# 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	Inst	allation	
	1.1	Installing Fenics	
	1.2	Installing tlm_adjoint	
	1.3	Installing Fenics Ice	
	1.4	Creating environment variables	
	1.5	Modifying the Python Path	
<b>2</b>	Program structure		
	2.1	Directory Structure	
		Overview	
3	Tut	orial: A Walkthrough of IsmipC	
4	Ism	$\mathbf{ipC}$	
	4.1	Generating the Domain	
	4.2	Solving the Momentum Equations	
	4.3	Generating Synthetic Observations	

# 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-fic 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

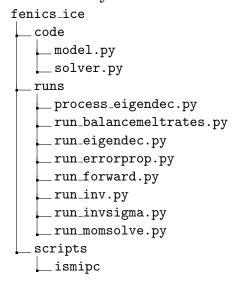
### 1.5 Modifying the Python Path

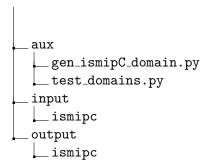
Modify the default paths python looks for modules to include tlm\_adjoint and fenics ice. Add to the end of .bashrc:

 $\label{eq:python} PYTHONPATH=``\$\{PYTHONPATH\}:/PATH/TO/tlm\_adjoint/python:/PATH/TO/fenics\_ice/code`` export PYTHONPATH$ 

# 2 Program structure

### 2.1 Directory Structure





#### 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 <code>gen\_ismipC\_domain.py</code> generates the ismipC domain, based off definitions in <code>test\_domains.py</code>. The /input folder is where input files, such as topography and ice thickness, for specific simulations are located. Similarly, 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  $\beta^2$  coefficient for linear sliding law  $(\tau_b = \beta^2 u)$ ;
- 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 \text{ (abs)} = 4.913e+00 \text{ (tol} = 1.000e-05) r \text{ (rel)} = 7.387e-02
(tol = 1.000e-05)
Newton iteration 2: r \text{ (abs)} = 4.393e-02 \text{ (tol} = 1.000e-05) r \text{ (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 ..