fc meshtools Python package, User's Guide*

version 0.1.1

F. Cuvelier†

May 14, 2019

Abstract

The experimental fc meshtools Python package contains some simplicial meshes given by their vertices array q and connectivity array q. Theses meshes can be easily used in other Python codes for debugging or testing purpose. Some functions are provided to compute, for each mesh elements, volumes, gradient of barycentric coordinates, ...

*LaTeX manual, revision 0.1.1.a, compiled with Python 3.7.3, and packages `fc-meshtools`[0.1.1], `fc-tools`[0.0.22],

†LAGA, UMR 7539, CNRS, Université Paris 13 - Sorbonne Paris Cité, Université Paris 8, 99 Avenue J-B Clément, F-93430 Villetaneuse, France, cuvelier@math.univ-paris13.fr

0 Contents

1 Introduction

A simplicial mesh is given by its vertices array q and its connectivity array me . For demonstration purpose, some simplicial meshes are given in this package and stored in the `fc_meshtools/data` directory. They can be load by using the functions `getMesh2D`, `getMesh3D` or `getMesh3Ds` of the `fc_meshtools.simplicial` module. Here are the kind of simplicial meshes present in this toolbox:

- a triangular mesh in dimension 2, made with 2-simplices (ie. triangles),
- a tetrahedral mesh in dimension 3, made with 3-simplices (ie. tetrahedron),
- a triangular mesh in dimension 3 (surface mesh), made with 2-simplices,
- a line mesh in dimension 2 or 3 made with 1-simplices (ie. lines).

This package was tested under

OS	**Python**
CentOS 7.6	python.org [3] 2.7.16, 3.5.7, 3.6.8, 3.7.3
Debian 9.9	python.org [3] 2.7.16, 3.5.7, 3.6.8, 3.7.3
Ubuntu 18.04 LTS	python.org [3] 2.7.16, 3.5.7, 3.6.8, 3.7.3
OpenSuse 15.0	python.org [3] 3.5.7, 3.6.8, 3.7.3
Fedora 29	python.org [3] 2.7.16, 3.5.7, 3.6.8, 3.7.3
MacOS Mojave 10.14.4	python.org [3]: 3.5.4, 3.6.8, 3.7.3
Windows 10 (1809)	python.org [3]: 3.5.4, 3.6.8, 3.7.3

2 Installation

The fc meshtools Python package is available from the Python Package Index [2].

2.1 On Linux

For Python 2, the installation of this package can be done with the `pip` or `pip2` command.

- For an installation which isolated to the current user, one can do:

```
$ pip install --user fc_meshtools
```

- For an installation for all users, one can do:

```
$ sudo pip install fc_meshtools
```

For Python 3 installation, sometimes `pip3` must be used instead of `pip`.

2.2 On windows

to do

2.3 On macOS

to do

3 Simplicial meshes

The functions `getMesh2D`, `getMesh3D` and `getMesh3Ds` of the module `fc_meshtools.simplicial` return a mesh vertices array `q`, a mesh elements connectivity array `me` and an indices array `toGlobal` associated with the input argument d (simplex dimension). The vertices array `q` is a dim-by-n_q array where dim is the space dimension (2 or 3) and n_q the number of vertices. The connectivity array `me` is a $(d+1)$-by-n_{me} array where n_{me} is the number of mesh elements and $0 \leqslant d \leqslant dim$ is the simplicial dimension:

- $d = 0$: points,
- $d = 1$: lines,
- $d = 2$: triangle,
- $d = 3$: tetrahedron.

So we can use theses functions to obtain

- 3D mesh: `getMesh3D(3)` (*main* mesh), `getMesh3D(2)`, `getMesh3D(1)`, `getMesh3D(0)`,
- 3D surface mesh: `getMesh3Ds(2)` (*main* mesh), `getMesh3Ds(1)` , `getMesh3Ds(0)`,
- 2D mesh: `getMesh2D(2)` (*main* mesh), `getMesh2D(1)`, `getMesh2D(0)`.

The indices array `toGlobal` contains the indices of the vertices in the *main* mesh
For example,

- `q3,me3,toGlobal3=fc_meshtools.simplicial.getMesh3D(3)` return a 3-simplicial mesh (main mesh) in space dimension $dim = 3$,
- `q2,me2,toGlobal2=fc_meshtools.simplicial.getMesh3D(2)` return a 2-simplicial mesh in space dimension $dim = 3$.
- `q1,me1,toGlobal1=fc_meshtools.simplicial.getMesh3D(1)` return a 1-simplicial mesh in space dimension $dim = 3$.

The third output contains the indices of the vertices in the *main* mesh:

```
q3[:,toGlobal2] == q2
q3[:,toGlobal1] == q1
```

4 Functions

4.1 getMesh functions

Returns a vertices array `q`, a connectivity array `me` and an indices array `toGlobal`.

Description

`q,me,toGlobal=fc_meshtools.simplicial.getMesh3D(d)`

`q,me,toGlobal=fc_meshtools.simplicial.getMesh3Ds(d)`

`q,me,toGlobal=fc_meshtools.simplicial.getMesh2D(d)`

Returns a vertices array `q`, a connectivity array `me` and an indices array `toGlobal` depending on the value of the `d`. For a 3D mesh, `d` $\in [\![0, 3]\!]$, and for a 2D or 3Ds mesh, `d` $\in [\![0, 2]\!]$.

In Listing 1, some examples are provided.

Listing 1: : examples of fc_meshtools.simplicial.getMesh3D function usage

```
from fc_meshtools.simplicial import getMesh3D
q2,me2,toG2=getMesh3D(2)
print('q2: %s, me2: %s, toG2: %s: '%(str(q2.shape),str(me2.shape),str(toG2.shape)))
q3,me3,toG3=getMesh3D(3)
print('q3: %s, me3: %s, toG3: %s: '%(str(q3.shape),str(me3.shape),str(toG3.shape)))
print('Error: %.5e'%abs(q3[:,toG2]-q2).max())
```

Output

```
q2: (3, 3133), me2: (3, 6274), toG2: (3133,):
q3: (3, 5168), me3: (4, 22234), toG3: (5168,):
Error: 0.00000e+00
```

4.2 Volumes function

Syntaxe Returns all the element volumes of a mesh given by a vertices array q and a connectivity array me. One can refer to [1] for computationnal details.

Description

vols=fc_meshtools.simplicial.Volumes(q,me)

vols[k] is the volume of the (k+1)-th mesh element where its vertices are the columns of q[me[:,k]].

In Listing 2, some examples are provided.

Listing 2: : examples of fc_meshtools.simplicial.Volumes function usage

```
from fc_meshtools.simplicial import getMesh3D,Volumes
q,me,toG=getMesh3D(2)
vols=Volumes(q,me)
print('q: %s, me: %s, vols: %s: '%(str(q.shape),str(me.shape),str(vols.shape)))
```

Output

```
q: (3, 3133), me: (3, 6274), vols: (6274,):
```

4.3 Gradient of barycentric coordinates

Syntaxe Returns all the gradients of barycentric coordinates of each element of a mesh given by a vertices array q and a connectivity array me. One can refer to [1] for computationnal details.

Description

G=fc_meshtools.simplicial.GradBaCo(q,me)

G[k,:,i] is the gradient of the i-th barycentric coordinate of the k-th mesh element.

In Listing 3, some examples are provided.

Listing 3: : examples of fc_meshtools.simplicial.GradBaCo function usage

```
from fc_meshtools.simplicial import getMesh3D,GradBaCo
q,me,toG=getMesh3D(3)
G=GradBaCo(q,me)
print('q: %s, me: %s, G: %s: '%(str(q.shape),str(me.shape),str(G.shape)))
```

Output

```
q: (3, 5168), me: (4, 22234), G: (22234, 3, 4):
```

4 References

[1] F. Cuvelier. Exact integration for products of power of barycentric coordinates over d-simplexes in R^n. `http://hal.archives-ouvertes.fr/hal-00931066v1`, June 2018. preprint.

[2] Python Software Foundation. Pypi, the python package index. `https://pypi.python.org/`, 2003–.

[3] Python.org. Python. `http://www.python.org/`, 2013.