3dVPM - 3D Unsteady Vortex Panel Method

Generated by Doxygen 1.8.6

Mon May 11 2015 17:24:03

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 Domain Class Reference

Domain class to compute velocity off-surface.

```
#include <domain.hpp>
```

Public Member Functions

- void **set_domain_ranges** (const int i, const int j, const int k)
- int get_IMAX () const

  returns maximum nodes in x direction
- int get_JMAX () const

  returns maximum nodes in y direction
- int get_KMAX () const

  returns maximum nodes in z direction
- int n_nodes () const

  returns total number of nodes

Public Attributes

- std::vector< vector3d > nodes

  point data of the domain mesh file

3.1.1 Detailed Description

Domain class to compute velocity off-surface.

The documentation for this class was generated from the following files:

- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/domain.hpp
- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/domain.cpp

3.2 matlab_writer Class Reference

writes output in matlab format. DO NOT USE - NOT a standard format

```
#include <matlab_writer.hpp>
```

Public Member Functions

- void **write_surface_mesh** (std::string filename, const std::shared_ptr< Surface > surface)
- template<class T >
  void **write_surface_data** (std::string filename, const std::shared_ptr< Surface > surface, const std::vector< T > &data, std::string name, bool writemesh=false)
- template<class T >
  void **write_domain_data** (std::string filename, std::shared_ptr< Domain > domain, std::vector< T > data, std::string name, bool writemesh)

3.2.1 Detailed Description

writes output in matlab format. DO NOT USE - NOT a standard format

The documentation for this class was generated from the following files:

- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/matlab_writer.hpp
- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/matlab_writer.cpp

3.3 Parameters Class Reference

Defines certain parameters used in the panel method.

```
#include <parameters.hpp>
```

Static Public Attributes

- static double inversion_tolerance = 1e-12

  tolerance check for division
- static double farfield_factor = 10.0

  farfield factor
- static double trailing_edge_wake_shed_factor = 0.25

  controls the trailing edge wake panel distance
- static bool unsteady_problem = false

  decides whether problem is steady or unsteady
- static bool static_wake = false

  decides whether to use static wake or force free wake model
- static double static_wake_length = 1.0

  sets static wake length
- static bool use_vortex_core_model = false

  decides whether to use vortex core model

3.3.1 Detailed Description

Defines certain parameters used in the panel method.

The documentation for this class was generated from the following files:

- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/parameters.hpp
- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/parameters.cpp

3.4 PLOT3D Class Reference

PLOT3D class to read plot3D mesh file and manipulates surface object.

`#include <plot3d.hpp>`

Public Member Functions

- void set_surface_filename (std::string)

  set file name to read
- void set_surface (std::shared_ptr< Surface >)

  connect to surface object
- void flip_normals (bool)

  Flips the normals of the surface.
- void read_surface (std::string name)

  read the plot3d mesh file
- void build_topology ()

  buid topology from mesh file
- void set_domain (std::shared_ptr< Domain >)

  connect to domain object
- void read_domain (std::string name)

  read mesh file for the domain

3.4.1 Detailed Description

PLOT3D class to read plot3D mesh file and manipulates surface object.

The documentation for this class was generated from the following files:

- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/plot3d.hpp
- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/plot3d.cpp

3.5 Solver Class Reference

Solver class for 3D unsteady panel method.

`#include <solver.hpp>`

Public Member Functions

- Solver (int argC, char **argS)

  constructor - takes command line arguments
- void add_surface (const std::shared_ptr< Surface >)

  attaches surface object with solver
- void add_wake (const std::shared_ptr< Wake >)

  attaches wake object with solver
- void add_logger (const std::shared_ptr< vtk_writer >)

  attaches vtk-logger object with solver
- void add_logger (const std::shared_ptr< matlab_writer >)

  attaches matlab-logger object with solver
- void set_free_stream_velocity (const vector3d &)

Set free stream velocity.

- void set_reference_velocity (const vector3d &)

  Set reference velocity.
- void set_fluid_density (const double value)

  Set fluid density velocity.
- void solve (const double dt, int iteration=0)

  driver function of solver
- void convect_wake (const double &dt)

  convects the wake with local induced velocity
- void compute_domain_velocity (const std::shared_ptr< Domain > domain)

  compute velocity at each node in the domain
- void finalize_iteration ()

  performs some post-solution operations
- vector3d get_body_forces () const

  Returns body force vectors.
- vector3d get_body_force_coefficients () const

  Returns body force coefficents.
- double get_pressure_coefficient (const int panel) const

  Returns pressure coefficients.
- void write_output (const int &interation) const

  Write output to a file.
- void write_matlab_output () const

  Write output to matlab format - do not use!

3.5.1 Detailed Description

Solver class for 3D unsteady panel method.

Solver class calculates the solution of the given problem. Takes input in terms of surface, wake, configurational parameters such as surface velocity, free stream velocity, reference velocity, density, etc. Calculates the solution in terms of surface velocity, pressure, body forces (and force coefficients). Also write output to vtk files.

3.5.2 Member Function Documentation

3.5.2.1 void Solver::solve (const double *dt,* int *iteration* = 0)

driver function of solver

The function is responsible for solution of given problem. The function calculates and influence coefficents, applied boundary conditions, builds system of equations, solves them, post-processes solution, write them in output file.

The documentation for this class was generated from the following files:

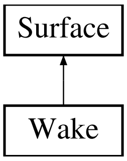
- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/solver.hpp
- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/solver.cpp

3.6 Surface Class Reference

Surface class to represent geometry and perform various operations on it.

```
#include <surface.hpp>
```

Inheritance diagram for Surface:



Public Member Functions

- int n_panels () const

  returns number of panels in the current surface
- int n_nodes () const

  returns total number of nodes in the current surface
- void compute_panel_components ()

  Computes the panel components.
- vector3d & get_collocation_point (int panel)

  returns panel's collocation point by reference
- vector3d get_collocation_point (int panel) const

  returns panel's collocation point
- void translate_surface (const vector3d &dX)

  translates the surface by distance dX
- void rotate_surface (vector3d dTheta, bool isRadian)

  rotates the surface by an angle
- void set_linear_velocity (const vector3d &)

  Set the linear velocity of the surface.
- void set_angular_velocity (vector3d vel, bool isRadian_sec)

  Set angular velocity of the body.
- int n_trailing_edge_nodes () const

  returns number of trailing edge nodes
- int n_trailing_edge_panels () const

  returns nuber of trailing edge panels
- vector3d get_trailing_edge_bisector (const int) const

  returns trailing edge bisecor
- vector3d get_kinematic_velocity (const vector3d &) const

  returns kinematic velocity at a point on surface
- vector3d get_panel_normal (const int) const

  return panel's normal vector
- double compute_source_panel_influence (const int panel, const vector3d &node) const

  computes the influence coefficient due to source panel
- double compute_doublet_panel_influence (const int panel, const vector3d &node) const

  computes the influence coefficient due to doublet panel
- std::pair< double, double > compute_source_doublet_panel_influence (const int panel, const vector3d &node) const

  computes the influence coefficient due to source and doublet panel
- vector3d transform_point_panel (int panel, const vector3d &x) const

  transforms a point in a panel's local coordinate
- vector3d transform_vector_panel_inverse (int panel, const vector3d &x) const

  transforms a point from panel's local coordinate to global coordinate
- vector3d transform_vector_panel (int panel, const vector3d &x) const

  transforms a vector in a panel's local coordinate
- double get_panel_area (const int &panel) const

transforms a vector from panel's local coordinate to global coordinate

- vector3d compute_source_panel_unit_velocity (const int &panel, const vector3d &node) const

    computes the induced velocity due to source panel at a point
- vector3d compute_doublet_panel_unit_velocity (const int &panel, const vector3d &node) const

    computes the induced velocity due to doublet panel at a point

Public Attributes

- std::vector< vector3d > nodes

    stores node data containing x,y,z coordinate
- std::vector< std::vector< int > > panels

    panel vector containing vector of its nodes
- std::vector< std::vector< int > > panel_neighbours

    stores the neighbouring panel ids
- std::vector< int > trailing_edge_nodes

    nodes on trailing edge
- std::vector< int > upper_TE_panels

    panels connected to upper trailing edge
- std::vector< int > lower_TE_panels

    panels connected to lower trailing edge

3.6.1 Detailed Description

Surface class to represent geometry and perform various operations on it.

Class for containing geometry entities and functions to perform operations on them

3.6.2 Member Function Documentation

3.6.2.1 double Surface::compute_doublet_panel_influence (const int *panel,* const vector3d & *node*) const

computes the influence coefficient due to doublet panel

Parameters

in	*panel*	doublet panel whose influence is sought
in	*node*	point which is being influenced by **panel**
in	*double*	return influence due to unit doublet strength panel

3.6.2.2 vector3d Surface::compute_doublet_panel_unit_velocity (const int & *panel,* const vector3d & *node*) const

computes the induced velocity due to doublet panel at a point

Parameters

in	*panel*	doublet panel
in	*node*	point at which induced velocity by **panel** is sought
in	*vector3d*	return induced velocity vector

3.6.2.3 void Surface::compute_panel_components ( )

Computes the panel components.

computes panel collocation point, panel local coordinates, area, panel transformations and farfield factors

3.6.2.4 std::pair< double, double > Surface::compute_source_doublet_panel_influence (const int *panel*, const vector3d & *node*) const

computes the influence coefficient due to source and doublet panel

Parameters

in	*panel*	panel whose influence is sought
in	*node*	point which is being influenced by **panel**
in	*pair<double,double>*	return influence due to unit strength source and doublet panel

3.6.2.5 double Surface::compute_source_panel_influence (const int *panel*, const vector3d & *node*) const

computes the influence coefficient due to source panel

Parameters

in	*panel*	source panel whose influence is sought
in	*node*	point which is being influenced by **panel**
in	*double*	return in the influence due to unit source strength panel

3.6.2.6 vector3d Surface::compute_source_panel_unit_velocity (const int & *panel*, const vector3d & *node*) const

computes the induced velocity due to source panel at a point

Parameters

in	*panel*	source panel
in	*node*	point at which induced velocity by **panel** is sought
in	*vector3d*	return induced velocity vector

3.6.2.7 vector3d & Surface::get_collocation_point (int *panel*)

returns panel's collocation point by reference

Parameters

in	*panel*	panel number
out	*vector3d&*	panel collocation point by reference

3.6.2.8 vector3d Surface::get_collocation_point (int *panel*) const

returns panel's collocation point

Parameters

in	*panel*	panel number
out	*vector3d*	panel collocation point

3.6.2.9 vector3d Surface::get_kinematic_velocity (const vector3d & *x*) const

returns kinematic velocity at a point on surface

Parameters

in	*x*	input location
out	*vector3d*	kinematic velocity due to surface's motion

3.6.2.10 vector3d Surface::get_trailing_edge_bisector (const int *TE_node*) const

returns trailing edge bisecor

Parameters

in	*TE_node*	trailing edge node
out	*vector3d*	normalized vector which points in the bisector at **TE_node**

3.6.2.11 int Surface::n_nodes ( ) const

returns total number of nodes in the current surface

Parameters

out	*int*	total number of nodes in the surface

3.6.2.12 int Surface::n_panels ( ) const

returns number of panels in the current surface

Parameters

out	*int*	total number of panels in the surface

3.6.2.13 void Surface::rotate_surface (vector3d *dTheta,* bool *isRadian*)

rotates the surface by an angle

Parameters

in	*dTheta*	angle by which to rorate the surface
in	*isRadian*	true if **dTheta** is in radians, false otherwise

3.6.2.14 void Surface::set_angular_velocity (vector3d *vel,* bool *isRadian_sec*)

Set angular velocity of the body.

Parameters

in	*vel*	angular velocity
in	*isRadian_sec*	true if **vel** is in radians per seconds

3.6.2.15 void Surface::translate_surface (const vector3d & *dX*)

translates the surface by distance dX

Parameters

`in`	*dX*	distance vector

3.6.3 Member Data Documentation

3.6.3.1 std::vector<std::vector<int> > Surface::panel_neighbours

stores the neighbouring panel ids

stores neighbouring panel ids - required in surface velocity calculations

3.6.3.2 std::vector<std::vector<int> > Surface::panels

panel vector containing vector of its nodes

represents panel vector. Each panel contains node number associated with it the node numbers are in counter-clockwise order

The documentation for this class was generated from the following files:

- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/surface.hpp
- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/surface.cpp

3.7 vector3d Class Reference

vector3d class to represent an array of size 3

`#include <vector3d.h>`

Public Member Functions

- **vector3d** (double a, double b, double c)
- **vector3d** (const vector3d &vec)
- double **operator[ ]** (int i) const
- double & **operator[ ]** (int i)
- void **operator=** (const vector3d &vec)
- const vector3d & **operator=** (const vector3d &vec) const
- double **dot** (const vector3d &vec)
- vector3d **cross** (const vector3d &vec) const
- void **normalize** ()
- double **squared_norm** () const
- double **norm** () const
- int **size** () const
- vector3d **operator-** (const vector3d &vec) const
- vector3d **operator+** (const vector3d &vec) const
- void **print** ()
- vector3d **operator/** (const double &val) const
- vector3d **operator*** (const double &val) const
- void **operator=** (const double val)
- vector3d **operator-** () const
- **operator vector3d &** ()
- **operator vector3d const &** () const
- double * **begin** ()
- void **operator+=** (const vector3d &vec)

- void **operator-=** (const vector3d &vec)
- void **operator*=** (const double &x)
- vector3d **operator/** (const vector3d &vec) const

Friends

- std::ostream & **operator<<** (std::ostream &os, const vector3d &vec)

3.7.1 Detailed Description

vector3d class to represent an array of size 3

useful in representing 3d point coordinates, vectors, forces, etc. Also performs some basic operations, helps in reducing equations sizes

The documentation for this class was generated from the following file:

- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/vector3d.h

3.8 vtk_writer Class Reference

vtk_writer class to write output in vtk format

```
#include <vtk_writer.hpp>
```

Public Member Functions

- template<class T >
  void write_surface_data (std::string filename, const std::shared_ptr< Surface > surface, const std::vector< T > &data, std::string name, bool writemesh)

  writes surface data to a file
- void write_surface_mesh (std::string filename, std::shared_ptr< Surface >)

  write mesh data to vtk format
- void write_domain_mesh (std::string filename, std::shared_ptr< Domain >)

  write domain mesh to vtk format
- template<class T >
  void write_domain_data (std::string filename, std::shared_ptr< Domain > domain, std::vector< T > data, std::string name, bool writemesh)

  writes surface data to a file

3.8.1 Detailed Description

vtk_writer class to write output in vtk format

3.8.2 Member Function Documentation

3.8.2.1 template<class T > void vtk_writer::write_domain_data (std::string *filename,* std::shared_ptr< **Domain** > *domain,* std::vector< T > *data,* std::string *name,* bool *writemesh*) `[inline]`

writes surface data to a file

Parameters

in	*filename*	filename of the output file, extension will be attaced automatically
in	*domain*	domain to write
in	*data*	data to write for a given **domain**
in	*name*	name of the **data** variable
in	*writemesh*	true if mesh data needs to be written, else false

3.8.2.2 template<class T > void vtk_writer::write_surface_data (std::string *filename*, const std::shared_ptr< **Surface** > *surface*, const std::vector< T > & *data*, std::string *name*, bool *writemesh*) `[inline]`

writes surface data to a file

Parameters

in	*filename*	filename of the output file, extension will be attaced automatically
in	*surface*	surface to write
in	*data*	data to write for a given **surface**
in	*name*	name of the **data** variable
in	*writemesh*	true if mesh data needs to be written, else false

The documentation for this class was generated from the following files:

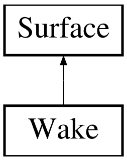
- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/vtk_writer.hpp
- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/vtk_writer.cpp

3.9 Wake Class Reference

Wake class to represent wake of a body.

`#include <wake.hpp>`

Inheritance diagram for Wake:



Public Member Functions

- void add_lifting_surface (const std::shared_ptr< Surface > surf)

  associate with surface object
- void initialize (const vector3d &free_stream_velocity, const double &dt)

  create first row of wake panels
- void build_topology ()

  build topology of the wake panels, such as connectivity information
- void shed_wake (const vector3d &free_stream_velocity, double dt)

  adds a row of wake panels

Additional Inherited Members

3.9.1 Detailed Description

Wake class to represent wake of a body.

The documentation for this class was generated from the following files:

- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/wake.hpp
- /home/pranav/Google Drive/Classes/wind-turbine/panel_method/code/3dVPM/3dVPM/src/wake.cpp