Soft object in MuJoCo

Creation of a deformable object from a 3D mesh

2024-07-31

Youenn Le Jeune
INSA Rennes / DIAG Sapienza

Step 1: Prepare the 3D model[1]

Step 1.1: Import the model

To begin with, you need to have a 3D model file to use. It can be of any filetype: we will import it in Blender, which supports many 3D model filetypes.

Once you have your model, open Blender. You should see something similar with the Figure 1.

Figure 1 — Blender welcome screen

Press *escape* to leave the welcome screen. Select the “Cube” from the right panel and press *delete*. Now you can import the model: go to “File”, “Import”, choose the right format and then import your file.

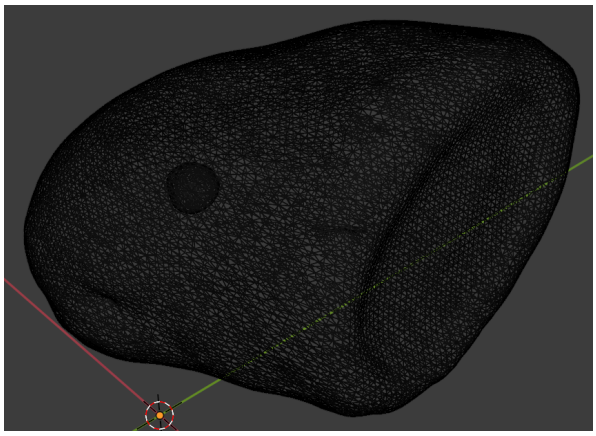
Step 1.2: Remove impurities

Some models may not be directly suitable to convert to MuJoCo. For instance, if they contain holes in their surface (Figure 2b) or voids inside their volume (Figure 2a), they must be fixed.

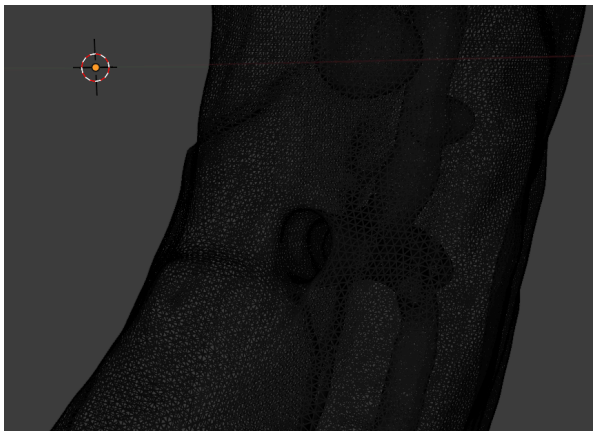
[1] **Disclaimer:** I am not at all a profesionnal 3D modeller. What I explain here, I understood it by myself and there are probably wrong things. If a person experienced in 3D modelling encounters this paper, please do not attempt to kill me.

Figure 2 — Impure models

(a) Kidney



(b) Liver



Step 1.3: Simplify mesh

Most 3D models are way too detailed to be imported in MuJoCo: it would cause the simulation to be extremely slow and require a gigantic amount of memory to run. Therefore, we must make our models "low-poly" by reducing the amount of vertices.

To do that, in Object Mode, select your mesh in the right panel. Go in the "Modifiers" menu at the bottom right and add a "Decimate" modifier (in the "Generate" submenu). See Figure 3a.

Figure 3a: Creation of the modifier

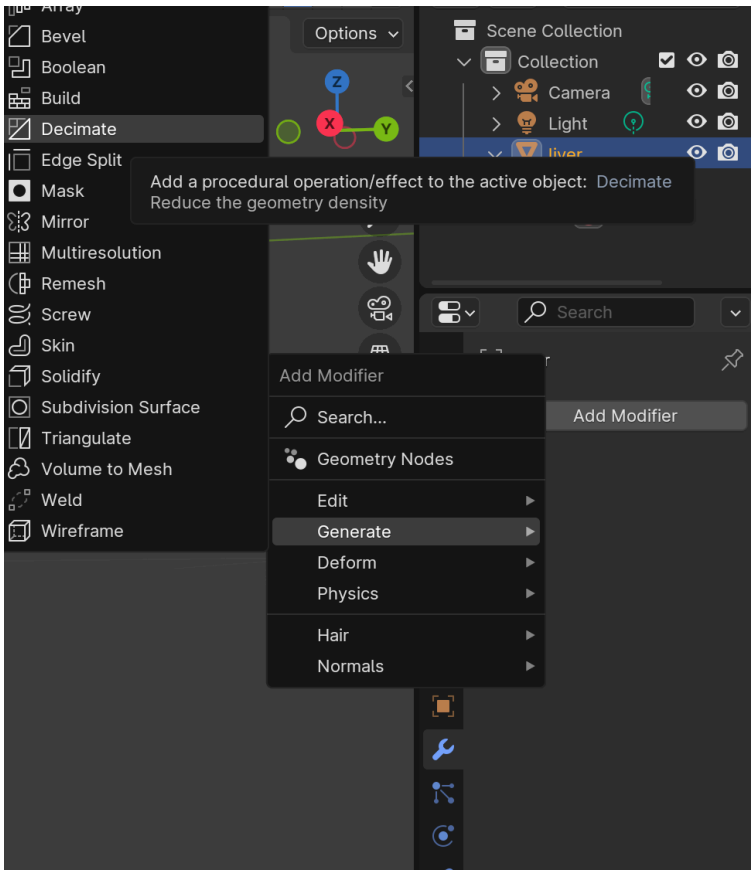
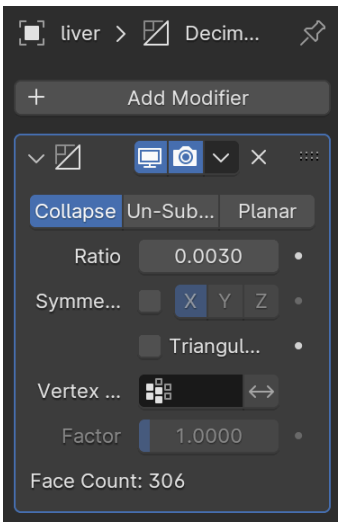


Figure 3b: Parameters



Once added, you can edit the "ratio" parameter (see Figure 3b). The goal is to have a low face count value (below 400 from my experience). Lowering the ratio will collapse vertices, converting the high-resolution model to a low-resolution one.

Figure 4 — Simplification of the liver model

(a) High resolution (b) Low resolution

Step 1.4: Export the mesh

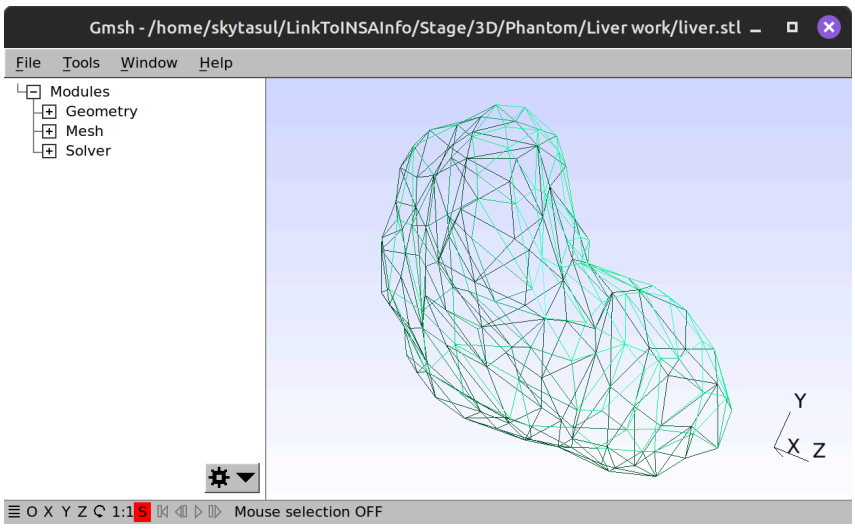
Once you are satisfied with the mesh simplification, you can export it. To do that, in Object Mode, select the mesh in the right panel. Then go to "File", "Export" and choose "STL (.stl)". In the options at the right side of the export window, tick "Selected Only". Export your mesh at the path of your choice.

Step 2: Conversion to volumetric mesh[2]

Step 2.1: Convert your mesh[3]

In Gmsh (installable for free on https://gmsh.info/), open your STL file using "File", "Open...". You should see a window similar to Figure 5.

Figure 5 — GMSH window after opening



[2]Informations have been gathered from this GitHub issue.

[3]This part is a transcription of this YouTube video.

In the left panel, click on “Modules” → “Geometry” → “Elementary Entities” → “Add” → “Volume”, then click on an edge of your mesh. You may get a popup asking you to create a .geo file – click on “Proceed as is”, then press on *e* to end the selection.

Now click on “Modules” → “Mesh” → “3D” to create a volumetric mesh. If you get an error at this point, see Section 2.2.

Once you have a valid 3D mesh, go to “File” → “Export”. Choose `Mesh - Gmsh MSH (.msh)` as the format and save. When it asks you for specific msh format, choose `Version 4 ASCII` .

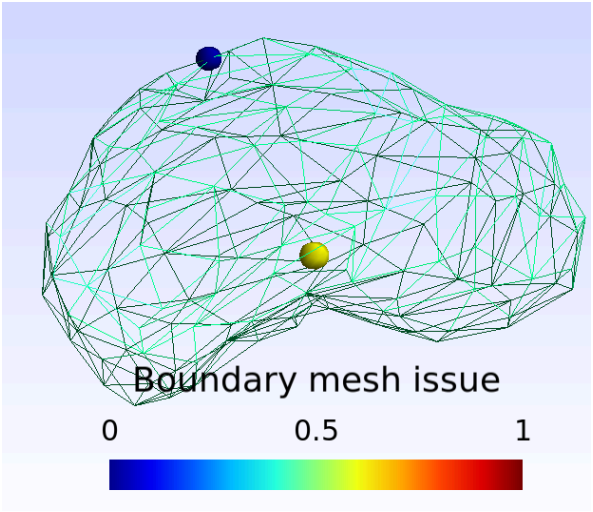
Step 2.2: Help! “3D” gives me errors

At the step of clicking on “3D” in Gmsh, you might get errors. This usually means the original mesh exported from Blender contains impurities or malformed sections. This must be resolved by hand on an individual basis.

See Figure 6 for an example of an issue. It was resolved by looking at the location specified by the blue sphere in Blender: there was a stray vertex creating an erroneous face. After deleting this face in Edit Mode, the 3D operation went successsfully.

Figure 6 — Example of a Gmsh issue when 3D meshing

(a) Mesh issue



(b) Full error

```
Info    : Reconstructing mesh...
Info    :  - Creating surface mesh
Info    : Found two duplicated facets.
Info    :   1st: [160, 161, 162] #1
Info    :   2nd: [160, 161, 162] #1
Error   : Invalid boundary mesh (overlapping facets) on surface 1 surface 1
Error   : No elements in volume 1
Info    : Done meshing 3D (Wall 0.00983517s, CPU 0.009077s)
Info    : 165 nodes 322 elements
Error   : ------------------------------
Error   : Mesh generation error summary
Error   :     0 warnings
Error   :     2 errors
Error   : Check the full log for details
Error   : ------------------------------
```

(This kind of full error is accessible by clicking on the bar at the bottom of the Gmsh window.)

Step 2.3: Clean the mesh

Your mesh file contains the necessary data and even more! Unfortunately, MuJoCo requires the .msh file to require *only* the necessary data (the volume and its associated tetrahedron). Therefore, we must clean the mesh of the unnecessary data.

Fortunately, someone created a Python script which does exactly that: https://github.com/mohammad200h/GMSHConverter. Clone it or download the “tree” branch, install the required libraries and run it using the following command:

```
$ python3 gmsh_cleaner.py -v 4.1 -i '<file>.msh' -o '<file>_converted.msh'
```

This will create many files, including `<file>_converted_vol.msh` . That is the important one, save it for later.

Step 3: Importing in MuJoCo

Now we have a file containing the necessary data for MuJoCo to create the soft body, we can include this file in an existing MJCF .xml file. There are multiple things to add in the file:

Step 3.1: Model parameters

In order to make a soft body work, there are some parameters to adjust in the whole model:

```
<option solver="CG" tolerance="1e-6" timestep=".001" integrator="implicitfast"
jacobian="sparse"/>

<size memory="100M"/>

<extension>
    <plugin plugin="mujoco.elasticity.solid" />
</extension>
```

The `option` tags seems to be necessary to have a stable soft simulation. The `timestep` can be set to be a little faster but not by much, 0.001 is a good start.

Step 3.2: Soft body

The most important thing to add is the actual declaration of the soft body. You will put it wherever you want in your bodies hierarchy, in the `<worldbody>` section:

```
<flexcomp name="soft_body_name" pos="x y z" type="gmsh" dim="3"
    file="file_converted_vol.msh" scale="1 1 1">
  <edge equality="true"/>
  <pin id="47 38 58"/>
  <contact internal="false" solref="0.005 1" solimp=".95 .99 .0001"
selfcollide="none" />
  <plugin plugin="mujoco.elasticity.solid">
      <config key="poisson" value="0.1" />
      <config key="young" value="5e4" />
      <config key="damping" value="0.005" />
  </plugin>
</flexcomp>
```

Among the parameters you can edit:

- the `pos` and `scale` attributes of `<flexcomp>`.
- `<pin />` to choose which bodies to pin in the world. The IDs are not guessable: use the MuJoCo "Simulate" executable, show the body names in the "Rendering" menu and choose which bodies to pin.
- the parameters of `<contact />`, which are not easily understandable. See MuJoCo XML reference.
- the 3 `<config />` values.

Finally, if you want to add a material or texture to the soft body, you'll have to create it in the `<asset>` section of your model and assign it in the `<flexcomp>`.