Fenics Ice Sheet Model User Guide

March 18, 2020

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

Contents

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 Installing 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 Installing 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 Creating environment variables

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 python scripts there are generic to any simulation. Each specific simulation then has its own primary folder in the `/scripts` folder, with simple bash scripts which call program files in `/runs` with specific parameters and data files.

The bash scripts in `/scripts` are where parameters and data file locations are specified; these are bash simple wrapper scripts for calling python scripts in `/runs`. 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 options of any python script in the `/runs` folder can be viewed by typing 'python run_xxx.py –help'.

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 of IsmipC

The Ice Sheet Model Intercomparison Project for Higher-Order ice sheet Models (ISMIP) provides a standardized set of idealized tests for ice sheet models. In this walkthrough, we apply Fenics_ice to the domain prescribed by experiment C (IsmipC). The original IsmipC is a static simulation, meaning time evolution is not considered. We'll extend it by running a dynamic simulation for the purposes of performing uncertainty quantification.

4 IsmipC

4.1 Generating the Domain

Navigate to the `/fenics_ice` base directory. Activate the fenics conda environment.

```
> source activate fenics
```

To begin, we'll generate the synthetic domain defined by the IsmipC experiment. The specifications are coded in the file `/aux/test_domains.py`. We'll use the python script 'gen_ismipC_domain.py' to create a domain with a given length and resolution.

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

This will generate a square domain with side-length 40km, at resolution of 100 x 100, placing the output in the folder `input/ismipc`. Let's observe the files that are generated.

```
> 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.xml  thick.xml
```

The .xml files contain discretized scalar fields over the IsmipC domain on a FEniCS mesh. The extension .npz indicates a numpy file format, and contains the domain resolution and length.

- B2.xml – β^2 coefficient for linear sliding law ($\boldsymbol{\tau_b} = \beta^2 \boldsymbol{u}$);
- Bglen.xml – parameter in Glen's flow law
- alpha.xml – variable in sliding law
- bed.xml – basal topography
- bmelt.xml – basal melt.
- mask.xml – mask of our domain
- mesh.xml – FEniCS mesh
- smb.xml – surface mass balance
- thick.xml – ice thickness

4.2 Solving the Momentum Equations

Having generated the files which describe our domain, we can solve the SSA momentum equations to determine ice velocities.

```
> 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-02)
Newton iteration 1: r (abs) = 1.139e+02 (tol = 1.000e-08) r (rel) = 7.186e-02
(tol = 5.000e-02)
Newton iteration 2: r (abs) = 1.307e+02 (tol = 1.000e-08) r (rel) = 8.248e-02
(tol = 5.000e-02)
Newton iteration 3: r (abs) = 9.443e+01 (tol = 1.000e-08) r (rel) = 5.958e-02
(tol = 5.000e-02)
Newton iteration 4: r (abs) = 5.682e+01 (tol = 1.000e-08) r (rel) = 3.585e-02
(tol = 5.000e-02)
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-05)
Newton iteration 1: r (abs) = 4.913e+00 (tol = 1.000e-05) r (rel) = 7.387e-02
(tol = 1.000e-05)
Newton iteration 2: r (abs) = 4.393e-02 (tol = 1.000e-05) r (rel) = 6.606e-04
(tol = 1.000e-05)
Newton iteration 3: r (abs) = 5.647e-06 (tol = 1.000e-05) r (rel) = 8.492e-08
(tol = 1.000e-05)
Newton solver finished in 4 iterations and 4 linear solver iterations.
Time for solve: 4.667648553848267

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

The script automatically places the output in the subdirectory of `input/` since we'll use the velocities we solved for in the next step, generating synthetic observations.

Opening `forward_solve.sh` with any text editor, we can confirm that this is a simple wrapper script.

```
#!/bin/bash
set -e

BASE_DIR=$FENICS_ICE_BASE_DIR
```

```
RUN_DIR=$BASE_DIR/runs

INPUT_DIR=$BASE_DIR/input/ismipC
OUTPUT_DIR=$INPUT_DIR/momsolve

cd $RUN_DIR

python run_momsolve.py -b -q 0 -d $INPUT_DIR -o $OUTPUT_DIR
```

The bash script specifies key folders, that we are solving momentum equations on a domain with periodic boundary conditions (-b option), and that we are using a linear sliding law (-q 0).

4.3 Generating Synthetic Observations

IsmipC is a synthetic experiment, meaning we don't have observational data of ice velocities. We can generate pseudo-oberservations by adding gaussian noise to the solved velocities. This script assumes the noise is additive rather than a multiplicative factor. The python script 'Uobs_from_momsolve.py' takes the vector field in U.xml and generates the files: data_mesh.xml, data_mask.xml, u_obs.xml, v_obs.xml, u_std.xml, and v_std.xml. The first file contains the data mesh, the second identifies where velocity data is available, the next two files contain the pseudo-observations in the x and y directions, with the final two files containing the standard deviation of the gaussian noise applied. A data_mask.xml and data_mask.pvd file are generated by forward_solve.sh since these are duplicates of the domain mask.

```
> cd $FENICS_ICE_BASE_DIR/aux/
> python Uobs_from_momsolve.py -b -L 40000 -d $FENICS_ICE_BASE_DIR/input/ismipC/momsolve
> find $FENICS_ICE_BASE_DIR/input/ismipC/momsolve -type f -regex '.*\(obs\|std\).xml'
/mnt/c/Users/ckozi/Documents/Python/fenics/fenics_ice/input/ismipC/u_obs.xml
/mnt/c/Users/ckozi/Documents/Python/fenics/fenics_ice/input/ismipC/u_std.xml
/mnt/c/Users/ckozi/Documents/Python/fenics/fenics_ice/input/ismipC/v_obs.xml
/mnt/c/Users/ckozi/Documents/Python/fenics/fenics_ice/input/ismipC/v_std.xml
```

Copy the five files generated into `$FENICS_ICE_BASE_DIR/input/ismipC/`. A study site in Antarctica or Greenland would require generating these files and the data_mask from a surface velocity dataset such as NSIDC MEaSUREs.

```
> cd $FENICS_ICE_BASE_DIR/input/ismipC/momsolve
> cp data_mask.xml data_mesh.xml mask_vel.xml u_*.xml v_*.xml ..
```