

FFEA Workshop part III: Meshing at the mesoscale

A Solernou

July 11, 2017

1 Introduction

This tutorial will guide you in building a volumetric mesh for biomolecular modelling. You will need to go through three steps: build a surface from an Electron Density Map (EDM), simplify or coarsen this surface, and create the final mesh to be parametrised into FFEA. Some times you will use Graphical User Interfaces (GUIs), i. e., programs where you can click, drag, explore menus, and write within fields. Some other times you will need to type some commands on a terminal. In order to help, this tutorial writes command-line text in monospace font, and you'll need to substitute the text in fields <within-brackets> correctly.


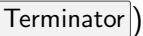
2 Build a surface from an electron density map

Electron density maps (EDM) describe the probability of electrons to be at a specific location. When looking at low resolution EDM, these reflect the shape and size of the object of interest. In this section you will learn how to build a surface file out of an EDM, i. e., a STL file that describes a surface as a set of triangles.



2.1 Obtain an EDM






Open a web browser, and go the EMDB (<https://www.ebi.ac.uk/pdbe/emdb>). We will work with a replicative DNA helicase lately referred as helicase, as it is complex enough to show some of the problems that can arise in meshing. Its 4-digit EMDB entry number is 2508. Look for it and download the map, but DO NOT close this site.

2.2 Visualise the map




Load the map in PyMOL. The map was downloaded into the “Downloads/” folder. Open a terminal ( ) to uncompress it and rename it so that you can load it on PyMOL:

```
cd Downloads/  
gunzip emd_2508.map.gz  
mv emd_2508.map emd_2508.ccp4
```

Now open PyMOL and load the map. You can do that clicking to   and selecting emd_2508.ccp4 from the Downloads/ folder. After doing so, the dimensions of the box in Angstroms will be printed, a new item “emd_2508” will appear on your right, but nothing will be displayed in the viewer window.

Visualise the map. Next to the item “emd_2508”, click onto   . You should now see the helicase, and if you click onto   next to “emd_2508_surf” you should see the box.



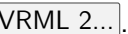
Now go back to the web browser, to the page where you downloaded the map, and click onto “Details” next to the “Experimental Information”. Check that the dimensions of the box printed by PyMOL are right and figure out the size of the helicase (Hint: multiply the voxel size by the number of grid points to work out the size of the box. Use this and the visualiser to estimate the size of your molecule).



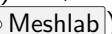

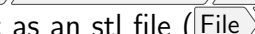


Experiment changing the level. Go back to PyMOL, and load again the surface, but changing its level (again next to “emd_2508”, click onto   ). What is happening?

2.3 Create a surface file

We need to get a surface file at the right contour level, and we will do that using PyMOL. Within the PyMOL command line type:

```
isosurface emd_2508_surf, emd_2508, <contour-level>
```

where you have to specify the right the right contour level from the experimental information details. When done, save the surface as a .wrl file by clicking onto   .

We need to convert this Virtual Reality Modelling Language (.wrl) file into a STerEO Litography (.stl) file, and we will use Meshlab for that. Open Meshlab (  ) , import the wrl file ( ) , and save it as an stl file ( ) giving it a new file name and type (state “Files of Type: STL File Format (*.stl)”). Further default saving options are fine.

You now have an “stl” file that is ready for simplification. Close Meshlab.


Note: [Chimera](#) provides much better support for EDMs, but we did not use it in this workshop as it is not free software. If you can accept their non-commercial license terms and install it, you would be able to inspect visually the different contour levels using its Volume Viewer (**Tools** > **Volume Data** > **Volume Viewer**), and obtain an “stl” file by exporting the scene (**File** > **Export Scene**).

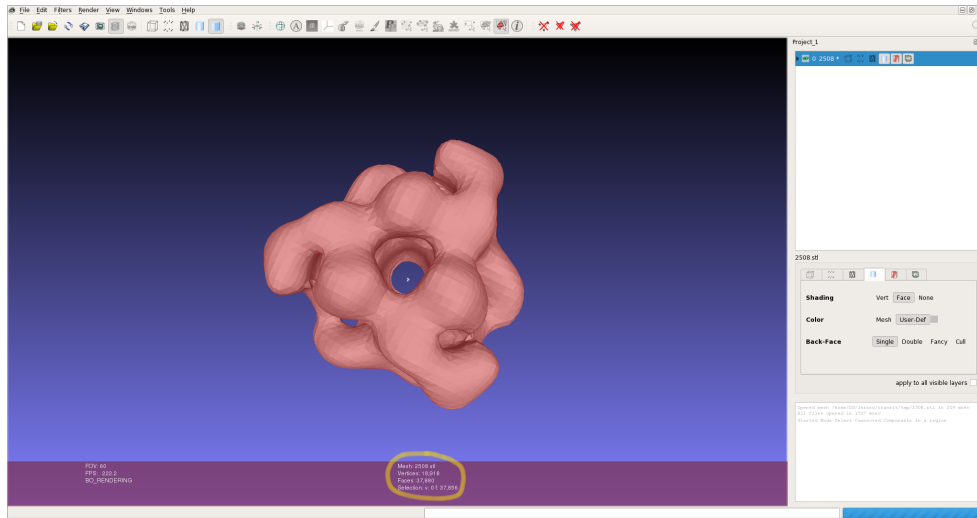
3 Coarsen a surface profile


Open Meshlab again (**Menu** > **FFEAWorkshop** > **Meshlab**) and import the stl file obtained at the end of the last section (**File** > **Import Mesh...**). If asked about unifying duplicated vertices, click **OK**.

Warning: Meshlab does not have any Undo button, so make sure you keep saving your work.



3.1 Clean the surface

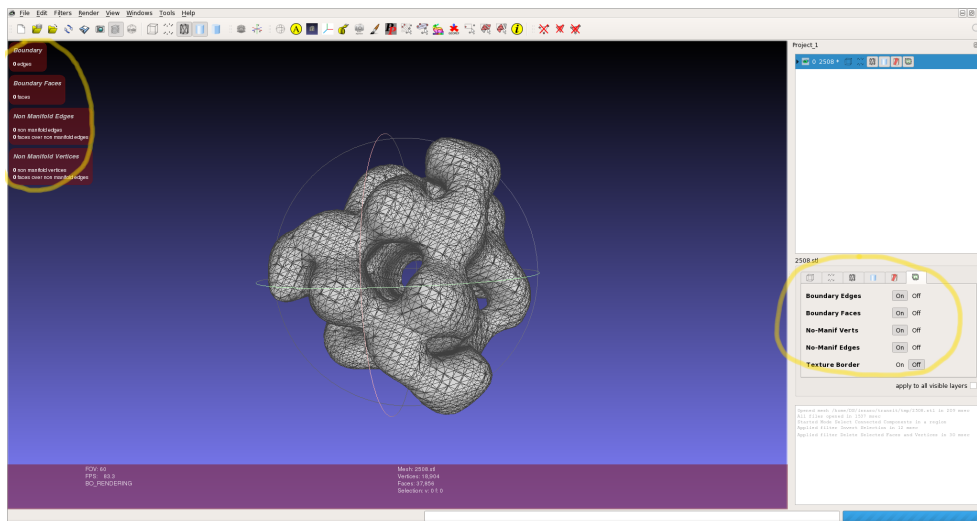
If you explore the geometry of the helicasé by rotating it in Meshlab, you will see that there are some independent floaters. Some of them are outside, and others may be inside the surface, and we need to remove them all. Click on the button , on the toolbar, to select connected components in a region, and select the helicasé drawing a small square with your mouse that carefully touches only its edge. The helicasé appears now shown in red, and you can see that the selection has thousands of faces, but no vertices (or nodes).



Now invert the selection (**Filters** **>** **Selection** **>** **Invert Selection**), and check only the faces. We can now remove all these faces (**Filters** **>** **Selection** **>** **Delete Selected Faces**), and see that the orphan nodes remain (click onto the “Points” button ). Remove these by clicking onto **Filters** **>** **Cleaning and Repairing** **>** **Remove Unreferenced Vertices**, and save this mesh (**File** **>** **Export Mesh...**).

3.2 Coarsen the surface

Check the surface Display the mesh clicking onto the button “Wireframe”  on the toolbar and see how dense it is (sometimes you even need to zoom in to see the triangles). It describes a closed 2-manifold surface (it has an interior and an exterior), and Meshlab can tell us so. To hand this, on the right side of the screen click on the border button  within the emd_2508.stl box (beware as there are two identical buttons) and select to display “Boundary Edges”, “Boundary Faces”, “Non-Manif Verts”, and “No-Manif Edges”,



as this will highlight these elements and show an updated summary on the left of the window. We can also check that there are no self-intersecting faces by clicking onto **Filters** **Selection** **Select Self Intersecting Faces** and noticing that “Selection” does not select any face (“v:0 f:0”).

Coarsen the surface. We need to simplify this surface dramatically, as the computational cost of running a simulation with this level of detail would be unaffordable. However, while simplifying the mesh the algorithm could go wrong by destroying manifoldness or creating self-intersecting faces, and a surface with any of these defects would be unusable. Thus, once satisfied with the level of coarsening, we need to check that the mesh is fine, and if not, fix it or restart again.

Start the simplification by clicking on **Filters** **Remeshing, Simplification & Reconstruction** **Simplification: Quadric Edge Collapse Decimation**. This will open a new window for one of the most robust decimators we have tried. Tick the boxes “Preserve boundary of the mesh”, “Preserve Normal” and “Preserve Topology”, while leaving the rest as they are. This instructs the algorithm to be conservative in the simplification process. Now type 0.9 within the “Percentage reduction” textbox, so that every time you click **Apply**, Meshlab will get rid of 10% of the faces. If you click **Help** Meshlab adds a description next to every field (which holds for almost every filter provided).



Simplify the surface through applying the filter successively until there are less than **450 vertices**. Close the filter and check that there are no self-intersecting faces (checking that no face is selected, after clicking onto **Filters** **Selection** **Select Self Intersecting Faces**). If the surface has none of such defects save the surface as a text .stl file: click on **File** **Export Mesh as...**, give a new file name and save it as .stl (state “Files of Type: STL File Format (*.stl)”). Then click **Save** and in the new appearing window make sure you untick the boxes “Binary

Encoding” and “Materialise Color Encoding”.

Open the decimation filter again, and iteratively simplify the mesh while checking for non-manifold edges or vertices, and self-intersecting faces. Remember to save the progress at every step. What is the coarsest clean mesh you can obtain? What happens if you keep simplifying it?

3.3 [Advanced] Fix the surface

If you have self-intersecting faces, or non-manifold edges or vertices, you can try still try to fix the surface. This is a tedious and complex process that that we do not intend to cover in detail here. Instead, the aim of this section is to provide some tips that can help you in the future when setting up surfaces for production.

The simplest approach is to select and remove problematic faces, edges and vertices, and then close the holes (Filters >> Remeshing, Simplification and Reconstruction >> Close Holes). Instead, selecting single faces can also be done more accurately. Once identified a problematic zone, click on the paintbrush button  and that will open a new window. Within this window, click on the red paintbrush , and that will let you select faces accurately. You can change from selection to movement modes by clicking `Esc`. Once selected and removed try to close the holes using the Meshlab filter.

Finally, you can move from Meshlab to Blender to fix some of these problems. [Blender](#) is an opensource 3D creation suite for modelling, animation, rendering, and game creating, that you can find installed under Menu >> FFEAWorkshop >> Blender. It will allow you to add, merge, move and remove nodes, vertices and faces with absolute flexibility. This should give you enough tools to solve almost any problem you encounter in surface meshing, and remember that as a modeller your priorities are to keep the shape, volume and topology of your objects.

4 From a surface to FFEA

4.1 Use Tetgen to build a mesh

Now that we have a surface profile we will use Tetgen to create a volumetric mesh, i. e., we will fill the closed surface of triangles with a tetrahedral mesh. [Tetgen](#) is a command line tetrahedral mesh generator that is able to keep the surface mesh. Now open a terminal, go to the working directory and type:

```
tetgen -Y <your-surface-mesh.stl>
```

where the option `-Y` informs Tetgen to keep the surface mesh intact. Detailed documentation on the different options available can be found [here](#). If the command is successful a message like:

Statistics:

```
Input points: 8982
Input facets: 2994
Input segments: 4491
Input holes: 0
Input regions: 0

Mesh points: 1457
Mesh tetrahedra: 4868
Mesh faces: 11233
Mesh faces on facets: 2994
Mesh edges on segments: 4491
```

will be printed onscreen. Otherwise, read well the error message (Cannot access file, A self-intersection was detected, ...) and address it. You should end this section with five new files:

```
<your-surface-mesh>.1.node <your-surface-mesh>.1.ele
<your-surface-mesh>.1.face <your-surface-mesh>.1.edge
<your-surface-mesh>.1.smesh
```

Note: [Netgen](#) provides an alternative to Tetgen, with nice graphical interface, and you can try it as it is installed in [Menu >> FFEAWorkshop >> Netgen](#). We hold some notes on its usage related to FFEA in our documentation <http://ffea.readthedocs.io/en/latest/surftovoltut.html>.

4.2 Build an FFEA system

We now have the structural information needed to build a FFEA model. Open a terminal, go to the relevant folder and type:

```
ffeatools voltoffea \
--mesh <your-surface-mesh>.1.node \
<your-surface-mesh>.1.ele <your-surface-mesh>.1.face \
--density 1.5e3 --shear-visc 1e-3 --bulk-visc 1e-3 \
--shear-mod 2.2e8 --bulk-mod 6.6e8 --make-script
```

to generate a complete set of FFEA input files with uniform parameters for density, bulk and shear viscosity, and bulk and shear modulus. FFEA takes all its input in SI units, so density here is 50% higher than water in kg m^{-3} , bulk and shear viscosity take the same values as if water in Pa s and shear and bulk modulus, in Pa come from considering a uniform Young's modulus of $6 \times 10^8 \text{Pa}$ with a Poisson ratio of 0.35. While these may be reasonable values for an initial approximation, we recommend to the user to calculate or measure these parameters for their system of interest.

You can now visualise the FFEA system on PyMOL and load it next to the EDM. Does they look similar? You should also be able to run a short simulation. Edit the newly created .ffea file, so that it it runs for 1,000 steps, with a time step of 10^{-14}s , and 10 snapshots are printed:

```
<dt = 1e-14>
<num_steps = 1000>
<check = 100>
```

Did it work? If yes, try whether the simulation is stable with a larger time step size, e. g., 10^{-13}s . If the time step is too big, you will get an error saying

ERROR:1 element has inverted since the last step.

Aborting simulation.

That means the fluctuations were too strong for a number of tetrahedra for them to invert, resulting in them having “negative” volume. You can fix that either decreasing the time step size or coarsening the mesh further with Meshlab. Try both approaches.