QBlade Guidelines

v0.5

David Marten

May 31, 2012

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

Contents

1 Introduction **4**
- 1.1 The BEM in the wind turbine industry 4
- 1.2 The software project 5

2 Software implementation **7**
- 2.1 Code limitations 7
- 2.2 Code structure 7
- 2.3 The QBlade module in XFOIL 9
- 2.4 The blade design and optimization sub-module 10
- 2.5 360° polar extrapolation sub-module 13
- 2.6 The rotor simulation sub-module 14
- 2.7 The turbine definition and simulation sub-module 15

3 How to create a rotor simulation **19**

4 Simulation parameters and corrections **26**
- 4.1 Number of elements 27
- 4.2 Epsilon 27
- 4.3 Maximum number of iterations 28
- 4.4 Relaxation factor 28
- 4.5 Density 30
- 4.6 Corrections to the simulation 30
- 4.7 Foil interpolation 31

5 Simulation results **33**
- 5.1 Rotor simulation variables 34
- 5.2 Turbine simulation variables 35
- 5.3 General validity checks on simulation results 36
- 5.4 Validation of the results against a reference BEM code 36
- 5.5 Global variables 38
- 5.6 Local variables 39
- 5.7 Sensitivity analysis 41

Bibliography **47**

1 Introduction

Solving the worlds energy problem 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 sited 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 The BEM in the wind turbine industry

The blade design methods for the wind turbines originate from the aircraft design industry and apply the same techniques. But because the flow conditions that a turbine blade experiences are quite different to those affecting a plane a lot of the assumptions made in flight aerodynamics cant't be applied to the complex flow field around a wind turbine. This complex flow field is unsteady, incompressible, three dimensional, turbulent and often separated. The aerodynamics of a Wind Turbine are influenced from the far field conditions, far up and downstream of the rotor and at the same time 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. Conducting a full CFD analysis that fulfills these requirements and accounts for all these effects, is very time consuming and expensive. An alternative to CFD simulations are the vortex methods, with the limitation that they can not model viscous behavior since they are based on potential flow theory. That is why only [8] design and evaluation tools that

are based on the *Blade Element Momentum Method* are used in the industry. The other (CFD, RANS and vortex) methods are almost only used in research environments. The main advantage of the BEM 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, that assumes a steady state wind field. The BEM, 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 under predicts the overall performance of a Turbine and over predicts the peak power [12]. But nevertheless the blade element momentum method is widely applied in the wind turbine industry because the use of analysis techniques of lower order-accuracy greatly simplifies the turbine design. With the BEM its 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. This, 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's ability for robust analysis and low computational cost make up for the shortcomings and inaccuracies. Virtually all the modern rotors for *Horizontal Axis Wind Turbines* that exist today were designed using the Blade Element Momentum Method.

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 work 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 allows for 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 rotor-simulation. This skips the step of exporting and importing foil and geometry data between different codes, and the troubles associated with that. At the same time the integration of the BEM code into XFOIL's sophisticated GUI will make this software accessible to many more interested people than the usual command line interface software tools. The software is especially adequate for teaching, as it provides a 'hands on' feeling for HAWT 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 well as a post processor to conducted rotor simulations 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 dependency's among the aerodynamic variables themselves. In addition to that the resulting software is a very flexible and user-friendly platform for wind turbine blade design that 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 should include the following features:

- Extrapolation of XFOIL generated or imported airfoil polar data to 360° AoA
- Advanced rotor blade design and optimization, including 3D visualization, using XFOIL generated or imported airfoils
- Wind turbine definition (rotor blade, turbine control, generator type, losses...)
- Computation of rotor performance over tip speed ratio range
- Computation of wind turbine performance over wind speed range
- Annual yield computation with WEIBULL distribution
- Manual selection of BEM correction algorithms
- Manual selection of all relevant simulation parameters
- Data browsing and visualization in 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. Whether 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 default-free and there will certainly be more bugs discovered after the distribution. However, a validation against other BEM software permits some trust in the results this software provides.

2.2 Code structure

An overview of the data objects, in which the blades, turbines, polars and simulations are stored, and their relation to XFOIL can be found in Fig. 2.1

360° polar Object

A 360° polar object is created in the 360° polar extrapolation sub-module. It is defined by a name and a parent polar (the 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 that include this object are deleted.

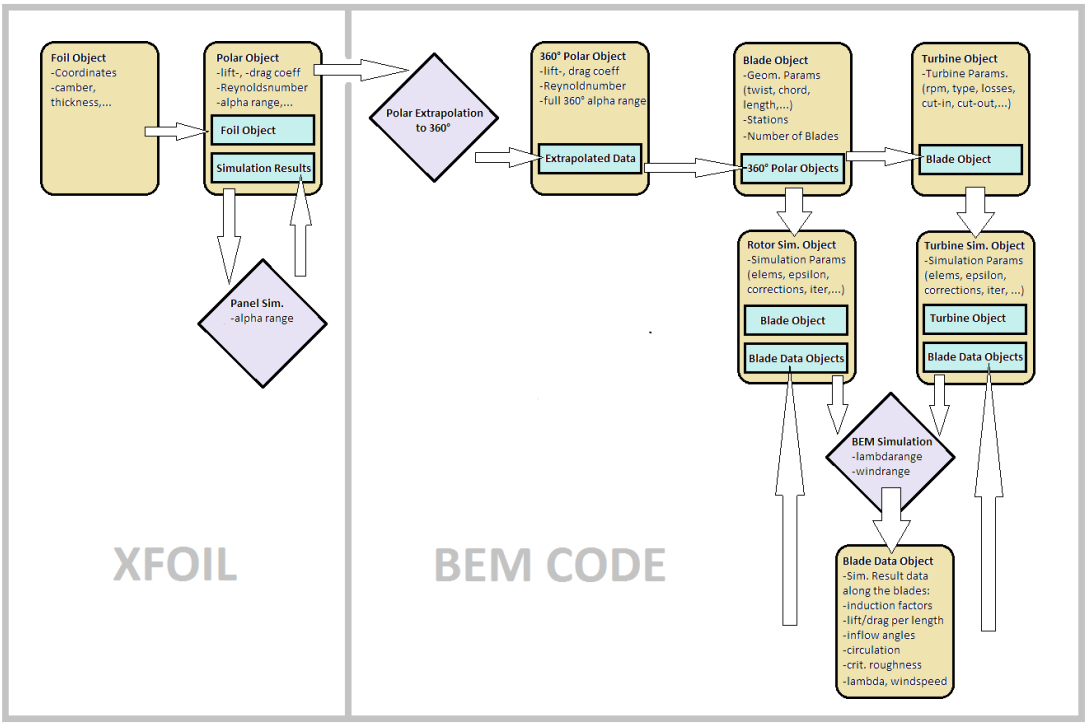

Figure 2.1: Data objects and data flow in the BEM module

Blade object

A blade object is created in the blade design and optimization submodule. It stores the geometric blade data (chord, radial position, twist and offset) 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 sub-module. It stores all turbine parameters (pitch, stall, single-, variable-, 2-step transmission,...) and a rotor-blade that needs to be associated with the turbine. If a turbine object is deleted all associated simulations deleted as well.

Simulation object

A simulation object is created when a simulation is defined. It stores the simulation parameters (max. iterations, corrections, elements, epsilon) and, if a simulation was executed, the global results. Global results are results that characterize the whole rotor ($\lambda, C_P, C_T, ...$). If a simulation object is deleted all associated blade data objects are deleted as well.

Blade data object

Blade data objects are created automatically during a simulation. They store all the data that is computed for every element along the blade. One blade data object is created for every increment of the simulation. The blade data objects require the most memory space.

2.3 The QBlade module in XFOIL

QBlade consists of four sub-modules, that will be discussed in greater detail on the following pages. These Modules are:

- Blade design and optimization
- 360° polar extrapolation
- Turbine definition and simulation
- Rotor simulation

2.4 The blade design and optimization sub-module

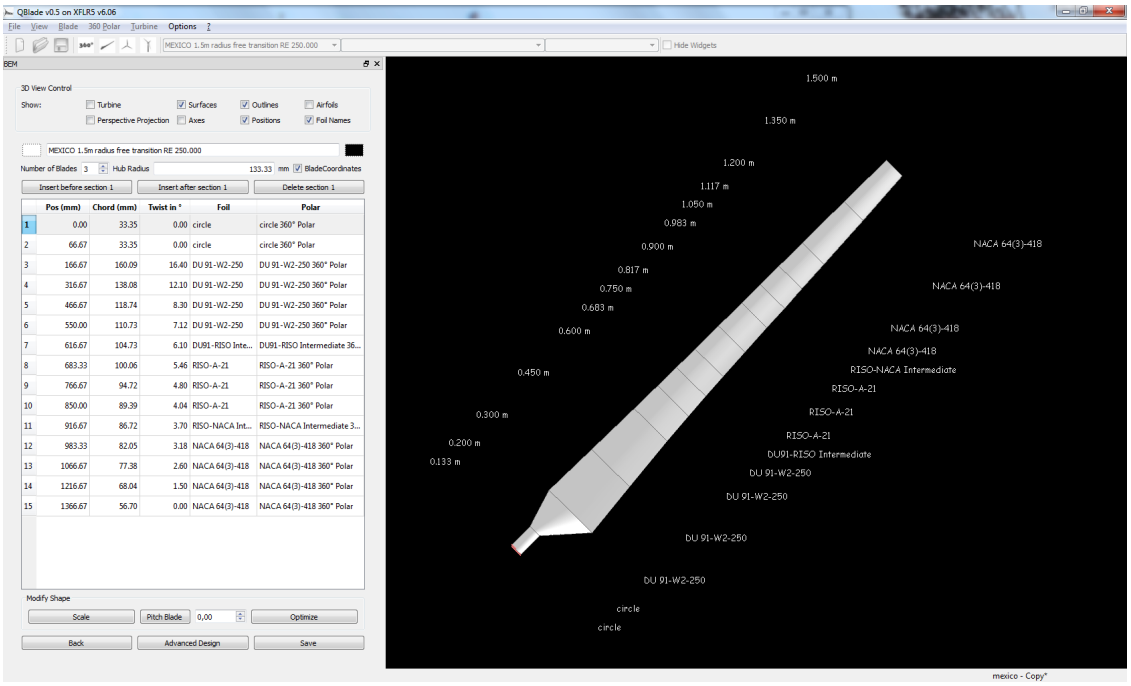

Figure 2.2: Blade design window

In the Blade design and optimization sub-module the user can create a rotor-blade. A blade consists of an arbitrary number of sections, see Fig. 2.3. 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, it can only be saved when a foil and a polar is selected at every station. Each rotor-blade has a *number of blades* associated with it that defines how many blades the rotor has. Also the /textithub 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 design can either be defined in /textitblade coordinates where the position are defined in distance from the blades root, or in /textitabsolute coordinates where the positions are defined in distance from the turbines hub center. The option *pitch blade* adds an offset to the rotorblades twist at every section of the blade.

Optimization of blade geometry

While a blade is edited, or created, and every section is fully defined the user can optimize the blade geometry with the blade optimization dialog, see Fig. 2.4.

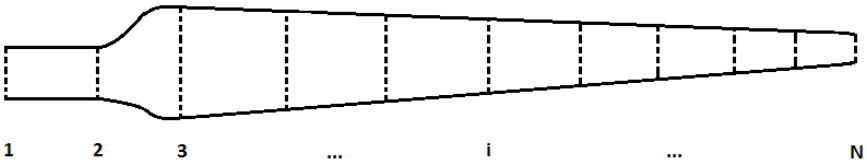

Figure 2.3: Sections along a Blade

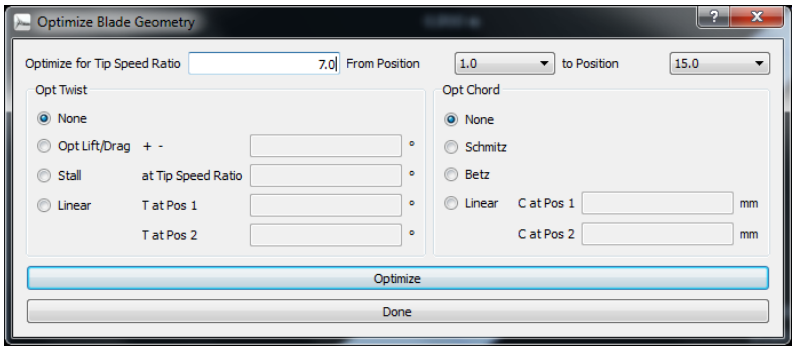

Figure 2.4: Optimization dialog

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

$$\alpha_{loc} = \tan^{-1}\left(\frac{1}{\lambda_{0,loc}}\frac{2}{3}\right) . \tag{2.1}$$

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]:

$$c(r) = \frac{16}{9}\frac{\pi R}{BC_L\lambda_0}\frac{1}{\sqrt{\left(\lambda_0\frac{r}{R}\right)^2 + \frac{4}{9}}} , \tag{2.2}$$

or SCHMITZ [4, p.202]:

$$c(r) = \frac{16\pi r}{BC_L}\sin^2\left(\frac{1}{3}\tan^{-1}\left(\frac{R}{\lambda_0 r}\right)\right) . \tag{2.3}$$

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

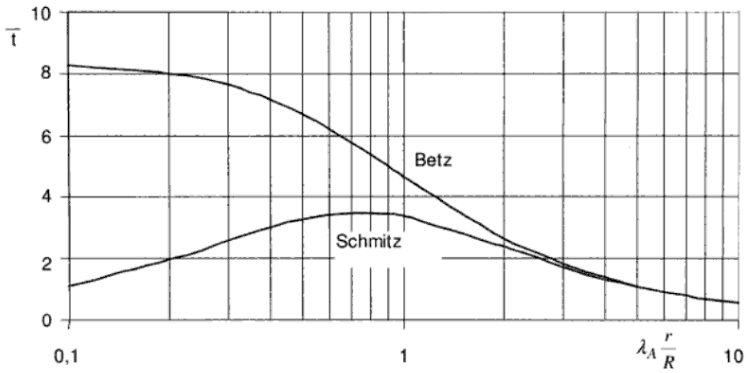

Figure 2.5: Comparison of dimensionless chord distribution after BETZ and Schmitz [4]

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.

Scaling of blade geometry

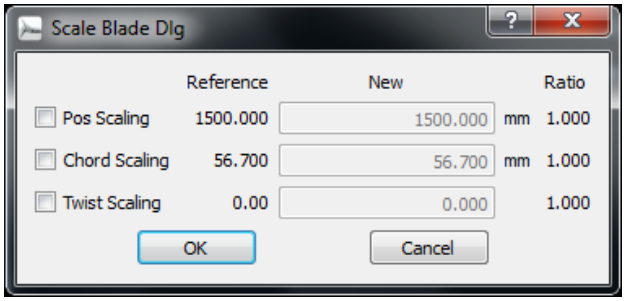

Figure 2.6: Scale blade dialog

The scale option allows to scale the blades 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.

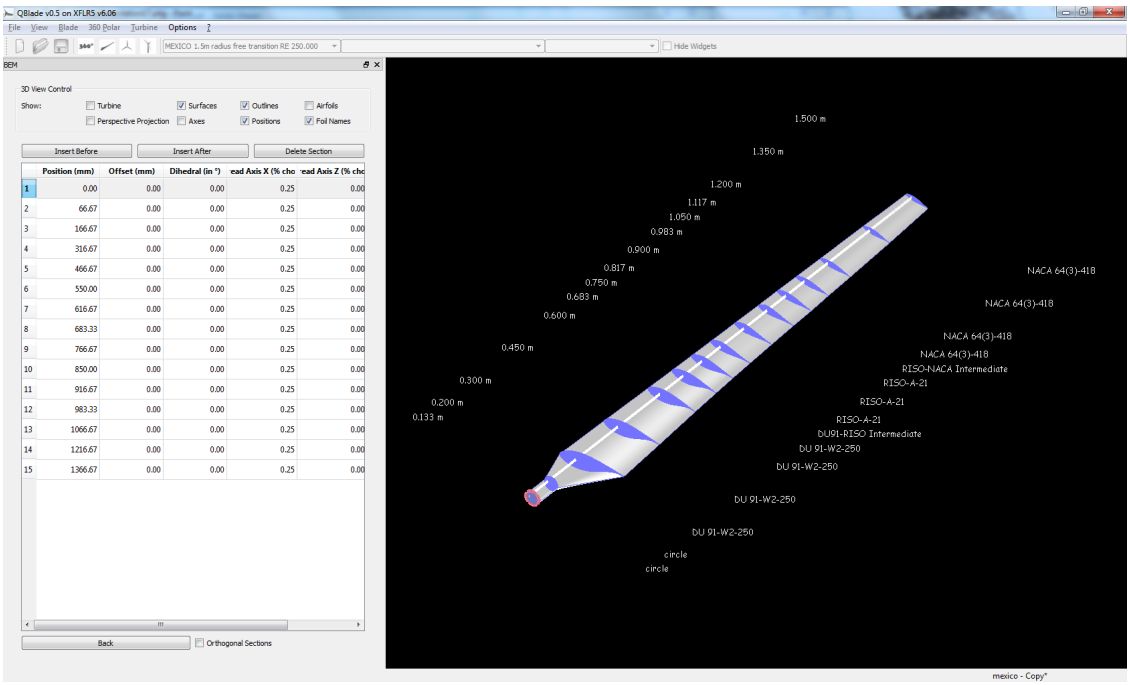

Figure 2.7: Advanced Design Window

Advanced Design

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.

2.5 360° polar extrapolation sub-module

Polars, that are a result of an XFOIL analysis or imported, can be extrapolated to the full 360° AoA range in the 360° polar extrapolation sub-module. A polar can be selected from the drop-down menu in the tool bar. When creating an extrapolation the user can set the $CD_{90,2D}$ value for the extrapolation manually. The two points C_{L1} and C_{L2}, from which the interpolation function f for

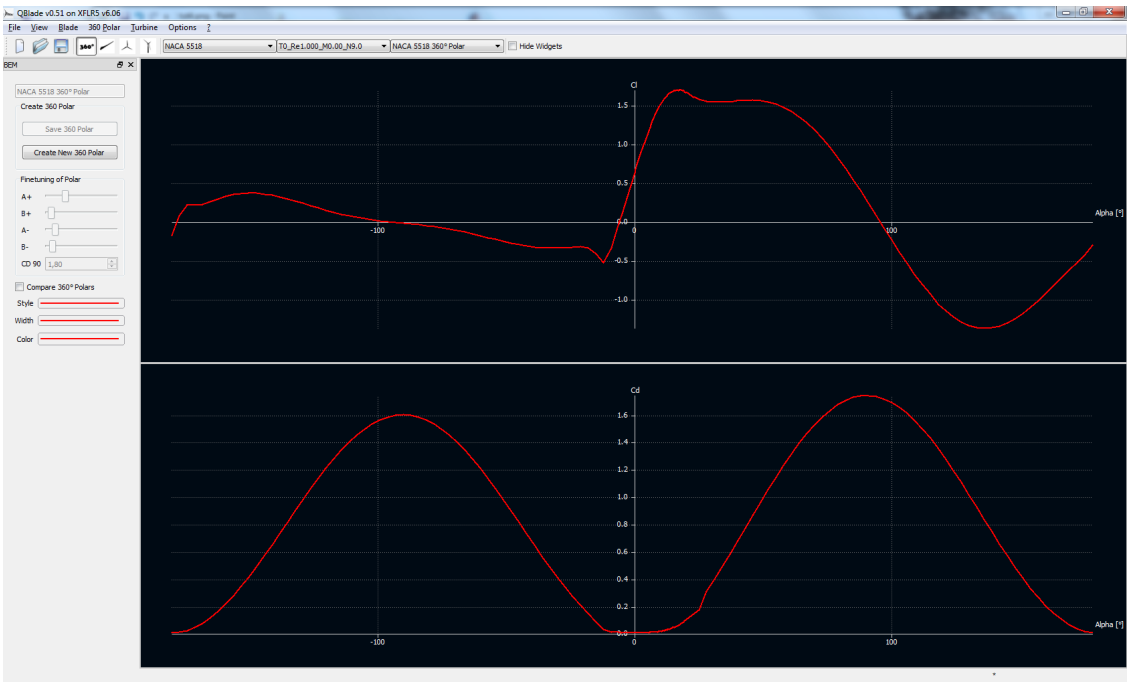

Figure 2.8: The 360° polar extrapolation sub-module

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 point for the negative extrapolation. The extrapolation is carried out exactly 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* [9]. Whenever a slider is moved, or the value for $C_{D90,2d}$ 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.

2.6 The rotor simulation sub-module

In the rotor simulation sub-module the user can commit rotor-blade simulations over a range of tip speed 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 lambda values (tip speed ratios), and the incremental lambda step for the simulation. A rotor simulation is always carried out dimensionless. The free stream velocity is assumed to be unity and the rotor radius is normalized for the

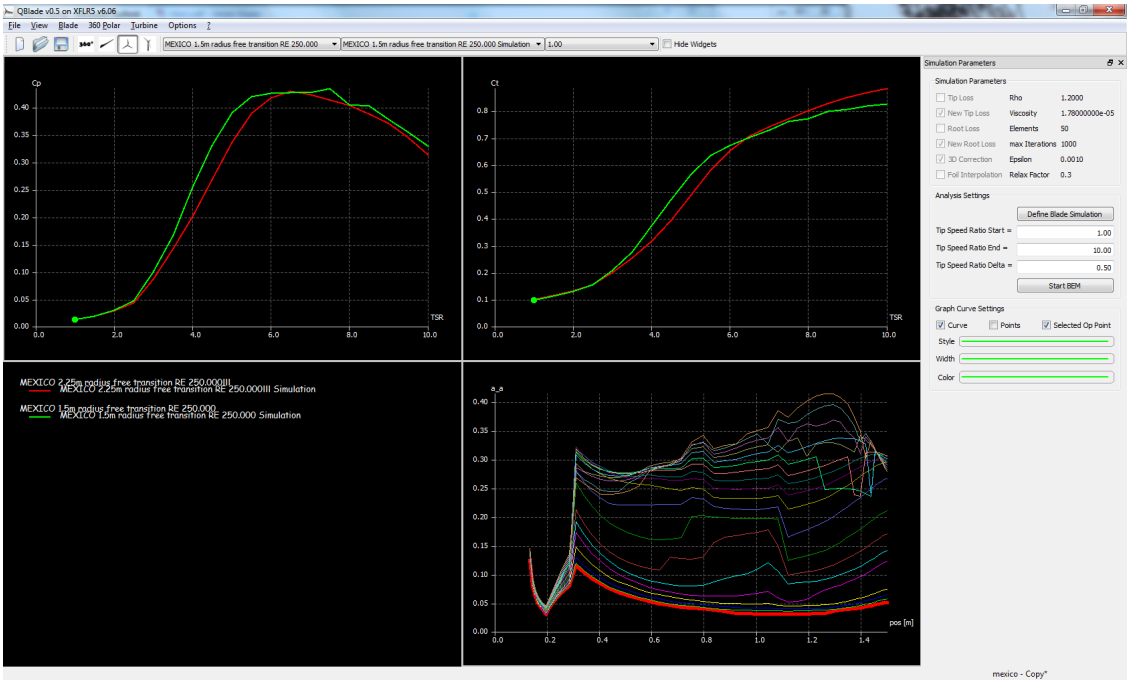

Figure 2.9: The Rotor simulation sub-module

computation. This implies that no power curve or load curves, like the bending moment, can be computed during a rotor simulation.

2.7 The turbine definition and simulation sub-module

In the turbine definition and simulation sub-module 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 wind speed , that yields the nominal power output is reached the blades are pitched to reduce the power for higher wind speeds 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 it power to the desired output and at the desired wind speed requires an

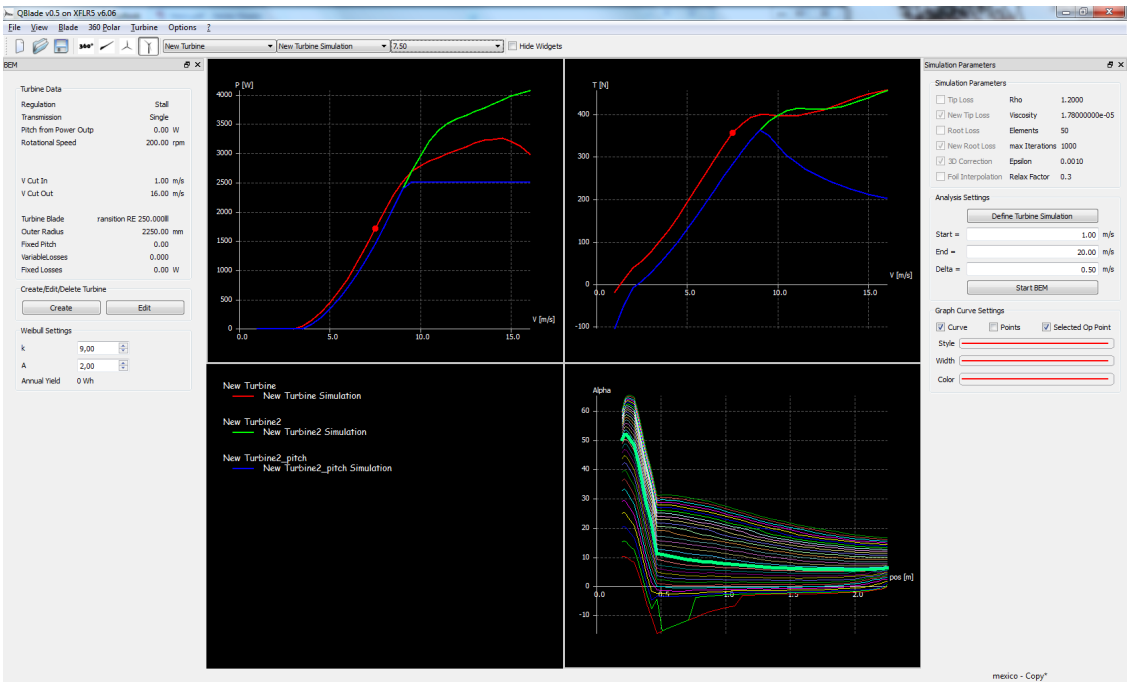

Figure 2.10: The turbine definition and simulation sub-module

iterative approach.

For a single speed transmission the user has to select only one rotational speed, in which the turbine operated over the whole range of wind speeds. For 2-step transmission two rotational speeds and a wind speed at which the transmission changes between these to rotational speeds have to be selected. A variable transmission turbine has a minimum and a maximum value for he rotational speed. Additionally the user selects a desired tip speed 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 wind speed
- Cut out wind speed
- Fixed losses
- Variable losses

Every turbine needs to have a rotor blade defined. This can be any blade that is stored in the runtime database. At the cut-in wind speed the turbine starts and at the cut-out wind speed the turbine stops operation. To account for power losses, that are not of aerodynamical nature, 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:

$$P_{out} = (1 - k_v)P_0 - P_{fixed} \ . \tag{2.4}$$

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

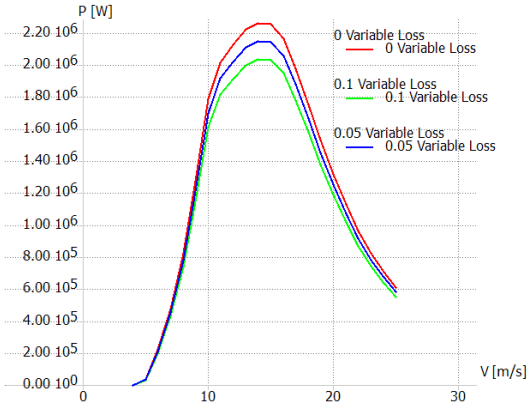

Figure 2.11: Different variable loss factors - effect on power output

If a turbine simulation has been conducted the user may calculate the annual yield of the turbine by specifying an annual wind speed distribution via the two parameters k and A of the WEIBULL distribution. The probability for a wind speed to occur is:

$$h_W(V_0) = \frac{k}{A}\left(\frac{V_0}{A}\right)^{k-1} \exp\left(-\left(\frac{V_0}{A}\right)^k\right) \ . \tag{2.5}$$

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

$$f(V_i < V_0 < V_{i+1}) = \exp\left(-\left(\frac{V_i}{A}\right)^k\right) - \exp\left(-\left(\frac{V_{i+1}}{A}\right)^k\right) \ . \tag{2.6}$$

Thus the annual energy production is calculated as:

$$AEP = \sum_{i=1}^{N-1} \frac{1}{2}\left(P(V_{i+1}) + P(V_i)\right) \cdot f(V_i < V_0 < V_{i+1}) \cdot 8760 \ . \tag{2.7}$$

A turbine simulation is carried out over a range of wind speeds, with the chosen incremental step size. Depending on the specified rotational speed of the turbine a tip-speed ratio is computed for every wind speed. Then a BEM simulation over he computed tip speeds, that is equivalent to a rotor simulation, is carried out.

3 How to create a rotor simulation

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.

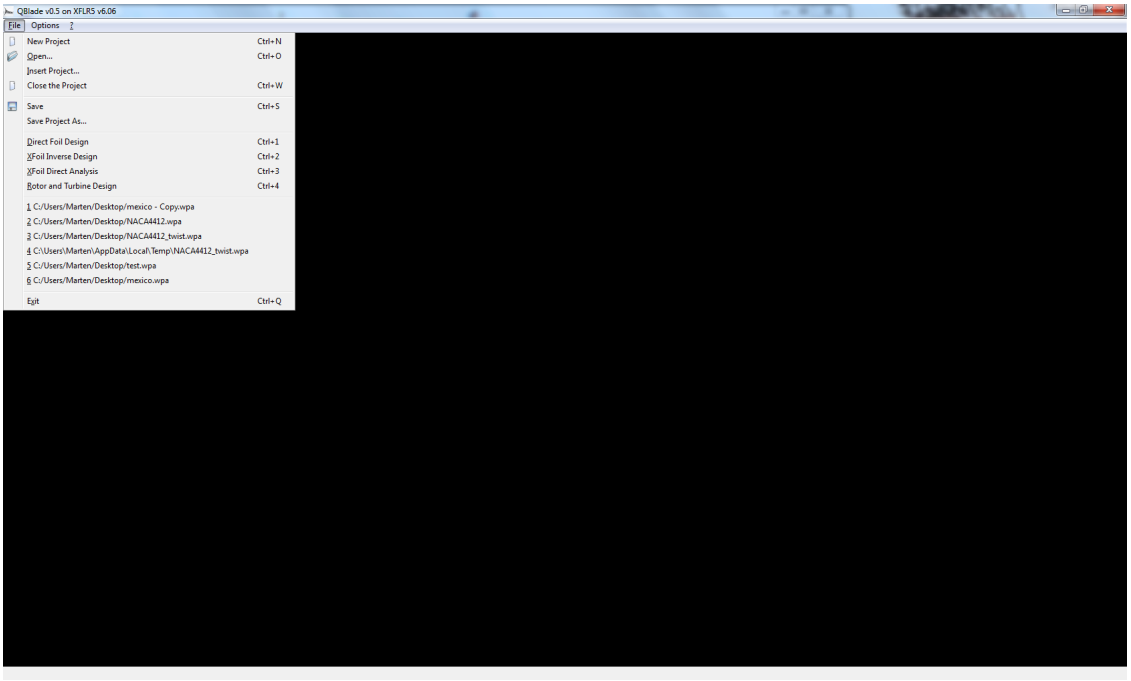

Figure 3.1: QBlade startup screen

Figure 3.1 shows the start screen of QBlade. From the *File* menu the user can select the different applications (the menu point *Rotor and Turbine Design* distinguishes QBlade from XFLR. During this tutorial the modules *Direct Foil Design*, *XFoil Direct analysis* and *Rotor and Turbine Design* are used. More information about XFOIL and XFLR5 in general can be found in [3] and [2].

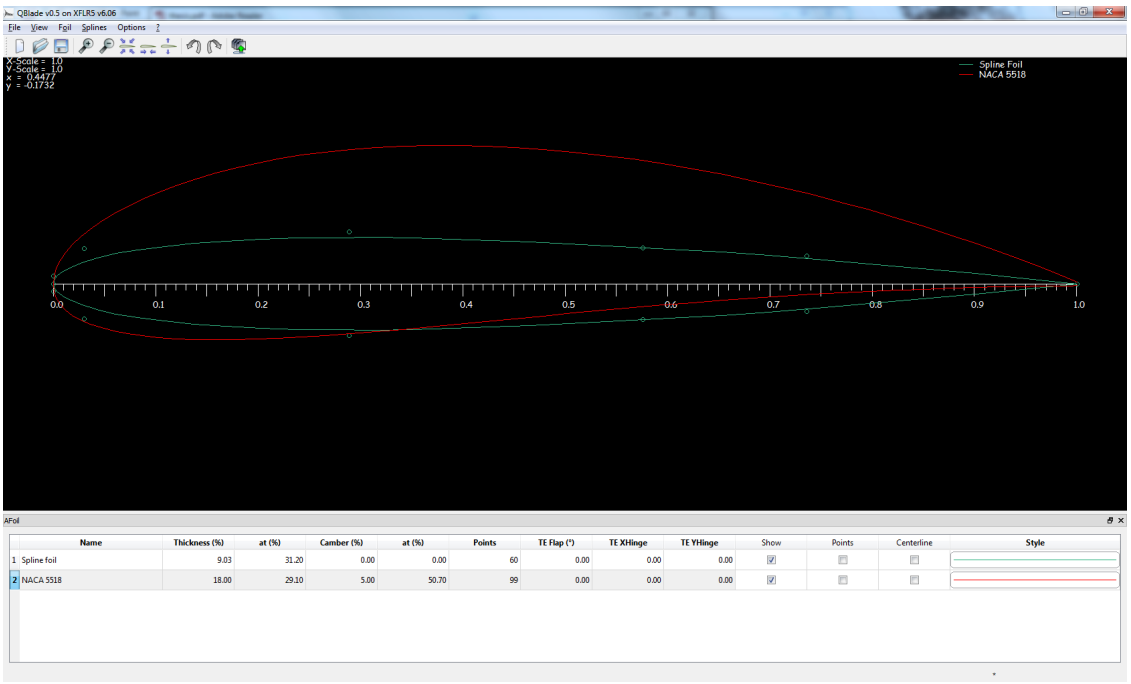

Figure 3.2: Direct Foil Design module

The introduction starts in the *Direct Foil Design* module. The first thing that needs to be done when creating a rotor is to create its airfoils. Airfoils can be created using splines, a NACA airfoil generator or via an import function in XFLR5. In this case only one airfoil, the NACA 5518 was created.

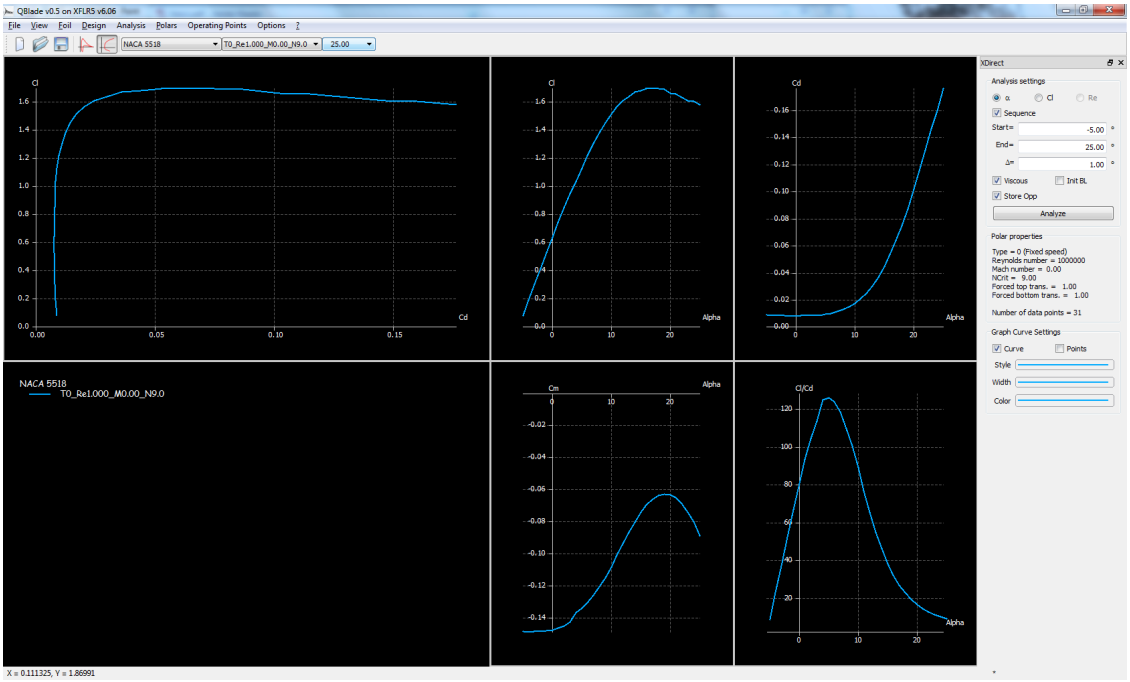

Figure 3.3: XFOIL Direct Analysis module

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, whenever importing polar data, 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.

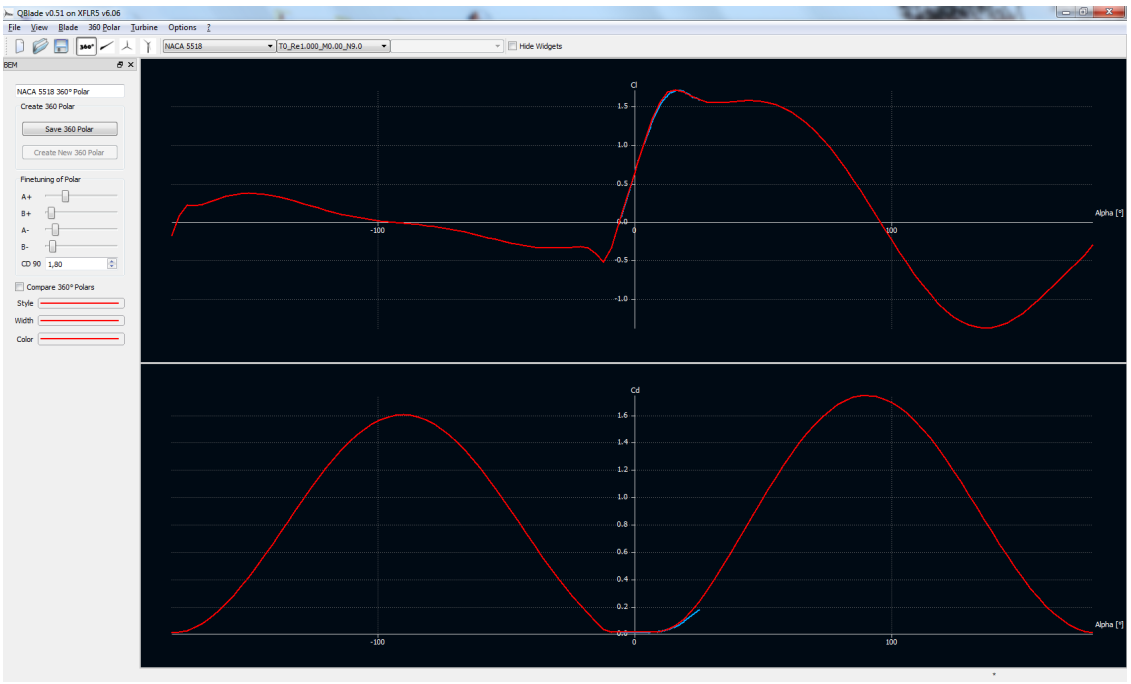

Figure 3.4: Rotor and Turbine Design module / 360° polar interpolation

To simulate a wind turbine the AoA range of the polar needs to be extrapolated to 360°, this is done in the submodule for polar extrapolation. 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 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. In the *360 Polar* menu, it is possible to automatically load a cylindrical foil. This foil has zero lift and a drag value that can be specified by the user. Cylindrical foils are used in the root region of a blade for structural reasons.

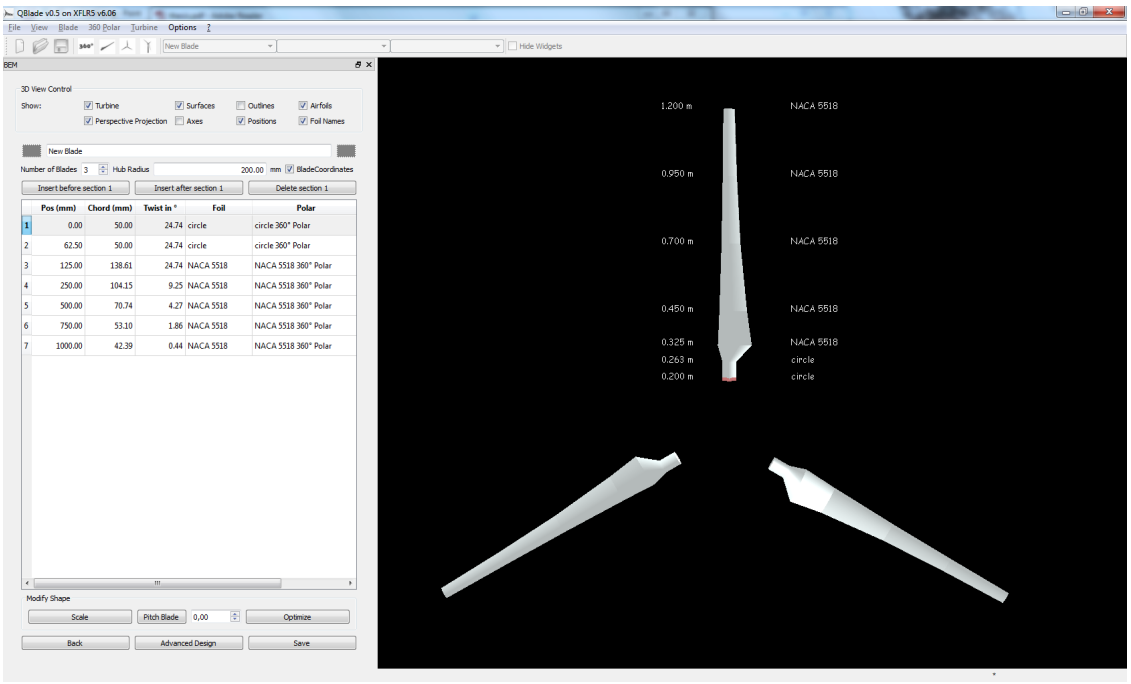

Figure 3.5: Rotor and Turbine Design module / Blade design

If one or more 360° polars have been created a blade can be designed. The *Optimize* and *Scale* buttons open the corresponding dialog windows. Stations can be added or removed, one airfoil and one polar has to be specified at every section. Additionally every blade has a "Number of Blades" defined, that specifies how of how many blades the rotor exists. When a blade is fully defined it can be saved. Instead of creating a new blade it is possible to edit stored blades. When this is done a copy of the stored blade is edited. When a blade from the database is deleted or overwritten all associated simulations and turbines are deleted as well.

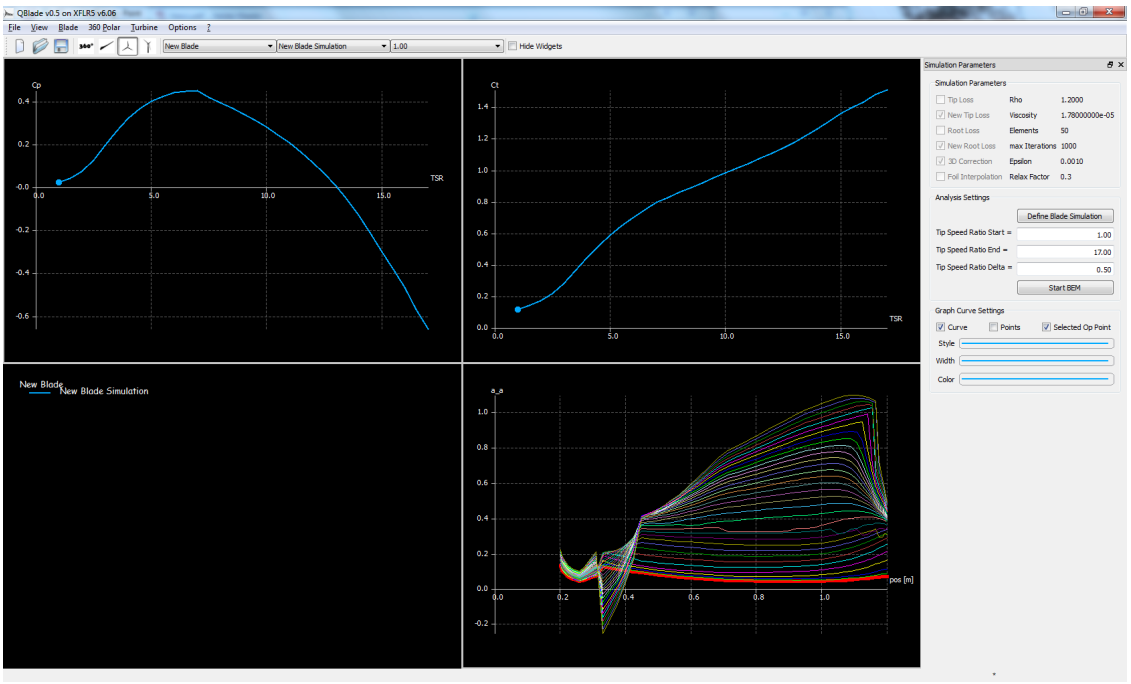

Figure 3.6: Rotor and Turbine Design module / Rotor simulation

All blades in the database can be simulated in the *Rotor Simulation* sub-module. Whenever a simulation is defined all simulation parameters need to be specified. The simulation then is conducted over the desired range of tip-speed ratios. The three graphs show the simulation results. By double-clicking on a graph the user may change the variables on the x- and y-axis. By right-clicking on a graph the user may change the graph type to display local or global variables. The local variables are always displayed for every computed lambda value. 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 variable curve of other simulations. If multiple simulations are stored in the database, the curves of global variables are always shown simultaneously.

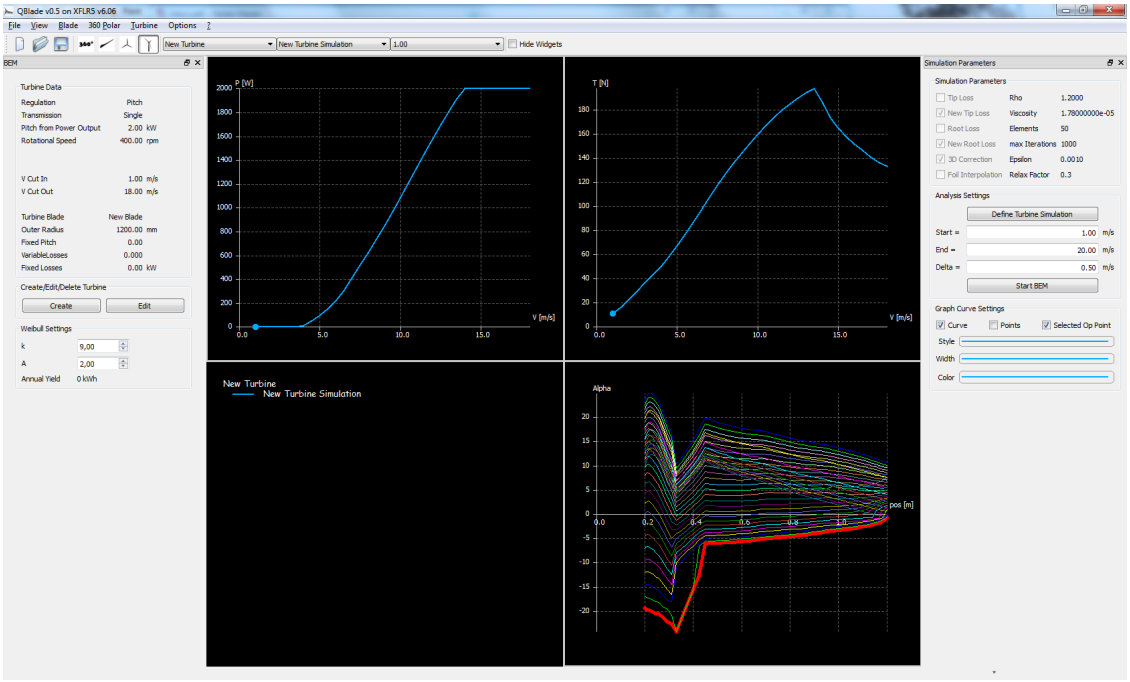

Figure 3.7: Rotor and Turbine Design module / Turbine definition and simulation

In the *Turbine definition and simulation* sub-module a turbine can be defined only if a blade is stored in the database. The user specifies all turbine parameters and saves the turbine. Then a turbine simulation, over a range of wind speeds, can be conducted. When the user sets the k and A values for the WEIBULL distribution, the annual yield is computed automatically.
In the file menu the user can save the whole conducted work, including foils, polars, blades and simulations as a `.wpa` file.

4 Simulation parameters and corrections

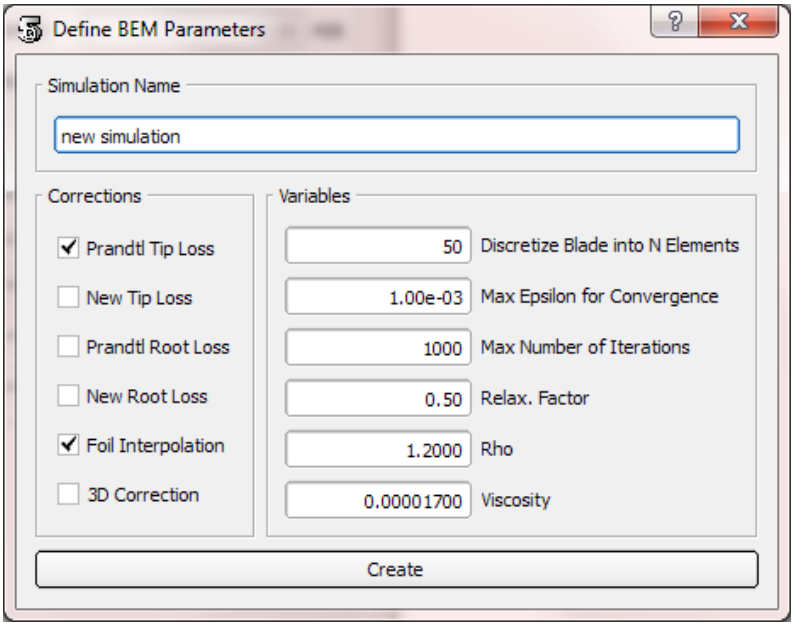

Figure 4.1: Simulation definition dialog

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

- Number of elements along blade
- Maximum epsilon for convergence
- Maximum number of iterations
- Relaxation factor
- Density
- Viscosity

4.1 Number of elements

The number of elements specifies into how many elements the blade is divided. This number is independent from the number of sections that a blade has. 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.

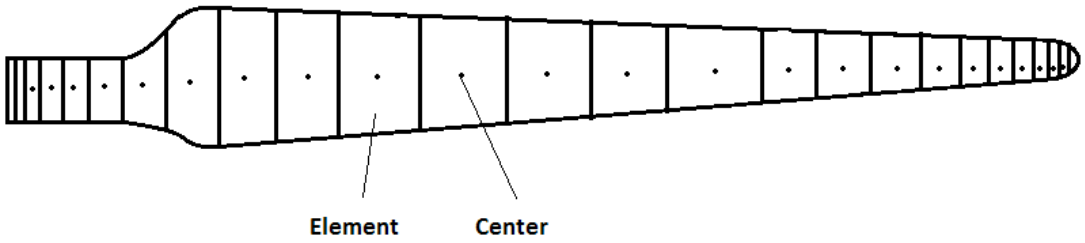

Figure 4.2: Sinusoidal spaced elements along the blade

All the variables, resulting from the BEM computation, 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 elements contribution to the total torque and thrust.

4.2 Epsilon

The epsilon value, ϵ, defines when an iteration is 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.

$$\texttt{max}(|a - a_{old}|, |a' - a'_{old}|) < \epsilon \ . \tag{4.1}$$

A recommendation for epsilon is 10^{-5}. See section *Sensitivity analysis* in the Appendix for details.

4.3 Maximum number of iterations

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

4.4 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 [7] (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 codes performance and its accuracy. In [7] MAHERI proposes to introduce a relaxation factor, ω_{relax}, to overcome these fluctuations.

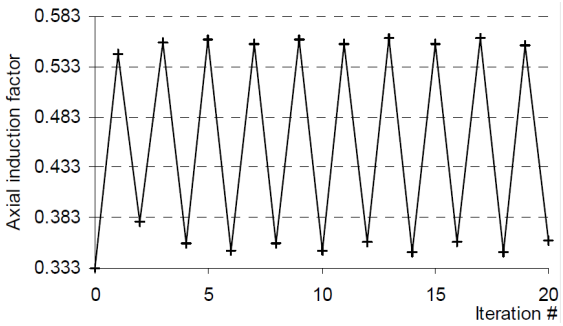

Figure 4.3: Fluctuation of the axial induction factor around the light and heavy loading state [7]

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

$$a_{k+1} = \omega_{relax} a_{k+1} + (1 - \omega_{relax}) a_k \ ; \quad 0 < \omega_{relax} < 1 \ . \tag{4.2}$$

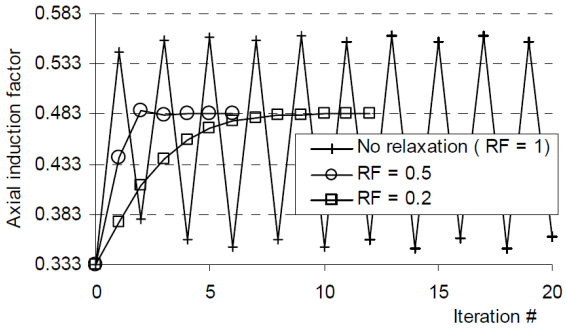

Figure 4.4: Damped fluctuation of the axial induction factor for different relaxation factors [7]

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 $\omega_{relax} = 1$ should be 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.

$$a_{k+1} = \frac{1}{4}a_{k+1} + \frac{1}{2}a_k + \frac{1}{4}a_{k-1} \, . \tag{4.3}$$

From there the iteration proceeds as normal with the desired relaxation factor and eq. 4.2 applied.

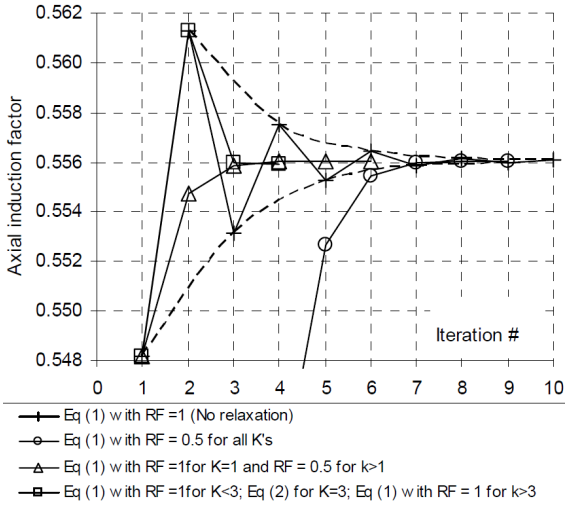

Figure 4.5: Accelerated convergence by placing the induction factor inside the neighborhood of the final result [7]

4.5 Density

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

$$P = \frac{1}{2}\rho A V_0^3 C_P \, . \tag{4.4}$$

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 tip speed ratio.

Dynamic viscosity

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

$$Re(r) = \frac{V_{rel}(r)c(r)\rho}{\mu} \, . \tag{4.5}$$

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 tip speed ratio.

4.6 Corrections to the simulation

The following correction algorithms can be selected to be included in the BEM simulation:

- PRANTL tip loss, found in [5]
- PRANTL root loss, found in [5]
- New tip loss model after SHEN et al. [11]
- New root loss model after SHEN et al. [11]

- 3D correction after Snel, found in [6]
- Foil interpolation

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

4.7 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. 4.6

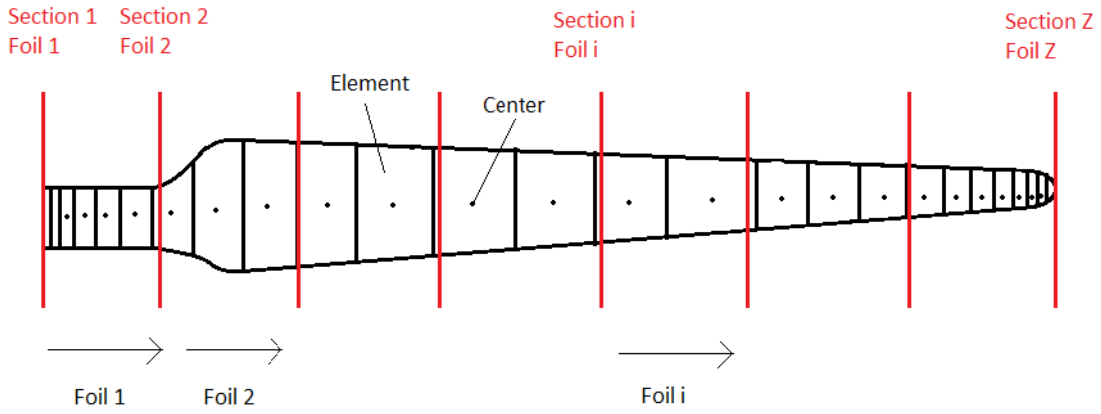

Figure 4.6: Foil distribution along the blade without interpolation

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. The equation used is:

$$\texttt{Polar}_{int} = \texttt{Polar}_2 \cdot \frac{r_{center} - r_{\texttt{sec}_1}}{r_{\texttt{sec}_2} - r_{\texttt{sec}_1}} + \texttt{Polar}_1 \cdot (1 - \frac{r - r_{\texttt{sec}_1}}{r\texttt{sec}_2 - r_{\texttt{sec}_1}}) \, . \tag{4.6}$$

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.

5 Simulation results

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 tip-speed ratio or wind speed.
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 tip-speed ratio or wind speed.

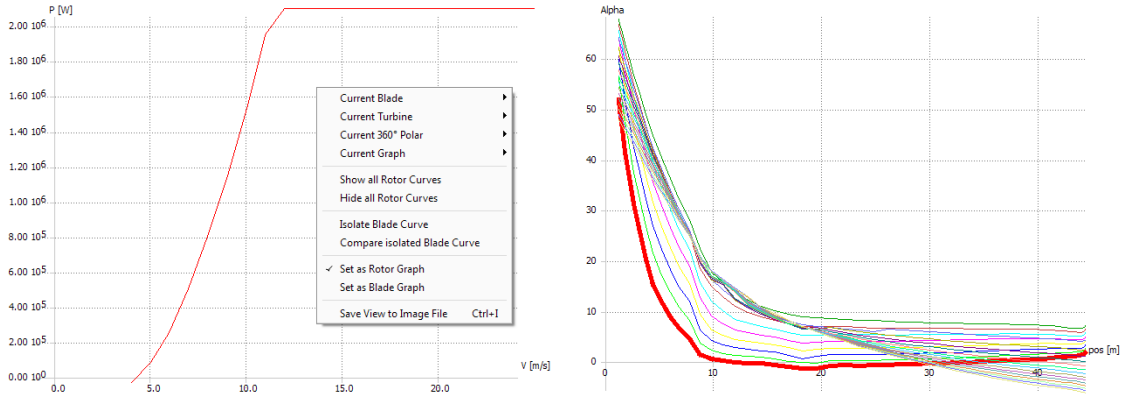

Figure 5.1: Rotor graph to the left and blade graph to the right, context menu

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.

5.1 Rotor simulation variables

As stated before all variables calculated in a rotor simulation are dimensionless. The global variables, computed during a rotor simulation are:

- Power coefficient C_P
- Thrust Coefficient C_T
- Tip-speed ratio λ
- Power coefficient based on tip-speed $K_P = \frac{C_P}{\lambda^3}$
- Inverse tip-speed ratio $\frac{1}{\lambda}$

The computed local variables of a rotor simulation are:

- Axial induction factor a
- Radial induction factor a'
- Local tip-speed 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 tip loss factor F
- Number of iterations n
- Annulus averaged axial induction factor
- Annulus averaged radial induction factor

5.2 Turbine simulation variables

In a turbine simulation these global variables are computed additionally to all the global variables from a rotor simulation:

- Power $P\,[W]$
- Thrust $T\,[N]$
- Wind speed $V_0\,\left[\frac{m}{s}\right]$
- Torque $[Nm]$
- Angular frequency $\Omega\,[\text{rpm}]$
- Pitch angle β
- WEIBULL probability h_w
- WEIBULL probability $\cdot$ wind speed^3 $h_w \cdot V_0^3$
- Root bending moment $M\,[Nm]$
- Power coefficient including losses $C_{P,loss}$

These additional blade variables are computed during a turbine simulation:

- Local REYNOLDS number $Re_{loc} = \frac{V_{rel} c \rho}{\mu}$
- Re deviation from polar simulation $Re_{loc} - Re_{polar}$
- Critical roughness $k_{critical} \approx 100 \frac{\mu}{V_{rel}}\,[mm]$ from [10]

- Resultant velocity V_{rel} $\left[\frac{m}{s}\right]$
- Tangential force per length P_T $\left[\frac{N}{m}\right]$
- Normal force per length P_N $\left[\frac{N}{m}\right]$
- Mach number Ma
- Circulation $\Gamma = \frac{1}{2} C_L V_{rel} c$ $\left[\frac{m^2}{s}\right]$

5.3 General validity checks on simulation results

Because of the assumptions made in the derivation of the BEM method, two checks need to be performed by the user to ensure the validity of the computation. It is assumed that the blade elements possess radial independence and that there is no cross flow in radial direction from one element to another. To ensure consistency with these assumptions the annulus averaged axial induction factor $F \cdot a$ and the circulation Γ have to be relatively constant along the blade. A large gradient in the averaged induction factor involves cross flow and a large gradient in circulation causes a radial dependent down wash. These checks should especially performed at wind speeds where the turbine operates at rated power and most notably in the outer part of the blade [6]. To check for the validity of the used polars, the user may check deviation of the REYNOLDS number at which the polar was computed to the local REYNOLDS number at the blade. When the difference is higher that 10^6, the polar data should be adapted to the flow conditions at the turbine.

5.4 Validation of the results against a reference BEM code

In this section the results, computed with the QBlade will be compared to the results produced by a reference inhouse BEM code, based on an EXCEL spreadsheet. Utilizing a classical BEM algorithm with optional PRANDTL tip and hub loss corrections the reference software is suitable to validate the BEM algorithm that has been implemented in XFOIL. The reference software utilizes a database of precalculated polars. In order to compare the results of the two BEM algorithms, the blade parameters and the airfoil polars have to be exactly the

same. For this comparison the polar data has been exported from the reference code and imported into QBlade. Both simulations were conducted with only the PRANDTL tip and hub loss corrections turned on. Foil interpolation was switched on in the new BEM code. In the reference software an element is always bounded by two sections, thus the number of elements is fixed.
Compared Turbine:
Regulation: Stall;
Transmission: Single;
Rotational speed: 15.47 rpm;
Cut in wind speed: 3.5 $\frac{m}{s}$, Cut out wind speed: 25.00 $\frac{m}{s}$;
Variable losses: 0.06, Fixed losses: 69000 W.
The turbine has a 3 bladed rotor:

Radius [m]	Chord [m]	Twist [°]	Foil
1.2	2.19	12	Cylinder
2.7	2.19	12	Cylinder
4.7	2.49	12	TRANSIT
7.7	3.09	12	TRANSIT
9.2	3.2	11.009	DU 00-W-401
12.2	3.13	8.874	DU 00-W-350
15.2	2.97	7.253	DU 97-W-300
18.2	2.775	5.924	DU 91-W2-250
21.2	2.533	4.724	NACA 63(4)-421
24.2	2.285	3.636	NACA 63(4)-421
27.2	2.073	2.71	NACA 63(4)-421
30.2	1.863	1.977	NACA 63(3)-418
33.2	1.636	1.415	NACA 63(3)-418
36.2	1.414	0.911	NACA 63(3)-418
39.2	1.217	0.466	NACA 63(3)-418
42.2	0.994	0.037	NACA 63(3)-418
43.2	0.84	0	NACA 63(3)-418
44	0.52	0	NACA 63(3)-418
44.25	0.3	0	NACA 63(3)-418
44.45	0.05	0	NACA 63(3)-418

Figure 5.2: The compared rotor blade

5.5 Global variables

For the WEIBULL distribution with $k = 2$ and $A = 9.591\frac{m}{s}$ the predicted annual yield of both programms is:

- reference code: 7671 MWh
- QBlade: 7495 MWh

This important result shows good agreement and has a relative difference of only 2.3%.

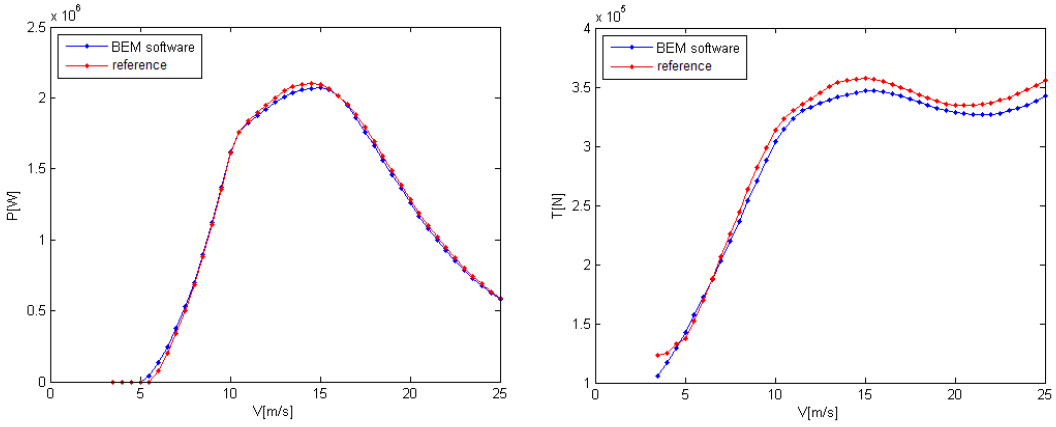

Figure 5.3: Power and thrust over wind speed

In general the power curves of the reference software and QBlade match well (Fig. 5.3). The reference software predicts the maximum power at 14.5 $\frac{m}{s}$, QBlade at 15 $\frac{m}{s}$. A larger discrepancy can be found in the curves for thrust. Here the reference software predicts a thrust thats approximately 5% higher than predicted by QBlade, however the overall tendency of the curves is the same.

The reference software predicts a lower C_P for low wind speeds (or high tip-speed ratios) as QBlade (Fig. 5.4). Even though this difference is quite large, its effect on the power curve is little, since the low wind speeds do not contribute much to the power output of the turbine. In contrast smaller differences between

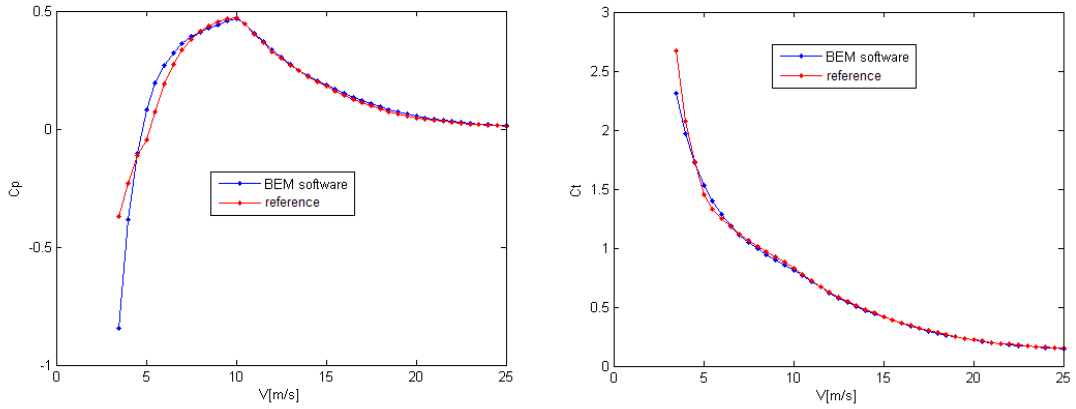

Figure 5.4: C_P and C_T over windspeed

the two C_P curves, in the region of high wind speeds are amplified in the power curve. In general there is a good matching between the computed C_P and C_T curves.

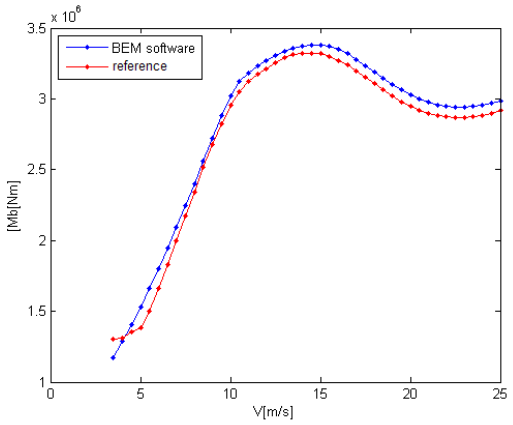

Figure 5.5: Bending moment over wind speed

The characteristics of the bending moment curves match (Fig. 5.5). QBlade predicts a moment that is about 3% higher than the moment computed by the reference software.

5.6 Local variables

The comparison of the distribution for local variables, at a wind speed $V_0 = 7.5\frac{m}{s}$, is shown on the following pages.

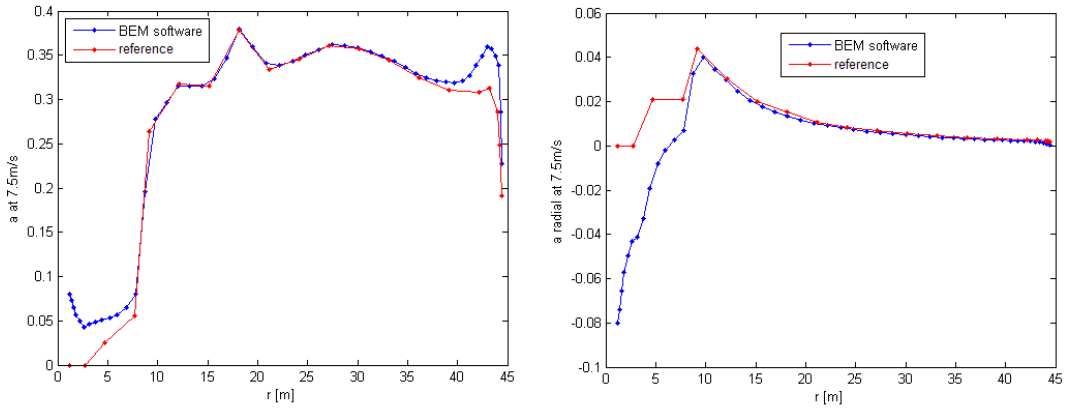

Figure 5.6: Axial and radial induction factor over radial position

The curves for the axial induction factor match everywhere except for the tip and hub region (Fig. 5.6). This might be due to a different implementation of the tip loss correction. The curves for the radial induction show a large difference in the hub region. The reference software might limit the radial induction factor to a value that cant be smaller than zero. In QBlade no such limit is implemented.

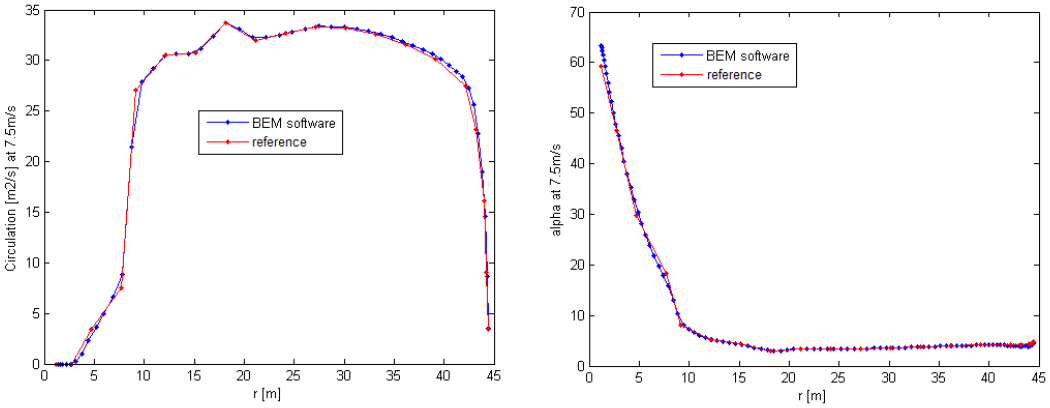

Figure 5.7: Γ and α over radial position

The curves for circulation Γ and AoA α both hardly show any differences (fig. 5.9).

Because the AoA curves were matching and the polar data in both codes are the same the C_L and C_D curves show good agreement (Fig. 5.7). The peak in C_D curve computed by QBlade is due to the higher radial resolution. In the reference software the number of elements is defined by the number of sections minus one. In QBlade the user can define an arbitrary number of elements for any blade.

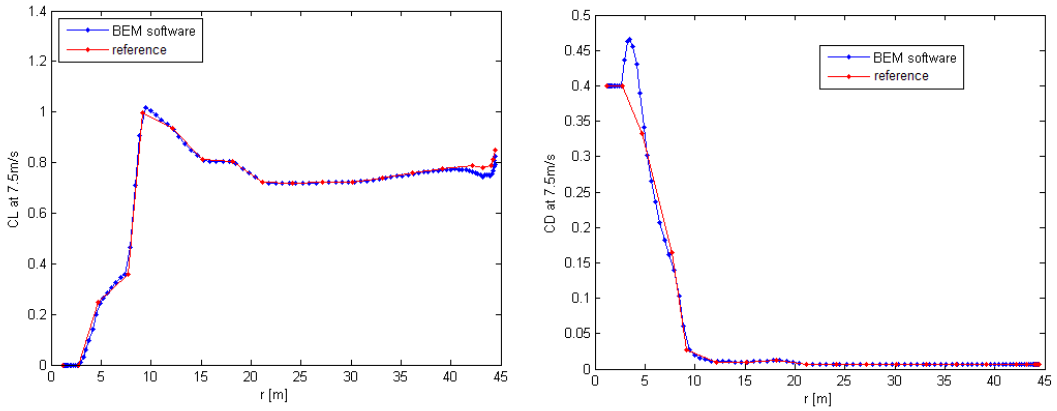

Figure 5.8: C_L and C_D over radial position

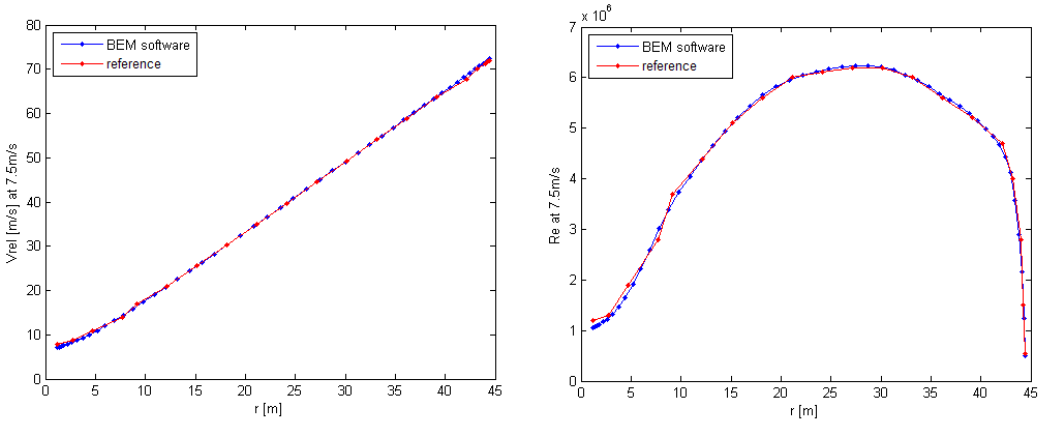

Figure 5.9: V_{rel} and Re over radial position

The V_{rel} and Re curves show very good agreement (Fig. 5.8). Generally all the compared curves have the same characteristics. There are slight differences for some variables but the reasons behind it have not been investigated. In any way the BEM method is a lower order accuracy analysis technique and draws its usefulness from predicting overall performance characteristics and only load and power approximations. The validation of QBlade can therefore be recorded as successful.

5.7 Sensitivity analysis

In this section the influence of the different simulation parameters and corrections on a BEM simulation are pointed out. All graphs presented in this section

are screen shots from QBlade. The rotor with which this analysis was conducted has three blades:

Radius [m]	Chord [m]	Twist [°]	Foil
3	2	0	Cylinder
5	2	0	Cylinder
7	4	14.62	NACA 4450
9	4	14.62	NACA 4450
11	3	6.89	NACA 4425
15	1.83	4	NACA 4420
19	1.4	1.73	NACA 4418
23	1.24	1.27	NACA 4418
27	1.15	1.17	NACA 4414
31	1.09	0.37	NACA 4414
33.5	0.8	0.37	NACA 4414
34.5	0.7	0.37	NACA 4414
34.8	0.5	0.37	NACA 4414
35	0.1	0	NACA 4414

Foil Interpolation

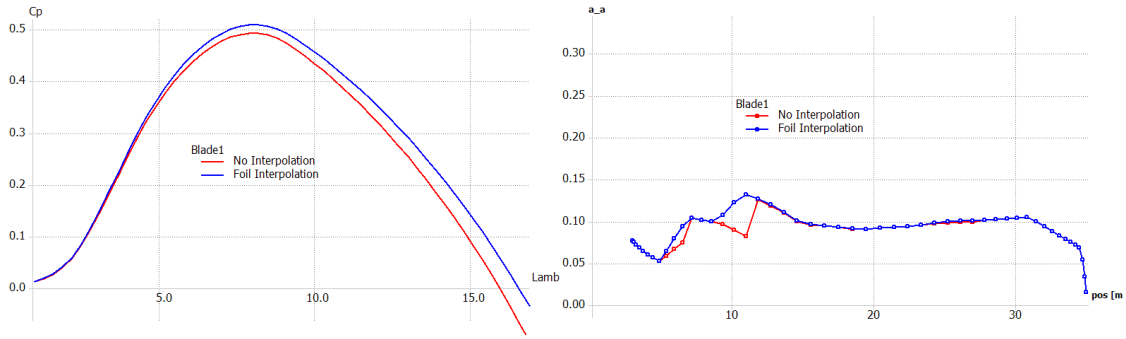

Figure 5.10: Effect of Foil Interpolation

Figure 5.10 shows the effect of Foil interpolation on a blade that was simulated with 50 elements. The left graph shows the effect on the C_P curve. The C_P value

is significantly higher with the interpolation turned on. This is because the whole root region from $r = 3m$ to $r = 7m$ is treated as being fully cylindric, with zero lift without the interpolation. When the interpolation is turned on, the root region contributes to the total torque of the rotor because from the interpolation between a cylindric polar with a NACA 4450 polar results a positive lift for elements between 5-7m. This is more realistic since the real geometry is not fully cylindric between 5-7m and should have a positive lift.
The distribution of the axial induction factor along the blade can be found with and without interpolation in the right graph. In the interpolated curve the "jumps" of the axial induction factor are smoothed to a more continuous behavior. For all the following investigations the foil interpolation is turned on.

Epsilon

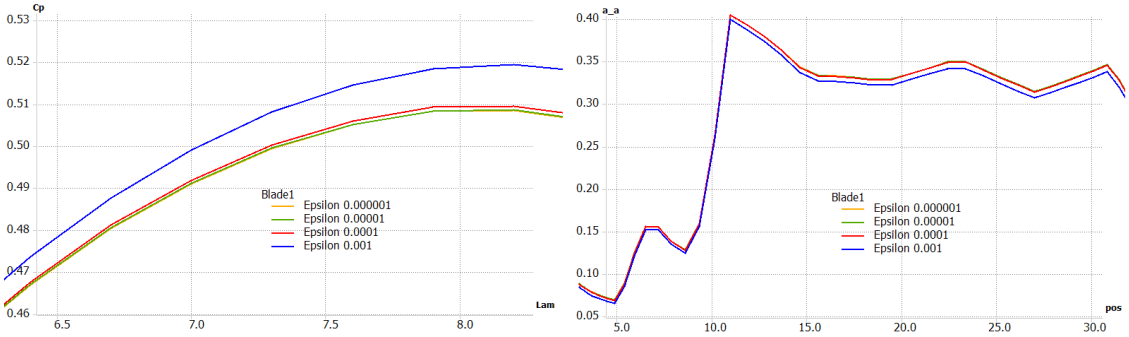

Figure 5.11: The effect of different epsilon values on simulation results

The left graph in Fig. 5.11 shows a section of the C_P curve for different epsilon values. The right graph shows the distribution of the axial induction factor. From these graphs an epsilon value of 10^{-5} should provide enough accuracy to any simulation.

Number of elements

In Fig. 5.12 different C_P curves for a discretization with 10, 50 and 100 elements are shown. For high tip-speed ratios a large difference between 10 and 50 elements is found. There is hardly a difference between 50 and 100 elements for the whole range. The right graph shows the effect of different element numbers for the distribution of the axial induction factor along the blade. Its obvious here that 10 elements are a relatively coarse distribution that filter a lot of the peaks

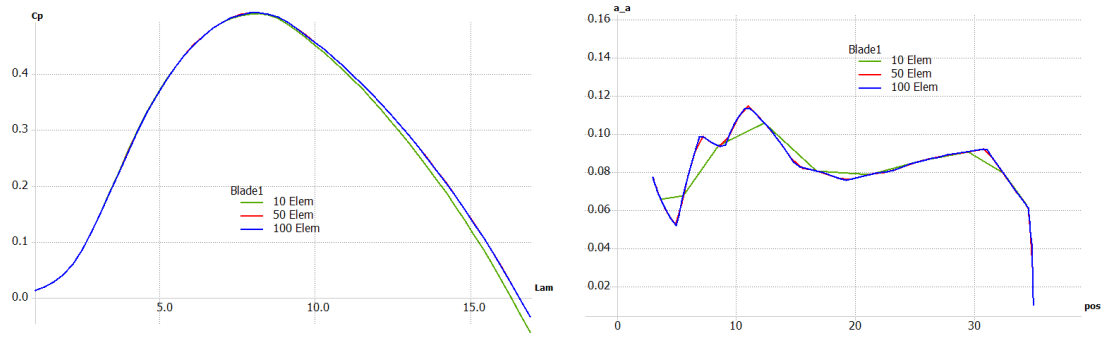

Figure 5.12: Effect of the number of elements

in the curve. From this a minimum element number of 50 is recommended for all simulations.

Tip loss corrections

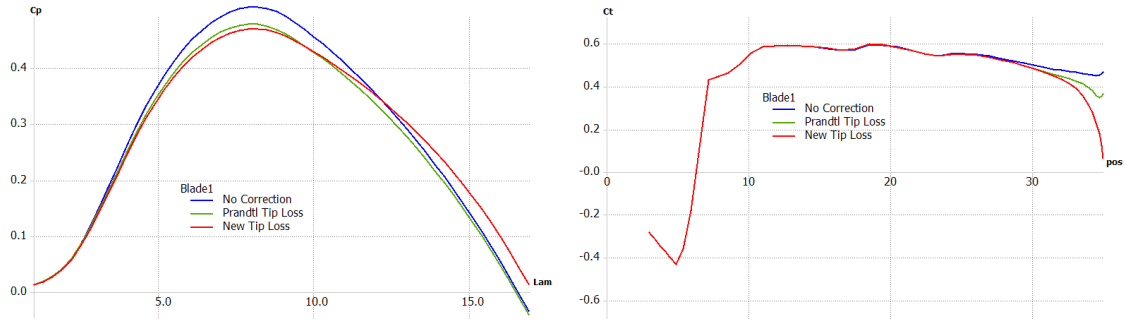

Figure 5.13: Different tip loss corrections

In Fig. 5.13 the effect of the different tip loss corrections on the C_P curve is illustrated. The right graph shows that the tip loss corrections only affect the distribution of the thrust coefficient in the tip region. C_T curves show that the New tip loss correction model after SHEN was motivated by the need to reduce the thrust coefficient to zero at the tip.

The left graph in Fig. 5.14 shows the effect of the different corrections on the axial induction factor, the right graph shows PRANTL's tip loss factor.

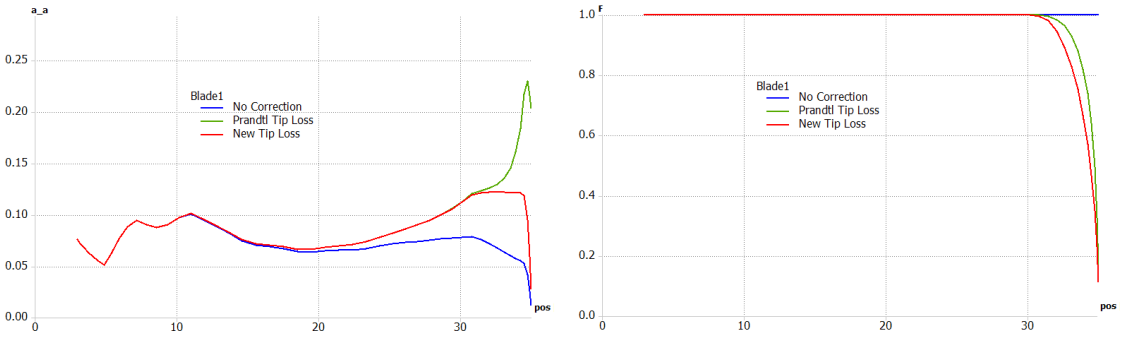

Figure 5.14: Different tip loss corrections

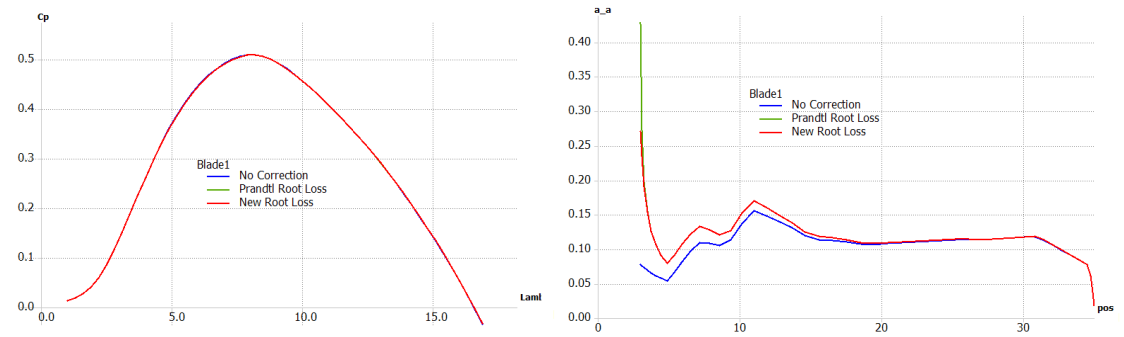

Figure 5.15: Different root loss corrections

Root loss corrections

The effect of the different root loss corrections on the C_P curve is little. It becomes more obvious in the distribution of the axial induction factor in the root region (Right graph, Fig. 5.15).

3D correction

The 3D correction accounts for the HIMMELSKAMP effect. An impact on the C_P curve for low tip speed ratios is visible in the left graph, Fig. 5.16. The right graph shows that the airfoils close to the root experience an increased lift coefficient.

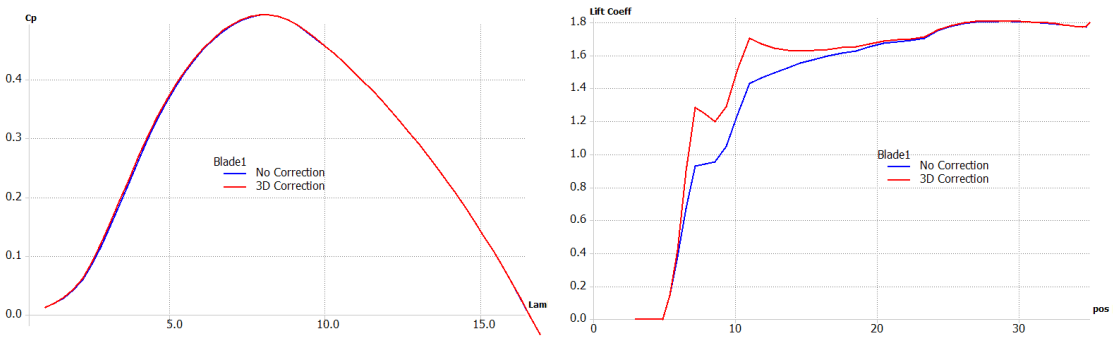

Figure 5.16: The 3D correction

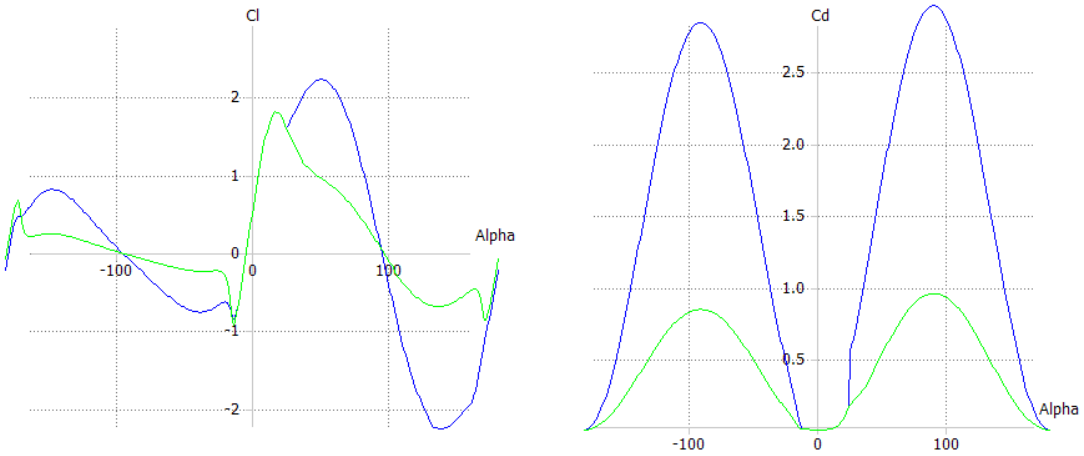

Figure 5.17: CD90 = 1.0 (green), CD90 = 3.0 (blue)

CD 90 Value / Extrapolation

To investigate the impact of the polar extrapolation on the accuracy of the simulation results the following blade was simulated:

Radius [m]	Chord [m]	Twist [°]	Foil
4	3.96	27	NACA 4414
6.375	3.18	16.43	NACA 4414
9.75	2.33	9.38	NACA 4414
16.5	1.47	3.62	NACA 4414
18.1875	1.34	2.82	NACA 4414
19.875	1.23	2.17	NACA 4414
23.25	1.06	1.14	NACA 4414
30	0.83	9	NACA 4414

For the first simulation the polar data from the NACA 4414 profile was extrapolated with a CD90 value of 1.0. The second simulation was carried out with polar data that was extrapolated with a CD90 value of 3.0. The different polars can be found in Fig. 5.17.

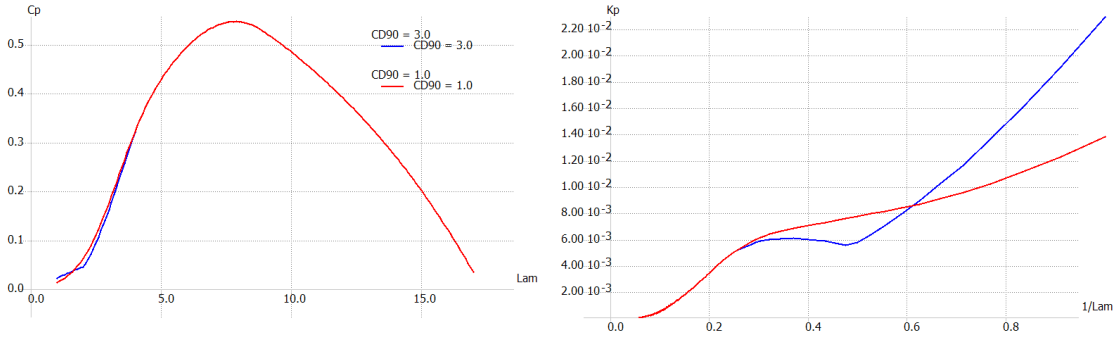

Figure 5.18: The effect of the extrapolation

The left graph in Fig. 5.18 shows that the large difference in the polar data only has an effect on the C_P curve for low tip-speed ratios, when the AoA's are high enough to be in the extrapolated region of the polars. When the turbine is operating at the highest efficiency the AoA's are relatively small, they are mostly on the linear part of the lift curve. The right graph shows the dimensionless power curve K_P over a range of tip-speed ratios. Here the extrapolated polars make a large difference in the region of high wind speeds ($V_0 \sim \frac{1}{\lambda}$), a fixed speed turbines operates at low tip speed ratios and the AoA's increase. The quality of the extrapolated polar data especially matters for the simulation of single speed stall turbines, where the point of maximum power output and the corresponding wind speed is of importance.

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] 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

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

[9] MONTGOMERIE, B: *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*, FOI Swedish Defence Research Agency, Scientific Report FOI-R-1035-SE, 2004

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

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

[12] 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