Fenics Ice Sheet Model Readme

March 17, 2020

This document briefly outlines how to get started with the Fenics ice sheet model.

1 Installation

The ice sheet mdoel is built using the open source Python finite element software Fenics, and depends on the package tlm-adjoint for implementing inversion and error propagation capabilities. The simplest way to install Fenics and tlm-adjoint is to create a conda environment.

1.1 Installing Fenics

1. Install Anaconda. This can be either Anaconda itself, or miniconda, which is a stripped down version. Ensure the Python version is greater than 3.6. Installer can be found here: https://www.anaconda.com/distribution/

2. Create a new conda environment.
conda create -n fenics -c conda-forge fenics fenics-dijitso fenics-dolfin fenics-ffc fenics-fiat fenics-libdolfin fenics-ufl

3. Enter the conda environment:
source activate fenics

4. Make sure the pip package manager is up to date:
pip install --upgrade pip

5. Install the following packages:
conda install matplotlib numpy ipython scipy

6. Install hdf5 for python:
`http://docs.h5py.org/en/latest/index.html`
pip install h5py

7. Install pyrevolve:
`https://github.com/opesci/pyrevolve`

Change to directory where you would like to download pyrevolve to. You can delete the pyrevolve directory after finishing this step.

git clone `https://github.com/opesci/pyrevolve.git`
cd pyrevolve/
python setup.py install

8. Install mpi4py:
`http://mpi4py.scipy.org/docs/`
pip install mpi4py

9. To enter this environment:
source activate fenics

10. To exit:
source deactivate fenics

1.2 tlm_adjoint

1. Clone the git repository to the local drive where you want it to live:
git clone `https://github.com/jrmaddison/tlm_adjoint.git`

1.3 Fenics ICE

1. Clone the git repository to the local drive where you want it to live:
git clone `https://github.com/cpk26/fenics_ice.git`

1.4 Create environment variable

Create an environment variable storing the fenics_ice base directory by adding the following to .bashrc, amending the path appropriately for your system.
FENICS_ICE_BASE_DIR="/XXXX/XXXX/XXXX/fenics_ice"
export FENICS_ICE_BASE_DIR

1.5 Modifying the Python Path

Modify the default paths python looks for modules to include tlm_adjoint and fenics ice.
Add to the end of .bashrc:
PYTHONPATH="${PYTHONPATH}:/PATH/TO/tlm_adjoint/python:/PATH/TO/fenics_ice/code"
export PYTHONPATH

2 Program structure

2.1 Directory Structure

```
fenics_ice
├── code
│   ├── model.py
│   └── solver.py
├── runs
│   ├── process_eigendec.py
│   ├── run_balancemeltrates.py
│   ├── run_eigendec.py
│   ├── run_errorprop.py
│   ├── run_forward.py
│   ├── run_inv.py
│   ├── run_invsigma.py
│   └── run_momsolve.py
├── scripts
│   └── ismipc
├── aux
│   ├── gen_ismipC_domain.py
│   └── test_domains.py
├── input
│   └── ismipc
└── output
    └── ismipc
```

2.2 Overview

The core of the ice sheet model is in two files: `/code/model.py` and `/code/solver.py`. These are utilized by the python scripts in the `/runs` folder, which execute specific parts of a simulation. The scripts there are generic to any simulation. Each specific simulation then has its own primary folder in the `/scripts` folder, which will call program files in tt /runs with specific parameters and data files.

The bash scripts in `/scripts` are where parameters and data file locations are specified. The data and parameters are used by the program files in `/runs` to create a model object (via a class defined in `model.py`) and subsequently a solver object (via a class defined in `solver.py`). The model object contains all the necessary data for a simulation, such as topography, constants, and velocity observations for inversions. The solver object contains the ice sheet physics/inversion code. The model object is passed as a parameter to your solver object. This object then allows you to solve the SSA equations on your domain, invert for basal drag or B_{glen}, and perform uncertainty quantification.

The /aux folder contains auxillary files; in here, the file gen_ismipC_domain.py generates the ismipC domain, based off definitions in test_domains.py. The /input folder is where input files, such as topography and ice thickness, for specific simulations are located. Similarily, the /output folder is where output is stored from specific simulations.

3 Tutorial: A walkthrough IsmipC

Begin by navigating to the /fenics_ice directory. Activate the fenics conda environment.

```
> source activate fenics
```

3.1 Generate synthetic domain

```
> cd $FENICS_ICE_BASE_DIR/aux
> python gen_ismipC_domain.py -o ../input/ismipC -L 40000 -nx 100 -ny 100

> ls $FENICS_ICE_BASE_DIR//input/ismipC
B2.xml  Bglen.xml  alpha.xml  bed.xml  bmelt.xml  grid_data.npz  mask.xml  mesh.xml  smb.xm
```

3.2 Solve the forward problem and generate observations

```
> cd $FENICS_ICE_BASE_DIR/scripts/ismipc/
> ./forward_solve.sh

Generating new mesh
Building point search tree to accelerate distance queries.
Computed bounding box tree with 39999 nodes for 20000 points.
Solving nonlinear variational problem.
Newton iteration 0: r (abs) = 1.585e+03 (tol = 1.000e-08) r (rel) = 1.000e+00 (tol = 5.000e
Newton iteration 1: r (abs) = 1.139e+02 (tol = 1.000e-08) r (rel) = 7.186e-02 (tol = 5.000e
Newton iteration 2: r (abs) = 1.307e+02 (tol = 1.000e-08) r (rel) = 8.248e-02 (tol = 5.000e
Newton iteration 3: r (abs) = 9.443e+01 (tol = 1.000e-08) r (rel) = 5.958e-02 (tol = 5.000e
Newton iteration 4: r (abs) = 5.682e+01 (tol = 1.000e-08) r (rel) = 3.585e-02 (tol = 5.000e
Newton solver finished in 5 iterations and 5 linear solver iterations.
Solving nonlinear variational problem.
Newton iteration 0: r (abs) = 6.650e+01 (tol = 1.000e-05) r (rel) = 1.000e+00 (tol = 1.000e
Newton iteration 1: r (abs) = 4.913e+00 (tol = 1.000e-05) r (rel) = 7.387e-02 (tol = 1.000e
Newton iteration 2: r (abs) = 4.393e-02 (tol = 1.000e-05) r (rel) = 6.606e-04 (tol = 1.000e
Newton iteration 3: r (abs) = 5.647e-06 (tol = 1.000e-05) r (rel) = 8.492e-08 (tol = 1.000e
Newton solver finished in 4 iterations and 4 linear solver iterations.
Time for solve: 4.667648553848267
```

```
ls $FENICS_ICE_BASE_DIR/input/ismipC
...
```

Notice that there are many more files in the directory now.

Next we'll convert our modelled velocities into pseudo-observations. The script