



Documentation of the FC-OOGMESH Octave package version 0.0.10*

François Cuvelier[†]

December 22, 2016

Abstract

This Octave package make it possible to generate mesh files from *.geo* file by using `gmsh`. It's also possible with the `ooGmsh` class to read the mesh file and to store its contains in more user-friendly form. This toolbox must be regarded as a very simple interface between gmsh files and Octave. So you are free to create any data structures or objects you want from an `ooGmsh` object.

Contents

1	Introduction	2
2	gmsh installation (Linux 64bit)	2
3	Installation of the FC-OOGMESH package	3
3.1	Automatic installation, all in one (recommanded)	3
4	gmsh interface	4
4.1	function <code>gmsh.buildmesh2d</code>	4
4.2	function <code>gmsh.buildmesh3d</code>	6
4.3	function <code>gmsh.buildmesh3ds</code>	7
4.4	function <code>gmsh.buildpartmesh2d</code>	7
4.5	function <code>gmsh.buildpartmesh3d</code>	8

*Compiled with Octave 4.2.0

[†]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 partially supported by ANR Dedales.

4.6	function <code>gmsh.buildpartmesh3ds</code>	8
4.7	function <code>gmsh.buildPartRectangle</code>	8
5	<code>ooGmsh</code> class	10
5.1	Sample 1	13
5.2	Sample 2	13
5.3	Sample 3	14

1 Introduction

The FC-OOGMSH Octave package is closely related to `gmsh`, see [2] or [1], which is a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. `gmsh` can also build two-dimensional meshes and three-dimensional surface meshes. This package was initially created to make it possible from Octave to rapidly

- generate mesh file from `.geo` file by using `gmsh`
- efficiently read this mesh file and store its contents in `ooGmsh` Octave object easy to manipulate.

The `ooGmsh` Octave object can be used to create, from a `.msh` file, any data structures or objects needed by your project. For example, the `fc-simesh` Octave package uses this package to create the `siMesh` object containing all the simplices elements of the mesh.

In a first step we quickly present the installation of `gmsh` on Linux (for Windows and Mac OS X precompiled software are also provided on <http://gmsh.info>). Then, we explain how to configure the FC-OOGMSH package for using `gmsh`. Finally, we describe the FC-OOGMSH's functions which use `gmsh` to create mesh files.

2 gmsh installation (Linux 64bit)

Binaries for Linux 64 bits are available on <http://gmsh.info>.

- To obtain the current stable release (version 2.15.0, December 4 2016) directly from a terminal one can run

```
Terminal
wget http://gmsh.info/bin/Linux/gmsh-2.15.0-
Linux64.tgz
```

- To obtain the development version (automated nightly snapshots) directly from a terminal one can run

```
Terminal
wget http://gmsh.info/bin/Linux/gmsh-svn-Linux64.
tgz
```

To install the version 2.15.0 on directory `~/software/GMSH`, one can do the commands

Terminal

```
mkdir -p ~/software/GMSH
cd ~/software/GMSH
wget http://gmsh.info/bin/Linux/gmsh-2.15.0-Linux64.tgz
tar zxvf gmsh-2.15.0-Linux64.tgz
```

To run `gmsh` we can use the following command in a terminal

Terminal

```
~/software/GMSH/gmsh-2.15.0-Linux/bin/gmsh &
```

3 Installation of the `fc-oogmsh` package

3.1 Automatic installation, all in one (recommended)

For this method, one just has to get/download the install file `ofc_oogmsh_install.m` and run it under Octave. This command downloads, extracts and configures the *fc-oogmsh* and the required *fc-tools* package in the current directory. By default, the `gmsh` binary is supposed to be located in `/usr/bin/gmsh` and to specify another location one can do

```
mfc_oogmsh_install('gmsh_bin', '<PATH>/gmsh')
```

where `<PATH>` is the path of the `gmsh` binary.

One also can change the location of the `gmsh` binary after the installation by using

```
fc_oogmsh.configure('gmsh_bin', '<PATH>/gmsh')
```

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

Terminal

```
mkdir -p ~/Octave/packages
cd ~/Octave/packages
wget http://www.math.univ-paris13.fr/~cuvelier/
software/codes/fc-oogmsh/0.0.10/ofc_oogmsh_install.
m
```

Then in an Octave terminal run the following commands



GNU Octave

```
>> cd ~/Matlab/toolboxes
>> mfc_oogmsh_install
```

There is the output of the `mfc_oogmsh_install` command:

```
1- Downloading and extracting the toolboxes
-> <fc-tools>[0.0.14] ... OK
-> <fc-oogmsh>[0.0.10] ... OK
2- Setting the <fc-oogmsh> toolbox
Write in ~/Matlab/toolboxes/fc-oogmsh-full/mfc-oogmsh-0.0.10/configure_loc.m ...
-> done
3- Using the <fc-oogmsh> toolbox
Under Matlab:
    addpath('~/Matlab/toolboxes/fc-oogmsh-full/mfc-oogmsh-0.0.10')
    fc_oogmsh.init()
```

See `~/Matlab/toolboxes/mfc_oogmsh_install.log`

The complete toolbox (i.e. with the **FC-TOOLS** toolbox included) is stored in the directory `~/Matlab/toolboxes/fc-oogmsh-full` and, for each Matlab session, one have to set the toolbox by:

 **GNU Octave**

```
>> addpath('~/Matlab/toolboxes/fc-oogmsh-full/mfc-oogmsh-0.0.10')
>> fc_oogmsh.init()
```

For **uninstalling**, one just have to delete directory `~/Matlab/toolboxes/fc-oogmsh-full`

4 gmsh interface

All the functions provided in this section use **gmsh** to create a mesh file from a **gmsh** geometry script file (extension `.geo`).

4.1 function `gmsh.buildmesh2d`

This function uses **gmsh** and a `.geo` file (describing a 2D-geometry) to generate a 2D-mesh.

Syntaxe

```
meshfile=gmsh.buildmesh2d(geofile,N)

meshfile=gmsh.buildmesh2d(geofile,N,Name,Value)
```

Description

`meshfile=gmsh.buildmesh2d(geofile,N)` create a 2D-mesh using **gmsh** and the `geo` file `geofile` (without path). The integer `N` has two functions : numbering the name of the generated mesh as `<geofile without extension and`

path> + <-N.msh> and passing this number to **gmsh** via the option "-setnumber N <N>". Usually we used this parameter in **gmsh** to set the prescribed mesh element size at the points. (see given *geo* files)
 As output return a file name (with full path) corresponding to the mesh generated by **gmsh**.

meshfile=gmsh.buildmesh2d(geofile,N,Name,Value, ...) specifies function options using one or more Name,Value pair arguments. The Name options can be

- 'geodir' : to specify the directory of the *geo* file geofile,
- 'meshdir' : to specify the directory where the mesh file will be written,
- 'meshfile' : to specify the name of the mesh file (with path and .msh extension),
- 'check' : to perform various consistency checks on mesh with **gmsh**, if Value is true. (default : false)
- 'force' : to force meshing even if the mesh file already exists if Value is true (default : false)
- 'verbose' : to specify the degree of verbosity (0, silence; 2, default; ...)
- 'strings' : cells array of strings corresponding to **gmsh** options given with -string "... (default empty) (see **gmsh** documentation)

Examples All the following examples use the *.geo* file **condenser11.geo** which is in the directory **geodir** of the toolbox.

Matlab commands with output

```
disp('****_gmsh.buildmesh2d_:1st_call')
meshfile=gmsh.buildmesh2d('condenser11',25,'force',true);
disp('****_gmsh.buildmesh2d_:2nd_call')
meshfile=gmsh.buildmesh2d('condenser11',25);
```

```
**** gmsh.buildmesh2d : 1st call
[fc-oogmsh] Input file : <fc-oogmsh>/geodir/condenser11.geo
[fc-oogmsh] Overwriting mesh file <fc-oogmsh>/meshes/condenser11-25.msh
[fc-oogmsh] Starting building mesh <fc-oogmsh>/meshes/condenser11-25.msh with gmsh 2.13.2
[fc-oogmsh] Using command : gmsh -2 -setnumber N 25 <fc-oogmsh>/geodir/condenser11.geo -o ...
                        <fc-oogmsh>/meshes/condenser11-25.msh
Be patient ...
**** gmsh.buildmesh2d : 2nd call
[fc-oogmsh] Input file : <fc-oogmsh>/geodir/condenser11.geo
[fc-oogmsh] Mesh file <fc-oogmsh>/meshes/condenser11-25.msh already exists.
-> Use "force" flag to rebuild if needed.
```

Matlab commands with output

```
meshfile=gmsl.buildmesh2d('condenser11',25,'force',true, ...
    'verbose',4, 'strings',{ 'Mesh.Algorithm=1;', ...
    'Mesh.ScalingFactor=2;'});
```

```
[fc-oogmsh] Input file : <fc-oogmsh>/geodir/condenser11.geo
[fc-oogmsh] Overwriting mesh file <fc-oogmsh>/meshes/condenser11-25.msh
[fc-oogmsh] Starting building mesh <fc-oogmsh>/meshes/condenser11-25.msh with gmsh 2.13.2
[fc-oogmsh] Using command : gmsh -2 -setnumber N 25 -string "Mesh.Algorithm=1;Mesh.ScalingFactor=2;" ...
    <fc-oogmsh>/geodir/condenser11.geo -o <fc-oogmsh>/meshes/condenser11-25.msh
Be patient...
[fc-oogmsh] gmsh output :
Info : Running '/usr/local/GMSH/gmsh-2.13.2-Linux/bin/gmsh -2 -setnumber N 25 -string ...
    Mesh.Algorithm=1;Mesh.ScalingFactor=2; <fc-oogmsh>/geodir/condenser11.geo -o ...
    <fc-oogmsh>/meshes/condenser11-25.msh' [Gmsh 2.13.2, 1 node, max. 1 thread]
Info : Started on Thu Dec 22 12:57:36 2016
Info : Reading '<fc-oogmsh>/geodir/condenser11.geo'...
Info : Reading '<fc-oogmsh>/geodir/options01_data.geo'...
Info : Done reading '<fc-oogmsh>/geodir/options01_data.geo'
Info : Reading '<fc-oogmsh>/geodir/shape_functions.geo'...
Info : Done reading '<fc-oogmsh>/geodir/shape_functions.geo'
Info : Removing duplicate mesh vertices...
Info : Found 0 duplicate vertices
Info : No duplicate vertices found
Info : Done reading '<fc-oogmsh>/geodir/condenser11.geo'
Info : Finalized high order topology of periodic connections
Info : Meshing 1D...
Info : Meshing curve 101 (Line)
Info : Meshing curve 102 (Line)
Info : Meshing curve 103 (Line)
Info : Meshing curve 104 (Line)
Info : Meshing curve 106 (Circle)
Info : Meshing curve 107 (Circle)
Info : Meshing curve 108 (Circle)
Info : Meshing curve 109 (Circle)
Info : Meshing curve 111 (Circle)
Info : Meshing curve 112 (Circle)
Info : Meshing curve 113 (Circle)
Info : Meshing curve 114 (Circle)
Info : Meshing curve 116 (Circle)
Info : Meshing curve 117 (Circle)
Info : Meshing curve 118 (Circle)
Info : Meshing curve 119 (Circle)
Info : Meshing curve 121 (Circle)
Info : Meshing curve 122 (Circle)
Info : Meshing curve 123 (Circle)
Info : Meshing curve 124 (Circle)
Info : Meshing curve 126 (Circle)
Info : Meshing curve 127 (Circle)
Info : Meshing curve 128 (Circle)
Info : Meshing curve 129 (Circle)
Info : Meshing curve 131 (Circle)
Info : Meshing curve 132 (Circle)
Info : Meshing curve 133 (Circle)
Info : Meshing curve 134 (Circle)
Info : Meshing curve 136 (Circle)
Info : Meshing curve 137 (Circle)
Info : Meshing curve 138 (Circle)
Info : Meshing curve 139 (Circle)
Info : Meshing curve 141 (Circle)
Info : Meshing curve 142 (Circle)
Info : Meshing curve 143 (Circle)
Info : Meshing curve 144 (Circle)
Info : Meshing curve 146 (Circle)
Info : Meshing curve 147 (Circle)
Info : Meshing curve 148 (Circle)
Info : Meshing curve 149 (Circle)
Info : Done meshing 1D (0.015988 s)
Info : Meshing 2D...
Info : Meshing surface 105 (Plane, MeshAdapt)
Info : Meshing surface 110 (Plane, MeshAdapt)
Info : Meshing surface 120 (Plane, MeshAdapt)
Info : Meshing surface 130 (Plane, MeshAdapt)
Info : Meshing surface 140 (Plane, MeshAdapt)
Info : Meshing surface 150 (Plane, MeshAdapt)
Info : Done meshing 2D (0.340838 s)
Info : 2999 vertices 6131 elements
Info : Writing '<fc-oogmsh>/meshes/condenser11-25.msh'...
Info : Done writing '<fc-oogmsh>/meshes/condenser11-25.msh'
Info : Stopped on Thu Dec 22 12:57:37 2016
```

4.2 function gmsl.buildmesh3d

This function uses `gmsh` and a `.geo` file (describing a 3D-geometry) to generate a 3D-mesh. See function `gmsl.buildmesh2d` for usage and options.

4.3 function gmsh.buildmesh3ds

This function uses `gmsh` and a `.geo` file (describing a 3D surface geometry or a 3D-geometry) to generate a 3D surface mesh. See function `gmsh.buildmesh2d` for usage and options.

4.4 function gmsh.buildpartmesh2d

This function uses `gmsh` and a `.msh` file (containing of a 2D-mesh) to generate a 2D partioned mesh.

Syntaxe

```
partmeshfile=gmsh.buildpartmesh2d(meshfile,np)
partmeshfile=gmsh.buildpartmesh2d(meshfile,np,Name,Value)
```

Description

`partmeshfile=gmsh.buildpartmesh2d(meshfile,np)` create a 2D partitioned mesh using `gmsh` and the `.msh` file `meshfile` (with path). The integer `np` is the number of partitions.

As output return a file name (with full path) corresponding to the partitioned mesh generated by `gmsh`. The output file name is construct as following : `<meshfile without extension>-part<np>.msh`

`partmeshfile=gmsh.buildpartmesh2d(meshfile,np,Name,Value, ...)` specifies function options using one or more `Name,Value` pair arguments. The `Name` options can be

- `'savedir'` : to specify the directory where the partitioned mesh file will be written,
- `'check'` : to perform various consistency checks on mesh with `gmsh`, if `Value` is true. (default : false)
- `'force'` : to force meshing even if the mesh file already exists if `Value` is true (default : false)
- `'verbose'` : to specify the degree of verbosity (0, silence; 2, default; ...)
- `'strings'` : cells array of strings corresponding to `gmsh` options given with `-string "..."` (default empty) (see `gmsh` documentation)

Examples All the following examples use the `meshfile` as output of the command :

```
meshfile=gmsh.buildmesh2d('condenser11',25);
```

Matlab commands with output

```
meshfile=gmsh.buildmesh2d('condenser11',25,'verbose',0);  
pmlfile=gmsh.buildpartmesh2d(meshfile,5,'force',true);
```

```
[fc-oogmsh] Input file : <fc-oogmsh>/meshes/condenser11-25.msh  
[fc-oogmsh] Overwriting mesh file <fc-oogmsh>/meshes/condenser11-25-part5.msh  
[fc-oogmsh] Starting building mesh <fc-oogmsh>/meshes/condenser11-25-part5.msh with gmsh 2.13.2  
[fc-oogmsh] Using command : gmsh -2 -saveall -part 5 <fc-oogmsh>/meshes/condenser11-25.msh -o ...  
                        <fc-oogmsh>/meshes/condenser11-25-part5.msh  
Be patient...
```

Matlab commands with output

```
meshfile=gmsh.buildmesh2d('condenser11',25,'verbose',0);  
pmlfile=gmsh.buildpartmesh2d(meshfile,5,'force',true, ...  
    'verbose',4, 'strings',{'Mesh.Partitioner=2;', ...  
    'Mesh.MetisAlgorithm=3;'});
```

```
[fc-oogmsh] Input file : <fc-oogmsh>/meshes/condenser11-25.msh  
[fc-oogmsh] Overwriting mesh file <fc-oogmsh>/meshes/condenser11-25-part5.msh  
[fc-oogmsh] Starting building mesh <fc-oogmsh>/meshes/condenser11-25-part5.msh with gmsh 2.13.2  
[fc-oogmsh] Using command : gmsh -2 -saveall -part 5 -string "Mesh.Partitioner=2;Mesh.MetisAlgorithm=3;" ...  
                        <fc-oogmsh>/meshes/condenser11-25.msh -o <fc-oogmsh>/meshes/condenser11-25-part5.msh  
Be patient...  
[fc-oogmsh] gmsh output :  
Info : Running '/usr/local/GMSH/gmsh-2.13.2-Linux/bin/gmsh -2 -saveall -part 5 -string ...  
      Mesh.Partitioner=2;Mesh.MetisAlgorithm=3; <fc-oogmsh>/meshes/condenser11-25.msh -o ...  
      <fc-oogmsh>/meshes/condenser11-25-part5.msh' [Gmsh 2.13.2, 1 node, max. 1 thread]  
Info : Started on Thu Dec 22 12:57:38 2016  
Info : Reading '<fc-oogmsh>/meshes/condenser11-25.msh'...  
Info : 2990 vertices  
Info : 6082 elements  
Info : Done reading '<fc-oogmsh>/meshes/condenser11-25.msh'  
Info : Finalized high order topology of periodic connections  
Info : Meshing 1D...  
Info : Done meshing 1D (1.9e-05 s)  
Info : Meshing 2D...  
Info : Done meshing 2D (2.69413e-05 s)  
Info : 2990 vertices 6082 elements  
Info : Building graph...  
Info : Partitioning graph...  
Info : Launching METIS graph partitioner  
METIS with weights  
Info : Number of Edges Cut : 138  
Info : Done partitioning graph  
Info : Writing '<fc-oogmsh>/meshes/condenser11-25-part5.msh'...  
Info : Done writing '<fc-oogmsh>/meshes/condenser11-25-part5.msh'  
Info : Stopped on Thu Dec 22 12:57:38 2016
```

4.5 function gmsh.buildpartmesh3d

This function uses **gmsh** and a *.msh* file (containing of a 3D-mesh) to generate a 3D partioned mesh.

4.6 function gmsh.buildpartmesh3ds

This function uses **gmsh** and a *.msh* file (containing of a 3D surface mesh) to generate a 3D partioned surface mesh.

4.7 function gmsh.buildPartRectangle

This function uses **gmsh** and the *geodir/rectanglepart.geo* file to generate a 2D regular partioned mesh of the rectangle $[0, Lx] \times [0, Ly]$ with $Nx \times Ny$ partitions.

Syntaxe

```
meshfile=gmsb.buildpartrectangle(Lx,Ly,Nx,Ny,N)

meshfile=gmsb.buildpartrectangle(Lx,Ly,Nx,Ny,N, ...
    Name,Value)
```

Description

`meshfile=gmsb.buildpartrectangle(Lx,Ly,Nx,Ny,N)` create a 2D regular partitioned mesh using `gmsb` of the rectangle $[0, Lx] \times [0, Ly]$ with $Nx \times Ny$ partitions. The `N` parameter is passed to `gmsb` to set the prescribed mesh element size at the points

As output return a file name (with full path) corresponding to the partitioned mesh generated by `gmsb`. The default output file name is construct as following : `rectanglepart-Lx%.3f-Ly%.3f-Nx%d-Ny%d-N%d.msh`

`meshfile=gmsb.buildpartrectangle(Lx,Ly,Nx,Ny,N,Name,Value, ...)` specifies function options using one or more `Name,Value` pair arguments. The `Name` options can be

- `'meshdir'` : to specify the directory where the partitioned mesh file will be written,
- `'meshfile'` : to specify the mesh file name with `.msh` extension. Without path, the file is written in `<meshdir>` directory.
- `'check'` : to perform various consistency checks on mesh with `gmsb`, if `Value` is true. (default : false)
- `'force'` : to force meshing even if the mesh file already exists if `Value` is true (default : false)
- `'verbose'` : to specify the degree of verbosity (0, silence; 2, default; ...)
- `'strings'` : cells array of strings corresponding to `gmsb` options given with `-string "..."` (default empty) (see `gmsb` documentation)

Examples All the following examples ...

Matlab commands with output

```
pmfile=gmsb.buildpartrectangle(1,1,3,2,100,'force',true);
```

```
[fc-oogmsb] Input file : <fc-oogmsb>/geodir/rectanglepart.geo
[fc-oogmsb] Overwriting mesh file <fc-oogmsb>/meshes/rectanglepart-Lx1.000-Ly1.000-Nx3-Ny2-N100.msh
[fc-oogmsb] Starting building mesh <fc-oogmsb>/meshes/rectanglepart-Lx1.000-Ly1.000-Nx3-Ny2-N100.msh with gmsb 2.13.2
[fc-oogmsb] Using command : gmsb -2 -setnumber N 100 -setnumber NX 3 -setnumber NY 2 -setnumber LX 1 -setnumber LY 1 ...
<fc-oogmsb>/geodir/rectanglepart.geo -o <fc-oogmsb>/meshes/rectanglepart-Lx1.000-Ly1.000-Nx3-Ny2-N100.msh
Be patient...
```

Matlab commands with output

```
pmfile=gmshtool.buildpartrectangle(1,1,3,2,100,'verbose',4,...
    'force',true,'meshfile','./toto.msh');
```

```
[fc-oogmsh] Input file : <fc-oogmsh>/geodir/rectanglepart.geo
[fc-oogmsh] Overwriting mesh file ./toto.msh
[fc-oogmsh] Starting building mesh ./toto.msh with gmshtool 2.13.2
[fc-oogmsh] Using command : gmshtool -2 -setnumber N 100 -setnumber NX 3 -setnumber NY 2 -setnumber LX 1 -setnumber LY 1 ...
    <fc-oogmsh>/geodir/rectanglepart.geo -o ./toto.msh
Be patient...
[fc-oogmsh] gmshtool output :
Info : Running '/usr/local/GMSH/gmsh-2.13.2-Linux/bin/gmsh -2 -setnumber N 100 -setnumber NX 3 -setnumber NY 2 ...
    -setnumber LX 1 -setnumber LY 1 <fc-oogmsh>/geodir/rectanglepart.geo -o ./toto.msh' [Gmsh 2.13.2, 1 node, max. ...
    1 thread]
Info : Started on Thu Dec 22 12:57:39 2016
Info : Reading '<fc-oogmsh>/geodir/rectanglepart.geo'...
Info : Reading '<fc-oogmsh>/geodir/partitions01.data.geo'...
Info : Done reading '<fc-oogmsh>/geodir/partitions01.data.geo'
Info : Reading '<fc-oogmsh>/geodir/partitions_shape.geo'...
Info : Done reading '<fc-oogmsh>/geodir/partitions_shape.geo'
Info : Done reading '<fc-oogmsh>/geodir/rectanglepart.geo'
Info : Finalized high order topology of periodic connections
Info : Meshing 1D...
Info : Meshing curve 1 (Line)
Info : Meshing curve 2 (Line)
Info : Meshing curve 3 (Line)
Info : Meshing curve 4 (Line)
Info : Meshing curve 5 (Line)
Info : Meshing curve 6 (Line)
Info : Meshing curve 7 (Line)
Info : Meshing curve 8 (Line)
Info : Meshing curve 9 (Line)
Info : Meshing curve 10 (Line)
Info : Meshing curve 11 (Line)
Info : Meshing curve 12 (Line)
Info : Meshing curve 13 (Line)
Info : Meshing curve 14 (Line)
Info : Meshing curve 15 (Line)
Info : Meshing curve 16 (Line)
Info : Meshing curve 17 (Line)
Info : Done meshing 1D (0.002211 s)
Info : Meshing 2D...
Info : Meshing surface 19 (Plane, Delaunay)
Info : Meshing surface 21 (Plane, Delaunay)
Info : Meshing surface 23 (Plane, Delaunay)
Info : Meshing surface 25 (Plane, Delaunay)
Info : Meshing surface 27 (Plane, Delaunay)
Info : Meshing surface 29 (Plane, Delaunay)
Info : Done meshing 2D (0.47318 s)
Info : 13685 vertices 27682 elements
Info : Writing './toto.msh'...
Info : Done writing './toto.msh'
Info : Stopped on Thu Dec 22 12:57:40 2016
```

5

ooGmsh class

The **ooGMSH** class can be used to read **gmsh** mesh files with the MSH ASCII file format described for example in [1], section 9.1.

In a **.msh** file the kind of mesh elements are identified by their *elm-type* integer values :

<i>elm-type</i>	description
1	2-node line
2	3-node triangle
3	4-node quadrangle
4	4-node tetrahedron
5	8-node hexahedron
6	6-node prism
7	5-node pyramid
8	3-node second order line (2 nodes associated with the vertices and 1 with the edge)

9	6-node second order triangle (3 nodes associated with the vertices and 3 with the edges)
10	9-node second order quadrangle (4 nodes associated with the vertices, 4 with the edges and 1 with the face)
11	10-node second order tetrahedron (4 nodes associated with the vertices and 6 with the edges)
12	27-node second order hexahedron (8 nodes associated with the vertices, 12 with the edges, 6 with the faces and 1 with the volume)
13	18-node second order prism (6 nodes associated with the vertices, 9 with the edges and 3 with the quadrangular faces)
14	14-node second order pyramid (5 nodes associated with the vertices, 8 with the edges and 1 with the quadrangular face)
15	1-node point
16	8-node second order quadrangle (4 nodes associated with the vertices and 4 with the edges)
17	20-node second order hexahedron (8 nodes associated with the vertices and 12 with the edges)
18	15-node second order prism (6 nodes associated with the vertices and 9 with the edges)
19	13-node second order pyramid (5 nodes associated with the vertices and 8 with the edges)
20	9-node third order incomplete triangle (3 nodes associated with the vertices, 6 with the edges)
21	10-node third order triangle (3 nodes associated with the vertices, 6 with the edges, 1 with the face)
22	12-node fourth order incomplete triangle (3 nodes associated with the vertices, 9 with the edges)
23	15-node fourth order triangle (3 nodes associated with the vertices, 9 with the edges, 3 with the face)
24	15-node fifth order incomplete triangle (3 nodes associated with the vertices, 12 with the edges)
25	21-node fifth order complete triangle (3 nodes associated with the vertices, 12 with the edges, 6 with the face)
26	4-node third order edge (2 nodes associated with the vertices, 2 internal to the edge)
27	5-node fourth order edge (2 nodes associated with the vertices, 3 internal to the edge)
28	6-node fifth order edge (2 nodes associated with the vertices, 4 internal to the edge)
29	20-node third order tetrahedron (4 nodes associated with the vertices, 12 with the edges, 4 with the faces)
30	35-node fourth order tetrahedron (4 nodes associated with the vertices, 18 with the edges, 12 with the faces, 1 in the volume)
31	56-node fifth order tetrahedron (4 nodes associated with the vertices, 24 with the edges, 24 with the faces, 4 in the volume)

92	64-node third order hexahedron (8 nodes associated with the vertices, 24 with the edges, 24 with the faces, 8 in the volume)
93	125-node fourth order hexahedron (8 nodes associated with the vertices, 36 with the edges, 54 with the faces, 27 in the volume)

When reading a .msh file generated by `gmsh`, we split the mesh elements by *elm-type* and generate an array of `Elmt` structure. The dimension of this array is the number of different *elm-type* founds on the .msh file. The `Elmt` structure is given by

 Fields of Elmt structure	
<code>type</code>	: integer refers to the type of the element : 1 for 2-node line, 2 for 3-node triangle, ... See the <i>elm-type</i> description of [1], section 9.1.
<code>geo</code>	: string contains the kind of geometry: 'line', 'triangle', 'tetrahedron', ...
<code>d</code>	: integer space dimension or <i>d</i> -simplex.
<code>order</code>	: integer order of the element
<code>n_{me}</code>	: integer number of mesh elements
<code>me</code>	: array of <i>d</i> + 1-by- <i>n_{me}</i> integers connectivity array
<code>phys_lab</code>	: array of <i>n_{me}</i> -by-... integers physical labels of the elements
<code>geo_lab</code>	: array of <i>n_{me}</i> -by-... integers geometrical labels of the elements
<code>nb_parts</code>	: array of <i>n_{me}</i> -by-1 integers number of mesh partitions to which the element belongs
<code>part_lab</code>	: array of <i>n_{me}</i> -by-max(<i>nb_parts</i>) integers part_lab(<i>i</i> , 1 : <i>nb_parts</i> (<i>i</i>)) contains all the partitions index to which the <i>i</i> -th element belongs.

The `ooGmsh` class was created to store a maximum of(all the) information(s) contained in the .msh file. The properties of this class are:



Properties of **ooGmsh** class

dim	:	integer space dimension
n _q	:	integer number of vertices/nodes
q	:	dim-by-n _q array of reals array of vertex coordinates
types	:	array of integers List of the element types found in the mesh file.
orders	:	array of integers List of the orders of the element types found in the mesh file.
sElts	:	array of Elmt structure One Elmt structure by element type, such that sElts(<i>i</i>) contains all the elements of type types(<i>i</i>) and order orders(<i>i</i>).

The **ooGmsh** class have only one constructor :

```
Gh=ooGmsh(meshfile)
```

where meshfile is the name of ... a mesh file

5.1 Sample 1

The 2d .geo file *condenser.geo* is used to create a .msh file : **condenser-25.msh**. This .msh file contains only 1 (2-node line) and 2 (3-node triangle) *elm-type*.

Matlab commands with output

```
meshfile=gmesh.buildmesh('condenser',25,'verbose',0);
Gh = ooGmsh(meshfile)
```

Gh =

```
ooGmsh with properties:
    dim: 2 double
    nq: 55670 double
    orders: 1 double
    partitionnedfile: 0 logical
    q: (2x55670 double)
    sElts: (2x1 struct)
    toGlobal: (1x55670 double)
    types: [ 1 2 ] (1x2 double)
```

5.2 Sample 2

The 3d .geo file *cylinderkey.geo* is used to create a .msh file : **cylinderkey-10.msh**. This .msh file contains 1 (2-node line), 2 (3-node triangle) and 4 (4-node tetrahedron) *elm-type*.

Matlab commands with output

```
meshfile=gmshtool.buildmesh3d('cylinderkey',10,'verbose',0,'force',true);  
Gh = ooGmsh(meshfile)
```

```
Gh =  
  
ooGmsh with properties:  
    dim: 3 double  
    nq: 5296 double  
    orders: 1 double  
    partitionedfile: 0 logical  
    q: (3x5296 double)  
    sElts: (3x1 struct)  
    toGlobal: (1x5296 double)  
    types: [ 1 2 4 ] (1x3 double)
```

5.3 Sample 3

The 3d .geo file *ball8.geo* is used to create a 3d surface .msh file : *ball8-50.msh*. This .msh file contains 1 (2-node line), 2 (3-node triangle) and 15 (1-node point) *elm-type*.

Matlab commands with output

```
meshfile=gmshtool.buildmesh3ds('ball8',50,'verbose',0,'force',true);  
Gh = ooGmsh(meshfile)
```

```
Gh =  
  
ooGmsh with properties:  
    dim: 3 double  
    nq: 37404 double  
    orders: 1 double  
    partitionedfile: 0 logical  
    q: (3x37404 double)  
    sElts: (3x1 struct)  
    toGlobal: (1x37404 double)  
    types: [ 1 2 15 ] (1x3 double)
```

References

- [1] Gmsh 2.15.0. <http://gmsh.info>, 2016.
- [2] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.