# Install and use pymedit to create standard meshes from level-set functions with FEniCS

Michel Duprez, Vanessa Lleras, Alexei Lozinski,
Vincent Vigon & Killian Vuillemot

June 2023

The 3 first parts are taken from :

`https://github.com/MmgTools/Mmg-Day-2018_TP/tree/master`

## 1 Install mmg

- Clone the repo and build the application :

```
$ git clone https://github.com/MmgTools/mmg.git
$ cd mmg
$ mkdir build
$ cd build
$ cmake ..
$ sudo make
$ sudo make install
```

- If you want to add to your path, in your bashrc add :

```
PATH=$PATH:/home/PATH_TO_YOUR_INSTALL/mmg/build/bin
```

## 2 Install Medit

- Clone the repo and build the application :

```
$ git clone https://github.com/ISCDtoolbox/Medit.git
$ cd Medit
$ mkdir build
$ cd build
$ sudo cmake ..
$ sudo make
$ sudo make install
```

- If you want to add to your path, in your bashrc add :

```
PATH=$PATH:/home/PATH_TO_YOUR_INSTALL/Medit/build
```

## 3 Some graphic packages for linux :

```
$ sudo apt-get install -y freeglut3-dev
```

```
$ sudo apt-get install -y libxi-dev
```

```
$ sudo apt-get install -y libxmu-dev
```

## 4 Install and configure pymedit

- Install pymedit :

```
 $ pip install pymedit
```

- Go to your site-packages folder, for example :

```
/home/username/.local/lib/python3.XX/site-packages/pymedit
```

- Open the file `abstract.py`
- Change the line 816 : modify "END" by "End"
- If you want to remove the details (maybe another simple way exists, but not found) :
  - In `abstract.py` : comment lines 138-140, 145-147, 153-156
  - In `mesh.py` : comment line 321,
  - In `mesh3D.py` : comment line 368.

## 5 Creation of a mesh without pymedit

- Create a unit square mesh with FEniCS :

```
import dolfin as df

mesh = df.UnitSquareMesh(100, 100)
df.File('boxmesh.xml') << mesh
```

- Convert the mesh, for example with meshio (pip install meshio) :

```
$ meshio convert boxmesh.xml boxmesh.mesh
```

- Generate an array of the level-set values (same size as the number of vertices in your mesh) and flatten it, for example :

```
import numpy as np

n = 101
X, Y = np.meshgrid(np.linspace(0.0, 1.0, n),
                   np.linspace(0.0, 1.0, n))
phi = (X - 0.5) ** 2 + (Y - 0.5) ** 2 - (0.3) ** 2
phi = phi.flatten()
```

- save your function :

```
f = open(
    'phi.txt',
    'w',
)
f.write('MeshVersionFormatted␣2␣\n')
f.write('\n')
f.write('Dimension␣2␣\n')
f.write('\n')
f.write('SolAtVertices␣\n')
f.write(f'{np.shape(domain)[0]}␣\n')
f.write('1␣1␣\n')
f.write('\n')

for i in range(len(domain)):
    f.write(f'{domain[i]}\n')

f.write('\n')
f.write('End')

import os

os.rename('phi.txt', 'phi.sol')
```

- Finally generate the mesh :

```
$ mmg2d_03 boxmesh -sol phi.sol -ls -nr -nsd 3 -hmax VALUE
```