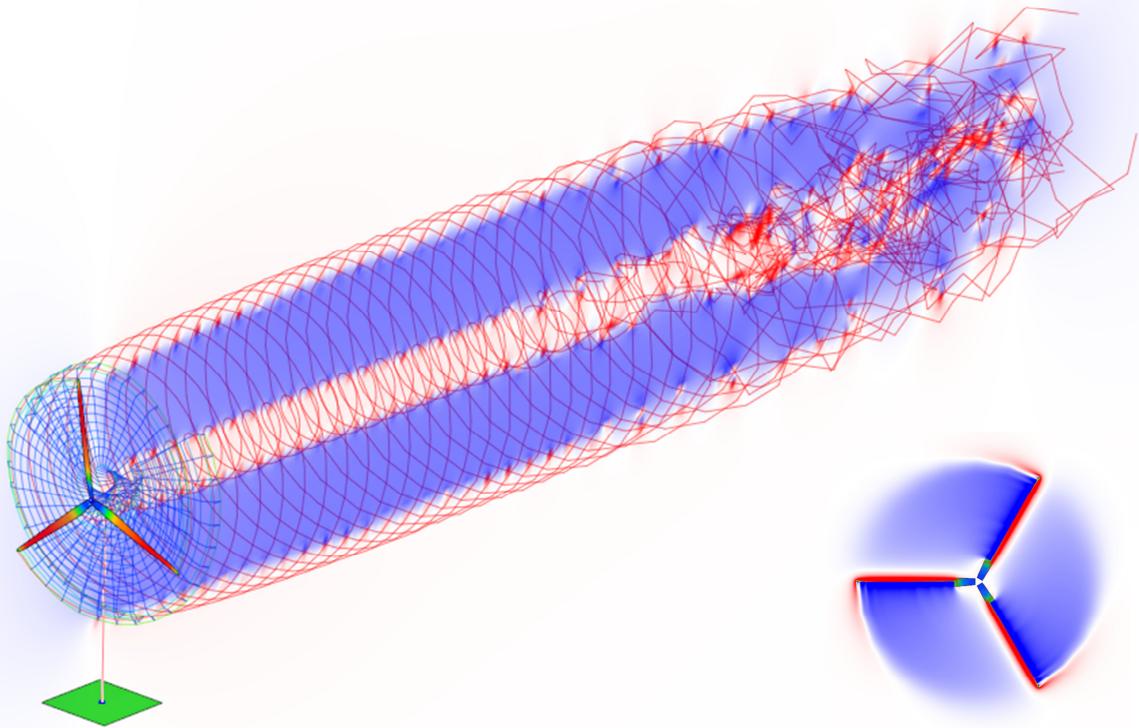


QBlade v0.9

Guidelines



David Marten

QBlade@outlook.com

Copyright 2015, TU Berlin



Notice:

This report was prepared as part of the research focus PAK780 "Wind Turbine Load Control Under Realistic Turbulent Inflow Conditions", funded by the German Science Foundation (DFG).

The software QBlade is being 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 bugs discovered after the distribution. If you discover a bug please report it on the discussion forum on QBlades sourceforge project page, found at: <http://sourceforge.net/p/qblade/discussion/>. The forum also covers a help section where questions are answered. For other inquiries contact the author directly under qblade@outlook.com.

QBlade is being developed at the Wind Energy Group of the Hermann Föttinger Institute (at the Chair of fluid Dynamics) of TU Berlin: <http://fd.tu-berlin.de/>. The project and code are hosted on sourceforge: <http://sourceforge.net/projects/qblade/> and the projects webpage is: <http://q-blade.org/>. If you want to reference QBlade inside a publication please refer to the list of QBlade publications at the end of this document.

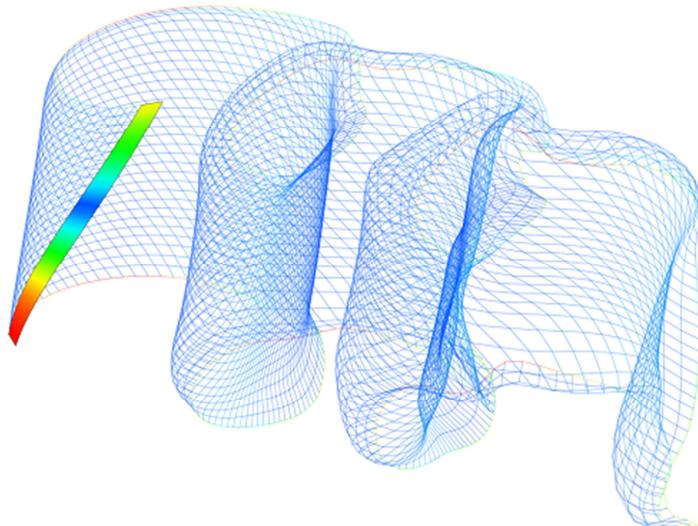


Table of Contents

Changes and bug fixes in v0.9	1
Nonlinear lifting line theory	1
The LLT algorithm implemented in QBlade.....	2
Common Controls	4
3D View	5
Simulation Output Variables:	11
Setting up a simulation; simulation parameters	13
Name and Rotor.....	14
Blade Discretization	15
Simulation Type:	15
Wind Input Data:.....	17
Rotor Angles:	19
Operational Point:.....	20
Length of Simulation:.....	20
Time Discretization:	20
Modeling Parameters:.....	21
Wake Modeling:.....	23
Vortex Modeling:	27
Output Parameters:.....	30
Algorithm Parameters:.....	30
Environmental Variables:	30
Validation	31
References.....	32
List of QBlade Publications.....	33

Introduction

This short manual is not intended to give a full overview of the applied simulation methods and their theory. This document is rather meant as a brief guide to enable the users to work with the new version of QBlade and give a short overview of the overall functionality that was implemented in QBlade v0.9.

In v0.9 a new module for unsteady aerodynamic simulations of wind turbines, based on a nonlinear lifting line free vortex wake algorithm was implemented [1]. As the implemented lifting line method is very general it can be used to simulate both horizontal and vertical axis wind turbines. Especially for the simulation of vertical axis wind turbines the lifting line method presents a large improvement over the double multiple stream-tube method, which was implemented in v0.6. The lifting line does not have any convergence problems and the accuracy is drastically improved. The lifting line method also outperforms the unsteady blade element momentum method, used for horizontal axis wind turbines, which was implemented through the coupling to FAST in v0.8. However, in contrast to FAST simulations the newly implemented lifting line algorithm is currently not coupled to a structural model and thus can only perform aerodynamic simulations with the assumption of rigid rotors. Setting up a lifting line simulation through the user interface in QBlade is a very quick task as reasonable values are already pre-selected for all simulation parameters. Nevertheless, if one wants to optimize the lifting line performance and accuracy to suit the current simulation, a good understanding of all governing parameters and their relevance for the investigated case is needed. The first part of this document gives an overview of the user interface of the Lifting Line module and its functionality, in the second part a short explanation of the underlying theory and the influence of all simulation parameters on the simulation is given.

Changes and bug fixes in v0.9

- Added: new unsteady nonlinear lifting line module for VAWT and HAWT mode
- Added: airfoil suction and pressure sides can now be switched in VAWT design
- Added: two visual points to 360° extrapolation graphs to help visualize the fine tuning of Montgomerie extrapolations
- Fixed: crash associated with the creation of 360° polars or blade loading objects
- Fixed: crash associated with the Eigen library under Windows 8
- Fixed: crash that could occur during polar extrapolation in some cases
- Ported QBlade to the newest Qt 5.4 version
- Completely removed the Miarex modules and classes
- Removed the gsl library and implemented custom polyfit function
- Compiled 64bit version for windows to circumvent 2GB memory limitation
- Several improvements of the GUI and many bug fixes

Nonlinear lifting line theory

The lifting line simulation method belongs to the family of the so called “vortex methods”. In Terms of computational cost, complexity and modeling of the physics the vortex methods are situated in between blade element momentum (BEM) methods and computational fluid dynamics (CFD). Using vortex methods, the flow field is modeled as inviscid, incompressible and irrotational; vortex elements are introduced in the form of straight or curved line segments to model both the rotor blades or wings and

the wake. The vortex lattice method models rotor blades or wings with a lattice of horseshoe vortices, which are located on the blades mid surface. The panel method models the blades upper and lower surfaces with a mesh of vortex rings. The lifting line method models the blades with a single line of vortices, located on the quarter chord points of the blade. With respect to the wake modeling two main classes of vortex methods can be distinguished. In prescribed wake methods the wake elements are convected on a prescribed path. Free wake methods update the position of the wake end nodes based on the local velocity, which is a combination of inflow velocity and induced velocity from all wake elements in the domain. The advantage of the free wake convection is a much higher accuracy as the wake shape is forming based on the underlying physical principles; however this comes at the price of a significantly higher computational cost compared to prescribed wake methods. The code implemented in QBlade is a “nonlinear lifting line – free vortex wake” algorithm. Nonlinear in this case implies that the circulation, computed for the bound vortices on the lifting line, is obtained from nonlinear lift and drag polars. One advantage of implementing the lifting line method in QBlade is that all existing blade designs that were used in blade element momentum simulations can be used in lifting line simulations without any conversion being necessary.

One large advantage of the vortex methods, compared to BEM methods, is that due to the sound modeling of the macroscopic flow physics, only very few empirical models related to microscopic fluid dynamics, where boundary layer effects play an important role, such as dynamic stall or stall delay need to be added. In many studies [2;3], vortex methods are identified as suitable to replace BEM codes in the near future to achieve a higher accuracy in turbine design and certification applications. Additionally the vortex methods not only provide simulation results concerning the rotor performance and the blade loads, but also result in the unsteady velocity field around the rotor and wake for every time step

The LLT algorithm implemented in QBlade

The lifting line algorithm implemented in QBlade roughly follows the work of van Garrel [4], but includes a range of extra functionality and improvements such as the vortex core modeling, time stepping, iteration loop and provisions for computational efficiency; multi-threading and wake connectivity tracking. Details of the implementation are not given in this document, however in the following section all simulation parameters are explained and short introductions or references to the theory are given if necessary. For completeness an overview over the simulation loop for one timestep is given in Figure 1. A large benefit of the implementation in QBlade is the wide range of functionality and the ease with which unsteady LLT simulations can be set up and evaluated. A brief overview of the functionality is listed in the following:

- Unsteady aerodynamic simulation of VAWT and HAWT rotors
- Upwind & downwind rotor configurations
- Blade discretization: custom, linear, sinusoidal
- Simulations of skewed inflow, yawed turbine, teeter, pre-cone and blade pitch angles
- Inflow conditions: stationary inflow, turbulent wind field, earth boundary layer, AeroDyn hub height wind file, custom transient QBlade simulation files
- Optional models: ground effects, tower shadow
- Custom wake discretization and vortex core modeling
- Velocity integration: 1st order: Euler forward integration, 2nd order: predictor corrector scheme
- Replay of conducted simulations and evaluation of 3D velocity field for every time step

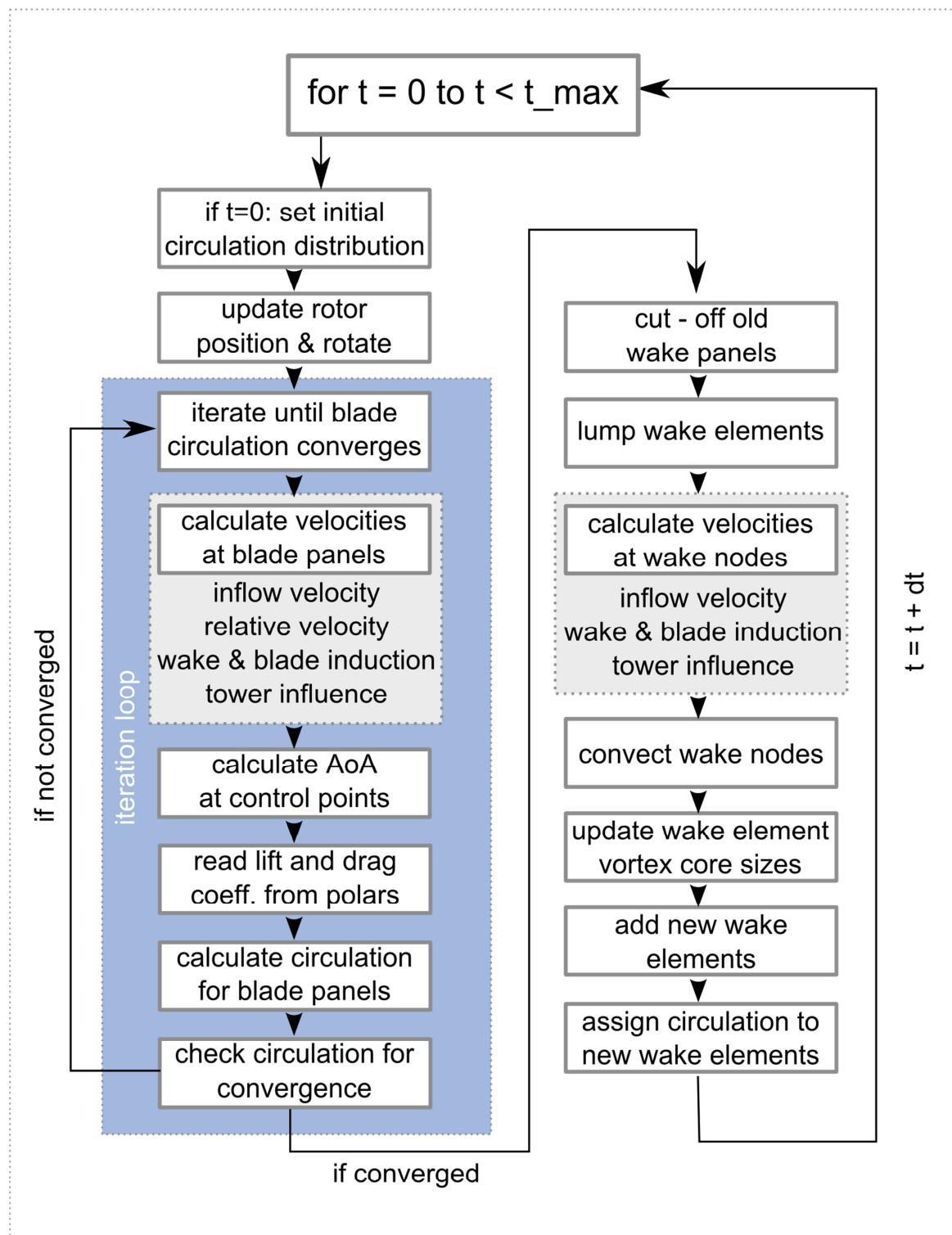


Figure 1. Flowchart of one time step loop of the implemented lifting line algorithm

Interface / Post Processing:

Common Controls

The Lifting Line Module is accessed through QBlade's Main Toolbar (Figure 2). To get into the Lifting Line Module for HAWT and/or VAWT turbines the HAWT/VAWT switch has to be set accordingly.



Figure 2. QBlade v0.9 Main Toolbar

Once inside the LLT Module the Control Bar (Figure 3) will appear below the main Toolbar. In the Control Bar the Graph/3D switch is used to switch between the 3D GL window, that shows a representation of the current Simulation and the graph window, which is used to show and export the various variables from the simulation results. The Simulation Object ComboBox shows the currently selected Lifting Line Simulation and is used to switch between the stored simulation objects. The time slider is used in multiple ways. If in the GL 3D window and the replay of the currently selected simulation was stored the time slider can be used to rewind or forward through the timesteps of the simulation. If the replay button is clicked an animation of the simulation is shown by automatically cycling through the timesteps. During replay a SpinBox will appear next to the replay button in which the time delay for the replay can be set in [s] to control the replay speed. If in the graph window the time slider is used to set the time step for which the *Blade Graphs* are plotting the simulation results. A large dot, that marks the currently selected time, will also appear in the *Time Graphs*.



Figure 3. Control Bar of the Lifting Line Module

Simulation Control

The Simulation Control part of the DockWindow (Figure 4) is used to setup, edit, rename, delete and start Lifting Line simulations. Lifting Line simulations can be created when a blade object is in the database. A press on the “New” button opens the simulation setup dialog (Figure 11, Figure 12), used to define new LLT simulations. When using the Edit/Copy button the simulation setup dialog of the currently selected LLT object is opened – this can be used to quickly check the simulation settings or to perform some quick changes of a few settings and create a new object under a different name.

Progress Bar

When a simulation is started the progress bar shows its progress and the remaining timesteps. A running simulation can be interrupted at any time by pressing the Stop button. The simulation can later be continued by pressing the Continue button. However, when a project is loaded from a project file unfinished simulations cannot be resumed but have to be restarted.

3D View

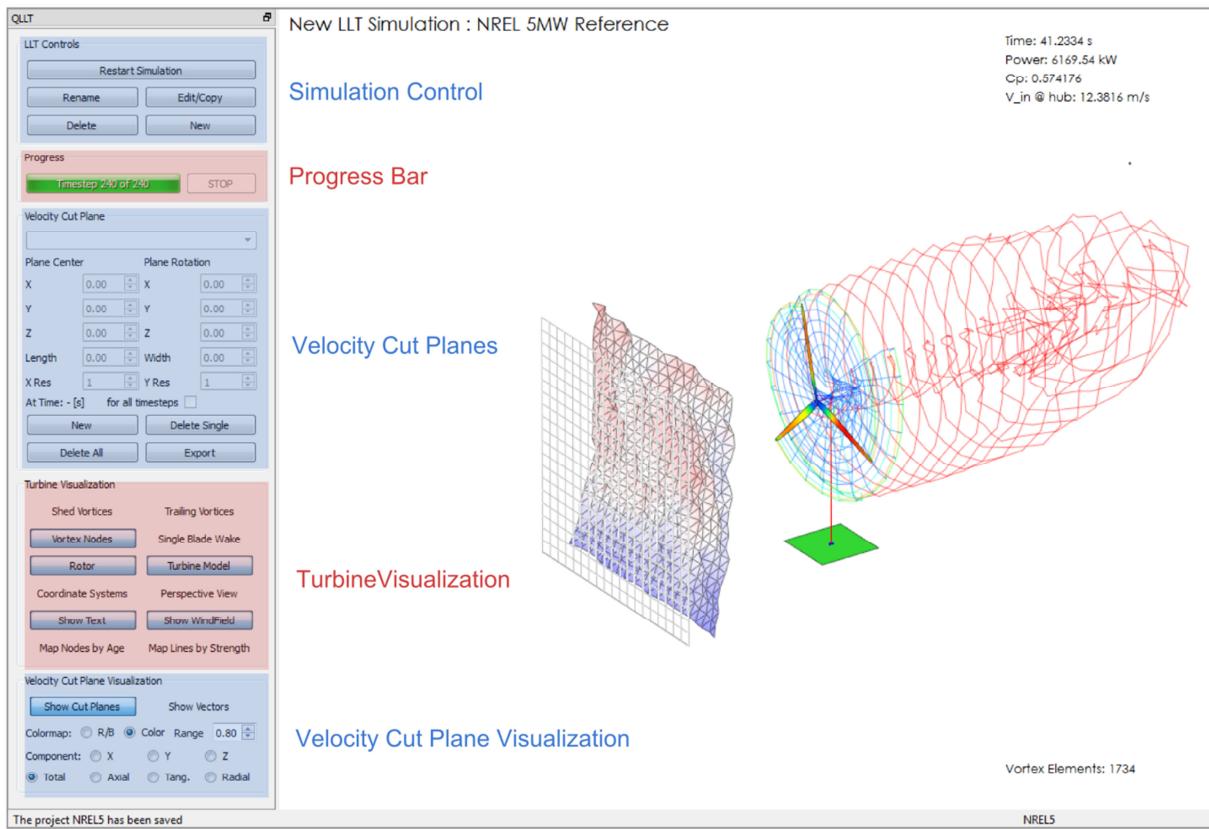


Figure 4. 3D View of The LLT Module

Velocity Cut Planes

A large benefit of the Lifting Line Theory over the Blade Element Momentum method is that as a result the velocity distribution in the flow field around the rotor can be obtained. The integrated “velocity cut planes” make it possible to access the information of the 3D velocity field in a general way. After a simulation is complete they can be used to visualize or export the velocity distribution. A cut plane is a 2D grid that can be oriented arbitrarily in the simulation domain, for which the velocity vectors are computed at the grid points. If a replay of the simulation has been stored, velocity fields can be reconstructed for any time step that is chosen with the TimeSlider (Figure 3). When the replay is not stored the cut planes can only be constructed for the last time step of the simulation. Cut planes are created by pressing the “New” button. The center of the cut plane is defined through the x-y-z coordinates under “Plane Center” in the cut plane dialog (Figure 5). The plane can be rotated around the x-y-z axis under “Plane Rotation”, angles are given in °. The rectangular grids edge lengths are given by the “Length” and “Width” parameters; “X Res” and “Y Res” define the grids resolution. The cut planes position and orientation are shown as a preview in the GL-window (Figure 5). If the “for all timesteps” checkbox is enabled, and a replay has been stored for the simulation, the cut plane is computed for all time steps. This allows to create animations where velocity distributions are included.

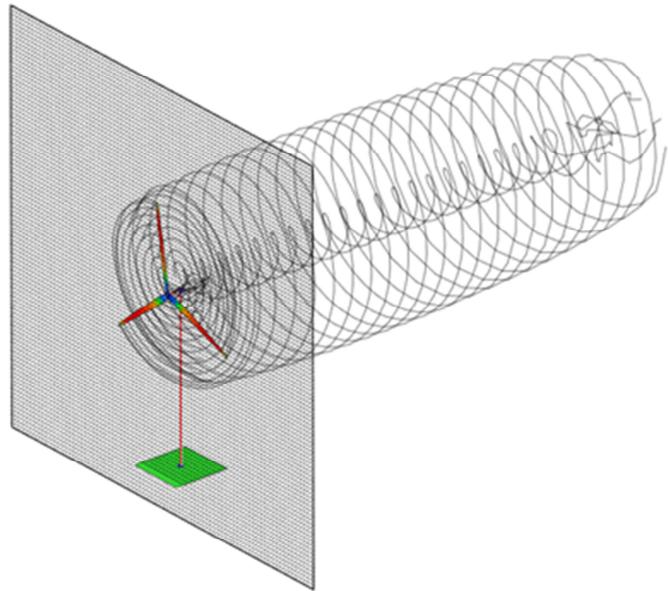
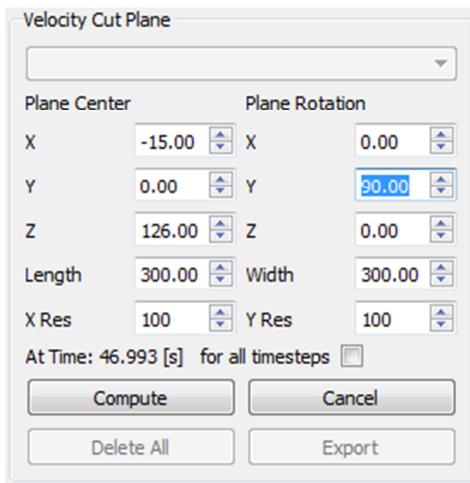


Figure 5. Cut plane dialog and newly created cut plane, upstream of the rotor parallel to the rotor plane

When the “Compute” button is pressed the velocities at the grid points are computed. Depending on the number of grid points and the number of wake elements the computation can require a significant amount of CPU time and can use up a large amount of memory, especially when the option “for all timesteps” is enabled. When a cut plane has been computed it can be viewed in the GL-window. Its appearance can be changed in the “Cut Plane Visualization” dialog. More importantly, the cut plane data can be exported to a plain text file, which has the format shown in Table 1. The first three columns are the position vector component of the grid points in [m]. The last three columns are the vector components of the total velocity at this point in [m/s]. The export functionality allows to easily produce velocity data such as wake velocity profiles or axial velocity profiles through the rotor plane to compare simulation results to experimental data.

Table 1. QBlade velocity cut plane export format

Export File Created with QBlade v0.9 on 08.07.2015 at 16:09:36					
Position Vector			Velocity Vector		
X	Y	Z	X	Y	Z
-15	-150	276	12.0274	-0.0970743	0.0992186
-15	-146.97	276	12.0274	-0.0984592	0.102661
-15	-143.939	276	12.0274	-0.0998126	0.106215
-15	-140.909	276	12.0273	-0.101129	0.109884
-15	-137.879	276	12.0273	-0.102404	0.113669
-15	-134.848	276	12.0272	-0.103631	0.117571
-15	-131.818	276	12.0271	-0.104803	0.121592
-15	-128.788	276	12.027	-0.105915	0.125731
-15	-125.758	276	12.0269	-0.106959	0.129991
-15	-122.727	276	12.0267	-0.107928	0.134369
-15	-119.697	276	12.0266	-0.108814	0.138867
-15	-116.667	276	12.0264	-0.109609	0.143482
-15	-113.636	276	12.0262	-0.110306	0.148213

Turbine Visualization

The buttons under “Turbine Visualization” activate/deactivate the rendering of the respective features, such as vortex lines or nodes or the turbine or rotor geometry. If “Map Nodes by Age” is selected the vortex nodes are colored, depending on the time they are released at the trailing edge, by cycling through the color spectrum. If “Map Lines by Strength” is selected the vortex line elements are colored according to their vortex strength.

Velocity Cut Plane Visualization

The velocity cut plane visualization dialog is used to change the appearance and the velocity component that is visualized in the cut planes. The velocity vectors at the grid points can be plotted and one of two color maps can be selected that maps the velocity magnitude in the plane. Additionally the velocity component that is to be plotted can be selected in the dialog. The options “Total”, “X”, “Y” and “Z” plot the magnitude or respective component of the velocity in the global coordinate system, while “Axial”, “Tang.” and “Radial” plot the respective component in the rotor coordinate system (see examples in Figure 6, Figure 7 and Figure 8). The SpinBox “range” is used to adjust the colormap range (values outside the range are clipped). This is a practical tool to highlight flow features and was used for the Y- and Z-component in Figure 8.

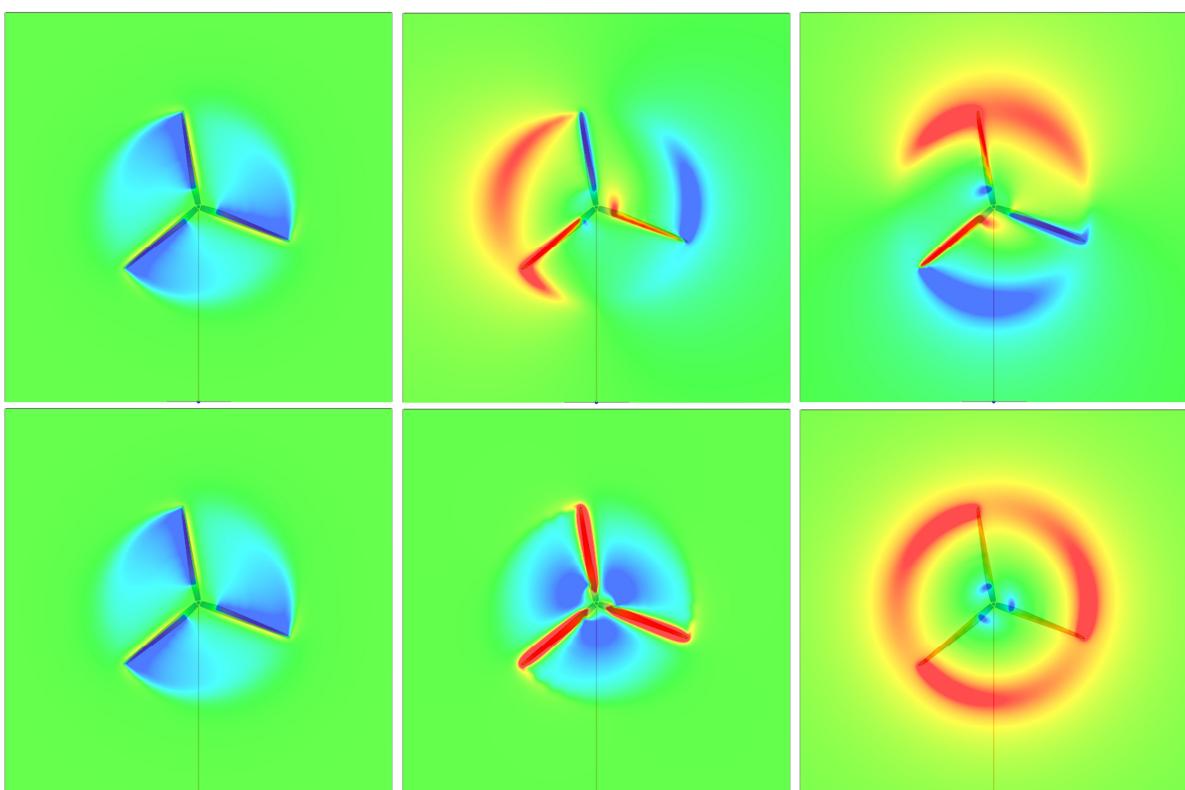


Figure 6. Velocity distribution in the rotor plane; Top row: X-, Y-, Z-component; bottom row: Axial- Tang., Radial- component,

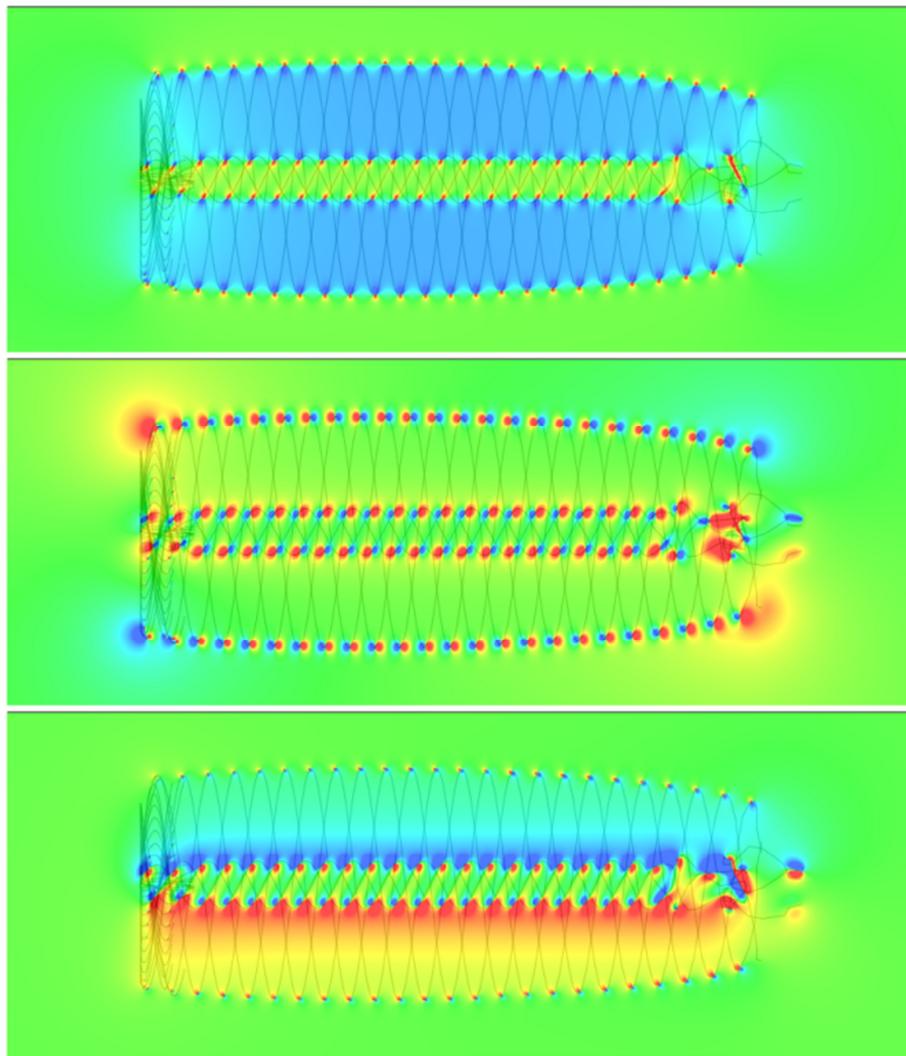


Figure 7. Exemplary velocity distribution in horizontal mid-plane; from top to bottom: X-, Y-, and Z-component

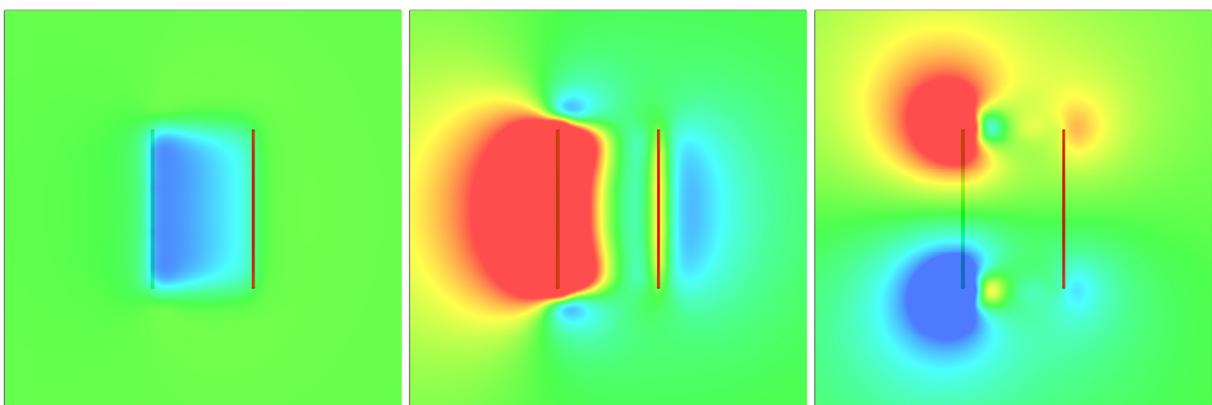


Figure 8. Exemplary velocity distribution in VAWT H-rotor mid (equilibrium) plane, from left to right: X-, Y- and Z-component

Graph View

In the graph view the simulation results are plotted. When a simulation is running the graphs are automatically updated for every time step, allowing the user to monitor the performance of the rotor in live. When in the graph window while a simulation is running the simulation runtime is slightly faster, as the GL visualization, that requires a small amount of CPU time during each timestep, is deactivated.

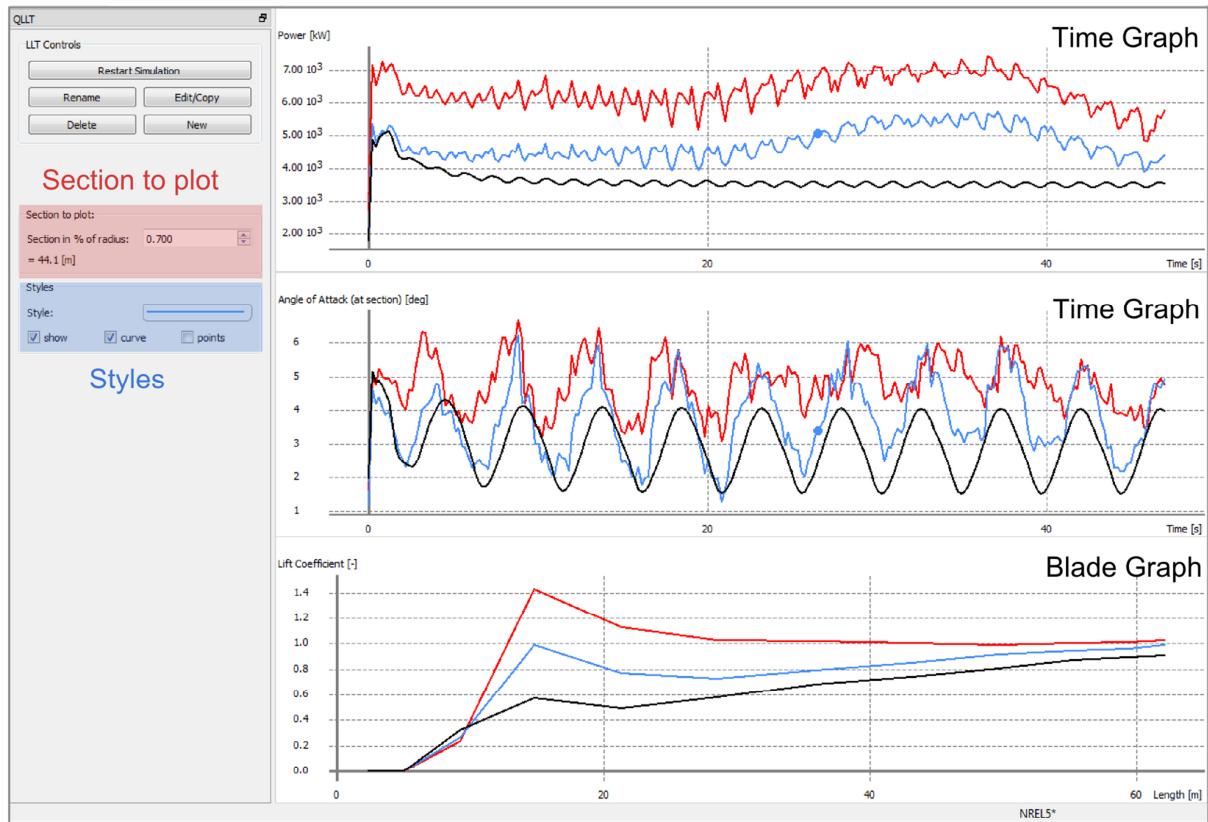


Figure 9. Graph View of the LLT Module

Graph Controls

The dynamic graphs in QBlade are fully controllable with the mouse. Zooming in and out of a graph is realized with the mouse wheel. Pressing the y or x keys during zooming allows only zooming the respective axis. The graph context menu (Figure 10) can be opened by pressing the right mouse button while on a graph. In the context menu the automatic resetting of the graph scales can be activated/deactivated and the graph type can be set (Time Graph or Blade Graph). A double click on a graph opens the graphs options menu (Figure 10). In the options menu the variables for the x- and y-axis are set and the graph style can be changed.

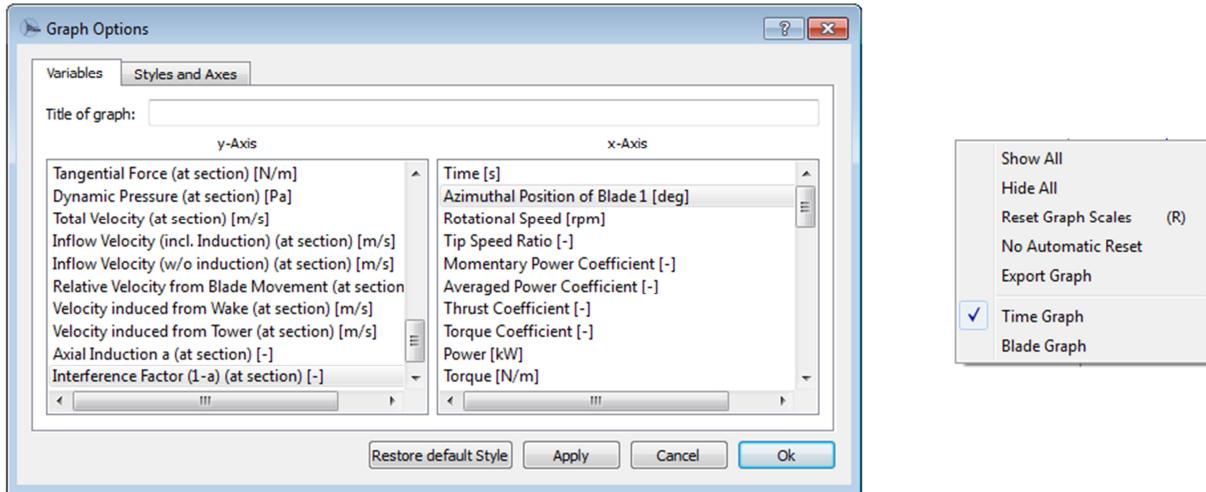


Figure 10. Graph Options(left) and Graph context menu (right)

Section to Plot

The SpinBox in “section to plot” is used to select a radial position of the rotor blade for which the local blade variables are plotted in the *time graphs*. The radial position is defined in % of blade radius. This also enables the comparison of local blade variables between rotors of different size, as the radial position is made dimensionless.

Styles

The styles dialog enables to set the appearance of the graph curves. Each simulation object can have its own style. If *points* is selected the computed points are highlighted. The *curve* switch enables the plotting of the point’s connections and *show* shows or hides the curve. All curves can be hidden or shown using Show All/Hide All in the graph context menu (Figure 10).

Time Graphs

The time graphs show the global variables for rotor performance over the simulation time, such as Cp Betz or Thrust. However it is also possible to plot local blade variables in the time graphs. The blade variables are marked with *(at station)*. Thus the blade variables are plotted for the radial position that is selected in the “section to plot” SpinBox (Figure 9).

Blade Graphs

The blade graphs show the distribution of local blade variables over the rotor blade (if length/height is chosen for the x-axis), such as the Lift coefficient or the local inflow velocity. The variables are always plotted for blade 1 only. The variables are plotted for the time step that is currently selected at the TimeSlider (Figure 3).

Simulation Output Variables:

To provide an overview the simulation output variables are listed here with a short description. The global rotor output variables can be plotted in time graphs only whereas the local blade variables can be plotted in both time and blade graphs.

The rotor output variables are integral values, usually obtained by integration of the local variables over the blade length.

Table 2. Rotor Output (Global) Variables

Name	Description	Unit
Time	The simulation time	[s]
Azimuthal Position of Blade 1	At 0° the blade is pointing upwards	[°]
Rotational Speed	The current rotor speed	[rpm]
Tip Speed Ratio	The current tip speed ratio	[‐]
Power Coefficient (HAWT)	The current Cp Betz for HAWT	[‐]
Momentary Power Coefficient (VAWT)	The current Cp Betz for VAWT	[‐]
Averaged Power Coefficient (VAWT)	Cp Betz averaged over one revolution	[‐]
Thrust Coefficient	The current thrust coefficient	[‐]
Torque Coefficient	The current thrust coefficient	[‐]
Power	The current rotor power	[kW]
Torque	The current rotor torque	[Nm]
Thrust in X	Thrust force in global X-direction	[N]
Thrust in Y	Thrust force in global Y-direction	[N]
Thrust in Z	Thrust force in global Z-direction	[N]
Wind Speed at Hub Height	Undisturbed inflow velocity w/o induction	[m/s]
Pitch Angle	The pitch angle of all blades	[°]
Yaw Angle (HAWT)	The rotor yaw angle	[°]
Horizontal Inflow Angle	The horizontal inflow angle	[°]
Vertical Inflow Angle	The vertical inflow angle	[°]
Out of Plane Root Bending Mom (HAWT)	Defined with respect to the current rotor plane	[Nm]
In Plane Root Bending Moment (HAWT)	Defined with respect to the current rotor plane	[Nm]
Iterations	Iterations needed to compute bound blade circulation	[‐]

The local blade output variables are values obtained at the individual blade stations, thus they can be plotted over the blades length or height.

Table 3. Blade Output Variables

Name	Description	Unit
Lift Coefficient	Lift coefficient from polar data	[\cdot]
Drag Coefficient	Drag coefficient from polar data	[\cdot]
Lift to Drag Ratio	L/D ratio from polar data	[\cdot]
Circulation	The bound circulation	[m^2/s]
Angle of Attack	The computed AoA that is used to get polar data	[$^\circ$]
Length / Height	Length (HAWT) or height (VAWT)	[m]
Chord	The chord length	[m]
Blade Twist Angle	The blade twist angle	[$^\circ$]
Normal Force Coefficient	C_n defined in the current direction of the rotor axis	[\cdot]
Tangential Force Coefficient	C_t defined in current tangential direction	[\cdot]
Normal Force	Computed from C_n	[N/m]
Tangential Force	Computed from C_t	[N/m]
Dynamic Pressure	From total velocity	[Pa]
Total Velocity	Total velocity magnitude at blade station (with induction)	[m/s]
Inflow Velocity (including Induction)	Total velocity magnitude less relative blade movement	[m/s]
Inflow Velocity (without Induction)	Free stream inflow velocity at station (no induction)	[m/s]
Relative Velocity from Blade Movement	The velocity magnitude from blade movement at station	[m/s]
Velocity induced from Wake	The wake induced velocity magnitude at the station	[m/s]
Velocity induced from Tower	The tower induced velocity magnitude at the station	[m/s]
Axial induction	The axial induction factor	[\cdot]
Interference Factor (1-a) (VAWT)	The axial interference factor to compare with the DMST	[\cdot]
Tangential Induction (HAWT)	The tangential induction factor at the station	[\cdot]
Reynolds Number	The Reynolds number at the station	[\cdot]

Setting up a simulation; simulation parameters

A LLT simulation can be defined through a simple setup dialog (Figure 11; Figure 12). In the following section all simulation parameters and their role in a simulation are described. The section is ordered into groups, in the same order as they appear in the setup dialog.

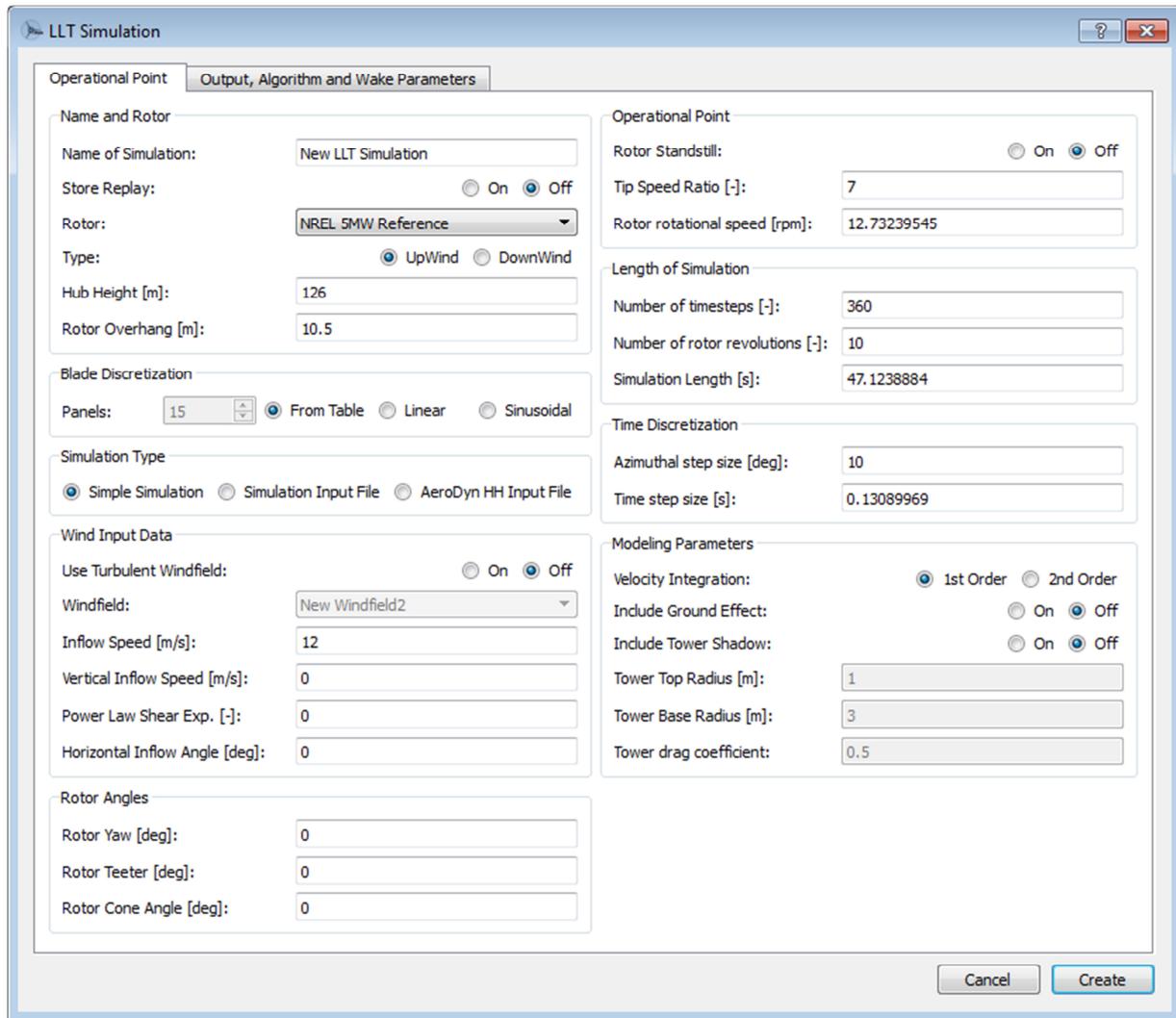


Figure 11. 1st tab of LLT simulation setup dialog

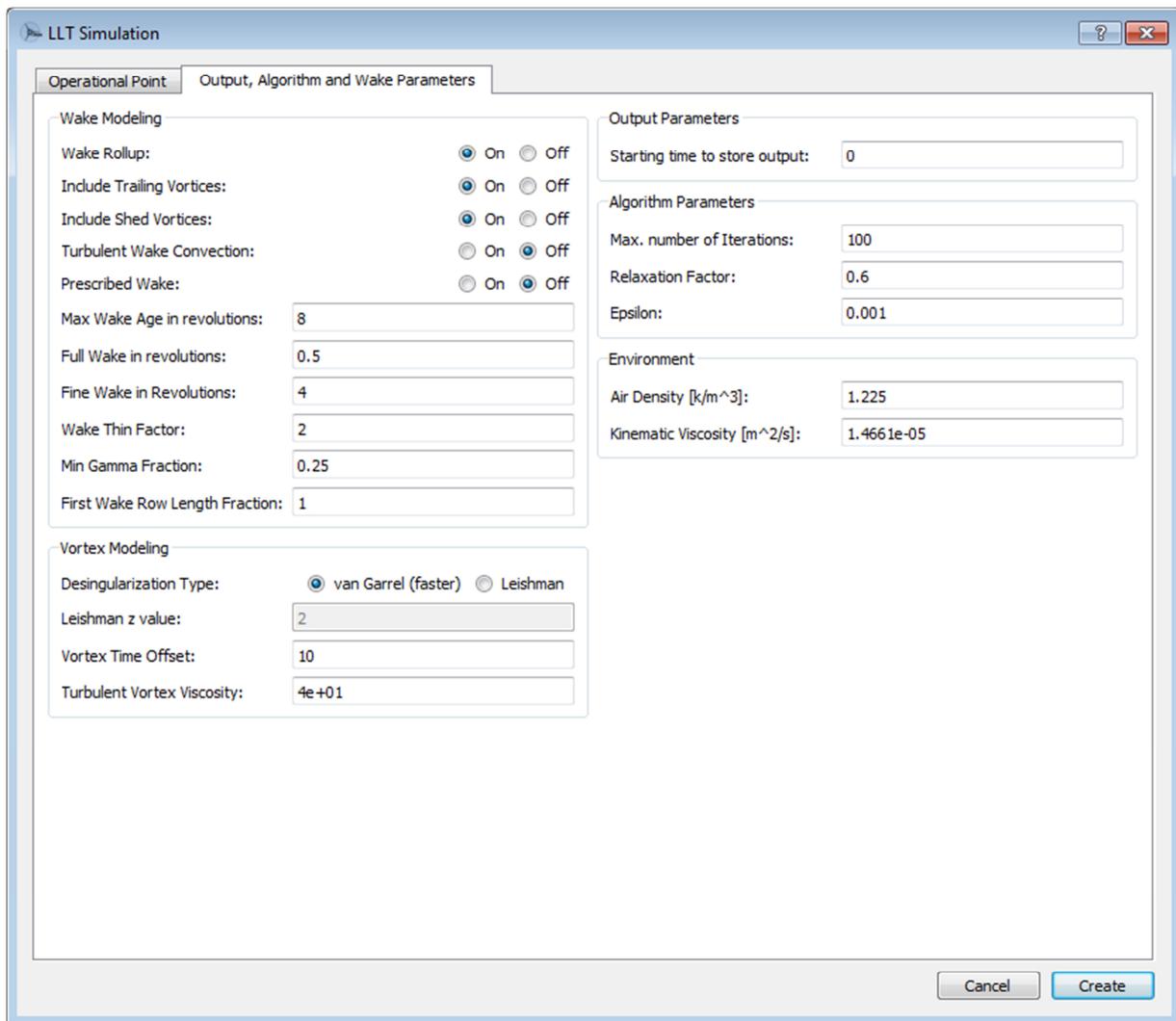


Figure 12. 2nd tab of LLT simulation setup dialog

Name and Rotor

Name of Simulation:

To assign a unique name to the simulation under which it will be stored in the database.

Store Replay:

If set to “On” all wake node positions are stored for every time step of the simulation. If the replay is stored, an animation of the simulation can be viewed after the simulation is complete. Storing of the wake nodes also enables to recreate the velocity field at any time step during post-processing. The store replay option can require, depending on the wake discretization and the number of time steps, a large amount of memory space. If you are using the 32bit version of QBlade, a crash will occur if more than 2GB of memory are requested. It is recommended to use the 64Bit version with at least 4GB of RAM if replays are stored.



Rotor:

The rotor for which the simulation is defined can be selected from this drop-down menu.

Type:

The selection defines the turbine as upwind or downwind. This option is only available for HAWT simulations.

Hub Height / Rotor Overhang / Height of Rotor Centre

These values define the wind turbine geometry and are important when calculating the position of the rotor with respect to the wind field (when turbulent wind field or earth boundary layer is used); tower (when tower model is turned on); ground (when ground effects are turned on).

Blade Discretization

The user can choose from linear or sinusoidal blade discretization and define the total number of blade panels. If “From Table” is selected the discretization is directly obtained from the blade table that was created in the “Blade Design Module”.

Simulation Type:

Simple Simulation:

A simple simulation does not require an input file. Fixed values for the rotational speed of the rotor, the wind inflow angle and rotor yaw have to be defined once and are then used throughout the whole simulation.

Simulation Input File:

A simulation input file is used to define a transient simulation. Values for rotor speed, horizontal and vertical wind speed, wind direction and yaw and pitch angle can be defined for different points in time in the input file (Table 4). When the simulation requests values in between the defined points in time a linear interpolation is employed. If the current simulation time is lower or larger than the minimum or maximum point in time defined in the input file then values at the lowest, respectively largest, point in time are used for the current time step. It is only necessary to define the first three columns of the input file; all other values (vertical wind speed, wind direction, yaw and pitch angle) are set to zero in this case. The file type has to be *.sim to be recognized for import in QBlade.

Table 4. Sample QBlade Simulation file, to be copy/pasted in a text file

Sample QBlade Simulation File .sim						
SIM	ROT	HOR WND	VER WND	WND	YAW	PITCH
TIME	SPEED	SPEED	SPEED	DIR	ANG	ANGLE
0.000	1.000	11.000	0.000	0.000	0.000	0.000
5.000	2.000	11.000	0.000	0.000	0.000	0.000
10.000	4.000	11.000	0.000	5.000	0.000	0.000
15.000	7.000	11.000	0.000	10.000	0.000	0.000
20.000	11.000	11.000	0.000	17.000	0.000	0.000
25.000	12.000	11.000	0.000	27.000	0.000	0.000
30.000	13.000	11.000	0.000	40.000	10.000	0.000
35.000	12.000	11.000	0.000	40.000	20.000	0.000
40.000	11.000	11.000	0.000	40.000	30.000	0.000
45.000	11.000	11.000	0.000	40.000	40.000	0.000
50.000	11.000	11.000	0.000	40.000	40.000	0.000
55.000	11.000	11.000	0.000	40.000	40.000	0.000
60.000	11.000	11.000	0.000	40.000	40.000	0.000
65.000	11.000	11.000	0.000	40.000	40.000	0.000
70.000	11.000	11.000	0.000	40.000	40.000	0.000
75.000	11.000	11.000	0.000	40.000	40.000	0.000
80.000	11.000	11.000	0.000	40.000	40.000	0.000
85.000	10.000	11.000	0.000	40.000	40.000	0.000
90.000	7.000	11.000	0.000	40.000	40.000	0.000
95.000	3.000	11.000	0.000	40.000	40.000	0.000

AeroDyn HH input File:

AeroDyn hub height input files (Table 5) can be used inside a simulation. The file should be of the *.wnd type to be imported. For a full description of AeroDyn hub height input files see the AeroDyn user's guide on pages 13-17 [5].

Table 5. Sample AeroDyn type hub-height wind file, to be copy/pasted in a text file

Sample hub-height wind file for AeroDyn .wnd							
Time	Wind Speed	Wind Dir	Vert. Speed	Horiz. Shear	Vert. Shear	LinV Shear	Gust Speed
0.000	15.000	5.000	-1.000	0.020	0.140	0.000	0.000
0.100	16.545	4.755	-0.900	0.022	0.140	0.000	0.000
0.200	17.939	4.045	-0.800	0.024	0.140	0.000	0.000
0.300	19.045	2.939	-0.700	0.027	0.140	0.000	0.000
0.400	19.755	1.545	-0.600	0.030	0.140	0.000	0.000
0.500	20.000	0.000	-0.500	0.033	0.140	0.000	0.000
0.600	19.755	-1.545	-0.400	0.036	0.140	0.000	0.000
0.700	19.045	-2.939	-0.300	0.040	0.140	0.000	0.000
0.800	17.939	-4.045	-0.200	0.045	0.140	0.000	0.000
0.900	16.545	-4.755	-0.100	0.049	0.140	0.000	0.000
1.000	15.000	-5.000	0.000	0.054	0.140	0.000	0.000
1.100	13.455	-4.755	0.100	0.060	0.140	0.000	0.000
1.200	12.061	-4.045	0.200	0.066	0.140	0.000	0.000
1.300	10.955	-2.939	0.300	0.073	0.140	0.000	0.000
1.400	10.245	-1.545	0.400	0.081	0.140	0.000	0.000
1.500	10.000	0.000	0.500	0.090	0.140	0.000	0.000
1.600	10.245	1.545	0.600	0.099	0.140	0.000	0.000
1.700	10.955	2.939	0.700	0.109	0.140	0.000	0.000
1.800	12.061	4.045	0.800	0.121	0.140	0.000	0.000
1.900	13.455	4.755	0.900	0.134	0.140	0.000	0.000
2.000	15.000	5.000	1.000	0.148	0.140	0.000	0.000

Wind Input Data:

Use Turbulent Windfield:

If the switch is set to "On" a turbulent windfield from the database of QBlade can be used in a simulation. In case a turbulent wind field is used in conjunction with a *Simulation Input File* or *AeroDyn input file* the column with wind speeds is overridden and replaced with wind data from the wind field. To use a windfield in a simulation, its dimensions need to be large enough for the rotor that is simulated. It is also important to check that the height of the wind field center and the hub height of the turbine match, such that the rotor is fully submerged in the windfield. In case of a wind speed being requested for a location outside of the spatial or temporal boundaries of the wind field (if a part of the rotor sticks out of the wind field or the LLT simulation time is larger than that of the wind field) the mean velocity at hub height of the selected wind field is used. Using Taylor's frozen turbulence hypothesis, the turbulent wind field is "marched" through the simulation domain along the mean inflow direction and with the mean inflow velocity (Figure 13). To make sure that the rotor is fully submerged in the quasi 3 dimensional wind field at the beginning of a simulation (such as when the simulation starts with a yawed rotor) the wind field is already marched by its width along the mean flow direction before the first time step. This approach is consistent with the treatment in AeroDyn and FAST, and thus ensures comparability of simulation results between these codes and QBlade.

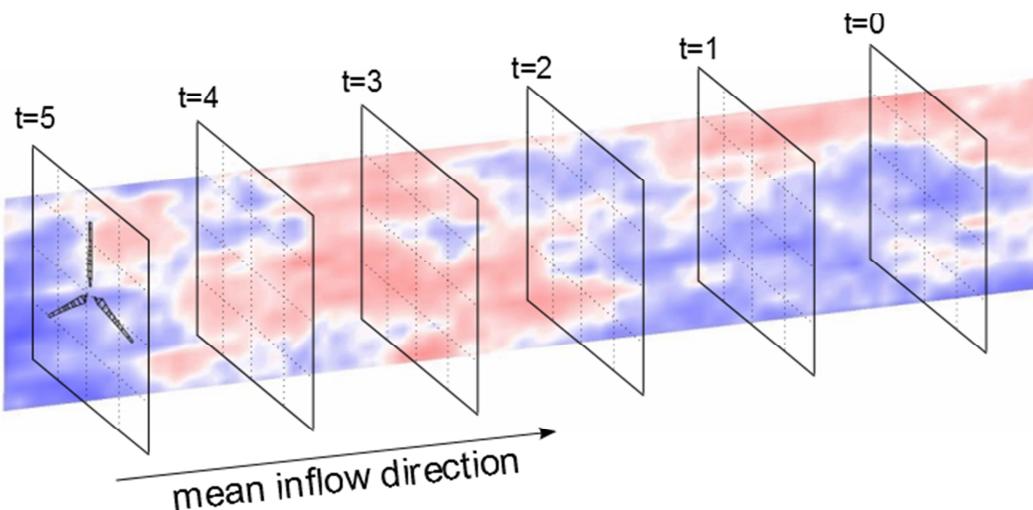


Figure 13. Turbulent wind field marched through domain under Taylor's frozen turbulence hypothesis

Windfield:

The windfield, to be used for the current simulation, can be selected from the drop down menu.

Inflow Speed [m/s]:

This value defines the horizontal inflow speed component for the current simulation. It is defined in the x-direction of the global coordinate system.

Vertical Inflow Speed [m/s]:

This value defines the vertical inflow speed component for the current simulation. A positive value corresponds to an upwards velocity.

Power Law Shear Exponent [-]:

A simple atmospheric boundary layer can be used through the definition of the power law shear exponent:

$$u(x) = u_{hub} \left(\frac{z}{z_{hub}} \right)^{\alpha}, \quad (1)$$

Where α is the Hellmann exponent, u_{hub} the velocity that is defined as inflow speed and z_{hub} the value defined as hub height. An example for values of the Hellmann exponent is given in (Table 6) [6].

Table 6: Exemplary values of the Hellmann exponent

location	Hellmann exponent α
Unstable air above open water surface:	0.06
Neutral air above open water surface:	0.10
Unstable air above flat open coast:	0.11
Neutral air above flat open coast:	0.16
Stable air above open water surface:	0.27
Unstable air above human inhabited areas:	0.27
Neutral air above human inhabited areas:	0.34
Stable air above flat open coast:	0.40
Stable air above human inhabited areas:	0.60

Horizontal Inflow Angle [$^{\circ}$]:

This value defines the horizontal inflow angle for the current simulation. It is defined in mathematically positive direction in a top down view (see Figure 14).

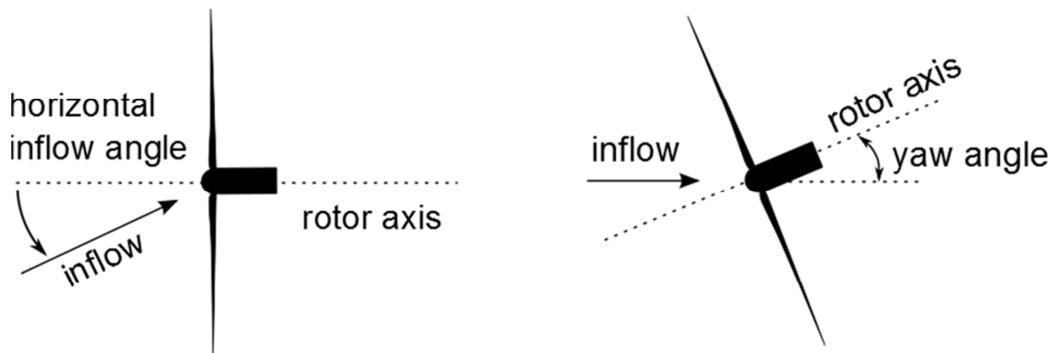


Figure 14. Definition of the horizontal inflow and yaw angle; top down view

Rotor Angles:

Rotor Yaw (HAWT only) [°]:

This value defines the rotor yaw angle for the current simulation. It is defined in mathematically positive direction in a top down view (see Figure 14).

Rotor Teeter (HAWT only) [°]:

This value defines the rotor teeter angle for the current simulation. Its definition is depicted in Figure 15.

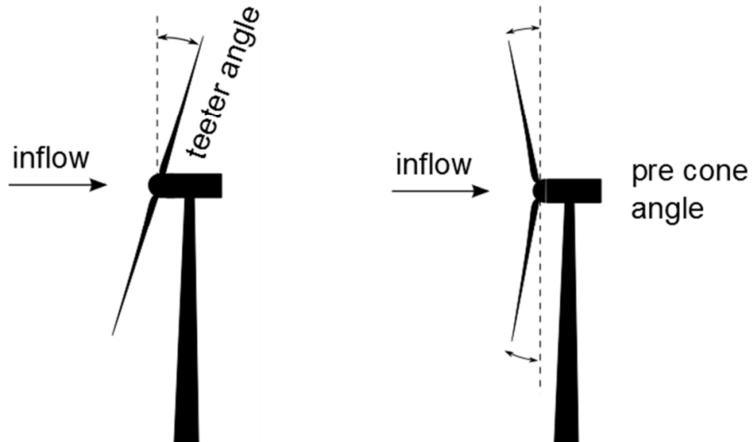


Figure 15. Definition of the rotor teeter and pre-cone angle; side view

Rotor Pre Cone Angle (HAWT only) [°]:

This value defines the rotor pre-cone angle for the simulation. Its definition is depicted in Figure 15.

Y-Roll / X-Roll Angle (VAWT only) [°]:

The Y- and X-roll angles define an inclination of the VAWT turbine around the X- and Y axis of the VAWT turbines tower bottom coordinate system (Figure 16).

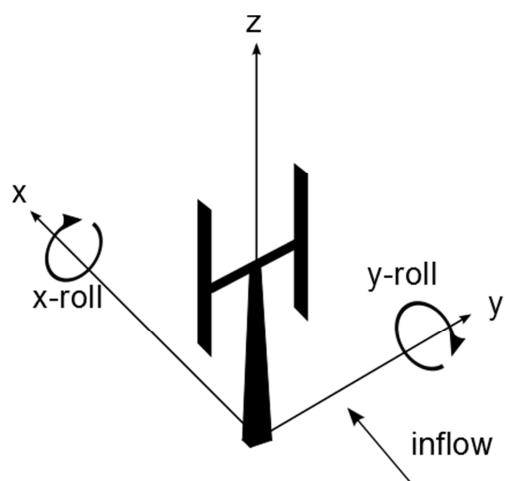


Figure 16. Definition of X- and Y- roll angles for VAWT

Operational Point:

Rotor Standstill:

If this option is switched on the simulation will be carried out at a rotational and a tip speed ratio of 0.

Tip Speed Ratio [-]:

This value defines the tip speed ratio for which the current simulation will be carried out. If the tip speed ratio is changed the rotational speed is updated to result in the chosen tip speed. Because the tip speed ratio depends on the wind inflow speed it can only be set as constant for a “Simple Simulation”.

Rotor Rotational Speed [rpm]:

This value defines the rotational speed of the rotor for which the current simulation will be carried out. If the rotational speed is changed the tip speed ratio is updated to result in the chosen rotational speed.

Length of Simulation:

Number of Timesteps [-]:

This value defines the total number of timesteps for which the current simulation will be carried out. If the number of timesteps is changed the number of rotor revolutions and the total simulated time are updated to result in the chosen number of timesteps.

Number of Rotor Revolutions [-]:

This value defines the number of full rotor revolutions for which the current simulation will be carried out. If the number of rotor revolutions is changed the number of timesteps and the total simulated time are updated to result in the chosen number of rotor revolutions. The total number of rotor revolutions cannot be chosen when a “Simulation Input File” is used because the rotational rotor speed is not constant in this case.

Simulation Length [s]:

This value defines the length of the current simulation. If the simulation length is changed the number of rotor revolutions and the total number of time steps are updated to result in the simulation length.

Time Discretization:

Azimuthal step size [°]:

This value defines the time step size in terms of azimuthal discretization. If the azimuthal step size is changed the time step size is updated to result in the azimuthal discretization. An azimuthal discretization step size cannot be defined when a “Simulation Input File” is used because the rotational rotor speed is not constant in this case and the implemented LLT method is currently only capable of constant time stepping.

Time step size [s]:

This value defines the size of a time step. If the time step size is changed the azimuthal step size is updated to result in the chosen time step size.

Modeling Parameters:

Velocity Integration:

One of two types of velocity integration can be selected for the free convection step of the wake. The 1st order Euler forward integration scheme:

$$\vec{x}_{t+1} = \vec{x}_t + (V_\infty + V_{ind}(\vec{x}_t))\Delta t, \quad (2)$$

and the 2nd order predictor-corrector scheme:

$$\vec{x}_{t+1,cor} = \vec{x}_t + (2V_\infty + V_{ind}(\vec{x}_t) + V_{ind}(\vec{x}_{t+1}))\frac{\Delta t}{2}. \quad (3)$$

Being of 2nd order the predictor-corrector scheme is more accurate, however two evaluations of the velocity field are required for every time step, which reduces the computational efficiency (the computational time is effectively doubled). On the other hand the 2nd order method allows for the selection of a larger time step, compared to the 1st order scheme.

Include Ground Effects:

If this switch is set to “On” ground effects are included in the simulation. The ground effects are modelled by mirroring the blade and wake vortices at the ground plane (Figure 17). As the mirroring doubles the number of vortices that are needed to be evaluated during the free wake convection step the computational time is effectively doubled as well.

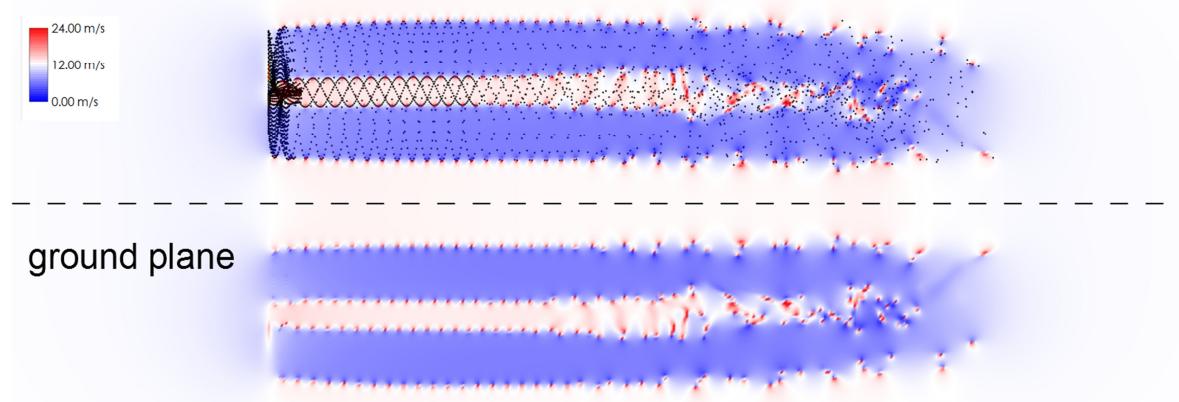


Figure 17. Vortex system mirrored at the ground plane; showing velocity distribution in the rotor mid-plane from side view

Include Tower Shadow:

When this switch is set to “On” a model for tower shadow effects is included in the simulation. The model implemented in QBlade is an adaption of the tower shadow model employed by AeroDyn and which is based on the work of Bak, et al [7]. This model is based on a superposition of the analytical solution for potential flow around a cylinder and a downwind wake model using a tower drag coefficient. The tower shadow model only affects velocity components in the x-y plane of the tower base coordinate system; the z-component of the velocity (perpendicular to the tower cross sections) remains unaffected. When the tower influence at an arbitrary point in space is calculated, the potential flow field of the tower is rotated such that its x-axis aligns with the respective velocity vector (free stream and induced

velocity at the point) at the evaluation point (Figure 18). It is then assumed that the velocity at the evaluation point is the uniform inflow velocity for the potential flow solution. For this solution we know the velocity distribution at all points in the tower model coordinate system. We then calculate the x- and y- velocity component at the evaluation point and transform this vector back into the global coordinate system. The tower model is only used when the z-component of the evaluation points position vector is smaller or equal to the tower height, this however introduces a discontinuity in the 3D velocity field of the simulation domain at the tower top (but its effect on simulation results usually is quite small).

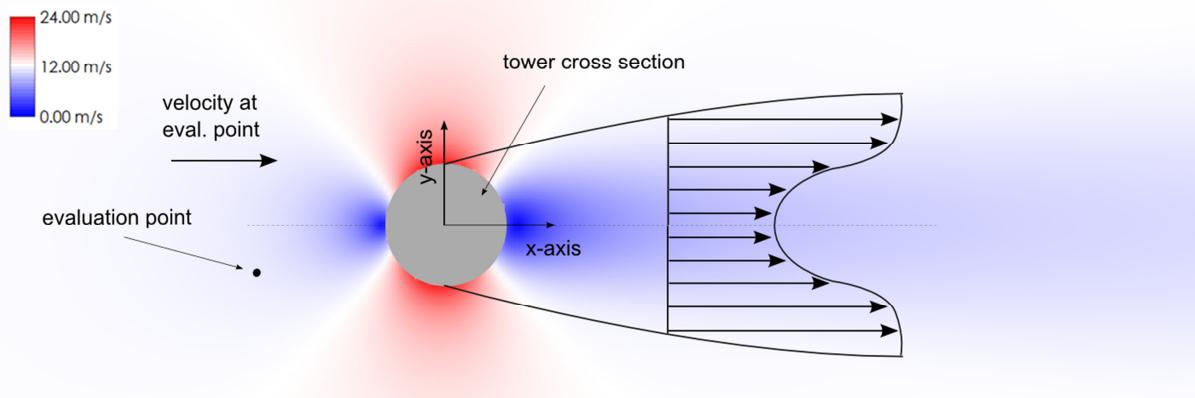


Figure 18. Schematic view of the tower shadow model, overlaid with velocity distribution from QBlade calculation, drawing reproduced from [5]

Tower Height (VAWT only) [m]:

For VAWT turbines this value defines the height of the tower. Contrary to a HAWT the tower height is not always the same as the hub (or rotor center) height. This means for VAWT that in one case the tower has an influence over the whole rotor height, in a different case only over its lower half (Figure 19).

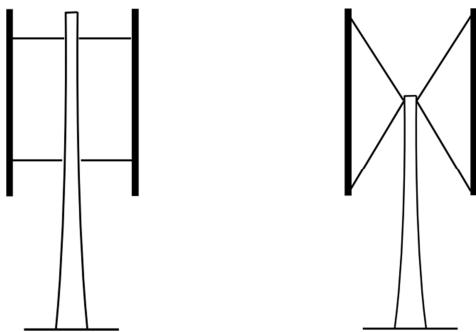


Figure 19. Different tower heights for VAWT rotors

Tower Top Radius/Tower Base Radius [m]:

The tower top radius and tower base radius define the tower radii at the respective position. Depending on the z-coordinate of the evaluation point a linear interpolation is used to obtain the tower radius for the tower shadow model.

Tower Drag Coefficient [-]:

This value defines the tower drag coefficient which is used to compute the wake deficit in the tower shadow model (see [5]). This value is more important for downwind or vertical rotor configurations.

Wake Modeling:

The modeling of the wake is highly important in a lifting line simulation. It affects both; the computational cost and the accuracy of a simulation. It is often possible to speed up the simulation by a factor of 10 or more without greatly affecting the accuracy. On the contrary in some situations (such as high induction cases at a high TSR) the wake discretization has a very large effect on the computed performance of the rotor. As the computational cost for the evaluation of one timestep is roughly proportional to N^2 , where N is the number of vortex elements, it is important to limit the maximum number of vortex elements in a simulation. In QBlade the wake is separated into three zones; the near field zone 1, where shed and trailing vorticity (as shed from the blades) exist; zone 2, from which the shed vorticity is removed and in which the trailing vorticity is concentrated into lumped vortices and zone 3, which has a coarser spatial discretization than zone 2. This zoning strategy is employed to enable a high accuracy of the near field wake and the rotor performance, while keeping the computational cost as low as possible. The parameters, governing the extensions of the three zones are: *max wake age*, *full wake length*, *fine wake length* and *wake thin factor* and are discussed in the following. The process that is employed to concentrate the trailing vorticity into lumped vortices is explained under the point “*Min. Gamma Fraction*” in this document.

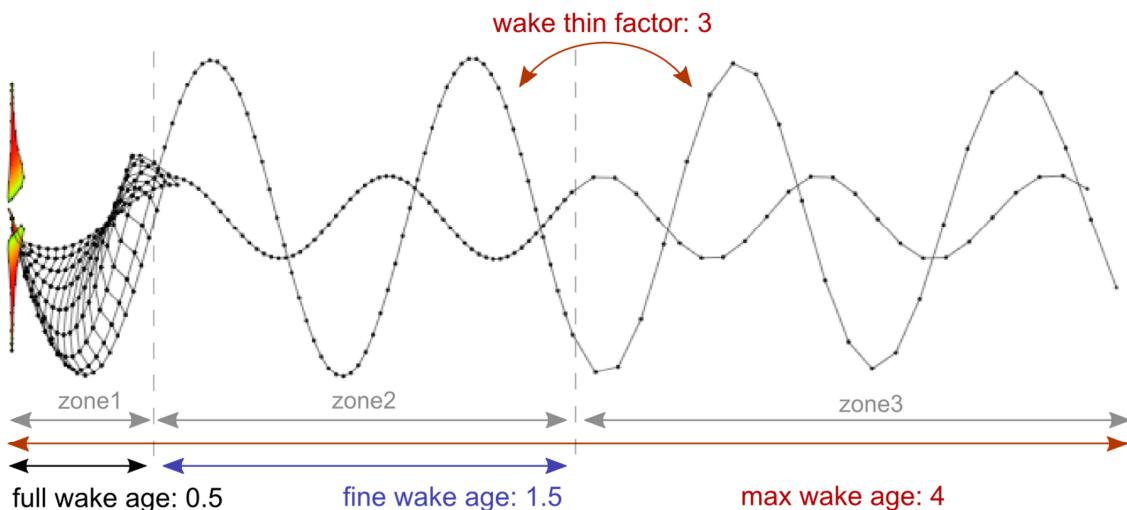


Figure 20: Example of wake zones and parameters with exemplary values given; single blade wake shown for overview

Wake Rollup:

If this switch is set to “On” the wake elements are convected with the local inflow velocity and the induction computed from all other wake elements. If set to “Off” free wake simulation is disabled and the wake elements are only convected by the inflow velocity.

Include Trailing Vortices:

If this switch is set to “On” trailing vorticity (oriented in stream wise direction) is shed from the trailing edges and included in the simulation.

Include Shed Vortices:

If this switch is set to “On” shed vorticity (oriented in span wise direction) is shed from the trailing edges and included in the simulation.

Turbulent Wake Convection:

If this switch is set to “On” the wake elements are convected with the local turbulent inflow velocity (in case a turbulent wind field is used). In this case the convection of the wake elements is also influenced by the tower shadow model. If the turbulent wake convection is switched “Off”, the wake elements are only convected with the mean inflow velocity.

Prescribed Wake:

If this switch is set to “On” all wake elements move on a straight, prescribed path that is parallel to the rotor axis and originates from their release point at the trailing edge. This also means that in this case the induction from other wake elements only has an effect in the direction of the prescribed path. This feature is intended to be used in case the wake is breaking down very early in a highly turbulent wind field.

Max Wake Age in Revolutions / in Timesteps [-]:

This value defines the maximum allowable age of a wake vortex element (see Figure 20). The age is defined dimensionless in rotor revolutions, starting from the time the vortex element is shed from the trailing edge of the blade. After a vortex element reaches the maximum age it is removed from the simulation. This parameter is highly important to limit the maximum number of vortex elements in the simulation which largely defines the computational time needed for a simulation. At the same time the accuracy of the computed rotor parameters and performance is affected by this value. By how much the accuracy is affected is a function of the TSR that is currently being simulated. The graph in Figure 21 shows that for higher TSR’s a larger number of rotor revolutions is needed. In the example shown a wake age of 14 rotor revolutions is needed at a TSR of 10 to reach a relative error of 1%. When a simulation input file (*.sim) is used during a simulation the maximum wake age cannot be defined in rotor revolutions (as the azimuthal discretization is not constant) and is defined as the maximum number of timesteps a wake element is allowed to exist. Make sure you adjust the value Max Wake Age value in this case, as the default value of 8 is far too small when the wake age is measured in timesteps.

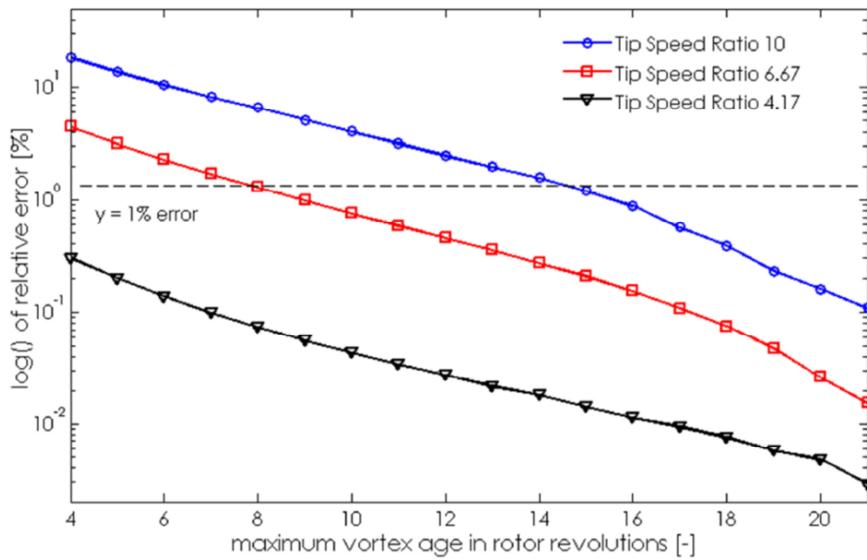


Figure 21. The effect of maximum wake age on accuracy (Cp Betz value was used as integral value to compare accuracy)

Full Wake Age in Revolutions / in Timesteps (HAWT only) [-]:

The parameter “full wake age” defines after how many rotor revolutions the shed vorticity is removed from the wake and the trailing vortices are concentrated into lumped vortices (see Figure 20). In highly unsteady simulations a “longer” full wake increases the accuracy, whereas in steady state simulations (such as Cp – TSR predictions) where the shed vorticity tends to zero a relatively short “full wake” length is sufficient. Similar to the parameter “max wake age” the “full wake age” influence on the accuracy of the result is also a function of the TSR (see Figure 22). The full wake age is usually defined in rotor revolutions, however when a *.sim file is used the age is defined in “number of timesteps”.

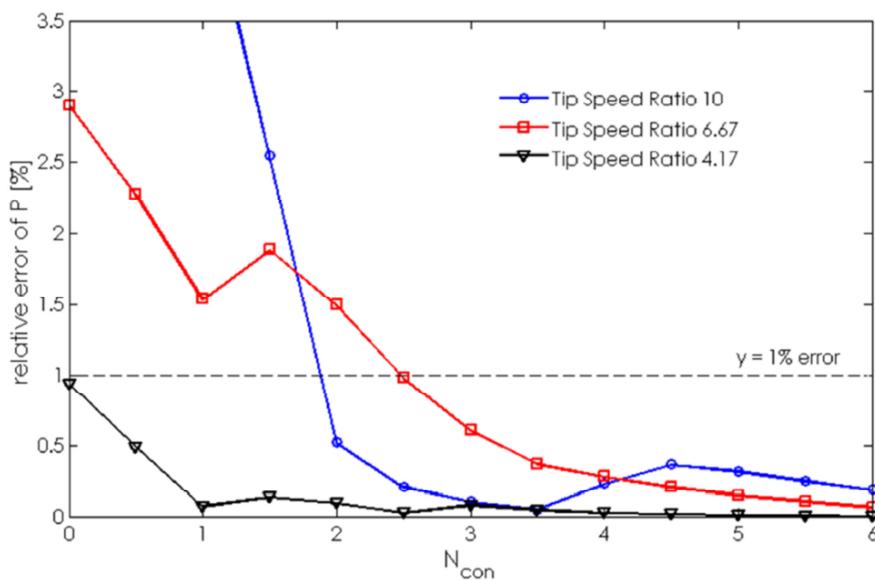


Figure 22. The effect of full wake age on accuracy (Cp Betz value was used as integral value to compare accuracy)

Fine Wake Age in Revolutions / in Timesteps (HAWT only) [-]:

The parameter “fine wake age” defines the length of zone 2 in Figure 20. The length of zone 3 (where the spatial discretization of the lumped trailing vorticity is coarser than in zone 2) is defined by subtracting the *full wake* and the *fine wake* length from the *total wake* length.

Wake Thin Factor (HAWT only) [-]:

The “wake thin factor” defines by how much the wake is thinned out in zone 3, resulting in a coarser spatial resolution of the wake and a smaller number of vortex elements in the far wake region. A factor of 4 means that the number of wake nodes in zone 3 is by a factor of 4 smaller than in zone 2 (or 4 vortex elements are combined into 1 between zone 2 and zone 3).

Min Gamma Fraction (HAWT) [-]:

To limit the number of free vortex elements in the wake, the trailing vorticity is concentrated into lumped vortices between zone 1 and zone 2. The strength of the lumped vortices is obtained by integrating the circulation of the wake sheet from local minima to local maxima, as shown in Figure 23.

$$\Gamma_{con} = \int_{x_{min}}^{x_{max}} \Gamma_{trail}(x) dx \quad (4)$$

In the illustrated case (double hump circulation distribution) this approach leads to four concentrated vortices. Their emission points are obtained by weighting the positions of the converted vortices (in the full wake sheet) by their contributing vortex strength. The circulation distribution of the wake sheet at the end of zone 1 can have very general shapes. If the distribution is only slightly jittering many local minima and maxima would be detected by the algorithm and no significant reduction of number of trailing vortex elements could be obtained. To prevent this, the parameter “min gamma fraction” defines the smallest strength that a concentrated trailing vortex is allowed to have. If the strength of the concentrated vortex is below this limit the vortex is lumped with the nearest concentrated vortex in its vicinity. The minimum strength that is allowed is defined as a fraction of the largest strength in the vortex sheet (usually the tip vortex), so a value of “min gamma fraction” of 0.1 would mean that the vortex strength of the weakest concentrated vortex cannot be smaller than 10% of the strongest vortexes strength. If “min gamma fraction” is continuously increased only the tip and the hub vortex will remain eventually, if its value is 0 no concentration of trailing vorticity will happen at all. Overall “min gamma fraction” can be seen to control the spatial resolution of the trailing vorticity in zone2.

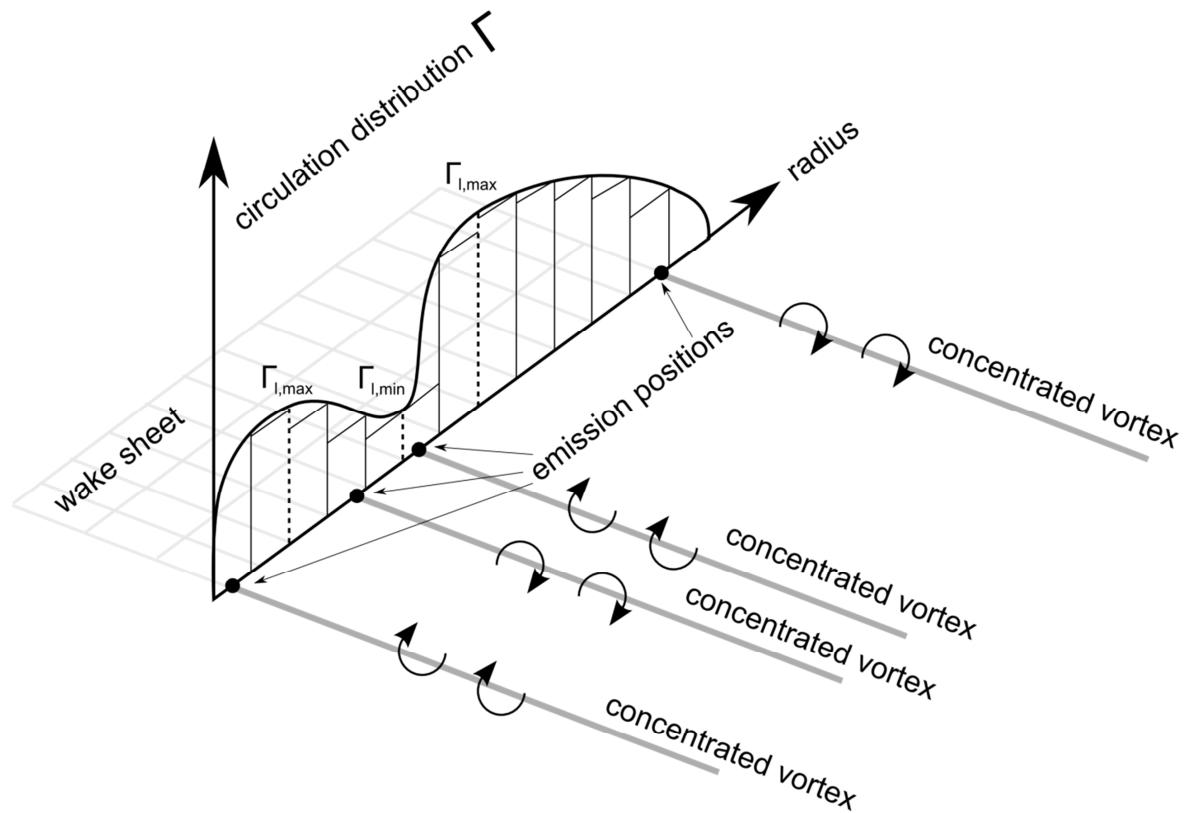


Figure 23. Illustration of the implemented approach for the concentration of trailing vorticity

Min Gamma Fraction (VAWT) [-]:

In case a VAWT rotor is simulated the far wake cannot be easily concentrated or simplified as both shed and trailing vorticity have an equally important influence on the flow field, even in a steady state simulation. The only strategy that is employed in this case, to reduce the number of vortex elements in the wake, is to limit the minimum vortex strength that is allowed to exist. As an example: If “Min Gamma Fraction” is set to 0.01, all wake vortices that have a vortex strength which is smaller than the strongest bound vortex (on the blades) multiplied by 0.01, are automatically removed.

First Wake Row Length Fraction [-]:

This sets the length of the first wake row in stream wise direction. A value of 0.1 would mean that the length of the first wake row is 10% of its normal length.

Vortex Modeling:

The Biot-Savart-Equation (Eq. 5) exhibits a singularity (see Figure 24) at the core where $\vec{r} = 0$:

$$\vec{u}_\Gamma(x) = \frac{\Gamma}{4\pi r_1 r_2 (r_1 r_2 + \vec{r}_1 \cdot \vec{r}_2)} (\vec{r}_1 + \vec{r}_2) (\vec{r}_1 \times \vec{r}_2) \quad (5)$$

To prevent this singularity from affecting the simulation, and also to model the viscous core of the bound and free vortices more accurately, a model for a viscous vortex core was implemented in QBlade. Many

different models to describe the tangential velocity distribution around the core exist, such as the Ramasay and Leishman model (Eq. 6):

$$\vec{u}_\Gamma(x) = K_v \frac{\Gamma}{4\pi r_1 r_2} \frac{(r_1 + r_2)(\vec{r}_1 \times \vec{r}_2)}{(r_1 r_2 + \vec{r}_1 \cdot \vec{r}_2)} \quad (6)$$

The viscous parameter K_v is defined as follows:

$$K_v = \frac{h^2}{(r_c^{2z} + h^{2z})^{1/z}}; \text{ with } h = \frac{|r_1 \times r_2|}{|r_2 - r_1|} \quad (7)$$

Van Garrel suggests in his report [4] to use a cut-off radius, that is simply added to the denominator of Eq.5 in the form of $(\delta l_0)^2$, where the length l_0 is the length of the vortex element and δ a fixed number between 0 and 0.1. This ensures that the induced velocity smoothly approaches zero in the vicinity of the core (Eq. 8). This is a very elegant and computationally efficient implementation, as the viscous core modeling is directly implemented in the calculation of the induced velocity. In QBlade the growth of the viscous vortex core is modeled and the core size r_c is used directly instead of a fixed length fraction δl_0 :

$$\vec{u}_\Gamma(x) = \frac{\Gamma}{4\pi r_1 r_2} \frac{(r_1 + r_2)(\vec{r}_1 \times \vec{r}_2)}{(r_1 r_2 + \vec{r}_1 \cdot \vec{r}_2) + r_c^2} \quad (8)$$

For other vortex models a viscous parameter (such as K_v in Eq.7) needs to be reevaluated whenever the Biot-Savart Equation is computed which is considerably increasing the computational cost. To ensure comparability to other codes both approaches are implemented in QBlade and the user has to choose one in the simulation setup. However due to considerable advantages in terms of computational efficiency it is suggested to use Van Garrel's approach.

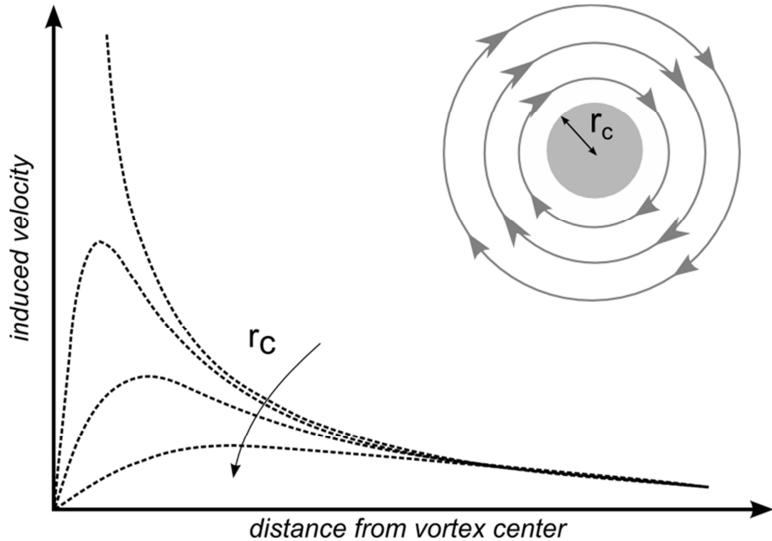


Figure 24. Schematic view of the influence of the core size on the tangential velocity induced by a vortex element

Both implementations require the radius of the viscous vortex core that is computed with the Lamb-Oseen vortex model, including filament straining, where Leishman added an additional parameter δ_v to model turbulent dissipation:

$$r_c = \left[\frac{4a \cdot \delta_v \cdot v \cdot (t_v + S_c)}{1 + \varepsilon} \right] \quad (9)$$

The two free parameters, the time offset parameter S_c and the turbulent viscosity δ_v are used to adjust the initial core size and its growth rate in time (see Figure 25).

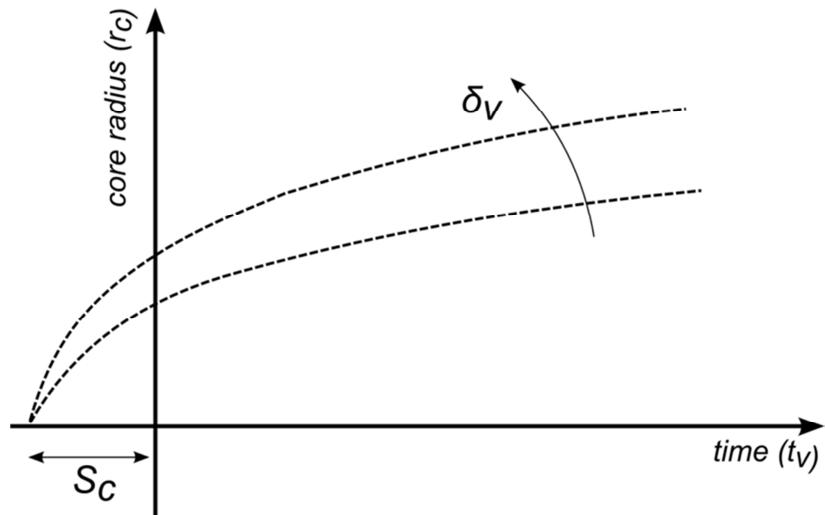


Figure 25. Schematic view of the influence of S_c and δ_v on the vortex core size

No empirical relationships to obtain values for these parameters exist to date. Sant [8] reports that experimental investigations suggest using a δ_v value of roughly 10 for small scale rotors and a value between 100-1000 for full scale rotors. The parameter S_c is needed to prevent vortices from having a zero core radius when being released from the trailing edge. These two parameters can have a significant effect on the computed rotor performance and choosing too large values for them reduces the induction in the near wake and leads to over predictions of rotor power output.

Desingularization Type:

Here the user can choose to either use Van Garrel's or Leishman's model.

Leishman z exponent [-]:

This defines the z-exponent of Eq.7. $z=1$ corresponds to the Scully model [9] and $z=2$ approximates the Lamb-Oseen vortex model [10].

Vortex Time Offset [s]:

This defines the S_c parameter in Eq.9.

Turbulent Vortex Viscosity [-]:

This defines the δ_v parameter in Eq.9.



Output Parameters:

Starting time to store output [s]:

Defines the starting time from which the output is stored (both visualization and performance parameters). This is to be used when the initial transient should not be present in the results of a simulation.

Algorithm Parameters:

Maximum number of iterations [-]:

This parameter defines the maximum number of iterations when calculating the bound circulation on the lifting line.

Relaxation factor [-]:

The relaxation factor r is in effect when updating the bound circulation at the blade during the iteration loop:

$$\Gamma_{i+1} = r \Gamma_i + (1 - r) \Gamma_{i-1}. \quad (10)$$

Epsilon [-]:

This parameter defines the convergence criterion. If the largest difference in bound circulation, between two successive iterations, is smaller than ε the iteration loop ends (if $(\Gamma_{i+1} - \Gamma_i) < \varepsilon$).

Environmental Variables:

Air Density [kg/m^3]:

This parameter defines the air density, used to compute forces and torque.

Kinematic Viscosity [m^2/s]:

This parameter defines the kinematic viscosity, used to compute the Reynolds number.

Validation

The HAWT part of QBlades Lifting Line module has been extensively validated against numerical and experimental data from the literature. The data used for the validation was obtained from the MexNext-I [11; 12] project, where numerical tools were compared to the MEXICO [13] experiment, and the NREL Phase VI [14] campaign. Overall the implementation in QBlade showed surprisingly accurate results and even outperformed some higher fidelity models. An excerpt of the HAWT validation is shown in the following graphs (Figure 26, Figure 27 Figure 28). Due to a lack of similar detailed and large scale test cases for VAWT turbines, a thorough validation of the VAWT part of the Lifting Line module has not yet been accomplished, however comparisons to DMST- and detailed 2D CFD simulations showed promising results and very good agreement in most cases. The VAWT validation part will be added to this document in the near future.

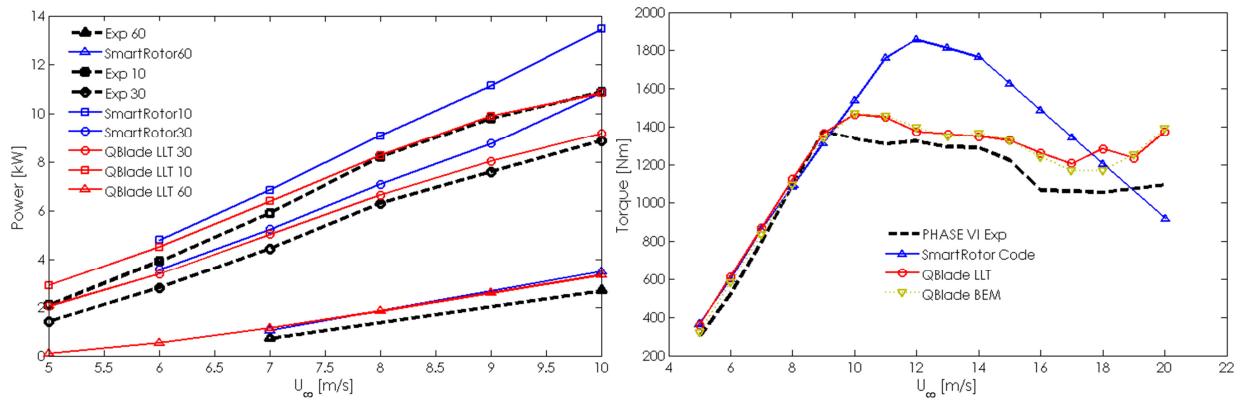


Figure 26. Validation with PHASE VI data and the SmartRotor code from [15]. Left: Power output for different yaw cases; right: power curve

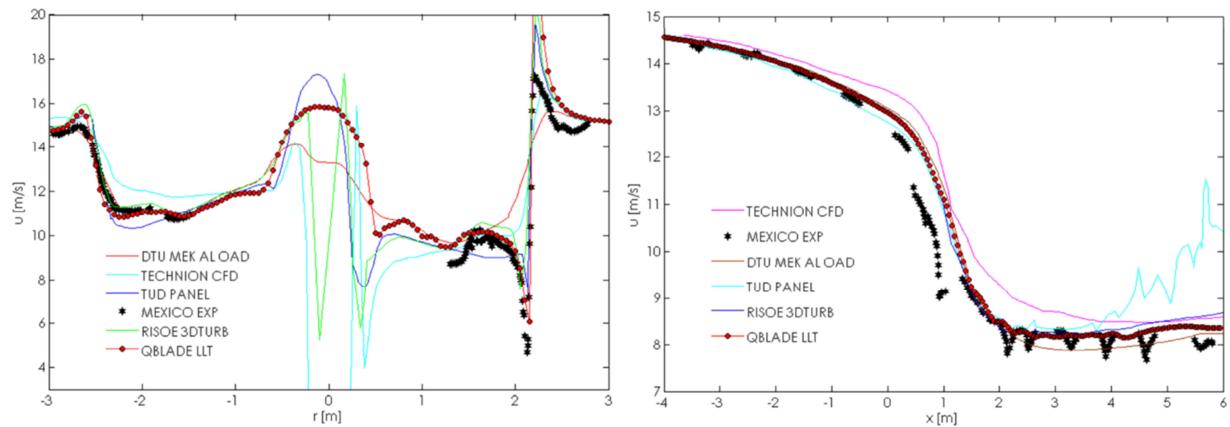


Figure 27. Validation with MexNext-I data from [12]. Left: radial traverse 0.15m behind and parallel to rotor plane showing axial velocity component under 30° yaw; right: axial traverse through rotor plane showing axial-component of velocity under 30° yaw

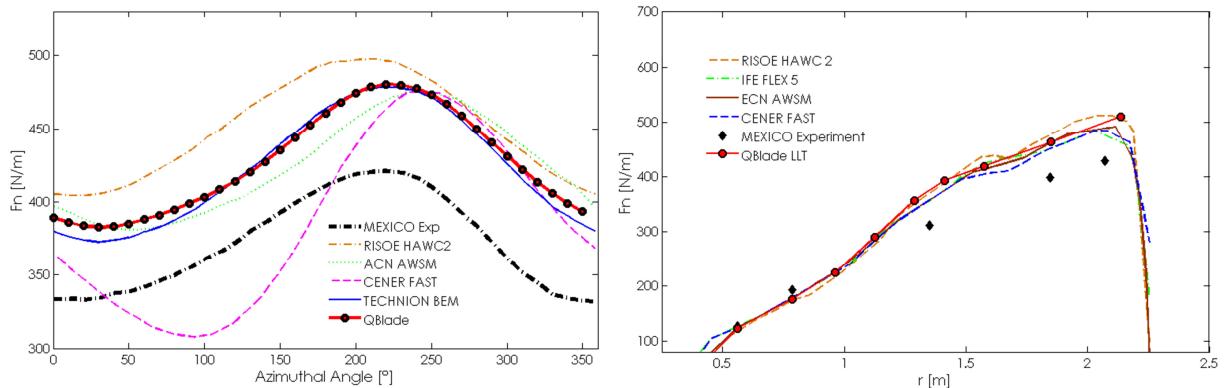


Figure 28. Validation with MexNext-I data from [12]. Left: azimuthal variation of normal force under 30° yaw; right: normal force variation over the blade radius

References

- [1] D. Marten, M. Lennie, G. Pechlivanoglou, C.N. Nayeri, C.O. Paschereit, „Implementation, Optimization and Validation of a Nonlinear Lifting Line Free Vortex Wake Module Within the Wind Turbine Simulation Code QBlade”, Proceedings of ASME Turbo Expo 2015, GT2015-43265, 2015
- [2] J. G. Schepers, “Engineering models in wind energy aerodynamics”, Doctoral Thesis, TU Delft, 2012.
- [3] M. Borg, A. Shires, M. Collu, “Offshore floating vertical axis wind turbines, dynamics modeling state of the art. part I: Aerodynamics”, Renewable and Sustainable Energy Reviews 39, 2014
- [4] A. Van Garrel, “Development of a wind turbine aerodynamics simulation module,” Technical Report, ECN, 2003
- [5] P. J. Moriarty and A.C. Hansen, “AeroDyn Theory Manual”, NREL Report, NREL/TP-500-36881, 2005
- [6] M. Kaltschmitt, W. Streicher , A. Wiese, "Renewable energy: technology, economics, and environment", page 55, Springer, 2007
- [7] C. Bak, H. Aagaard Madsen, J. Johansen, "Influence from blade-tower interaction on fatigue loads and dynamics (poster)." Proceedings of Wind Energy for the New Millennium, 2001
- [8] T. Sant, “Improving BEM-based aerodynamic models in wind turbine design codes,” Doctoral Thesis, TU Delft, 2007
- [9] M.P. Scully, J.P. Sullivan, “Helicopter Rotor Wake Geometry and Airloads and Development of Laser Doppler Velocimeter for Use in Helicopter Rotor Wakes,” Massachusetts Institute of Technology, Technical Report 183, MIT DSR No. 73032, 1972
- [10] P.G. Saffman, M.J. Ablowitz, J. Hinch, J.R. Ockendon, J. Peter, “Vortex dynamics”, p. 253 Cambridge University Press, 1992

- [11] J. G. Schepers, et al. *Final report of IEA Task 29, Mexnext (Phase 1)*”, Analysis of Mexico wind tunnel measurements, ECN-E-12-004, 2012
- [12] J. G. Schepers, K. Boorsma, X. Munduate, “Final Results from the Mexnext-I: Analysis of detailed aerodynamic measurements on a 4.5m diameter rotor placed in the large German Dutch Wind Tunnel DNW”, Proceedings: The Science of Making Torque from Wind 2012, Oldenburg, Germany
- [13] H. Snel, G. Schepers, and N. Sicama, “Mexico Project: the database and results of data processing and interpretation,” 47th AIAA Aerospace Sciences Meeting, pp. 1–12, 2009
- [14] D. Simms, S. Schreck, M. Hand, and L. J. Fingersh, “NREL Unsteady Aerodynamics Experiment in the NASA-Ames Wind Tunnel : A Comparison of Predictions to Measurements NREL Unsteady Aerodynamics Experiment in the NASA-Ames Wind Tunnel : A Comparison of Predictions to Measurements,” NREL/TP-500-29494, 2001
- [15] McTavish S., Feszty D., and F. Nitzsche, “Aeroelastic Simulations of the NREL Phase VI Wind Turbine using a Discrete Vortex Method,” Proceedings of the Canadian Aeronautics and Space Institute AERO Conference, 2009

List of QBlade Publications

QBlade general:

- D. Marten, G. Pechlivanoglou, C.N. Nayeri, C.O. Paschereit, “Integration of a WT Blade Design Tool in XFLR5”, Proceedings of DEWEK 2010, Bremen, Germany, 2010
- D. Marten, J. Wendler, G. Pechlivanoglou, C. N. Nayeri, and C. O. Paschereit, “QBlade : An Open Source Tool For Design And Simulation Of Horizontal And Vertical Axis Wind Turbines,” IJETAE, 3 special issue, pp. 264–269, 2013
- D. Marten, J. Wendler, G. Pechlivanoglou, C.N. Nayeri, C.O. Paschereit, “Development and application of a simulation tool for vertical and horizontal axis wind turbines,” in Proc. of ASME Turbo Expo, GT2013-94979, 2013

Lifting Line:

- D. Marten, M. Lennie, G. Pechlivanoglou, C.N. Nayeri, C.O. Paschereit, „Implementation, Optimization and Validation of a Nonlinear Lifting Line Free Vortex Wake Module Within the Wind Turbine Simulation Code QBlade”, in Proc. of ASME Turbo Expo 2015, GT2015-43265, 2015

Structural Blade Model:

- M. Lennie, D. Marten, G. Pechlivanoglou, C.N. Nayeri, C.O. Paschereit, „ Development and Validation of a Modal Analysis Code for Wind Turbine Blades”, in Proc. of ASME Turbo Expo 2014, GT2014-27151, 2014



Application:

- S. Niether, B. Bobusch, D. Marten, G. Pechlivanoglou, C.N. Nayeri, C.O. Paschereit, „Development of a Fluidic Actuator for Adaptive Flow Control on a Thick Wind Turbine Airfoil”, Journal of Turbomachinery 06/2015, DOI:10.1115/1.4028654, 2015
- D. Marten, H. Spiegelberg, G. Pechlivanoglou, C.N. Nayeri, C.O. Paschereit, C. Tropea, „Configuration and Numerical Investigation of the Adaptive Camber Airfoil as Passive Load Alleviation Mechanism for Wind Turbines”, 33rd AIAA Applied Aerodynamics Conference 2015, Dallas, Texas, 2015