

## Ice Sheet System model

### ISMIP-HOM

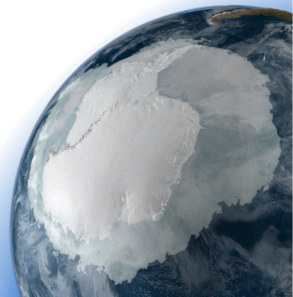
Eric LAROUR<sup>1</sup>, Eric RIGNOT<sup>1,3</sup>, Mathieu MORLIGHEM<sup>1,2</sup>, **Hélène SEROUSSI**<sup>1,2</sup> Chris BORSTAD<sup>1</sup>, Feras HABBAL<sup>1,3</sup>, Daria HALKIDES<sup>1,4</sup>, Behnaz KHAKBAZ<sup>1</sup>, John SCHIERMEIER<sup>1</sup>, Nicole SCHLEGEL<sup>1</sup>

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

<sup>2</sup>Laboratoire MSSMat, École Centrale Paris, France

<sup>3</sup>University of California, Irvine

<sup>4</sup>Joint Institute for Regional Earth System Science & Engineering, UCLA



ISMIP-HOM

Larour et al.

## Outline

### ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

### ISMIP-HOM Test F

Test Description

Transient model

Results

#### ① ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

#### ② ISMIP-HOM Test F

Test Description

Transient model

Results

ISMIP-HOM

Larour et al.

## Test A

### ISMIP-HOM Test A

#### Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

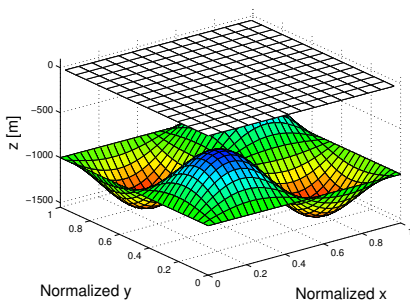
### ISMIP-HOM Test F

Test Description

Transient model

Results

## Square ice sheet flowing over a bumpy bed



- Sinusoidal bedrock
- Ice frozen on the bed
- Periodic boundary conditions

Description of test in `ismiphom_description.pdf`

ISMIP-HOM

Larour et al.

## Mesh

ISMIP-HOM Test A

Test Description

**Mesh**

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

ISMIP-HOM Test F

Test Description

Transient model

Results

Use `squaremesh` to create the mesh:

```
1  >> help squaremesh
2      SQUAREMESH - create a structured square mesh
3
4      This script will generate a structured square mesh
5      Lx and Ly are the dimension of the domain (in meters)
6      nx anx ny are the number of nodes in the x and y direction
7      The coordinates x and y returned are in meters.
8
9      Usage:
10         [md]=squaremesh (md,Lx,Ly,nx,ny)
```

Example for 80 km and with 20 points in each direction

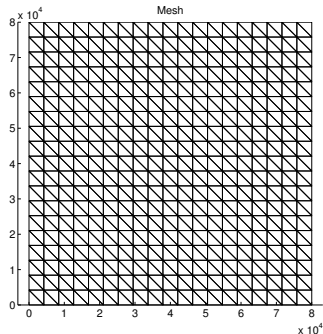
ISMIP-HOM

Larour et al.

## Mesh

Example for 80 km and with 20 points in each direction:

```
1 >> md=squar mesh (md,80000,80000,20,20)
2 >> plotmodel (md,'data','mesh')
```



ISMIP-HOM

Larour et al.

# Set Mask

ISMIP-HOM Test A

Test Description

Mesh

**Mask**

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

ISMIP-HOM Test F

Test Description

Transient model

Results

## Use `setmask` to specify that everything is grounded ice

```
1  >> help setmask
2  SETMASK- establish boundaries between grounded and floating ice.
3
4  By default, ice is considered grounded. The contour floatingicename defines nodes
5  for which ice is floating. The contour groundedicename defines nodes inside an floatingice,
6  that are grounded (ie: ice rises, islands, etc ...)
7  All input files are in the Argus format (extension .exp).
8
9  Usage:
10     md=setmask(md,floatingicename,groundedicename)
11
12  Examples:
13     md=setmask(md,'all','');
14     md=setmask(md,'Iceshelves.exp','Islands.exp');
```

```
>> md=setmask(md,'','')
```

ISMIP-HOM

Larour et al.

# Set Mask

## ISMIP-HOM Test A

Test Description

Mesh

**Mask**

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

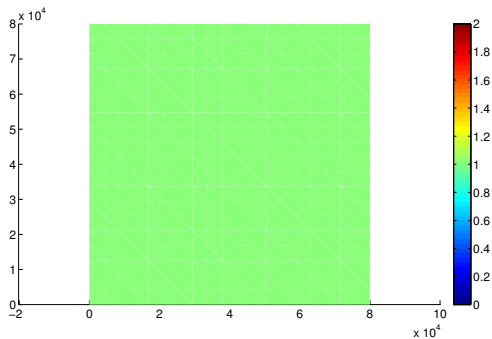
## ISMIP-HOM Test F

Test Description

Transient model

Results

```
1 >> plotmodel(md, 'data', md.mask.elementongroundedice)
```



ISMIP-HOM

Larour et al.

## Parameterization

### ISMIP-HOM Test A

Test Description

Mesh

Mask

**Parameterization**

Extrusion

Flow equation

Boundary conditions

Results

### ISMIP-HOM Test F

Test Description

Transient model

Results

Use ISMIPA.par file to parameterize the model:

```
1  >> help parameterize
2  PARAMETERIZE - parameterize a model
3
4  from a parameter matlab file, start filling in all the @model fields that were not
5  filled in by the mesh.m and mask.m @model methods.
6  Warning: the paramter file must be able to be run in Matlab
7
8  Usage:
9      md=parameterize(md,parametername)
10
11  Example:
12      md=parameterize(md,'Square.par');
```



ISMIP-HOM

Larour et al.

## Parameterization

ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

ISMIP-HOM Test F

Test Description

Transient model

Results

Use ISMIPA.par file to parameterize the model:

```
1  >> help parameterize
2  PARAMETERIZE - parameterize a model
3
4  from a parameter matlab file, start filling in all the @model fields that were not
5  filled in by the mesh.m and mask.m @model methods.
6  Warning: the paramter file must be able to be run in Matlab
7
8  Usage:
9      md=parameterize(md,parametername)
10
11  Example:
12      md=parameterize(md,'Square.par');
```

```
1  >> md=parameterize(md,'./ISMIPA.par');
```

ISMIP-HOM

Larour et al.

## Parameterization

ISMIP-HOM Test A

Test Description

Mesh

Mask

**Parameterization**

Extrusion

Flow equation

Boundary conditions

Results

ISMIP-HOM Test F

Test Description

Transient model

Results

Use `ISMIPA.par` file to parameterize the model:

First geometry:

- surface:  $s(x, y) = -x \tan(\alpha)$ ,  $\alpha = 0.5^\circ$
- bed:  $b(x, y) = s(x, y) - 1000 + 500 \sin(\omega x) \sin(\omega y)$ ,  $\omega = \frac{2\pi}{L}$
- thickness:  $h(x, y) = s(x, y) - b(x, y)$

ISMIP-HOM

Larour et al.

## Parameterization

Use `ISMIPA.par` file to parameterize the model:

First geometry:

- surface:

```
md.geometry.surface=-md.mesh.x*tan(0.5*pi/180);
```

- bed:

```
md.geometry.bed=md.geometry.surface-1000 ...  
... +500*sin(md.mesh.x*2*pi/L).*sin(md.mesh.y*2*pi/L);
```

- thickness:

```
md.geometry.thickness=md.geometry.surface-md.geometry.bed;
```

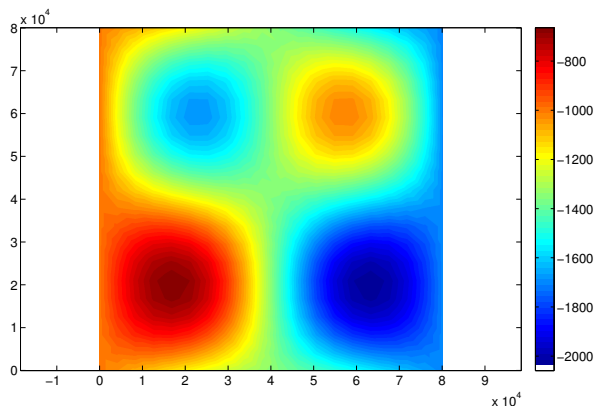
ISMIP-HOM

Larour et al.

## Parameterization

Use ISMIPA.par file to parameterize the model:

```
1 >> plotmodel(md, 'data', md.geometry.bed)
```



ISMIP-HOM

Larour et al.

ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

ISMIP-HOM Test F

Test Description

Transient model

Results

## Parameterization

Use `ISMIPA.par` file to parameterize the model:

Friction: frozen bed

→ Does not matter (use default value in `ISMIPA.par` file)

Rheology:

- Ice-flow parameter:  $A = 10^{16} \text{ Pa}^n \text{ a}^{-1}$
- Exponent in Glen's flow law:  $n = 3$

ISMIP-HOM

Larour et al.

ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

ISMIP-HOM Test F

Test Description

Transient model

Results

## Parameterization

Use ISMIPA.par file to parameterize the model:

Friction: frozen bed

→ Does not matter (use default value in ISMIPA.par file)

Rheology:

- Ice-flow parameter:  $B = \frac{1}{A^{1/n}}$  in Pa s<sup>1/n</sup>

```
1 md.materials.rheology_B= ...
2 ... 6.8067*10^7*ones(md.mesh.numberofvertices,1);
```

- Exponent in Glen's flow law:

```
1 md.materials.rheology_n= ...
2 ... 3*ones(md.mesh.numberofelements,1);
```

ISMIP-HOM

Larour et al.

ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

ISMIP-HOM Test F

Test Description

Transient model

Results

## Parameterization

Use ISMIPA.par file to parameterize the model:

Friction: frozen bed

→ Does not matter (use default value in ISMIPA.par file)

Rheology:

- Ice-flow parameter:  $B = \frac{1}{A^{1/n}}$  in Pa s<sup>1/n</sup>

```
1 md.materials.rheology_B= ...
2 ... 6.8067*10^7*ones(md.mesh.numberofvertices,1);
```

- Exponent in Glen's flow law:

```
1 md.materials.rheology_n= ...
2 ... 3*ones(md.mesh.numberofelements,1);
```

Boundary condition:

- Ice Sheet default boundary conditions

ISMIP-HOM

Larour et al.

ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

ISMIP-HOM Test F

Test Description

Transient model

Results

## Parameterization

Use ISMIPA.par file to parameterize the model:

Friction: frozen bed

→ Does not matter (use default value in ISMIPA.par file)

Rheology:

- Ice-flow parameter:  $B = \frac{1}{A^{1/n}}$  in Pa s<sup>1/n</sup>

```
1 md.materials.rheology_B= ...
2 ... 6.8067*10^7*ones(md.mesh.numberofvertices,1);
```

- Exponent in Glen's flow law:

```
1 md.materials.rheology_n= ...
2 ... 3*ones(md.mesh.numberofelements,1);
```

Boundary condition:

- Ice Sheet default boundary conditions

```
1 md=SetIceSheetBC(md);
```



ISMIP-HOM

Larour et al.

## Extrusion

Use `extrude.m` to extrude the model:

```
1  >> help extrude
2  EXTRUDE - vertically extrude a 2d mesh
3
4  vertically extrude a 2d mesh and create corresponding 3d mesh.
5  The vertical distribution can:
6  - follow a polynomial law
7  - follow two polynomial laws, one for the lower part and one for the upper part
8  of the mesh
9  - be described by a list of coefficients (between 0 and 1)
10
11  Usage:
12      md=extrude(md,numlayers,extrusionexponent);
13      md=extrude(md,numlayers,lowerexponent,upperexponent);
14      md=extrude(md,listofcoefficients);
15
16  Example:
17      md=extrude(md,8,3);
18      md=extrude(md,8,3,2);
19      md=extrude(md,[0 0.2 0.5 0.7 0.9 0.95 1]);
20
21  See also:
22  MODELEXTRACT, COLLAPSE
```

ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

ISMIP-HOM Test F

Test Description

Transient model

Results

ISMIP-HOM

Larour et al.

## Extrusion

Use `extrude.m` to extrude the model:

```
1  >> help extrude
2  EXTRUDE - vertically extrude a 2d mesh
3
4  vertically extrude a 2d mesh and create corresponding 3d mesh.
5  The vertical distribution can:
6  - follow a polynomial law
7  - follow two polynomial laws, one for the lower part and one for the upper part
8  of the mesh
9  - be described by a list of coefficients (between 0 and 1)
10
11  Usage:
12      md=extrude(md,numlayers,extrusionexponent);
13      md=extrude(md,numlayers,lowerexponent,upperexponent);
14      md=extrude(md,listofcoefficients);
15
16  Example:
17      md=extrude(md,8,3);
18      md=extrude(md,8,3,2);
19      md=extrude(md,[0 0.2 0.5 0.7 0.9 0.95 1]);
20
21  See also:
22  MODELEXTRACT, COLLAPSE
```

```
1  md=extrude(md,5,1);
```

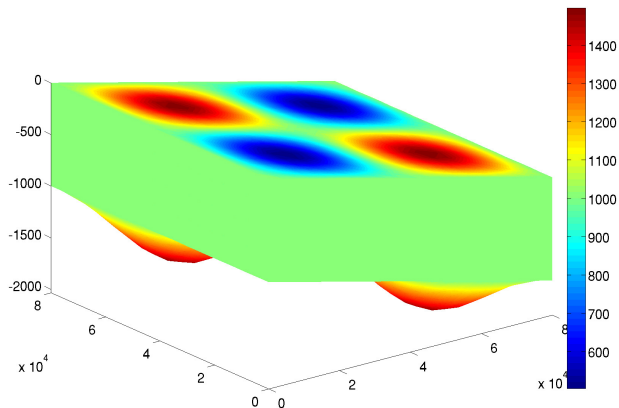
ISMIP-HOM

Larour et al.

## Extrusion

Use `extrude.m` to extrude the model:

```
1 >> plotmodel(md, 'data', md.geometry.thickness)
```



ISMIP-HOM

Larour et al.

## Flow equation

Use `setflowequation.m` to specify the approximation :

```

1  >> help setflowequation
2  SETELEMENTSTYPE - associate a solution type to each element
3
4  This routine works like plotmodel: it works with an even number of inputs
5  'hutter','macayeal','pattyn','stokes' and 'fill' are the possible options
6  that must be followed by the corresponding exp file or flags list
7  It can either be a domain file (argus type, .exp extension), or an array of element flags.
8  If user wants every element outside the domain to be
9  setflowequationd, add '-' to the name of the domain file (ex: '-Pattyn.exp');
10 an empty string '' will be considered as an empty domain
11 a string 'all' will be considered as the entire domain
12 You can specify the type of coupling, 'penalties' or 'tiling', to use with the input ...
    'coupling'
13
14 Usage:
15     md=setflowequation(md,varargin)
16
17 Example:
18     md=setflowequation(md,'pattyn','Pattyn.exp','macayeal',md.mask.elementonfloatingice,'fill',
19     md=setflowequation(md,'pattyn','Pattyn.exp',fill,'hutter','coupling','tiling');
```

ISMIP-HOM

Larour et al.

## Flow equation

Use `setflowequation.m` to specify the approximation :

```

1  >> help setflowequation
2  SETELEMENTSTYPE - associate a solution type to each element
3
4  This routine works like plotmodel: it works with an even number of inputs
5  'hutter','macayeal','pattyn','stokes' and 'fill' are the possible options
6  that must be followed by the corresponding exp file or flags list
7  It can either be a domain file (argus type, .exp extension), or an array of element flags.
8  If user wants every element outside the domain to be
9  setflowequationd, add '-' to the name of the domain file (ex: '-@pattyn.exp');
10 an empty string '' will be considered as an empty domain
11 a string 'all' will be considered as the entire domain
12 You can specify the type of coupling, 'penalties' or 'tiling', to use with the input ...
    'coupling'
13
14 Usage:
15     md=setflowequation(md,varargin)
16
17 Example:
18     md=setflowequation(md,'pattyn','Pattyn.exp','macayeal',md.mask.elementonfloatingice,'fill')
19     md=setflowequation(md,'pattyn','Pattyn.exp',fill,'hutter','coupling','tiling');
```

```
1  md=setflowequation(md,'pattyn','all');
```

ISMIP-HOM

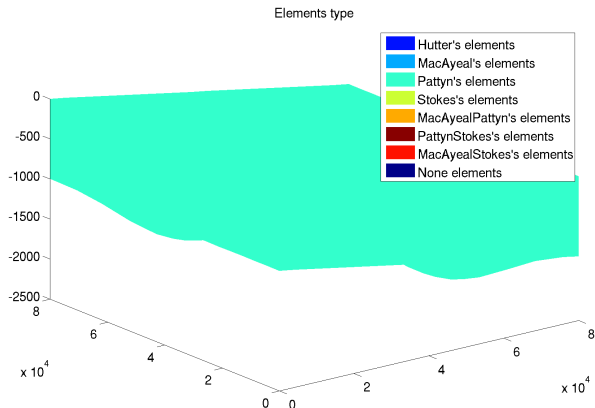
Larour et al.

## Flow equation

Use `setflowequation.m` to specify the approximation :

```
1 md=setflowequation(md, 'pattyn', 'all');
```

```
1 >> plotmodel(md, 'data', 'elements_type')
```



ISMIP-HOM

Larour et al.

## Boundary conditions

### ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

**Boundary conditions**

Results

### ISMIP-HOM Test F

Test Description

Transient model

Results

### Ice frozen on the bedrock: zero velocity

```
1  %Create dirichlet on the bed only
2  md.diagnostic.spcvx=NaN*ones(md.mesh.numberofvertices,1);
3  md.diagnostic.spcvy=NaN*ones(md.mesh.numberofvertices,1);
4  md.diagnostic.spcvz=NaN*ones(md.mesh.numberofvertices,1);
5  pos=find(md.mesh.vertexonbed);
6  md.diagnostic.spcvx(pos)=0;
7  md.diagnostic.spcvy(pos)=0;
8  md.diagnostic.spcvz(pos)=0;
```

ISMIP-HOM

Larour et al.

## Boundary conditions

ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

**Boundary conditions**

Results

ISMIP-HOM Test F

Test Description

Transient model

Results

### Periodic boundary conditions: md.diagnostic.vertex\_pairing

```
1  %Create MPCs to have periodic boundary conditions
2  posx=find(md.mesh.x==0);
3  posx2=find(md.mesh.x==max(md.mesh.x));
4  posy=find(md.mesh.y==0 & md.mesh.x≠0 & ...
            md.mesh.x≠max(md.mesh.x)); %Don't take the same grids twice
5  posy2=find(md.mesh.y==max(md.mesh.y) & md.mesh.x≠0 & ...
            md.mesh.x≠max(md.mesh.x));
6  md.diagnostic.vertex_pairing=[posx,posx2;posy,posy2];
```



ISMIP-HOM

Larour et al.

## Solve model

### ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

**Results**

### ISMIP-HOM Test F

Test Description

Transient model

Results

Set cluster and print convergence while running:

```
1 %Set cluster and print messages
2 md.cluster=generic('name',oshostname(),'np',2);
3 md.verbose=verbose('convergence',true,'solution',true);
```

ISMIP-HOM

Larour et al.

## Solve model

### ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

**Results**

### ISMIP-HOM Test F

Test Description

Transient model

Results

Set cluster and print convergence while running:

```
1 %Set cluster and print messages
2 md.cluster=generic('name',oshostname(),'np',2);
3 md.verbose=verbose('convergence',true,'solution',true);
```

```
1 md=solve(md,DagnosticSolutionEnum);
```

ISMIP-HOM

Larour et al.

## Results

### ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

**Results**

### ISMIP-HOM Test F

Test Description

Transient model

Results

```
1 plotmodel(md, 'data', md.results.DiagnosticSolution.Vx)
```

ISMIP-HOM

Larour et al.

## Results

### ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

**Results**

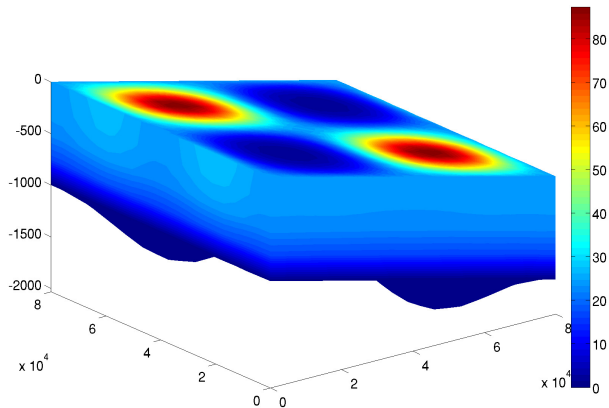
### ISMIP-HOM Test F

Test Description

Transient model

Results

```
1 plotmodel(md, 'data', md.results.DiagnosticSolution.Vx)
```



ISMIP-HOM

Larour et al.

## Results

### ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

**Results**

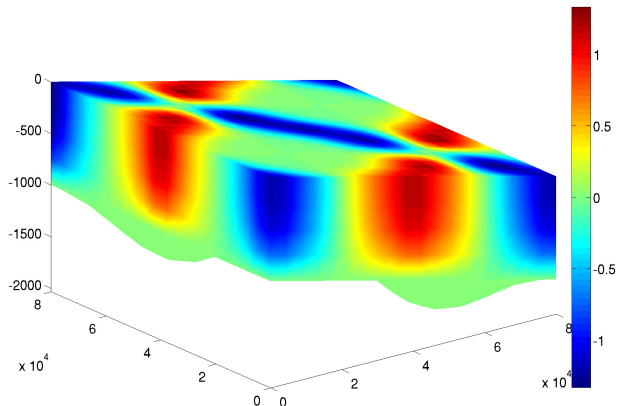
### ISMIP-HOM Test F

Test Description

Transient model

Results

```
1 plotmodel(md, 'data', md.results.DiagnosticSolution.Vx)
```



ISMIP-HOM

Larour et al.

## Results

## ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

**Results**

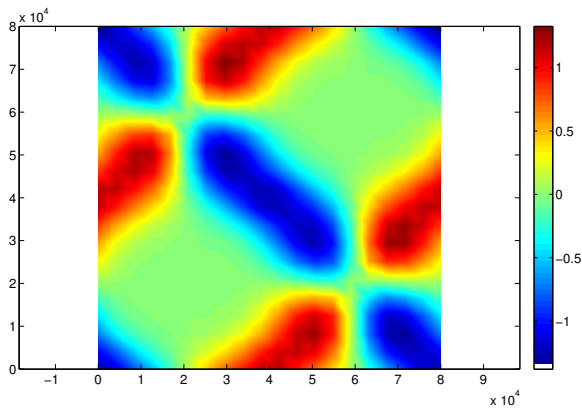
## ISMIP-HOM Test F

Test Description

Transient model

Results

```
1 plotmodel(md, 'data', md.results.DiagnosticSolution.Vy, 'layer', md.mesh,
```



ISMIP-HOM

Larour et al.

## Test F

### ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

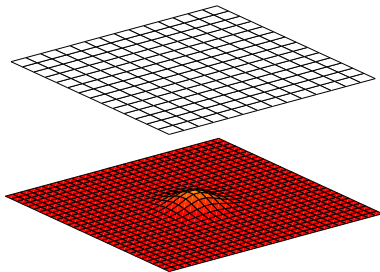
### ISMIP-HOM Test F

Test Description

Transient model

Results

## Square ice sheet flowing over a sinusoidal bed



- Sinusoidal bedrock but flat surface
- Ice frozen or sliding on the bed
- Periodic boundary conditions
- Transient model until steady-state

ISMIP-HOM

Larour et al.

# Transient

## ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

```

1  >> md.transient
2
3  ans -
4
5  transient solution parameters:
6      isprognostic      : 1      -- indicates if a prognostic solution is used in the transient
7      isthermal         : 0      -- indicates if a thermal solution is used in the transient
8      isdiagnostic      : 1      -- indicates if a diagnostic solution is used in the transient
9      isgroundingline   : 0      -- indicates if a groundingline migration is used in the transient
10     requested_outputs : N/A    -- list of additional outputs requested

```

## ISMIP-HOM Test F

Test Description

Transient model

Results

```

1  >> md.prognostic
2
3  ans -
4
5  Prognostic solution parameters:
6      spcthickness      : (2000x1) -- thickness constraints (NaN means no constraint)
7      hydrostatic_adjustment : 'Absolute' -- adjustment of ice shelves surface and bed elevations: . . .
8      stabilization      : 1      -- 0->no, 1->artificial_diffusivity, 3->discontinuous Galerkin
9
10     Penalty options:
11         penalty_factor    : 3      -- offset used by penalties: penalty = Kmax*10^offset
12         vertex_pairing    : (200x2) -- pairs of vertices that are penalized

```

```

1  >> md.timestepping
2
3  ans -
4
5  timestepping parameters:
6      time_step          : 4      -- length of time steps [yrs]
7      final_time         : 80     -- final time to stop the simulation [yrs]
8      time_adapt         : 0      -- use cfl condition to define time step ? (0 or 1)
9      cfl_coefficient    : 0.5    -- coefficient applied to cfl condition

```



ISMIP-HOM

Larour et al.

## Parameterization file

### ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

### ISMIP-HOM Test F

Test Description

Transient model

Results

```

1  disp('      creating geometry');
2  L=100000;
3  alpha=3*pi/180;
4  md.geometry.surface=-md.mesh.x*tan(alpha);
5  md.geometry.bed=md.geometry.surface-1000+100*exp(-(md.mesh.x-L/2).^2+(md.mesh.y-L/2).^2)/(10000^2));
6  md.geometry.thickness=md.geometry.surface-md.geometry.bed;
7
8  disp('      creating drag');
9  md.friction.coefficient=sqrt(365.25*24*3600/(1000*2.140373*10^-7))*ones(md.mesh.numberofvertices,1);
10 md.friction.p=ones(md.mesh.numberofelements,1);
11 md.friction.q=zeros(md.mesh.numberofelements,1);
12
13 disp('      creating flow law parameter');
14 md.materials.rheology_B=(1/(2.140373*10^-7/(365.25*24*3600)))*ones(md.mesh.numberofvertices,1);
15 md.materials.rheology_n=1*ones(md.mesh.numberofelements,1);
16
17 disp('      boundary conditions for diagnostic model');
18 md=SetIceSheetBC(md);
19
20 %Field initialization
21 md.initialization.vx=zeros(md.mesh.numberofvertices,1);
22 md.initialization.vy=zeros(md.mesh.numberofvertices,1);
23 md.initialization.vz=zeros(md.mesh.numberofvertices,1);
24 md.initialization.pressure=zeros(md.mesh.numberofvertices,1);

```

ISMIP-HOM

Larour et al.

## Runme file

## ISMIP-HOM Test A

Test Description

Mesh

Mask

Parameterization

Extrusion

Flow equation

Boundary conditions

Results

## ISMIP-HOM Test F

Test Description

Transient model

Results

```

1  %Create model
2  md=model;
3
4  %Create mesh
5  L=100000; %in m
6  nx=20; %numberof nodes in x direction
7  ny=20;
8  md=squaremesh(md,L,L,nx,ny);
9
10 md=setmask(md,''); %ice sheet test
11 md=parameterize(md,'./ISMIPF.par');
12 md=extrude(md,5,1);
13 md=setflowequation(md,'pattyn','all');
14
15 %Boundary conditions
16 md.diagnostic.spcvx(:)=NaN;
17 md.diagnostic.spcvy(:)=NaN;
18 md.diagnostic.spcvz(:)=NaN;
19 %Create dirichlet on the bed if no slip
20 pos=find(md.mesh.vertexonbed);
21 md.diagnostic.spcvx(pos)=0;
22 md.diagnostic.spcvy(pos)=0;
23 md.diagnostic.spcvz(pos)=0;
24
25 %Create MPCs to have periodic boundary conditions
26 posx=find(md.mesh.x==0);
27 posx2=find(md.mesh.x==max(md.mesh.x));
28 posy=find(md.mesh.y==0); %Don't take the same grids two times
29 posy2=find(md.mesh.y==max(md.mesh.y));
30 md.diagnostic.vertex_pairing=[posx,posx2;posy,posy2];
31 md.prognostic.vertex_pairing=[posx,posx2;posy,posy2];
32
33 %Transient parameters
34 md.timestepping.time_step=4;
35 md.timestepping.final_time=4*20;
36 md.transient.isthermal=0;
37
38 %Compute the diagnostic
39 md.cluster-generic('name',oshostname(),'np',2);
40 md.verbose-verbose('convergence',true,'solution',true);
41 md=solve(md,TransientSolutionEnum);

```

ISMIP-HOM

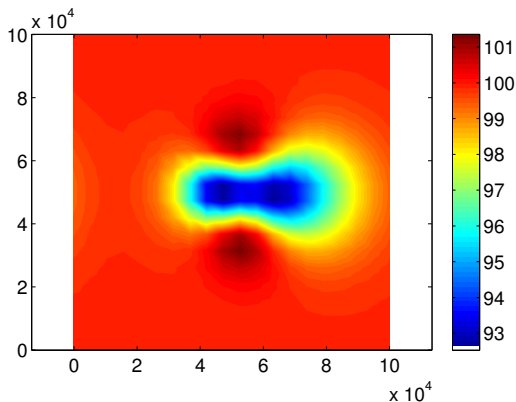
Larour et al.

## Results

Results after 20 iterations of 4 years:

```
1  >> plotmodel(md, 'data', md.results.Transient(20).Vel, 'layer', md.mesh.numberoflayers)
```

Upper surface velocity



ISMIP-HOM

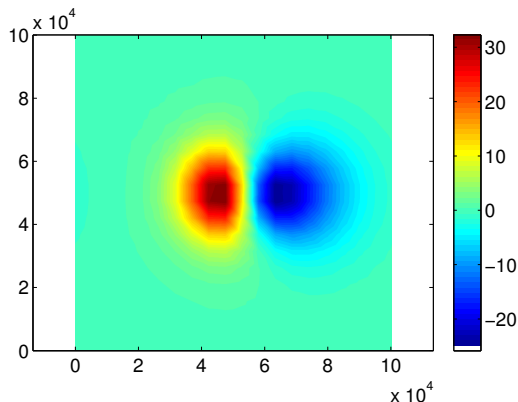
Larour et al.

## Results

Results after 20 iterations of 4 years:

```
1  >> plotmodel(md,'data',md.results.Transient(20).Surface-md.geometry.surface,'layer',md.mesh.numberoflayers)
```

Difference between final and initial upper surface



ISMIP-HOM

Larour et al.

## Results

### Results after 20 iterations of 4 years:

```

1  >> plotmodel(md,'data',md.results.Transient(20).Surface-md.geometry.surface,'layer',md.mesh.numberoflayers,...
2  ...'sectionvalue','Profile.exp')

```

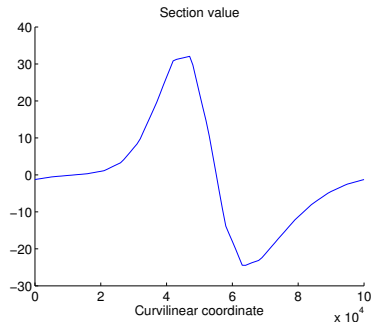
### Define profile in Profile.exp

```

1
2  ## Name:icefront
3  ## Icon:0
4  # Points Count  Value
5  2 1.
6  # X pos Y pos
7  0 50000

```

### Difference between final and initial surface on a section



Thanks!

