Multiscale kinematic growth coupled with mechanosensitive systems biology in open-source software

Authors and developers:
Steven A. LaBelle, Ph. D.,
Mohammadreza Soltany Sadrabadi, Ph. D.,
Amirhossein Arzani, Ph. D.

1-Introduction:

This repository contains the input files required to reproduce nonlinear transient structural mechanics problems presented in LaBelle et al. 2024. Input files are provided for reproduction in Finite Elements for Biomechanics (FEBio) or Python (FEniCS). These simulations couple the kinematic growth and remodeling (G&R) of a biological tissue/structure with systems biology and cell signaling pathways. At the cell scale, we implement a system of ordinary differential equations (ODEs) or partial differential equations (PDEs) representing the reactions between the biochemicals causing the growth. We implemented the weak form of the governing continuum mechanics equations using finite element analysis representing the G&R at the tissue level. Three test subjects (cube, aneurysm, and aortic valve) are implemented in the package.

2- Installation:

FEBio installation:

The FEBio software suite includes FEBio and FEBioStudio which provides a GUI to create models and analyze results. The best reference to install FEBio + FEBioStudio: https://febio.org/downloads/. We additionally need the FEBio Elastic Reaction Diffusion (ERD) solver plugin which may either be built from this repository or you may use the pre-build .dll in this repository (built for Windows). Model input files have the extension .feb which is based on xml. These can be viewed and edited in FEBioStudio or a text editor. An excellent introduction to FEBio can be found at https://febio.org/knowledgebase/.

Once built, users need to specify a configuration file for FEBio to find the .dll for the solver plugin. FEBio uses an xml document for the configuration file which is included with the installation as febio.xml. This file needs to be modified with the following line to allow FEBio to find the plugin:

```
<import>[full path]/FEBioERD.dll</import>
```

Now, the plugin may be run either through the FEBioStudio GUI or the command line.

Running models in FEBioStudio:

First, the plugin must be registered in FEBioStudio. Open FEBioStudio and navigate to FEBio>Manage FEBio Plugins. Select “load…” and find your copy of FEBioERD.dll. Now, you can create models and run models with features from the plugin. Once your model is created and saved, the model is run by selecting FEBio>Run FEBio. A menu will appear with options for specifying the model run parameters. Expand the "Advanced Settings" section and add the path to your .dll to "Config file".

Running models from the command line:
To point FEBio to the .dll from the command line, one only needs to add a flag for the config file. The model may thus be run as:

```
$ febio4.exe -i [input_file.feb] -cnf [config_file.xml]
```

FEniCS installation:
To run the software, users need to install legacy FEniCS via DOLFIN (NOT DOLFINx). The best reference to install FEniCS is: https://fenicsproject.org/download/archive/. FEniCS may be run either through the conda Python package manager or with a docker image. Input files for either approach can be run with the command:

```
$ python3.6 [input_file.py] > [log_file.txt]
```

Installing FEniCS through conda:
We tested our code using the miniforge3 distribution of conda (https://github.com/conda-forge/miniforge). The project is created and activated with:

```
$ conda create -n fenicsproject -c conda-forge fenics=2019.2
$ source activate fenicsproject
```

Installing FEniCS through Docker:
Use the latest image (stable:latest not stable:current):

```
docker run -ti -p 127.0.0.1:8000:8000 -v $(pwd):/home/fenics/shared -w /home/fenics/shared quay.io/fenicsproject/stable:latest
```

3-Files and folders:
The software contains three folders for the cube and aneurysm models. All of the code is implemented in C++. The code can be run in parallel by specifying the number of threads before running the model:

```
$ set OMP_NUM_THREADS=[number_of_threads]
```

5- Post-processing:
You can see the results in the open-source software FEBioStudio or ParaView. Model results may be viewed in FEBioStudio by opening the associated ".xplt" file generated by FEBio. To visualize in ParaView, the model must first be opened in FEBio. Once in FEBio, select "Save As" and choose ".vtk" as the file type. A time-series data set can be output by selecting "Export all states"

and enabling “output VTK series file.” Then, open ParaView and select the .vtk-series file. ParaView can be downloaded at: https://www.paraview.org/download/

Additional steps are needed to see the displacement and the growth for each time span in ParaView. This can be accomplished with the “warp by vector” option in Filters->alphabetical->warp by vector.