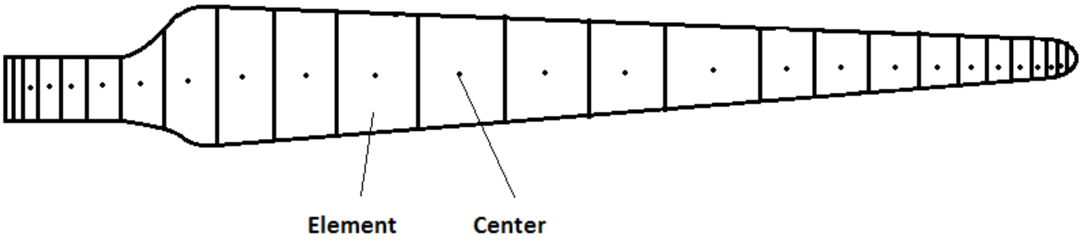


Figure 6.13: Sinusoidal spaced elements along the blade

All the variables resulting from the BEM simulation are computed for the center of an element and are treated as the averaged values over an element. This solves a problem that arises if the values would be computed for the boundaries of an element and then interpolated to the center from there. When the PRANDTL tip loss factor F goes to zero at the tip or at the hub, numerical instabilities arise and result in a non converging iteration. This problem is skipped because the center of an element can be arbitrarily close to the tip or hub of the blade but never be on the same position, it is always $\frac{\Delta_i}{2}$ away (Δ_i is the width of the i^{th} element).

The forces per length, P_N and P_T, that were computed for the element center are integrated over the whole element to yield the element's contribution to the total torque and thrust.

Convergence criterion

The convergence criterion ϵ defines when an iteration has converged. The maximum of the difference of axial and radial induction factor between the last and the current iteration has to be below ϵ for convergence. A recommendation is $\epsilon = 10^{-5}$.

$$\texttt{max}(|a - a_{old}|, |a' - a'_{old}|) < \epsilon \; . \tag{6.14}$$

Maximum number of iterations

The maximum number of iterations prevents that the algorithm may get stuck in an infinite loop.

Relaxation factor

A common problem during the iteration loop of a BEM computation is the fluctuating behavior of the axial induction factor. The reason for this fluctuation is the periodical switching of the turbines loading state between light and heavy loading [8] (see section *The turbulent wake state*). This may lead to a stop of the iteration after the maximum number of iterations is reached and impacts both the code's performance and its accuracy. In [8] MAHERI proposes to introduce a relaxation factor ω_{relax}, to overcome these fluctuations.

The relaxation factor is introduced in the iteration after a new value, a_{k+1}, for the axial induction factor has been calculated:

$$a_{k+1} = \omega_{\text{relax}} a_{k+1} + (1 - \omega_{\text{relax}}) a_k \; ; \quad 0 < \omega_{\text{relax}} < 1 \; . \tag{6.15}$$

The convergence rate of the BEM code strongly depends on the initial guess value for the axial induction factor. If the initial guess ($a = 0$ in this implementation) is in the neighborhood of the final result, convergence is achieved significantly faster. To further accelerate the convergence rate of the BEM code MAHERI proposes that for the first few iterations a relaxation factor $\omega_{\text{relax}} = 1$ should be

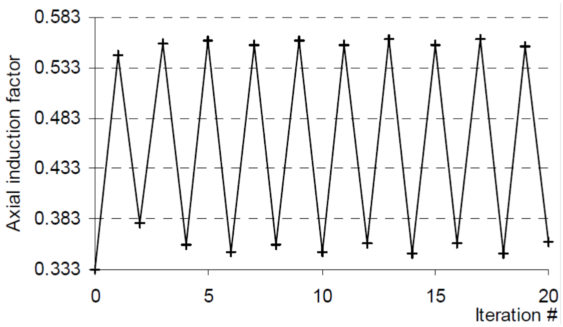


Figure 6.14: Fluctuation of the axial induction factor around the light and heavy loading state [8]

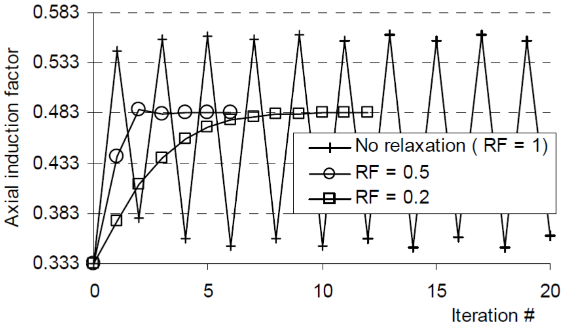


Figure 6.15: Damped fluctuation of the axial induction factor for different relaxation factors [8]

applied, to let the first few oscillations happen. These oscillations then mark the boundary of the neighborhood of the final result. With a three-point-equation the axial induction factor is then placed inside this neighborhood.

$$a_{k+1} = \frac{1}{4}a_{k+1} + \frac{1}{2}a_k + \frac{1}{4}a_{k-1} \, . \tag{6.16}$$

From there the iteration proceeds as normal with the desired relaxation factor and eq. 6.15 applied.

6.6.2 Corrections

Due to the two dimensional nature of the BEM theory, three dimensional effects, can not be accounted for by the classical BEM. This leads to large deviations of the computed data compared with measured turbine data, especially under the influence of stall. The three dimensional effects, responsible for this are:

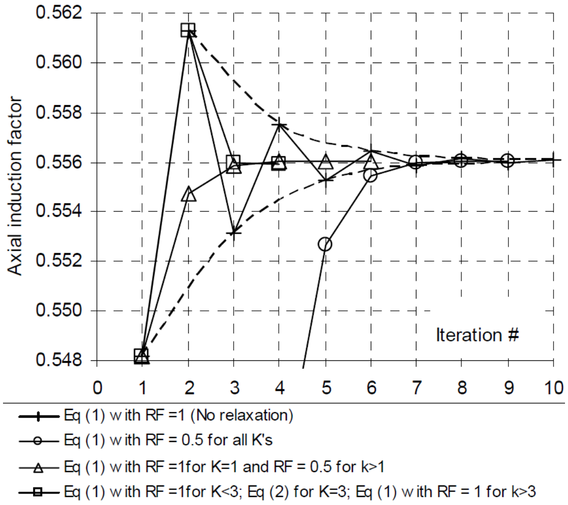


Figure 6.16: Accelerated convergence by placing the induction factor inside the neighborhood of the final result [8]

- The wind velocity in the rotor plane is assumed to be constant. In reality there are large variations in the wind velocities.

- In the area of the blade root and close to the blade tip the finiteness of the blades leads to vortices being shed. This results in a loss of energy extraction.

- The rotation of the rotor is only considered in the velocity distribution along the blade, but leads to a dynamic pressure gradient that particularly affects detached flow.

- The blades' boundary layer is subject to centrifugal forces.

- Centrifugal pumping, a radial flow along the blades caused by Coriolis forces and the radial pressure gradient, occurs.

- The momentum balance is only valid in the rotor plane. Bending of the blades out of the rotor plane leads to errors.

Different, often semi empirical correction algorithms to account for these effects exist. The following ones can be selected to be included in the QBlade BEM simulation:

- Prandtl tiploss [7]

- Prandtl rootloss [7]

- New tiploss model after Shen et al. [13]
- New rootloss model after Shen et al. [13]
- 3D correction after SNEL [17]
- Reynolds number drag correction, from Hernandez and Crespo [18]
- Airfoil interpolation

Any combination of these corrections can be added to a simulation with one exception. The tip- or rootloss model after Shen can never be used in combination with the Prandtl root- or tiploss model. That is because the Prandtl tiploss factor F is included in the models by Shen a priori.

Prandtl tip- and rootloss

The Prandtl tiploss factor is an approved procedure for the correction of profile data to get a better agreement between measured and computed data. Prandtl modeled the helicoidal vortex sheet wake behind the rotor as a succession of solid disks moving with the wake. Prandtl's analysis can be found in detail in [7]. In combination with the tiploss model, a rootloss model is often used to account for the influence of the vortex shedding at every blade's root on the induction factors.

New tip- and rootloss after Shen

In 2005, Shen examined the various tiploss models by Prandtl, Wilson & Lissaman and Mikkelsen that are used in modern BEM computations. He found that they all lack rigorous consistency when the tip of the blade is approached [13]. At the tip the tiploss factor F always tends to zero, thus the axial induction factor tends to unity. This implies that the axial velocity and the angle ϕ always tend to zero at the tip, no matter what shape, pitch or other airfoil parameters. On the other hand, he found that the Prandtl correction overestimates the loads at the tip when compared to experimental data. Based on data from the NREL experimental rotor, he proposed a new tiploss model to overcome these inconsistencies. Based on the assumption that the force should tend to zero at the tips because of pressure equalization, he introduced a new correction term to correct the force coefficients C_n and C_t. It is implemented in combination with the Prandtl tiploss correction.

3D correction after Snel

Himmelskamp discovered in 1945 that the maximum lift coefficient of profiles on a rotating rotor blade is significantly higher than the maximum lift coefficient of the same profile measured on a stationary rotor. The centrifugal force accelerates the boundary layer radially. this results in a thinner boundary layer where the stall is delayed. At the same time air flowing radially, in a rotating reference system, generates a Coriolis force opposite to the rotational direction of the rotor. This force is opposing the rise in pressure of the profiles suction side and delays the stall even more. This effect is called stall delay or Himmelskamp effect and can be taken into account by modifying the two dimensional profile data. For the affected profiles the stalled region will shift to higher angles of attack.

With a Viscous Inviscid Interaction Method, Snel et al. investigated the flow around a rotating rotor blade and developed a semi-empirical formula to correct 2D profile data [17] based on these investigations. According to Snel, only the lift but not the drag coefficient, needs to be modified.-

Reynolds number drag correction

The changes in lift and drag polars due to Reynolds number effects are not included in QBlade. The polars are always computed for a fixed Reynolds number. During the simulation of a turbine, the Reynolds number is changing for every operational point. The user should carefully check how large the deviation is for every individual case. Hernandez and Crespo [18] suggested a correction in which the lift polar remains unchanged and the drag polar is corrected by scaling the drag coefficient inversely with the Reynolds number.

$$C_D = C_{D,Ref}\left(\frac{Re_{Ref}}{Re}\right)^{0.2} \tag{6.17}$$

In 6.17 Re_{Ref} depicts the Reynolds number for which the polar data was computed. It is important to note that this correction represents a very simplistic approach to the estimation of the drag coefficient. Especially for low Reynolds numbers the drag behavior can be very complex.

Foil interpolation

The foil interpolation, in effect, is not a correction to the BEM algorithm. It merely is the most simple solution to a problem that arises during the discretization of the blade. As stated previously, a blade is defined in sections. Every section may have a different airfoil, that defines the sections geometry. The geometry in between two sections, of a real blade, is a liner interpolation between the two airfoils. The problem now is that only polar data for the airfoils at every section, but no data for the interpolated airfoils in between are present in the database. If the option Foil interpolation is not selected, the BEM treats the blade as in Fig. 6.17.

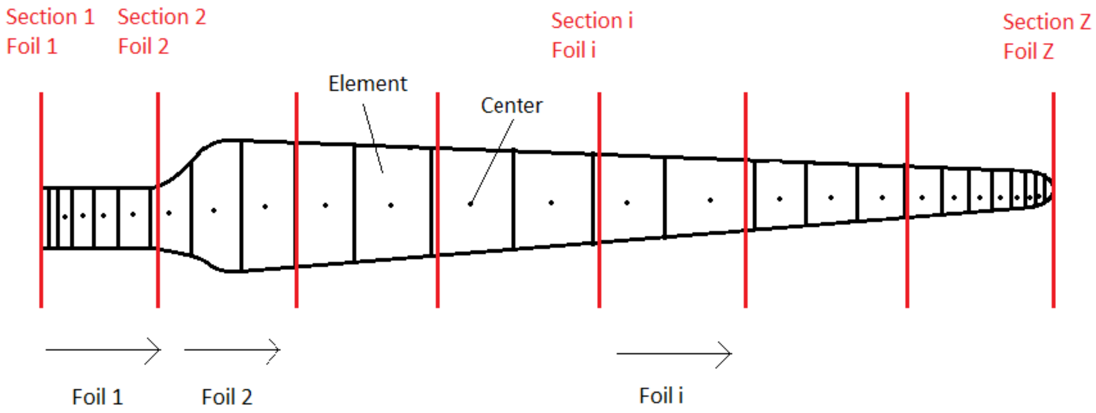


Figure 6.17: Foil distribution along the blade without interpolation

All elements, whose centers lie between section 1 and section 2 are linked to the polar data from foil 1. The last airfoil, at position Z, is not included at all in the simulation. From the element that lies just before section 2 , to the first element that lies after section 2 there is a discontinuity, as the foils rapidly change from foil 1 to foil 2. This is expressed in the simulation results, if interpolation is not included.

When the foil interpolation is switched on, the polar data, that is used for the BEM computation of an element, is a linear interpolation between the polar data of the bounding airfoils. This interpolation more accurately represents a "true blade" geometry. Strictly speaking, the linear interpolation between two polars never represents the true polar of the intermediate airfoil. This interpolation is just the most simple approximation to the real polar, that is not present in the database. However, the accuracy can be arbitrarily improved by importing the geometric data for these intermediate airfoils and create new sections where the intermediate airfoils are placed. Another possibility is to use XFOIL's dynamic coordinate mixing function, where intermediate airfoil geometries can be created and simulated in XFOIL.

6.7 Simulation results

6.7.1 Data storage and visualization

There are two different types of simulation results for a BEM computation. Global variables, or rotor variables, are values that characterize the rotor, or turbine, as a whole. The C_P value is such a global variable. Every increment of tip speed ratio or wind speed, that was simulated yields one C_P value. The C_P over λ curve gives only information about the overall rotor performance, but not about the local events taking place at the blades. These global variables are computed out of the local variables. Every point in the C_P curve represents a BEM computation for one tipspeed ratio or windspeed.

The thrust T is calculated by adding up the normal forces acting on the elements. The local variables, or blade variables, like the AoA α, give insight in the local conditions at the blade. Every curve of a local variable represents a BEM computation for one tipspeed ratio or windspeed.

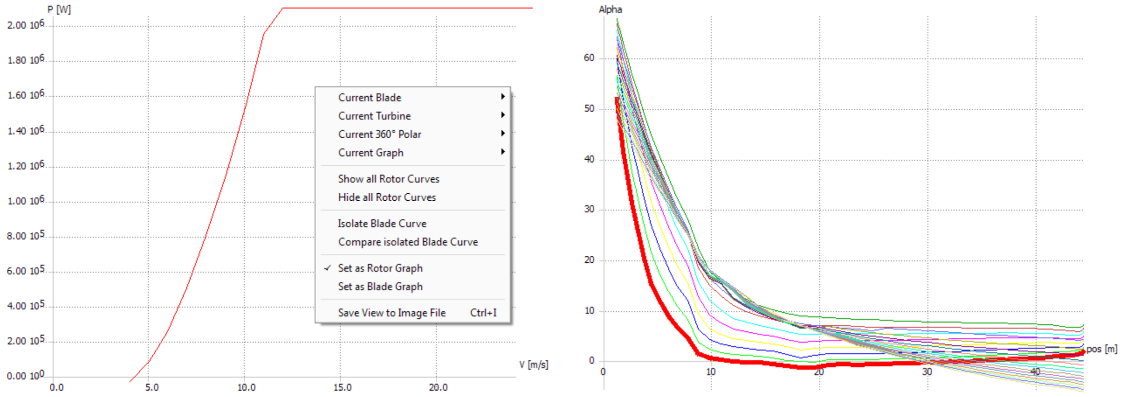


Figure 6.18: Rotor graph to the left and blade graph to the right, context menu

In the context menu of a graph, the user can set the graph as rotor graph, to display global variables, or as blade graph to display local variables.

6.7.2 Variable listings

As stated before, all the variables calculated in a rotor simulation are dimensionless:

1. global:
   - power coefficient C_P
   - thrust coefficient C_T
   - tipspeed ratio λ
   - power coefficient based on tipspeed $K_P = \frac{C_P}{\lambda^3}$
   - inverse tipspeed ratio$\frac{1}{\lambda}$
2. local:
   - axial induction factor a
   - radial induction factor a'
   - local tipspeed ratio λ_{loc}
   - radial position r
   - dimensionless normal force component C_n
   - dimensionless tangential force component C_t
   - inflow angle φ
   - relative angle α
   - twist angle θ
   - chord c
   - lift coefficient C_L
   - drag coefficient C_D

- lift to drag Ratio $\frac{C_L}{C_D}$
- PRANDTL tiploss factor F
- number of iterations n
- annulus averaged axial induction factor
- annulus averaged radial induction factor

In a turbine simulation, the following variables are computed additionally to the variables resulting from a rotor simulation:

1. global:
   - power $P[\mathrm{W}]$
   - thrust $T[\mathrm{N}]$
   - windspeed $V_0[\mathrm{m/s}]$
   - angular speed $\omega[\mathrm{rpm}]$
   - pitch angle β
   - WEIBULL probability h_w
   - WEIBULL probability · windspeed^3 $h_w \cdot V_0^3[\mathrm{m^3/s^3}]$
   - root bending moment $M[\mathrm{Nm}]$
   - power coefficient including losses $C_{P,loss}$
2. local:
   - local REYNOLDS number $Re_{loc} = \frac{V_{rel} c \rho}{\mu}$
   - Re deviation from polar simulation $Re_{loc} - Re_{polar}$
   - critical roughness $k_{critical} \approx 100 \frac{\mu}{V_{rel}}[\mathrm{mm}]$ from [12]
   - resultant velocity $V_{rel}[\mathrm{m/s}]$

- tangential force per length $P_T[\mathrm{N/m}]$
- normal force per length $P_N[\mathrm{N/m}]$
- Mach number Ma
- circulation $\Gamma = \frac{1}{2}C_L V_{rel} c[\mathrm{m^2/s}]$

7 The QBlade VAWT Module

7.1 Basics

There are many different types and shapes of wind turbines. Those with a vertical axis of rotation are called Vertical Axis Wind Turbines (VAWT). QBlade includes a module for the simulation of those wind turbines. The implemented algorithm is applicable for the performance analysis of lift based VAWT's such as the classical "Eggbeater" Darrieus rotor. Based on the calculated airfoils, polars and 360 polars, the blade shape is defined in the *blade design submodule*. This part of the program provides a range of scaling and optimization functions as well. In the following, the user can choose between *rotor simulation* and *turbine simulation* . The former one is a dimensionless calculation for a range of tip-speed ratios whereas the latter one requires the definition of concrete turbine parameters such as rpm and is executed for a range of windspeeds. In either module, simulation parameters and corrections can be selected for the calculation. For visualization purposes, three different graph types are available to plot the simulation data: the blade graph for vertical plots, the azimuthal graph for circumferential plots and the rotor graph for plots of global variables, such as the power coefficient.

7.1.1 Method of operation

The cross section of a VAWT blade is an airfoil producing lift and drag that drive the wind turbine. The force on a blade section depends on both the undisturbed air velocity V_∞ and the rotational speed ω of the moving blade. The relative velocity at the blade is

$$W = V_\infty - R\omega \tag{7.1}$$

and the angle of attack of the airfoil section is the angle between the relative velocity W and the chord line c. The resultant force on each airfoil section can either be decomposed in lift and drag or in tangential and normal components. The lift and drag coefficients for all angles of attack are already known in terms

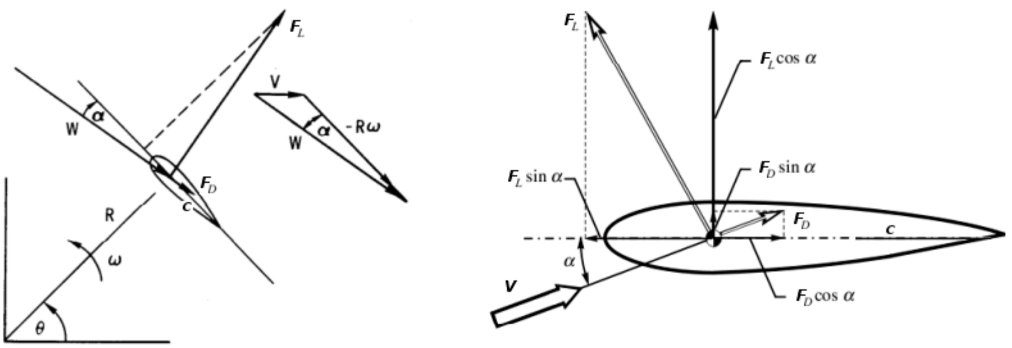


Figure 7.1: Relative velocity [15] and forces on a VAWT blade cross section [16]

of the extrapolated 360 polars. The tangential force drives the turbine and is hence vitally important for the calculation of the power coefficient.

7.1.2 The Double-Multiple Streamtube Model

Concept

This model has been developed by Ion Paraschivoiu for the performance analysis of Darrieus type rotors. It is basically a derivation of the actuator disk theory and the blade element theory.

The streamtube flowing through the VAWT rotor is splitted up into a set of smaller streamtubes. The blades of the rotor pass through each of these streamtubes on their 360 degree path and extract energy from the fluid by reducing its velocity. Thus, the standard actuator disk theory can be applied for every streamtube in particular.

Due to the circular path of a VAWT blade, it passes each streamtube twice. These two steps of energy extraction are taken into consideration in the DMS model by dividing the rotor into an upstream and downstream half. Each one is represented by a separate rotor disk. This double disk acts like two single actuator disks in tandem. The subsequent iteration algorithm is hence executed twice for every streamtube.

Additionally, the DMS algorithm is executed for each height position where the fluid flows through the respective blade section. As already described, a rotor blade is composed of several blade sections. According to the selected number of elements, the intermediate blade sections are interpolated from the given geometry. All sections can be treated as independent 2D foils producing lift and

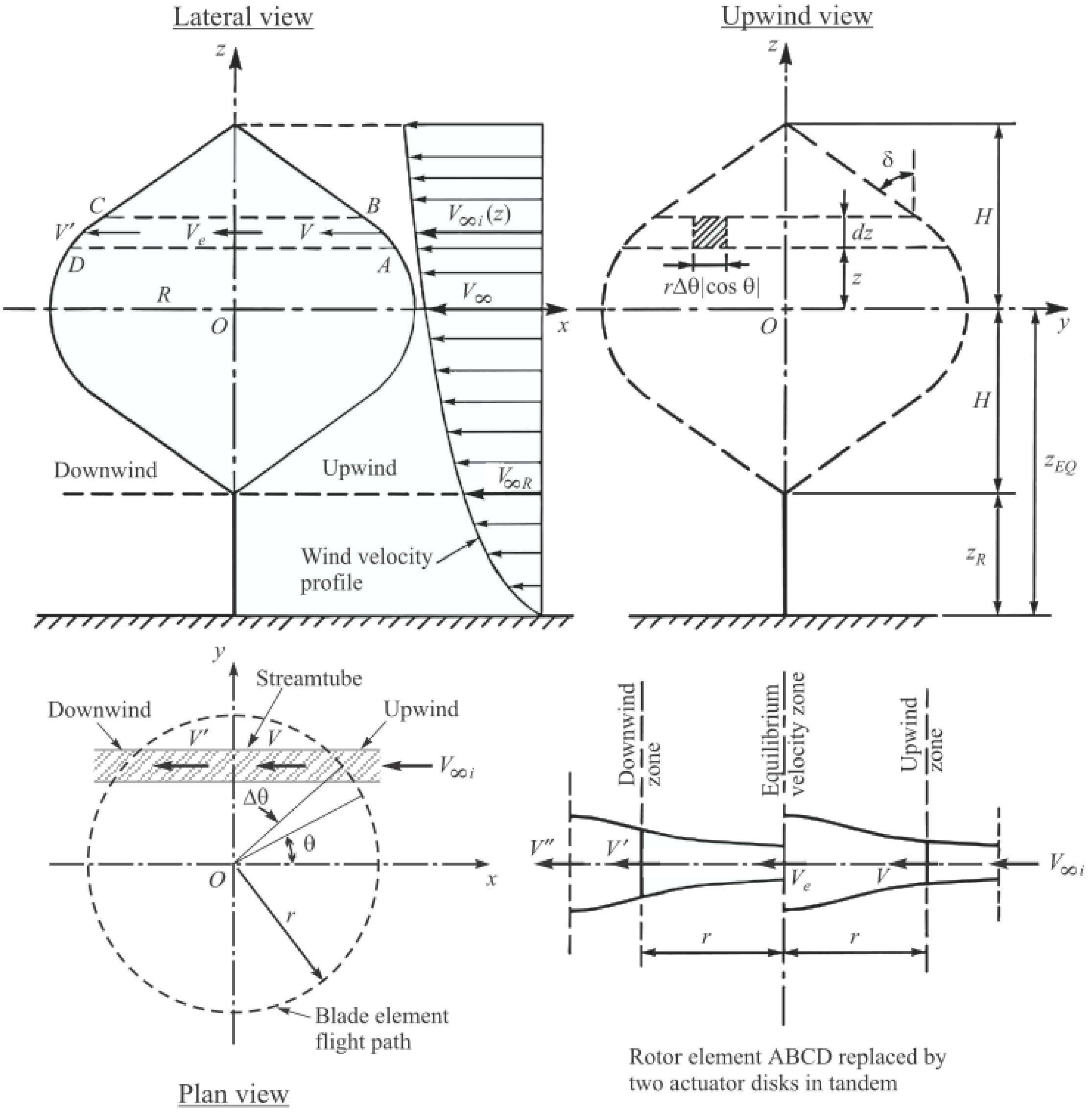


Figure 7.2: VAWT rotor geometry definition and two actuator disks in tandem [16]

drag as a function of their local angle of attack α. The resultant total force on a blade can be found by integrating over the whole blade length.

Streamtube models combine the preceding approaches of the actuator disk theory and the blade element method by iteratively balancing momentum conservation and the forces on the rotor blades until system convergence is reached.

7.1.3 Velocities

There are five important velocities in a DMS calculation:

1. **inflow velocity** V_∞
   of the undisturbed freestream flow

2. **upwind induced velocity** V
   due to the energy extraction of the blade in the upstream rotor half

3. **equilibrium velocity** V_e
   in the plane between up- and downstream rotor half (represents wake velocity of upstream rotor disk and inflow velocity of downstream rotor disk)

4. **downwind induced velocity** V'
   due to the energy extraction of the blade in the downstream rotor half

5. **wake velocity** V''
   of the whole double disk

According to these velocity determinations, one can define the interference factors for the energy extraction in the up- and downstream rotor half:

$$u = \frac{V}{V_\infty} \tag{7.2}$$

$$u' = \frac{V'}{V_e} \tag{7.3}$$

An interference factor of $u = 1$ stands for zero energy extraction. If $u = 0$, the fluid velocity is slowed down to zero.

7.1.4 Iteration procedure

The iteration variable of the DMS algorithm is the interference factor u. It can be variable or constant for all azimuthal positions (streamtubes) depending on user selection.

A rotor simulation is executed for a range of tipspeed ratios. The global tipspeed ratio is the relative rotational speed of a VAWT rotor and is defined as

$$TSR = \frac{R\omega}{V_\infty} \tag{7.4}$$

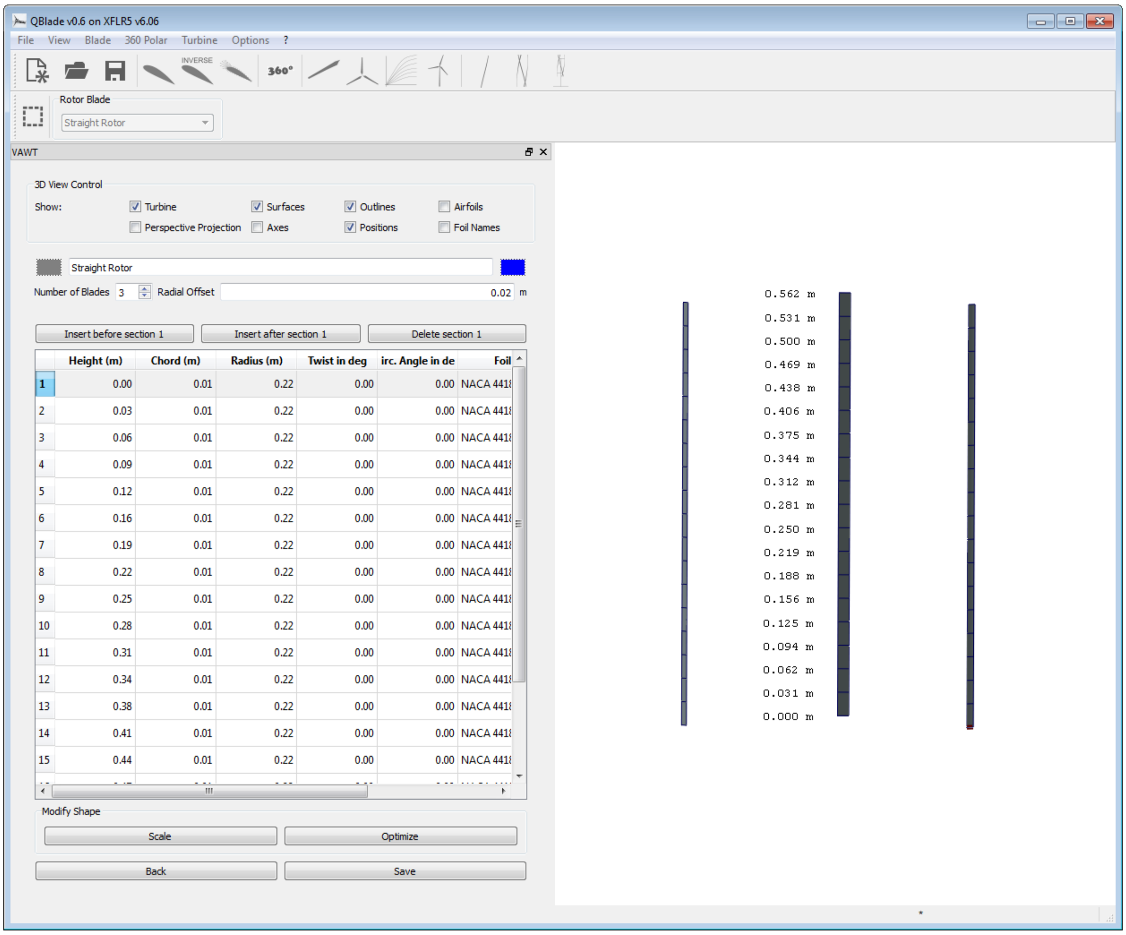
where R is the radius (of the equator), ω is the angular speed of the blade and V_∞ is the freestream inflow velocity. A turbine simulation is generally executed for a range of windspeeds, although in fact the windspeed is translated into a tipspeed ratio on the basis of the turbine data as well.

For each tipspeed ratio, the iteration is executed at every height position for all upwind azimuthal angles (streamtubes) until the user defined convergence condition is met for either the whole rotor half (constant u) or the particular azimuthal position (variable u). In the process, local and global charcteristic simulation data is stored and can be visualized after the calculation.

7.1.5 Limitations

Please note that there have been observed some convergence and plausibility problems for unrealistic geometries (very high solidities) as well as for low windspeeds (high tipspeed ratios) as the model does not consider backflow effects. Consequently, the results at high tipspeed ratios should not be trusted after the "kinks" of the power curve occur. As most of the reference data does not show this area either, it might be a general problem of the DMS algorithm that has to be reconsidered in the future.

7.2 The blade design and optimization submodule



Rotor Blade: Straight Rotor

3D View Control

Show: ☑ Turbine ☑ Surfaces ☑ Outlines □ Airfoils

□ Perspective Projection □ Axes ☑ Positions □ Foil Names

Straight Rotor

Number of Blades 3 Radial Offset 0.02 m

	Height (m)	Chord (m)	Radius (m)	Twist in deg	irc. Angle in de	Foil
1	0.00	0.01	0.22	0.00	0.00	NACA 4418
2	0.03	0.01	0.22	0.00	0.00	NACA 4418
3	0.06	0.01	0.22	0.00	0.00	NACA 4418
4	0.09	0.01	0.22	0.00	0.00	NACA 4418
5	0.12	0.01	0.22	0.00	0.00	NACA 4418
6	0.16	0.01	0.22	0.00	0.00	NACA 4418
7	0.19	0.01	0.22	0.00	0.00	NACA 4418
8	0.22	0.01	0.22	0.00	0.00	NACA 4418
9	0.25	0.01	0.22	0.00	0.00	NACA 4418
10	0.28	0.01	0.22	0.00	0.00	NACA 4418
11	0.31	0.01	0.22	0.00	0.00	NACA 4418
12	0.34	0.01	0.22	0.00	0.00	NACA 4418
13	0.38	0.01	0.22	0.00	0.00	NACA 4418
14	0.41	0.01	0.22	0.00	0.00	NACA 4418
15	0.44	0.01	0.22	0.00	0.00	NACA 4418

Modify Shape

Figure 7.3: Blade design submodule

The *blade design and optimization submodule* allows the rotor design for different VAWT types. A blade consists of a finite number of blade sections. In a tabular, vertical, radial and circumferential position, chord length, airfoil, 360 polar and other parameters of each blade section can be defined. Additionally, scaling and optimization functionalities can be used to improve and speed up the blade design process.

7.2.1 Blade optimization

- **Straight blades**

  The Giromill and H-Darrieus rotor are straight-bladed VAWT rotor geome-

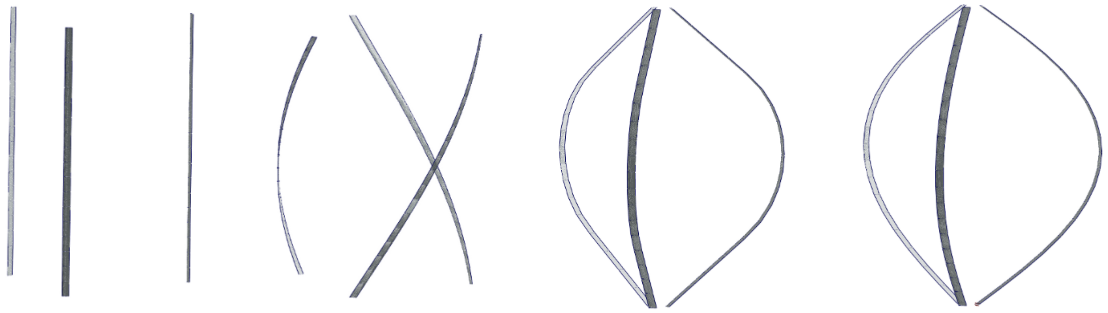


Figure 7.4: Blade shapes: straight, helical, Sandia and Troposkien

tries. Their airfoil sections all have the same radius and circumferential position, that has to be set in the optimization dialog.

- **Helical blades**

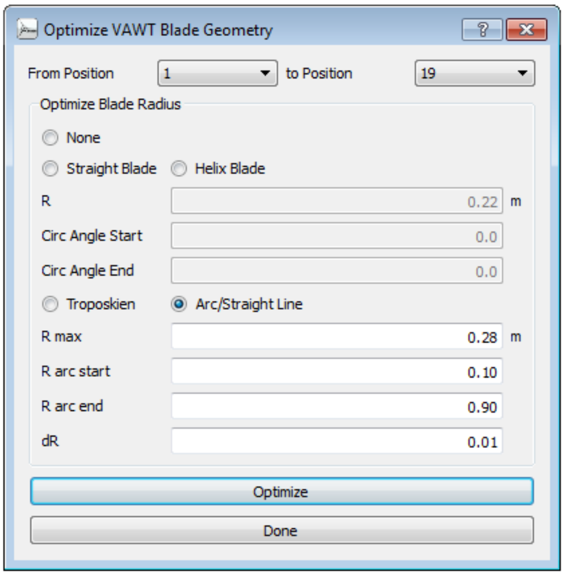
  The helical blade shape takes account of destructive torque pulsations resulting from the finite number of blades. By altering the cirmcumferential position of the sections, the torque production is spread evenly over the blade revolution. The circular angles of a helical blade can be defined for a start and end section in the optimize dialog, the intermediate angles are derived from a linear interpolation.

- **Troposkien-shaped blades**

  The Troposkien shape is the shape of an idealized rope that is attached at its ends and rotated around a vertical axis. As a rope transmits only drag forces and no bending moments, this shape is used to reduce the stress experienced by VAWT blades. In the optimization dialog, the radial maximum deflection of the blade from the rotation axis needs to be defined.

- **Sandia-shaped blades**

  To simplify the blade manufacturing process of Darrieus wind turbines, the Troposkien shape can be approximated by a circular arc segment and two straight line parts. This approximation by the Sandia National Laboratories is implemented as well. To find a fitting arc radius for a certain troposkien-shape, the user has to define a start and end radius as well as a step size. The best radius is calculated on the basis of the smallest maximum distance between Troposkien and Sandia shape.



From Position: 1 to Position: 19

Optimize Blade Radius

○ None

○ Straight Blade ○ Helix Blade

R: 0.22 m

Circ Angle Start: 0.0

Circ Angle End: 0.0

○ Troposkien ● Arc/Straight Line

R max: 0.28 m

R arc start: 0.10

R arc end: 0.90

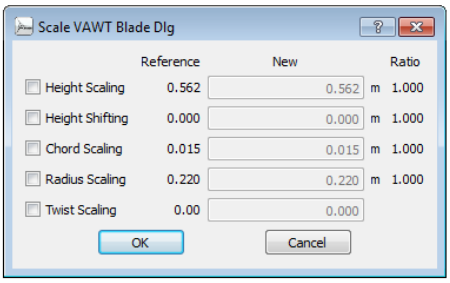
dR: 0.01

Figure 7.5: Blade optimization dialog

7.2.2 Blade scaling

The following scaling functionalities speed up the rotor modification:

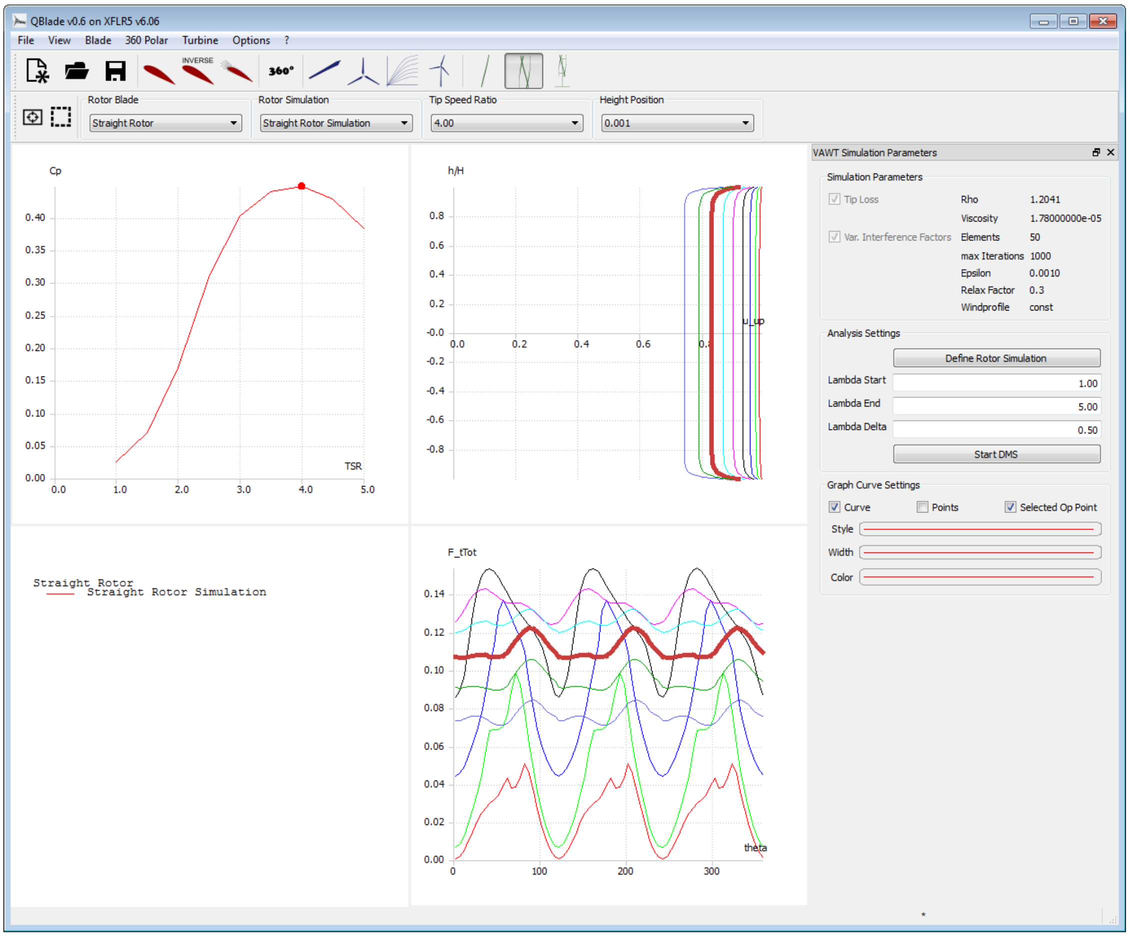
- **height scaling**: change the overall rotor height
- **height shifting**: vertically shift whole rotor without changing it
- **chord scaling**: scales the chord length
- **radius scaling**: scales the radial position
- **twist scaling**: sets a constant twist angle



	Reference	New		Ratio
□ Height Scaling	0.562	0.562	m	1.000
□ Height Shifting	0.000	0.000	m	1.000
□ Chord Scaling	0.015	0.015	m	1.000
□ Radius Scaling	0.220	0.220	m	1.000
□ Twist Scaling	0.00	0.000		

Figure 7.6: Blade scaling dialog

7.3 The rotor simulation submodule



Rotor Blade: Straight Rotor
Rotor Simulation: Straight Rotor Simulation
Tip Speed Ratio: 4.00
Height Position: 0.001

☑ Tip Loss
☑ Var. Interference Factors

Rho	1.2041
Viscosity	1.78000000e-05
Elements	50
max Iterations	1000
Epsilon	0.0010
Relax Factor	0.3
Windprofile	const

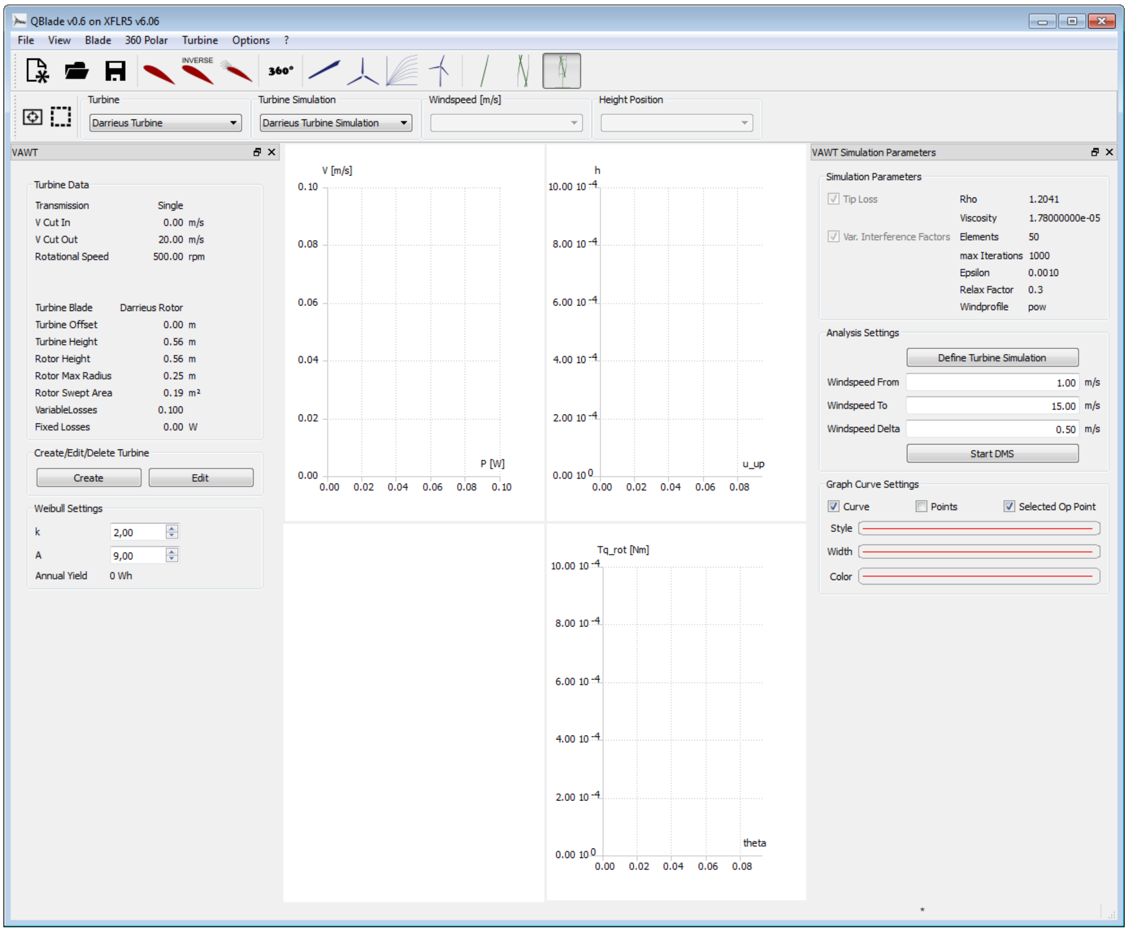
Lambda Start 1.00
Lambda End 5.00
Lambda Delta 0.50

☑ Curve ☐ Points ☑ Selected Op Point

Figure 7.7: The rotor simulation submodule

In the rotor simulation submodule, the user can commit rotor blade simulations over a range of tipspeed ratios. A rotor simulation can only be defined when at least one rotor blade is present in the runtime database. When defining a rotor simulation, the user has to select the desired corrections to the DMS algorithm and the simulation parameters. Once a simulation is defined, the user can select a range of λ-values (tipspeed ratios), and the incremental step for the simulation. A rotor simulation is always dimensionless. The freestream velocity is assumed to be unity and the rotor radius and height are normalized for the computation. This implies, that no power curve or load curves can be computed during a rotor simulation.

7.4 The turbine definition and simulation submodule



Turbine: Darrieus Turbine

Turbine Simulation: Darrieus Turbine Simulation

Windspeed [m/s]

Height Position

Turbine Data

Transmission	Single
V Cut In	0.00 m/s
V Cut Out	20.00 m/s
Rotational Speed	500.00 rpm
Turbine Blade	Darrieus Rotor
Turbine Offset	0.00 m
Turbine Height	0.56 m
Rotor Height	0.56 m
Rotor Max Radius	0.25 m
Rotor Swept Area	0.19 m²
VariableLosses	0.100
Fixed Losses	0.00 W

Create/Edit/Delete Turbine

Weibull Settings

k	2,00
A	9,00
Annual Yield	0 Wh

Simulation Parameters

☑ Tip Loss

☑ Var. Interference Factors

Rho	1.2041
Viscosity	1.78000000e-05
Elements	50
max Iterations	1000
Epsilon	0.0010
Relax Factor	0.3
Windprofile	pow

Analysis Settings

Windspeed From	1.00 m/s
Windspeed To	15.00 m/s
Windspeed Delta	0.50 m/s

Graph Curve Settings

☑ Curve ☐ Points ☑ Selected Op Point

Figure 7.8: The turbine definition and simulation submodule

In the turbine definition and simulation submodule, the user can define and simulate a wind turbine. To define a wind turbine, a rotor blade must be present in the runtime database and the turbine parameters have to be specified. The parameters that define a turbine are:

- rotor blade
- cut-in windspeed
- cut-out windspeed
- turbine offset
- fixed and variable losses

The cut-in and -out windspeed define the operational windspeed range of the turbine. The turbine offset allows the positioning of a rotor on buildings with different heights, where it experiences varying inflow velocities due to the natural wind profile caused by friction. The fixed and variable losses account for gear and generator losses.

Furthermore, the transmission can be selected. The turbine angular speed can be a constant value, variable or prescribed. Either the constant angular speed for single speed turbines or the minimum and maximum angular speed for variable speed need to be defined. For the latter, a design opoint and corresponding TSR has to be selected. For prescribed ω, the values can be set in a tabular.

The turbine simulation has to be defined by the selection of simulation parameters , a wind profile and corrections. A turbine simulation is carried out over a range of windspeeds with the chosen incremental step size. If a turbine simulation has been conducted, the user may calculate the annual yield of the turbine by specifying an annual windspeed distribution via the two parameters k and A of the WEIBULL distribution. Thus, the annual energy production can be calculated.

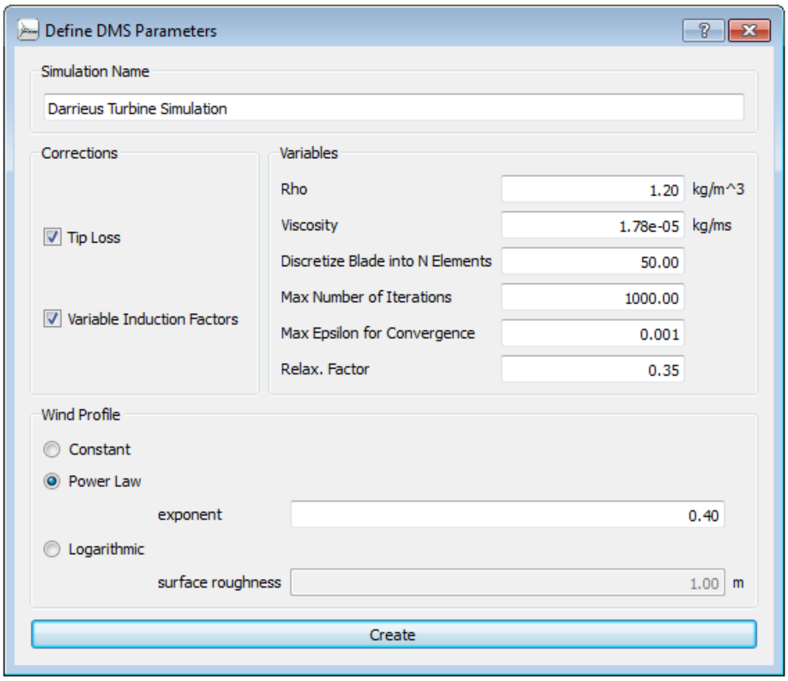
For further information about the turbine definition, see section 6.5.

7.5 Simulation settings

7.5.1 Simulation Parameters

For a simulation, the following user-specific simulation parameters can be adjusted:

- fluid density ρ
- fluid viscosity ν
- number of blade elements N
- maximum number of iterations
- maximum ϵ for convergence
- relaxation factor ω_{relax}



Define DMS Parameters

Simulation Name

Darrieus Turbine Simulation

Corrections

☑ Tip Loss

☑ Variable Induction Factors

Variables

Rho	1.20	kg/m^3
Viscosity	1.78e-05	kg/ms
Discretize Blade into N Elements	50.00	
Max Number of Iterations	1000.00	
Max Epsilon for Convergence	0.001	
Relax. Factor	0.35	

Wind Profile

○ Constant

● Power Law

exponent 0.40

○ Logarithmic

surface roughness 1.00 m

Figure 7.9: Simulation definition dialog

The relaxation factor is implemented as

$$u_{k+1} = \omega_{\text{relax}} \cdot u_{k+1} + (1 - \omega_{\text{relax}}) \cdot u_k \; ; \quad 0 < \omega_{\text{relax}} < 1 \tag{7.5}$$

and the convergence criterion ϵ is applied to the alteration rate of the interference factor Δu. The iterative process stops when either convergence or the maximum number of iterations is reached. For further information, see section 6.6.1.

7.5.2 Wind profile

Besides the above simulation parameters, the turbine simulation additionally allows the specification of an inflow wind profile. There are three possible choices:

1. **constant**
   uniform velocity profile (like in a rotor simulation)

2. **power law**

$$V_\infty(z) = v_{\text{ref}} \cdot \left(\frac{z}{z_{\text{ref}}} \right)^\alpha \tag{7.6}$$

where v_{ref} is the simulated windspeed, z_{ref} is the equator (or half-hight position) of the turbine and α is the roughness exponent (e.g. $\alpha = 1/7 = 0.143$ for open land and $\alpha = 0.11$ for open water).

3. **logarithmic**

$$V_\infty(z) = v_{\text{ref}} \cdot \left(\frac{\log(z/z_0)}{\log(z_{\text{ref}})/z_0} \right)^\alpha \tag{7.7}$$

with the already stated settings for v_{ref} and z_{ref} and the surface roughness z_0 depending on the surroundings of the wind turbine (e.g. $z_0 = 0.2\,\text{mm}$ for open water and $z_0 = 2\,\text{m}$ for highly built-up areas).

7.5.3 Corrections

To take account for the finite length of the blade, a **tiploss factor** can be included in the DMS algorithm. Due to the pressure difference between the upper and lower surface of the airfoils, downwash and a spanwise velocity component can be observed regarding a real blade. These two effects result in a decreased energy extraction at the blade tips. Therefore, the tiploss factor assumes values from 0 to 1 and is $F = 1$ in the middle of the blade and $F = 0$ at the end section. It is introduced into both, actuator disk theory and blade element method by altering the angle of attack α, the relative velocity W, the lift and drag coefficients c_L and c_D and by considering the downwind induced angle of attack α_i.

Another effect of the spanwise velocity component that has not yet been taken into consideration is the vertical streamtube expansion that increases the streamtube crosssection.

The primary DMS model proposed constant interference factors for a whole rotor half. The differing azimuthal blade position and hence altering energy extraction are unaccounted for. To correct this simplification, **variable interference factors** can be selected. Thus, the iteration is executed for each azimuthal position in particular.

Other secondary effects like the influence of the tower wake, struts, guy ropes and dynamic stall are neclegted to date.

7.6 Simulation results

7.6.1 Data storage and visualization

Local and global variables

In general, there are local and global variables. Local variables depend either on the local position such as height, radius or azimuthal angle whereas global variables only depend on the simulated tipspeed ratio or windspeed and describe the whole rotor.

Rotor and turbine variables

Rotor variables describe the rotor in a dimensionless manner. Turbine variables whatsoever include turbine specific information from the turbine data in a turbine simulation.

Plots

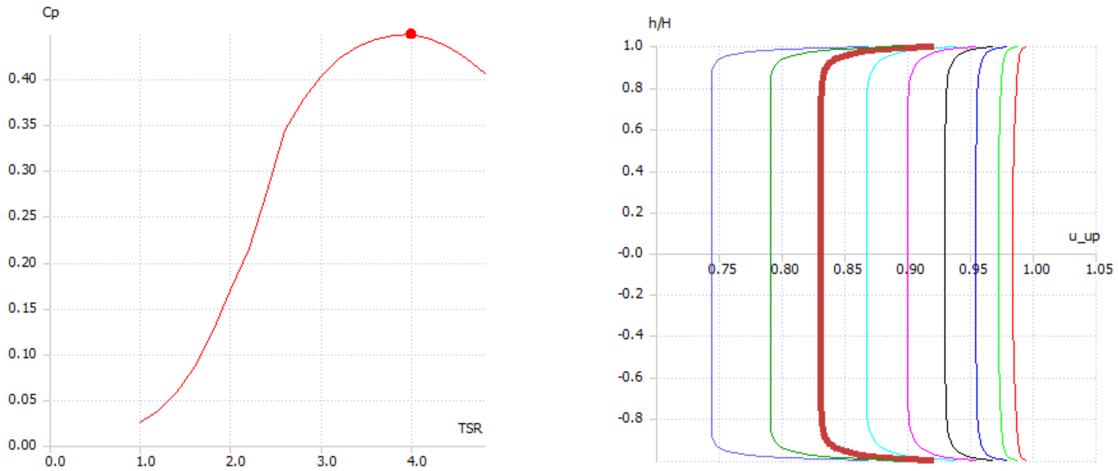


Figure 7.10: Rotor Graph: $C_P(\lambda)$ and Blade Graph: $u_{\mathrm{up}}(h)$

Three different graph types are available in the QBlade VAWT module: rotor graph, blade graph and azimuthal graph. The first one is destined for global and the other ones for local simulation data. They can be switched by the appropriate selection in the context menu that appears after a right click on the plot.

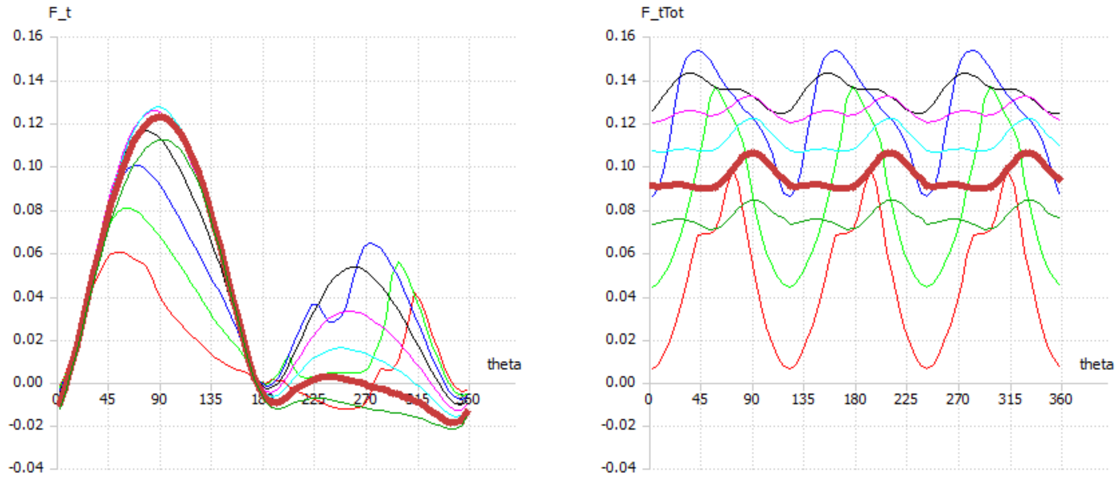


Figure 7.11: Azimuthal Graph: blade tangential force coefficient (single blade) and rotor tangential force coefficient (all blades, here: 3)

7.6.2 Variable listings

The plotted variables are selected by double clicking on the graph. After a **rotor simulation**, the following variables are available:

1. global:
   - power coefficient C_P
   - upwind power coefficient C_{P1}
   - downwind power coefficient C_{P2}
   - torque coefficient C_M
   - upwind torque coefficient C_{M1}
   - downwind torque coefficient C_{M2}
   - power coefficient based on tipspeed K_P
   - tipspeed ratio TSR
   - inverse tipspeed ratio $1/TSR$

2. local:
    - height-dependent:
        - relative height $\frac{h}{H}$
        - relative radius $\frac{r}{R}$
        - chord c
        - inclination angle δ
        - upwind interference factor u_{up}
        - downwind interference factor u_{dw}
        - upwind induction factor a_{up}
        - downwind induction factor a_{dw}
        - inflow velocity V_{∞}
        - upwind induced velocity V_{up}
        - equilibrium induced velocity V_{eq}
        - downwind induced velocity V_{dw}
        - wake velocity V_{wake}
        - local upwind tipspeed ratio TSR_{up}
        - local downwind tipspeed ratio TSR_{dw}
        - swept area S
        - upwind tiploss factor F_{up}
        - downwind tiploss factor F_{dw}
        - upwind iterations
        - downwind iterations

- azimuthal:
  - azimuthal angle θ
  - iterations
  - interference factor u
  - induced velocity V
  - relative velocity W
  - Reynolds number Re
  - tiploss factor F
  - angle of attack α
  - lift coefficient C_L
  - drag coefficient C_D
  - lift to drag ratio $\frac{L}{D}$
  - normal force coefficient C_n
  - tangential force coefficient C_t
  - blade tangential force coefficient CF_t
  - blade lengthwise force coefficient CF_x
  - blade crosswise force coefficient CF_y
  - rotor tangential force coefficient $CF_{t,\text{tot}}$
  - rotor lengthwise force coefficient $CF_{x,\text{tot}}$
  - rotor crosswise force coefficient $CF_{y,\text{tot}}$

In a turbine simulation, the dimensionsless height and radius are replaced. In addition to the other above mentioned variables, a turbine simulation provides:

1. global:
   - power $P[\mathrm{W}]$
   - torque $T[\mathrm{Nm}]$
   - windspeed $V[\mathrm{m/s}]$
   - angular speed $\omega[\mathrm{rpm}]$
   - power including losses $P_{\mathrm{loss}}[\mathrm{W}]$
   - power coefficient including losses $C_{P,loss}$
   - WEIBULL probability h_w
   - WEIBULL probability · windspeed^3 $h_w \cdot V^3[\mathrm{m^3/s^3}]$
2. local:
   - height-dependent:
     - height $h[\mathrm{m}]$
     - radius $r[\mathrm{m}]$
   - azimuthal:
     - blade torque T
     - blade lengthwise force $F_{x,\mathrm{tot}}$
     - blade crosswise force $F_{y,\mathrm{tot}}$
     - rotor torque T_{tot}
     - rotor lengthwise force $F_{x,\mathrm{tot}}$
     - rotor crosswise force $F_{y,\mathrm{tot}}$

For constant interference factors, the iteration gives one value each for the upwind and downwind zone for all height-dependent "upwind" and "downwind" data (u, a, TSR, F, iterations etc.). For variable interference factors, u varies over the azimuthal angle. Consequently, all "upwind" or "downwind" height-dependent variables are averaged values for the respective zone at the chosen height position. The azimuthal data is stored for every simulated height position and therefore always height-dependent, too.

The blade forces and force coefficients for a single blade result from an integration over the whole blade length. The rotor forces and force coefficients for all rotor blades depend on the number of blades and their resulting azimuthal position. The tangential force has always the same direction as the blade flight path. The direction of the lengthwise force is the inflow direction whereas the crosswise force is always perpendicular to the lengthwise force.

Bibliography

[1] BMU *Kurzinfo Windenergie*, 2010 [online], Available from: *http://www.erneuerbare-energien.de/inhalt/4642/* [Accessed 26 May 2010]

[2] DEPERROIS, A.: *XFLR5 Analysis of foils and wings operating at low reynolds numbers*, 2009 [online], Availiable from: *http://xflr5.sourceforge.net/xflr5.htm* [Accessed 19 February 2010]

[3] DRELA, M.; YOUNGREN, H.: *XFOIL 6.94 User Guide*, MIT Aero & Astro, 2001

[4] GASCH, R.; TWELE, J.: *Windkraftanlagen Grundlagen, Entwurf, Planung und Betrieb*, Teubner, Wiesbaden, 2007

[5] HANSEN, Martin O. L.: *Aerodynamics of Wind Turbines*. Earthscan, London, 2nd Edition, 2008

[6] VAN LANGEN, P.J.: *Blade Optimization Tool User Manual*, ECN-C-06-006, 2006

[7] GLAUERT, Hermann: Airplane Propellers. In: DURAND, William, F.: *Aerodynamic theory* 4. Springer, Berlin, 1935

[8] MAHERI, A.; NOROOZI, S.; TOOMER, C.; VONNEY, J.: *Damping the fluctuating behavior and improving the convergence rate of the axial induction factor in the BEMT-based aerodynamic codes*, University of West England BS16 1QJ, Bristol, 2006

[9] MIKKELSEN, R.: *Actuator Disk Methods Applied to Wind Turbines*, Dissertation MEK-FM-PHD 2003-02, Technical University of Denmark, 2003

[10] MONTGOMERIE, B: *Methods for Root Effects, Tip Effects and Extending the Angle of Attack Range to +-100deg, with Application to Aerodynamics for Blades on Wind Turbines and Propellers*, FOI Swedish Defence Research Agency, Scientific Report FOI-R-1035-SE, 2004

[11] VITERNA, L.A.; CORRIGAN, R.D.: *Fixed Pitch Rotor Performance of Large Hor-*

izontal Axis Wind Turbines, NASA Lewis Research Center, Cleveland, Ohio, 1982

[12] PECHLIVANOGLOU, G.: *The Effect of Distributed Roughness on the Power Performance of Wind Turbines*, GT2010-23512, Berlin University of Technology, Berlin, 2010

[13] SHEN, W.Z.; MIKKELSEN, R.; SORENSEN, J.N.; BAK, C.: *Tip Loss Corrections for Wind Turbine Computations*. Wind Energy 2005. Wiley, 2005

[14] TANGLER, J.: *The Nebulous Art of using Wind-Tunnel Airfoil Data for Predicting Rotor Performance*, NREL/CP-500-31243, National Renewable Energy Laboratory, Golden, Colorado, 2002

[15] BLACKWELL, B.F.: *The Vertical-Axis Wind Turbine "'How It Works"'*, SLA-74-0160, Sandia Laboratories, Albuquerque, New Mexico, April 1974

[16] PARASCHIVOIU, I.: *Wind Turbine Design – With Emphasis on Darrieus Concept*. Presses Internationales Polytechnique, Canada, 2002

[17] SNEL, H.; HOUWKING, R.;VAN BUSSEL, G.J.W.;BRUINING, A.: *Sectional Prediction of 3D Effects for Stalled Flow on Rotating Blades and Comparison with Measurements*. Proc. European Community Wind Energy Conference, H.S. Stevens and Associates, LÃ¼beck-TravemÃ¼nde,1993

[18] HERNANDEZ, J.; CRESPO, A.: *Aerodynamics Calculation of the Performance of Horizontal Axis Wind Turbines and Comparison with Experimental Results*. Wind Eng., 11(4), pp. 177-187, 1987

[19] ZHANG, J.H.: *Numerical modeling of a vertical axis wind turbine (VAWT)*. MEK-FM-EP 2004-11, Technical University of Denmark, 2004

[20] FUGLSANG, P.; ANTONIOU, I.; SORENSEN, N.; MADSEN, H.A.: *Validation of a wind tunnel testing facility for blade surface pressure measurements*, Riso National Laboratory, Denmark, 1998