

Fenics Ice Sheet Model Readme

January 18, 2018

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

1 Installing Fenics + dolfin-adjoint

The ice sheet model is built using the open source Python finite element software Fenics, and depends on the package dolfin-adjoint for implementing inversion and error propagation capabilities. The simplest way to install Fenics and dolfin-adjoint is to use a pre-built docker image. Installation instructions are available online via the Fenics website at: <http://fenics.readthedocs.io/projects/containers/en/latest/>.

1.1 Eddie

On Eddie, the following procedure was used successfully:

1. Login into Eddie, go onto worker node. It was recommended to specify the memory usage:

```
qlogin -pe interactivemem 4 -l h_vmem=4G
```

2. Install Anaconda. This can be either Anaconda itself, or miniconda, which is a stripped down version. I went for the latter, Python 2.7. <https://conda.io/miniconda.html> for the link, or

```
wget https://repo.continuum.io/miniconda/Miniconda2-latest-Linux-x86_64.sh
```

3. It takes up a little space, so I installed it in the iceocean group directory:

```
/exports/csce/eddie/geos/groups/geos_iceocean/ckoziol
```

4. During the install it asks whether you want to change the pythonpath, I said no.

5. At the end it gives you an export command, I added this to my .bashrc

```
export PATH=/exports/csce/eddie/geos/groups/...
geos_iceocean/ckozio1/miniconda2/bin:\$PATH
```

6. Installing dolfin adjoint:

```
conda create -n dolfinproject -c conda-forge dolfin-adjoint=2016.2.0
```

Note, I couldnt get the latest version 2017.1.0 to work.

This creates an environment 'dolfinproject' which will contain dolfin/fenics. (note can use a different name).

7. to enter the environment, we use `source activate dolfinproject`

8. then you can type `python`, and `from dolfin_adjoint import *`

2 Program structure

The core of the ice sheet model is in two files: `/code/model.py` and `/code/solver.py`. Each simulation then has its own main routine located in the `/runs` folder. The run routine loads any necessary data specific to your simulation, and is where you can specify the parameters of your run. These data and parameters are then used to create a model object (via a class defined in `model.py`), which contains all the necessary data for a simulation, such as topography, constants, and velocity observations for inversions. The ice sheet physics/inversion code is contained within in `solver.py`, which contains the code to create a solver object. The model object created in your run file is then passed as a parameter to your solver object. This object then allows you to solve the SSA equations on your domain, as well perform an inversion.

For an example of a run file, see `/runs/run_inv_smith.py`. This is under development, and is the most up to date at present.