AFLR2 Utilities

0.9

Generated by Doxygen 1.8.6

Sat Mar 26 2016 01:41:53

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 aflr2utils Namespace Reference

Namespaces

- distributions
- geometry
- mesh

4.2 aflr2utils.distributions Namespace Reference

Classes

- class Uniform
- class HyperbolicTangent

4.3 aflr2utils.geometry Namespace Reference

Classes

- class Point
- class Segment

Functions

- def import_geometry

4.3.1 Function Documentation

4.3.1.1 def aflr2utils.geometry.import_geometry (*file_name*)

```
Function for importing segments from a file.

Format of data file:
NS
NP_1
x_1  y_1
x_2  y_2
```

```
  .   .
  .   .
  .   .
x_NP y_NP
NC_2
NP_1
...

where
NS   - number of segments.
NP_i - number of knots for segment i.

@param file_name name of file which contains data to be imported.
@return a list of segment objects.
```

4.4 aflr2utils.mesh Namespace Reference

Classes

- class Connector
- class Edge
- class Mesh

Functions

- def create_circular_connector
- def create_circular_arc_connector

4.4.1 Function Documentation

4.4.1.1 def aflr2utils.mesh.create_circular_arc_connector (*xc*, *yc*, *r*, *theta_start*, *theta_stop*, *num_points*)

```
Function for creating a circular connector.
This function creates a circular connector when provided with
the x- and y-coordinates of the center of the circle and the
radius of the circle.  A uniform point distribution is created on
the circle.

@param xc x-coordinate of the center of the circle.
@param yc y-coordinate of the center of the circle.
@param r radius of the circle.
@param theta_start starting angle in radians.
@param theta_stop final angle in radians.
@param num_points number of points on the circle.
```

4.4.1.2 def aflr2utils.mesh.create_circular_connector (*xc*, *yc*, *r*, *num_points*)

```
Function for creating a circular connector.
This function creates a circular connector when provided with
the x- and y-coordinates of the center of the circle and the
radius of the circle.  A uniform point distribution is created on
the circle.

@param xc x-coordinate of the center of the circle.
@param yc y-coordinate of the center of the circle.
@param r radius of the circle.
@param num_points number of points on the circle.
```

Chapter 5

Class Documentation

5.1 aflr2utils.mesh.Connector Class Reference

Public Member Functions

- def __init__
- def create_point_distribution

Public Attributes

- seg
- num_points
- nodes
- bc

5.1.1 Detailed Description

```
Class for connector (in Pointwise terminology).
```

5.1.2 Constructor & Destructor Documentation

5.1.2.1 def aflr2utils.mesh.Connector.__init__ (*self*, *seg*, *num_points*, *bc* = 0)

```
Constructor.
Possible boundary conditions (bc) values are
0 - inviscid
1 - viscous
2 - farfield

@param seg input segment object on which connector will be created.
```

5.1.3 Member Function Documentation

5.1.3.1 def aflr2utils.mesh.Connector.create_point_distribution (*self*, *distribution*, *cluster_end* = False)

```
Method for creating points on a connector.
This method assumes that you want to cluster towards the start of the connector.
If this is not the case, set cluster_end to True.

@param distribution a point distribution object from distributions.py.
@param cluster_end boolean variable used to control clustering direction.
```

5.1.4 Member Data Documentation

5.1.4.1 aflr2utils.mesh.Connector.bc

5.1.4.2 aflr2utils.mesh.Connector.nodes

5.1.4.3 aflr2utils.mesh.Connector.num_points

5.1.4.4 aflr2utils.mesh.Connector.seg

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

- mesh.py

5.2 aflr2utils.mesh.Edge Class Reference

Public Member Functions

- def __init__

Public Attributes

- connectors

5.2.1 Detailed Description

```
Edge class which is a collection of connectors
```

5.2.2 Constructor & Destructor Documentation

5.2.2.1 def aflr2utils.mesh.Edge.__init__ (*self*, *connectors*)

```
Constructor.

@param connectors list of connectors which belong to this edge.
```

5.2.3 Member Data Documentation

5.2.3.1 aflr2utils.mesh.Edge.connectors

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

- mesh.py

5.3 aflr2utils.distributions.HyperbolicTangent Class Reference

Public Member Functions

- def __init__
- def find_delta
- def s

Public Attributes

- delta
- imax
- ds
- reverse
- s_fun

5.3.1 Detailed Description

```
Class which represents a hyperbolic tangent point distribution.
```

5.3.2 Constructor & Destructor Documentation

5.3.2.1 def aflr2utils.distributions.HyperbolicTangent.__init__ (*self*, *imax*, *ds*, *lx* = `1.0`, *reverse* = `False`)

```
Constructor.

@param imax number of points.
@param ds clustering toward beginning.
@param lx total x-length of segment.
@param reverse boolean variable used to reverse direction of clustering.
```

5.3.3 Member Function Documentation

5.3.3.1 def aflr2utils.distributions.HyperbolicTangent.find_delta (*self*)

```
Method for finding delta.
This method finds the value of delta required to match with the given ds.
```

5.3.3.2 def aflr2utils.distributions.HyperbolicTangent.s (*self*, *i*)

```
Method for getting the intermediate spacing function.

@param i coordinate for the i-th point.
@return s(i).
```

5.3.4 Member Data Documentation

5.3.4.1 aflr2utils.distributions.HyperbolicTangent.delta

5.3.4.2 aflr2utils.distributions.HyperbolicTangent.ds

5.3.4.3 aflr2utils.distributions.HyperbolicTangent.imax

5.3.4.4 aflr2utils.distributions.HyperbolicTangent.reverse

5.3.4.5 aflr2utils.distributions.HyperbolicTangent.s_fun

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

- distributions.py

5.4 aflr2utils.mesh.Mesh Class Reference

Public Member Functions

- def __init__
- def write_bedge

Public Attributes

- edges

5.4.1 Detailed Description

```
Class which represents a mesh.
A mesh is a collection of edges which represent the boundaries.
```

5.4.2 Constructor & Destructor Documentation

5.4.2.1 def aflr2utils.mesh.Mesh.__init__ (*self*, *edges*)

```
Constructor.

@param edges list of edges which represent the boundaries in the mesh.
```

5.4.3 Member Function Documentation

5.4.3.1 def aflr2utils.mesh.Mesh.write_bedge (*self*, *bedge_name*)

```
Method for writing boundaries (edges) to a bedge file.
This method writes a bedge file which is input for
AFLR2.

@param bedge_name string which contains the name of the output file.
```

5.4.4 Member Data Documentation

5.4.4.1 aflr2utils.mesh.Mesh.edges

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

- mesh.py

5.5 aflr2utils.geometry.Point Class Reference

Public Member Functions

- def __init__

Public Attributes

- x
- y

5.5.1 Detailed Description

```
Class which represents a point in 2D.
```

5.5.2 Constructor & Destructor Documentation

5.5.2.1 def aflr2utils.geometry.Point.__init__ (*self, xc, yc*)

```
Constructor.

@param xc x-coordinate of point.
@param yc y-coordinate of point.
```

5.5.3 Member Data Documentation

5.5.3.1 aflr2utils.geometry.Point.x

5.5.3.2 aflr2utils.geometry.Point.y

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

- geometry.py

5.6 aflr2utils.geometry.Segment Class Reference

Public Member Functions

- def __init__
- def find_inner_control_points
- def obj_func
- def get_point

Public Attributes

- points
- is_bezier
- spacing_function
- bezier_fun_x
- bezier_fun_y
- straight_fun_x
- straight_fun_y
- p0
- p1
- p2
- p3
- x0
- x_data
- y_data

5.6.1 Detailed Description

```
Class which represents a line segment.

This class represents a line segment via control points.  If only two
control points are used, the segment is assumed to be straight.  Otherwise,
the other supported option is for a line with four control points.  In this
case, the segment is represented using a cubic Bezier curve.
```

5.6.2 Constructor & Destructor Documentation

5.6.2.1 def aflr2utils.geometry.Segment.__init__ (*self*, *points*, *spacing_function* = `None`, *is_bezier* = `True`)

```
This constructor must be called with a list which contains points.

@param points list of Point objects.
```

5.6.3 Member Function Documentation

5.6.3.1 def aflr2utils.geometry.Segment.find_inner_control_points (*self*)

```
Utility function for finding the inner control points.
```

5.6.3.2 def aflr2utils.geometry.Segment.get_point (*self*, *i*)

```
Method for getting the i-th point on a line.

@param i coordinate in the computational plane (i.e., the i-th point)
@return a Point object which has the x- and y-coordinates of the i-th point.
```

5.6.3.3 def aflr2utils.geometry.Segment.obj_func (*self*, *y*)

```
Objective function used in fitting the curve to the data points.

@param y numpy array with two entries which represent y-coordinates of control pts.
@return Sum of squares error between curve and data.
```

5.6.4 Member Data Documentation

5.6.4.1 aflr2utils.geometry.Segment.bezier_fun_x

5.6.4.2 aflr2utils.geometry.Segment.bezier_fun_y

5.6.4.3 aflr2utils.geometry.Segment.is_bezier

5.6.4.4 aflr2utils.geometry.Segment.p0

5.6.4.5 aflr2utils.geometry.Segment.p1

5.6.4.6 aflr2utils.geometry.Segment.p2

5.6.4.7 aflr2utils.geometry.Segment.p3

5.6.4.8 aflr2utils.geometry.Segment.points

5.6.4.9 aflr2utils.geometry.Segment.spacing_function

5.6.4.10 aflr2utils.geometry.Segment.straight_fun_x

5.6.4.11 aflr2utils.geometry.Segment.straight_fun_y

5.6.4.12 aflr2utils.geometry.Segment.x0

5.6.4.13 aflr2utils.geometry.Segment.x_data

5.6.4.14 aflr2utils.geometry.Segment.y_data

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

- geometry.py

5.7 aflr2utils.distributions.Uniform Class Reference

Public Member Functions

- def __init__
- def s

Public Attributes

- imax

5.7.1 Detailed Description

```
Class which represents a uniform point distribution.
```

5.7.2 Constructor & Destructor Documentation

5.7.2.1 def aflr2utils.distributions.Uniform.__init__ (*self*, *imax*)

```
Constructor.

@param imax number of points.
```

5.7.3 Member Function Documentation

5.7.3.1 def aflr2utils.distributions.Uniform.s (*self*, *i*)

```
Method for getting the intermediate spacing function.

@param i coordinate for the i-th point.
@return s(i).
```

5.7.4 Member Data Documentation

5.7.4.1 aflr2utils.distributions.Uniform.imax

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

- distributions.py

Chapter 6

File Documentation

6.1 __init__.py File Reference

Namespaces

- aflr2utils

6.2 distributions.py File Reference

Classes

- class aflr2utils.distributions.Uniform
- class aflr2utils.distributions.HyperbolicTangent

Namespaces

- aflr2utils.distributions

6.3 geometry.py File Reference

Classes

- class aflr2utils.geometry.Point
- class aflr2utils.geometry.Segment

Namespaces

- aflr2utils.geometry

Functions

- def aflr2utils.geometry.import_geometry

6.4 mesh.py File Reference

Classes

- class aflr2utils.mesh.Connector
- class aflr2utils.mesh.Edge
- class aflr2utils.mesh.Mesh

Namespaces

- aflr2utils.mesh

Functions

- def aflr2utils.mesh.create_circular_connector
- def aflr2utils.mesh.create_circular_arc_connector

Index