

Muscular fascicle arrangement based on Laplacian vector fields

Jan Kusterer, Niven Ratnamaheson, Raimund Rolfs, and Tobias Walter

Abstract—

Index Terms—muscle, fascicle, mesh, Laplace, electrostatic, streamline

1 INTRODUCTION

2 REASSEMBLING

The stl-file doesn't connect the surface triangles with its neighbours. Gmsh however needs one big surface to create a volume in which we can compute the streamlines. To connect all the small surfaces, we use the reclassify option with a threshold of 0. This runs an edge-detection for all triangles with an angle to its neighbour greater than 0. This excludes the two cut surfaces, since these surfaces all have an angle of 0. Finally we can recombine all the detected surfaces to one big surface.

With this new reassembled 3D-structure we now run our python script to detect the two cut surfaces. Since we cut the biceps orthogonal to the Z-axis the process is fairly simple. We look for all vertices with the highest z-coordinate for the upper boundary and the lowest z-coordinate for the lower boundary, respectively. These two sets are grouped and form two new surfaces. With this surface we save the maximum and minimum of the y- and x- coordinates for later use of the streamlines. We do this to cover the whole surface with the starting points of the streamlines.(bearbeitet)

3 MESHING

Meshing with Gmsh is fairly easy. On default gmsh selects between three 2D algorithms and two 3D algorithms. The automatic algorithm selection tries to select the most appropriate for the given structure. For 2D algorithms there are "MeshAdapt", "Delauny" and the "Frontal" algorithm. Every one of them has different uses. According to Gmsh The "MeshAdapt" works best for very complex, curved surfaces. "Frontal" is the best choice, when high element quality is important. And "Delauny" is fastest for large meshes of plane Surfaces. As stated in the manual for Gmsh the automatic selection chooses "Delauny" for plane surface and "MeshAdapt" for all other surfaces. For 3D algorithms there are "Delauny" and "Frontal". The Delaunay algorithm is the most robust and the fastest. However, this algorithm will sometimes modify the surface mesh, and is thus not suitable for producing hybrid structured/unstructured grids. In that case the Frontal algorithm should be preferred. As our mesh is only unstructured, the "Delauny" is our algorithm. The quality of the elements produced by both algorithms is comparable. For our 3D Mesh, first, the 2D surfaces, then the Volume is meshed.

4 SIMULATION

Our goal is to calculate streamlines, which according to [1] are close to equal to muscle fascicles. Gmsh's solver getDP features a plugin, which calculates the Streamlines based on the Laplacian vector field. Therefore we need a simulation on fluid flow or Electrostatics. The Laplace equation can be used in 3D just as in 2D. getDP's problem definition files (.pro) are used to describe the models for simulation. In this model we consider the calculation of the electric field given a static distribution of electric potential. This matches to an "electrostatic" physical model. On the one End of the Muscle we have a conducting surface on top of a dielectric Volume, called "Body". A Dirichlet boundary sets the potential on the

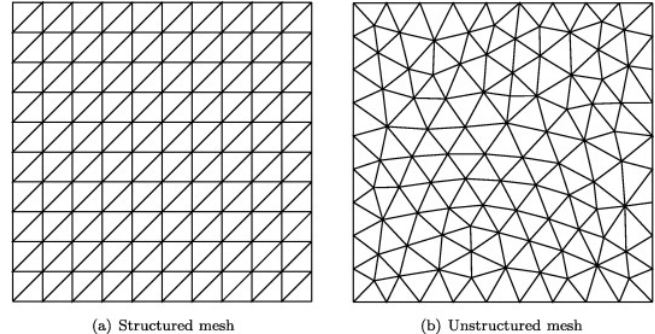


Figure 1. Comparisson of structured and unstructured grids []

boundary of the conducting surface, called "Electrode", to 10 mV and to 0 V on the other end of the Muscle, called "Ground". A homogeneous Neumann boundary condition is defined on the surface of the muscle to truncate the domain.

The Structure of the File is as Follows:

Group start by giving meaningful names to physical regions defined in the mesh file. We only use the Regions Body, Electrode and Ground. After that we define abstract regions, that are used below.

Function Here we define Material laws.

Constraint The Dirichlet boundary condition is defined piecewise. The constraint "Dirichlet_Ele" is invoked later in the FunctionSpace.

Group This is the domain definition of The FunctionSpace, which lists al regions on wich a field is defined. The Domain contains both the Volume and the Surface of the muscle.

FunctionSpace The functionSpace is used to pick the electric scalar potential. The solution is defined by:
-the domain defenition
-a type
-a set of basis functions
-a set of entities to wich the basis functions are associated (here all the nodes of the domain)
-a constraint (here the Dirichlet boundary conditions)

Jacobian

5 STREAMLINES

So now we use the earlier mentioned Plugin, which calculates Streamlines. The starting points of the Streamlines are on the upper surface of the biceps, which was calculated in the Reassembling section. The number of Streamlines can be calculated by multiplying the x- and y- coordinates from the mentioned surface. Afterwards instead of extracting the complete streamlines, we can only

extract the stepwise vector data of the streamlines into a postprocessing file in our case this is called BicepsStreamline.pos.

6 REPRESENTATION OF DATA

Our next step is to add up the vectors of each streamline separately. This is done by our Python Script Streamline-converter, which gets the vector data from the BicepsStreamline.pos file. Furthermore, the script creates a new file streamlines.geo with the data of each streamline in it. All in all, we can now use GMSH to merge the streamlines.geo file with our biceps surface to visualize the result.

ACKNOWLEDGMENTS

This work was partially funded by cake and coffee.

REFERENCES

- [1] H. F. Choi and S. S. Blemker. Skeletal muscle fascicle arrangements can be reconstructed using a laplacian vector field simulation. *PLOS ONE*, 8:1–7, 10 2013.