

Contents

0.1	Coupled Electrophysiology and Solid Mechanics	1
0.2	Simulations of the Neuromuscular System	11

0.1 Coupled Electrophysiology and Solid Mechanics

Next, we coupled the electrophysiology models given by the fiber based model presented in ?? or the multidomain model presented in ?? with a model of solid mechanics. The solver for nonlinear hyperelasticity was demonstrated in ?? in simulations of the passive behavior of muscle tissue. The goal in the current section is to simulate the contraction of active muscle tissue.

0.1.1 Fiber Based Electrophysiology and Muscle Contraction

In the first scenario, we couple the fiber based electrophysiology solver with the solid mechanics model and, thus, simulate muscle contraction. We use the subcellular model of Shorten et al. [Sho07], which computes the microscopic activation parameter $\gamma \in [0, 1]$. The parameter is mapped and homogenized from the 0D subcellular points to the 3D mesh to obtain $\bar{\gamma}$. In the macroscopic 3D mechanics description, the factor is multiplied with a maximum active stress parameter and a force-velocity characteristic, as described in ???. The 3D mechanics model updates the geometry of the 3D domain and transfers the fiber stretch λ_f and the contraction velocity $\dot{\lambda}_f$ back to the subcellular model.

We simulate a scenario with 169 fibers, which are associated with 15 MUs. This association is generated by method 1 in ???. The MUs are subsequently activated in a ramp in the first 1.4 s.

The muscle geometry is fixed at its lower end and no external forces are considered in this scenario. The dynamic formulation with the transversely isotropic Mooney-Rivlin material is used, as described in ???. The 3D muscle mesh contains $2 \times 3 \times 18 = 108$ quadratic elements with 1295 nodes in total and a partitioning for four processes is used. Time step widths of $dt_{0D} = dt_{1D} = dt_{\text{splitting}} = 10^{-4} \text{ ms}$ and $dt_{3D} = 1 \text{ ms}$ are used. The numeric solvers and other settings of the electrophysiology and contraction parts are used as described in ??? and ??.

[Figure 0.1](#) shows the fibers of the contracting muscle at four different timesteps between $t = 44 \text{ ms}$ and $t = 2.084 \text{ s}$. The fibers are colored according to the resulting activation parameter γ , which is a measure for the generated force on the sarcomere level. Between $t = 44 \text{ ms}$ and $t = 844 \text{ ms}$, shown in [Figures 0.1a](#) and [0.1b](#), the smallest MUs are activated, which in this example are mainly located at the left side such that the muscle domain initially bends slightly to the left. As more MUs become active at $t = 1684 \text{ ms}$ and depicted in [Fig. 0.1c](#), the deformation increases and the bending direction is reversed. However,

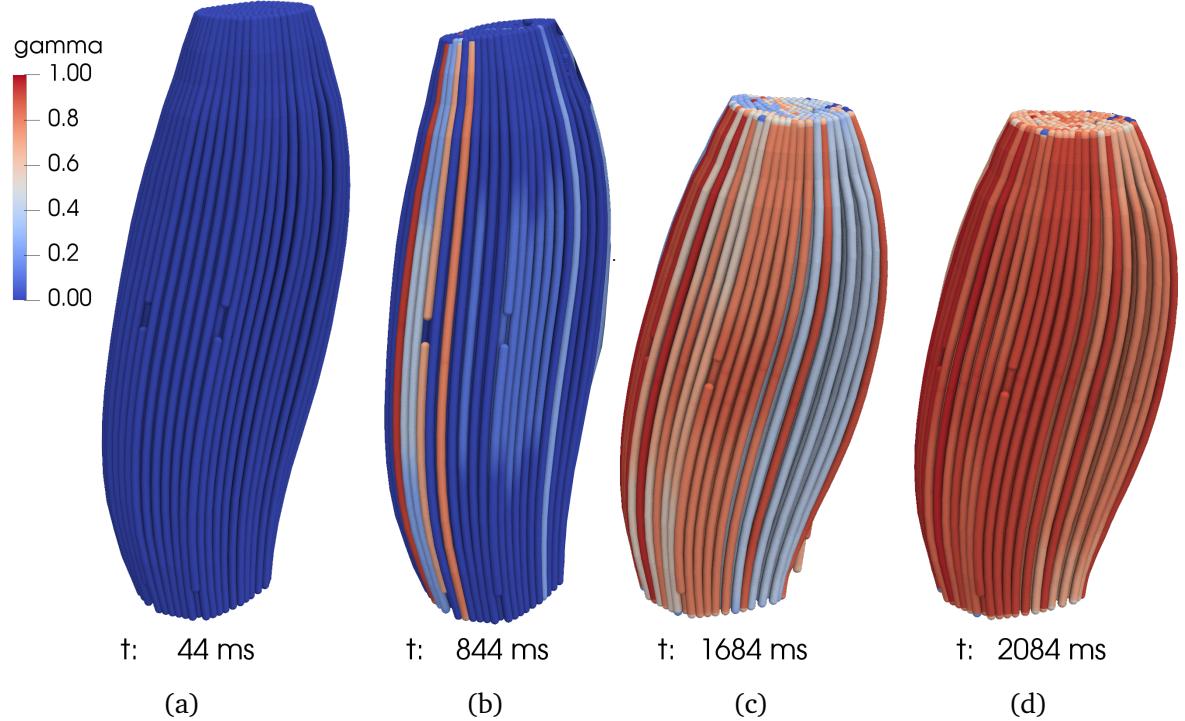


Figure 0.1: Simulation of fiber based electrophysiology and muscle contraction: Activation of the fibers and overall deformation at various timesteps.

the fibers at the left side still in Fig. 0.1c have the highest value of γ , as they have been stimulated most often at that time. At $t = 1684\text{ ms}$, visualized in Fig. 0.1d, almost all fibers show a maximum γ value near to one.

In Fig. 0.2, several quantities are given at the simulation end time of $t = 2084\text{ ms}$. Figure 0.2a shows the transmembrane voltage V_m on the muscle fibers. Action potentials can be seen on almost all fibers as the whole muscle is activated at this time in the scenario. Figure 0.2b shows a comparison between the reference configuration given by the yellow mesh and the current configuration given by the red mesh. The muscle domain is colored according to the stretch λ , which has a nearly constant value of $\lambda \approx 85\%$ at the end time of this scenario. A similar visualization is given in Fig. 0.2c for the active stress in the muscle.

Because of the high level of activation and the corresponding active stress distribution in the muscle, our mechanics solver only converges up to the shown simulation time of 2084 ms in this scenario, which aims at demonstrating the amount of contraction of a fully activated muscle. Other scenarios, where the activation is applied more slowly reach longer simulation times.

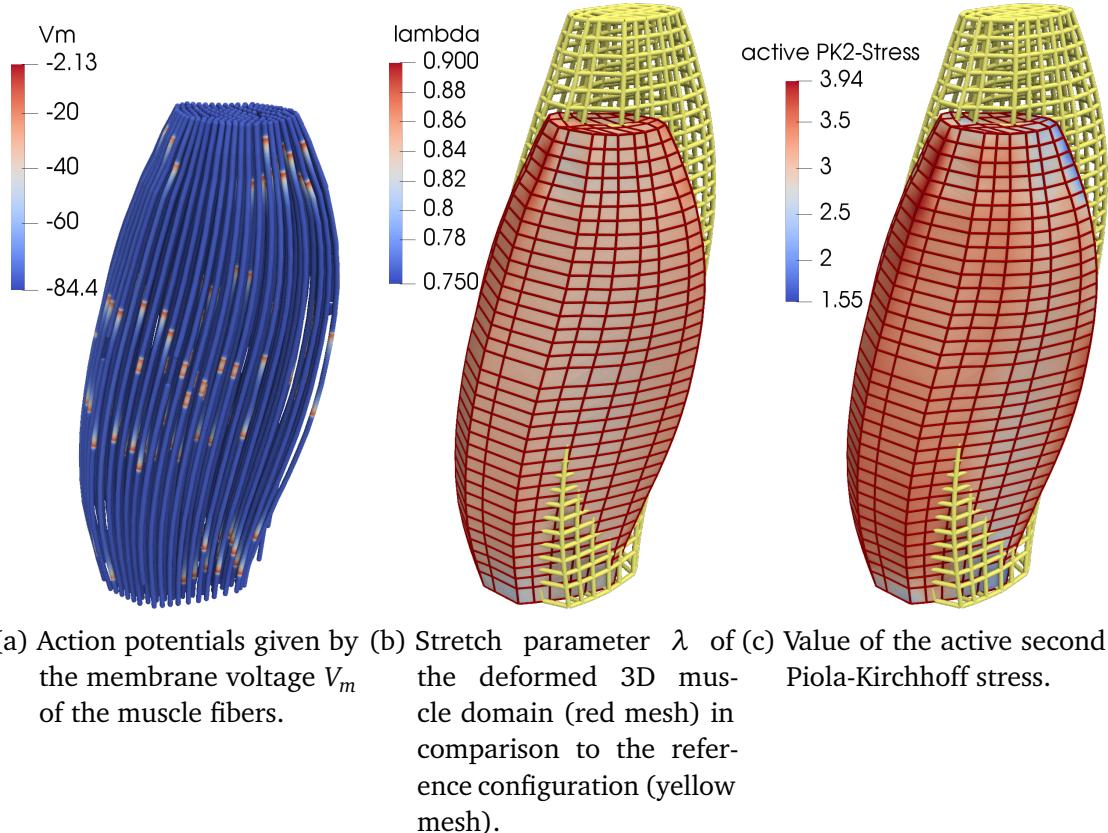


Figure 0.2: Simulation of fiber based electrophysiology and muscle contraction: Simulation results at $t = 2084\text{ms}$.

The presented scenario showed that a fully activated biceps muscle contracted to about 85 % of its original length. However, in reality, larger contractions are possible. In the shown scenario, the muscle was initially in a stress free configuration. More realistic scenarios can incorporate pretension forces, where the undeformed reference configuration is subject to a constant stress level in the muscle's direction of the line of action. This is considered in the next scenario.

How To Reproduce

The simulation can be run as follows:

```
cd $OPENDIHU_HOME/examples/electrophysiology/fibers/
    ↪ fibers_contraction/no_precice/build_release
mpirun -n 4 ./biceps_contraction ../settings_biceps_contraction.py
    ↪ ramp.py
```

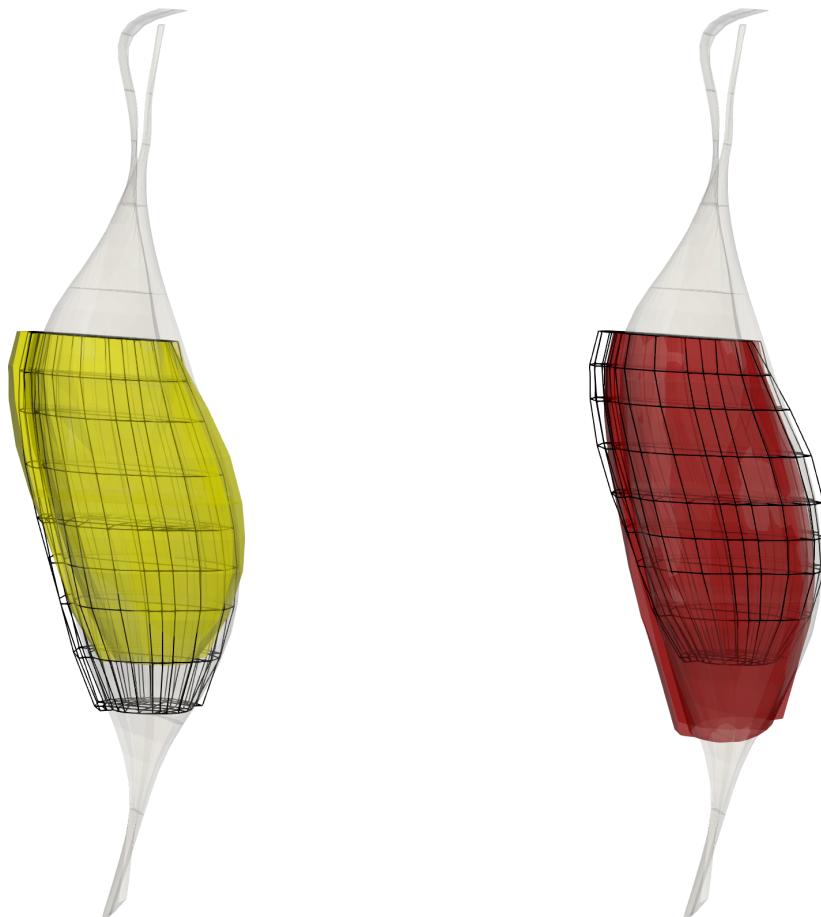
0.1.2 Simulation of Prestress

To obtain more realistic muscle contraction ranges, the goal is to have a nonzero constant prestress in the undeformed configuration of the muscle. In our solid mechanics formulation, the reference configuration always has zero stress. Thus, we construct a separate reference configuration of a shorter muscle geometry and extend this configuration to the original muscle length by applying external forces. The resulting configuration resembles the original muscle geometry and has the desired prestress characteristics.

The detailed steps of this algorithm are visualized in Fig. 0.3 and are described in the following. We begin with the given geometry of the muscle with body fat layer, which is shown as black wireframe mesh in Fig. 0.3a. In a first static simulation step, a constant active stress $\alpha_{\text{pre}} S_{\text{max,active}}$ is prescribed in the entire muscle volume. The resulting muscle deformation is computed, using the usual nonlinear hyperelastic muscle material. $S_{\text{max,active}}$ refers to the maximum active stress value as used in the mechanics model described in ???. The result of this first step is a shortened muscle with the same volume as the original geometry. Figure 0.3 shows the result by the yellow volume for $\alpha_{\text{pre}} = 0.3$. It can be seen that the length of the muscle has shortened by approximately 13 %.

In the next step, we use the computed geometry as new stress-free reference configuration and extend the muscle again by applying a constant surface load F_{pre} on the lower face pointing to the bottom. This step is again simulated by solving a static problem. The result is a similar muscle geometry as the original one, and contains prestress according to the applied force. The muscle volume is exactly preserved due to the incompressible material formulation. Figure 0.3b shows the starting point for the second step by the black wireframe mesh and the resulting geometry for a total applied force of $F_{\text{pre}} = 30 \text{ N}$ by the red volume. The comparison of the original, black mesh in Fig. 0.3a with the red volume in Fig. 0.3b shows a good match of the geometry. For the subsequent dynamic simulations of, e.g., muscle contraction, the surface load has to be constantly applied. It corresponds to the tendon forces and the loads of the musculoskeletal system that act on the muscle.

The active stress parameter α_{pre} and the corresponding preload force F_{pre} can be chosen according to the desired amount of prestress. However, the higher these values are chosen, the more difficult is it for the nonlinear solid mechanics solvers to converge to a solution. Especially for irregular or large mechanics meshes, a lower stress factor of, e.g., $\alpha_{\text{pre}} = 0.1$ has to be chosen. To improve convergence, we apply the load in the second step of the algorithm incrementally by several load steps. To yield better convergence properties, it



(a) In the first step, the original mesh (black wireframe) is contracted by an artificial active stress $\alpha_{\text{pre}} S_{\max, \text{active}}$ to yield a shortened muscle (yellow mesh).

(b) In the second step, the mesh is extended again by an external surface load. The black wireframe corresponds to the yellow volume in (a), the red volume is the resulting geometry.

Figure 0.3: Simulation of biceps muscle contraction with prestress: The two steps of the algorithm to generate a reference geometry with prestress, shown with the geometry of the tendons for reference.

also helps to reduce the number of unknowns and increase the mesh width in the mechanics problem, as this improves the conditioning of the problem.

0.1.3 Coupling of the Multidomain Model and Solid Mechanics Model with Prestress

Both the fiber based electrophysiology model and the multidomain model can be coupled with the nonlinear solid mechanics model to simulate muscle contraction. In the following,

we present a scenario that uses the prestress algorithm of the last section and couples the multidomain and mechanics models to simulate surface EMG signals on the skin surface over a contracting muscle.

We choose $\alpha_{\text{pre}} = 0.1$ and apply the prestretch force $F_{\text{pre}} = 10\text{ N}$ in three load steps. The multidomain model considers 5 MUs with stimulation frequencies between 7 Hz and 24 Hz and a 3D mesh of $8 \times 8 \times 28 = 1792$ elements. We execute the simulation with four processes. All other parameters and settings of the multidomain model and the solid mechanics model, which are not explicitly mentioned in the following are chosen the same as in ?? and ??.

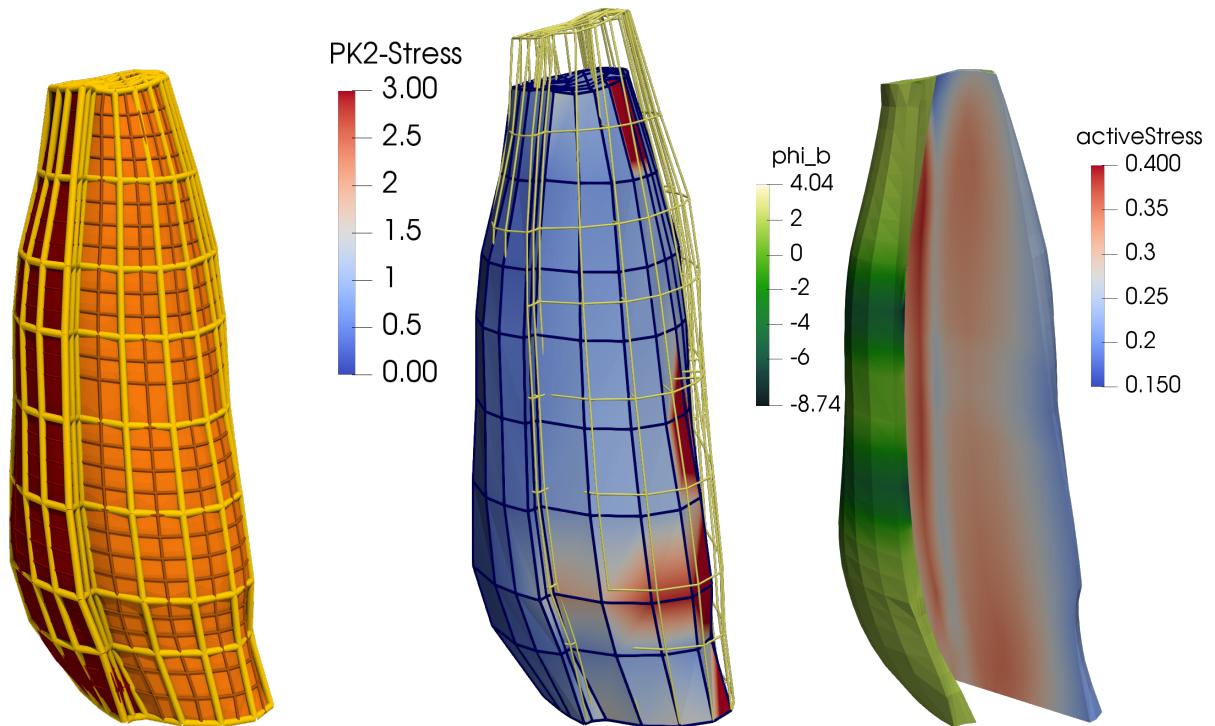
For the mechanics model, we use a coarser mesh than for the discretization of the multidomain model. Furthermore, we need quadratic elements instead of linear elements. The Python implementation of the settings script contains functionality to create the mechanics mesh by subsample the multidomain mesh with specified factors. In this scenario, we set these factors for the x , y and z directions to 0.7, 0.7, and 0.3, respectively. In result, we get a quadratic meshes with $5 \times 7 \times 9 = 315$ elements for the muscle and $5 \times 1 \times 4 = 20$ elements for the body fat domain. [Figure 0.4a](#) visualizes all meshes used in this scenario: The orange muscle mesh and the red body mesh are used for the multidomain model, and the yellow mesh is used for the solid mechanics model.

[Figures 0.4b](#) and [0.4c](#) depict results of the simulation at time $t = 920\text{ ms}$. [Figure 0.4b](#) shows the reference geometry after applying the prestress by the yellow wireframe. The muscle is colored by the second Piola-Kirchhoff stress. During the dynamic simulation, the muscle elastically moves slightly to the left and right, as it is only fixed at its bottom side in [Fig. 0.4](#). This explains the stress distribution at the snapshot in [Fig. 0.4b](#), where higher stresses occur at the right side.

[Figure 0.4c](#) shows the electric potential ϕ_b of the body domain by the green color scale on the left of the image. The visible part of the fat layer shows two action potentials visualized by the two dark green stripes.

Moreover, [Fig. 0.4c](#) displays the total active stress in the interior of the muscle by the color scale that ranges from blue to red color. In the multidomain model, the total active stress $\mathbf{S}_{\text{active}}$ is calculated by a weighted sum over the values $\mathbf{S}_{\text{active}}^k$ of the MU compartments, scaled by the occupancy factors f_r^k (cf. ??):

$$\mathbf{S}_{\text{active}} = \sum_{k=1}^{N_{\text{MU}}} f_r^k \mathbf{S}_{\text{active}}^k.$$



(a) Multidomain meshes of the muscle domain (orange) and body fat domain (red) and the coarser mesh used for the solid mechanics model (yellow).

(b) Reference configuration (yellow mesh) and current configuration of the muscle colored according to the distribution of the second Piola-Kirchhoff stress.

(c) Electric potential ϕ_b in the body domain (green color scale, in millivolts) and active stress in the interior of the muscle (blue-red color scale, in $\frac{N}{cm^2}$).

Figure 0.4: Simulation of muscle contraction. Used meshes and simulation results at $t = 920$ ms of a scenario of the multidomain electrophysiology model coupled to the solid mechanics model.

The muscle domain is cut open such that interior distribution at the cut plane can be seen. The image shows two regions of higher active stress which run vertically through the muscle, given by red color. They depend on the location of the MUs in this scenario. The legend shows that the active stress inside the muscle is below $0.4 \frac{N}{cm^2}$, while the maximum active stress parameter is chosen as $7.3 \frac{N}{cm^2}$. The low activation level is a result of the chosen MU recruitment. In result, the muscle only slightly contracts, as shown in Fig. 0.4b.

How To Reproduce

The simulation can be run with the following commands. Instead of four processes also other numbers are possible. A lot of parameters can be fine-tuned in the `../variables/multidomain.py` settings file.

```
cd $OPENDIHU_HOME/examples/electrophysiology/multidomain/
→ multidomain_prestretch/build_release
mpirun -n 4 ./multidomain_prestretch ../
→ settings_multidomain_prestretch.py multidomain.py
```

0.1.4 Coupling of Solid Mechanics Models using the Software preCICE

One problem of solid mechanics models with low mesh resolutions is the limited amount of parallelism of the scenario. The domain can only be partitioned to so many subdomains as there are elements in the 3D mechanics mesh. While this is not an issue for small scale simulations like the ones shown in the previous sections, it prohibits exploitation of High Performance Computing resources, e.g., if a large number of muscle fibers is considered as in ??.

The reason for the limited parallelism lies in the partitioning scheme, where every node in the 3D domain corresponds to the subdomain of exactly one process, regardless of the mesh. Thus, OpenDiHu does not allow to partition, e.g., the finely resolved 1D muscle fiber meshes differently than the coarse 3D mechanics mesh or the 3D multidomain mesh. However, the restriction can be circumvented by using multiple OpenDiHu programs with different partitioning schemes and by performing the data transfer between the meshes using external coupling software.

We provide support for the black-box solver coupling library preCICE [Bun16]. This open source library allows to map data between different meshes, can communicate values between subdomains that reside on different processors, and implements implicit numerical coupling schemes with quasi-Newton methods. The implementation is known to scale well on small-scale clusters and supercomputers. The preCICE library targets a minimally-invasive approach, where the user application implements a preCICE adapter. Multiple, potentially different solver codes can then be numerically coupled and solve

individual parts of a common simulation. Moreover, an active and growing community exists and open source adapters are available for several popular solvers.

This makes the library suited for our use case. We provide two different types of preCICE adapters, one for surface coupling of 2D meshes and one for volume coupling of 3D meshes. These adapters integrate with the structure of nested solvers in OpenDiHu and can be positioned anywhere in the solver tree (cf. ??). The meshes and variables to expose to preCICE can be configured in the settings file. In the following, we present an example using volume coupling adapters. The next section presents a simulation that uses surface coupling.

We simulate muscle contraction and surface EMG of the biceps muscle using the fiber based electrophysiology model. To run the computation on an 18-core Intel Core i9-10980XE processor, we want to compute the electrophysiology model using 16 processes and the mechanics model using 2 processes. The data mapping between the differently partitioned 3D meshes is performed by preCICE.

[Figure 0.6](#) shows the scheme of used meshes and exchanged variables in this simulation. Two different OpenDiHu programs are executed at the same time, given by the gray boxes. The left program solves the electric conduction problem, given by the bidomain equation ??, on the 3D domain and the action potential propagation model, given by the monodomain equation ??, on a large number of 1D muscle fiber meshes. The 3D and 1D meshes in this program are correspondingly partitioned to 16 subdomains in total for 16 processes. [Figure 0.6](#) visualizes the meshes and their partitioning the colored inset images. TODO

How To Reproduce

```
cd $OPENDIHU_HOME/examples/electrophysiology/fibers/
    ↵ fibers_contraction/with_precice_volume_coupling/build_release
    ./muscle_contraction ../settings_muscle_contraction.py ramp.py
    mpirun -n 16 ./fibers_with_3d ../settings_fibers_with_3d.py ramp.py
```

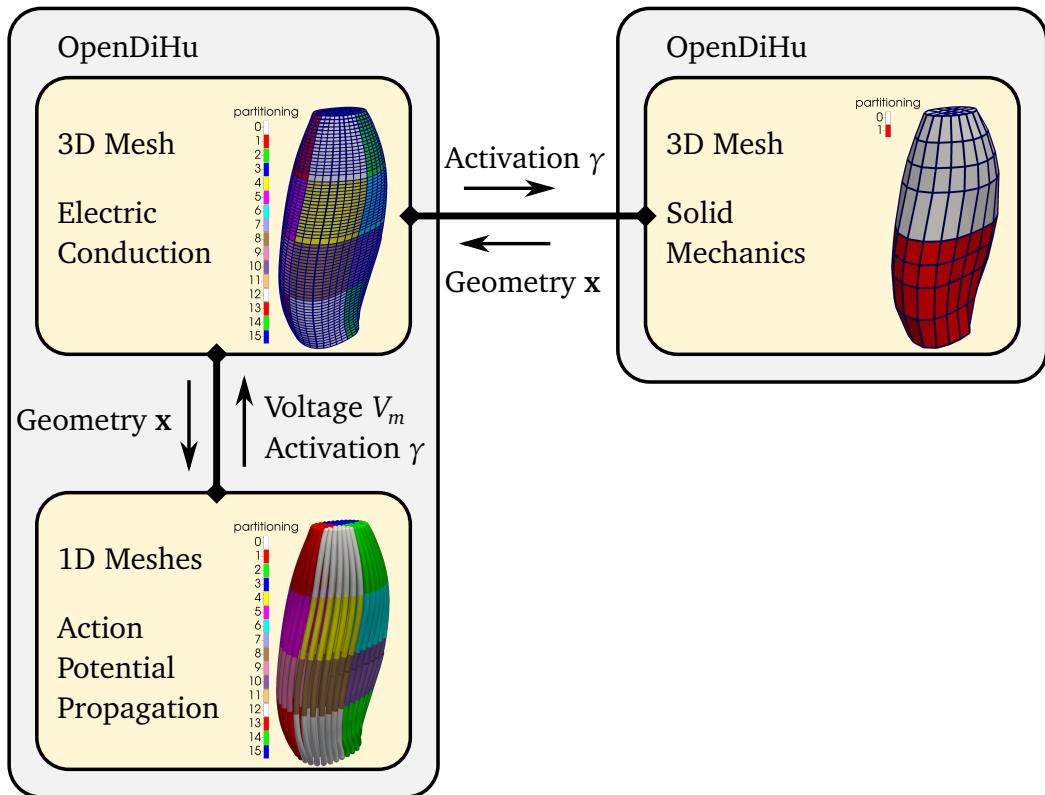


Figure 0.5: preCICE

0.1.5 With tendons preCICE

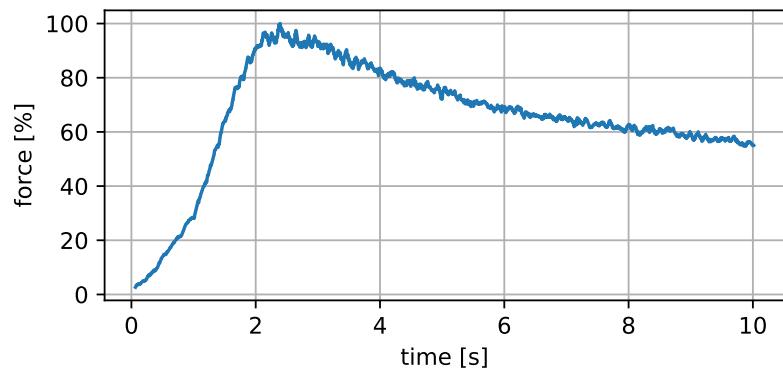


Figure 0.7: Muscles

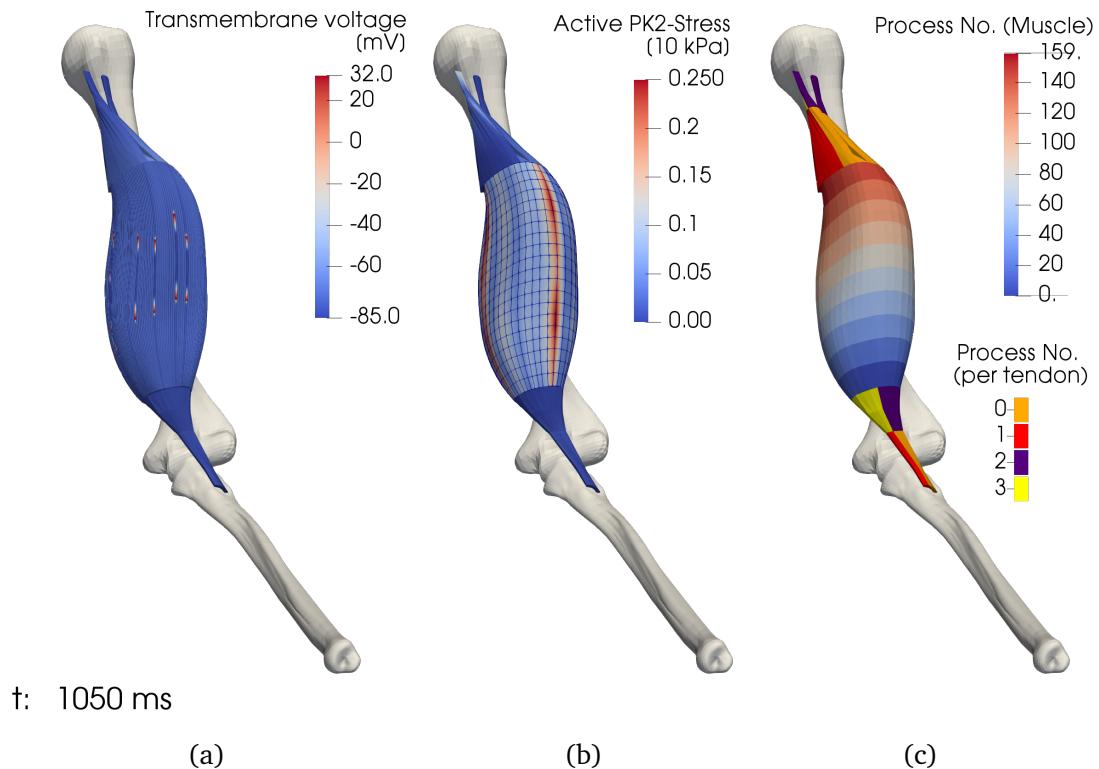


Figure 0.6: Coupled simulation of muscles and tendons.

0.2 Simulations of the Neuromuscular System

0.2.1 Neuromuscular system with Spindles and Prestretch

0.2.2 Neuromuscular system with More Sensor Organs

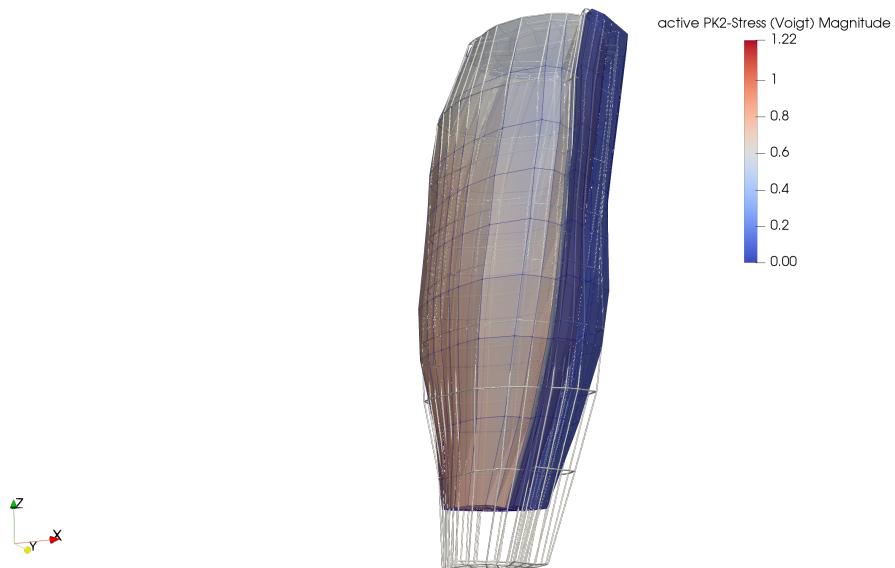


Figure 0.8: Reference and current configuration. neuromuscular contraction0

Bibliography

- [Bun16] **Bungartz**, H.-J. et al.: *Precice – a fully parallel library for multi-physics surface coupling*, Computers & Fluids 141, 2016, Advances in Fluid-Structure Interaction, pp. 250–258, ISSN: 0045-7930, doi:<https://doi.org/10.1016/j.compfluid.2016.04.003>, <https://www.sciencedirect.com/science/article/pii/S0045793016300974>
- [Sho07] **Shorten**, P. R. et al.: *A mathematical model of fatigue in skeletal muscle force contraction*, Journal of Muscle Research and Cell Motility 28.6, 2007, pp. 293–313, doi:[10.1007%5C2Fs10974-007-9125-6](https://doi.org/10.1007%5C2Fs10974-007-9125-6), <http://dx.doi.org/10.1007%5C2Fs10974-007-9125-6>