Open-source software for coupling growth and remodeling of tissues with cell signaling pathways

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

1-Introduction:

This repository contains the input files required to reproduce nonlinear transient structural mechanics problems presented in Soltany et al. 2023. Input files are provided for reproduction in Python (FEniCS) or Finite Elements for Biomechanics (FEBio). 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- FEniCS Installation:

FEniCS interfaces with MPI and linear algebra backends like PETSc to solve finite element problems in Python. To work with this package, a user should know the basic knowledge of how to solve PDEs through

FEniCS and basic programming skills in Python. An excellent introduction to FEniCS can be found at https://fenicsproject.org/tutorial/

The input files provided are compatible with the legacy version of DOLFIN/FEniCS, not the current release versions of DOLFINx/FEniCSx. To run the software, users need to install FEniCS, an open-source finite element software written in Python: the best reference to install FEniCS is: https://fenicsproject.org/download/
The simulations were originally generated using a containerized version of DOLFIN/FEniCS via docker. The container contains a build of the software with most of the necessary dependencies already installed. To install via Docker, the following command may be used:

```
$ docker run -ti quay.io/fenicsproject/stable:latest
```

A local repository can be added as a volume to the container through:

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

Alternatively, the package can be installed via conda:

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

Some useful packages:

- scipy:
    - Scipy is a built-in python package that provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations and etc. We used this package to solve the systems of ODEs for cell signaling pathways. This will need to be installed in the FEniCS environment or Docker container:

      Docker:
      ```
      $ pip install scipy
      ```

Conda:

```
$ source activate fenicsproject
$ conda install scipy
```

- mshr:
  - This package is a built-in python package and is a supplementary package in the FEniCS environment for making simple geometries and discretizing and meshing the geometries to tetrahedral elements. The project is no longer active so it is not recommended for use with newer versions of DOLFINx/FEniCSx. It does however come preinstalled with the container. For use with conda, install mshr via:

```
$ conda install -c conda forge mshr
```

- Growth and Remodeling Software/Input files:
  - The solver can be installed simply by cloning the GitHub repository to your own computer:

```
$ git clone https://github.com/amir-cardiolab/valve-growth.git
$ cd code
```

  - Alternatively, you can download software from the link below:

```
$ https://github.com/amir-cardiolab/valve-growth.git
$ unzip code.zip
$ cd code
```

3-Files and folders:

The software contains three folders for the cube, aneurysm, and aortic valve models. All of the codes are implemented in Python. The code can be run in parallel via:

```
$ time mpi -n [# processors] python [filename.py]
```

Note 1: The aortic valve geometry was made in another software than mshr due to the complexity. The aortic valve geometry is available at: https://drive.google.com/file/d/1QVIGgRxDCMKsZHVPIoSGhBz3DxKBMSlv/view?usp=sharing

5- Post-processing:

You can see the results in the open-source software Paraview. To download the software, you can use the link below:
https://www.paraview.org/download/

To see the displacement and the growth for each time span, you need to use the “warp by vector” option in Filters->alphabetical->warp by vector.