
polyhedron*potential Documentation*

Release 0.0.1

Shankar Kulumani

Dec 20, 2018

INSTALLATION

1	Meshes	1
2	Indices and tables	7

This module provides a MeshData class for defining a Surface_Mesh and performing related operations.

class MeshData

Class to hold a polyhedron mesh.

This is a wrapper of the Surface_mesh from CGAL and is the foundation used for the asteroid polyhedron potential model.

Author Shankar Kulumani

Version 28 October 2018

Public Functions

void **update_mesh** (**const** Eigen::Ref<**const** Eigen::MatrixXd> &V, **const** Eigen::Ref<**const** Eigen::MatrixXi> &F)

Completely update the mesh with new vertices and faces. This will also completely regenerate the mesh properties

Return void

Author Shankar Kulumani

Version 9 June 2018

Parameters

- V: Eigen matrix with the vertices
- F: Eigen matrix with the faces

build_edge_factor (**const** Eigen::Ref<**const** Eigen::Vector3d> &pos)

Build the per edge factor L for the polyhedron potential model

Return bool Success if true

Author Shankar Kulumani

Version 28 October 2018

Parameters

- `pos`: Position of test point

build_face_factor (`const Eigen::Ref<const Eigen::Vector3d> &pos`)
Build the per face factor `w` for the polyhedron potential model

Return bool success if true

Author Shankar Kulumani

Version 28 October 2018

Parameters

- `pos`: Position of test point

get_verts (void) **const**
Return the vertices of the mesh as an Eigen array

Return vertices Eigen arrays for the mesh `nx3`

Author Shankar Kulumani

Version 28 October 2018

`Eigen::Matrix<int, Eigen::Dynamic, 3>` **get_faces** (void) **const**
Return the faces of the mesh as an eigen array

Return faces Eigen array for the faces of the mesh `nx3`

Author Shankar Kulumani

Version 28 October 2018

`std::size_t` **number_of_vertices** (void) **const**
Return the number of vertices of the mesh

Return `num_vert` Return the number of vertices

Author Shankar Kulumani

Version 28 October 2018

`std::size_t` **number_of_edges** (void) **const**
Return number of edges of the mesh

Return `num_edge` Number of edges

Author Shankar Kulumani

Version 28 October 2018

`std::size_t` **number_of_faces** (void) **const**
Return number of faces of the mesh

Return num_face Number of faces

Author Shankar Kulumani

Version 28 October 2018

std::size_t **number_of_halfedges** (void) **const**
Return number of halfedges of the mesh

Return num_half Number of halfedges

Author Shankar Kulumani

Version 28 October 2018

Mesh::Vertex_range **vertices** (void) **const**
Return vertex range for iteration

Return Vertex_range Range of vertices of the mesh

Author Shankar Kulumani

Version 28 October 2018

Mesh::Face_range **faces** (void) **const**
Return face range for iteration

Return Face_range Range of faces of the mesh

Author Shankar Kulumani

Version 28 October 2018

Mesh::Edge_range **edges** (void) **const**
Return edge range for iteration

Return Edge_range Range of edges of the mesh

Author Shankar Kulumani

Version 28 October 2018

Mesh::Halfedge_range **halfedges** (void) **const**
Return halfedge range for iteration

Return Halfedge_range Range of halfedges of the mesh

Author Shankar Kulumani

Version 28 October 2018

bool **refine_faces** (**const** std::vector<Face_index> &face_vec, std::vector<Face_index>
&new_faces, std::vector<Vertex_index> &new_vertices, **const** int &den-
sity = 4.0)

Given a set of faces, this will refine them by adding new faces/ vertices within them. The new faces and vertices are returned. The surface mesh member variable is updated.

This tends to make a very craggy shape. Better to use remesh instead

Return new_faces Vector of new faces

Return new_vertices Vector of new vertices

Author Shankar Kulumani

Version 10 June 2018

Parameters

- face_vec: Vector of faces to refine
- density: integer defining the density control factor

remesh_faces (**const** std::vector<Face_index> &face_vec, **const** double &target_edge_length, **const** int &number_of_iterations = 3)

Perform isotropic remeshing of the desired faces of the mesh.

Return bool True if success

Author Shankar Kulumani

Version 28 October 2018

Parameters

- face_vec: Vector of faces to remesh
- target_edge_length: Target length of the new edges
- number_of_iterations: Iterations for isotropic_remeshing

std::vector<Face_index> **faces_in_fov** (**const** Eigen::Ref<**const** Eigen::Vector3d> &pos, **const** double &max_fov = 0.52)

Find the faces that are within a FOV of the current position

Return face_vec Vector of face indices

Author Shankar Kulumani

Version 11 June 2018

Parameters

- pos: Position of spacecraft in the asteroid fixed frame

std::vector<Face_index> **vertices_in_fov** (**const** Eigen::Ref<**const** Eigen::Vector3d> &pos, **const** double &max_fov = 0.52)

Find the vertices that are within a FOV of the current position

Return vertex_vec Vector of vertex indices

Author Shankar Kulumani

Version 28 October 2018

Parameters

- pos: Position of spacecraft in the asteroid fixed frame

Eigen::Matrix<double, Eigen::Dynamic, 3> **refine_faces_in_view**(const Eigen::Ref<const Eigen::Vector3d> &pos, const double &max_fov = 0.52)

Use the refinement function for the faces within a certain angle of the position

Return Face_centers get the center of the new faces in view

Author Shankar Kulumani

Version 28 October 2018

Parameters

- pos: Eigen vector for position
- max_fov: Maximum cosine(angle) for faces to refine

bool **remesh_faces_in_view**(const Eigen::Ref<const Eigen::Vector3d> &pos, const double &max_fov = 0.52, const double &edge_length = 0.01)

Use isotropic remeshing to remesh the faces that are in view of the point

Return bool Success if true

Author Shankar Kulumani

Version 28 October 2018

Parameters

- pos: Position vector
- max_fov: Field of view to use to compute the faces in view
- edge_length: Target edge length for the new faces

template <typename Index>

Eigen::RowVector3d **get_vertex**(const Index &index) const

Get the desired vertex given an index

Return vertex Eigen array of the specific vertex

Author Shankar Kulumani

Version 28 October 2018

Parameters

- Index: Index of vertex to get

bool **set_vertex**(const Vertex_index &vd, const Eigen::Ref<const Eigen::Vector3d> &vec)

Update the position of a single vertex and recompute the affected mesh properties

Return bool True if good

Author Shankar Kulumani

Version 9 June 2018

Parameters

- vd: Vertex index to modify
- vec: Eigen vector of the new vertex position

template <typename Index>

Eigen::RowVector3i **get_face_vertices** (const Index &index) const

Get the vertex indecies of all vertices in the face

Return row a row vector of indices

Author Shankar Kulumani

Version 6 June 2018

Parameters

- index: A face index (basically an integer)

Eigen::VectorXd **get_all_face_area** (void) const

Get the area of each face of the mesh

Return Vector of all face areas

Author Shankar Kulumani

Version 12 June 2018

Parameters

- void: none

Public Members

Mesh **surface_mesh**

Surface_mesh object

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

M

MeshData (*C++ class*), 1
MeshData::build_edge_factor (*C++ function*),
1
MeshData::build_face_factor (*C++ function*),
2
MeshData::edges (*C++ function*), 3
MeshData::faces (*C++ function*), 3
MeshData::faces_in_fov (*C++ function*), 4
MeshData::get_all_face_area (*C++ function*),
6
MeshData::get_face_vertices (*C++ function*),
6
MeshData::get_faces (*C++ function*), 2
MeshData::get_vertex (*C++ function*), 5
MeshData::get_verts (*C++ function*), 2
MeshData::halfedges (*C++ function*), 3
MeshData::number_of_edges (*C++ function*), 2
MeshData::number_of_faces (*C++ function*), 2
MeshData::number_of_halfedges (*C++ func-
tion*), 3
MeshData::number_of_vertices (*C++ func-
tion*), 2
MeshData::refine_faces (*C++ function*), 3
MeshData::refine_faces_in_view (*C++ func-
tion*), 4
MeshData::remesh_faces (*C++ function*), 4
MeshData::remesh_faces_in_view (*C++ func-
tion*), 5
MeshData::set_vertex (*C++ function*), 5
MeshData::surface_mesh (*C++ member*), 6
MeshData::update_mesh (*C++ function*), 1
MeshData::vertices (*C++ function*), 3
MeshData::vertices_in_fov (*C++ function*), 4