pipemesh Documentation

Release 0.1.2

Duncan Hunter

Oct 18, 2019

Contents:

pipemesh is a python package allowing modular, sequential creation of pipes and pipe networks using GMSH. The most useful tool here is the Network class in pipes. A readme with installation and examples can be found on github.

Contents:

Pipes

Create pipes and pipe networks using the Network class.

See the Readme for more details and examples.

Author: Duncan Hunter

class `pipemesh.pipes.`**`Network`**(*length*, *radius*, *direction*, *lcar=0.1*)
> Represents a pipe or network of pipes.
>
> Pipes are built from an inlet in a sequential, modular fashion. When a junction is added, a new "out surface" is added, which can be added to. In this way, a network of pipes can be built.
>
> **`physical_in_out_surfaces`**
> > Dictionary of Physical surface tags to GMSH surface tags for inlets/outlets.
>
> **`phyiscal_no_slip`**
> > Dictionary of Physical surface tags for walls/outside of cylinder.
>
> **`physical_volume`**
> > Physical tag of the volume. Only available after generate.
>
> **`add_cylinder`**()
> > Add a cylinder to the Network.
>
> **`add_curve`**()
> > Add a curve to the Network.
>
> **`add_mitered`**()
> > Add a mitered bend to the Network.
>
> **`add_change_radius`**()
> > Add a cylinder with change in radius to the Network.
>
> **`add_t_junction`**()
> > Add a T junction the the network.
>
> **`generate`**()
> > Fuse the network and generates the msh file.

get_inlet_outlet_phys_ids()
  Returns a list of physical ids of inlets.

get_cyl_phys_ids()
  Returns a list of physical ids of cylinder surfaces.

get_velocities_reynolds()
  Returns velocity vectors for inlets using Reynolds number.

get_velocities_vel_mag()
  Returns velocity vectors for inlets using velocity magnitude.

add_change_radius(*length*, *new_radius*, *change_length*, *lcar=0.1*, *out_number=0*)
  Adds a piece that changes the radius of the outlet.

  The piece is length long, and changes the Network radius to new_radius, over change_length, which controls how gentle the change is.

  > **Parameters**
  >
  > - **length** – (float) Length of the piece.
  >
  > - **new_radius** – (float) radius to change to.
  >
  > - **change_length** – (float) Length that the change takes place over. Must be less than length and > 0.
  >
  > - **lcar** – (float) mesh size for this piece.
  >
  > - **out_number** – Out surface to add to. If <= 1, will add to the first out surface.
  >
  > **Raises** ValueErrors – change_length is not between length and 0. If radius does not change.

add_curve(*new_direction*, *bend_radius*, *lcar=0.1*, *out_number=0*)
  Adds a curve to the Network at the outlet.

  > **Parameters**
  >
  > - **new_direction** – (list) Direction pipe will be facing in x, y, z vector format. e.g. [0, 1, 0] faces positive y.
  >
  > - **bend_radius** – (float) Radius of the bend.
  >
  > - **lcar** – (float) Size of mesh in this piece.
  >
  > - **out_number** – Out surface to add to. If <= 1, will add to the first out surface.
  >
  > **Raises** ValueError – new_direction vector isn't right size. Bend radius isn't big enough (<1.1 inlet radius).

add_cylinder(*length*, *lcar=0.1*, *out_number=0*)
  Adds a pipe to the Network at the outlet.

  > **Parameters**
  >
  > - **length** – (float) length of pipe.
  >
  > - **lcar** – (float) mesh size of piece.
  >
  > - **out_number** – Out surface to add to. If <= 1, will add to the first out surface.

add_mitered(*new_direction*, *lcar=0.1*, *out_number=0*)
  Adds a mitered bend to the Network at the outlet.

  A mitered bend is a sharp change in direction. Hard to simulate.

  > **Parameters**

- **new_direction** – (list, length 3) xyz vector representing the new direction of the pipe.

- **lcar** – (float) size of mesh of this piece.

- **out_number** – Out surface to add to. If <= 1, will add to the first out surface.

add_t_junction(*t_direction*, *lcar=0.1*, *t_radius=-1*, *out_number=0*)
Adds a T junction to the Network at the outlet.

This represents a pipe joining this pipe, creating a place to add a Network to this Network.

> **Parameters**
>
> - **t_direction** – (list, length 3) representing the direction that the joining pipe's inlet is facing.
>
> - **lcar** – (float) mesh size for this piece.
>
> - **t_radius** – radius of the piece joining the pipe. If <= 0, will default to radius of the pipe.
>
> - **out_number** – Out surface to add to. If <= 1, will add to the first out surface.

generate(*filename=None*, *binary=False*, *mesh_format='msh2'*, *write_info=False*, *write_xml=False*, *run_gui=False*)
Generates mesh and saves if filename.

> **Parameters**
>
> - **filename** – (string) filename (without extension) to save as.
>
> - **binary** – (Bool) Save mesh as binary or not. Default (False).
>
> - **mesh_format** – (string) mesh format to save as. Default is msh2. To save as msh4, use 'msh4'.
>
> - **write_info** – (Bool) write filename.txt with mesh mesh information (physical surfaces, locations and directions).
>
> - **write_xml** – (Bool) write information in an xml file. Still under development.
>
> - **run_gui** – (Bool) run the gmsh gui to view the mesh. May stop saving of information/meshes.

get_cyl_phys_ids()
Returns a list of physical ids of cylinder surfaces.

get_inlet_outlet_phys_ids()
Returns a list of physical ids of inlets.

By default, the inlet phys_id is 1, then the default outlet is 2, followed by any added outlets in the order they were added.

get_velocities_reynolds(*physical_ids*, *reynolds_no*, *density*, *viscosity*)
Creates velocity vectors for inlets using reynolds number.

Must be run after generate(). Physical ids are turned into indices, which are used to select surfaces from self.physical_in_out_surfaces. By default, the inlet phys_id is 1, then the default outlet is 2, followed by any added outlets in the order they were added. Reynolds is turned into velocity mag using Reynolds*visc/(2*radius*density).

> **Parameters**
>
> - **physical_ids** – (list of ints) physical ids of inlets/outlet surfaces to get velocity vectors for.
>
> - **reynolds_no** – (float) reynolds number resulting from velocity vectors created.

- **density** – (float) density of the fluid simulated. Should match density used in simulation for accurate results.

- **viscosity** – (float) viscosity of the fluid simulated. Should match viscosity used in simulation for accurate results.

get_velocities_vel_mag(*physical_ids*, *velocity_magnitude*)

Returns velocity vectors for inlets using velocity magnitude.

Must be run after generate(). Physical ids are turned into indices, which are used to select surfaces from self.physical_in_out_surfaces. By default, the inlet phys_id is 1, then the default outlet is 2, followed by any added outlets in the order they were added.

rotate_network(*axis*, *angle*)

Rotates the network from old_direction to new_direction.

> **Parameters**
>
> - **axis** – (array-like, shape (3,)) xyz vector representing the axis of rotation.
>
> - **angle** – angle to rotate network about axis.

translate_network(*vector*)

Translates a network by vector.

> **Parameters** **vector** – (list length 3) representing xyz vector to translate network by.

Pieces

Base classes for cylindrical GMSH pieces.

Also contains useful functions for these classes. Author: Duncan Hunter

class pipemesh.pieces.**ChangeRadius**(*length*, *change_length*, *in_radius*, *out_radius*, *direction*, *lcar*)
    Class representing a cylinder with a change in radius.

class pipemesh.pieces.**Curve**(*radius*, *in_direction*, *out_direction*, *bend_radius*, *lcar*)
    Class representing a GMSH curve by revolution.

class pipemesh.pieces.**Cylinder**(*length*, *radius*, *direction*, *lcar*)
    Class representing a GMSH cylinder with base at 0,0,0 facing upwards.

class pipemesh.pieces.**Mitered**(*radius*, *in_direction*, *out_direction*, *lcar*)
    Class representing a mitered (sharp) pipe bend.

    Piece creation is done by masking (intersect) a cylinder with a chamfered box. The piece is then mirrored, rotated and fused.

    The piece is then rotated to face the direction of the outflow. It is then rotated about the direction of outflow to match the new direction

class pipemesh.pieces.**PipePiece**(*radius*, *vol_tag*, *in_tag*, *out_tag*, *in_direction*, *out_direction*, *lcar*)
    Parent class of pieces.

    Pieces are GMSH objects that can be used in creating pipes. This class has common information that all pieces have, such as radius. It also has functions that all the classes use, such as the need to update centres of pieces after they have been transformed.

class pipemesh.pieces.**Surface**(*dimtag*, *centre*, *direction*, *radius*)
    Class representing a surface of a piece.

    Pieces are PipePieces, and are all cylindrical in nature, which is why the pieces have radius.

class pipemesh.pieces.**TJunction**(*radius*, *t_radius*, *direction*, *t_direction*, *lcar*)
    Class representing a T-junction in GMSH

`pipemesh.pieces.`**`proj`**`(`*vec1*`, `*vec2*`)`
> Returns the component of vec1 along vec2

> > **Parameters** **`vec2`** (`vec1,`) – (np.array shape 3) xyz vector.

`pipemesh.pieces.`**`vec_angle`**`(`*vec1*`, `*vec2*`)`
> Returns the angle between two numpy array vectors

AutoMPML

Edits a basic .mpml file with the AutoMPML class.

Create the class with options = AutoMPML. Then call the set_methods to set options. Then use write to generate the file. MPML files generated will only work with latest version of IC-FERST. Contact author if you want support for older versions.

class pipemesh.icferst.auto_mpml.**AutoMPML**

> **set_all** (*sim_name=None*, *msh_file='src/pipe'*, *dump_period=0.1*, *dump_ids=[]*, *finish_time=1*, *timestep=0.005*, *cfl_no=2.0*, *density=0.001*, *viscosity=0.001*, *inlet_phys_ids=[]*, *inlet_velocities=[]*, *outlet_phys_ids=[]*, *cyl_phys_ids=[]*, *min_mesh_size=0.01*, *max_mesh_size=0.5*, *max_no_nodes=300000.0*, *t_adapt_delay=0.5*, *aspect_ratio=0.5*)
> Sets all of the options at once.
>
> See base functions for information on what each option sets.

> **set_inlets** (*phys_ids*, *velocities*)
> Sets the properties for inlets.
>
> > **Parameters**
> >
> > - **phys_ids** – (list of ints) physical ids of inlet surfaces.
> >
> > - **velocities** – (list of xyz vector lists) velocities for respective physical ids. E.g. velocities[0] corresponds to phys_ids[0]. If not enough vectors are given, the first vector is used for all inlets.

> **set_io_options** (*dump_period=0.1*, *dump_ids=[]*)
> Sets the input/output settings.
>
> > **Parameters**
> >
> > - **dump_period** – How often to dump vtu files.
> >
> > - **dump_ids** – Physical surfaces to record fluxes.

> **set_material_properties** (*density=1000.0*, *viscosity=0.001*)
> Sets the material phase properties.

> **Parameters**
>
> - **density** – density of the fluid.
>
> - **viscosity** – viscosity of the fluid.

set_mesh_adaptivity(*min_size=0.01*, *max_size=0.5*, *max_no_nodes=300000*, *t_adapt_delay=0.5*, *aspect_ratio=5*)

Sets mesh adaptivity options.

> **Parameters**
>
> - **min_size** – Minimum mesh size.
>
> - **max_size** – Maximum mesh size.
>
> - **max_no_nodes** – Maximum number of nodes. 100,000 is ~16GB of memory.
>
> - **t_adapt_delay** – Start adaptive meshing at this time.
>
> - **aspect_ratio** – Limit the mesh shape to maximum this aspect ratio.

set_msh_options(*msh_file*)

Sets the location of the msh_file.

Default is src/pipe. :param msh_file: (string) location .msh file is stored.

set_no_slip(*phys_ids*)

Sets the no-slip boundary condition.

> **Parameters** **phys_ids** – The physical surfaces to set to no-slip.

set_outlets(*phys_ids*)

Sets the outlet surfaces.

Sets the physical surfaces phys_ids to 0 pressure.

> **Parameters** **phys_ids** – The physical surfaces to set to 0 pressure.

set_sim_name(*simname*)

Sets the name of the simulation.

Default is "3d_pipe".

set_timestepping(*finish_time*, *timestep=0.005*, *cfl_no=2.0*)

Sets the timestepping options.

> **Parameters**
>
> - **finish_time** – Time the simulations stops.
>
> - **timestep** – Initial timestep.
>
> - **cfl_no** – CFL number adaptive timestepping aims for.

write_mpml(*fname*)

Writes the settings to fname.mpml.

Indices and tables

- genindex
- modindex
- search
- pipemesh on github

Python Module Index

p

Index

T

TJunction (*class in pipemesh.pieces*), 7

translate_network() (*pipemesh.pipes.Network method*), 6

V

vec_angle() (*in module pipemesh.pieces*), 8

W

write_mpml() (*pipemesh.icferst.auto_mpml.AutoMPML method*), 10