FC-HYPERMESH Octave package, User's Guide * under rhum-ubuntu-16-04-3lts computer

François Cuvelier†

April 18, 2018

Abstract

This object-oriented Octave package allows to mesh any d-orthotopes (hyperrectangle in dimension d) and their m-faces by simplices or orthotopes. It was created to show the implementation of the vectorized algorithms presented in []. The FC-HYPERMESH package uses Octave objects and is provided with meshes visualisation tools for dimension less than or equal to 3.

0 Contents

*Compiled with Octave 4.2.1

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

This work was supported by the ANR project DEDALES under grant ANR-14-CE23-0005.

1 Introduction

The FC-HYPERMESH package contains a simple class object `OrthMesh` which permits, in any dimension $d \geqslant 1$, to obtain a simplicial mesh or orthotope mesh with all their m-faces, $0 \leqslant m < d$. It is also possible with the method function `plotmesh` of the class object `OrthMesh` to represent a mesh or its m-faces for $d \leqslant 3$.

This package was tested under

Windows 10.0.16299: with Octave 4.2.0 and 4.2.1

Mac OS X 10.12.6: with Octave 4.2.1 (installed with homebrew)

Ubuntu 14.04.5 LTS: with Octave 4.2.0 and 4.2.1 (both compiled from source)

Ubuntu 16.04.3 LTS: with Octave 4.2.0 and 4.2.1 (both compiled from source)

Ubuntu 17.10: with Octave 4.2.0 and 4.2.1 (both compiled from source)

Fedora 27: with Octave 4.2.0 and 4.2.1 (both compiled from source)

It is not compatible with Octave 4.0.x and previous.

In the following section, the class object `OrthMesh` is presented. Thereafter some warning statements on the memory used by these objects in high dimension are given. Finally computation times for orthotope meshes and simplicial meshes are given in dimension $d \in [\![1, 5]\!]$.

2 Installation

2.1 Installation via pkg command (recommended)

- Download the packages. For example, in a terminal:

```
# wget http://www.math.univ-paris13.fr/~cuvelier/software/codes/
    Octave/fc-tools/0.0.20/fc-tools-0.0.20.tar.gz
# wget http://www.math.univ-paris13.fr/~cuvelier/software/codes/
    Octave/fc-hypermesh/0.0.7/fc-hypermesh-0.0.7.tar.gz
```

- Under Octave :

```
>> pkg install fc-tools-0.0.20.tar.gz
>> pkg install fc-hypermesh-0.0.7.tar.gz
```

- Now to use `fc-hypermesh` in any Octave session, it is necessary to load the package:

```
>> pkg load fc-hypermesh
```

- To try the package, one can launch a demo:

```
>> fc_hypermesh.demo01
```

For uninstalling the package, just do in an Octave session:

```
>> pkg uninstall fc-hypermesh
>> pkg uninstall fc-tools
```

2.2 All-in-one installation

For this method, one just has to get/download the install file

ofc_hypermesh_install.m

or get it on the dedicated web page. Thereafter, it should be run under Octave. This command downloads, extracts and configures the *fc-hypermesh* and the required *fc-tools* packages in the current directory.

For example, to install this package in directory `~/Octave/packages`, in a terminal one can do:

```
# mkdir -p ~/Octave/packages
# cd ~/Octave/packages
# wget http://www.math.univ-paris13.fr/~cuvelier/software/codes/Octave/fc
    -hypermesh/0.0.7/ofc_hypermesh_install.m
```

Then in a Octave terminal run the following commands

```
>> cd ~/Octave/toolboxes
>> ofc_hypermesh_install
```

This is the output of the `ofc_hypermesh_install` command:

```
1- Downloading and extracting the packages
   -> <fc-tools>[0.0.20] ... OK
   -> <fc-hypermesh>[0.0.7] ... OK
2- Setting the <fc-hypermesh> package
Write in ~/Octave/toolboxes/fc-hypermesh-full/fc_hypermesh-0.0.7/
    configure_loc.m ...
  -> done
3- Using the <fc-hypermesh> package
   Under Octave:
     addpath('~/Octave/toolboxes/./fc-hypermesh-full/fc_hypermesh-0.0.7')
     fc_hypermesh.init()

   See ~/Octave/toolboxes/ofc_hypermesh_set.m
```

The complete package (i.e. with all the other needed packages) is stored in the directory `~/Octave/packages/fc-hypermesh-full` and, for each Octave session, one has to set the package by:

```
>> addpath('~/Octave/packages/fc-hypermesh-full/ofc-hypermesh-0.0.7')
>> fc_hypermesh.init()
```

To **uninstall**, one just has to delete directory `~/Octave/packages/fc-hypermesh-full`

3 Using the fc-hypermesh package

First of all, the main class object `OrthMesh` is presented. Thereafter some usage samples are given.

3.1 Class object `OrthMesh`

The aim of the class object `OrthMesh` is to efficiently create an object which contains a mesh of a d-orthotope and all its m-face meshes. An elementary mesh class object `EltMesh` is used to store only one mesh, the main mesh as well as any of the m-face meshes. This class `EltMesh` also simplifies (for me) the codes writing. Its fields are the following:

- d, space dimension
- m, kind of mesh ($m = \mathrm{d}$ for the main mesh)
- type, 0 for simplicial mesh or 1 for orthotope mesh
- $\mathrm{n_q}$, number of vertices
- q, vertices array of dimension d-by-$\mathrm{n_q}$
- $\mathrm{n_{me}}$, number of mesh elements
- me, connectivity array of dimension $(\mathrm{d}+1)$-by-$\mathrm{n_{me}}$ for simplices elements or 2^{d}-by-$\mathrm{n_{me}}$ for orthotopes elements
- `toGlobal`, index array linking local array q to the one of the main mesh
- label, name/number of this elementary mesh
- color, color of this elementary mesh (for plotting purpose)

Let the d-orthotope defined by $[a_1, b_1] \times \cdots \times [a_{\mathrm{d}}, b_{\mathrm{d}}]$. The class object `OrthMesh` corresponding to this d-orthotope contains the main mesh and all its m-face meshes, $0 \leqslant m < \mathrm{d}$. Its Fields are the following

- d: space dimension
- type: string 'simplicial' or 'orthotope' mesh
- Mesh: main mesh as an `EltMesh` object
- Faces: list of arrays of `EltMesh` objects such that Faces(1) is an array of all the $(\mathrm{d}-1)$-face meshes, Faces(2) is an array of all the $(\mathrm{d}-2)$-face meshes, and so on
- box: a d-by-2 array such that $\mathrm{box}(i,1) = a_i$ and $\mathrm{box}(i,2) = b_i$.

Constructor

The `OrthMesh` constructor is :

$$\text{Oh} = \text{OrthMesh(d,N)}$$

where N is either a 1-by-d array/list such that N[i−1] is the number of discretization for $[a_i, b_i]$ or either an integer if the the number of discretization is the same in all space directions.
Some options are proposed with the constructor:

$$\text{Oh} = \text{OrthMesh(d,N,Name,Value)}$$

Options are defined with one or more Name,Value pair arguments. The Name argument could be the string

- `'box'` : used to specify the d-orthotope $[a_1, b_1] \times \ldots \times [a_d, b_d]$ by setting Value as an d-by-2 array such that a_i =Value(i,1) and b_i =Value(i,2).
- `'type'` : used to select the kind of elements used for meshing. The default value is `'simplicial'` and otherwise `'orthotope'` can be used.

plotmesh method

The plotmesh() member function can be used to represent the mesh given by an `OrthMesh` object if the space dimension is less than or equal to 3. Some options are proposed with this function:

$$\text{plotmesh(Name,Value)}$$

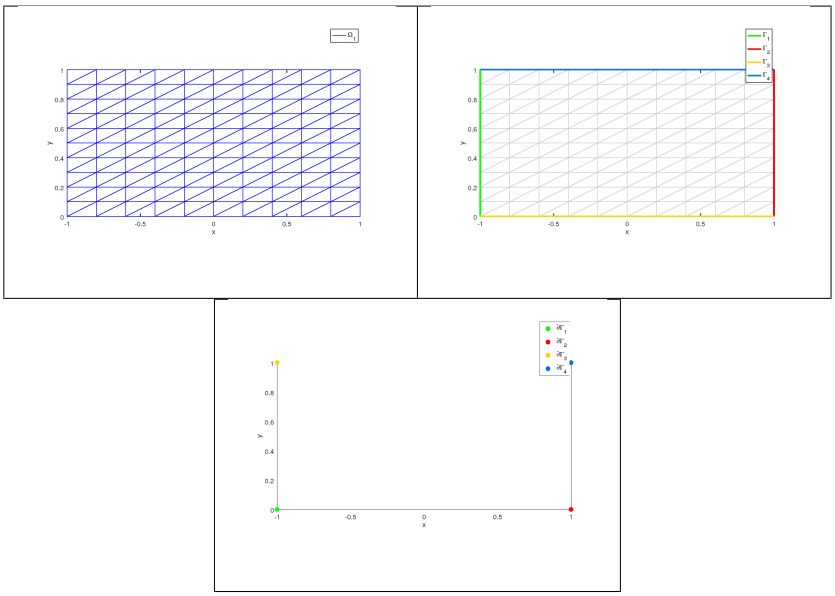
- 'legend = value : if value is True, a legend is displayed. Default is False.
- m = value : plots all the m-faces of the mesh. Default m = d i.e. the main mesh. (0 ⩽m⩽d)
- ...

3.2 2d-orthotope meshing by simplices

In Listing 15, an `OrthMesh` object is built under Octave by using command

```
Oh=OrthMesh(2,10,'box',[-1,1;0,1])
```

So the Oh object is the tessellations of the orthotope $[-1, 1] \times [0, 1]$ with simplicial elements. In each direction $10 + 1(= 11!)$ points are taken. So we have 11^2 vertices in this mesh. The main mesh and all the m-face meshes of the resulting object are plotted by using plotmesh method.

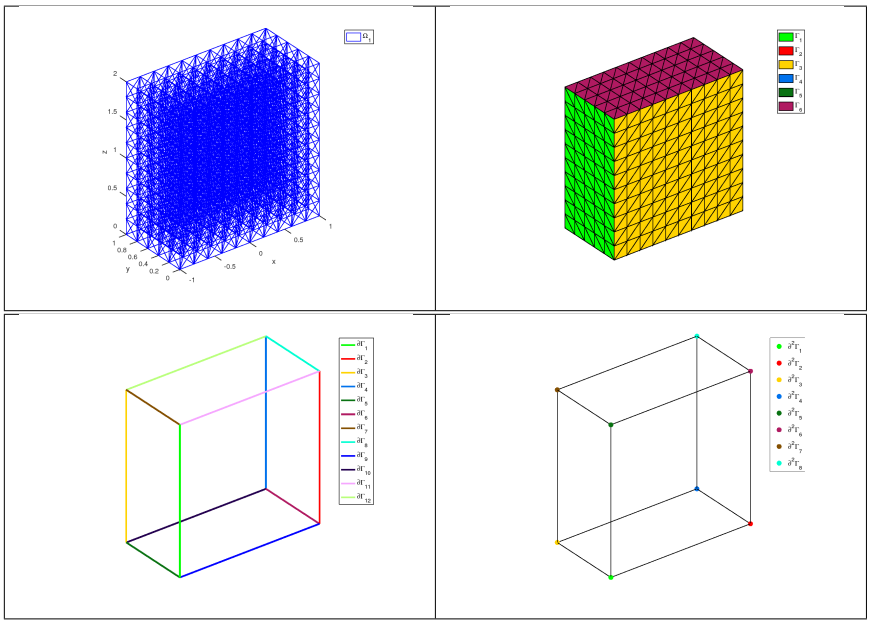


```
Oh=OrthMesh(2,10,'box',[-1,1;0,1])
% plot the main mesh
figure(1)
Oh.plotmesh('legend',true)
% plot the 1-face meshes
figure(2)
Oh.plotmesh('color',[0.8,0.8,0.8])
hold on
Oh.plotmesh('m',1,'Linewidth',2,'legend',true)
% plot the 0-face meshes
figure(3)
Oh.plotmesh('m',1,'color','k')
hold on
Oh.plotmesh('m',0,'legend',true)
```

Listing 1: 2D simplicial `OrthMesh` object with Octave 4.2.1, main mesh (upper left), 1-face meshes (upper right), and 0-face meshes (bottom)

3.3 3d-orthotope meshing by simplices

In Listing 22, an `OrthMesh` object is built under Octave for the orthotope $[-1,1] \times [0,1] \times [0,2]$ with simplicial elements and $\mathbf{N} = (10,5,10)$. The main mesh and all the m-face meshes of the resulting object are plotted.

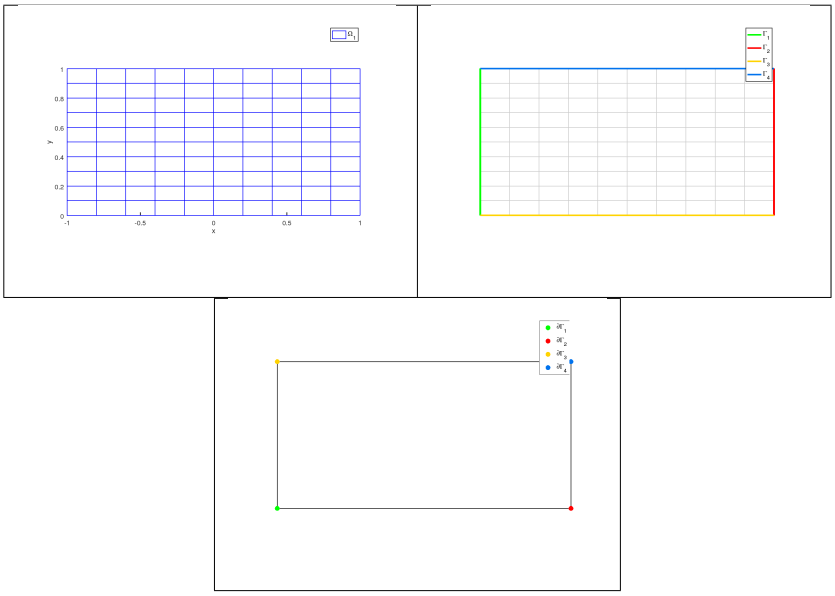


```
Oh=OrthMesh(3,[10,5,10],'box',[-1,1;0,1;0,2])
% plot the main mesh
figure(1)
Oh.plotmesh('legend',true)
axis equal;xlabel('x');ylabel('y');zlabel('z')
% plot the 2-face meshes
figure(2)
Oh.plotmesh('m',2,'legend',true)
axis equal;axis off
% plot the 1-face meshes
figure(3)
Oh.plotmesh('m',2,'color',[0.8,0.8,0.8],'EdgeAlpha',0.2, ...
    'FaceColor','none')
hold on
Oh.plotmesh('m',1,'Linewidth',2,'legend',true)
axis equal;axis off
% plot the 0-face meshes
figure(4)
Oh.plotmesh('m',1,'color','k')
hold on
Oh.plotmesh('m',0,'legend',true)
axis equal;axis off
```

Listing 2: 3D simplicial `OrthMesh` object with Octave 4.2.1, main mesh (upper left), 2-face meshes (upper right), 1-face meshes (bottom left) and 0-face meshes (bottom right)

3.4 2d-orthotope meshing by orthotopes

In Listing 22, an `OrthMesh` object is built under Octavefor the orthotope $[-1,1]\times[0,1]\times[0,2]$ with orthotope elements and $\mathbf{N}=(10,5,10)$. The main mesh and all the m-face meshes of the resulting object are plotted.



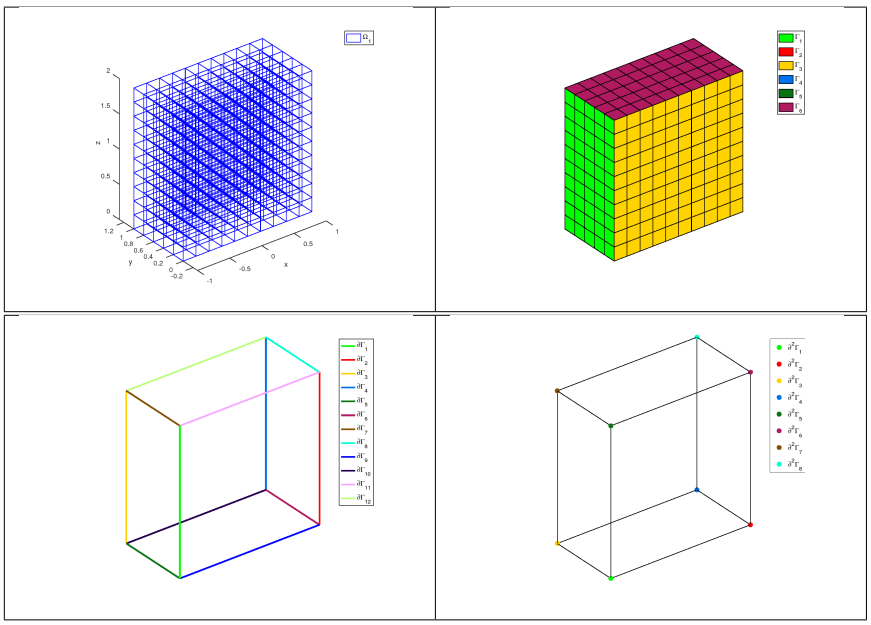
```
Oh=OrthMesh(2,10,'box',[-1,1;0,1],'type','orthotope')
% plot the main mesh
figure(1)
Oh.plotmesh('legend',true)
axis equal;xlabel('x');ylabel('y')
% plot the 1-face meshes
figure(2)
Oh.plotmesh('color',[0.8,0.8,0.8])
hold on
Oh.plotmesh('m',1,'Linewidth',2,'legend',true)
axis equal;axis off
% plot the 0-face meshes
figure(3)
Oh.plotmesh('m',1,'color','k')
hold on
Oh.plotmesh('m',0,'legend',true)
axis equal;axis off
```

Listing 3: 2D orthotope `OrthMesh` object with Octave 4.2.1, main mesh (upper left), 1-face meshes (upper right), and 0-face meshes (bottom)

3.5 3d-orthotope meshing by orthotopes

In Listing 22, an `OrthMesh` object is built under Octave for the orthotope $[-1,1]\times[0,1]\times[0,2]$ with orthotope elements and $\mathbf{N}=(10,5,10)$. The main

mesh and all the m-face meshes of the resulting object are plotted.



```
Oh=OrthMesh(3,[10,5,10],'box',[-1,1;0,1;0,2], ...
    'type','orthotope')
% plot the main mesh
figure(1)
Oh.plotmesh('legend',true)
axis equal;xlabel('x');ylabel('y');zlabel('z')
% plot the 2-face meshes
figure(2)
Oh.plotmesh('m',2,'legend',true)
axis equal;axis off
% plot the 1-face meshes
figure(3)
Oh.plotmesh('m',2,'color',[0.8,0.8,0.8],'EdgeAlpha',0.2, ...
    'FaceColor','none')
hold on
Oh.plotmesh('m',1,'Linewidth',2,'legend',true)
axis equal;axis off
% plot the 0-face meshes
figure(4)
Oh.plotmesh('m',1,'color','k')
hold on
Oh.plotmesh('m',0,'legend',true)
axis equal;axis off
```

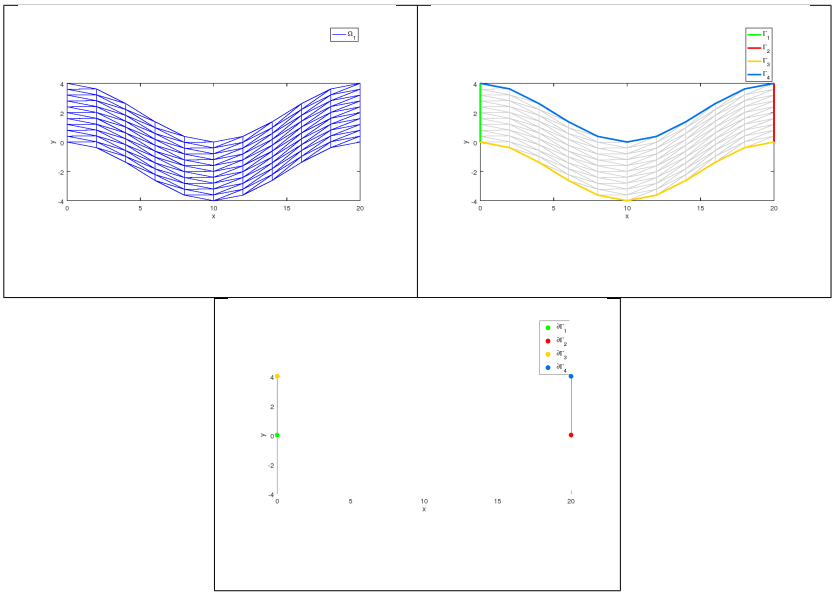
Listing 4: 3D orthotope `OrthMesh` object with Octave 4.2.1, main mesh (upper left), 2-face meshes (upper right), 1-face meshes (bottom left) and 0-face meshes (bottom right)

3.6 Mapping of a 2d-orthotope meshing by simplices

For example, the following 2D geometrical transformation allows to deform the reference unit hypercube.

$$[0,1]\times[0,1] \longrightarrow \mathbb{R}^2$$

$$\begin{pmatrix} x \\ y \end{pmatrix} \longrightarrow F(x,y) = \begin{pmatrix} 20\,x \\ 2\,(2\,y - 1 + \cos(2\pi x)) \end{pmatrix}$$



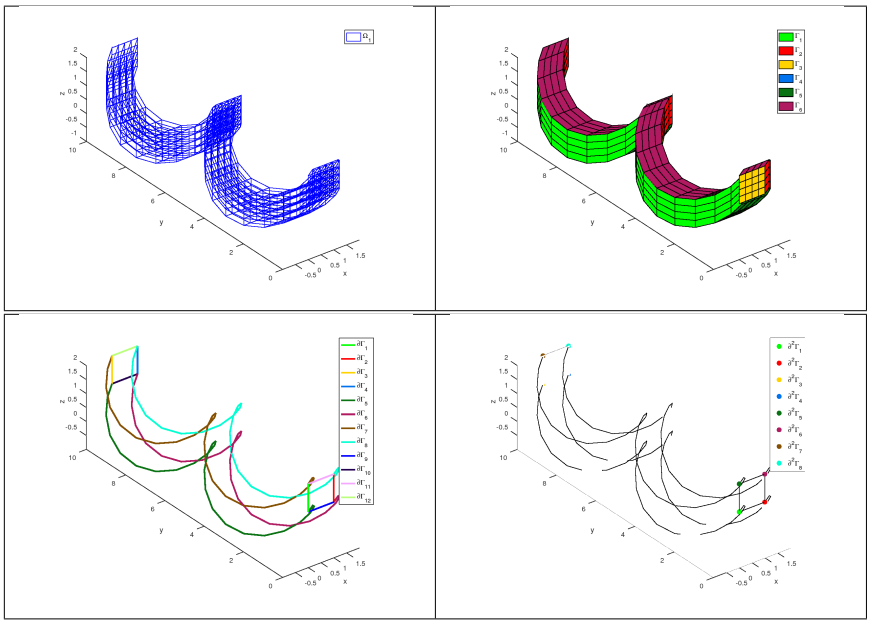
```
Oh=OrthMesh(2,10,'mapping',@(q) [20*q(1,:); ...
    2*(2*q(2,:)-1+cos(2*pi*q(1,:)))]);
% plot the main mesh
figure(1)
Oh.plotmesh('legend',true)
% plot the 1-face meshes
figure(2)
Oh.plotmesh('color',[0.8,0.8,0.8])
hold on
Oh.plotmesh('m',1,'Linewidth',2,'legend',true)
% plot the 0-face meshes
figure(3)
Oh.plotmesh('m',1,'color','k')
hold on
Oh.plotmesh('m',0,'legend',true)
```

Listing 5: Mapping of a 2D simplicial `OrthMesh` object with Octave 4.2.1, main mesh (upper left), 1-face meshes (upper right), and 0-face meshes (bottom)

3.7 3d-orthotope meshing by orthotopes

For example, the following 3D geometrical transformation allows to deform the reference unit hypercube.

$$
\begin{array}{rcl}
[0,1]\times[0,1]\times[0,1] & \longrightarrow & \mathbb{R}^2 \\
\begin{pmatrix} x \\ y \\ z \end{pmatrix} & \longrightarrow & F(x,y,y) = \begin{pmatrix} x+\sin(4\pi y) \\ 10y \\ z+\cos(4\pi y) \end{pmatrix}
\end{array}
$$



```
Map=@(q) [q(1,:)+ sin(4*pi*q(2,:)); 10*q(2,:); ...
    q(3,:)+cos(4*pi*q(2,:))];
Oh=OrthMesh(3,[4,25,4],'mapping',Map,'type','orthotope');
% plot the main mesh
figure(1)
Oh.plotmesh()
legend('show')
% plot the 2-face meshes
figure(2)
Oh.plotmesh('m',2)
legend('show')
% plot the 1-face meshes
figure(3)
Oh.plotmesh('m',2,'color',[0.8,0.8,0.8],'EdgeAlpha',0.2, ...
    'FaceColor','none')
hold on
Oh.plotmesh('m',1,'Linewidth',2,'legend',true)
% plot the 0-face meshes
figure(4)
Oh.plotmesh('m',1,'color','k')
hold on
Oh.plotmesh('m',0,'legend',true)
```

Listing 6: Mapping of a 3D orthotope `OrthMesh` object with Octave 4.2.1, main mesh (upper left), 2-face meshes (upper right), 1-face meshes (bottom left) and 0-face meshes (bottom right)