Ice Sheet System Model

Ice flow models

Eric Larour

1 Jet Propulsion Laboratory, Pasadena, CA, USA
Outline

1. Ice flow equations
   - Approximations implemented
   - Ice flow equation
   - Diagnostic parameters
   - Boundary conditions

2. Combining models
   - Methods implemented in ISSM
   - Penalties
   - Tiling method
   - Utilization
Ice Sheet flow equations

**Incompressibility**

\[ \forall x \in \Omega \quad \nabla \cdot \mathbf{v} = \text{Tr}(\dot{\varepsilon}) = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \]  

\( \mathbf{v} = (u, v, w) \) ice velocity (m/yr)

\( \dot{\varepsilon} \) strain rate tensor (yr\(^{-1}\))

**Incompressible viscous fluid**

\[ \sigma' = 2\mu \dot{\varepsilon} \]  

\( \sigma' \) deviatoric stress

\( \mu \) ice viscosity

\( \dot{\varepsilon} \) strain rate tensor

**Glen’s flow law**

\[ \mu = \frac{B}{2\dot{\varepsilon}_e^{n-1}} \]  

\( B \) ice hardness

\( n \) Glen’s law coefficient \((n = 3)\)

\( \dot{\varepsilon}_e \) effective strain rate (related to the second invariant)
# Ice Sheet flow equations

## Conservation of momentum

\[ \forall x \in \Omega \quad \nabla \cdot \sigma' - \nabla P + \rho g = 0 \quad (4) \]

**Assumptions:**

1. Stokes flow (quasi-static assumption)
2. Coriolis effect negligible

## Boundary conditions

<table>
<thead>
<tr>
<th>Interface</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ice/Air interface: Free surface</td>
<td>[ \Gamma_s \quad \sigma \cdot n = P_{atm} \quad n \simeq 0 ]</td>
</tr>
<tr>
<td>Ice/Ocean interface: water pressure</td>
<td>[ \Gamma_w \quad \sigma \cdot n = P_w \quad n ]</td>
</tr>
<tr>
<td>Ice/Bedrock interface (1): lateral friction</td>
<td>[ \Gamma_b \quad (\sigma \cdot n + \beta v)_{\parallel} = 0 ]</td>
</tr>
<tr>
<td>Ice/Bedrock interface (2): impenetrability</td>
<td>[ \Gamma_b \quad v \cdot n = 0 ]</td>
</tr>
<tr>
<td>Side boundaries: Dirichlet</td>
<td>[ \Gamma_u \quad v = v_{obs} ]</td>
</tr>
</tbody>
</table>
**Models description**

Full-Stokes model (FS):
- Momentum balance + incompressibility
- 3D model
- Four unknowns \((v_x, v_y, v_z, p)\)

**Model equations**

\[
\begin{align*}
\frac{\partial}{\partial x} \left( 2\mu \frac{\partial v_x}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial v_x}{\partial y} + \mu \frac{\partial v_y}{\partial x} \right) + \frac{\partial}{\partial z} \left( \mu \frac{\partial v_x}{\partial z} + \mu \frac{\partial v_z}{\partial x} \right) - \frac{\partial p}{\partial x} &= 0 \\
\frac{\partial}{\partial x} \left( \frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) + \frac{\partial}{\partial y} \left( 2\mu \frac{\partial v_y}{\partial y} \right) + \frac{\partial}{\partial z} \left( \mu \frac{\partial v_y}{\partial z} + \mu \frac{\partial v_z}{\partial y} \right) - \frac{\partial p}{\partial y} &= 0 \\
\frac{\partial}{\partial x} \left( \frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial v_z}{\partial z} + \mu \frac{\partial v_y}{\partial y} \right) + \frac{\partial}{\partial z} \left( 2\mu \frac{\partial v_z}{\partial z} \right) - \frac{\partial p}{\partial z} - \rho g &= 0
\end{align*}
\]

\[\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} = 0\]
Models description

Higher-order model (HO):

- [Blatter, 1995, Pattyn, 2003]
- 3D model
- Horizontal and vertical velocity decoupled
- $2 (v_x, v_y) + 1 (v_z)$ unknowns

Blatter-Pattyn (BP)

Blatter-Pattyn (BP) $(v_x, v_y)$

Model equations

\[
\begin{align*}
\frac{\partial}{\partial x} \left( 2\mu \frac{\partial v_x}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial v_x}{\partial y} + \mu \frac{\partial v_y}{\partial x} \right) + \frac{\partial}{\partial z} \left( \mu \frac{\partial v_x}{\partial z} + \mu \frac{\partial v_z}{\partial x} \right) - \frac{\partial p}{\partial x} &= 0 \\
\frac{\partial}{\partial x} \left( \mu \frac{\partial v_y}{\partial y} + \mu \frac{\partial v_z}{\partial x} \right) + \frac{\partial}{\partial y} \left( 2\mu \frac{\partial v_y}{\partial y} \right) + \frac{\partial}{\partial z} \left( \mu \frac{\partial v_y}{\partial z} + \mu \frac{\partial v_z}{\partial y} \right) - \frac{\partial p}{\partial y} &= 0 \\
\frac{\partial}{\partial x} \left( \mu \frac{\partial v_z}{\partial z} + \mu \frac{\partial v_x}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial v_z}{\partial z} + \mu \frac{\partial v_y}{\partial y} \right) + \frac{\partial}{\partial z} \left( 2\mu \frac{\partial v_z}{\partial z} \right) - \frac{\partial p}{\partial z} - \rho g &= 0 \\
\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} &= 0
\end{align*}
\]
Models description

Shelfy-stream approximation (SSA):
- [MacAyeal, 1989]
- 2D model
- Horizontal and vertical velocity decoupled
- $2 (v_x, v_y) + 1 (v_z)$ unknowns

Model equations

$$
\begin{align*}
\frac{\partial}{\partial x} \left( 2\mu \frac{\partial v_x}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial v_x}{\partial y} + \mu \frac{\partial v_y}{\partial x} \right) + \frac{\partial}{\partial z} \left( \mu \frac{\partial v_x}{\partial z} + \mu \frac{\partial v_z}{\partial x} \right) - \frac{\partial p}{\partial x} &= 0 \\
\frac{\partial}{\partial x} \left( \mu \frac{\partial v_y}{\partial y} + \mu \frac{\partial v_y}{\partial x} \right) + \frac{\partial}{\partial y} \left( 2\mu \frac{\partial v_y}{\partial y} \right) + \frac{\partial}{\partial z} \left( \mu \frac{\partial v_y}{\partial z} + \mu \frac{\partial v_z}{\partial y} \right) - \frac{\partial p}{\partial y} &= 0 \\
\frac{\partial}{\partial x} \left( \mu \frac{\partial v_z}{\partial z} + \mu \frac{\partial v_z}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial v_z}{\partial z} + \mu \frac{\partial v_z}{\partial y} \right) + \frac{\partial}{\partial z} \left( 2\mu \frac{\partial v_z}{\partial z} \right) - \frac{\partial p}{\partial z} - \rho g &= 0 \\
\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} &= 0
\end{align*}
$$
Models description

Shallow ice approximation (SIA):
- [Hutter, 1983]
- 3D analytical model
- 2 unknowns \((v_x, v_y)\) computed separately

Model equations

\[
\begin{align*}
\frac{\partial}{\partial x} \left( 2\mu \frac{\partial v_x}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial v_x}{\partial y} + \mu \frac{\partial v_y}{\partial x} \right) + \frac{\partial}{\partial z} \left( \mu \frac{\partial v_x}{\partial z} + \mu \frac{\partial v_z}{\partial x} \right) - \frac{\partial p}{\partial x} &= 0 \\
\frac{\partial}{\partial x} \left( \mu \frac{\partial v_x}{\partial y} + \mu \frac{\partial v_y}{\partial x} \right) + \frac{\partial}{\partial y} \left( 2\mu \frac{\partial v_y}{\partial y} \right) + \frac{\partial}{\partial z} \left( \mu \frac{\partial v_y}{\partial z} + \mu \frac{\partial v_z}{\partial y} \right) - \frac{\partial p}{\partial y} &= 0 \\
\frac{\partial}{\partial x} \left( \mu \frac{\partial v_x}{\partial z} + \mu \frac{\partial v_z}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu \frac{\partial v_y}{\partial z} + \mu \frac{\partial v_z}{\partial y} \right) + \frac{\partial}{\partial z} \left( 2\mu \frac{\partial v_z}{\partial z} \right) - \frac{\partial p}{\partial z} - \rho g &= 0 \\
\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} &= 0
\end{align*}
\]
Material non-linearity

Model equations

\[
\begin{align*}
\frac{\partial}{\partial x} \left( 2\mu \frac{\partial v_x}{\partial x} \right) &+ \frac{\partial}{\partial y} \left( \mu \frac{\partial v_x}{\partial y} + \mu \frac{\partial v_y}{\partial x} \right) &+ \frac{\partial}{\partial z} \left( \mu \frac{\partial v_x}{\partial z} + \mu \frac{\partial v_z}{\partial x} \right) - \frac{\partial p}{\partial x} &= 0 \\
\frac{\partial}{\partial x} \left( \mu \frac{\partial v_y}{\partial z} + \mu \frac{\partial v_z}{\partial x} \right) &+ \frac{\partial}{\partial y} \left( 2\mu \frac{\partial v_y}{\partial y} \right) &+ \frac{\partial}{\partial z} \left( \mu \frac{\partial v_y}{\partial z} + \mu \frac{\partial v_z}{\partial y} \right) - \frac{\partial p}{\partial y} &= 0 \\
\frac{\partial}{\partial x} \left( \mu \frac{\partial v_z}{\partial z} \right) &+ \frac{\partial}{\partial y} \left( \mu \frac{\partial v_z}{\partial z} \right) &+ \frac{\partial}{\partial z} \left( 2\mu \frac{\partial v_z}{\partial z} \right) - \frac{\partial p}{\partial z} - \rho g &= 0 \\
\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} &= 0
\end{align*}
\]

Glen’s flow law

\[
\mu = \frac{B}{2 \varepsilon_e^n} \left( \frac{n-1}{n} \right)
\]

- \( B \) ice hardness
- \( n \) Glen's law coefficient (\( n = 3 \))
- \( \varepsilon_e \) effective strain rate (second invariant)

- Treatment of non-linearity with fixed point (Picard) or Newton’s method
Ice flow equations

Material non-linearity

Treatment of non-linearity with fixed point:

Initial velocity $u = u^h + w$

Compute $u^h$

$\alpha (u^h, \Phi) = l (\Phi)$

Viscosity convergence

Compute $w$

$\frac{\partial w}{\partial z} = -\text{div} \left( u^h \right)$

Solution $u = u^h + w$

Horizontal velocity computed with incompressibility for SSA and HO

Initial velocity $u_h$

Compute $u_h$

$\alpha_h (u_h, \Phi) = l_h (\Phi)$

Viscosity convergence

Compute $w$

Solution $u = u^h + w$

Vertical velocity computed with incompressibility for SSA and HO
Flow equation

`setflowequation` is used to generate the approximation used to compute the velocity

- Arguments:
  1. model
  2. approximation names
  3. approximation domains

- Domains can be Argus files or array of element flags

- Approximation available
  - FS (Full-Stokes model)
  - HO (Higher-order model)
  - SSA (Shallow Shelf Approximation)
  - SIA (Shallow Ice Approximation)

- Possibility of coupling models
Flow equation

`setflowequation` is used to generate the approximation used to compute the velocity

- Examples

```matlab
1 md=setflowequation(md,'SIA','all')
2 md=setflowequation(md,'FS','all')
3 md=setflowequation(md,'SSA','all')
4 md=setflowequation(md,'HO','all')
```

- To display the type of approximation:

```matlab
1 >> plotmodel(md,'data','elements_type')
```
Flow equation

- To display the type of approximation:

```matlab
1  >> plotmodel(md,'data','elements_type')
```
## Flow equation class

```matlab
>> md.flowequation
ans =
flow equation parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>isSIA</td>
<td>0</td>
</tr>
<tr>
<td>isSSA</td>
<td>1</td>
</tr>
<tr>
<td>isL1L2</td>
<td>0</td>
</tr>
<tr>
<td>isHO</td>
<td>0</td>
</tr>
<tr>
<td>isFS</td>
<td>0</td>
</tr>
<tr>
<td>fe_SSA</td>
<td>'P1'</td>
</tr>
<tr>
<td>fe_HO</td>
<td>'P1'</td>
</tr>
<tr>
<td>fe_FS</td>
<td>'MINIcondensed'</td>
</tr>
<tr>
<td>vertex_equation</td>
<td>N/A</td>
</tr>
<tr>
<td>element_equation</td>
<td>N/A</td>
</tr>
<tr>
<td>borderSSA</td>
<td>N/A</td>
</tr>
<tr>
<td>borderHO</td>
<td>N/A</td>
</tr>
<tr>
<td>borderFS</td>
<td>N/A</td>
</tr>
</tbody>
</table>
```

- `isSIA`: 0 -- is the Shallow Ice Approximation (SIA)?
- `isSSA`: 1 -- is the Shelfy-Stream Approximation (SSA)?
- `isL1L2`: 0 -- is the L1L2 approximation used?
- `isHO`: 0 -- is the Higher-Order (HO) approximation?
- `isFS`: 0 -- are the Full-FS (FS) equations used?
- `fe_SSA`: 'P1' -- Finite Element for SSA 'P1', 'P1bubble'...
- `fe_HO`: 'P1' -- Finite Element for HO 'P1' 'P1bubble'...
- `fe_FS`: 'MINIcondensed' -- Finite Element for FS 'MINIcondensed'...
- `vertex_equation`: N/A -- flow equation for each vertex
- `element_equation`: N/A -- flow equation for each element
- `borderSSA`: N/A -- vertices on SSA's border (for tiling)
- `borderHO`: N/A -- vertices on HO's border (for tiling)
- `borderFS`: N/A -- vertices on FS' border (for tiling)
StressBalance class

```
>> md.stressbalance
ans =

StressBalance solution parameters:

Convergence criteria:
restol     : 0.0001  -- mechanical equilibrium residual convergence criterion
reltol     : 0.01    -- velocity relative convergence criterion, NaN: not applied
abstol     : 10      -- velocity absolute convergence criterion, NaN: not applied
isnewton   : 0        -- 0: Picard's fixed point, 1: Newton's method, 2: hybrid
maxiter    : 100     -- maximum number of nonlinear iterations
viscosity_overshoot : 0  -- over-shooting constant new=new+C*(new-old)

boundary conditions:
spcvx       : N/A     -- x-axis velocity constraint (NaN means no constraint) [m/yr]
spcvy       : N/A     -- y-axis velocity constraint (NaN means no constraint) [m/yr]
spcvz       : N/A     -- z-axis velocity constraint (NaN means no constraint) [m/yr]

Rift options:
rift_penalty_threshold : 0  -- threshold for instability of mechanical constraints
rift_penalty_lock       : 10 -- number of iterations before rift penalties are locked

Penalty options:
penalty_factor          : 3   -- offset used by penalties: penalty = Kmax*10^offset
vertex_pairing          : N/A  -- pairs of vertices that are penalized

Other:
shelf_dampening         : 0    -- use dampening for floating ice? Only for FS model
FSreconditioning        : 100000000000000 -- multiplier for incompressibility equation. Only for FS model
referential             : N/A   -- local referential
loadingforce            : N/A   -- loading force applied on each point [N/m^3]
requested_outputs       : {'default'} -- additional outputs requested
```
Boundary conditions

Boundary conditions created automatically or manually

- Automatically:

```
1   >> md=SetIceSheetBC(md)
2   >> md=SetIceShelfBC(md,'Front.exp')
3   >> md=SetMarineIceSheefBC(md,'Front.exp')
```

- Manually: fields to change
  - `md.stressbalance.spcvx`
  - `md.stressbalance.spcvy`
  - `md.stressbalance.spcvz`
  - `md.mask.ice_levelset`

- To display the boundary conditions

```
1   >> plotmodel(md,'data','BC')
```
## Models description

"Everything should be made as simple as possible, but no simpler." Albert Einstein

<table>
<thead>
<tr>
<th>Model</th>
<th>Dim.</th>
<th>Unknowns</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS (Full-Stokes)</td>
<td>3d</td>
<td>4</td>
<td>[Stokes, 1845]</td>
</tr>
<tr>
<td>HO (Blatter-Pattyn)</td>
<td>3d</td>
<td>2 + 1</td>
<td>[Blatter, 1995, Pattyn, 2003]</td>
</tr>
<tr>
<td>SSA (Shallow shelf)</td>
<td>2d</td>
<td>2 + 1</td>
<td>[MacAyeal, 1989]</td>
</tr>
<tr>
<td>SIA (Shallow ice)</td>
<td>2d</td>
<td>2 + 1</td>
<td>[Hutter, 1983]</td>
</tr>
</tbody>
</table>
Penalty method

- Only to couple SSA and HO
- Very stiff spring to penalize differences between velocities at the interface

Using penalties to couple models:

```matlab
1 md=setflowequation(md,'SSA','FloatingIce.exp','fill','HO','coupling','penalties')
```
**Domain Decomposition**

- $\Omega = \Omega_1 \cup \Omega_2$
- $\Omega_S = \Omega_1 \cap \Omega_2 \neq \emptyset$
- $\mathbf{u} = u_1|_{\Omega_1} + u_2|_{\Omega_2} \in \tilde{V}(\Omega) = (V_1(\Omega_1) + V_2(\Omega_2))$

Find $\mathbf{u} = u_1|_{\Omega_1} + u_2|_{\Omega_2} \in \tilde{V}$,

$$\forall (\mathbf{v}_1, \mathbf{v}_2) \in \tilde{V} \quad a(u_1 + u_2, v_1 + v_2) = l(v_1 + v_2)$$

→ Infinite number of solutions for the continuous problem
Discretization

We take advantage of the discretization to avoid the redundancy:

→ Create one layer of elements in the superposition zone
Multi-model formulation

Two different models: $a_1$, $a_2$ and $l_1$, $l_2$

Find $u = u_1 |_{\Omega_1} + u_2 |_{\Omega_2} \in (V_1 + V_2)$, such that:

$$\forall v = v_1 |_{\Omega_1} + v_2 |_{\Omega_2} \in (V_1 + V_2)$$

$$a_1 \left( u_1 |_{\Omega_1}, v_1 |_{\Omega_1} \right) + a_2 \left( u_2 |_{\Omega_2}, v_2 |_{\Omega_2} \right) + a_2 \left( u_1 |_{\Omega_1}, v_2 |_{\Omega_2} \right) + a_1 \left( u_2 |_{\Omega_2}, v_1 |_{\Omega_1} \right) = l_1 \left( v_1 |_{\Omega_1} \right) + l_2 \left( v_2 |_{\Omega_2} \right)$$

- Coupling different mechanical models
- Easy to implement (local modification of stiffness matrices)
**Flow equation**

`setflowequation` is used to generate the approximation used to compute the velocity

- **Examples**

```matlab
1   md=setflowequation(md,'HO',md.mask.groundedice_levelset>0,'fill','SSA','coupling','penalties')
2   md=setflowequation(md,'HO',md.mask.groundedice_levelset>0,'fill','SSA','coupling','tiling')
3   md=setflowequation(md,'FS','Contour.exp','fill','HO')
```

- **Use exptool to create EXP contours**

```matlab
1   >> exptool('Contour.exp')
```
Flow equation

- To display the type of approximation:

```matlab
>> plotmodel(md,'data','elements_type','edgecolor','k','expdisp','Contour.exp')
```
Bibliography I

Velocity and stress-fields in grounded glaciers: A simple algorithm for including deviatoric stress gradients.  

*Theoretical glaciology: material science of ice and the mechanics of glaciers and ice sheets.*  

Large-scale ice flow over a viscous basal sediment: Theory and application to Ice Stream B, Antarctica.  

A new three-dimensional higher-order thermomechanical ice sheet model: Basic sensitivity, ice stream development, and ice flow across subglacial lakes.  
Bibliography II

Stokes, G. (1845).
On the theories of internal friction of fluids in motion.
Thanks!