Fenics Ice Sheet Model Readme

January 28, 2020

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

1 Installing Fenics and tlm_adjoint

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 tp to date the pip package manager:
```
pip install –upgrade pip
```

5. Install the following packages:
```
pip 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

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

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.