ISSM Workshop 2014

Ice Sheet System model

Application to Pine Island Glacier

Basile DE FLEURIAN[1],
BASED ON ISSM TEAM WORK,

[1] University of California, Irvine

Modeling Flowchart

The aim of the exercise is to present the different step that should be taken to build an operational simulation.

- Define the modelling region
  - Define a region of interest from a velocity map and preceding shape file
- Create a Mesh
  - Create a mesh optimized to deal with the observed velocities
- Create masks
  - Use SeaRISE thickness to specify floating vs grounded elements
- Parameterize model
  - To treat the ice flow with a SSA based model
- Control method for basal drag
  - Inverting for the friction coefficient
- Plot results
  - Observed velocity, modelled velocities, bed elevation, friction coefficient ...

File description

ISSM needs a combination of four types of files to perform a simulation. For this case they are gathered in `workshop/06PIG_Application` and `workshop/Data`

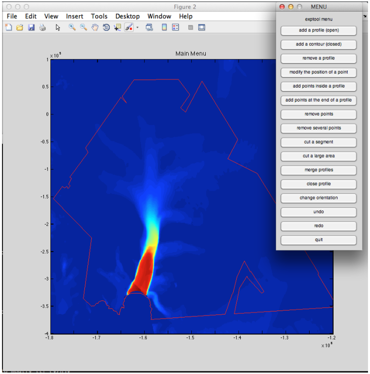
- Matlab “runme.m” file
  - Structure of the simulation
- Matlab “.par” file
  - Most of the ’standard’ parameters of a simulation
- Shape files “.exp”
  - Shape files needed to define geometric boundaries for the simulation
- Data files
  - Usually NetCDF files which are loaded by other routines
  - Could also be whatever other format that you could read with Matlab (Python)

Setting-up domain outline

We are drawing a new domain for PIG function of the surface velocities and an existing shapefile (`./DomainOutline.exp`)

Here starts the Matlab work

- `PigRegion`
  velocity map in the PIG vicinity
- `exptool('Name')`
  `Name=./DomainOutline.exp`
  Open the shapefile and allow to modify it



TIPS

- Re-size the map from `PigRegion` before opening `exptool`
- Avoid to change the front line for this application

Step 1: Mesh

Work is now taking place in the `runme.m` file. The mesh generation is a two step process

- We first create a uniform initial mesh of given resolution (`hinit`) with `bamg`

```
% Generate an initial uniform mesh (resolution = hinit m)
md=bamg(model,'domain',domain,'hmax',hinit,'MaxCornerAngle',1);
```

This first command creates a new ISSM model (`md`) with a mesh of the provided PIG domain

- The mesh is then improved by an anisotrop refinement. We use here the surface velocity as a control parameter for the refinement

```
vy_obs=InterpFromGridToMesh(x,y,flipud(vy'),md.mesh.x,md.mesh.y,0);
vel_obs=sqrt(vx_obs.^2+vy_obs.^2);
clear vx vy x y;
```

Step 2: Mask

Specify grounded vs ungrounded areas of the PIG domain.

- ISSM default: ice is grounded
- `md.mask` is the model field containing the mask information
  - Whether there is ice or not [-1;1]
  - Weather the ice is grounding floating or we are at grounding line [1;0;-1]

```
thkmask=double(ncread(searise,'thkmask'));

%interpolate onto our mesh vertices
groundedice=double(InterpFromGridToMesh(x1,y1,thkmask',md.mesh.x,md.mesh.y,0));
```

TIPS

Documentation is present for each model field. Typing the model field in Matlab will give you the sufields and short explanation

Step 3: Parameterize Model 1/2

Parameterization of the models is usually done through a different file (`./Pig.par`).
Parameters which are unlikely to be change for a given set of experiments are fixed here. to lighten the `runme` file.
In this example we use the SeaRise data to parameterize the following model fields.

- Geometry
  - Only define two variables and compute the third
  - `find` function allow to define subsets of nodes
- Initialization parameters
- Material parameters
- Forcings
- Friction and inversion set up
- Boundary Conditions

Step 4: Control Method for Coefficient of Friction

The friction coefficient β is inversed from the surface velocities

$$\tau_b = -\beta^2 N^r \|\mathbf{v_b}\|^{s-1} \mathbf{v_b}$$

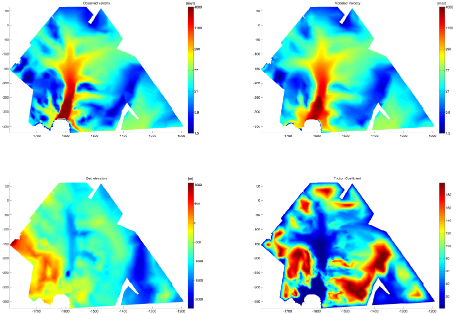
- τ_b : Basal drag
- N : Effective pressure
- v_b : Basal velocity (equal surface in SSA approximation)
- r : Exponent (equals q/p of the parameter file)
- s : Exponent (equals $1/p$ of the parameter file)

The procedure for the inversion method is as follow

- Velocity is computed from the SSA approximation
- Misfit of the cost function is computed
- Friction coefficient is modified
- $\vdots$

Step 5: Plot Results

Plotting ability is from `plotmodel` for simple graphs. But can use every plotting ability from matlab.



TIPS

`plotdoc` give the more usual options for `plotmodel`

Step 6: Higher Order (HO) Ice Flow Model

Set up of the HO ice flow model

- Load the initial step
  - model to load is `Control_drag`
- Disable the inversion process
  - put to zero the inversion flag (cf. `md.inversion`)
- Extrude the mesh
  - `help extrude`
  - keep the number of layer low to avoid to long computation
- Change the flow model
  - `setflowequation`
- Solve
  - we are still solving for a `StressBalanceSolution`
- save the model
  - as in the preceding