geomdl-shapes Documentation

Onur Rauf Bingol

Jul 22, 2019

Contents:

geomdl-shapes is an extension module for creating common shapes using NURBS-Python (geomdl).

CHAPTER 1

Installing geomdl-shapes

The recommended method for installation is using pip.

```
$ pip install --user geomdl.shapes
```

Alternatively, you can install the latest version from the GitHub repository:

- Clone the repository: `git clone https://github.com/orbingol/geomdl-shapes.git`
- Inside the directory containing the cloned repository, run: `pip install --user .`
- The setup script will install all required dependencies

CHAPTER 2

API Documentation

2.1 Class Reference

2.1.1 Analytic Geometry

class geomdl.shapes.analytic.**Circle**(***kwargs*)

Bases: geomdl.shapes.analytic.AnalyticGeometry

Analytic circle geometry

Finds the points on a circle using the following equation:

$$x = x_0 + r \cos \theta$$
$$y = y_0 + r \sin \theta$$

Keyword Arguments:

- `radius`: radius of the circle. *Default: 1*
- `origin`: coordinates of the circle center. *Default: (0, 0)*

data

Returns a dict which contains the geometry data.

Please refer to the wiki for details on using this class member.

dimension

Spatial dimension.

Please refer to the wiki for details on using this class member.

Getter Gets the spatial dimension, e.g. 2D, 3D, etc.

Type int

evalpts

Evaluated points.

Please refer to the wiki for details on using this class member.

Getter Gets the coordinates of the evaluated points

Type list

evaluate(***kwargs*)
Evaluates the circle.

Keyword Arguments:

- `start`: start angle θ in degrees. *Default: 0*
- `stop`: stop angle θ in degrees. *Default: 360*
- `jump`: angle θ increment in degrees. *Default: 0.5*

id
Object ID (as an integer).

Please refer to the wiki for details on using this class member.

Getter Gets the object ID

Setter Sets the object ID

Type int

name
Object name (as a string)

Please refer to the wiki for details on using this class member.

Getter Gets the object name

Setter Sets the object name

Type str

opt
Dictionary for storing custom data in the current geometry object.

`opt` is a wrapper to a dict in *key* => *value* format, where *key* is string, *value* is any Python object. You can use `opt` property to store custom data inside the geometry object. For instance:

```
geom.opt = ["face_id", 4]  # creates "face_id" key and sets its value to an␣
↪integer
geom.opt = ["contents", "data values"]  # creates "face_id" key and sets its␣
↪value to a string
print(geom.opt)  # will print: {'face_id': 4, 'contents': 'data values'}

del geom.opt  # deletes the contents of the hash map
print(geom.opt)  # will print: {}

geom.opt = ["body_id", 1]  # creates "body_id" key  and sets its value to 1
geom.opt = ["body_id", 12]  # changes the value of "body_id" to 12
print(geom.opt)  # will print: {'body_id': 12}

geom.opt = ["body_id", None]  # deletes "body_id"
print(geom.opt)  # will print: {}
```

Please refer to the wiki for details on using this class member.

Getter Gets the dict

Setter Adds key and value pair to the dict

Deleter Deletes the contents of the dict

opt_get(*value*)
Safely query for the value from the *opt* property.

Parameters **value** (*str*) – a key in the *opt* property

Returns the corresponding value, if the key exists. None, otherwise.

reverse()
Reverses the evaluated points

type
Geometry type

Please refer to the wiki for details on using this class member.

Getter Gets the geometry type

Type str

class geomdl.shapes.analytic.**Sphere**(***kwargs*)
Bases: geomdl.shapes.analytic.AnalyticGeometry

Analytic sphere geometry

Finds the points on a sphere using the following equation:

$$x = x_0 + r \sin \phi \cos \theta$$
$$y = y_0 + r \sin \phi \sin \theta$$
$$z = z_0 + r \cos \phi$$

Keyword Arguments:

- radius: radius of the sphere. *Default: 1*
- origin: coordinates of the sphere center. *Default: (0, 0, 0)*

data
Returns a dict which contains the geometry data.

Please refer to the wiki for details on using this class member.

dimension
Spatial dimension.

Please refer to the wiki for details on using this class member.

Getter Gets the spatial dimension, e.g. 2D, 3D, etc.

Type int

evalpts
Evaluated points.

Please refer to the wiki for details on using this class member.

Getter Gets the coordinates of the evaluated points

Type list

evaluate(***kwargs*)
Evaluates the sphere.

Keyword Arguments:

- start_theta: start angle θ in degrees. *Default: 0*
- stop_theta: stop angle θ in degrees. *Default: 360*

- `jump_theta`: angle θ increment in degrees. *Default: 0.5*
- `start_phi`: start angle ϕ in degrees. *Default: 0*
- `stop_phi`: stop angle ϕ in degrees. *Default: 180*
- `jump_phi`: angle ϕ increment in degrees. *Default: 0.25*

id

Object ID (as an integer).

Please refer to the wiki for details on using this class member.

Getter Gets the object ID

Setter Sets the object ID

Type int

name

Object name (as a string)

Please refer to the wiki for details on using this class member.

Getter Gets the object name

Setter Sets the object name

Type str

opt

Dictionary for storing custom data in the current geometry object.

`opt` is a wrapper to a dict in *key* => *value* format, where *key* is string, *value* is any Python object. You can use `opt` property to store custom data inside the geometry object. For instance:

```
geom.opt = ["face_id", 4]  # creates "face_id" key and sets its value to an␣
↪integer
geom.opt = ["contents", "data values"]  # creates "face_id" key and sets its␣
↪value to a string
print(geom.opt)  # will print: {'face_id': 4, 'contents': 'data values'}

del geom.opt  # deletes the contents of the hash map
print(geom.opt)  # will print: {}

geom.opt = ["body_id", 1]  # creates "body_id" key  and sets its value to 1
geom.opt = ["body_id", 12]  # changes the value of "body_id" to 12
print(geom.opt)  # will print: {'body_id': 12}

geom.opt = ["body_id", None]  # deletes "body_id"
print(geom.opt)  # will print: {}
```

Please refer to the wiki for details on using this class member.

Getter Gets the dict

Setter Adds key and value pair to the dict

Deleter Deletes the contents of the dict

opt_get(*value*)

Safely query for the value from the *opt* property.

Parameters **value** (*str*) – a key in the *opt* property

Returns the corresponding value, if the key exists. `None`, otherwise.

type
Geometry type

Please refer to the wiki for details on using this class member.

Getter Gets the geometry type

Type str

class geomdl.shapes.analytic.**Rectangle**(***kwargs*)
Bases: geomdl.shapes.analytic.AnalyticGeometry

Analytic rectangle geometry

Finds the points on a rectangle with the size of $2p \times 2q$ using the following equation:

$$x = a(|\cos\theta| \cos\theta + |\sin\theta| \sin\theta)$$
$$y = b(|\cos\theta| \cos\theta - |\sin\theta| \sin\theta)$$

Keyword Arguments:

- `a`: length of the side on the u-direction. *Default: 1*
- `b`: length of the side on the v-direction. *Default: 1*
- `origin`: coordinates of the rectangle center. *Default: (0, 0)*

data
Returns a dict which contains the geometry data.

Please refer to the wiki for details on using this class member.

dimension
Spatial dimension.

Please refer to the wiki for details on using this class member.

Getter Gets the spatial dimension, e.g. 2D, 3D, etc.

Type int

evalpts
Evaluated points.

Please refer to the wiki for details on using this class member.

Getter Gets the coordinates of the evaluated points

Type list

evaluate(***kwargs*)
Evaluates the rectangle.

Keyword Arguments:

- `start`: start angle θ in degrees. *Default: 0*
- `stop`: stop angle θ in degrees. *Default: 360*
- `jump`: angle θ increment in degrees. *Default: 0.5*

id
Object ID (as an integer).

Please refer to the wiki for details on using this class member.

Getter Gets the object ID

Setter Sets the object ID

Type int

name

Object name (as a string)

Please refer to the wiki for details on using this class member.

Getter Gets the object name

Setter Sets the object name

Type str

opt

Dictionary for storing custom data in the current geometry object.

`opt` is a wrapper to a dict in *key* => *value* format, where *key* is string, *value* is any Python object. You can use `opt` property to store custom data inside the geometry object. For instance:

```
geom.opt = ["face_id", 4]  # creates "face_id" key and sets its value to an␣
↪integer
geom.opt = ["contents", "data values"]  # creates "face_id" key and sets its␣
↪value to a string
print(geom.opt)  # will print: {'face_id': 4, 'contents': 'data values'}

del geom.opt  # deletes the contents of the hash map
print(geom.opt)  # will print: {}

geom.opt = ["body_id", 1]  # creates "body_id" key  and sets its value to 1
geom.opt = ["body_id", 12]  # changes the value of "body_id" to 12
print(geom.opt)  # will print: {'body_id': 12}

geom.opt = ["body_id", None]  # deletes "body_id"
print(geom.opt)  # will print: {}
```

Please refer to the wiki for details on using this class member.

Getter Gets the dict

Setter Adds key and value pair to the dict

Deleter Deletes the contents of the dict

opt_get(*value*)

Safely query for the value from the *opt* property.

Parameters **value** (*str*) – a key in the *opt* property

Returns the corresponding value, if the key exists. `None`, otherwise.

reverse()

Reverses the evaluated points

type

Geometry type

Please refer to the wiki for details on using this class member.

Getter Gets the geometry type

Type str

2.2 Function Reference

2.2.1 Curves

geomdl.shapes.curve2d.**full_circle**(*radius=1*)
Generates a full NURBS circle from 9 control points.

Parameters **radius** (*int, float*) – radius of the circle

Returns a NURBS curve

Return type NURBS.Curve

geomdl.shapes.curve2d.**full_circle2**(*radius=1*)
Generates a full NURBS circle from 7 control points.

Parameters **radius** (*int, float*) – radius of the circle

Returns a NURBS curve

Return type NURBS.Curve

2.2.2 Surfaces

geomdl.shapes.surface.**cylinder**(*radius=1*, *height=1*)
Generates a cylindrical NURBS surface.

Parameters

- **radius** (*int, float*) – radius of the cylinder
- **height** (*int, float*) – height of the cylinder

Returns a NURBS surface

Return type NURBS.Surface

2.2.3 Volumes

geomdl.shapes.volume.**scordelis_lo**(*radius=25*, *thickness=0.25*, *length=50*, *angle=40*, ***kwargs*)
Generates a Scordelis-Lo Roof.

The Scordelis-Lo roof is a classical test case for linear static analysis. Please refer to the following articles for details:

- https://doi.org/10.14359/7796
- https://doi.org/10.1016/0045-7825(85)90035-0
- https://doi.org/10.1016/j.cma.2010.03.029

Keyword Arguments:

- `jump_angle`: iteration step for *angle* value. *Default: 2*
- `jump_length`: iteration step for *length* value. *Default: 2*
- `degree_u`: degree of the volume (u-dir). *Default: 2*
- `degree_v`: degree of the volume (v-dir). *Default: 2*

- `size_u`: number of control points (u-dir). *Default: degree_u + 2*
- `size_v`: number of control points (v-dir). *Default: degree_v + 2*

Parameters

- **radius** (*int, float*) – radius (R)
- **thickness** (*int, float*) – thickness (t)
- **length** (*int, float*) – length (L)
- **angle** (*int, float*) – angle in degrees (Theta)

Returns Scordelis-Lo Roof as a shell/volume

Return type BSpline.Volume

Python Module Index

Index

V