



© hypermesh Matlab toolbox, User's Guide *

François Cuvelier[†] Gilles Scarella[‡]

April 20, 2018

Abstract

This object-oriented Matlab toolbox allows to generate conforming meshes of hypercubes, hyperrectangles or of any d -orthotopes by simplices or orthotopes with their m -faces. It was created to show the implementation of the algorithms of [1]. The © hypermesh toolbox uses Matlab objects and is provided with meshes visualisation tools for dimension less than or equal to 3.

Contents

1	Introduction	2
2	Installation	2
2.1	Installation automatic, all in one (recommended)	2
2.2	Manual installation	3
3	Using the fc_hypermesh toolbox	3
3.1	Class object OrthMesh	4
3.2	2d-orthotope meshing by simplices	6
3.3	3d-orthotope meshing by simplices	7
3.4	2d-orthotope meshing by orthotopes	8
3.5	3d-orthotope meshing by orthotopes	9
3.6	Mapping of a 2d-orthotope meshing by simplices	10
3.7	3d-orthotope meshing by orthotopes	11

*Compiled with Matlab 2017a, toolboxes `fc_hypermesh-10.0.7` and `fc_tools-10.0.21` under hercule computer

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

[‡]Université Côte d'Azur, CNRS, LJAD, F-06108 Nice, France, gilles.scarella@unice.fr.

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

4 Benchmarking	12
4.1 <code>fc_bench.bench01</code> function	12
4.2 Examples	13

1 Introduction

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

This toolbox was tested under

Windows 10.0.16299: with Matlab R2015b to R2018a

macOS High Sierra 10.13.4: with Matlab R2015b to R2018a

Ubuntu 16.04.3 LTS: with Matlab R2015b to R2018a

Ubuntu 17.10: with Matlab R2015b to R2018a

centOS 7.4: with Matlab R2015b to R2018a

Fedora 27: with Matlab R2015b to R2018a

OpenSUSE Leap 42.3: with Matlab R2015b to R2018a

With Matlab R2015b there is a trouble with the legend of the 0-faces (points) : same color for all points!

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 $d \in \llbracket 1, 5 \rrbracket$.

2 Installation

2.1 Installation automatic, all in one (recommended)

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

```
mfc_hypermesh_install.m
```

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

For example, to install this toolbox in `~/Matlab/toolboxes` directory, one has to copy the file `mfc_hypermesh_install.m` in the `~/Matlab/toolboxes` directory. Then in a Matlab terminal run the following commands

```
>> cd ~/Matlab/toolboxes
>> mfc_hypermesh_install
```

This is the output of the `mfc_hypermesh_install` command on a Linux computer:

```
Parts of the <fc-hypermesh> Matlab toolbox.
Copyright (C) 2017-2018 F. Cuvelier <cuvelier@math.univ-paris13.fr>

1- Downloading and extracting the toolboxes
2- Setting the <fc-hypermesh> toolbox
Write in ~/Matlab/toolboxes/fc-hypermesh-full/fc_hypermesh-0.0.7/
  configure_loc.m ...
3- Using toolboxes :
->          fc-tools : 0.0.21
->          fc-hypermesh : 0.0.7
*** Using instructions
To use the <fc-hypermesh> toolbox:
  addpath('~/Matlab/toolboxes/fc-hypermesh-full/fc_hypermesh-0.0.7')
  fc_hypermesh.init()

See ~/Matlab/toolboxes/mfc_hypermesh_set.m
```

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

```
>> addpath('~/Matlab/toolboxes/fc-hypermesh-full/mfc-hypermesh-0.0.7')
>> fc_hypermesh.init()
```

To **uninstall**, one just has to delete directory

```
~/Matlab/toolboxes/fc-hypermesh-full
```

2.2 Manual installation

This package uses the `fc_tools` toolbox. So one has to install it as explain in the dedicated web page.

Thereafter, on the `fc_hypermesh` dedicated web page, one can found link to archives (*zip*, *7z* or *tar.gz* format)

- Downloads an archive and extracts it on a folder, for example `~/Matlab/toolboxes`. The toolbox path is `~/Matlab/toolboxes/mfc-hypermesh-0.0.7`
- Adds the toolbox path in Matlab with `addpath` command.
- Verifies that the `fc_tools` toolbox is in the Matlab path. Otherwise, adds it...

3 Using the hypermesh toolbox

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 and its fields are the following:

- `d`, space dimension
- `m`, kind of mesh ($m = d$ for the main mesh)
- `type`, 0 for simplicial mesh or 1 for orthotope mesh
- `nq`, number of vertices
- `q`, vertices array of dimension d -by-`nq`
- `nme`, number of mesh elements
- `me`, connectivity array of dimension $(d + 1)$ -by-`nme` for simplices elements or 2^d -by-`nme` 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 $[a_1, b_1] \times \dots \times [a_d, b_d]$. The class object `OrthMesh` corresponding to this d -orthotope contains the main mesh and all its m -face meshes, $0 \leq m < d$. 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 $(d - 1)$ -face meshes, `Faces(2)` is an array of all the $(d - 2)$ -face meshes, and so on
- `box`: a d -by-2 array such that `box(i,1) = ai` and `box(i,2) = bi`.

3.1.1 Constructor

```
Oh = OrthMesh(d,N)
Oh = OrthMesh(d,N, key, value, ...)
```

Description

```
Oh = OrthMesh(d,N)
```

Generates the `OrthMesh` object `Oh` which contains which contains a simplicial mesh of the unit `d`-orthotope and all its `m`-face meshes.

```
Oh = OrthMesh(d,N, key,value, ...)
```

Some optional `key/value` pairs arguments are available with `key`:

- `'type'` : used to select the kind of elements used for meshing. The default `value` is `'simplicial'` and otherwise `'orthotope'` can be used.

```
Oh = OrthMesh(3,10, 'type','orthotope')
```

- `'box'` : used to specify the `d`-orthotope $[a_1, b_1] \times \dots \times [a_d, b_d]$ by setting `value` as an `d`-by-2 array such that $a_i = \text{value}(i,1)$ and $b_i = \text{value}(i,2)$.

```
Oh = OrthMesh(3,10, 'box',[-1 1;-2 2;0 3])
```

- `'m_min'` : used to only mesh the `m`-Faces for `m` in $[[m,d]]$. Default `value` is 0.

```
Oh = OrthMesh(3,10, 'm_min',2)
```

- `'mapping'` : used to apply on the mesh a mapping function given by a function handle.

```
Oh = OrthMesh(3,10, 'mapping',@(q) [q(1,:)+sin(q(2,:));q(2,:);q(3,:)])
```

3.1.2 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.

Syntaxe

```
obj.plotmesh()  
obj.plotmesh(key, value, ...)
```

Description

```
obj.plotmesh()
```

```
obj.plotmesh(key, value, ...)
```

Some optional `key/value` pairs arguments are available with `key`:

- `'legend'` : if `value` is `True`, a legend is displayed. Default is `False`.

- 'm' : plots all the m-faces of the mesh. Default $m = d$ i.e. the main mesh. ($0 \leq m \leq d$)
- 'color' : use to specify the color of the mesh.
- ...

Other **key/value** pairs arguments can be used depending of **obj.d** and **obj.m** values and they are those of the plotting function used:

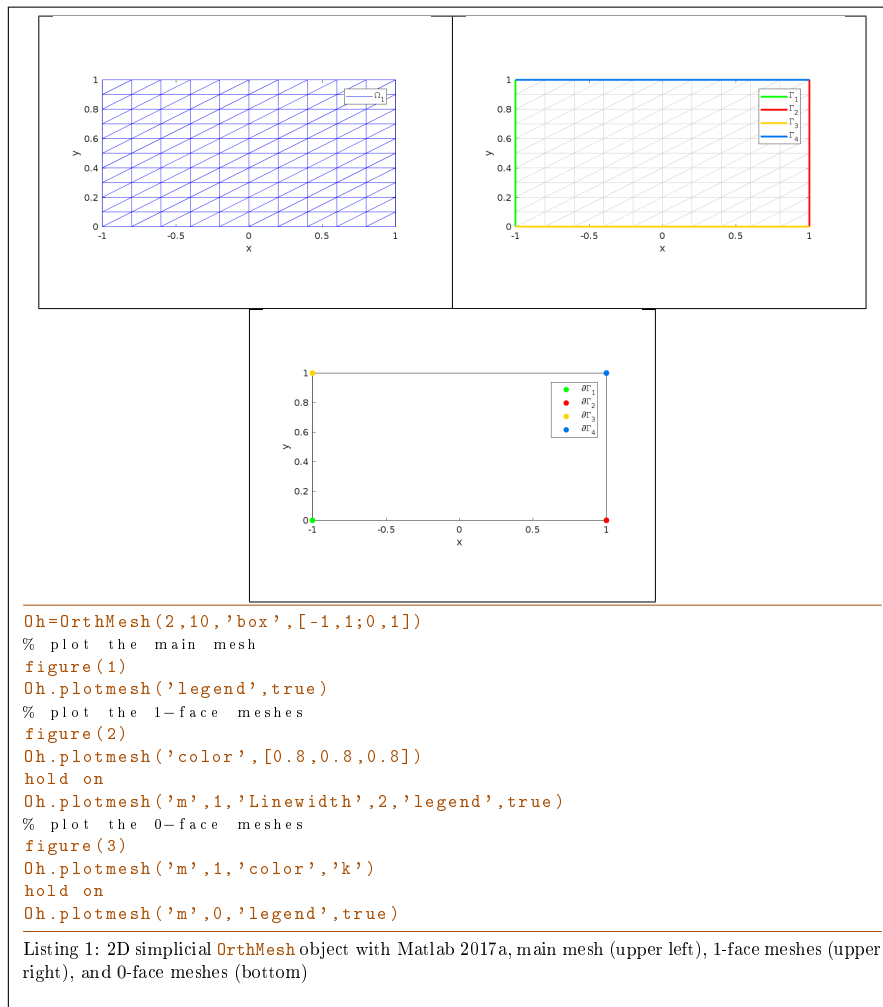
- with **obj.d=3** and **obj.m=3** , **patch** function is used;
- with **obj.d=3** and **obj.m=2** , **trimesh** function is used for simplicial mesh and **patch** function is used for orthotope mesh;
- with **obj.d=3** and **obj.m=1** , **line** function is used;
- with **obj.d=3** and **obj.m=0** , **scatter3** function is used;
- with **obj.d=2** and **obj.m=2** , **triplot** function is used for simplicial mesh and **patch** function is used for orthotope mesh;
- with **obj.d=2** and **obj.m=1** , **line** function is used;
- with **obj.d=2** and **obj.m=0** , **scatter** function is used;
- with **obj.d=1** and **obj.m=1** , **line** function is used;
- with **obj.d=1** and **obj.m=0** , **scatter** function is used;

3.2 2d-orthotope meshing by simplices

In Listing 1, an **OrthMesh** object is built under Matlab by using command

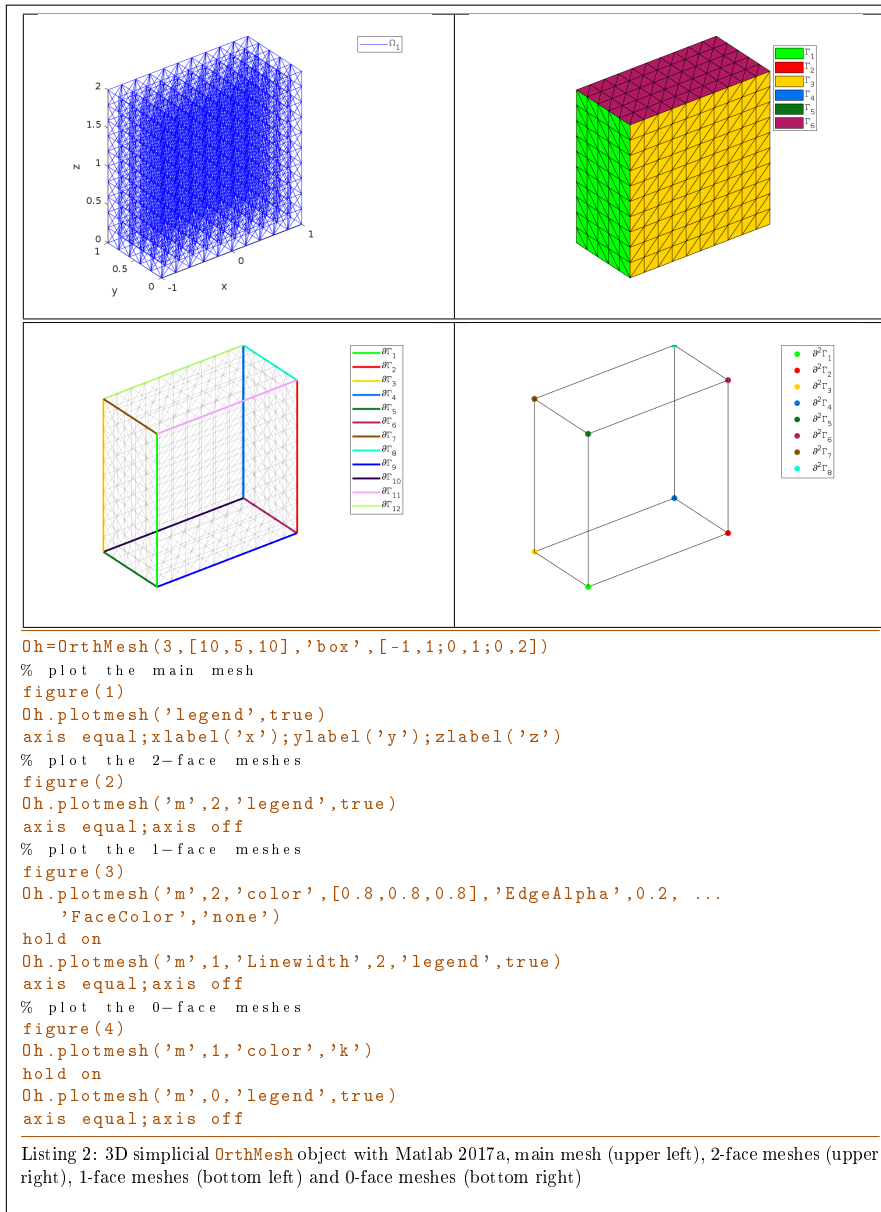
```
Oh=OrthMesh(2,10,'box',[-1,1;0,1])
```

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.



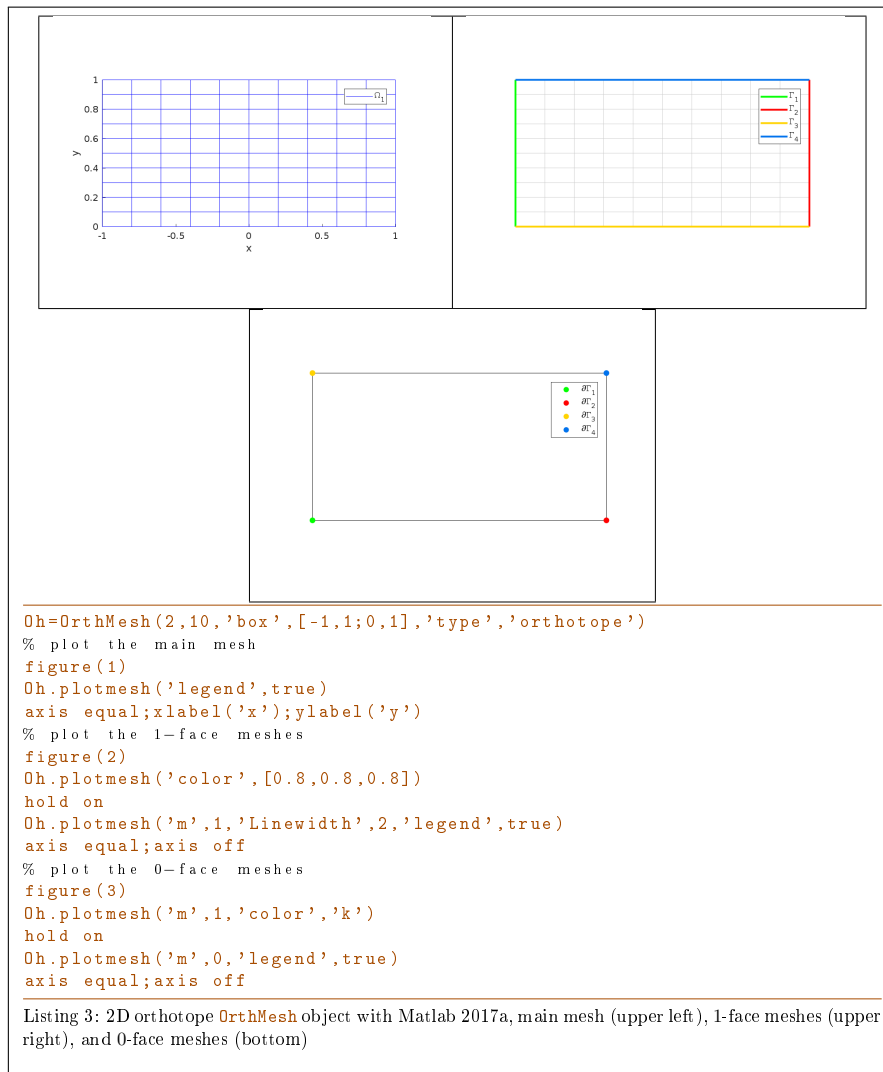
3.3 3d-orthotope meshing by simplices

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



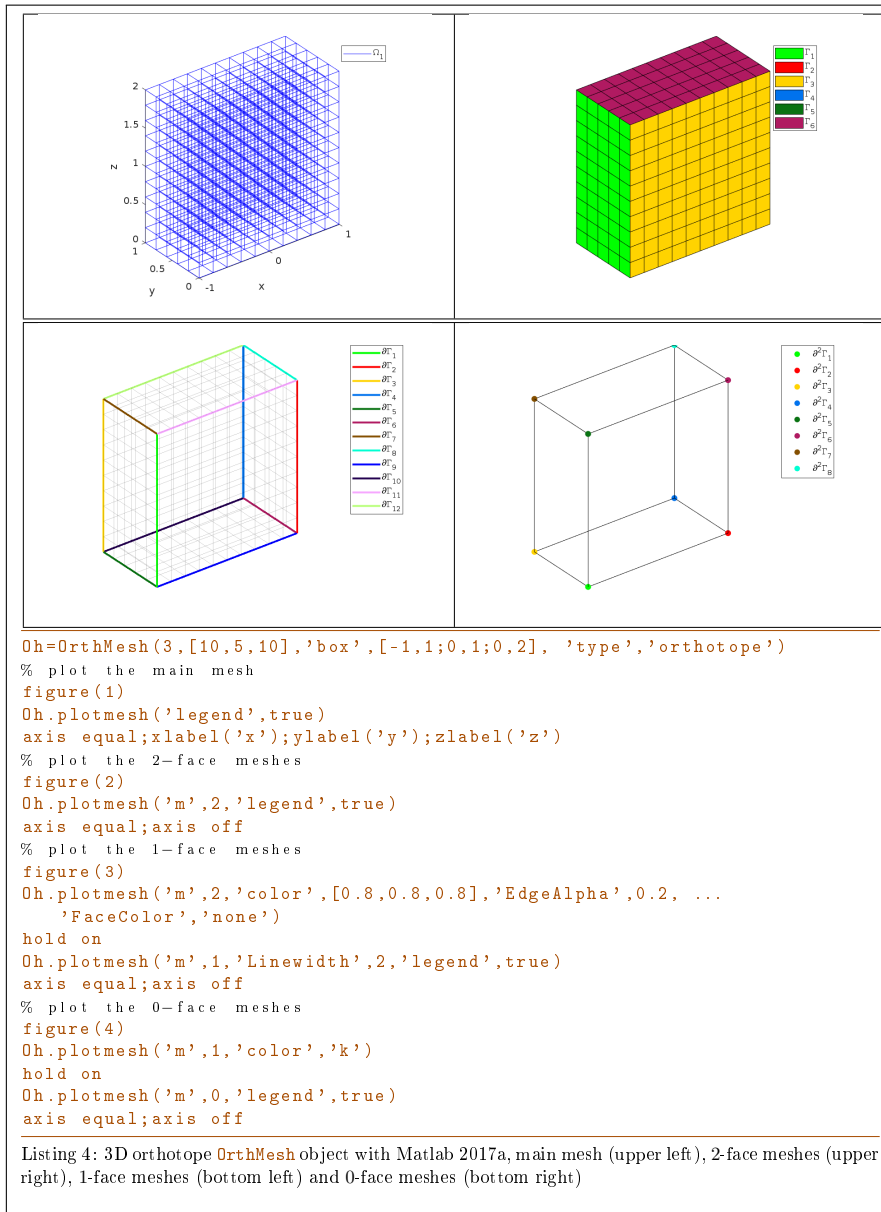
3.4 2d-orthotope meshing by orthotopes

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



3.5 3d-orthotope meshing by orthotopes

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

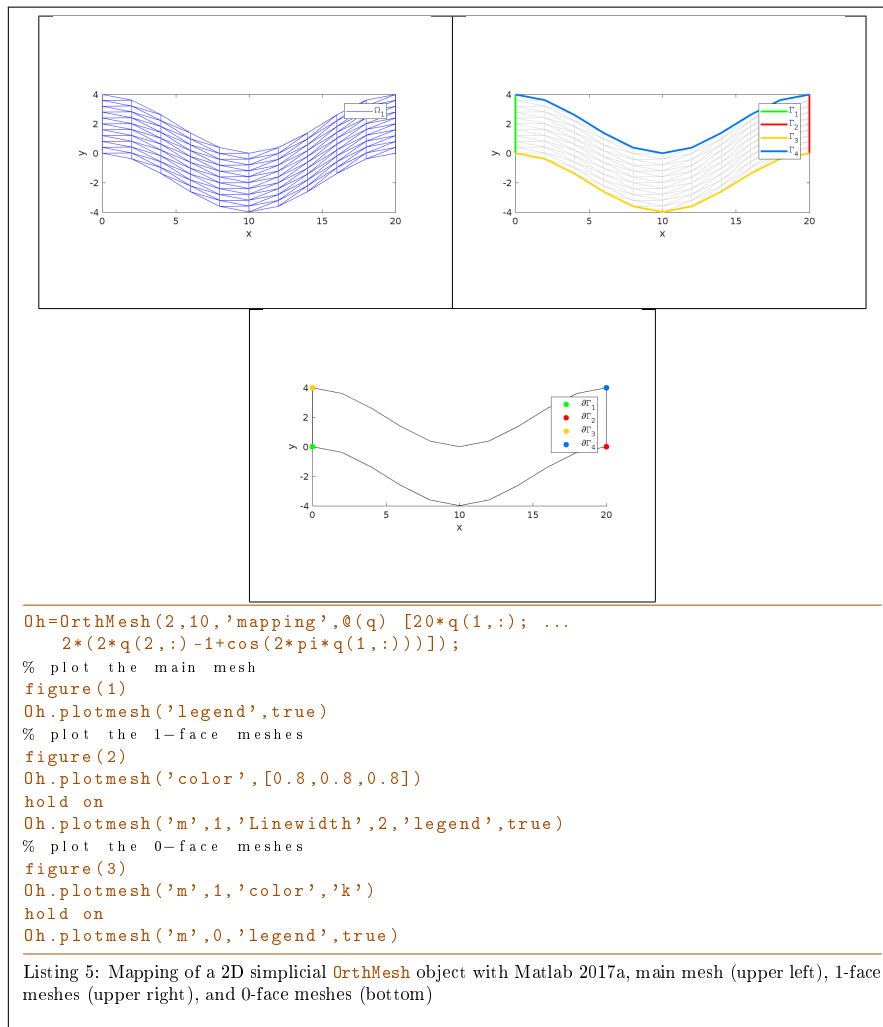


3.6 Mapping of a 2d-orthotope meshing by simplices

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

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

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

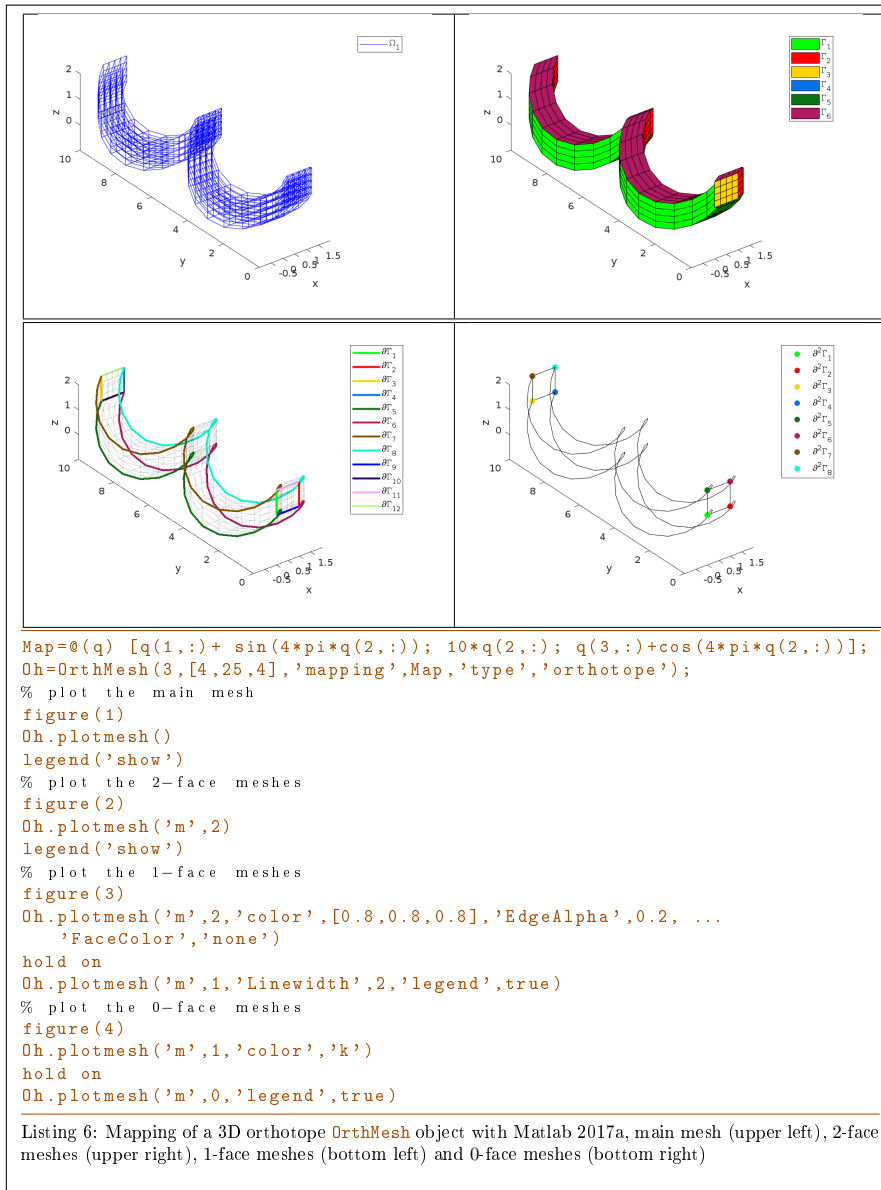


3.7 3d-orthotope meshing by orthotopes

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

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

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \longrightarrow F(x, y, y) = \begin{pmatrix} x + \sin(4\pi y) \\ 10y \\ z + \cos(4\pi y) \end{pmatrix}$$



4 Benchmarking

4.1 `fc_bench.bench01` function

The `fc_bench.bench01` function can be used to obtain computational times of the `OrthMesh` constructor.

Syntaxe

```
fc_bench.bench01(d, ctype, Box, LN)
```

Description

```
fc_bench.bench01(d, ctype, Box, LN)
```

displays computational times of the `OrthMesh` constructor as follows

```
ts=tic(); Oh=OrthMesh(d,N,'box',Box,'type',ctype); tcpu=toc(ts);
```

for each `N` in `LN`.

4.2 Examples

Listing 7: : Computational times of `OrthMesh` constructor in dimension `d=3` (simplicial mesh)

```
fc_hypermesh.bench01(3,'simplicial',[-1 1;-1 1;-1 1],25:25:175)
```

Output

```
# BENCH in dimension 3 with simplicial mesh
#d: 3
#type: simplicial
#box: [-1 1; -1 1; -1 1]
#desc: N      nq      nme  time(s)
      25    17576    93750  0.489
      50   132651   750000  0.171
      75   438976  2531250  0.307
     100  1030301  6000000  0.552
     125  2000376 11718750  0.984
     150  3442951 20250000  1.569
     175  5451776 32156250  2.417
```

Listing 8: : Computational times of `OrthMesh` constructor in dimension `d=5` (orthotope mesh)

```
fc_hypermesh.bench01(5,'orthotope',[-1 1;-1 1;-1 1;-1 1;-1 ...
1],[5:5:25,27])
```

Output

```
# BENCH in dimension 5 with orthotope mesh
#d: 5
#type: orthotope
#box: [-1 1; -1 1; -1 1; -1 1; -1 1]
#desc: N      nq      nme  time(s)
      5     7776     3125  0.637
     10    161051    100000  0.357
     15   1048576   759375  0.633
     20   4084101  3200000  1.622
     25  11881376  9765625  4.276
     27  17210368 14348907  6.347
```

References

- [1] François Cuvelier and Gilles Scarella. Vectorized algorithms for regular tessellations of d -orthotopes and their faces. *HAL archives ouvertes*, November 2017. preprint.