Bladed "Project Info" features

SUMMARY

Some Bladed features are not yet available through the User Interface and are instead accessed through the "Project Info" screen in Bladed. This document summarises the syntax required to activate and control these extra features.

Version: O

Date of issue: 14/06/2023

Doc No: 110052-UKBR-T-29-O

Contents

SUMMARY 1

1 INTRODUCTION 4
1.1 Entering Project Info Data 4

2 HYDRODYNAMICS 6
2.1 Heave Plates 6
2.1.1 Bladed 4.6.0.91 and earlier 6
2.1.2 Bladed 4.6.0.92 and later 6
2.2 User-defined direction shear for currents 6
2.3 Anisotropic C_D and C_M 7
2.4 Custom wave-train extent 8
2.5 Global viscous drag 8
2.6 Global linear drag 8
2.7 Timestep for Morison calculation with SEA file 9
2.8 Use total acceleration (convective and time derivative) for Morison loading when using stream function 9
2.9 Adjusting translational drift 9

3 STRUCTURAL 11
3.1 Rigid members 11
3.2 Local Joint Flexibilities 11
3.3 Pre-tensioned Elements 11
3.4 Torsion Bending Coupling 12
3.5 Rotate linear foundations with tower "X-axis clockwise from South" 13
3.6 Removing Modes from Simulations 13
3.7 Multi-part Blade model 13
3.8 Blade geometric stiffness 13
3.9 Non-linear tower damper 14

4 MECHANICAL SYSTEMS 15

4.1 Ailerons 15
4.2 Generator Brake 16
4.3 Pitch rate dependent torque limits 16
4.4 Speeding up simulations slowed by stick-slip 16
4.4.1 Yawing against friction 16
4.4.2 Shaft rotation against the brake 17
4.5 Yaw quadratic damper 17
4.6 Yaw Friction Dependent on Bearing Loads 17
4.7 Additional teeter restraint models 18
4.7.1 Non-linear teeter stiffness 18
4.7.2 Non-linear teeter damping 18
4.7.3 Teeter Friction and Stiction 19
4.7.4 Combining teeter restraint functions 19
4.7.5 Teeter offset and hub centres of mass 20
4.8 Pendulum tower damper functionality 22
4.9 Ignore damper in mode shapes 22

5 ELECTRICAL SYSTEMS 23
5.1 Generator Cogging Torque 23
5.2 Speed Dependent Electrical Losses 23

6 CONTROL SYSTEMS 24
6.1 Strain Gauges 24
6.1.1 Strain Gauges on Tower 24
6.1.2 Strain Gauges on Blades 25
6.2 Accelerometers 26
6.2.1 Accelerometers on Tower 26
6.2.2 Accelerometers on the Nacelle 26
6.2.3 Accelerometers on Blades 27
6.3 Delaying the start-up sequence 27
6.4 Use Legacy DISCON 27
6.5 Tower inclinometers 27
6.6 Blade aerodynamic information 28

7 MULTIPLE ROTOR FUNCTIONALITY 29
7.1 Initial azimuth angle (power production) 29
7.2 Rotor rotational directions 29
7.2.1 Inputs for Project Info 29
7.2.2 Sign convention for counter-rotating rotors 29

8 OUTPUTS 30
8.1 Blade acceleration output 30
8.2 Blade refined deflection output 30
8.3 State values output 30
8.4 Tower buoyancy 30
8.5 Avoiding interpolation in post-processing 31

9 ENVIRONMENT 32
9.1 Evolving turbulence (for Lidar) 32
9.2 Gust Propagation 32

9.3 Wind Dll 32
9.4 Meandering Wake 32
9.4.1 Generating low-pass filtered wind file 33
9.4.2 Time domain simulation 33

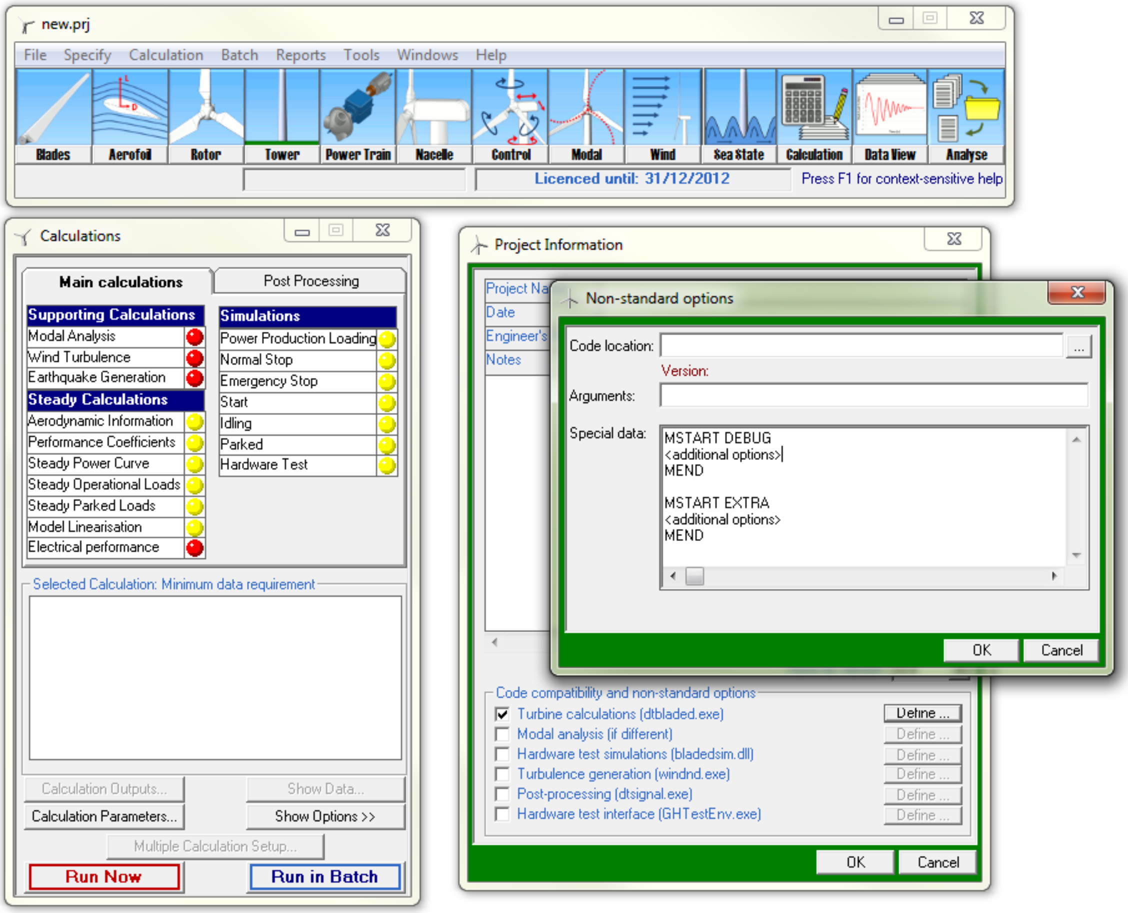
10 SIMULATION CONTROL....34
10.1 Solve_system iterations and tolerance 34

11 AERODYNAMICS CONTROL....35
11.1 Lift slope model in old aerodynamics 35

1 INTRODUCTION

1.1 Entering Project Info Data

The Bladed "Project Info" screen allows options to be specified which are either in development and do not yet have an associated GUI screen.
These "Project Info" options can be activated by going to File → Project Info… and then ticking "Turbine calculations (dtbladed.exe) and then clicking "Define":



All three boxes on this "Non-standard options" dialog box are optional, so the "Special data" can be provided without specifying a new executable location or additional arguments. The "Special data" field is a free-form text box, the contents of which appear within the bulk of the DTBLADED.IN file (the input file for the dtbladed executable).

Most options in this document are added in a module called "EXTRA". Entries are added as part of this module in the following manner

```
MSTART EXTRA
…
MEND
```

The extra module contains miscellaneous additional options, each described here in their own section. If many options are added in the same MSTART-MEND block, then the order of these sections is critical, as the file gets read sequentially.

Generally, several MSTART EXTRA modules can be used together in Project Info for increased clarity. In this case, the data groups can be added in any order.

```
MSTART EXTRA
Use Discon
MEND

MSTART EXTRA
Strain Gauges on Tower
MEND
```

Code in the input data can be commented out by using the asterisk (*). Any characters on a line after an asterisk will not be read in. This has been used extensively in this document to describe each line.

Each block has been written in such a way that they can be copied directly into the special data box in Bladed. Some other text editors will add new lines when you paste the data.

When there is a line that requires many values to be entered, they may be space, tab or comma separated. No other delimiters are accepted by the parser.

2 HYDRODYNAMICS

2.1 Heave Plates

Versions: 4.2 and later.

Heave plates represent members with axial drag and fluid inertial properties (for default Morison equation, the drag and inertia forces act only normal to the member axis).

2.1.1 Bladed 4.6.0.91 and earlier

In Bladed 4.6.0.91 and earlier, the following code is used to add Heave Plates.

```
MSTART EXTRA
NHeavePlates    * Number of heave plates
HeavePlates     * A comma separated list of indexes of the members that are
representing heave plates as they appear in the user interface screen
HeavePlateCm    * A comma separated list of hydrodynamic inertia coefficient
for each member
HeavePlateCd    * A comma separated list of drag coefficient for each member
MEND
```

The Cd is based on the plan area of the element (i.e. the length multiplied by the diameter), and the Ca is referenced to the volume of the element.

2.1.2 Bladed 4.6.0.92 and later

From Bladed 4.6.0.92, the Froude-Krylov force is not included on heave plates. This is because the Froude-Krylov force is already accounted when applying forces due to dynamic pressure from wave kinematics. Heave plates specified in the project info are only included when using one of the following hydro loading models: “Morison’s Equation”, “BEM Hydrodynamics (with Morison drag)” or if the heave plate element is a Morison element in the "Specify different model for each member". Only the contribution due to quadratic drag (HeavePlateCd values) are included when using “BEM Hydrodynamics (with Morison drag)”.

The following code should be used to add Heave Plates

```
MSTART EXTRA
NHeavePlates    * Number of heave plates
HeavePlates     * A comma separated list of indexes of the members that are
representing heave plates as they appear in the user interface screen
HeavePlateCa    * A comma separated list of added mass coefficient for each
member
HeavePlateCd    * A comma separated list of drag coefficient for each member
MEND
```

The Cd is based on the plan area of the element (i.e. the length multiplied by the diameter), and the Ca is referenced to the volume of the element.

Note that when upgrading from a pre-4.6.0.92 version, it will be necessary to subtract 1 from the HeavePlateCm values to get the correct HeavePlateCa values.

2.2 User-defined direction shear for currents

Versions: 4.7 and later

In 4.7 and later, this allows the user to apply a depth-dependent offset to the subsurface current direction. It is specified as a lookup table, with direction offset values being plotted against height above the seabed, normalised to mean water depth.

Linear interpolation is used to extract the values, which are then added on to the current direction before it is used in the calculations.

```
MSTART EXTRA
USERCURRDIROFFSETPTS        *Number of points in the table
USERCURRDIROFFSETHT         *A space-separated list of (height divided by
depth) values
USERCURRDIROFFSET           *A space-separated list of direction offsets, in
radians
MEND
```

2.3 Anisotropic C_D and C_M

Versions: 4.5 and later

On support structure members, C_D and C_M are usually the same in the member Y and Z directions. It's possible to specify different values of C_D and C_M in the Y and Z directions, using the following inputs.

```
MSTART EXTRA
N_ANISOTROPIC_CDS           * number of members on which to specify y and z
drag coefficients
ANISOTROPIC_CDS             * list of member numbers on which y and z drag
coefficients are specified
ANISOTROPIC_CDY             * drag coefficient in local member y direction at
end 1 and end 2 of each member {(mem1, end1), (mem1, end2), (mem2, end1)…….}
- there should be double the number as in row above
ANISOTROPIC_CDZ             * drag coefficient in local member z direction at
end 1 and end 2 of each member
N_ANISOTROPIC_CMS           * number of members on which to specify y and z
hydrodynamic inertia coefficients
ANISOTROPIC_CMS             * list of member numbers on which y and z
hydrodynamic inertia coefficients are specified
ANISOTROPIC_CMY             * hydrodynamic inertia coefficient in local
member y direction at end 1 and end 2 of each member {(mem1, end1), (mem1,
end2), (mem2, end1)…….} - there should be double the number as in row above
ANISOTROPIC_CMZ             * hydrodynamic inertia coefficient in local
member z direction at end 1 and end 2 of each member
MEND
```

2.4 Custom wave-train extent

Versions: 4.3.0.63 and later revisions; 4.4 and later

For floating turbines that move a large horizontal distance during the simulation, resulting in the “Turbine has floated beyond calculated wave train” message. It is possible to define your own wave train extent, as follows.

```
MSTART EXTRA
WaveExtent     150.0       * Extent of wave train in metres (upstream and
downstream)
MEND
```

The custom wave train will extend by at least this distance, both upstream and downstream. This value will only be applied, in either direction, if its absolute size in that direction is greater than the default wave train extent calculated by DTBladed.

2.5 Global viscous drag

Versions: 4.6. This feature is in the User Interface from Bladed 4.7

This is intended for use in conjunction with BEM hydrodynamics, because there is no viscous damping term in BEM. Specify one or more matrices of drag coefficients, each associated with a tower node. For instance, one for each BEM body if there are several. Water motion is not taken into account. Both velocities and loads are in global coordinates. All off-diagonal elements in the matrices are assumed zero.

```
MSTART EXTRA
NDRAGBODIES 2         *number of drag matrices to be specified
BODYNODES 4 5         *tower nodes at which drag forces are to be applied
DRAGCOEFFS 1.3 4.55 6.3 54.45 77.4 76.34      *each row is a separate set of
6 values
55.2 433.45 8.45 5.34 44.2 66.7               *which are the 6 elements on
the diagonal of the matrix
MEND
```

2.6 Global linear drag

Versions: 4.6. This feature is in the User Interface from Bladed 4.7

This is analogous to the global viscous drag feature, with the only difference being that the loads are linear multiples of the structural node’s velocity, not quadratic. As for viscous drag, specify one or more matrices of drag coefficients, each associated with a tower node. For instance, one for each BEM body if there are several. Water motion is not taken into account. Both velocities and loads are in global coordinates. All off-diagonal elements in the matrices are assumed zero.

```
MSTART EXTRA
NDRAGBODIES_L 2 *number of linear drag matrices to be specified
BODYNODES_L 4 5 *tower nodes at which linear drag forces are to be applied
DRAGCOEFFS_L 1.3 4.55 6.3 54.45 77.4 76.34 *each row is a separate set of 6
values
55.2 433.45 8.45 5.34 44.2 66.7 *which are the 6 elements on the diagonal of
the matrix
MEND
```

2.7 Timestep for Morison calculation with SEA file

When using a SEA file, the Morison hydrodynamic forces are calculated on a fixed timestep of 0.05s, as this calculation is computationally expensive. An alternative timestep for this calculation can be specified with the following code

```
MSTART EXTRA
HydroTimeStep 0.01          * Time in seconds
MEND
```

2.8 Use total acceleration (convective and time derivative) for Morison loading when using stream function

Versions: 4.7 and later

Total acceleration is the sum of the time derivatives and a convective term:

$$\frac{Du}{Dt} = \frac{\partial u}{\partial t} + (u \cdot \nabla)u$$

The water particle accelerations in Bladed only include the time derivative terms by default. The convective terms can be added when using the stream function for the water particle acceleration calculations, by adding the text below into project info.

If using irregular waves and a constrained wave is applied defined by a stream function then it is not recommended to turn total acceleration on in this way. This is because the water particle acceleration will be calculated using time derivatives for the irregular waves and total acceleration for the constrained waves and would therefore be inconsistent.

```
MSTART EXTRA
TotalAccWaveNonLinear 1     * 0: off (default) i.e. time derivative term only
                            * 1: on i.e. total acceleration
MEND
```

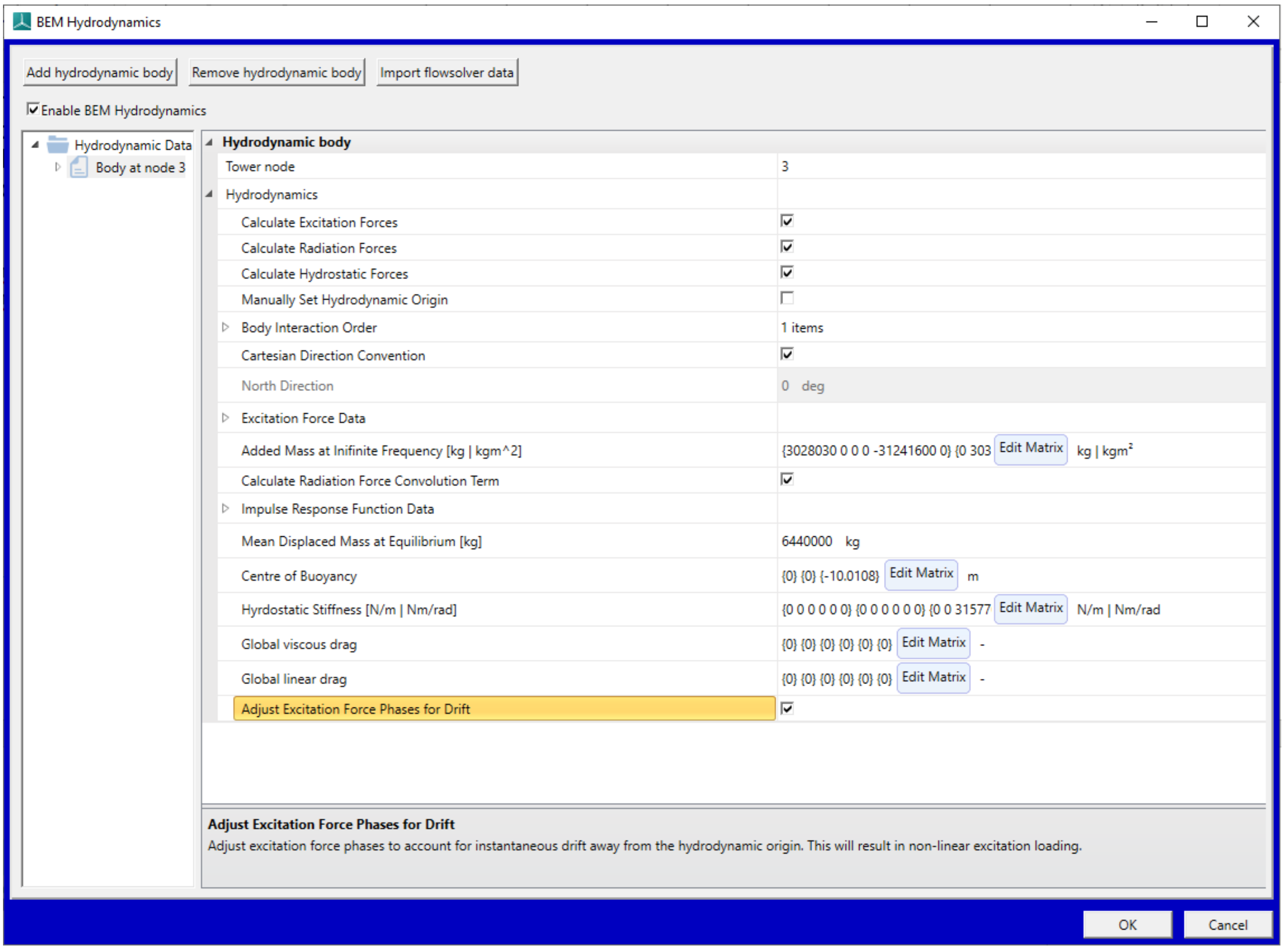
2.9 Adjusting translational drift

Versions: 4.11 and earlier

BEM hydrodynamic data that are imported to Bladed are defined in the origin of the hydrodynamic coordinate system. When a floating structure is exposed to a wave, the structure is drifted off the origin. The drift generates a phase lag in the calculation of excitation force (see theory manual for more information). The phase lag is calculated as k*d where k is wave number, and d is the length of the drift which is a distance between hydrodynamic origin and the position of structure at each time step. The instantaneous position of the drifted structure is not smooth and needs to be adjusted. Hence, to filter out high-frequency data to obtain a smooth time series of the drift position, a two-parameter low pass filter can be introduced to Bladed using the following project info code:

```
MSTART EXTRA
WaveDriftParamZeta 0.707
WaveDriftParamOmega 0.10472
MEND
```

To activate this feature, in BEM hydrodynamics screen, the option "*Adjust Excitation Force Phases for Drift*" needs to be ticked (highlighted in the following picture).



It is worth noting that setting *WaveDriftParamZeta* and *WaveDriftParamOmega* using the above code is not supported from Bladed 4.12 and onward. Instead, from Bladed 4.12 and onward, these parameters will be set in the BEM hydrodynamics screen.

3 STRUCTURAL

3.1 Rigid members

Versions: 4.0 and later

Bladed modal analysis cannot cope with a support structure which contains too large a variation in element stiffness. It is not recommended that users model rigid members by entering a very large stiffness value. It is better to make some of the stiffer elements rigid.

```
MSTART EXTRA
NRM 6                           * number of rigid members
RIGID 36 37 38 39 40 41         * member numbers that are rigid
MEND
```

3.2 Local Joint Flexibilities

Versions: 4.0 to 4.12. Not available in 4.13+

Local joint flexibilities can be defined based on the Buitrago's equations described in Appendix B of the DNV-OS-J101 standard and then implemented in Bladed through the Special Data window in Project Info. For details about the implementation in Bladed, please refer to document "Local Joint Flexibility" on the Bladed User Portal in the "Engineering Documentation" section.

Versions: 4.13 and later

Local joint flexibilities can be modelled using the Hinges Feature which allows the user to define 6DOF spring stiffness at each member end, for each member in the multi-member tower. The spring stiffness can be defined based on the Buitrago's equations described in Appendix B of the DNV-OS-J101 standard. It is recommended to define the Hinges in Sesam and export the model to a Bladed project file. Sesam Wind Manager 6.0+ is required to include the hinges data in a jacket structure model. Please refer to the bladed user manual section '11.3 Support structure hinges' for full details on the bladed hinges feature.

3.3 Pre-tensioned Elements

Versions: 4.0 and later

Pre-tension can be added to support structure beam or bar elements in Bladed. If an element represents a wire, it must also be assigned as a bar element (no bending, shear or torsional stiffness).

If two or more pre-tensioned beam elements are included end to end in the model, the pre-tension must only be applied to one of the elements (otherwise it will be double counted).

Cables should be modelled using a single bar element.

In the example below, there are two wires, each consisting of one element

```
MSTART EXTRA
NumBarElements 2
BarElements 40,41
NumPreTensionEls 2
PreTensionEls 40,41
PreTension 5000, 5000
MEND
```

3.4 Torsion Bending Coupling

Versions: 4.4.0.108 and later revisions; 4.5 and later

Extra coupling terms can be included in the constitutive matrix that couples bending and torsional moment to bending and torsional strain. Without bend-twist coupling, the relationship between bending strains and moments are as follows. Note that the coordinate system is the element coordinate system (i.e. x along the element axis).

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} GI & 0 & 0 \\ 0 & EI_y & 0 \\ 0 & 0 & EI_z \end{bmatrix} \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_z \end{bmatrix}$$

The bend-twist coupling terms are added to this matrix as shown here

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} GI & C_{xy} & C_{xz} \\ C_{xy} & EI_y & C_{yz} \\ C_{xz} & C_{yz} & EI_z \end{bmatrix} \begin{bmatrix} \kappa_x \\ \kappa_y \\ \kappa_z \end{bmatrix}$$

So, there are extra coupling that mean that a bending strain gives a torsional moment and vice-versa.

Fill in the following text in project info to enable torsion bending coupling. You need enough entries for the number of elements in the blade – each node in the Bladed user interface defines the properties for both the end of one element and the beginning of the next, unless it is a split station so there must be 2N-2 entries.

```
MSTART EXTRA
TorsionEdgeCStiff                  * list of Cxz values
TorsionFlapCStiff                  * list of Cxy values
FlapEdgeCStiff                     * list of Cyz values
MEND
```

For example, for the demo_a turbine (which has 10 blade stations):

```
MSTART EXTRA
TorsionEdgeCStiff 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0
10000.0 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0
10000.0 10000.0
TorsionFlapCStiff 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0
10000.0 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0
10000.0 10000.0
FlapEdgeCStiff 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0
10000.0 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0 10000.0
10000.0 10000.0
MEND
```

3.5 Rotate linear foundations with tower "X-axis clockwise from South"

Versions: 4.4 and 4.5. In 4.6 and later it is in the tower screen

When using the "X-axis clockwise from South", the support structure is rotated about the global Z axis, but the foundations are not. This is potentially a problem if the foundations are not rotationally symmetric (i.e. if the "Symmetrical in X and Y" check-box is not selected in the foundation screen).

Linear foundation stiffness and mass matrices can also be rotated according to the "X-axis rotated from South" variable. The extra line in Project Info to enable this is

```
MSTART EXTRA
ROTATELINEARFOUND 1          * 0 = off, 1 = on
MEND
```

Note that you can use ROTATELINEARFOUND 0 to disable the rotation, or just omit the line.

Note that this flag will not have any effect for foundation stiffness p-y look-up tables, which cannot be rotated simply. The support structure "X-axis rotate from South" functionality should therefore be used with extra care if using non-linear foundations.

3.6 Removing Modes from Simulations

Versions: 4.0 to 4.7. Not available in 4.8+

It is not possible through the UI to exclude modes at a lower frequency than the highest frequency mode that is being modelled. Lower frequency modes can be excluded via the following code

```
MSTART EXTRA
NBladeModeToExclude 2             * number of blade modes to exclude
BladeModesToExclude 3 4           * blade mode numbers to exclude
NTowModeToExclude 1               * number of tower modes to exclude
TowModesToExclude 1               * tower mode numbers to be excluded
MEND
```

3.7 Multi-part Blade model

Versions: 4.7

In Bladed 4.7 and later it is possible to allow the blade to be split into several different linear FE bodies to give a non-linear blade deflection model.

For details about the implementation in Bladed, please refer to document "Multi-part Blade, User Guide for Bladed 4.7" on the Bladed User Portal in the "User Guides" section.

3.8 Blade geometric stiffness

Versions: 4.3 to 4.6. In 4.7, it is in the blade screen

Geometric stiffness attempts to account for the change in blade or tower structural properties due to deflection.

For details about these options in Bladed, please refer to document "Geometric stiffening options in Bladed" on the Bladed User Portal in the "User Guides" section.

3.9 Non-linear tower damper

Versions: 4.3.0.92 and later revisions of 4.3 only. Not available in other Bladed versions.

The passive linear vibration damper on the tower can be replaced with a non-linear definition where stiffness and damping vary according to damper displacement. The user must specify the damper in the normal way in the UI, but the frequency and damping factor inputs will not be used.

The damper is defined in radial coordinates where the displacement is the radial coordinate from the damper equilibrium position. The elastic force and damping coefficient definition is isotropic (i.e. the same at all angles). Elastic force is defined as positive for a positive displacement.

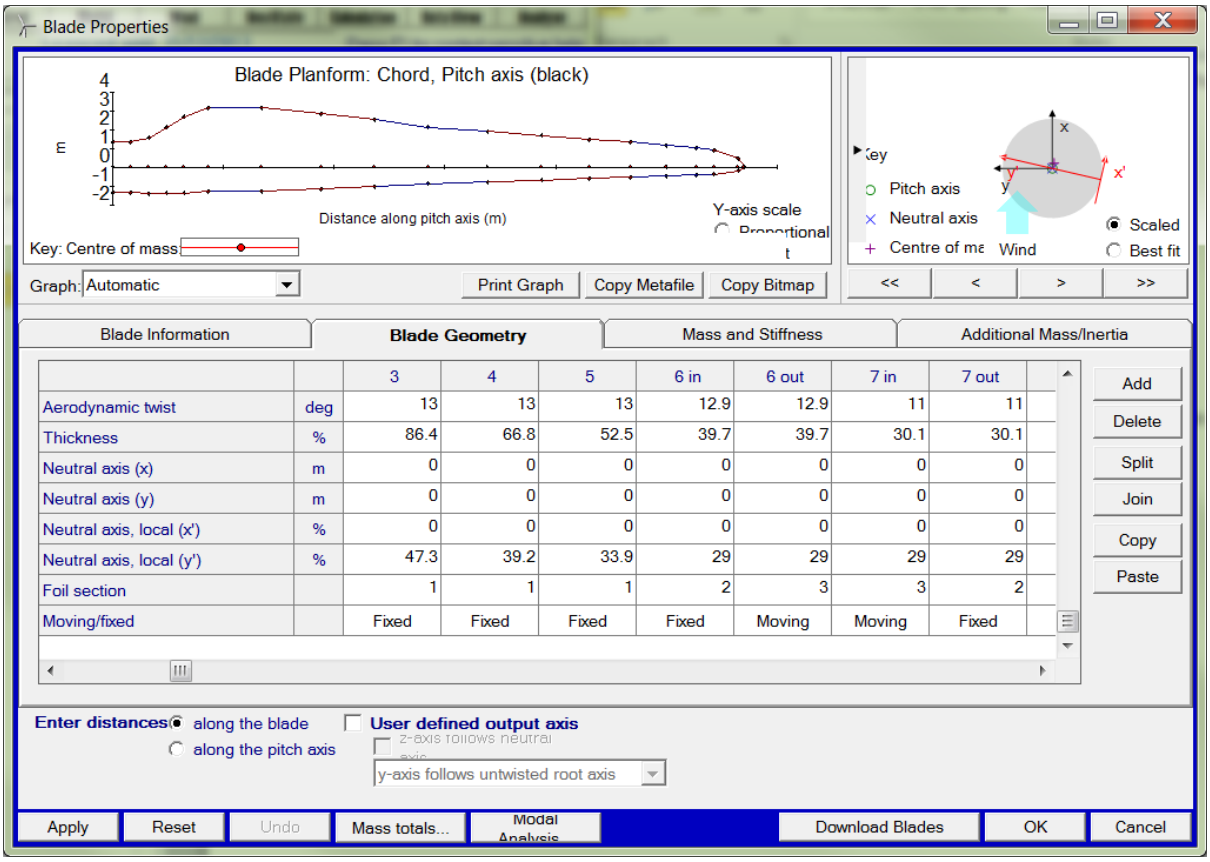
```
MSTART EXTRA
NonLinTowDampPoints         * number of points in look-up table
NonLinearDisps              * displacement values for look-up table
NonLinearElasticForce       * elastic force values for each displacement
NonLinearDamping            * damping values for each displacement
MEND
```

4 MECHANICAL SYSTEMS

4.1 Ailerons

Versions: 4.0 and later

Ailerons can be added via the Blade planform screen:



An aileron is denoted as being a succession of "Moving" sections between one or more "Fixed" sections. It is not currently possible to have two adjacent ailerons.
Project Info can be used to provide additional information about the ailerons:

```
MSTART EXTRA
PitchAndAileron     1        * 1 means whole blade pitches according to pitch
demand, and ailerons move additionally according to aileron angle demand
Angle           0            *Initial or steady state aileron angle (rad)
(default = 0)
AileronActuator       1      *Default 0 for instantaneous response
*Only if AileronActuator=1, we then need the transfer function for the
aileron response:
*Example for first-order response, time constant 0.1s:
Aileron_NUMORD 0             * order of transfer function numerator
Aileron_NUM    1             * numerator values of transfer function
Aileron_DENORD 1             * order of transfer function denominator
Aileron_DEN    0.01 1        * denominator values of transfer function
MEND
```

4.2 Generator Brake

Versions: 4.0 and later

There is an option to add a generator brake, e.g.

```
MSTART EXTRA
GENBRAKE_TYPE  1      * 0 = off, 1 = look-up table
GENBRAKE_N            * Number of points in the speed/torque look-up table
* This block must be repeated for every point in the look-up table so that
there are GENBRAKE_N speed/torque pairs
GENBRAKE_SPEED
GENBRAKE_TORQUE
* end of block
GENBRAKE_NUMORD       * order of transfer function numerator
GENBRAKE_NUM          * numerator values of transfer function
GENBRAKE_DENORD       * order of transfer function denominator
GENBRAKE_DEN          * denominator values of transfer function
GENBRAKE_RATLIM       * maximum generator speed at which torque is applied
MEND
```

4.3 Pitch rate dependent torque limits

Versions: Bladed 4.3 and earlier. In the UI in 4.4 and later

Note that the code below does not work in Bladed versions 4.4 and later. These limits are now included in the UI.

Here is an example of code used to implement a pitch rate - torque curve of a pitch actuator in Bladed project info. This can be used when actuator details have been defined in the interface. The rate which is referred to is the actuator rate, i.e. the pitch rate multiplied by pitch actuator gear ratio.

```
MSTART EXTRA
ACTTRQLIM_ENTRIES          * number of entries in look-up table
ACTTRQLIM_RATES            * list of actuator rates in look-up table
ACTTRQLIM_TRQMINS          * list of minimum torques
ACTTRQLIM_TRQMAXS          * list of maximum torques
MEND
```

4.4 Speeding up simulations slowed by stick-slip

Versions: 4.0 and later

Sometimes Bladed runs can be very slow due to the stick-slip model causing excessive re-steps. This happens sometimes when a certain modelled friction surface oscillates between a 'stuck' and 'slipping' status. When determining a change from a 'slipping' status to a 'stuck' status, a rate tolerance is used where the contact can only become 'stuck' if the rate is below this value. The size of these rate bands have sensible defaults but can be changed in project info.

4.4.1 Yawing against friction

Relevant only if yaw friction and stiction are non-zero.

```
MSTART TOLER
ABSTOL_YAWRATE 0.01
MEND
```

4.4.2 Shaft rotation against the brake

```
MSTART TOLER
ABSTOL_ROTSPD 0.01
MEND
```

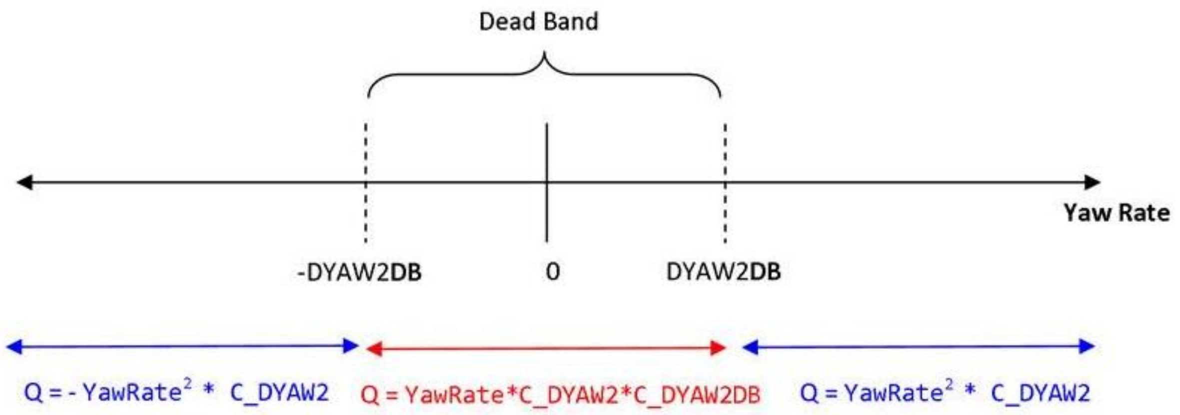
4.5 Yaw quadratic damper

Versions: 4.0 and later

You can defined a quadratic damper using the following code

```
MSTART EXTRA
DYAW2                    *Quadratic yaw damping (Nms^2/rad^2)
DYAW2DB                  *Dead band (rad/s)
DYAW2LIMIT               *Quadratic yaw torque limit (Nm)
YAWMLIMIT                *Total yaw torque limit (Nm)
MEND
```

There is a “dead band” in the yaw rate for the damper, where the applied torque is linear rather than quadratic as shown in the diagram below (a small linear region helps to avoid numerical problems). The diagram shows the damping torque Q that will be applied to the yaw bearing when the yaw rate is inside or outside of the Dead Band. Note also that the damping torque is limited by the yaw torque limit, DYAW2LIMIT.



4.6 Yaw Friction Dependent on Bearing Loads

Versions: 4.0 and later

You can define the yaw friction to be a function of thrust and bending moment on the yaw bearing.

```
MSTART EXTRA
QCOUL                    *constant kinetic friction
NYAWBTHRUST              *number of items in thrust look-up table
YAWBTHRUST               *NYAWBTHRUST values of bearing thrust loading
QCOUL                    *NYAWBTHRUST values of additional friction due to
thrust loading
NYAWBMOMENT              * number of items in bending moment look-up table
YAWBMOMENT               * NYAWBMOMENT values of bearing bending moment
QCOUL                    * NYAWBMOMENT values of friction due to bending moment
MEND
```

The values are entered as look-up tables of additional yawing friction as a function of load.

4.7 Additional teeter restraint models

Versions: 4.4 and later

The Bladed user interface allows definition of simple linear stiffness and damping of the teeter hinge, as well as the ability to have a free teeter angle (below which teeter stiffness and damping = 0) and a spring preload torque. This section describes some additional models for the restraint torque, covering non-linear stiffness and damping terms and teeter hinge friction.

4.7.1 Non-linear teeter stiffness

A look-up table for teeter restraint stiffness can be defined as below.

```
MSTART EXTRA
NKTEET2 4                                  * number of points in look-up table
KTEET2ANGLE   -0.1  -0.01 0.01 0.1         * teeter angle (radians)
KTEET2TORQUE   -10000  0  0  10000         * teeter restraint torque (Nm)
MEND
```

4.7.2 Non-linear teeter damping

There are two non-linear teeter damping options available through Project Info, with either a linear or quadratic dependence on teeter velocity.
The damping follows one of the following formulae:

Linear damper: $$Damping\ torque = K_a * e^{b(\gamma-\gamma_0)} * \dot{\gamma}$$
Quadratic damper: $$Damping\ torque = K_a * e^{b(\gamma-\gamma_0)} * \dot{\gamma} * |\dot{\gamma}|$$

where γ is the teeter angle in delta-3 coordinates. Ka and b are parameters that must be specified.

For the quadratic damper model, it is also possible to specify a teeter rate below which a linear damping model is used. This can help with stability of the quadratic damper at low teeter rates. When the teeter rate is below (the constant value) $\dot{\gamma}_{D2LIN}$, a different equation is used to work out the damping.

Quadratic damper linear region: $$Damping\ torque = K_a * e^{b(\gamma-\gamma_0)} * \dot{\gamma} * \dot{\gamma}_{D2LIN}$$

The damper is active when the teeter angle is greater than γ_0, **and only when the motion is away from zero degrees**. Additionally the damping is capped at a maximum damping torque and when the teeter angle is greater than γ_1 , the damping torque always takes the maximum value.

Example parameters to be entered in Project Info:

```
MSTART EXTRA
DELTA3DAMP     1           * Enables non-linear damper. Value = 1.
TEETERK  120000            * Ka
TEETERGAM0     0.0872      * γ0 (rad)
TEETERGAM1     0.1919      * γ1 (rad)
EXPONENT       100         * b (if b = 0 is wanted, use b = 0.0000001)
TEETERMMAX     400000      * Maximum damping torque (Nm)
LINEARDAMP     0           * 0 for "Linear damper" and 1 for "Quadratic
damper"
D2LINGAMMA     0.1         * Teeter rate (rad/s) below which "quadratic
damper linear region" used
MEND
```

4.7.3 Teeter Friction and Stiction

A stick-slip friction model can be applied to the teeter hinge. Constant friction and stiction torques can be defined. The friction torque is the "kinetic friction" that describes the friction torque experienced while the hinge is in motion. The stiction torque is additional friction torque only present when the hinge is stationary. In order for the teeter hinge to move (assuming no other restraints), the torque acting on the hinge must be greater than the sum of the constant friction and stiction terms.

Enter the following code into Project Info.

```
MSTART EXTRA
TEETERFRICTION 1000        * Teeter spring friction (Nm)
TEETERSTICTION  600        * Teeter spring stiction (Nm)
MEND
```

4.7.4 Combining teeter restraint functions

If it is desired to include more than one of the above teeter functions in a single simulation, the lines must be entered in the order shown below. Any complete set of lines (e.g. all three lines relating to non-linear stiffness) can be omitted if that functionality is not required.

```
MSTART EXTRA
DELTA3DAMP      1
TEETERK         120000
TEETERGAM0      0.001745329
TEETERGAM1      0.013962634
EXPONENT        100
TEETERMMAX      4000
LINEARDAMP      1
D2LINGAMMA      0
NKTEET2         4
KTEET2ANGLE     -0.1  -0.01  0.01  0.1
KTEET2TORQUE    -10000  0  0   10000
TEETERFRICTION 1000
TEETERSTICTION 1000
MEND
```

4.7.5 Teeter offset and hub centres of mass

This section describes the functionality and conventions regarding mass and geometry definitions for the hub of a teetered turbine. The sign conventions for defining the hub centres of mass and teeter offset are shown in Figure 4-1 and Figure 4-2.

4.7.5.1 Hub centres of mass

For teetered turbines, the overall mass and inertia of the hub is considered in two separate parts. Firstly, the total mass of the 'non-teetering parts' (labelled as the 'Hub CoM' on figures 1 and 2) is defined relative to the end of the Low Speed Shaft (LSS). Secondly, the mass of the 'teetered' parts of the hub (labelled 'Teetered Hub CoM' on the figures) is defined relative to the rotor centre. The sum of the 'Hub' mass and the 'Teetered Hub' mass equals the total hub mass.

The hub mass and inertia defined in the User Interface refers to the parts of the hub that are 'non-teetered', in other words that are placed inboard of the teeter hinge, directly on the LSS. Teetered hub mass (i.e. outboard of the teeter hinge) should be defined with the following Project Info code.

```
MSTART EXTRA
THUBMAS   11460    *Teetered hub mass (kg)
THUBCTR    0.15    *Teetered hub centre of mass offset from rotor centre (m)
THUBINE   1000     *Teeter hub inertia about rotor rotational axis (kgm2)
THUBINE2  1000     *Teeter hub inertia about other orthogonal axes (kgm2)
MEND
```

4.7.5.2 Teeter Offset

In practice, teetered designs with coned rotors or blades with pre-bend will usually have a 'Teeter Offset' which is the axial distance from the rotor centre to the teeter hinge. This offset is designed to ensure that the rotor centre of mass is directly above the teeter hinge axis. If this were not the case, then at low rotor speeds the rotor would tend to fall away from the intended vertical position.

In Bladed, this offset is defined in Project Info. According to the conventions described in figures 1 & 2, the Teeter Offset in Bladed will be a negative number if the rotor is coned away from the support structure. This would be the case for both upwind and downwind designs.

```
MSTART EXTRA
TEETEROFFSET        -0.3         *Teeter offset (m)
MEND
```

4.7.5.3 Hub geometry sign conventions

The sign conventions for defining the teeter offset and hub centres of mass are shown in Figure 4-1 and Figure 4-2.

The non-teetered hub centre of mass and Teeter Offset are defined relative to the end of the LSS, whereas the teetered hub centre of mass is defined relative to the rotor centre. A change in Teeter Offset therefore changes the position of the teetered hub centre of mass. All of these parameters are defined as "positive away from the tower" for both upwind and downwind turbines. This sign convention has the advantage that the same geometrical parameters should be suitable for both an upwind and a downwind turbine.

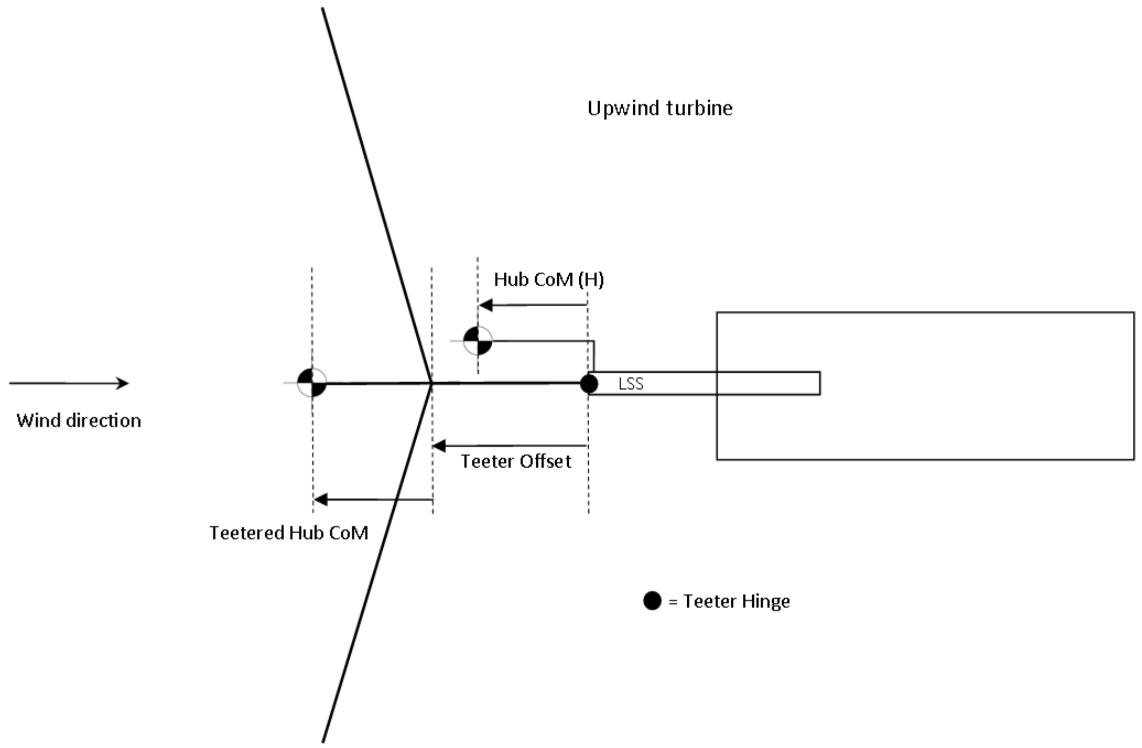


Figure 4-1: Hub geometry sign conventions for an upwind turbine

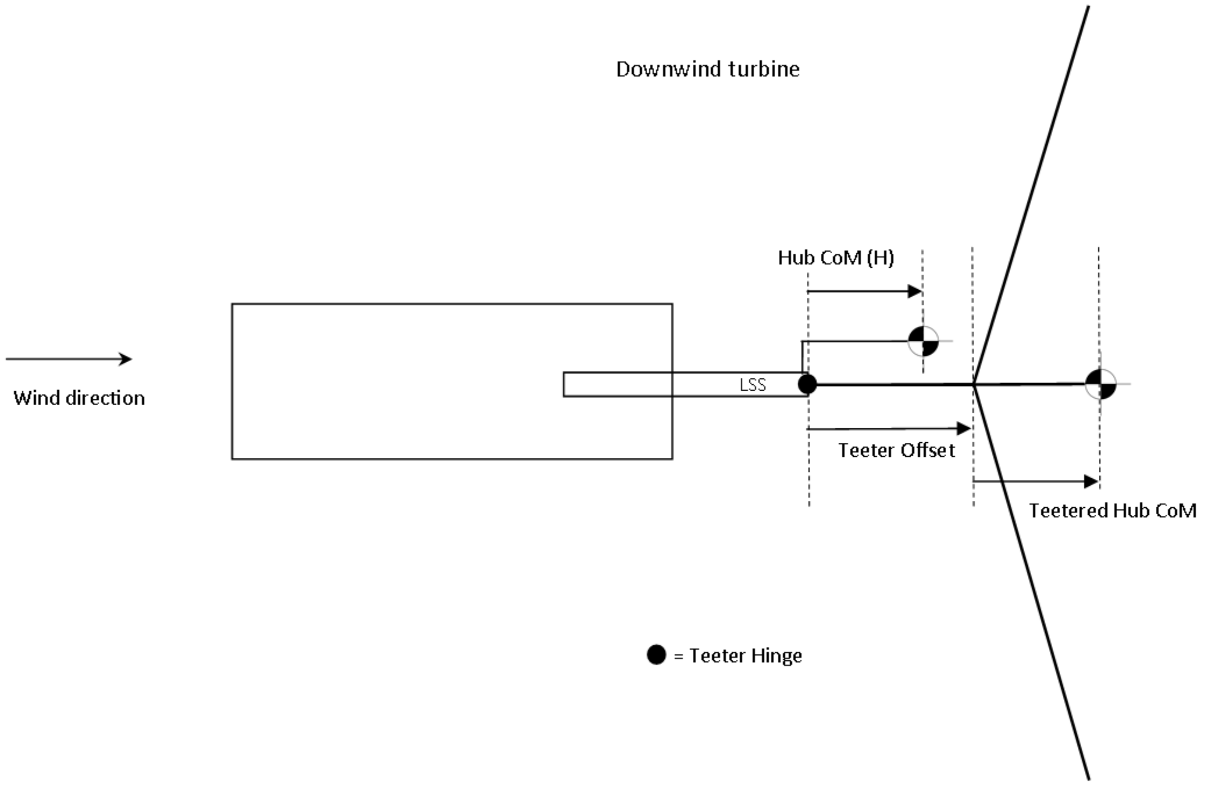


Figure 4-2: Hub geometry sign conventions for a downwind turbine

4.8 Pendulum tower damper functionality

Versions: 4.6, 4.8 and later

A pendulum tower damper can be included on any support structure node. For full details, see the document “Pendulum Damper User Manual”, available on the Bladed User Portal.

Example project info text below:

```
MSTART PENDULUMDAMPERS
NumPendulumDampers 1
Station 08
Length 3.6
Direction 1.57
Mass 6120
Inertia 13225
NumStiffnessPoints 23
StiffnessAngles -0.4471 -0.4454 -0.4438 -0.4432 -0.4419 -0.4409 -0.4391 -
0.4377 -0.4371 -0.4365 -0.4363 0 0.4363 0.4365 0.4371 0.4377 0.4391 0.4409
0.4419 0.4432 0.4438 0.4454 0.4471
StiffnessTorques -1290000 -1290000 -1250000 -1230000 -1120000 -1010000 -
796000 -570000 -415000 -172000 0 0 0 172000 415000 570000 796000 1010000
1120000 1230000 1250000 1290000 1290000
Damping 27100
InitialAngle 0.0
MEND
```

4.9 Ignore damper in mode shapes

Versions: 4.6, 4.8 and later

When tower dampers are added at nodes other than the tower top, extra support structure attachment mode shapes will be generated based on the points at which the damper is attached. This can be undesirable as the simulation time can be increased as the modal basis is less efficient as the modes are less physically realistic.

Assuming that the damper does not have a significant effect on the tower mode shapes, it is therefore desirable to ignore tower dampers for the purposes of calculating mode shapes. This can be achieved with the following Project Info code

```
MSTART EXTRA
IgnoreTowerDamperInModalAnalysis Y
MEND
```

5 ELECTRICAL SYSTEMS

5.1 Generator Cogging Torque

Versions: 4.0 and later

Cogging torque for direct drive generator

```
MSTART EXTRA
COGGINGTORQUE 1
COGTORQUEAMP          * Amplitude of cogging torque (Nm)
COGTORQUEFAC          * Frequency of cogging torque = COGTORQUEFAC *
generator speed (rad/s)
MEND
```

5.2 Speed Dependent Electrical Losses

Versions: 4.0 and later

Electrical losses can be made dependent on rotor speed.

```
MSTART LOSS
LMODEL 3                 *Loss model, 3=2D look-up table against gen speed and
air-gap torque
SPDVAR 0                 * 0 = based on speed; 1 = based on slip
NESPD 2                  *Number of speed values for look-up table
ESPDS 0 161              *Speed values for look-up table separated by spaces
NETRQ 2                  *Number of torque values for look-up table
ETRQS 0 1280000          *Torque values for look-up table separated by spaces
ELOSSPOW 0 0             *2D look-up table: Torque across, speed down
100000 100000
MEND
```

6 CONTROL SYSTEMS

6.1 Strain Gauges

The external controller is already provided with a set of loading data, but it is possible to add additional strain gauges at any station on the tower or blade. Although called "strain gauges", these actually report forces and moments. Once specified, these will be updated before each call to the discrete external controller.

6.1.1 Strain Gauges on Tower

Versions: Bladed 4.5 and 4.6

Strain gauges on the tower are specified differently for monopole and multi-member towers. Only enter the entries applicable to the tower type.

```
MSTART EXTRA
NUMTOWERSTRAINGAUGES         * The number of tower strain gauge definitions
* Repeat this block for every strain gauge definition
SGHEIGHT                     * (Monopile only) Z-coordinate of strain gauge
SGMEMBER                     * (Multi-member only) member number
SGFRACTION                   * (Multi-member only) Fraction along the member
from end 1 to end 2
* end of block
MEND
```

This means that the strain gauge location for a monopile tower could be specified in two different ways, whereas a multi-member tower only one. This is illustrated in

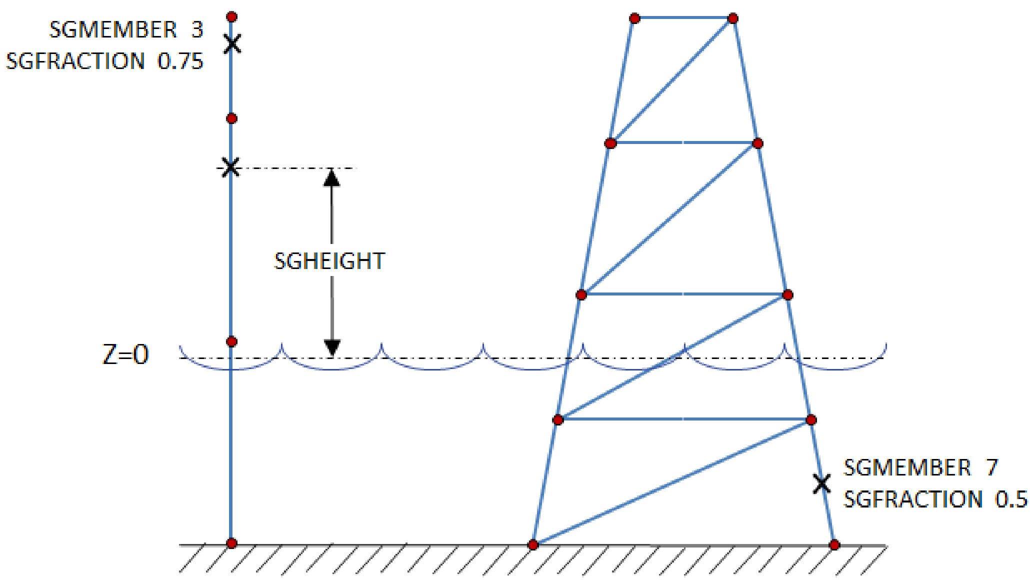


Figure 6-1: Tower strain gauge position specification

It is strongly recommended not to mix specification methods for a monopile tower, as this can be ambiguous. All tower strain gauges report three forces and three moments: Fx, Fy, Fz, Mx, My, Mz. For monopile towers, these are in the global coordinate system (i.e. Z is vertical), otherwise it will be in the member's coordinate system (i.e. X is axial along the member).

Versions: Bladed 4.7 and later

Strain gauge definitions must be entered using the member number and fraction along the member. Note that the fraction must be either 0 or 1 so that the load sensor is at an end of the member.

```
MSTART EXTRA
NUMTOWERSGAUGES      * The number of tower strain gauge definitions
* Repeat this block for every strain gauge definition
SGMEMBER                   * member number
SGFRACTION                 * Must be 0 or 1
* end of block
MEND
```

All tower strain gauges report three forces and three moments: Fx, Fy, Fz, Mx, My, Mz in the member's coordinate system (i.e. X is axial along the member).

6.1.2 Strain Gauges on Blades

Versions: Bladed 4.5 and later. In 4.5 and 4.6, the angle is used and the controller interface returns only a single angle in that direction. In 4.7 and later, the angle is not defined and all six load components are returned.

Strain gauges on the blades are defined with a distance and an angle.

```
MSTART EXTRA
NUMSGAUGES           * number of blade strain gauge definitions to follow
SGDIST               * Distance along the blade
SGANG                * (4.5 & 4.6 only) The angle from the chord line
MEND
```

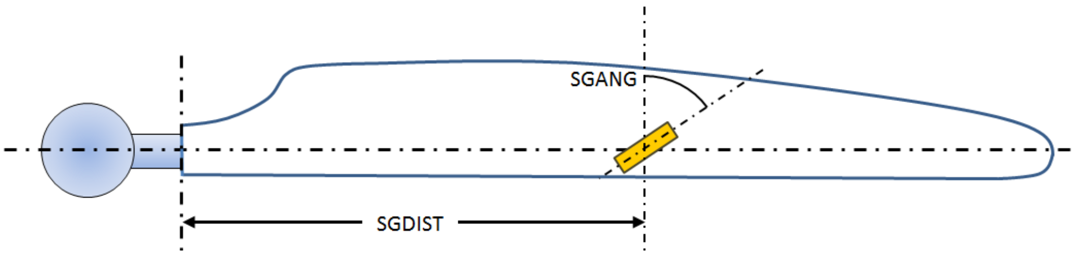


Figure 6-2: Blade strain gauge position specification

In Bladed 4.7 and later, the resulting load can be retrieved in the external controller using the function *GetMeasuredBladeStrainGaugeFx*, changing the last two letters for the relevant load component. In Bladed 4.5 and 4.6, the load is accessed through *GetMeasuredBladeStrainGaugeF.*

6.2 Accelerometers

As for strain gauges, there are already a number of locations where the accelerations are measured on each discrete controller time-step. Additional locations can be added to the tower and blades in a similar manner as for strain gauges.

6.2.1 Accelerometers on Tower

Versions: 4.5 to 4.10

As for strain gauges, accelerometers on the tower are specified differently for monopole and multi-member towers. Only enter the entries applicable to the tower type.

```
MSTART EXTRA
NUMTOWERACCELEROMETERS      * The number of accelerometer definitions
* Repeat block for each accelerometer
ACCHEIGHT                   * (Monopile only) Z-coordinate of accelerometer
ACCMEMBER                   * (Multi-member only) Member number
ACCFRACTION                 * (Multi-member only) Fractional distance from
end1 to end2
* end of block
MEND
```

For monopile towers, the accelerations in X,Y and Z are provided in the global coordinate system (i.e. Z is vertical), otherwise it will be in the member's coordinate system (i.e. X is axial along the member).

Versions: 4.11 and later

Accelerometers can be simulated in Bladed and the 6DOF acceleration outputs are accessed either via a result output group or in the external controller using the appropriate interface functions. The accelerometers can be attached to nodes on the tower. The acceleration signals are the node accelerations in global coordinates. The user defines the node by selecting the member and then the end of the member.
This is supported for both multi-member and axis-symmetrical towers.

```
MSTART EXTRA
NUMTOWERACCELEROMETERS      * The number of accelerometer definitions
* Repeat block for each accelerometer
ACCMEMBER                   * Member number
ACCFRACTION                 * end1/end2 (0/1 respectively) of the member
MEND
```

The accelerations (X, Y and Z) are given in the global coordinate system.
Note that providing a tower height is no longer supported. Accelerometers can only be placed at member ends (e.g., ACCFRACTION must be either 0 or 1).
In axis-symmetrical towers each tower station node is connected with a member with increasing ID from tower base to the top.

6.2.2 Accelerometers on the Nacelle

Versions: 4.5 and later

Accelerometers on the nacelle are defined with an offset from the yaw bearing.

```
MSTART EXTRA
NUMNACELLEACCELEROMETERS          * The number of nacelle accelerometer
definitions that will follow.
ACCOFFSET                         * 3 values for the offset {X, Y, Z} in
global coordinates from the tower top. Repeated for each accelerometer
MEND
```

The resulting accelerations are provided for X, Y, Z in the local nacelle's coordinates.

6.2.3 Accelerometers on Blades

Versions: 4.5 and later

Accelerometers on the blades are defined in a similar manner to strain gauges, but only with a distance and not an angle.

```
MSTART EXTRA
NUMBLADEACCELEROMETERS      * The number of blade accelerometer definitions
that will follow.
ACCDIST                     * Distance along the blade from blade root. This
line is repeated for each accelerometer definition
MEND
```

The resulting accelerations are provided for X, Y, Z in the local blade's coordinates.

6.3 Delaying the start-up sequence

Versions: Bladed 4.3 and later

You can specify a delay before the start-up sequence occurs, as follows:

```
MSTART EXTRA
STARTTIME   75.0            * Delay, in sec, from start of simulation to
beginning turbine start-up
MEND
```

During the delay period the turbine will be in Idling mode.

6.4 Use Legacy DISCON

Versions: Bladed 4.5. In the UI in 4.6 and later as "Force Legacy DISCON"

For external controller dlls which contain both current and legacy controllers (CONTROLLER and DISCON functions), by default, the current controller will be used in preference. By setting the following flag, the DISCON legacy controller will be used. If this flag is set and DISCON is not found, then an error will be raised.

```
MSTART EXTRA
USE_DISCON 1
MEND
```

6.5 Tower inclinometers

Versions: Bladed 4.7 and later. Syntax is different for 4.9 (and later) compared to 4.7 and 4.8.

In Bladed 4.9, inclinometers can added to the tower using the following text:

```
MSTART EXTRA
NUMTOWERINCLINOMETERS  2     *number of inclinometer signals to apply
INCLNODE  3, 4               *nodes onto which the inclinometers are applied
MEND
```

In Bladed 4.7 and 4.8, each INCLNODE must be specified on a new line:

```
MSTART EXTRA
```

```
NUMTOWERINCLINOMETERS  2   *number of inclinometer signals to apply
INCLNODE   3               *apply inclinometer at node 3
INCLNODE   4               *apply inclinometer at node 4
MEND
```

The inclinometer reading is then available to the controller through the controller API. An example function is shown below.

```
/// <summary>
/// Returns the rotation of the sensor from the horizontal, about the local Y-axis.
/// Measured in rad.
/// </summary>
/// <param name="turbine_id">  The id of the turbine model that is provided by the simulation.  This sl
/// <param name="index_turbine_inclinometer">  Index of turbine inclinometer, starting at 0.</param>
double GetMeasuredTurbineInclinometerAngleY(const turbine turbine_id, int index_turbine_inclinometer);
```

6.6 Blade aerodynamic information

It is possible to access the angle of attack and relative wind speed measured signals at a blade station via the external controller. This can be achieved using the *GetMeasuredBladeStationWindSpeed* and *GetMeasuredBladeStationAngleOfAttack* interface functions. This returns signals that match the aerodynamic information output signals angle of attack and relative wind speed. The user will not be able to obtain these results if the model has encrypted and the aerodynamic information output has been restricted by the user.

To enable this feature the user needs to add the following code to project info.

```
MSTART EXTRA
PassBladeValsToController 1   *0 disables the measured signal (default), *1
enables the signal output
MEND
```

NOTE: This feature is not recommended for calculation of external applied loading as the external controller returns a time lagged value (dependent on controller call step size). This will produce a zero order value for the during of a time step (during time step integration) assuming the controller step size is larger than the integrator step size. Thus the value is not be updated as part of sub-stepping or iteration during a time step and so will not be representative of the instantaneous blade motions/wind velocity. The most suitable approach for applying loads to the Bladed structural model is using the applied loads call made available in the external loads dll where the user can also access the blade structural motion and the wind velocity.

7 MULTIPLE ROTOR FUNCTIONALITY

7.1 Initial azimuth angle (power production)

Versions: Bladed 4.4 and later. Syntax is different in 4.6 and later compared to 4.4 and 4.5

In Bladed 4.4 / 4.5 you can specify the initial azimuth angle on each rotor. For example, with two rotors

```
MSTART EXTRA
MULTIINITAZI 0.1, 1.5       * Initial azimuth angles (radians) for each rotor
MEND
```

In Bladed 4.6, it is entered as in the INITCON module.

```
MSTART INITCON
NUMROTORS 2           * number of rotors
* Repeat for each rotor
INAZI 0.5             * Initial azimuth
INIMD 0               * Initial machine yaw direction
* end of block
MEND
```

7.2 Rotor rotational directions

7.2.1 Inputs for Project Info

Versions: 4.4 and later

You can specify the rotational direction for each rotor. For example, with two rotors

```
MSTART EXTRA
MULTIROTORDIRS C,A          * Direction of each rotor (C = clockwise, A =
anti-clockwise)
MEND
```

Note that the first rotor direction in MULTIROTORDIRS must be the same as the rotor direction specified in the UI to ensure correct elastic axis information is passed to the UI.

Note for versions 4.6 and earlier (resolved in 4.7)

There is an issue with the blade definition for counter-rotating rotors in Bladed 4.6 and earlier. Inside the Bladed code, when changing the arrangement from anti-clockwise to clockwise, the direction of the “Neutral Axis (y)” offset in the blade definition is reversed to give the correct blade geometry. However, it is not yet implemented within Bladed to account for this difference due to rotor direction when there are counter-rotating rotors. Therefore, it is recommended to select “Neutral Axis (y)” to be equal to zero.

7.2.2 Sign convention for counter-rotating rotors

- **Rotor speed** should generally be positive during operation. The rotor speed is defined as positive in the direction of expected rotation when operating. **Rotor torque** should generally be positive during operation for the same reason.

- For upstream rotors, the x direction is downstream, so hub Mx is clockwise when viewed from upstream (for both clockwise and anticlockwise rotors). **Rotating hub Mx** should take an opposite sign for the clockwise and anticlockwise rotors.

8 OUTPUTS

8.1 Blade acceleration output

Versions: 4.2 and later

Blade accelerations can be output using the following code:

```
MSTART EXTRA
OutputBladeAcc  1
MEND
```

Accelerations will be output at the same stations as the blade deflections. Accelerations are output in the blade principal axes coordinates.

8.2 Blade refined deflection output

Versions: 4.6 and later

Usually the blade deflections output are modal deflections. In Bladed 4.6 and later, the blade deflections calculated from the FE model can also be output if desired.

```
MSTART EXTRA
OutputRefinedBladeDeflections 1
MEND
```

Note that this function is not allowed for multi-part blades.

8.3 State values output

Versions: 4.3 to 4.6. In 4.7, there is an output group called “structural state positions” output by default for all unencrypted models.

State values (i.e. degree of freedom position and velocity values) can be output using the following code:

```
MSTART EXTRA
OUTPUT_STATES 1
MEND
```

8.4 Tower buoyancy

Versions: 4.7 and later

The buoyancy force and centre of buoyancy can be output for the support structure or tower. This is especially useful for floating wind turbines. The centre of mass of the undeflected turbine is output into the $VE file for comparison.

```
MSTART EXTRA
OUTPUT_BUOYANCY1      *0 : No buoyancy output.
                      *1: Outputs the centre of buoyancy and the centre of
gravity for the support structure.
                      *2: Outputs the centre of buoyancy and the centre of
gravity for each support structure member.
MEND
```

This information is output for time domain simulations only. There are two extra output groups called “Turbine Centre of Gravity” and another called “Buoyancy of support structure”.

8.5 Avoiding interpolation in post-processing

Versions: 4.9 and later

When post-processing a signal with a secondary axis value, such as distance along a blade or tower, small numerical discrepancies can result in an out of range error if the distance is slightly outside the range of the secondary axis values. To avoid this, a small tolerance is applied at the end stations.

Numerical discrepancy at non-end stations can result in signals being interpolated instead of the exact values being used. To avoid this, the same tolerance can be applied by using the following:

```
MSTART EXTRA
NEARESTNEIGHBOUR Y        * Y = tolerance applied, N = not applied (default)
MEND
```

9 ENVIRONMENT

9.1 Evolving turbulence (for Lidar)

Versions: 4.5 and later

The evolving turbulence model can be changed via Project Info until the Wind screen has been redesigned and released.

```
MSTART EXTRA
EVOLVING_TURBULENCE   2      * 0 = None, 1 = Kristensen, 2 = Exponential
EXPONENTIAL_FACTOR 0.3       * The value to be used with Exponential model
MEND
```

9.2 Gust Propagation

Versions: 4.2 and later

By default, Bladed gusts are spatially invariant, which means that the gust happens at the same time at all locations (including upstream). This can be changed to make the gust propagate longitudinally. The gust moves in the longitudinal direction at the speed of the start value of the gust.

```
MSTART EXTRA
GUSTPROPAGATION  1
MEND
```

9.3 Wind Dll

Versions: 4.7 and later

The user is able to completely substitute the wind field created by Bladed with their own dynamic link library (dll). The dll is asked to return a three dimensional wind velocity vector in global coordinates at any requested time and location in three dimensional global space. The dll is called every time the wind velocity is requested, which will be at least for every blade station on every time-step.

The dll must implement a single function in a cdecl calling convention (default in C-based languages), with the following name and signature:

```
int GetWindVelocity(const double time, const double Location[3], double WindVelocity[3])
```

The arguments are the simulation time, global position [x,y,z] at which wind velocity is requested and the wind velocity [u,v,w]. The latter must be set by the dll. The return value is the error code and should be zero under normal conditions whereas a non-zero value will indicate an error and cause the simulation to end.

```
MSTART EXTRA
WindDllPath  'C:\...'        * Path to user-compiled dll
MEND
```

9.4 Meandering Wake

Versions: 4.0 and later

When an upwind turbine wake model (either Gaussian or Eddy viscosity) is chosen in Bladed then the wake centreline is fixed throughout the simulation. The meandering wake model allows the “meandering” of the wake centre-line with simulation time. In order to do this, the user must firstly generate a bespoke low-pass filtered wind file, separately to the normal wind file used in turbulent wind simulations.

9.4.1 Generating low-pass filtered wind file

The following option can be used during the wind turbulence generation calculation to apply a low-pass filter to the turbulent velocities saved in a wind file. The filter is applied to the velocity signal in the longitudinal direction for each velocity component at each (y,z) grid point.

```
MSTART EXTRA
CUTOFF 1              * Activate low-pass filter (1 = sharp cut-off)
CutoffFreq 0.04       * sharp cut-off frequency for turbulence spectrum in Hz
MEND
```

Note that a low-pass filter cannot be applied to:

1. A wind file generated using the Mann turbulence model.
2. A wind file generated to match a wind time history at specific (y,z) grid node in the file.

9.4.2 Time domain simulation

For the time domain simulation, the user must specify a wind file that determines the movement of the meandering wake. The turbulence and wind speed of this can be overwritten from the default. Furthermore, the user may also add small length scale turbulence to the wake deficit with the intention to represent eddies created by tip vortices.

```
MSTART WINDP
WPMODEL 3
WAKEP_NPTS 40         * Number of points defining the wake deficit radius
WAKEP_RAD 0 6.3...    * Wake deficit radii in metres
WAKEP_DEF 25 31...    * Wake deficit depths, percent of free-stream wind
speed
CENTRE 0              * horizontal offset between the meandering wake
centreline & the turbine hub
VCENTRE 0             * vertical offset between the meandering wake
centreline & the turbine hub
MEND

MSTART EXTRA
MEANDERINGWAKE   5   * 5 = recommended setting
MWWINDF         * location of filtered wind file generated by 9.4.1
MW_TI_HERE  1   * [optional] flag to indicate that turbulence and mean
velocity of wind file is specified below, otherwise it uses information from
the wind file
MW_TI_V         * [MW_TI_HERE>0] Lateral turbulence intensity
MW_TI_W         * [MW_TI_HERE>0] Vertical turbulence intensity
MW_UBAR         * [MW_TI_HERE>0] Mean wind velocity of wind field
MWAKEDIST       * Perpendicular distance to upstream turbine from which wake
is generated
* [optional] following options add additional turbulence to the wind file
TURBWAKE 1      * adds turbulence within the wake
TWWINDF         *Path to small scale turbulence wind file
TW_TI_U 0       * Longitudinal turbulence intensity as decimals
TW_TI_V 0       * Lateral turbulence intensity as decimals
TW_TI_W 0       * Vertical turbulence intensity as decimals
DEFGAIN 0       * Constant associated with deficit part of the added
turbulence scaling equation
DEFDERGAIN 0    * Constant associated with the deficit derivative part of the
scaling equation
MEND
```

10 SIMULATION CONTROL

10.1 Solve_system iterations and tolerance

Versions: 4.5.0.86 and later revisions. All revisions of 4.6 and later

There is an iteration in the Multibody solver to find the structural state accelerations.
The maximum number of iterations can be changed from the default value of 4. If all of the state accelerations have changed by less than *SolveSystemTolerance* since the last iteration, then the number of iterations can be reduced to less *NumberSolveSystemIterations*. The tolerance can also be set as shown.

```
MSTART EXTRA
NumberSolveSystemIterations 5      * maximum number of iterations (default = 4)
SolveSystemTolerance 0.003         * relative tolerance in finding
accelerations (default = 0.05)
MEND
```

11 AERODYNAMICS CONTROL

11.1 Lift slope model in old aerodynamics

Versions: 4.3.0.91 up to 4.7

By default the aerodynamics model in Bladed calculates the slope of the lift curve at the point where the lift curve crosses the x-axis. The first data points above and below the x-axis then determine the slope of the lift curve. In the new aerodynamics module (from version 4.7) the slope of the curve is determined with the crossing points of the x-axis and the y-axis. This latter method is considered more accurate and is therefore also added in as an option in the old aerodynamics.

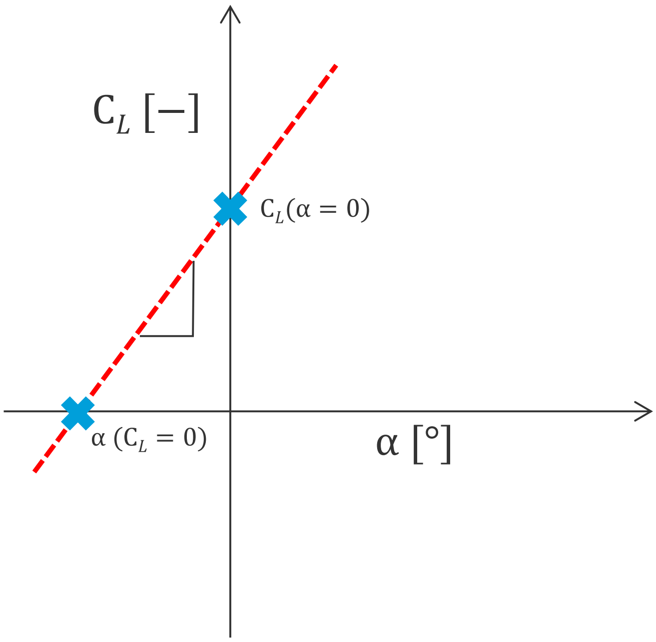


Figure 11-1: Illustration of linear part of the lift curve with crossings at the x, - and y-axis.

To use the triangle method then the following code can be added into the ProjectInfo section:

```
MSTART EXTRA
LIFTSLOPEMODEL 2
MEND
```

Using the value of “1” will select the old method of the lift slope; this is also the default value in case nothing is entered in the ProjectInfo section.