

## Ice Sheet System model

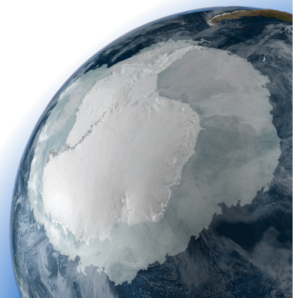
### Application to IceBridge dataset, Greenland Ice Sheet

Eric LAROUC<sup>1</sup>, Eric RIGNOT<sup>1,2</sup>, Mathieu MORLIGHEM<sup>2</sup>, Hélène SEROUSSI<sup>1</sup>  
Nicole SCHLEGEL<sup>1</sup>, Chris BORSTAD<sup>1</sup>, Feras HABBAL<sup>1,2</sup>, Daria HALKIDES<sup>1,3</sup>,  
Gilberto PEREZ<sup>1</sup>, John SCHIERMEIER<sup>1</sup>, **Michiel HELSEN**<sup>2</sup>

<sup>1</sup> Jet Propulsion Laboratory - California Institute of Technology

<sup>2</sup> University of California, Irvine

<sup>3</sup> Joint Institute for Regional Earth System Science & Engineering, UCLA



Application IceBridge

Larour et al.

Introduction

Mesh

Parameterization

Control Method  
Solution

Transient Solution

Plot Results

# Overview

- 1 Introduction
- 2 Mesh
- 3 Parameterization
- 4 Control Method Solution
- 5 Transient Solution
- 6 Plot Results

## Goals

Use your new ISSM skills to adjust a coarse Greenland model by adding IceBridge data.

We refine the mesh in the Jakobshavn basin and add higher resolution bedrock and surface elevation data from IceBridge in this area.

### Steps:

- Refine Greenland mesh using given Jakobshavn outline
- Parameterize, include the high-resolution IceBridge bedrock and surface data
- Plot bedrock and surface data
- Diagnostic: run 2 inverse method runs to solve for control drag (20 steps recommended)
- Transient: run 20 year runs, with coarse and refined bedrock and surface elevation data
- Plot transient results

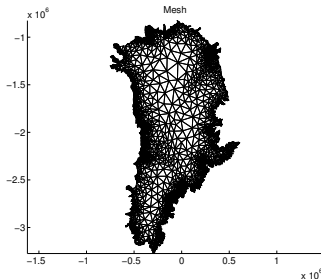
## First Run Step: Mesh

We modify the experiment from the Greenland SeaRISE talk, and improve from there.

First, run the first step in `runme.m` in directory `13_IceBridge` to mesh the Greenland domain as done in the previous talk.

Step 1 is interrupted after making the default mesh. Plot the result.

It should look like this:



Application IceBridge

Larour et al.

Introduction

Mesh

Parameterization

Control Method  
Solution

Transient Solution

Plot Results

## Mesh, 2/5

### Adapt

We want to refine the mesh in the Jakobshavn area. An outline of this area **Jak\_outline.exp** can be found in directory **Exp\_Par**.

Try using **exptool** to view this outline:

```
1 >> exptool('Exp_Par/Jak_outline.exp');
```

Application IceBridge

Larour et al.

Introduction

Mesh

Parameterization

Control Method  
Solution

Transient Solution

Plot Results

## Mesh, 3/5

### Adapt

Next, we modify the `bamg` command by imposing a 1 km resolution within the Jakobshavn area: using `hmaxVertices`.

Note that you need to deactivate the previous `bamg` command.

```
26 %Refine mesh in the region of Jakobshavn (resolution = 1000 m)
27 hmaxVertices=NaN*ones(md.mesh.numberofvertices,1);
28 in=ContourToMesh(md.mesh.elements,md.mesh.x,md.mesh.y,...
29 './Exp_Par/Jak_outline.exp','node',1);
30 hmaxVertices(find(in))=1000;
31 md=bamg(md,'hmax',400000,'hmin',5000,'gradation',1.7,'field',vel,...
32 'err',8,'hmaxVertices',hmaxVertices);
```

Check your results using `plotmodel`.

Application IceBridge

Larour et al.

Introduction

Mesh

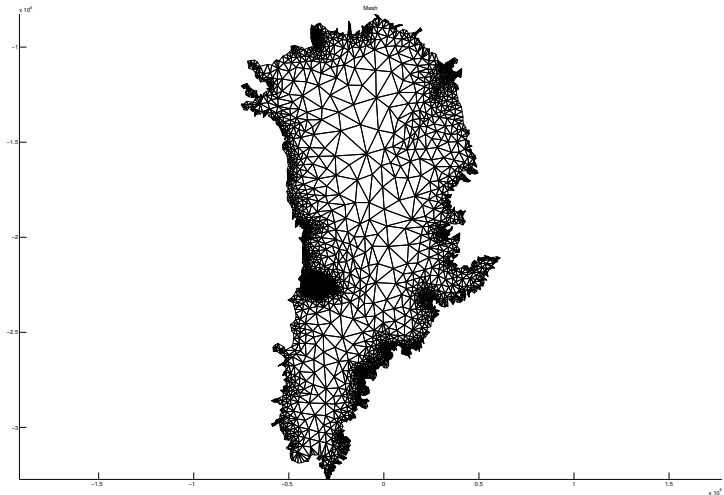
Parameterization

Control Method  
Solution

Transient Solution

Plot Results

## Mesh, 4/5 Solution



Try **zoom** to make a close-up of the Jakobshavn domain.

Application IceBridge

Larour et al.

Introduction

Mesh

Parameterization

Control Method  
Solution

Transient Solution

Plot Results

## Mesh, 5/5

### Full Solution

```
22 % %Mesh greenland without refinement in Jak basin
23 % md=bamg(md,'hmax',400000,'hmin',5000,'gradation',1.7,'field',vel,'err',8);
24 % return
25
26 %Refine mesh in the region of Jakobshavn (resolution = 1000 m)
27 hmaxVertices=NaN*ones(md.mesh.numberofvertices,1);
28 in=ContourToMesh(md.mesh.elements,md.mesh.x,md.mesh.y,...
29     './Exp_Par/Jak_outline.exp','node',1);
30 hmaxVertices(find(in))=1000;
31 md=bamg(md,'hmax',400000,'hmin',5000,'gradation',1.7,'field',vel,...
32     'err',8,'hmaxVertices',hmaxVertices);
33
34 %convert x,y coordinates (Polar stereo) to lat/lon
35 [md.mesh.lat,md.mesh.long]=xy2ll(md.mesh.x,md.mesh.y,+1,39,71);
36
37 save ./Models/Greenland.Mesh_generation md;
38 end
```

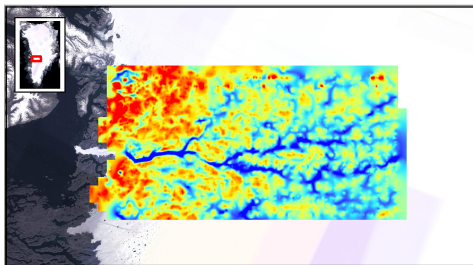


## Parameterization

We want to include high-resolution bedrock and surface elevation data acquired in the IceBridge mission. Accessible on:

<https://data.cresis.ku.edu/data/grids/>

**Jakobshavn\_2008\_2011\_Composite\_XYZGrid.txt**



Ice Bottom (m)



Jakobshavn  
Bottom



## Parameterization, 2/6

Bedrock data is read, transformed into a usable grid, and interpolated to the mesh in the parameter file **Exp\_Par/Greenland.par**:

```
30 %Reading IceBridge data for Jakobshavn
31 disp('      reading IceBridge Jakobshavn bedrock');
32 fid = fopen('Data/Jakobshavn_2008_2011_Composite_XYZGrid.txt');
33 data = fscanf(fid, '%g,%g,%g,%g,%g',[5 266400]);
34 fclose(fid);
35
36 [xi,yi]= ll2xy(md.mesh.lat,md.mesh.long,+1,45,70);
37 bed = flipud(reshape(data(:,5),[360 740])); bed(find(bed==-9999))=NaN;
38 bedy = flipud(reshape(data(:,1),[360 740]));
39 bedx = flipud(reshape(data(:,2),[360 740]));
40
41 %Insert Icebridge bed and recalculate thickness
42 bed_jks=InterpFromGridToMesh(bedx(1,:)',bedy(:,1),bed,xi,yi,NaN);
43 in=ContourToMesh(md.mesh.elements,md.mesh.x,md.mesh.y,...
44     './Exp_Par/Jak_grounded.exp','node',1);
45 bed_jks(~in)=NaN;
46 pos=find(~isnan(bed_jks));
47 md.geometry.bed(pos)=bed_jks(pos);
```

Modify **Greenland.par** such that the surface elevation data is also included for the Jakobshavn area.

## Parameterization, 3/6

### Solution

```
30 %Reading IceBridge data for Jakobshavn
31 disp('          reading IceBridge Jakobshavn bedrock');
32 fid = fopen('Data/Jakobshavn_2008_2011_Composite_XYZGrid.txt');
33 data = fscanf(fid, '%g,%g,%g,%g,%g', [5 266400]);
34 fclose(fid);
35
36 [xi,yi]= ll2xy(md.mesh.lat,md.mesh.long,+1,45,70);
37 bed = flipud(reshape(data(:,5),[360 740])); bed(find(bed== -9999))=NaN;
38 surf = flipud(reshape(data(:,4),[360 740])); surf(find(surf== -9999))=NaN;
39 bedy = flipud(reshape(data(:,1),[360 740]));
40 bedx = flipud(reshape(data(:,2),[360 740]));
41
42 %Insert Icebridge bed and recalculate thickness
43 bed_jks=InterpFromGridToMesh(bedx(1,:)',bedy(:,1),bed,xi,yi,NaN);
44 surf_jks=InterpFromGridToMesh(bedx(1,:)',bedy(:,1),surf,xi,yi,NaN);
45 in=ContourToMesh(md.mesh.elements,md.mesh.x,md.mesh.y,...
46     './Exp_Par/Jak_grounded.exp','node',1);
47 bed_jks(~in)=NaN;
48 surf_jks(~in)=NaN;
49 pos=find(~isnan(bed_jks));
50 md.geometry.bed(pos)=bed_jks(pos);
51 md.geometry.surface(pos)=surf_jks(pos);
52 md.geometry.thickness=md.geometry.surface-md.geometry.bed;
```

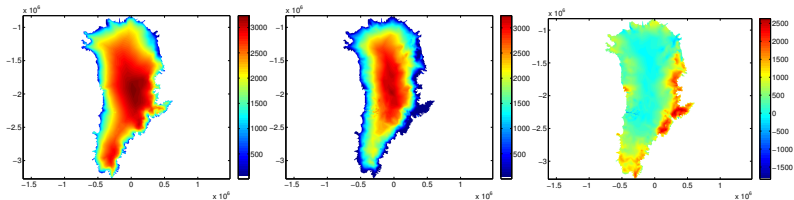
Plot surface elevation, thickness and bedrock.

## Parameterization, 4/6

### Solution

```
1 >> plotmodel(md, 'data', md.geometry.surface);  
2 >> plotmodel(md, 'data', md.geometry.thickness);  
3 >> plotmodel(md, 'data', md.geometry.bed);
```

They should look like:



## Parameterization, 5/6

### Solution

To plot the difference in bed topography between the SeaRISE and IceBridge datasets:

Modify the parameterization step in your runme and save the model under a different name.

a difference in fields can be plotted using:

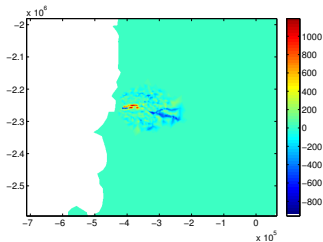
```
1  >> md2=loadmodel('Models/Greenland.parameterization2');  
2  >> md=loadmodel('Models/Greenland.parameterization');  
3  >> plotmodel(md, 'data', md.geometry.bed-md2.geometry.bed);
```

## Parameterization, 6/6

### Solution

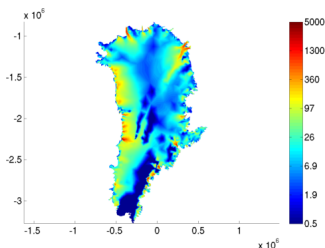
a difference in fields can be plotted using:

```
1 >> md2=loadmodel('Models/Greenland.parameterization2');  
2 >> md=loadmodel('Models/Greenland.parameterization');  
3 >> plotmodel(md,'data',md.geometry.bed-md2.geometry.bed);
```



## Diagnostic

Use control methods to inversely solve for Greenland friction coefficient  
Comparable to 12\_SeaRISE experiment



The observed velocity map contains some gaps; exclude these from the inversion by creating a new exp file that outlines the gaps in the velocity data, using **exptool**:

```
1 >> exptool('Exp_Par/data_gaps.exp');
```

## Diagnostic, 2/3

```
74 in=ContourToMesh(md.mesh.elements,md.mesh.x,md.mesh.y,'./Exp_Par/data_gaps.exp',  
75 md.inversion.cost_functions_coefficients(find(in),1)=0.0;  
76 md.inversion.cost_functions_coefficients(find(in),2)=0.0;
```

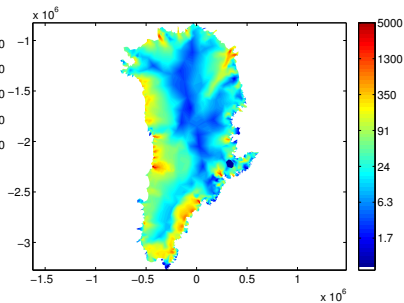
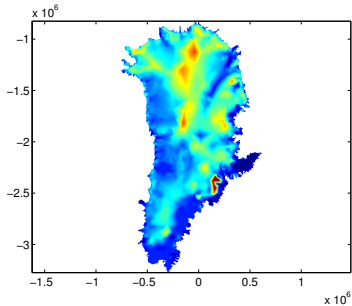
Next, run the model and plot the resulting friction coefficient and velocity pattern

```
1 >> plotmodel(md,'data',md.results.DiagnosticSolution.FrictionCoefficient);  
2 >> plotmodel(md,'data',md.results.DiagnosticSolution.Vel,...  
3 'log',10,'caxis',[0.5 5000]);
```



## Diagnostic, 3/3

```
1 >> plotmodel(md, 'data', md.results.DiagnosticSolution.FrictionCoefficient);  
2 >> plotmodel(md, 'data', md.results.DiagnosticSolution.Vel, ...  
3     'log', 10, 'caxis', [0.5 5000]);
```



Application IceBridge

Larour et al.

Introduction

Mesh

Parameterization

Control Method  
Solution

Transient Solution

Plot Results

## Transient Forcing

Do a transient run..

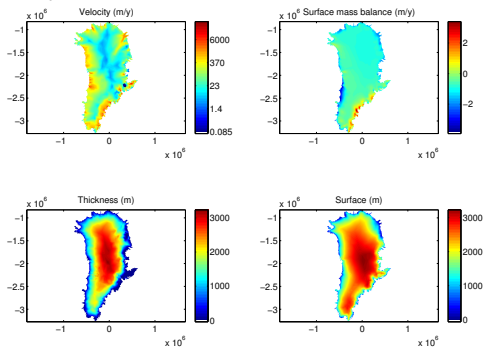
# Transient Results

## Plot Plan

Your results are located in **md.results.TransientSolution**. Plot your results.

First, plot the initial plan view of velocity, surface mass balance, thickness, and surface.

They should look like:



Application IceBridge

Larour et al.

Introduction

Mesh

Parameterization

Control Method  
Solution

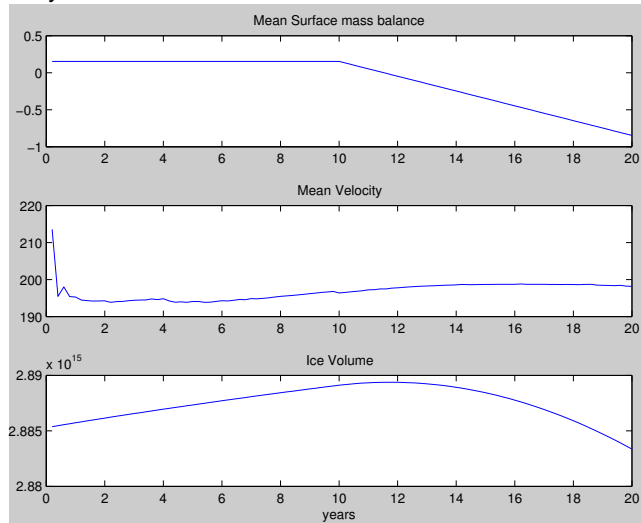
Transient Solution

Plot Results

## Transient Results

Plot time series

They should look like:



## Transient Results

Some suggestions what to explore further:

- How would you make a plot of time series of results from the SeaRISE and IceBridge experiments?
- How would you make a plot of the difference between final and initial ice thickness?
- ...

We can help you to implement your own ideas in the code.

Thanks!

