

# CACTUS User Guide

Jonathan C. Murray  
Aerosciences Department

Matthew Barone  
Wind Energy Technologies

Phillip K. Chiu  
Wind Energy Technologies

Sandia National Laboratories  
P.O. Box 5800  
Albuquerque, NM 87185

# Contents

<b>Contents</b>	<b>1</b>
<b>List of Figures</b>	<b>2</b>
<b>List of Tables</b>	<b>2</b>
<b>1 Introduction and overview</b>	<b>4</b>
1.1 Blade loads and wake models . . . . .	4
1.1.1 Dynamic blade loads . . . . .	5
1.2 Solid and free-surface boundaries . . . . .	6
<b>2 Normalization parameters</b>	<b>7</b>
<b>3 Input files</b>	<b>8</b>
3.1 CACTUS namelist input . . . . .	8
3.1.1 Input configuration . . . . .	8
3.1.2 Case inputs . . . . .	11
3.1.3 Output configuration . . . . .	13
3.1.4 Example namelist input file . . . . .	14
3.2 Geometry input file . . . . .	15
3.2.1 Header parameters . . . . .	16
3.2.2 Blade parameters . . . . .	17
3.2.3 Strut parameters . . . . .	18
3.2.4 Example geometry input file . . . . .	20
3.2.5 Geometry creation tools . . . . .	23
3.3 Airfoil input file . . . . .	23
3.4 Wall input file . . . . .	25
<b>4 Output description</b>	<b>27</b>
4.1 Revolution-averaged rotor-integrated loads . . . . .	27
4.2 Blade-integrated loads . . . . .	28
4.3 Blade element loads . . . . .	28
4.4 Ground plane, wall, and free surface output . . . . .	30
4.4.1 Ground plane data . . . . .	30
4.4.2 Free surface data . . . . .	30

4.5	Field velocities . . . . .	31
4.6	Vortex filament data . . . . .	32
<b>5</b>	<b>Details</b>	<b>33</b>
5.1	OpenMP acceleration . . . . .	33
	<b>Bibliography</b>	<b>34</b>
	<b>Appendix A Example geometry script</b>	<b>35</b>

## List of Figures

1.1	Blade element with associated vortex lattice system. . . . .	5
-----	--	---

## List of Tables

2.1	Parameters used for non-dimensionalization. . . . .	7
3.1	Available input configuration options in the <code>&amp;ConfigInputs</code> namelist. . . . .	8
3.2	Available input configuration options in the <code>&amp;CaseInputs</code> namelist. . . . .	11
3.3	Available input configuration options in the <code>&amp;ConfigOutputs</code> namelist. . . . .	13
3.4	Parameters in the header of the geometry input file. . . . .	16
3.5	Parameters in the blade section of the geometry input file. . . . .	17
3.6	Parameters in the strut section of the geometry input file. . . . .	19
3.7	Header block for airfoil input files . . . . .	24
3.8	Reynolds number data block for airfoil input files . . . . .	24
4.1	Revolution-averaged output data. . . . .	27
4.2	Blade integrated loads. . . . .	28
4.3	Temporal blade element loads. . . . .	29
4.4	Wall outputs. . . . .	30
4.5	Free surface outputs. . . . .	31

4.6	Field velocity output data. . . . .	31
4.7	Vortex filament output data. . . . .	32

# 1 Introduction and overview

CACTUS (**C**ode for **A**xial and **C**rossflow **T**urbine **S**imulation) is a turbine performance simulation code using a blade element discretization and a free vortex line description of the turbine wake. The code was originally based on **VDART3**, a free vortex wake simulation of the Darrieus wind turbine, developed by Strickland [3]. The codebase has been largely upgraded to the Fortran 9x standard, and a number of modifications have been made to the original **VDART3** methods including updates to the blade loads models, and new models to handle generic device geometry and marine turbine specific physics.

This document serves as a user’s manual for CACTUS. A brief overview of the code is provided below, however, more detailed information on the methods used can be found in [2].

## 1.1 Blade loads and wake models

CACTUS simulates a turbine device consisting of an arbitrary configuration of blade element sections. Each section can be assigned arbitrary load coefficient vs. angle of attack characteristics, which typically correspond to two-dimensional lift and drag coefficient data for a particular foil section. Since data from two-dimensional wind tunnel tests or foil performance calculations are used to represent element loads, it is generally assumed that these elements are in locally two-dimensional flow.

A rotor blade consisting of an arbitrary planform shape and foil sections can be modeled by the synthesis of a number of blade elements. The blade loads and wake of the turbine rotor are evolved in time over a certain number of rotor revolutions, until the revolution-averaged rotor power is converged. The code output includes the blade aerodynamic forces, wake vortex trajectories, and performance metrics such as torque and power.

CACTUS uses a potential flow model comprised of free vortex line elements to represent the turbine wake flow field. The vortex line structure attached to a single blade element is shown in Section 1.1. At each point in time, the bound vorticity ( $\Gamma_B$ ) on each blade element is related to the element lift coefficient through the Kutta-Joukowski theorem, and the spanwise ( $\Gamma_S$ ) and trailing vorticity ( $\Gamma_T$ ) are recovered through the application of the Helmholtz theorem of conservation of circulation along a vortex line [1].

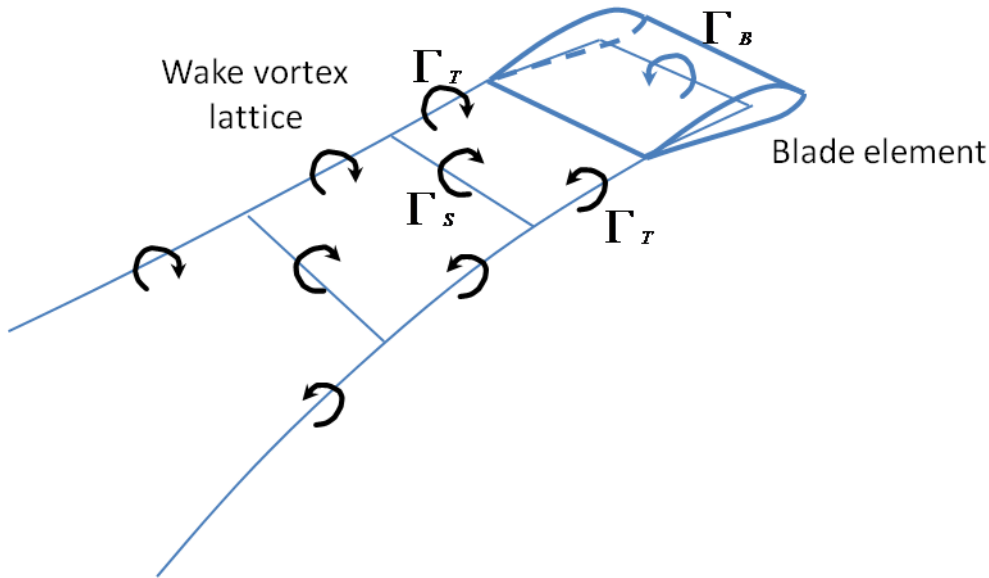


Figure 1.1: Blade element with associated vortex lattice system.

### 1.1.1 Dynamic blade loads

The operational cycle of some turbines, most notably cross-flow turbines or axial flow turbines in yawed flow conditions, cause the turbine blades to operate in dynamically variable flow conditions. The effects of blade rotation with respect to the surrounding fluid and the effects of dynamically variable flow angle of attack are captured with additional models.

The effects of blade section pitch rate (rotation around an axis normal to the section plane) are captured by analogy to an analytical solution for a pitching flat plate. Improvements have been made to the original methodology used in VDART3, and modifications have been made to handle non-zero section pitching moment due to cambered foil sections.

Under certain operational conditions, the turbine blades may operate at angles of attack beyond their steady-state stall limits for significant lengths of time. The transient behavior of the blade section loads during this “dynamic stalling” process must be modeled as it is not captured by the steady load coefficient data input for each foil section. The primary effect of dynamic stalling is a delay in the appearance of stalled flow effects on blade loads to higher angles of attack than would be expected in steady flow.

Two models for dynamic stall effects on blade section loads are included in CACTUS. The modified Boeing-Vertol method of Gormont is the default. This algebraic method approximates dynamic stall effects with a “lagged” angle of attack, where the magnitude of the lag is empirically correlated to the angle of attack rate. The Leishman-Beddoes model incorporates more physical models and attempts to model the temporal evolution of dynamic stall flow phenomena and associated effects on blade loads. This model may provide more accurate results than the algebraic Boeing-Vertol method, but requires many

more simulation time steps to be taken per turbine revolution to achieve converged results.

## 1.2 Solid and free-surface boundaries

CACTUS can simulate the effects of proximity to a ground plane or free (water) surface on turbine performance. The boundary conditions, either zero normal flow for a ground plane or constant surface pressure for a free surface, are applied using rectangular source panel elements. The free surface boundary condition is currently implemented as a quasi-static boundary, allowing it to respond only to the average flow created by the turbine and wake over a full revolution.

The user is allowed to specify the time step interval between updates to the wall panel system. For the free surface model, the wall update interval specifies the number of time steps between updates to the revolution averaged quantities. It is often not necessary to update these quantities on every time step. If it's possible to reduce the frequency of wall panel system updates and still obtain convergence of the simulation output of interest, this can reduce simulation run time considerably.

A recent addition to CACTUS is the ability to model *general* solid boundaries, or walls, using generalized quadrilateral source panels. A 3-D mesh file describing the wall geometry must be generated externally and passed as an input to CACTUS.

## 2 Normalization parameters

Some definitions of parameters used in normalize CACTUS input and output parameters are given in Table 2.1.

Table 2.1: Parameters used for non-dimensionalization.

Variable	Description
$\rho$	Fluid density
$U_\infty$	Freestream fluid flow speed
$A_T$	Turbine reference area. Typically this reference area is chosen to be the projected frontal area of the volume swept by the rotor.
$R$	Turbine reference radius
$\omega$	Turbine rotation rate
$U_{\text{local}}$	Fluid flow speed local to an element.
$U_{\text{tip}}$	Turbine tip speed. Defined as $U_{\text{tip}} = \omega R$ .
$c$	Element chord length
$A_E$	Element planform area

In general, most parameters have been normalized at the machine scale. Unless otherwise noted in the input and output descriptions below, the force, torque, and power coefficients are normalized as:

$$C_F = \frac{F}{\frac{1}{2}\rho U_\infty^2 A_T}$$

$$C_T = \frac{T}{\frac{1}{2}\rho U_\infty^2 A_T R}$$

$$C_P = \frac{P}{\frac{1}{2}\rho U_\infty^3 A_T}$$



## 3 Input files

This section describes the namelist, geometry, and airfoil data input files required by CACTUS. CACTUS is run from the command line with the path to the namelist input file passed as the only argument. The geometry input file and the airfoil data table files are referenced in the namelist input file.

### 3.1 CACTUS namelist input

This section describes the FORTRAN namelist input file for CACTUS. There are two namelist groups in the input file, `&ConfigInputs` and `&CaseInputs`. The parameters that can be input in each group are given in the tables below. Parameters not specified in the namelist input are left at the indicated default values. The parameters listed in bold font below are the parameters most commonly specified in an input file (with the rest being left at default values).

#### 3.1.1 Input configuration

Table 3.1: Available input configuration options in the `&ConfigInputs` namelist.

<i>Regression testing</i>	
Variable name	Description
<b>RegTFlag</b>	Set to 1 to perform a regression test (two iterations, generates <code>_RegData.out</code> output file), 0 for normal operation (default).
<i>Wall calculation</i>	
Variable name	Description
<b>GPFlag</b>	Set to 1 to use a ground plane, 0 otherwise (default 0).
<b>FSFlag</b>	Set to 1 to use a free surface, 0 otherwise (default 0).
<b>GPGridSF</b>	Factor on default ground plane grid spacing (default 1).

<b>FSGridSF</b>	Factor on default free surface near-field grid spacing (default 1).
<b>GPGridExtent</b>	Distance the ground plane will extend from the turbine location. Measured in rotor radii. (default 10.0)
<b>WPFlag</b>	Set to 1 to use a wall geometry read in from file, 0 otherwise (default 0).

---

*Calculation inputs*

---

Variable name	Description
<b>nr</b>	Number of revolutions to perform (default 10)
<b>nti</b>	Number of time steps per revolution (default 20)
<b>convrg</b>	Convergence level for the revolution average power coefficient. Iteration will finish before nr revs if this level is hit. Input -1 to skip convergence check (default).
<b>iut</b>	Number of iterations between wake convection velocity updates. If set to zero, the interval will be calculated automatically. If negative, wake convection velocities will be left at the values calculated at the time the wake element is created (no wake convection velocity updates).
<b>iWall</b>	Number of iterations between wall model updates (if wall calculation is active).
<b>TSFilFlag</b>	Flag to enable timestep filtering. Set to 1 to enable filtering of the blade bound vorticity smooth over ntsf timesteps (often needed for stability when blade chord to radius ratio is high). Set to 0 for no filtering (default).
<b>ntsf</b>	Number of timesteps over which the blade bound vorticity is filtered smooth when <b>TSFilFlag</b> = 1 (default 3).
<b>ivtxcor</b>	Flag to specify the finite vortex core model to use. Input 1 for constant velocity in the core (default). Input 2 to use linear radial velocity distribution in core. Input 0 to turn off core model.
<b>vcrgb</b>	Factor on nominal bound vortex core radius used if <b>ivtxcor</b> =1 (default 1). Nominal bound vortex core radius is specified by the maximum blade chord value input in the turbine geometry specification.

---

<code>vcrft</code>	Factor on nominal trailing wake vortex core radius used if <code>ivtxcor=1</code> (default 1). Nominal trailing wake vortex core radius is specified by the maximum blade element span value input in the turbine geometry specification.
<code>vcrfs</code>	Factor on nominal spanwise wake vortex core radius used if <code>ivtxcor=1</code> (default 1). Nominal spanwise wake vortex core radius is calculated from a reference distance between spanwise wake lines given the temporal discretization level used.
<code>vcutoffrad</code>	Cutoff radius used for vortex induced velocity calculation (default $1.0 \times 10^{-7}$ ).
<code>Incompr</code>	1 to ignore any compressibility effects in models, 0 to include compressibility effects (default).
<code>ifc</code>	1 to use final convergence step, 0 to not (default). If selected, the temporal discretization level is refined once/if initial convergence is reached before <code>nr</code> revolutions have been performed.
<code>nrhc</code>	Revolution number after which to switch to final convergence, if initial convergence level has not yet been achieved. Input -1 to skip this check (default).
<code>ntif</code>	Final number of time steps per revolution. This value will replace <code>nti</code> during final convergence. Input -1 to leave <code>ntif=nti</code> (default).
<code>convrgr</code>	Final convergence level. This level will replace <code>convrgr</code> during final convergence. Input -1 to skip final convergence check (default).
<code>iutf</code>	Final number of iterations between wake updates. This value will replace <code>iut</code> during final convergence. Default behavior is the same as <code>iut</code> .
<code>ixterm</code>	1 to ignore wake points beyond $x/R = \text{xstop}$ , 0 to use all wake points (default).
<code>xstop</code>	If <code>ixterm=1</code> , defines $x/R$ beyond which wake points are ignored (default 5)

---

*Unsteady aerodynamics*

---

Variable name	Description
<code>DSFlag</code>	0 for no dynamic stall, 1 for Modified Boeing-Vertol model (default), 2 for Leishman-Beddoes model.
<code>PRFlag</code>	0 for no element pitch rate aerodynamic effects, 1 to include these effects (default).

---

## 3.1.2 Case inputs

Table 3.2: Available input configuration options in the `&CaseInputs` namelist.

<i>Operation point inputs</i>	
Variable name	Description
<code>RPM</code>	Rotor rotation rate (revs per minute).
<code>Ut</code>	Tip speed ratio with freestream flow speed ( $U_{\text{tip}}/U_{\infty}$ )
<code>rho</code>	Density (slug/ft <sup>3</sup> )
<code>vis</code>	Dynamic viscosity (slug/(ft s))
<code>tempr</code>	Temperature (°F)
<code>hBLRef</code>	Height above ground of the effective freestream to be used in ground shear layer model (ft).
<code>slex</code>	Exponent for ground shear layer model (Ex. 1/2 for parabolic laminar BL model, 1/7 turbulent approx., 0 for constant freestream).
<code>hAG</code>	Height above ground at turbine geometry origin point (ft). Note that the ground plane is assumed to be oriented with its normal vector in the $+y$ direction.
<code>dFS</code>	Depth below un-deflected free surface of the turbine geometry origin point (ft). Only used if free surface calculation is active. Note that the un-deflected free surface is assumed to be oriented with its normal vector in the $+y$ direction.
<code>igust</code>	1 to activate sinusoidal gust model, 0 to use nominal constant freestream model (default). The sinusoidal gust perturbation is modeled per the IEC 61400-1 Wind Turbine Design Standard.
<code>gustamp</code>	Amplitude of sinusoidal gust perturbation (m/s) used if <code>igust</code> = 1.
<code>gusttime</code>	Timescale of sinusoidal gust perturbation (s) used if <code>igust</code> = 1.
<code>gustX0</code>	Starting x location of the gust divided by reference radius (location where perturbation is zero at initial simulation time). Used if <code>igust</code> = 1.

<code>itower</code>	1 to activate the tower wake model, 0 otherwise (default). This model adds an empirically defined viscous wake deficit to the nominal freestream to model the presence of the support tower. The model currently assumes this tower to be oriented along the y-axis.
<code>tower_Npts</code>	Number of elements used to represent the tower (default 10). <code>itower = 1</code> .
<code>tower_x</code>	Tower location x coordinate divided by reference radius. Used if <code>itower = 1</code> .
<code>tower_ybot</code>	Lower tower y coordinate divided by reference radius. Used if <code>itower = 1</code> .
<code>tower_ytop</code>	Upper tower y coordinate divided by reference radius. Used if <code>itower = 1</code> .
<code>tower_D</code>	Tower diameter divided by reference radius. Used if <code>itower = 1</code> .
<code>tower_CD</code>	Tower 2D sectional drag coefficient based on tower diameter (default 1.0). Used if <code>itower = 1</code> .

---

*Geometry and airfoil data*

---

Variable name	Description
<code>GeomFilePath</code>	(string in single quotes) Path to turbine geometry input file.
<code>nSect</code>	Number of airfoil section data tables to use (default 1).
<code>AFDPath</code>	(string in single quotes) Array (comma separated) of section data file path strings (size= <code>nsect</code> ).
<code>AFDPath</code>	(string in single quotes) Array (comma separated) of section data file path strings (size= <code>nsect</code> ).
<code>WallMeshPath</code>	(string in single quotes) Path to wall mesh file in Plot3D format. Only read if <code>WallFlag = 1</code> .

---

*Other parameters*

---

Variable name	Description
<code>Jbtitle</code>	(string in single quotes) Job title.
<code>CDPar</code>	Additional parasitic interference drag coefficient based on “chord area” (chord squared) to be applied to the blade/strut interference drag calculation (default 0).

---

CTExcrM	Additional machine level excrescence torque coefficient, specified by $C_{T,ExcrM} = T_{ExcrM} / \frac{1}{\rho} U_{tip}^2 R^3$ (default 0).
---------	---

---

### 3.1.3 Output configuration

Table 3.3: Available input configuration options in the &ConfigOutputs namelist.

Variable name	Description
DiagOutFlag	1 to output diagnostic info to standard output device each iteration, 0 to omit this output (default).
Output_ELFlag	1 to output full detail element loads .csv file, 0 to omit this output (default)
Output_DSFlag	1 to output dynamic stall data, 0 to omit this output (default).
WakeElementOutFlag	1 to output wake element locations, 0 to omit this output (default).
WakeElementOutIntervalTimesteps	Number of timesteps in between wake element outputs. 5 by default.
WakeElementOutStartTimestep	Start timestep for wake element output. 1 for first timestep (default).
WakeElementOutEndTimestep	End timestep for wake element output, -1 to output until last timestep (default)
WakeGridOutFlag	1 to output induced velocity on a 3-D Cartesian grid, 0 to omit this output (default).
WakeGridOutIntervalTimesteps	Number of timesteps in between wake grid outputs. 5 by default.
WakeGridOutStartTimestep	Start timestep for wake grid output. 1 for first timestep (default).
WakeGridOutEndTimestep	End timestep for wake grid output, -1 to output until last timestep (default)
nxgrid,nygrid,nzgrid	Number of grid elements in each (x,y,z-) direction. (1, 100, 100) by default.
xgridL,xgridU,ygridL,ygridU,zgridL,zgridU	Extents of Cartesian grid to calculate induced velocity on. Defaults are: (-0.0, 0.0), (-2.0, 2.0), (-2.0, 2.0)

---

WallOutFlag	1 to output ground plane/wall panel strengths, 2 to also output velocities at panel centers. 0 for no output (default).
WallOutIntervalTimesteps	Number of timesteps in between wake grid outputs. 1 by default.
WallOutStartTimestep	Start timestep for wake grid output. 1 for first timestep (default).
WallOutEndTimestep	End timestep for wake grid output, -1 to output until last timestep (default)

---

### 3.1.4 Example namelist input file

A file format example is given below. This example can also be found in `./Test/TestCase2/TestVAWT.in` in the CACTUS repository.

```
&ConfigInputs
  GPFlag = 0
  nr = 10
  nti = 16
  convrg = .0001
  iut = 0

  ifc = 0
  interm = 0

  ntif = 16
  iutf = 1
  nric = 9
  convrgf = .0001
/End

&CaseInputs
  jbtitle = 'Test VAWT'

  rho = .002378
  vis = .3739E-6
  tempr = 60.0
  hBLRef = 56.57
  sllex = 0.0
  hAG = 15.0

  RPM = 52.0
  Ut = 5.0

  ! Turbine geometry
```

```

GeomFilePath='../TestGeom/TestVAWT.geom'

! Airfoil section data
nSect      = 1
AFDPath = '../Airfoil_Section_Data/NACA_0015.dat'
/End

&ConfigOutputs
  DiagOutFlag=1
  Output_ELFlag=1
  WakeElementOutFlag = 1
  WakeGridOutFlag = 1

! Start output at the 100th timestep
WakeGridOutStartTimestep = 100

! Stop output at the last timestep
WakeGridOutStartTimestep = -1

! Output wake velocities on every timestep
WakeGridOutIntervalTimesteps = 1

! Output a 12x40x40 grid
! (12 y-z planes at every 1.0 x, from -1.0 to 10.0
nxgrid = 12
nygrid = 40
nzgrid = 40

xgridL = -1.0
xgridU = 10.0
ygridL = -2.0
ygridU = 2.0
zgridL = -2.0
zgridU = 2.0
/End

```

## 3.2 Geometry input file

This section describes the CACTUS turbine geometry input file format, and the tools provided for generating an input file for a generic turbine configuration. CACTUS considers the turbine rotor to be comprised of lifting-line blades and optional support struts. Both blades and struts are decomposed into elements. The primary difference between the two is that the strut elements do not shed a free vortex wake and use simplified empirical models for element loads.



When generating geometry, it should be noted that the nominal freestream flow direction in CACTUS is the  $+x$  direction. If a ground plane or free surface calculation is being performed, these surfaces are oriented with their normal vectors in the  $+y$  direction.

The CACTUS geometry input file is comprised of a header section and individual sections for each of the blades and struts. Each blade and strut section contains the details of the element decomposition at the initial time in the turbine simulation. In general, the blade and strut geometry is specified by the user at the element end points. The element specific geometry maintained in the geometry input file can be calculated consistent with the element end geometry using the geometry creation tools described in Section 3.2.5.

The parameters and file format are described below. Note that each parameter occupies one line in the geometry file and array valued parameters are input as component values separated by spaces.

### 3.2.1 Header parameters

The header section of the input geometry file provides global information about the rotor. In addition, it is here that the length and area scales for non-dimensionalization are specified. Variables in this header are described in Table 3.4.

Table 3.4: Parameters in the header of the geometry input file.

Variable name	Description
<b>NBlade</b>	Number of blades.
<b>NStrut</b>	Number of struts.
<b>RotN</b>	Turbine rotation axis normal vector ( $x, y, z$ values).
<b>RotP</b>	Turbine rotation origin point ( $x, y, z$ values).
<b>RefAR</b>	Turbine reference area (for force/torque/power normalization) divided by reference radius squared.
<b>RefR</b>	Turbine reference radius (reference length dimension) for scaling dimensional output values (ft). Corresponds to the reference radius used to normalize all geometry inputs below.
<b>Type</b>	String indicating the turbine geometry generation function used to create this file. This line is for reference only, and is not used internally in CACTUS.

### 3.2.2 Blade parameters

The blade geometry is specified by the user at the element end points in terms of the locations of the blade quarter-chord line, the local blade tangent vector components, and the local chord-to-radius ratio at the initial time of the turbine simulation. The element center geometry should be calculated consistent with the element end point geometry using the geometry creation tools described in Section 3.2.5. The complete set of geometry inputs is described in Table 3.5. The parameters directly required from the user are highlighted. The other parameters are computed via the geometry creation tools. A separate block is needed to describe each blade of the turbine rotor.

Note that the direction of the element normal vector defines the sign of the flow angle of attack calculated on that element and used to interpolate the foil data tables. Positive angle of attack will be calculated when the relative flow velocity component in the element normal direction is positive. Flow angle of attack is zero when the relative flow velocity is aligned with the element tangent vector.

Table 3.5: Parameters in the blade section of the geometry input file. Independent (non-derived) quantities required for geometry creation scripts are highlighted.

Variable name	Description
<code>Blade[i]</code>	Header indicating the blade number index. This line is for reference only, and is not used internally in CACTUS.
<code>NElem</code>	Number of elements.
<code>FlipN</code>	Set to 1 to flip the element normal direction, 0 otherwise. Nominal element normal direction in CACTUS is calculated to align with the cross product of the blade tangent vector and the quarter-chord line direction.
<code>QCx</code>	Blade quarter-chord line x coordinates at element ends divided by reference radius ( <code>NElem + 1</code> values).
<code>QCy</code>	Blade quarter-chord line y coordinates at element ends divided by reference radius ( <code>NElem + 1</code> values).
<code>QCz</code>	Blade quarter-chord line z coordinates at element ends divided by reference radius ( <code>NElem + 1</code> values).
<code>tx</code>	Blade unit tangent vector (rearward chord line direction) $x$ -component at element ends ( <code>NElem + 1</code> values).
<code>ty</code>	Blade unit tangent vector (rearward chord line direction) $y$ -component at element ends ( <code>NElem + 1</code> values).
<code>tz</code>	Blade unit tangent vector (rearward chord line direction) $z$ -component at element ends ( <code>NElem + 1</code> values).

<b>CtoR</b>	Blade chord to turbine reference radius ratio at element ends ( <b>NElem</b> + 1 values).
<b>PEx</b>	Element center $x$ -coordinate divided by reference radius ( <b>NElem</b> values).
<b>PEy</b>	Element center $y$ -coordinate divided by reference radius ( <b>NElem</b> values).
<b>PEz</b>	Element center $z$ -coordinate divided by reference radius ( <b>NElem</b> values).
<b>tEx</b>	Element unit tangent vector (rearward chord line direction) $x$ -component ( <b>NElem</b> values).
<b>tEy</b>	Element unit tangent vector (rearward chord line direction) $y$ -component ( <b>NElem</b> values).
<b>tEz</b>	Element unit tangent vector (rearward chord line direction) $z$ -component ( <b>NElem</b> values).
<b>nEx</b>	Element unit normal vector $x$ -component ( <b>NElem</b> values).
<b>nEy</b>	Element unit normal vector $y$ -component ( <b>NElem</b> values).
<b>nEz</b>	Element unit normal vector $z$ -component ( <b>NElem</b> values).
<b>sEx</b>	Element unit spanwise vector $x$ -component ( <b>NElem</b> values).
<b>sEy</b>	Element unit spanwise vector $y$ -component ( <b>NElem</b> values).
<b>sEz</b>	Element unit spanwise vector $z$ -component ( <b>NElem</b> values).
<b>ECtoR</b>	Element chord to turbine reference radius ratio ( <b>NElem</b> values).
<b>EAreaR</b>	Element area divided by turbine reference radius squared ( <b>NElem</b> values).
<b>iSect</b>	Index of the foil section data to be applied to each element ( <b>NElem</b> values). This index corresponds to the order in which the foil section data table files are supplied in the namelist input file.

### 3.2.3 Strut parameters

The strut geometry is specified by the user at the element end points in terms of the locations of the strut mid-chord line, and the local chord-to-radius ratio at the initial time of the turbine simulation. The element center geometry should be calculated consistent with the element end geometry using the geometry creation tools described in Section 3.2.5. The parameters directly required from the user are bolded in the table below, with the others being filled out by the geometry creation tools.

There should be a separate strut data block for each strut in the turbine rotor.

Table 3.6: Parameters in the strut section of the geometry input file. Independent (non-derived) quantities required for geometry creation scripts are highlighted.

Variable name	Description
<b>Strut[i]</b>	Header indicating the strut number index. This line is for reference only, and is not used internally in CACTUS.
<b>NElem</b>	Number of elements.
<b>TtoC</b>	Strut thickness to chord ratio (single value, assumed constant over strut).
<b>MCx</b>	Strut mid-chord $x$ -coordinate at element ends divided by reference radius ( <b>NElem</b> + 1 values).
<b>MCy</b>	Strut mid-chord $y$ -coordinate at element ends divided by reference radius ( <b>NElem</b> + 1 values).
<b>MCz</b>	Strut mid-chord $z$ -coordinate at element ends divided by reference radius ( <b>NElem</b> + 1 values).
<b>CtoR</b>	Strut chord to turbine reference radius ratio at element ends ( <b>NElem</b> + 1 values).
<b>PEx</b>	Element center $x$ -coordinate divided by reference radius ( <b>NElem</b> values).
<b>PEy</b>	Element center $y$ -coordinate divided by reference radius ( <b>NElem</b> values).
<b>PEz</b>	Element center $z$ -coordinate divided by reference radius ( <b>NElem</b> values).
<b>sEx</b>	Element unit spanwise vector $x$ -component ( <b>NElem</b> values).
<b>sEy</b>	Element unit spanwise vector $y$ -component ( <b>NElem</b> values).
<b>sEz</b>	Element unit spanwise vector $z$ -component ( <b>NElem</b> values).
<b>ECtoR</b>	Element chord to turbine reference radius ratio ( <b>NElem</b> values).
<b>EAreaR</b>	Element area divided by turbine reference radius squared ( <b>NElem</b> values).
<b>BIndS</b>	Index of the blade to which the first strut element connects. Used for blade-strut interference drag calculation. Set to zero if start element connected to rotor shaft.
<b>EIndS</b>	Index of the element on the above blade at which the first strut element connects. Used for blade-strut interference drag calculation. Set to zero if first element connected to rotor shaft.

BIndE	Index of the blade to which the last strut element connects. Used for blade-strut interference drag calculation. Set to zero if last element is connected to rotor shaft.
EIndE	Index of the element on the above blade to which the last strut element connects. Used for blade-strut interference drag calculation. Set to zero if last element is connected to rotor shaft.

### 3.2.4 Example geometry input file

An example rotor for a three-bladed two-strut rotor is shown below. This geometry file can be found in `./Test/TestGeom/TestVAWT.geom`.

```

NBlade: 2
NStrut: 2
RotN: 0.00000e+00 1.00000e+00 0.00000e+00
RotP: 0.00000e+00 0.00000e+00 0.00000e+00
RefAR: 3.52000e+00
RefR: 3.15000e+01
Type: VAWT
Blade 1:
  NElem: 5
  FlipN: 0
  QCx: -1.25936e-02 -1.25936e-02 -1.25936e-02 -1.25936e-02 -1.25936e-02
        ↪ -1.25936e-02
  QCy: 0.00000e+00 5.28000e-01 1.05600e+00 1.58400e+00 2.11200e+00
        ↪ 2.64000e+00
  QCz: -0.00000e+00 -6.40000e-01 -9.60000e-01 -9.60000e-01 -6.40000e-01
        ↪ -0.00000e+00
  tx: 1.00000e+00 1.00000e+00 1.00000e+00 1.00000e+00 1.00000e+00
        ↪ 1.00000e+00
  ty: 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
        ↪ 0.00000e+00
  tz: 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
        ↪ 0.00000e+00
  CtoR: 7.40800e-02 7.40800e-02 7.40800e-02 7.40800e-02 7.40800e-02
        ↪ 7.40800e-02
  PEx: -1.25936e-02 -1.25936e-02 -1.25936e-02 -1.25936e-02 -1.25936e-02
  PEy: 2.64000e-01 7.92000e-01 1.32000e+00 1.84800e+00 2.37600e+00
  PEz: -3.20000e-01 -8.00000e-01 -9.60000e-01 -8.00000e-01 -3.20000e-01
  tEx: 1.00000e+00 1.00000e+00 1.00000e+00 1.00000e+00 1.00000e+00
  tEy: 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
  tEz: 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
  nEx: -0.00000e+00 -0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00
  nEy: 7.71373e-01 5.18302e-01 0.00000e+00 -5.18302e-01 -7.71373e-01

```

```

nEz:  6.36383e-01  8.55198e-01  1.00000e+00  8.55198e-01  6.36383e-01
sEx:  -0.00000e+00 -0.00000e+00 -0.00000e+00 -0.00000e+00 -0.00000e+00
sEy:  -6.36383e-01 -8.55198e-01 -1.00000e+00 -8.55198e-01 -6.36383e-01
sEz:  7.71373e-01  5.18302e-01 -0.00000e+00 -5.18302e-01 -7.71373e-01
ECtoR:  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e
      ↪ -02
EAreaR:  6.14634e-02  4.57371e-02  3.91142e-02  4.57371e-02  6.14634e
      ↪ -02
iSect:  1  1  1  1  1
Blade 2:
NElem:  5
FlipN:  0
QCx:  1.25936e-02  1.25936e-02  1.25936e-02  1.25936e-02  1.25936e-02
      ↪ 1.25936e-02
QCy:  0.00000e+00  5.28000e-01  1.05600e+00  1.58400e+00  2.11200e+00
      ↪ 2.64000e+00
QCz:  1.54227e-18  6.40000e-01  9.60000e-01  9.60000e-01  6.40000e-01
      ↪ 1.54227e-18
tx:  -1.00000e+00 -1.00000e+00 -1.00000e+00 -1.00000e+00 -1.00000e+00
      ↪ -1.00000e+00
ty:  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00
      ↪ 0.00000e+00
tz:  -1.22465e-16 -1.22465e-16 -1.22465e-16 -1.22465e-16 -1.22465e-16
      ↪ -1.22465e-16
CtoR:  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e-02
      ↪ 7.40800e-02
PEx:  1.25936e-02  1.25936e-02  1.25936e-02  1.25936e-02  1.25936e-02
PEy:  2.64000e-01  7.92000e-01  1.32000e+00  1.84800e+00  2.37600e+00
PEz:  3.20000e-01  8.00000e-01  9.60000e-01  8.00000e-01  3.20000e-01
tEx:  -1.00000e+00 -1.00000e+00 -1.00000e+00 -1.00000e+00 -1.00000e+00
tEy:  2.41486e-19 -9.83382e-19  0.00000e+00  9.83382e-19 -2.41486e-19
tEz:  -1.22172e-16 -1.23061e-16 -1.22465e-16 -1.23061e-16 -1.22172e-16
nEx:  7.79344e-17  1.04732e-16  1.22465e-16  1.04732e-16  7.79344e-17
nEy:  7.71373e-01  5.18302e-01  0.00000e+00 -5.18302e-01 -7.71373e-01
nEz:  -6.36383e-01 -8.55198e-01 -1.00000e+00 -8.55198e-01 -6.36383e-01
sEx:  9.40865e-17  6.46235e-17 -0.00000e+00 -6.46235e-17 -9.40865e-17
sEy:  -6.36383e-01 -8.55198e-01 -1.00000e+00 -8.55198e-01 -6.36383e-01
sEz:  -7.71373e-01 -5.18302e-01 -0.00000e+00  5.18302e-01  7.71373e-01
ECtoR:  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e
      ↪ -02
EAreaR:  6.14634e-02  4.57371e-02  3.91142e-02  4.57371e-02  6.14634e
      ↪ -02
iSect:  1  1  1  1  1
Strut 1:
NElem:  5
TtoC:  1.50000e-01

```

```

MCx:  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00
      ↪  0.00000e+00
MCy:  1.32000e+00  1.32000e+00  1.32000e+00  1.32000e+00  1.32000e+00
      ↪  1.32000e+00
MCz:  -0.00000e+00 -1.92000e-01 -3.84000e-01 -5.76000e-01 -7.68000e-01
      ↪ -9.60000e-01
CtoR:  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e-02
      ↪  7.40800e-02
PEx:  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00
PEy:  1.32000e+00  1.32000e+00  1.32000e+00  1.32000e+00  1.32000e+00
PEz:  -9.60000e-02 -2.88000e-01 -4.80000e-01 -6.72000e-01 -8.64000e-01
sEx:  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00
sEy:  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00
sEz:  -1.00000e+00 -1.00000e+00 -1.00000e+00 -1.00000e+00 -1.00000e+00
ECToR:  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e
      ↪ -02
EAreaR:  1.42234e-02  1.42234e-02  1.42234e-02  1.42234e-02  1.42234e
      ↪ -02
BIndS:  0
EIndS:  0
BIndE:  1
EIndE:  3
Strut 2:
NElem:  5
TtoC:  1.50000e-01
MCx:  0.00000e+00 -2.35132e-17 -4.70264e-17 -7.05397e-17 -9.40529e-17
      ↪ -1.17566e-16
MCy:  1.32000e+00  1.32000e+00  1.32000e+00  1.32000e+00  1.32000e+00
      ↪  1.32000e+00
MCz:  0.00000e+00  1.92000e-01  3.84000e-01  5.76000e-01  7.68000e-01
      ↪  9.60000e-01
CtoR:  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e-02
      ↪  7.40800e-02
PEx:  -1.17566e-17 -3.52698e-17 -5.87830e-17 -8.22963e-17 -1.05809e-16
PEy:  1.32000e+00  1.32000e+00  1.32000e+00  1.32000e+00  1.32000e+00
PEz:  9.60000e-02  2.88000e-01  4.80000e-01  6.72000e-01  8.64000e-01
sEx:  -1.22465e-16 -1.22465e-16 -1.22465e-16 -1.22465e-16 -1.22465e-16
sEy:  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00
sEz:  1.00000e+00  1.00000e+00  1.00000e+00  1.00000e+00  1.00000e+00
ECToR:  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e-02  7.40800e
      ↪ -02
EAreaR:  1.42234e-02  1.42234e-02  1.42234e-02  1.42234e-02  1.42234e
      ↪ -02
BIndS:  0
EIndS:  0
BIndE:  2

```

### 3.2.5 Geometry creation tools

A set of MATLAB tools has been created to allow the user to generate a CACTUS geometry input file for an arbitrary turbine rotor. This set of MATLAB scripts is located in the `./CreateGeom/` folder in the CACTUS source code directory. Note that these scripts should also run without modification under GNU Octave (a free software package that mimics much of the basic MATLAB syntax), with the possible exception of the plotting functions in the script `PlotTurbineGeom.m`.

The script `CreateTurbine.m` generates an empty turbine geometry structure and optionally fills it out with data for a parameterized generic turbine type. See the comments at the top of the file for details. The user can then modify the data in the created geometry structure as necessary to represent their particular problem. When defining blade and strut geometry, the user should first fill out the *independent* fields (highlighted in Tables 3.5 and 3.6), and then use the scripts `CalcBEGeom.m` and `CalcSEGeom.m` to fill out the blade and strut element geometry fields.

Once the turbine geometry structure has been finalized, the script `WriteTurbineGeom.m` will create the CACTUS geometry input file. The script `ReadTurbineGeom.m` will read an existing CACTUS geometry input file and create a corresponding geometry structure in MATLAB for further modification.

The script `PlotTurbineGeom.m` will plot the geometry contained in a turbine geometry structure at a particular phase angle of rotation around the turbine rotation axis. The plotting functions in this script have not been verified to work in GNU Octave.

An example script that creates and plots turbine rotor geometry for a vertical axis wind turbine is provided in Appendix A.

## 3.3 Airfoil input file

Foil data table files contain the force coefficient data and dynamic stall parameters for a foil to be used in the calculation. The user can specify multiple foil data table files in the CACTUS namelist input file and they are indexed in the order they appear in the namelist input file. The `iSect` parameter in the geometry input file identifies the index of the foil data table to be applied to each blade element section.

The foil coefficients are input as lift, drag, and pitching moment (about the 25% chord point) coefficients (per span) as a function of angle of attack (AOA) and Reynolds number. A maximum of 20 Reynolds number values are allowed. A maximum of 1000 AOA values for each Reynolds number are allowed. The AOA data must go from -180 deg to 180 deg. The sign convention for AOA is such that positive AOA on a blade element is generated by positive relative flow velocity in the positive element normal direction. Note that the cross-flow turbine geometry generator provided with CACTUS creates element normal vectors in the machine inward direction. The axial flow turbine



geometry generator creates element normal vectors in the machine rearward (downwind) direction at zero blade incidence angle. Airfoil data is split into a header section, described in Table 3.7, and a number of data blocks, described in Table 3.7, each corresponding to a different Reynolds number. An example airfoil data file is shown in Listing 3.1. A number of foil data table files are provided in the `.Test/Airfoil_Section_Data/` folder in the CACTUS source code directory.

Table 3.7: Header block for airfoil input files

Variable name	Description
<b>Title</b>	Title for this foil data table. For reference only; not used internally in CACTUS.
<b>ThicknessToChordRatio</b>	Thickness to chord ratio for this foil section.
<b>ZeroLiftAOA</b>	Angle of attack (deg) at zero lift for this section.
<b>ReverseCamberDirection</b>	Set to 1 to reverse the orientation of a non-symmetric foil with respect to the normal vector of the blade element section to which it is applied.

Table 3.8: Reynolds number data block for airfoil input files

Variable name	Description
<b>BVDyn. StallModel</b>	The Boeing-Vertol dynamic stall model uses reference angle of attack values to switch from a steady attached flow state to a dynamic stalled state. While these are denoted as “stall” AOA values, they should be set back from the foil stall AOA such that the lift coefficient is still fairly linear with AOA at this point. Generally, a value 50 - 75% of the way between the zero lift AOA and the stall AOA works well.
<b>LBDyn. StallModel</b>	The Leishman-Beddoes dynamic stall model uses a reference lift slope and a critical lift coefficient value to indicate the onset of leading edge stall. The critical lift coefficient value should be approximately equal to the value of lift coefficient that would have been obtained at the foil stall AOA, had the lift coefficient remained linear with AOA (with slope given by the reference lift coefficient slope).
<b>ForceandMomentCoefficients</b>	Force and moment coefficient data at angle of attack from -180 deg to 180 deg. (AOA, lift coefficient, drag coefficient, pitching moment coefficient about the 25% chord point).

Listing 3.1: An example airfoil data file.

```

Title: AFTitle
Thickness to Chord Ratio: 0.2
Zero Lift AOA (deg): 0.0
Reverse Camber Direction: 0

Reynolds Number: 1e6
BV Dyn. Stall Model - Positive Stall AOA (deg): 10
BV Dyn. Stall Model - Negative Stall AOA (deg): -10
LB Dyn. Stall Model - Lift Coeff. Slope at Zero Lift AOA (per radian): 6.28
LB Dyn. Stall Model - Positive Critical Lift Coeff.: 1.3
LB Dyn. Stall Model - Negative Critical Lift Coeff.: -1.3
AOA (deg) CL CD Cm25
-180.0 0.0 1.0 0.0
... ..
180.0 0.0 1.0 0.0

Reynolds Number: 5e6
...

```

## 3.4 Wall input file

If `WPFlag = 1`, CACTUS will include the influence of a specified wall geometry. The wall geometry must be generated externally; the path to this wall geometry is specified with the `WallMeshPath` namelist variable.

The wall mesh file is a multi-block, formatted (ASCII), Plot3d structured mesh. More details on the Plot3d file format can be found at <http://www.grc.nasa.gov/WWW/wind/valid/plot3d.html>. Each block contains one structured mesh, and thus complex wall geometries can be specified with a single mesh file. The normals of each cell defined by the node locations must face inward toward the domain in which the wake elements lie. The format of this file is demonstrated in Listing 3.2. For inspection, this file format may easily be read in by ParaView.

Listing 3.2: Example file format for a multi-block Plot3d mesh file.

```

[nblocks]
[nx] [ny] [nz]
[x_1]
[x_2]
...
[x_nx]
[y_1]
[y_2]
...

```

```
[y_ny]  
[z_1]  
[z_2]  
...  
[z_nz]  
EOF
```

## 4 Output description

By default, CACTUS will output rotor-integrated loads to a text file. CACTUS is capable of saving much more information, such as blade-integrated loads, blade element loads, wall panel data, field velocities, and the complete state of vortex filaments describing the wake. These outputs can be enabled through the appropriate input file flags, described in Table 3.3. This section describes the format of the various output files.

### 4.1 Revolution-averaged rotor-integrated loads

CACTUS writes revolution-averaged rotor-integrated loads for each revolution to the comma delimited file `_RevData.csv`. The output columns of this file are described in Table 4.1.

Table 4.1: Revolution-averaged output data.

Variable name	Description
Rev	Revolution number
PowerCoeff.(-)	Revolution average machine power coefficient
TipPowerCoeff.(-)	Revolution average machine power coefficient normalized with $U_{\text{tip}}$ instead of $U_{\infty}$
TorqueCoeff.(-)	Revolution average torque coefficient
FxCoeff.(-)	Revolution average $x$ -component of force coefficient
FyCoeff.(-)	Revolution average $y$ -component of force coefficient
FzCoeff.(-)	Revolution average $z$ -component of force coefficient
Power(kW)	Revolution average machine power
Torque(ft-lbs)	Revolution average machine torque

## 4.2 Blade-integrated loads

If `Output_ELFlag = 1`, blade-integrated loads are written at each to the comma delimited file `_TimeData.csv`. The output columns of this file are described in Table 4.2.

Table 4.2: Blade integrated loads.

Variable name	Description
<code>NormalizedTime(-)</code>	Normalized simulation time, $t_N = tU_\infty/R$
<code>Theta(rad)</code>	Turbine rotational phase angle
<code>Rev</code>	Revolution number
<code>TorqueCoeff(-)</code>	Torque coefficient
<code>PowerCoeff(-)</code>	Power coefficient
<code>FxCoeff.(-)</code>	$x$ -component of force coefficient
<code>FyCoeff.(-)</code>	$y$ -component of force coefficient
<code>FzCoeff.(-)</code>	$z$ -component of force coefficient
<code>BladeFxCoeff(-)</code>	Contribution to $x$ -component of force coefficient from blade
<code>BladeFyCoeff(-)</code>	Contribution to $y$ -component of force coefficient from blade
<code>BladeFzCoeff(-)</code>	Contribution to $z$ -component of force coefficient from blade
<code>BladeTorqueCoeff(-)</code>	Contribution to torque coefficient from blade
<code>StrutFxCoeff(-)</code>	Contribution to $x$ -component of force coefficient from strut
<code>StrutFyCoeff(-)</code>	Contribution to $y$ -component of force coefficient from strut
<code>StrutFzCoeff(-)</code>	Contribution to $z$ -component of force coefficient from strut
<code>StrutTorqueCoeff(-)</code>	Contribution to torque coefficient from strut

## 4.3 Blade element loads

If `Output_ELFlag = 1`, loads on the individual blade elements at each time step are written to the comma delimited file `_ElementData.csv`. The output columns of this file are described in Table 4.3.

Table 4.3: Temporal blade element loads.

Variable name	Description
NormalizedTime(-)	Normalized simulation time, $t_N = tU_\infty/R$
Theta(rad)	Turbine rotational phase angle
Blade	Blade number
Element	Element number
Rev	Revolution number
AOA25(deg)	Local flow angle of attack, defined at element quarter-chord location.
AOA50(deg)	Reference 50% chord flow angle of attack. Different from AOA25 when element is rotating in the local spanwise direction.
AOA75(deg)	Reference 75% chord flow angle of attack. Different from AOA25 when element is rotating in the local spanwise direction.
AdotNorm(-)	Normalized AOA rate $\dot{\alpha}_{\text{norm}} = \dot{\alpha}c/2U_{\text{loc}}$
Re(-)	Element Reynolds number based on element chord
Mach(-)	Element Mach number
Ur(-)	Local flow speed ratio with freestream, $U_r = U_{\text{loc}}/U_\infty$
CL(-)	Element lift coefficient, $C_L = L/\frac{1}{2}\rho U_{\text{loc}}^2 A_E$
CD(-)	Element drag coefficient, $C_D = D/\frac{1}{2}\rho U_{\text{loc}}^2 A_E$
CM25(-)	Element pitching moment coefficient about the quarter-chord location, $C_{M,25} = L/\frac{1}{2}\rho U_{\text{loc}}^2 A_E c$
CLCirc(-)	Circulatory component of element lift coefficient, $C_{L,\text{circ}} = L_{\text{circ}}/\frac{1}{2}\rho U_{\text{loc}}^2$
CN(-)	Element normal force coefficient, $C_N = N/\frac{1}{2}\rho U_{\text{loc}}^2 A_E$
CT(-)	Element tangential force coefficient, $C_T = T/\frac{1}{2}\rho U_{\text{loc}}^2 A_E$
Fx(-)	Contribution to $x$ -component of force coefficient from element
Fy(-)	Contribution to $y$ -component of force coefficient from element
Fz(-)	Contribution to $z$ -component of force coefficient from element
te(-)	Contribution to torque coefficient from element

## 4.4 Ground plane, wall, and free surface output

### 4.4.1 Ground plane data

If either a ground plane calculation or wall calculation is being performed ( $\text{GPFlag} = 1$  or  $\text{WPFlag} = 1$ ), information about the wall may optionally be written to file. The verbosity of this output file is specified with the `WallOutFlag` namelist variable. If `WallOutFlag` = 1, the source panel strength is written to file. If `WallOutFlag` = 2, the induced velocities at each panel center are also computed and written to file.

The interval of ground plane/wall data output may be controlled via the variables described in Table 3.3.

- `WallOutIntervalTimesteps`,
- `WallOutStartTimestep`, and
- `WallOutEndTimestep`

The ground plane/wall data output filenames have the format `[runname]_WPData_[timestepnumber].tp`. The files themselves use a TecPlot file format (finite-element structured mesh, cell-centered field data). The output fields are described in Tables 4.4 and 4.5. These files may be easily read in to ParaView for post-processing.

Table 4.4: Wall outputs.

Variable name	Description
<code>X</code>	$x$ -coordinate of panel node normalized by $R$
<code>Y</code>	$y$ -coordinate of panel node normalized by $R$
<code>Z</code>	$z$ -coordinate of panel node normalized by $R$
<code>sigma(-)</code>	Source density of the panel normalized by $U_\infty$
<code>u</code>	$x$ -velocity at the panel center normalized by $U_\infty$
<code>v</code>	$y$ -velocity at the panel center normalized by $U_\infty$
<code>w</code>	$z$ -velocity at the panel center normalized by $U_\infty$

### 4.4.2 Free surface data

If a free surface calculation is being performed ( $\text{FSFlag} = 1$ ), information about the free surface is automatically written to file. A single comma delimited file is written which contains the information described in Table 4.5 *averaged over the final revolution*.

Table 4.5: Free surface outputs.

Variable name	Description
X/R(-)	$x$ -coordinate of the panel center normalized by $R$
Y/R(-)	$y$ -coordinate of the panel center normalized by $R$
Z/R(-)	$z$ -coordinate of the panel center normalized by $R$
U/Uinf(-)	Wall tangential velocity (nominal freestream direction) normalized by $U_\infty$
dH/R(-)	Free surface height (above un-deflected height) normalized by $R$

## 4.5 Field velocities

If `WakeGridOutFlag=1`, the induced velocity field on a Cartesian grid is computed and written to a file. This output data is split into multiple files, each file containing the field data at a single timestep.

This can add considerable time to the simulation, since the induced velocity due to every wake node must be calculated at every point in the specified Cartesian grid.

The output filenames have the format `[runname]_WallData_[timestepnumber].csv`.

Table 4.6: Field velocity output data.

Variable name	Description
NormalizedTime(-)	Normalized simulation time, $t_N = tU_\infty/R$
x/R(-)	$x$ -coordinate of the data point normalized by $R$
y/R(-)	$y$ -coordinate of the data point normalized by $R$
z/R(-)	$z$ -coordinate of the data point normalized by $R$
U/Uinf(-)	$x$ -component of induced velocity
V/Uinf(-)	$y$ -component of induced velocity
W/Uinf(-)	$z$ -component of induced velocity
Ufs/Uinf(-)	$x$ -component of free-stream velocity
Vfs/Uinf(-)	$y$ -component of free-stream velocity
Wfs/Uinf(-)	$z$ -component of free-stream velocity



Table 4.7: Vortex filament output data.

Variable name	Description
<code>NormalizedTime(-)</code>	Normalized simulation time, $t_N = tU_\infty/R$
<code>NodeID</code>	A unique ID given to each distinct filament node (useful for tracing a particle's path in time)
<code>OriginNode</code>	ID of the element node from which this filament was generated
<code>x/R(-)</code>	$x$ -coordinate of the data point normalized by $R$
<code>y/R(-)</code>	$y$ -coordinate of the data point normalized by $R$
<code>z/R(-)</code>	$z$ -coordinate of the data point normalized by $R$
<code>U/Uinf(-)</code>	$x$ -component of induced velocity
<code>V/Uinf(-)</code>	$y$ -component of induced velocity
<code>W/Uinf(-)</code>	$z$ -component of induced velocity

## 4.6 Vortex filament data

If `WakeElementOutFlag` = 1, information about each vortex filament is written to a file. This output data is split into multiple files, each file containing the filament data at a single timestep. Output data is defined at the endpoints of the vortex filaments.

The output filenames have the format `[runname]_WakeData_[timestepnumber].csv`.

## 5 Details

### 5.1 OpenMP acceleration

OpenMP acceleration is enabled in the main Biot-Savart calculation loop as well as in the calculation of field velocities as described in 5.5. OpenMP is enabled by default, provided the appropriate flags are selected during compilation and that the OpenMP libraries are properly linked.

If OpenMP is enabled, CACTUS will write to standard out a few lines stating so. If CACTUS has not been compiled correctly with OpenMP flags and libraries, or if OpenMP is otherwise disabled, those lines about OpenMP will be omitted.

If the number of threads displayed is less than the number of cores/threads available on the machine, check that the environment variable `OMP_NUM_THREADS` is correctly set.

```
[phil@localhost CACTUS-CK]$ ./bin/cactus
```

```
Starting CACTUS Execution.
```

```
-----
```

```
OpenMP is Enabled.
```

```
Executing with           2  threads.
```

Please call the program with the name of the input file on the command line. Ex

```
→ . CACTUS INPUTFILE.in
```

# Bibliography

- [1] Joseph Katz and Allen Plotkin. *Low-speed aerodynamics*. Cambridge University Press, 2001.
- [2] Jonathan Murray and Matthew Barone. The Development of CACTUS, a Wind and Marine Turbine Performance Simulation Code. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, number January, pages 1–21, Reston, Virginia, January 2011. American Institute of Aeronautics and Astronautics.
- [3] J. H. Strickland, B. T. Webster, and T. Nguyen. A Vortex Model of the Darrieus Turbine: An Analytical and Experimental Study. *Journal of Fluids Engineering*, 101(4):500, 1979.

## A Example geometry script

An example geometry creation MATLAB script for is provided here for a vertical axis wind turbine. This example can also be found in Test/TestGeom/TestVAWT.m in the CACTUS repository.

```
clear
close all

% Creates test VAWT geometry file

% Add geom creation scripts to path
path(path,'.././CreateGeom');

% Params
R=31.5;           % Center radius (ft)
HR=2.64;          % Height to radius ratio
CRr=0.07408;      % Root chord to radius
eta=.42;          % Blade mount point ratio (mount point behind leading edge
    ↪ as a fraction of chord)
NBlade=2;
NBElem=5;
NStrut=2;         % number of struts
NSElem=5;
CRs=CRr;          % strut chord to radius
TCs=.15;          % strut thickness to chord

% Output filename
FN='TestVAWT.geom';

% Plot data?
PlotTurbine=1;

% Convert
dToR=pi/180;

% Create basic parabolic blade VAWT
Type='VAWT';
BShape=1;
```

---

```

T=CreateTurbine(NBlade,NBElem,NStrut,NSElem,R,[],[],[],Type,1,CRr,HR,eta,BShape
    ↪ ,CRs,TCs);

% Write geom file
WriteTurbineGeom(FN,T);

% Plot if desired
if PlotTurbine

    % Plot animated turbine rotation
    XLim=[-4,4];
    YLim=[-2,4];
    ZLim=[-4,4];

    % Plot controls
    PlotVec=1;
    SFVec=.5;
    Trans=.5;

    hf=figure(1);
    set(hf,'Position',[303 124 956 610])
    set(gca,'Position',[5.2743e-002 5.1245e-002 8.9979e-001 8.8141e-001])
    set(gca,'CameraPosition',[-52.1999 30.4749 62.2119])
    set(gca,'CameraUpVector',[1.8643e-001 9.7433e-001 -1.2615e-001])
    set(gca,'CameraViewAngle',6.3060e+000)
    grid on
    set(gcf,'Color','white');
    hl=light('Position',[-1,0,0]);
    set(gca,'Color','white');
    set(gca,'DataAspectRatio',[1,1,1])
    set(gca,'XLim',XLim,'YLim',YLim,'ZLim',ZLim)

    HIn=[];
    PhasePlot=linspace(0,2*pi,150);
    for i=1:length(PhasePlot)
        H=PlotTurbineGeom(T,hf,PhasePlot(i),HIn,Trans,PlotVec,SFVec);
        HIn=H;
        pause(.01);
    end
end
end

```