Nonlinear Stokes Flow in FEniCS

About

The purpose of this document is to guide the user in running the nonlinear Stokes code flow program contained in the Nonlinear Stokes Flow in FEniCS (NSFF) package. The user should have basic knowledge of the Python programming language (http://www.python.org/). The user must also install FEniCS finite element software (http://fenicsproject.org/). Instructions for installing FEniCS are provided on the FEniCS website.

The programs written in the NSFF package are compatible with FEniCS version 2016.2 and may not be compatible with other versions. For questions about the NSFF package, contact the authors at stephen.k.jimenez@vanderbilt.edu and ravindra.duddu@vanderbilt.edu.

Package Contents

The programs contained within the NSFF package use relative paths for accessing files and generating plots, and so the package layout should not be altered. The original layout is:

```
Nonlinear Stokes Flow in Fenics/
        mesh/
                hdf5/
                xml/
        output/
                data/
                details/
                figs/
        depth-vs-time.py
        generate-mesh.py
        nonlinear-stokes.py
        quad-plot.py
        plot.py
```

The **mesh/** directory is used for storing finite element mesh files in HDF5 or XML format. The **output/** directory is used for storing output. The subdirectory **data/** contains the actual simulation output in .h5 format, whereas **details/** stores simulation logs (i.e., convergence and time-stepping information) in text format. The **figs/** subdirectory contains all the figures that get generated during post-processing.

Using the Package

The programs included in the NSFF package are written in Python (specifically, Python 2.7.10). A brief description of each file and instructions for use is given below:

- **depth-vs-time.py**
  This file generates a plot of the surface crevasse depth d_s versus time in hours. The plot is saved as a PDF file in the **`output/figs`** directory. In order to run the file, the user needs to enter **`python depth-vs-time.py`** into the computer terminal. Upon running, the program will prompt the user to input the number of the last data file to be included. The user should only supply the number and not the ".h5" file extension.

- **generate-mesh.py**
  This file uses FEniCS built-in libraries to generate a mesh. The mesh is generated in both .h5 and .xml formats, which are stored in the **`mesh/hdf5`** and **`mesh/xml`** directories, respectively. A picture of the mesh is saved as a PDF file in the **`output/figs`** directory. In order to run the file, the user needs to enter **`python generate-mesh.py`** into the computer terminal. Note that FEniCS will generate all mesh files as a set of triangular cells and nodes at vertices. FEniCS will automatically generate additional nodes during the simulation if the user defines a higher-order finite element function space (e.g., quadratic or cubic shape functions).

- **nonlinear-stokes.py**
  This file contains the actual simulation program. The user must edit the file to change the material parameters, boundary conditions, and the amount of time the simulation runs. Output is stored in .h5 format in the **`output/data`** directory. Details about the simulation (including the number of Picard iterations for convergence at each time step) are logged in the **`output/details`** directory. In order to run the simulation in serial, the user needs to enter **`python nonlinear-stokes.py`** into the computer terminal. The simulation can also be run in parallel by entering **`mpirun -n # python generate-mesh.py`**, where **`#`** is replaced with the number of processors.

- **plot.py**
  This file reads simulation output from the **`output/data`** folder and plots the damage, stresses, and velocities at finite element nodes. All plots are saved in PNG format in the **`output/figs`** directory. In order to run the simulation in serial, the user needs to enter **`python plot.py`** into the computer terminal. Upon running, the program will prompt the user to input the number of the data file to be plotted. The user should only supply the number and not the ".h5" file extension.

- **quad-plot.py**
  This file reads simulation output from the **`output/data`** folder and plots the damage stresses at finite element quadrature (integration) points, and the velocities at finite element nodes. All plots are saved in PNG format in the **`output/figs`** directory. In order to run the simulation in serial, the user needs to enter **`python plot.py`** into the computer terminal. Upon running, the program will prompt the user to input the number of the data file to be plotted. The user should only supply the number and not the ".h5" file extension.