

Medit: fast and light mesh inspection and visualization

Algiane Froehly

3 mai 2022

Contents

1	Installation	1
2	Prerequisite	2
3	Mesh inspection with Medit	3
4	Solution visualization	8
5	Animation	8
6	Modification of default view and preferences	9
7	Mesh edition:	10
8	To go further: read the documentation or the code	10

This file is available online at:

<https://sed-bso.gitlabpages.inria.fr/formations/org/MdB-Medit/MdB-Medit.html>

Medit is a software dedicated to fast and light mesh inspection and visualization (based on shortcuts).

1 Installation

Official repository: <https://github.com/ISCDtoolbox/Medit>.

My fork (adds support of retina screen + multiple solution fields):<https://github.com/Algiane/Medit>.

1. Get sources

```
wget https://github.com/Algiane/Medit/archive/refs/heads/master.zip -O medit.zip
unzip medit.zip
```

2. Install dependencies

- on Ubuntu
 - development tools:
`apt install -y cmake g++`
 - graphic tools:
`apt install -y freeglut3-dev`
- on Mac OSX Here I suppose that Xcode development tools and homebrew are already installed
 - development tools:
`brew install -y cmake`
 - graphic tools:
`brew install freeglut`

3. Build and install Medit (and add it to your PATH)

```
cd Medit-master && cmake -S. -Bbuild
make install -Cbuild
export PATH=$HOME/bin:$PATH
```

2 Prerequisite

- A mouse;
- A terminal;
- A mesh at medit .mesh file format (GMF file format v5, readable using the libMeshb library);

```
cat > 1tri.mesh <<EOF
MeshVersionFormatted 2
```

```
Dimension 2
```

```
Vertices 3
0. 0. 0
1. 0. 0
0. 1. 1

Triangles 1
1 2 3 0

Edges
2
1 2 0
1 3 1

End
EOF
```

- Optionnally: a solution file at medit .sol file format:

```
cat > 1tri.sol <<EOF
MeshVersionFormatted 2
```

```
Dimension 2
```

```
SolAtVertices 3
1
1
5.
4.
3.
End
EOF
```

Download the meshes needed for this MdB:

```
wget https://static.bordeaux.inria.fr/mmg/MdB-meshes.zip
unzip MdB-meshes.zip
```

3 Mesh inspection with Medit

- File opening:

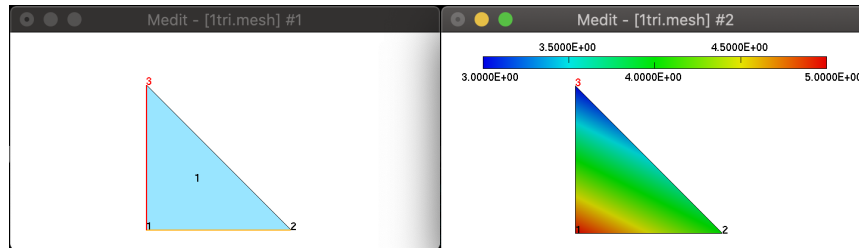


Figure 1: Mesh example

– In your terminal:

```
medit MdB-meshes/medit-visu.mesh
```

– `/!\` bug on hdpi screen: run medit with `src r` option if mesh appears on lower half quarter of the graphic window (see [[2]])

```
medit -r MdB-meshes/medit-visu.mesh
```

- Number of entities in the mesh and bounding box size are printed in terminal;
- Medit can be launched with other options (tested later):
 - `-a`: to load a series of files and create 'a' animations;
 - `-s`: to forbid the construction 's'urfaces in a mesh containing only volume elements.
- Mouse commands:
 - left click + movement: mesh rotation;
 - middle click + movement: mesh translation;
 - right click: menu with shortcuts reminder;
 - `Maj` + left click on mesh element: highlight element and prints element index, connectivity, reference and point coordinates in terminal;
 - `Maj` + middle click on mesh point: highlight point (and solution if exists) and prints point index, coordinates and reference (and solution value if exists).
- Main shortcuts (has to be runned with the medit window selected):

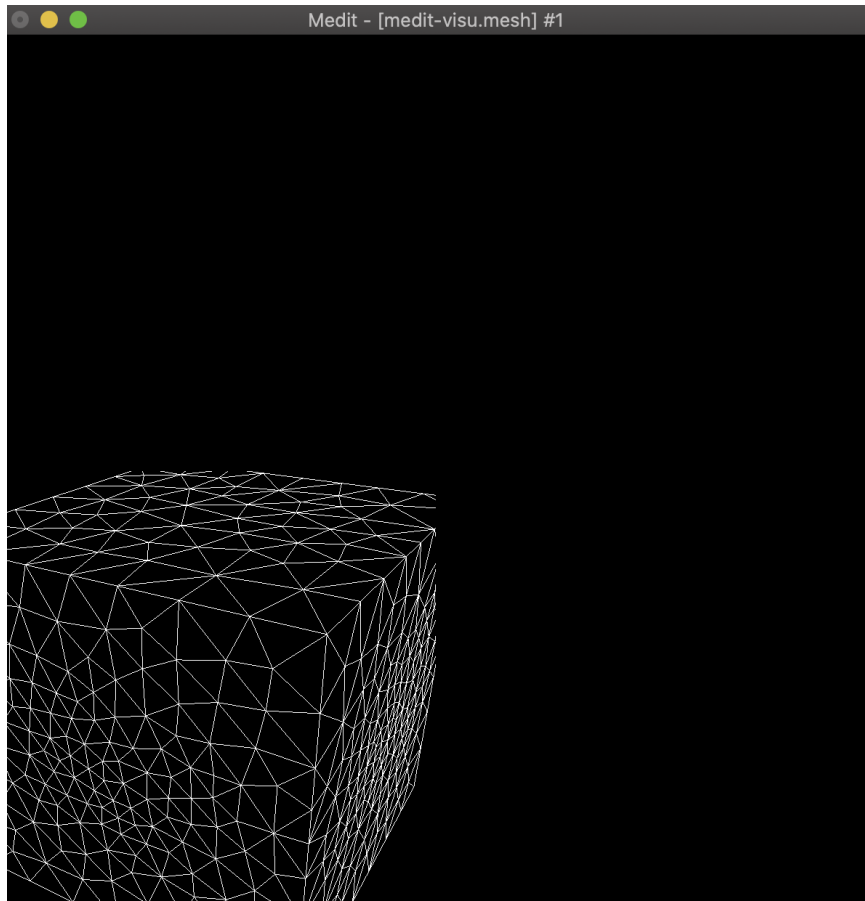


Figure 2: Hdpi bug

- **h**: online **'h'**elp;
 - **z**: **'z'**oom in, **Z**: **'Z'**oom view;
 - **+~/-~**: scale object;
 - **b**: **'b'**ackground color;
 - **g**: show/hide **'g'**eometric features (ridge edges, corners and required points provided in the mesh);
 - **c**: show/hide **'c'**olors (from element type?), **e**: show/hide **'e'**ntity colors from materials (ref fields of the mesh)
 - **l**: show/hide mesh **'l'**ines;
 - **f**: show/hide mesh **'f'**aces;
 - **P**: show/hide **'P'**oints numbering;
 - **F**: show/hide **'F'**aces numbering;
 - **#** in Medit window + **<n>** in the terminal: shows element, face and point of index **<n>**;
 - **V**: center **'V'**iew on selected entity.
 - **s**: **'s'**uppress selected element; (**/!** no possible undo);
Useful to check for element duplication or recovering. Try to suppress triangle num 29... there is a duplicated triangle below (you can suppress it too, to check that it is the only one)!
 - **r**: **'r'**emove selected material, **R**: undo previous **'R'** removal;
 - **N**: show/hide **'N'**ormals and tangent (if present in the mesh file);
 - **O**: **'O'**pposite normals;
 - **n**: show/hide smooth shading computed from **'n'**ormals;
- Deal with cutting planes in 3D:
 - **F1**: Clip/ unclip plane;
 - **F2**: Enable/disable plane displacement (same movements as for mesh displacement);
 - **F3**: Enable/disable mesh displacement with freezed plane;
 - **F4**: Show/hide volume elements;
 - **C**: cut/uncut elements that intersect cutting plane;
 - right click -> clipping -> Toggle plane: show/hide representation of the cutting plane;

- right click -> clipping -> Inverse orient: invert cutting plane;
- Link multiple meshes for comparative views: Open multiple files (or multiple times the same file) at the same time

```
medit MdB-meshes/medit-visu.mesh MdB-meshes/medit-visu.mesh MdB-meshes/medit-visu
```

- you can link mesh serially using `Alt-c` on origin window (for 'c'opy), `Alt-l` on window to link (for 'l'ink);
- try to link window B with window A, then window C with B. Now, modifications of the view done in A are reported in B and C too.
- Exploded view:
 - `F5`: switch between Eploded / normal view;
 - `F6`: increase explosion factor;
 - `F7`: decrease explosion factor;

Look between triangles 2389 and 2720: there is quasi degenerated triangle! By default, only the surface of mesh is represented, exploded view of tetra can be printed using cutting plane.

- View without surface reconstruction:
 - You can set the number of a given entity to 0 in the .mesh file to ignore it (check the diff between `medit-visu.mesh` and `medit-visu-nosurf.mesh`). By default, Medit reconstruct surface triangles from tetra.

```
diff MdB-meshes/medit-visu.mesh MdB-meshes/medit-visu-nosurf.mesh
medit MdB-meshes/medit-visu-nosurf.mesh
```

Shadows in colors are due to the orientation of created triangles. Removing shading with `n` solve the visu issue.

- To forbid this reconstruction and visualize only volume elements:

```
medit -s MdB-meshes/medit-visu-nosurf.mesh
```

Try to find vertex 2816, to center the mesh and to explode the view: there is a degenerated tetra too. To have a better view, we can remove some of the neighbouring tets.

4 Solution visualization

If a `.sol` file is attached to your mesh, it is automatically opened. Open `naca-multiSols.mesh` file:

```
medit MdB-meshes/naca-multiSols
```

Main shortcuts to visualize sol fields are:

- `m`: to show/hide the color 'm'ap of the solution;
- `p`: to show/hide the palette of colors (palette range can be modified using Medit preferences);
- `o`: to show isolines (In 3D isosurfaces can be shown using menu).
- `k`: to view scalar field as an elevation coeff (to combine with mesh lines `l` and isolines `o`)
- `>` and `<`: to view next or previous solution field;
- `w`: show/hide vectors of vectorial field;
- `v`: add streamline passing to a selected vertex (for a vectorial field).

You can test the last 2 options on the second field of the mesh. When a vertex is clicked, the index of the field and the field value at point is printed in the terminal.

Go to the field number 9:

- a SDP tensorial field is seen as an ellipsoid that represent the unit ball prescribed by the SDP tensor (surface of the ellipsoid is at distance 1 of the vertex in the metric space prescribed by the tensor). It is very useful when computing anisotropic size map for mesh adaptation

Note that the **Data** menu appears only if there is a solution attached to the mesh and changes depending on the solution type (with multiple solutions it doesn't change and shows only a part of the possibilities).

5 Animation

You can open and visualize a temporal solution by:

- saving a series of meshes under the form <basename>.<i>.mesh, where i is an integer that provides the time step (i values have to be contiguous);
- running medit with -a option followed by the indices of the first and last time step that you want to visualize:

```
medit -a 1 400 MdB-meshes/anim/lag-mot2D
```

- right-click -> animation -> play sequence: runs the animation;
- animation can be saved at ppm format (series of images) using right-click -> animation -> Toogle imgSave and playing the animation;
- then, ppm files can be converted into animated gif using convert (ImageMagick: <https://imagemagick.org/index.php>):

```
infile="lag-mot2D"
lastit=100

# rename files to have all indices in same format (order of images is wrong otherwise)
for i in {1..$lastit}
do
  printf -v ic "%03d" $i
  mv $infile.$i.ppm $infile.$ic.ppm
done
convert *.ppm -coalesce -delay 1 video.gif
```

6 Modification of default view and preferences

Open mesh medit-visu.mesh:

```
medit MdB-meshes/medit-visu.mesh
```

- E: allows to 'E'dit colors and transparency of a given material (an element has to be selected).
 - Diffuse values allow to change colors (first 3 values) transparency (last value) of material.
- W: 'W'rites configuration file

- if file is named DEFAULT.medit, it applies to all meshes, otherwise, preferences applies only to mesh with same basename that the .medit file.
- file can be manually edited to choose a given color palette, background colors, windows size, etc..
- entire list of readed keywords is available in the `parsop` function of `parsop.c` file. For example, you can change: `linecolor`, `linewidth`, `edgewidth`, `isowidth`, `pointsize`, `sunposition`...

7 Mesh edition:

Open mesh `medit-visu.mesh`:

```
medit MdB-meshes/medit-visu.mesh
```

- '%' in Medit window + <n> in terminal: Change entity reference (material) to <n>. Mesh must be saved (in <basename>.d.mesh) and reloaded to see effect.

8 To go further: read the documentation or the code

Available documentation of Medit is pretty old and not complete but you will find tips:

- In french : <https://www.ljll.math.upmc.fr/frey/logiciels/Docmedit.dir/Docmedit.html>
- In english : <https://hal.inria.fr/inria-00069921/document>

Medit is a small C code pretty easy to understand and modify, do not hesitate to open it and add things!