Numerical modelling of ice sheets, streams, and shelves

Ed Bueler

Dept of Mathematics and Statistics
and Geophysical Institute
University of Alaska, Fairbanks

September 2012
Karthaus Summer School

slogans & scope

slogans:

- focus on approximating ice flow
- example numerical codes that actually work

scope:

- continuum models
  - shallow ice approximation (SIA) in 2D
  - shallow shelf approximation (SSA) in 1D
  - mass continuity & surface kinematical equations
- numerical ideas
  - finite difference schemes
  - solving algebraic systems from stress balances
  - verification

notation: see printed notes

outside of scope

not covered here:

- Stokes and "higher order" flow equations
- thermomechanical coupling or polythermal ice
- subglacial hydrology/processes
- mass balance and snow/firn processes
- constitutive relations other than Glen isotropic
- grounding lines, calving fronts, ocean interaction
- paleo-climate and "spin-up"
- earth deformation under ice sheet load
- other numerics: FEM, spectral, multigrid, parallel, ...
- etc.

Matlab/Octave codes

- lectures are structured around 16 Matlab codes
- they also work in Octave
- each is 1/2 to one page
- `.zip` and `.tar.gz` forms available from memory stick
- online:

`http://www.dms.uaf.edu/~bueler/karthaus/`

my goal for the first hour

- my goal is to get to an equation for which I can say:

  numerically solve just this equation, and you've got a usable model for a flowing ice sheet

- a "usable" model is *understood* more than it is *correct*
- to get to my goal I will quickly recall some continuum mechanics

ice in glaciers is a *fluid*

- we describe fluids primarily by a *velocity field* $\mathbf{u}(t, x, y, z)$
- if the ice fluid were
  - faster-moving than it actually is, and
  - linearly-viscous like liquid water

  then ice flow would be a “typical” fluid
- we would use the Navier-Stokes equations as our flow model:

$$\nabla \cdot \mathbf{u} = 0 \qquad \textit{incompressibility}$$

$$\rho\left(\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \nu\nabla^2\mathbf{u} + \rho\mathbf{g} \qquad \textit{force balance}$$

- so, to numerically model our fluid, we grab a textbook on “computational fluid dynamics”
- right?

hmmm . . . does not sound like glaciology to me!

is numerical ice flow modeling a part of computational fluid dynamics?

- yes
- large scale like atmosphere/ocean
- . . . but it is a weird one
- consider what makes atmosphere/ocean flow modeling exciting:
  - turbulence
  - convection
  - coriolis force
  - density variation
  - chemistry (methane, ozone, . . .)
- none of the above is very relevant to ice flow
- so what could be interesting about the flow of slow, cold, stiff, laminar, inert old ice?

  . . . it's *ice dynamics!*

ice is a slow, shear-thinning fluid

- our fluid is
  slow: $\rho\left(\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}\right) \approx 0$
  non-Newtonian: viscosity ν is not constant
- "slow":

$$\rho\left(\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}\right) \approx 0 \qquad \Longleftrightarrow \qquad \begin{pmatrix}\text{forces of inertia} \\ \text{are negligible}\end{pmatrix}$$

- "non-Newtonian" flow: it is "shear-thinning", so larger strain rate means smaller viscosity
- so the standard ice flow model is Glen-law ($n = 3$) Stokes:

$$\begin{aligned} \nabla\cdot\mathbf{u} &= 0 && \textit{incompressibility} \\ 0 &= -\nabla p + \nabla\cdot\tau_{ij} + \rho\mathbf{g} && \textit{force balance} \\ \mathbf{D}u_{ij} &= A\tau^2\tau_{ij} && \textit{flow law} \end{aligned}$$

ice is a slow fluid 2

- equations on previous slide are true at every instant
- *geometry, boundary stress, and ice viscosity determine velocity field instantaneously*
- a time-stepping ice sheet code recomputes the velocity field at every time step, without requiring velocity from the previous step[1]
- thus no memory of previous momentum/velocity, so velocity is a "diagnostic" output of an ice flow model

---

[1] to be a weatherman you've got to know which way the wind blows ... but don't expect that from a glaciologist

plane flow Stokes

- in the x, z plane flow case the Stokes equations say

$$\begin{aligned} u_x + w_z &= 0 && \textit{incompressibility} \\ p_x &= \tau_{11,x} + \tau_{13,z} && \textit{stress balance } (x) \\ p_z &= \tau_{13,x} - \tau_{11,z} - \rho g && \textit{stress balance } (z) \\ u_x &= A\tau^2\tau_{11} && \textit{flow law } \text{(diagonal)} \\ u_z + w_x &= 2A\tau^2\tau_{13} && \textit{flow law } \text{(off-diagonal)} \end{aligned}$$

- x, z subscripts are partial derivatives
- τ_{13} is a “vertical” shear stress
- τ_{11} and $\tau_{33} = -\tau_{11}$ are deviatoric longitudinal stresses
- we have five equations in five unknowns $(u, w, p, \tau_{11}, \tau_{13})$
- complicated enough ...
- what about in a simplified situation?

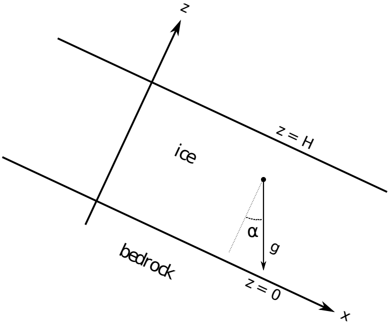
slab-on-a-slope

- rotated coordinates:

$$\mathbf{g} = g \sin\alpha\, \hat{x} - g\cos\alpha\, \hat{z}$$

- so p_x, p_z equations are now:

$$p_x = \tau_{11,x} + \tau_{13,z} + \rho g \sin\alpha$$
$$p_z = \tau_{13,x} - \tau_{11,z} - \rho g \cos\alpha$$



- for a slab-on-a-slope there is *no variation in x*
- so equations simplify:

$$\boxed{\begin{aligned} w_z &= 0 & \qquad 0 &= \tau_{11} \\ \tau_{13,z} &= -\rho g \sin\alpha & \qquad u_z &= 2A\tau^2\tau_{13} \\ p_z &= -\rho g\cos\alpha & & \end{aligned}}$$

slab-on-a-slope 2

- add some boundary conditions:

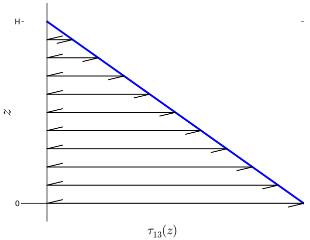
$$w(\text{base}) = 0, \qquad p(\text{surface}) = 0, \qquad u(\text{base}) = u_0$$

- by integrating vertically, get :

$$w = 0, \qquad p = \rho g \cos\alpha (H - z), \qquad \tau_{13} = \rho g \sin\alpha (H - z)$$

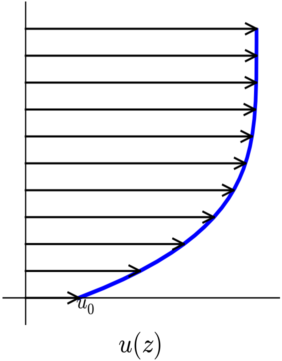
- and from "$u_z = 2A\tau^2\tau_{13}$" get

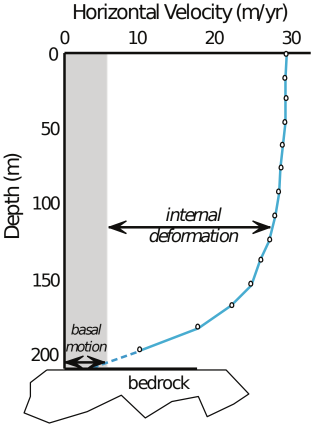
$$u(z) = u_0 + 2A(\rho g \sin\alpha)^3 \int_0^z (H - z')^3\, dz'$$
$$= u_0 + \frac{1}{2} A (\rho g \sin\alpha)^3 \left(H^4 - (H - z)^4\right)$$



slab-on-a-slope 3

- do we believe these equations?
- velocity on last slide (and below) was from a *formula*
- compare to observations at right





velocity profile of the Athabasca Glacier from inclinometry
(Savage and Paterson, 1963)

mass continuity

- now we know the velocity $u = u(t, x, z)$ ... so what?
- suppose our slab has variable thickness $H(t, x)$
- compute the vertical average of velocity:

$$\bar{u}(x, t) = \frac{1}{H} \int_0^H u(t, x, z)\, dz$$

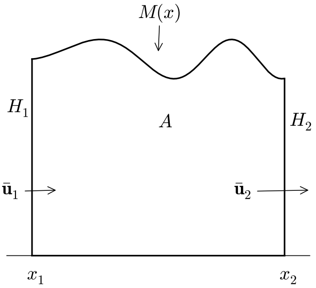
- consider change of area (ice volume in 3D) in the figure:

$$\frac{dA}{dt} = \int_{x_1}^{x_2} M(x)\, dx + \bar{u}_1 H_1 - \bar{u}_2 H_2$$

- but, assuming width $dx = x_2 - x_1$ is small, $A \approx dx\, H$; divide by dx and get

$$H_t = M - (\bar{u} H)_x$$

- this is a *mass continuity equation*



rough explanation of “shallow ice approximation” (SIA)

- consider only $u_0 = 0$ case for now (“non-sliding SIA”)
- from slab-on-slope velocity formula

$$\begin{aligned}\bar u H &= \int_0^H \frac{1}{2} A(\rho g \sin\alpha)^3 \left(H^4 - (H-z)^4\right)\,dz \\ &= \frac{1}{2} A(\rho g \sin\alpha)^3 \left(\frac{4}{5} H^5\right) \\ &= \frac{2}{5} A(\rho g \sin\alpha)^3 H^5\end{aligned}$$

- note $\sin\alpha \approx \tan\alpha = -h_x$
- combine with mass continuity $H_t = M - (\bar u H)_x$:

$$H_t = M + \left(\frac{2}{5}(\rho g)^5 A H^5 |h_x|^2 h_x\right)_x.$$

- this is SIA evolution equation for ice thickness
- I’ll return to this topic ...

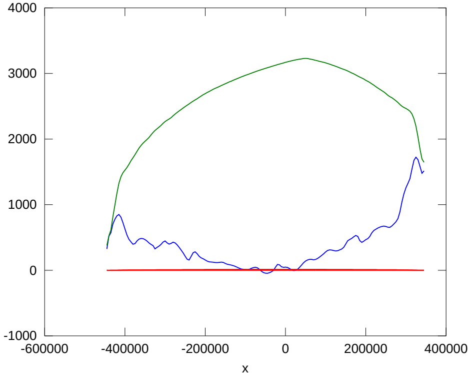
slow, non-Newtonian, some basal slip, and shallow

- ice sheets have four outstanding properties *as fluids*:
  1. slow
  2. non-Newtonian
  3. shallow
  4. contact slip (sometimes)

- the first model will capture the first three properties

regarding "shallow"

- below in red is a no-vertical-exaggeration cross section of Greenland at 71°
  - green and blue: standard vertically-exaggerated cross section



- you can scale Stokes equation using smallness of $\epsilon = [H]/[L]$, where $[H]$ is a typical thickness of an ice sheet and $[L]$ is a typical horizontal dimension, …

non-sliding, isothermal shallow ice approximation = (SIA)

a model which applies to

- shallow grounded ice sheets
- on not-too-rough bed topography,
- whose flow is not dominated by sliding and/or liquid water at the base or margin

“Polaris Glacier,” northwest Greenland, photo 122, Post & LaChapelle (2000)

SIA velocity equation

- here we "derive" the SIA by the simple slogan:

  the SIA uses the formulas from slab-on-a-slope

- shear stress approximation:

$$(\tau_{13}, \tau_{23}) \approx -\rho g(h - z)\nabla h$$

- notation: $\mathbf{U} = (u, v)$ = the horizontal velocity
- we further approximate

$$\begin{aligned}\mathbf{U}_z &\approx 2A|(\tau_{13}, \tau_{23})|^{n-1}(\tau_{13}, \tau_{23}) \\ &= -2A(\rho g)^n(h - z)^n|\nabla h|^{n-1}\nabla h\end{aligned}$$

- by integrating vertically, in the non-sliding case,

$$\mathbf{U} = -\frac{2A(\rho g)^n}{n+1}\left[H^{n+1} - (h - z)^{n+1}\right]|\nabla h|^{n-1}\nabla h$$

- but mass continuity remains, $H_t = M - \left(\overline{\mathbf{U}}H\right)_x$

SIA thickness equation

- combine last two equations on last slide
- get the non-sliding, isothermal shallow ice approximation for thickness changes:

$$\boxed{H_t = M + \nabla \cdot \left(\Gamma H^{n+2} |\nabla h|^{n-1} \nabla h\right)}$$

  - where H is ice thickness, h is ice surface elevation, b is bed elevation ($h = H + b$)
  - M combines surface and basal mass (im)balance: accumulation if $M > 0$, ablation if $M < 0$
  - n is the exponent in the Glen flow law
  - $\Gamma = 2A(\rho g)^n/(n+2)$ is a positive constant

SIA thickness equation 2

- numerically solve this equation

$$\boxed{H_t = M + \nabla \cdot \left(\Gamma H^{n+2} |\nabla h|^{n-1} \nabla h\right)} \qquad (1)$$

and you've got a usable model for ... *the Barnes ice cap* (Mahaffy, 1976)

good questions:

1. how to solve equation (1) numerically?
2. how to *think* about it?

heat equation

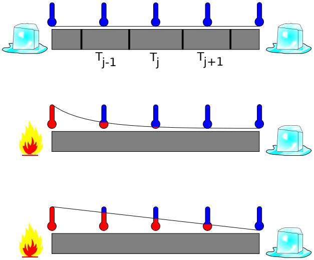
- for understanding the SIA, recall the heat equation describing conduction
- here's one quick way to derive it . . .
- Newton's law of cooling for each segment of the rod:

$$\frac{dT_j}{dt} = -C\left(T_j - \frac{1}{2}(T_{j-1} + T_{j+1})\right) = \frac{C}{2}\left(T_{j-1} - 2T_j + T_{j+1}\right)$$

- the limit as segments shrink $\Delta x \to 0$:

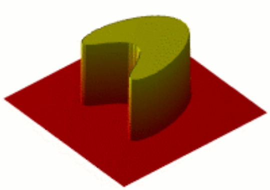
$$T_t = DT_{xx}$$

- $D =$ diffusivity, with units $\mathrm{m}^2\,\mathrm{s}^{-1}$
- $T_{xx} > 0 \implies T(t)$ decreases
- $T_{xx} < 0 \implies T(t)$ increases
- so $T(x)$ becomes smoother with time



more complete heat equation

- $T(t, x, y)$ is temperature in a 2D object at position x, y and time t
- Fourier rewrote Newton's law as a rule for heat flux: $\mathbf{q} = -k\nabla T$
- allow an additional heat source f
- coefficients: ρ is density, c is specific heat, k is conductivity
  - let's assume ρ, c are constant
- by conservation of energy:



$$\rho c T_t = f + \nabla \cdot (k \nabla T)$$

- define the diffusivity $D = k/(\rho c)$ and also $F := f/(\rho c)$
- for 2D object (e.g. a plate), the heat equation:

$$T_t = F + \nabla \cdot (D \nabla T) \qquad (2)$$

- the temperature T solves a *diffusive, time-evolving partial differential equation (PDE)* ... just like thickness H in SIA

analogy: SIA versus heat equation

- side-by-side comparison:

SIA: $H(t,x,y)$ is ice thickness

$$\boxed{H_t = M + \nabla \cdot \left(\Gamma H^{n+2} |\nabla h|^{n-1} \nabla h\right)}$$

heat: $T(t,x,y)$ is temperature

$$\boxed{T_t = F + \nabla \cdot (D \nabla T)}$$

- we identify the diffusivity in the SIA:

$$D = \Gamma H^{n+2} |\nabla h|^{n-1}$$

- *non-sliding shallow ice flow diffuses the ice sheet*
- some issues with this analogy:
  - D depends on solution $H(t,x,y)$
  - $D \to 0$ at margin, where $H \to 0$
  - $D \to 0$ at divides/domes, where $|\nabla h| \to 0$

basic ideas of finite differences

- numerical schemes for heat equation are good start for SIA
- for differentiable $f(x)$ and any Δ, *Taylor's theorem* says

$$f(x+\Delta) = f(x) + f'(x)\Delta + \frac{1}{2}f''(x)\Delta^2 + \frac{1}{3!}f'''(x)\Delta^3 + \dots$$

- you can replace "Δ" by its multiples, e.g.:

$$f(x-\Delta) = f(x) - f'(x)\Delta + \frac{1}{2}f''(x)\Delta^2 - \frac{1}{3!}f'''(x)\Delta^3 + \dots$$
$$f(x+2\Delta) = f(x) + 2f'(x)\Delta + 2f''(x)\Delta^2 + \frac{4}{3}f'''(x)\Delta^3 + \dots$$

- basic finite difference idea:

  combine expressions like these to give approximations of derivatives from values of $f(x)$ on a grid

finite differences for partial derivatives

- we want partial derivative expressions, for example with some function $u = u(t, x)$:

$$u_t(t,x) = \frac{u(t+\Delta t, x) - u(t,x)}{\Delta t} + O(\Delta t),$$

$$u_t(t,x) = \frac{u(t+\Delta t, x) - u(t-\Delta t,x)}{2\Delta t} + O(\Delta t^2),$$

$$u_x(t,x) = \frac{u(t, x+\Delta x) - u(t,x)}{\Delta x} + O(\Delta x),$$

$$u_{xx}(t,x) = \frac{u(t, x+\Delta x) - 2u(t,x) + u(t, x-\Delta x)}{\Delta x^2} + O(\Delta x^2)$$

- sometimes we want a derivative in-between grid points:

$$u_x(t, x+(\Delta x/2)) = \frac{u(t, x+\Delta x) - u(t,x)}{\Delta x} + O(\Delta x^2)$$

- "$+O(\Delta^2)$" is better than "$+O(\Delta)$" if Δ is a small number

explicit scheme for heat equation

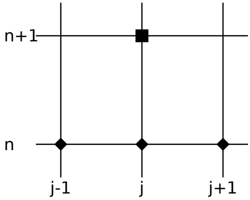
- recall 1D heat equation $T_t = DT_{xx}$
- an *explicit* scheme:

$$\frac{T(t+\Delta t, x) - T(t,x)}{\Delta t} = D\,\frac{T(t, x+\Delta x) - 2T(t,x) + T(t, x-\Delta x)}{\Delta x^2}$$

- the difference between the equation $T_t = DT_{xx}$ and the scheme is $O(\Delta t, \Delta x^2)$
- notation: (t_n, x_j) and $T_j^n \approx T(t_n, x_j)$
- let $\mu = D\Delta t/(\Delta x)^2$, so

$$T_j^{n+1} = \mu T_{j+1}^n + (1-2\mu)T_j^n + \mu T_{j-1}^n$$

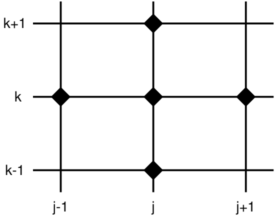
- scheme has stencil at right $\rightarrow$



explicit scheme in two space dimensions

- recall heat equation in 2D: $T_t = D(T_{xx} + T_{yy})$
- in two spatial variables we write $T^n_{jk} \approx T(t_n, x_j, y_k)$
- so the 2D explicit scheme is

$$\frac{T^{n+1}_{jk} - T^n_{jk}}{\Delta t} = D\left(\frac{T^n_{j+1,k} - 2T^n_{jk} + T^n_{j-1,k}}{\Delta x^2} + \frac{T^n_{j,k+1} - 2T^n_{jk} + T^n_{j,k-1}}{\Delta y^2}\right)$$



implementation

```
function T = heat(D,J,K,dt,N)

dx = 2 / J;     dy = 2 / K;
[x,y] = meshgrid(-1:dx:1, -1:dy:1);
T = exp(-30*(x.*x + y.*y));

mu_x = dt * D / (dx*dx);
mu_y = dt * D / (dy*dy);
for n=1:N
   T(2:J,2:K) = T(2:J,2:K) + ...
       mu_x * ( T(3:J+1,2:K) - 2 * T(2:J,2:K) + T(1:J-1,2:K) ) + ...
       mu_y * ( T(2:J,3:K+1) - 2 * T(2:J,2:K) + T(2:J,1:K-1) );
end

surf(x,y,T),  shading('interp'),  xlabel x,  ylabel y
```

heat.m

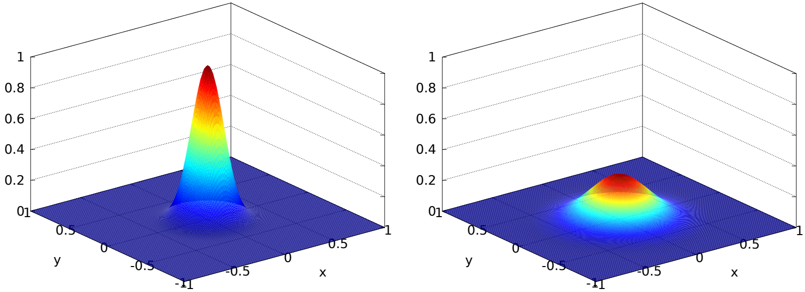
- solves $T_t = D(T_{xx} + T_{yy})$ on square $-1 < x < 1$, $-1 < y < 1$
- uses gaussian initial condition: $T(0, x, y) = e^{-30r^2}$
- uses "colon notation" to remove loops over spatial variables
- » `heat(1.0,30,30,0.001,20)`
  approximates T on 30×30 spatial grid, with $D = 1$ and $N = 20$ steps of $\Delta t = 0.001$

the look of success

- solving $T_t = T_{xx} + T_{yy}$ on 30×30 grid

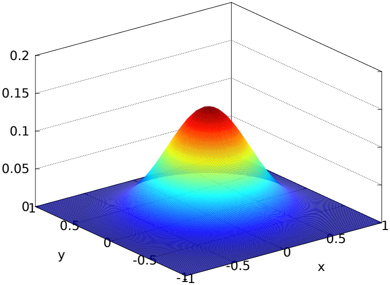
initial condition $T(0,x,y)$

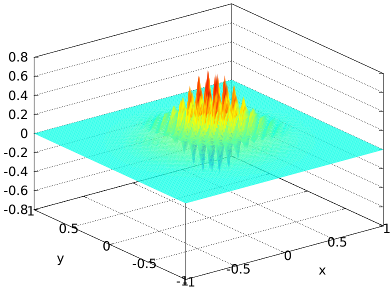
approximate solution $T(t,x,y)$ at $t = 0.02$ with $\Delta t = 0.001$



the look of instability

- figures below are from solving $T_t = T_{xx} + T_{yy}$ on the same space grid, but with slightly different time steps





$$\frac{D\Delta t}{\Delta x^2} = 0.2$$

$$\frac{D\Delta t}{\Delta x^2} = 0.4$$

avoid the instability

- recall 1D explicit scheme has form

$$T_j^{n+1} = \mu T_{j+1}^n + (1 - 2\mu) T_j^n + \mu T_{j-1}^n$$

- thus the new value T_j^{n+1} is an *average* of the old values, *if the middle coefficient is positive*:

$$1 - 2\mu \geq 0 \quad \iff \quad \frac{D\Delta t}{\Delta x^2} \leq \frac{1}{2} \quad \iff \quad \Delta t \leq \frac{\Delta x^2}{2D}$$

- averaging is always stable because averaged wiggles are always smaller than the original wiggles
- this condition is a sufficient *stability criterion*
- *the result was unstable because the time step was too big*
- in 2D case with $\Delta x = \Delta y$ the condition is

$$\frac{D\Delta t}{\Delta x^2} \leq \frac{1}{4}$$

adaptive implementation: guaranteed stability

```
function T = heatadapt(D,J,K,tf)

dx = 2 / J;     dy = 2 / K;
[x,y] = ndgrid(-1:dx:1, -1:dy:1);
T = exp(-30*(x.*x + y.*y));

t = 0.0;     count = 0;
while t < tf
   dt0 = 0.25 * min(dx,dy)^2 / D;
   dt = min(dt0, tf - t);
   mu_x = dt * D / (dx*dx);    mu_y = dt * D / (dy*dy);
   T(2:J,2:K) = T(2:J,2:K) + ...
       mu_x * ( T(3:J+1,2:K) - 2 * T(2:J,2:K) + T(1:J-1,2:K) ) + ...
       mu_y * ( T(2:J,3:K+1) - 2 * T(2:J,2:K) + T(2:J,1:K-1) );
   t = t + dt;
   count = count + 1;
end

surf(x,y,T),  shading('interp'),  xlabel x,  ylabel y
```

`heatadapt.m`

- same as `heat.m` except

choose time step from stability criterion

alternative instability fix: implicitness

- explicit scheme is only "conditionally stable"
    - the adaptive implementation uses the condition
- there are implicit methods which are stable for *any* Δt
- an implicit scheme for the heat equation is *Crank-Nicolson*
    - it has smaller error too: $O(\Delta t^2, \Delta x^2)$
- *but* you have to solve systems of equations at each time step
- in nonlinear case you may end up spending much more programmer time to implement implicit methods
    - this can impose a big opportunity cost

- Donald Knuth has advice for ice sheet modelers:

  > *We should forget about small efficiencies . . . : premature optimization is the root of all evil.*

variable diffusivity

- recall the analogy: (SIA) $\leftrightarrow$ (heat eqn)
- the SIA has a diffusivity $D(x,y)$ which varies in space
- and it has both H and $h = H + b$
- so consider a more general heat equation:

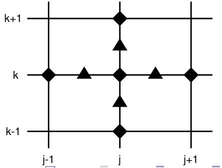
$$T_t = F + \nabla \cdot (D \nabla (T + b)) \qquad (*)$$

- the best explicit method for $(*)$ evaluates diffusivity D at staggered grid points:

$$\nabla \cdot (D \nabla X) \approx \frac{D_{j+1/2,k}(X_{j+1,k} - X_{j,k}) - D_{j-1/2,k}(X_{j,k} - X_{j-1,k})}{\Delta x^2} + \frac{D_{j,k+1/2}(X_{j,k+1} - X_{j,k}) - D_{j,k-1/2}(X_{j,k} - X_{j,k-1})}{\Delta y^2}$$

- best = just as stable as previous
- in stencil at right:

  diamonds: $X = T + b$
  triangles: D



general diffusion equation code

```
function [T,dtav] = diffusion(Lx,Ly,J,K,Dup,Ddown,Dright,Dleft,T0,tf,F,b)

dx = 2 * Lx / J;    dy = 2 * Ly / K;
[x,y] = ndgrid(-Lx:dx:Lx, -Ly:dy:Ly);
T = T0;
if nargin < 11, F = zeros(size(T0)); end
if nargin < 12, b = zeros(size(T0)); end

t = 0.0;    count = 0;
while t < tf
   maxD = [max(max(Dup)) max(max(Ddown)) max(max(Dleft)) max(max(Dright))];
   maxD = max(maxD);
   if maxD <= 0.0
     dt = tf - t;
   else
     dt0 = 0.25 * min(dx,dy)^2 / maxD;
     dt = min(dt0, tf - t);
   end
   mu_x = dt / (dx*dx);    mu_y = dt / (dy*dy);
   Tb = T + b;
   T(2:J,2:K) = T(2:J,2:K) + ...
       mu_y * Dup    .* ( Tb(2:J,3:K+1) - Tb(2:J,2:K)   ) - ...
       mu_y * Ddown  .* ( Tb(2:J,2:K)   - Tb(2:J,1:K-1) ) + ...
       mu_x * Dright .* ( Tb(3:J+1,2:K) - Tb(2:J,2:K)   ) - ...
       mu_x * Dleft  .* ( Tb(2:J,2:K)   - Tb(1:J-1,2:K) );
   T = T + F * dt;
   t = t + dt;    count = count + 1;
end
dtav = tf / count;
```

$\boxed{\texttt{diffusion.m}}$

- solves abstract diffusion equation $T_t = \nabla \cdot (D\nabla(T + b))$
- user supplies diffusivity D on staggered grid

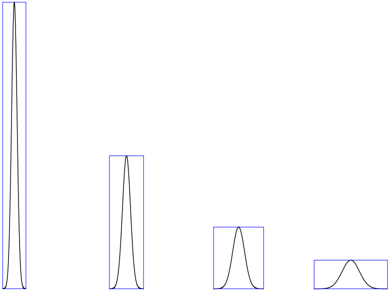
exact solution of heat equation

- before getting to ice flow, one more heat equation topic ...

- many *exact* solutions to the heat equation are known
- I'll show the "Green's function"
- ... also known as "heat kernel"
- it starts at time $t = 0$ with a "delta function" of heat at the origin $x = 0$ and then it spreads out over time
- we find it by a method which generalizes to the SIA

Green's function of heat equation

- the solution is "self-similar" over time
- with time it changes shape by
  - shrinking the output (vertical) axis and
  - lengthening the input (horizontal) axis
- ...but otherwise it is the same shape
- the integral over x is independent of time



increasing time $\longrightarrow$

similarity solutions

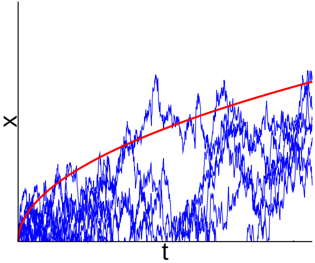
- Green's function of 1D heat equation ($T_t = DT_{xx}$) is

$$T(t,x) = C\,t^{-1/2} e^{-x^2/(4Dt)}$$

- "similarity" variables for 1D heat equation are

$$\underbrace{s = t^{-1/2}x,}_{\textit{input scaling}} \qquad \underbrace{T(t,x) = t^{-1/2}\phi(s)}_{\textit{output scaling}}$$

- *historical note*: in 1905 Einstein saw that the average distance traveled by particles in thermal motion scales like $\sqrt{t}$, so $s = t^{-1/2}x$ is an invariant



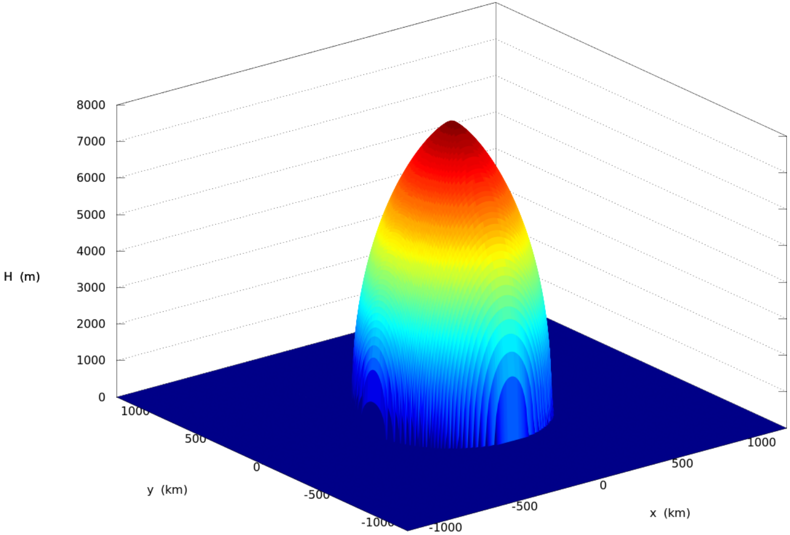
similarity solution to SIA

- jump forward to 1981
- P. Halfar found the similarity solution of the SIA in the case of flat bed and no surface mass balance
- Halfar's 2D solution for Glen flow law with $n = 3$ has scalings

$$\begin{array}{cc} \textit{input scaling} & \textit{output scaling} \\ s = t^{-1/18} r, & H(t,r) = t^{-1/9}\phi(s) \end{array}$$

- so: the nonlinear diffusion of the SIA quickly slows down the rate of change of the profile as the shape flattens out

Halfar solution to the SIA: the movie



frames from $t = 4$ months to $t = 10^6$ years, equal spaced in *exponential* time

Halfar solution to the SIA: the formula

- for $n = 3$ the solution formula is:

$$H(t,r) = H_0 \left(\frac{t_0}{t}\right)^{1/9} \left[1 - \left(\left(\frac{t_0}{t}\right)^{1/18} \frac{r}{R_0}\right)^{4/3}\right]^{3/7}$$

- the "characteristic time" is

$$t_0 = \frac{1}{18\Gamma} \left(\frac{7}{4}\right)^3 \frac{R_0^4}{H_0^7}$$

  if H_0, R_0 are central height and ice cap radius at $t = t_0$
- you choose H_0 and R_0 and then determine t_0
- it is a simple formula to use for verification!

is the Halfar solution *good for any modelling*?

- John Nye and others (2000) compared long-time consequences of different flow laws for the Mars polar caps
- they evaluated CO_2 ice versus H_2O ice parameters
- …by comparing long-time behavior of the Halfar solutions
- conclusions:

> *…none of the three possible [CO_2] flow laws will allow a 3000-m cap, the thickness suggested by stereogrammetry, to survive for 10^7 years, indicating that the south polar ice cap is probably not composed of pure CO_2 ice … the south polar cap probably consists of water ice, with an unknown admixture of dust.*

on “degenerate” diffusivity

- recall that the SIA is

$$H_t = M + \nabla \cdot (D \nabla h) \quad \text{where} \quad D = \Gamma H^{n+2} |\nabla h|^{n-1}$$

- thus the diffusivity “degenerates”, $D \to 0$, when either $H \to 0$ or $\nabla h \to 0$
- summary:

	why $D \to 0$	so what?
domes	$\nabla h \to 0$	H and ∇h are continuous but $\nabla^2 h$ is singular
margins	$H \to 0$	H is continuous but ∇h is singular

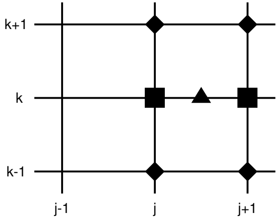
- in terms of numerical error, margin is worse than dome
- degenerate diffusion equations are automatically free boundary problems

computing diffusivity in SIA

- for numerical stability we compute $D = \Gamma H^{n+2} |\nabla h|^{n-1}$ on the staggered grid
- various schemes proposed
- all schemes involve
  - averaging H
  - differencing h
  - in a "balanced" way, for better accuracy,

  to get the diffusivity on staggered grid

- Mahaffy's scheme has stencil $\rightarrow$



SIA implementation: flat bed case

```
function [H,dtlist] = siaflat(Lx,Ly,J,K,H0,deltat,tf)

g = 9.81;     rho = 910.0;     secpera = 31556926;
A = 1.0e-16/secpera;     Gamma  = 2 * A * (rho * g)^3 / 5;
H = H0;

dx = 2 * Lx / J;    dy = 2 * Ly / K;
N = ceil(tf / deltat);     deltat = tf / N;
j  = 2:J;     k = 2:K;
nk = 3:K+1;     sk = 1:K-1;     ej = 3:J+1;     wj = 1:J-1;

t = 0;    dtlist = [];
for n=1:N
  Hup = 0.5 * ( H(j,nk) + H(j,k) );   Hdn = 0.5 * ( H(j,k) + H(j,sk) );
  Hrt = 0.5 * ( H(ej,k) + H(j,k) );   Hlt = 0.5 * ( H(j,k) + H(wj,k) );
  a2up = (H(ej,nk) + H(ej,k) - H(wj,nk) - H(wj,k)).^2 / (4*dx)^2 + ...
         (H(j,nk) - H(j,k)).^2 / dy^2;
  a2dn = (H(ej,k) + H(ej,sk) - H(wj,k) - H(wj,sk)).^2 / (4*dx)^2 + ...
         (H(j,k) - H(j,sk)).^2 / dy^2;
  a2rt = (H(ej,k) - H(j,k)).^2 / dx^2 + ...
         (H(ej,nk) + H(j,nk) - H(ej,sk) - H(j,sk)).^2 / (4*dy)^2;
  a2lt = (H(j,k) - H(wj,k)).^2 / dx^2 + ...
         (H(wj,nk) + H(j,nk) - H(wj,sk) - H(j,sk)).^2 / (4*dy)^2;
  Dup = Gamma * Hup.^5 .* a2up;  Ddn = Gamma * Hdn.^5 .* a2dn;
  Drt = Gamma * Hrt.^5 .* a2rt;  Dlt = Gamma * Hlt.^5 .* a2lt;
  [H,dtadapt] = diffusion(Lx,Ly,J,K,Dup,Ddn,Drt,Dlt,H,deltat);
  t = t + deltat;    dtlist = [dtlist dtadapt];
end
```

siaflat.m

- solves the $M = 0$, flat bed SIA: $\quad H_t = \nabla \cdot \left(\Gamma H^{n+2} |\nabla H|^{n-1} \nabla H\right)$
- calls `diffusion.m` at each major time step

verification of numerical ice flow codes

- how do we make sure a numerical scheme is correct?
- how do we make sure an *implemented* numerical scheme is correct?
  - *technique* 1: don't make any mistakes
  - *technique* 2: compare your model with others, and hope that the outliers are the ones with errors
  - *technique* 3: build-in a comparison to an exact solution, and actually measure the numerical error
- technique 3 is called verification

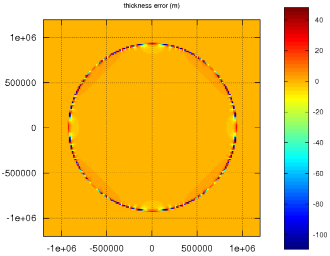
where to get exact solutions for ice flow models?

- textbook: Greve and Blatter (2009)
- similarity solutions to SIA (Halfar 1983; Bueler et al 2005)
- manufactured solutions to thermo-coupled SIA (Bueler et al 2007)
- flowline and cross-flow SSA solutions (van der Veen, 1985; Schoof, 2006)
- flowline Blatter solutions (Glowinski and Rappaz 2003)
- flowline Stokes solutions for constant viscosity (Ladyzhenskaya 1963; Balise and Raymond 1985)
- manufactured solutions to the Stokes equations (Sargent and Fastook 2010; Jouvet and Rappaz 2011)

verifying SIA code vs Halfar

test program `verifysia.m` calls `siaflat.m`

```
octave:40> verifysia(20)
average abs error              = 22.310
maximum abs error              = 227.849
octave:41> verifysia(40)
average abs error              = 9.490
maximum abs error              = 241.470
octave:42> verifysia(80)
average abs error              = 2.800
maximum abs error              = 155.796
octave:43> verifysia(160)
average abs error              = 1.059
maximum abs error              = 109.466
```



Trust but verify.

(Ronald Reagan)

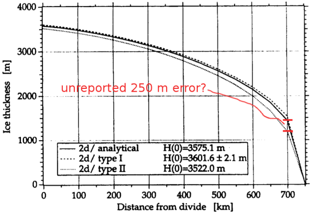
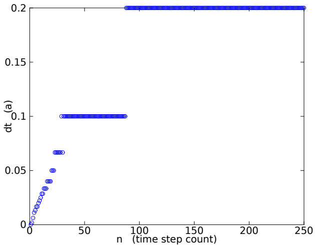
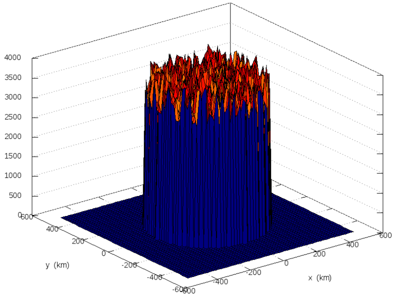


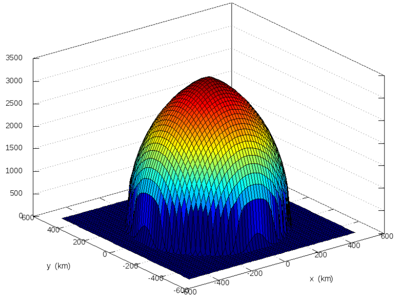
figure 2 in Huybrechts et al. (1996)

demonstrate robustness and adaptivity

- test program `roughice.m` calls `siaflat.m`
- sets up the nasty initial state (below left)
- runs for 50 years
- get final state (below right)
- time steps adapt (upper right)

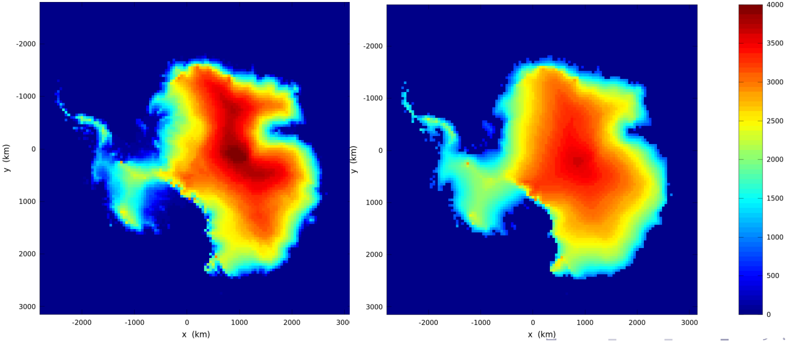






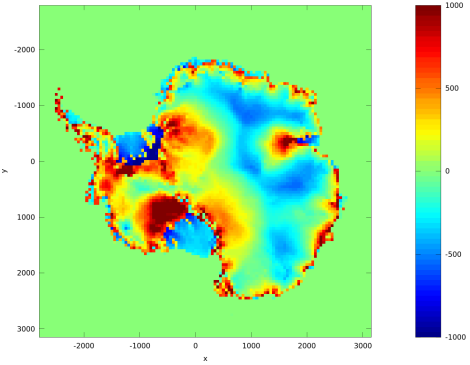
model of the Antarctic ice sheet

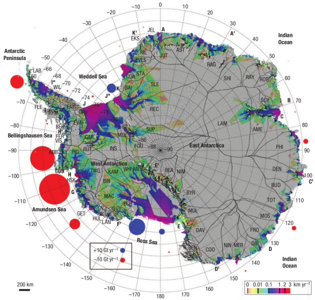
- careful-but-small modifications of `siaflat.m` give `siageneral.m`
  - allow non-flat bed
  - calve floating ice (*SIA can't handle it*)
- build a *toy* Antarctic flow model
  - observed accumulation as surface mass balance, sans melt
  - data from ALBMAPv1, a NetCDF file
- results from a 40 ka model run on a $\Delta x = 50$ km grid below
- runtime four minutes on laptop



model of the Antarctic ice sheet 2

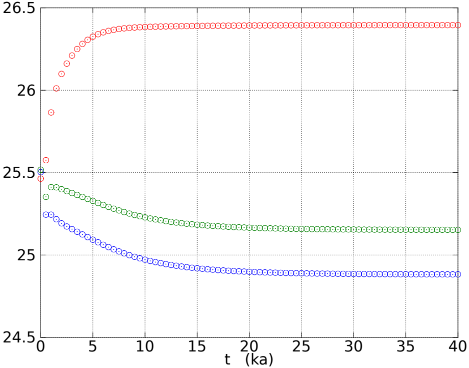
- thickness change from beginning to end of 50 km run
- thickness change is nearly a map of where flow is fast
  - why?





model of the Antarctic ice sheet 3

- volume time series from 50 km, 25 km, 20 km runs
- units of 10^6 km^3
- conclusion: look at your results on *multiple grid resolutions* before interpreting your results (e.g. their parameter-dependence)

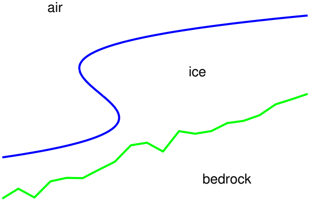


the most basic shallow assumption

- there are many shallow theories: SIA, SSA, hybrids, Blatter, . . .
- *all* make one assumption not required in the (non-shallow) Stokes theory:

  the surface and base of the ice are given by differentiable functions $z = h(t, x, y)$ and $z = b(t, x, y)$

- surface overhang is not allowed



not shallow!

not shallow!

kinematic and mass continuity equations

- what does this "most basic shallow assumption" get you?
- *answer:* a map-plane mass continuity equation
- consider these three equations:
  - the surface kinematical equation
  - the base kinematical equation
  - the map-plane mass continuity equation
- under the "most basic shallow assumption",

  any two imply the third

- these three equations are on the next slide

kinematic and mass continuity equations 2

- a is surface mass balance ($a > 0$ is accumulation)
- s is basal melt rate ($s > 0$ is basal melting)
- $M = a - s$
- define the map-plane flux of ice,

$$\mathbf{q} = \int_b^h (u, v)\, dz = \overline{\mathbf{U}}\, H$$

- the three equations are:

surface kinematical
$$\boxed{h_t = a - u\big|_h h_x - v\big|_h h_y + w\big|_h} \qquad (3)$$

base kinematical
$$\boxed{b_t = s - u\big|_b b_x - v\big|_b b_y + w\big|_b} \qquad (4)$$

mass continuity
$$\boxed{H_t = M - \nabla \cdot \mathbf{q}} \qquad (5)$$

kinematic and mass continuity equations 3

- how are the three equations related to each other?
- start with the incompressibility of ice

$$u_x + v_y + w_z = 0 \qquad \Longleftrightarrow \qquad w_z = -\nabla \cdot \mathbf{U}$$

- integrate vertically:

$$w\big|_h - w\big|_b = \int_b^h w_z\,dz = -\int_b^h \nabla \cdot \mathbf{U}\,dz$$

- now be careful and use the Leibniz rule for differentiating integrals:

$$\frac{d}{dx}\left(\int_{A(x)}^{B(x)} F(x,y)\,dy\right) = B'(x)F(x,B(x)) - A'(x)F(x,A(x)) + \int_{A(x)}^{B(x)} F_x(x,y)\,dy$$

- which implies that

$$-\nabla \cdot \int_b^h \mathbf{U}\,dz = -\nabla h \cdot \mathbf{U}\big|h + \nabla b \cdot \mathbf{U}\big|_b - \int_b^h \nabla \cdot \mathbf{U}\,dz$$

- so

$$w\big|_h - w\big|_b = -\nabla \cdot \mathbf{q} + \nabla h \cdot \mathbf{U}\big|h - \nabla b \cdot \mathbf{U}\big|_b$$

- and so on

kinematic and mass continuity equations 4

- why am I telling you this?
- because glaciology articles are filled with recapitulation of these equivalences

- my main points regarding mass continuity:
    - most ice sheet models use the mass continuity equation to describe change in ice sheet geometry
    - but they could instead use the surface kinematical equation
    - and they almost always use, and greatly simplify, the basal kinematical equation

standard recipe for ice sheet models

- the ingredients of a typical ice sheet model:
  1. numerical implementation of a stress balance: gives velocity $\mathbf{u} = (u, v, w)$
  2. from the horizontal velocity $\mathbf{U} = (u, v)$, or the horizontal flux $\mathbf{q}$, and the climatic mass (surface) balance M, do a time-step of the mass continuity equation to get $\Delta H = H_t \Delta t$
  3. update upper (and perhaps lower) surface elevation: $h \mapsto h + H_t \Delta t$
  4. decide on next time-step Δt, and repeat at 1.

- the non-sliding SIA is slightly atypical because we can write $\mathbf{q} = -D\nabla h$ in addition to $\mathbf{q} = \bar{\mathbf{U}} H$

shallow shelf approximation (SSA) stress balance

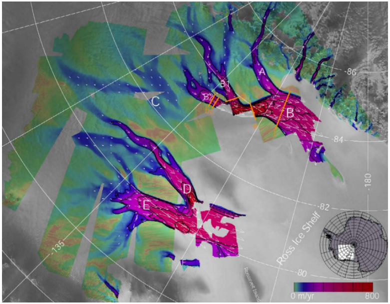
SSA model applies very well to ice shelves

- . . . for parts away from grounding lines
- . . . and away from calving fronts

shallow shelf approximation (SSA) stress balance 2

SSA also applies reasonably well to ice streams

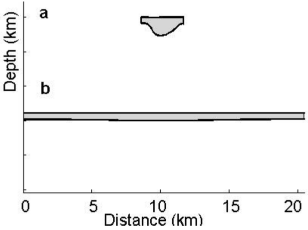
- ...with modest bed topography
- ...and weak bed strength
- ...but not so good near shear margins & grounding lines
- energy conservation (= ice temperature and basal melt) is a major aspect of ice stream flow, but not addressed here



RADARSAT-derived surface velocity for Siple Coast ice streams, Antarctica

what is, *and is not*, an ice stream?

- ice streams
  - slide (100 to 1000 $\mathrm{m\,a^{-1}}$)
  - have concentration of vertical shear in a thin layer near base
  - *liquid water* at bed has critical role
- "outlet glaciers"
  - have fast surface speed (up to 10 $\mathrm{km\,a^{-1}}$)
  - uncertain how much is true sliding
  - substantial vertical shear "up" in the ice column
  - not-at-all flat bed topography
  - soft, temperate ice plays a role
- therefore few simplifying assumptions are appropriate for outlet glaciers



Cross sections of Jakobshavns Isbrae (**a**) and Whillans Ice Stream (**b**). Plotted without vertical exaggeration. (Figure 1 in Truffer and Echelmeyer (2003), *Of isbrae and ice streams*)

SSA stress balance equation

- only plane flow case ("flow line") in these lectures
- the stress balance equation which determines velocity:

$$\boxed{\left(2BH|u_x|^{1/n-1}u_x\right)_x - C|u|^{m-1}u = \rho g H h_x} \qquad (6)$$

- the red term inside parentheses is the vertically-integrated "longitudinal" or "membrane" stress
- the blue term is basal resistance; $C = 0$ in an ice shelf
- the right-hand side is driving stress; see *next slide*
- *how to think about this equation*?
- *how do you solve it numerically*?

flotation criterion and grounding line

- the inequality "$\rho H < -\rho_w b$" is the flotation criterion
- at the grounding line $x = x_g$ the inequality switches
- driving stress:
  - on the grounded side $\rho H > -\rho_w b$, so

$$\rho g H h_x = \rho g H (H_x + b_x)$$

  - on the floating side $\rho H < -\rho_w b$, so $h = (1 - \rho/\rho_w)H$, and so

$$\rho g H h_x = \rho(1 - \rho/\rho_w) g H H_x$$

- best numerical models for moving grounding lines still an open question (e.g. MISMIP and all that)

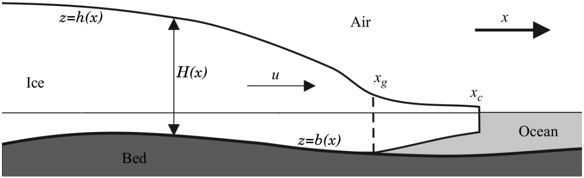
from stream to shelf across grounding line

$$u = u_0 \qquad \text{at } x = 0$$

$$\left.\begin{aligned}\left(2BH|u_x|^{1/n-1}u_x\right)_x - C|u|^{m-1}u &= \rho g H h_x \\ h &= H + b\end{aligned}\right\} \qquad \text{on } 0 < x < x_g$$

$$\left.\begin{aligned}\left(2BH|u_x|^{1/n-1}u_x\right)_x + 0 &= \rho g H h_x \\ h &= (1-\rho/\rho_w)H\end{aligned}\right\} \qquad \text{on } x_g < x < x_c$$

$$2BH|u_x|^{1/n-1}u_x = \frac{1}{2}\rho(1-\rho/\rho_w)gH^2 \qquad \text{at } x = x_c$$

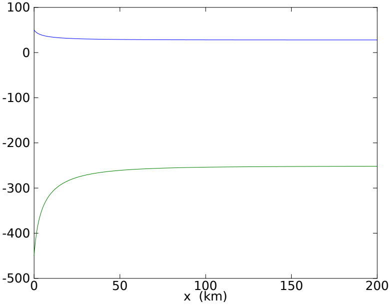


exact velocity and thickness for steady ice shelf

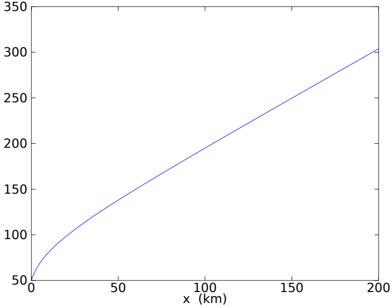
- limited goal here: describe a steady state, 1D ice shelf
- there is this nice by-hand result: the thickness and velocity in the ice shelf can be completely determined in terms of the
  - ice thickness H_g at the grounding line and
  - ice velocity u_g at the grounding line
- derived by van der Veen (1983)
- shown on next slide
- we will use this to
  - understand the SSA better
  - verify a numerical SSA code

exact velocity and thickness for steady ice shelf 2

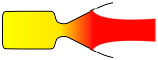
see `testshelf.m`:



surface and base elevation (m)



velocity (m/a)



numerically solving the SSA stress balance

- now we fix ice thickness $H(x)$ and find the velocity numerically
- the stress balance is a nonlinear equation in the velocity:

$$\left(2BH|u_x|^{1/n-1}u_x\right)_x - C|u|^{m-1}u = \rho g H h_x$$

- nonlinear so iteration is needed

Picard iteration

- coefficient $\bar{\nu} = B|u_x|^{1/n-1}$ is the "effective viscosity":

$$(2\,\bar{\nu}\,Hu_x)_x - C|u|^{m-1}u = \rho g H h_x$$

- *simplest iteration idea*: use initial velocity estimate to get effective viscosity estimate, then solve for new velocity, and repeat until things stop changing
  - this is "Picard" iteration
  - Newton iteration is a superior alternative
- the iteration in formulas:
  1. previous iterate $u^{(k-1)}$
  2. define $W^{(k-1)} = 2\bar{\nu}H = 2B|u_x^{(k-1)}|^{1/n-1}H$
  3. solve for current iterate (unknown) $u^{(k)}$:

$$\left(W^{(k-1)}u_x^{(k)}\right)_x - C|u^{(k-1)}|^{m-1}u^{(k)} = \rho g H h_x$$

  4. repeat at 1

where do you get an initial velocity $u^{(0)}$?

- *for floating ice*, an initial guess assumes uniform strain rate:

$$u^{(0)}(x) = \gamma(x - x_g) + u_g$$

  where γ is the value of u_x from calving front stress imbalance
- *for grounded ice*, an initial guess assumes ice is held by basal resistance only:

$$u^{(0)}(x) = \left(-C^{-1}\rho g H h_x\right)^{1/m}$$

a linear problem at each iteration

- abstract the problem:

$$\left(W(x)\,u_x\right)_x - \alpha(x)\,u = \beta(x)$$

on $0 < x < L$, with boundary conditions

$$u(0) = V, \qquad u_x(L) = \gamma$$

- an *linear*, elliptic PDE boundary value problem
- $W(x)$, $\alpha(x)$, $\beta(x)$ are known functions in the SSA context:
  - both $W(x)$ and $\alpha(x)$ come from previous iteration
  - $\beta(x)$ is driving stress

numerics of the linear problem

- equally-spaced grid $x_1, x_2, \dots, x_{J+1}$
  - $x_1 = x_g$ and $x_{J+1} = x_c$ are endpoints
- the finite difference approximation to the linear PDE is:

$$\frac{W_{j+1/2}(u_{j+1} - u_j) - W_{j-1/2}(u_j - u_{j-1})}{\Delta x^2} - \alpha_j u_j \stackrel{*}{=} \beta_j$$

- so $W(x)$ is needed on the staggered grid
- left-hand boundary condition: $u_1 = V$ given
- right-hand boundary condition is $u_x(L) = \gamma$
- to put right-hand b.c. into system:
  - introduce notional point x_{J+2}
  - $$\frac{u_{J+2} - u_J}{2\Delta x} = \gamma$$
  - using equation $*$ in $j = J + 1$ case, eliminate u_{J+2} variable by-hand before solving system

numerics of the linear problem 2

- thus discretized stress balance has form $A\mathbf{x} = \mathbf{b}$:

$$\begin{bmatrix} 1 & & & & \\ W_{3/2} & A_{22} & W_{5/2} & & \\ & W_{5/2} & A_{33} & & \\ & & \ddots & \ddots & \\ & & W_{J-1/2} & A_{JJ} & W_{J+1/2} \\ & & & A_{J+1,J} & A_{J+1,J+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_J \\ u_{J+1} \end{bmatrix} = \begin{bmatrix} 0 \\ \beta_2 \Delta x^2 \\ \beta_3 \Delta x^2 \\ \vdots \\ \beta_J \Delta x^2 \\ b_{J+1} \end{bmatrix}$$

- with specific formulas on diagonal (A_{jj}) and in the last equation
    - see printed notes
- this is a *tridiagonal* system
- hand the system to a linear algebra black box

solver for the linear problem

```
function u = flowline(L,J,gamma,W,alpha,beta,V0)

dx = L / J;
rhs = dx^2 * beta(:);
rhs(1) = V0;
rhs(J+1) = rhs(J+1) - 2 * gamma * dx * W(J+1);

A = sparse(J+1,J+1);
A(1,1) = 1.0;
for j=2:J
  A(j,j-1:j+1) = [ W(j-1), -(W(j-1) + W(j) + alpha(j) * dx^2), W(j) ];
end
A(J+1,J) = W(J) + W(J+1);
A(J+1,J+1) = - (W(J) + W(J+1) + alpha(J+1) * dx^2);

scale = full(max(abs(A),[],2));
for j=1:J+1,  A(j,:) = A(j,:) ./ scale(j);  end
rhs = rhs ./ scale;

u = A \ rhs;
```

flowline.m

- solves

$$(W(x)\, u_x)_x - \alpha(x)\, u = \beta(x)$$

testing the linear PDE solver

- before proceeding to solve nonlinear SSA problem, we can test the "abstracted" code `flowline.m`
- test by "manufacturing" solutions
  - see `testflowline.m`; not shown
- results: converges at optimal rate $O(\Delta x^2)$

numerical: SSA

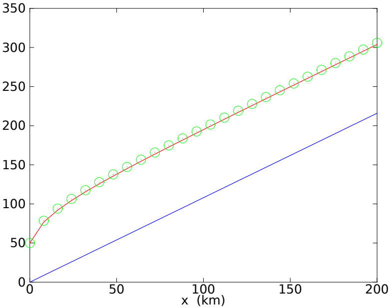
```
function [u,u0] = ssaflowline(p,J,H,b,ug,initchoice)

if nargin ~= 6, error('exactly 6 input arguments required'), end

dx = p.L / J;
x = (0:dx:p.L)';
xstag = (dx/2:dx:p.L+dx/2)';

alpha = p.C * ones(size(x));
h = H + b;
hx = regslope(dx,h);
beta = p.rho * p.g * H .* hx;
gamma = ( 0.25 * p.A^(1/p.n) * (1 - p.rho/p.rhow) *...
          p.rho * p.g * H(end) )^p.n;

u0 = ssainit(p,x,beta,gamma,initchoice);
u = u0;

Hstag = stagav(H);
tol = 1.0e-14;
eps_reg = (1.0 / p.secpera) / p.L;
maxdiff = Inf;
W = zeros(J+1,1);
iter = 0;
while maxdiff > tol
  uxstag = stagslope(dx,u);
  sqr_ux_reg = uxstag.^2 + eps_reg^2;
  W(1:J) = 2 * p.A^(-1/p.n) * Hstag .* sqr_ux_reg.^(((1/p.n)-1)/2.0);
  W(J+1) = W(J);

  unew = flowline(p.L,J,gamma,W,alpha,beta,ug);
  maxdiff = max(abs(unew-u));
  u = unew;
  iter = iter + 1;
end
```

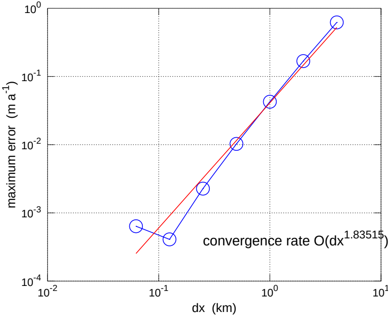
`ssaflowline.m`

- implements the Picard iteration
- calls the linear solver `flowline.m` at each iteration

numerical velocity for a steady ice shelf

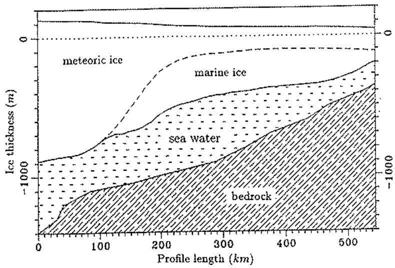
- left: exact (red curve) and numerical (green circles) velocity
  - for very coarse 8 km grid
  - blue is initial velocity iterate
- right: convergence analysis of `ssaflowline.m`
  - error = max(numerical - exact), over various grids





realistic ice shelf modeling

- flow lines are never very realistic
- you can add "side drag"
  - I don't know how to parameterize it
- also, ice shelves have surprises:
  - high basal melt near grounding lines
  - "reverse bed slope" instability
  - marine ice can freeze-on at bottom (below)

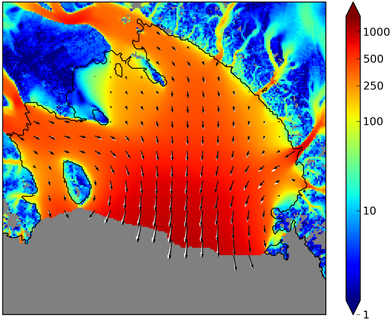


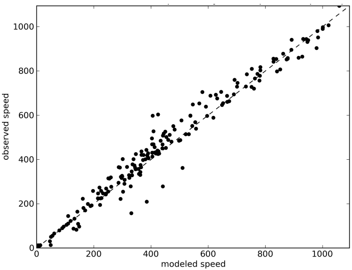
Filchner-Ronne ice shelf; from Grosfeld & Thyssen 1994

ice shelf modeling in 2D

- "diagnostic" (static geometry) ice shelf modeling in two horizontal variables has been quite successful
- observed surface velocities validate SSA stress balance
  - Ross ice shelf example below using PISM
  - many numerical models can do this

ice velocity (m/year); white=observed, black=model



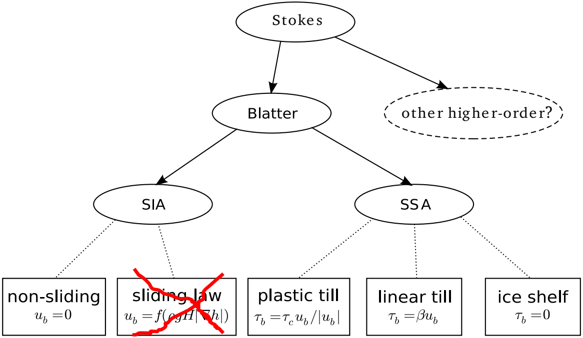


numerical solution of stress balances: a summary

- stress balance equations (e.g. SSA) determine horizontal velocity from geometry and boundary conditions
  - nonlinear so iteration is necessary
  - at each iteration a sparse matrix linear problem is solved
  - give the linear problem to a matrix solver software package

"higher-order" schemes

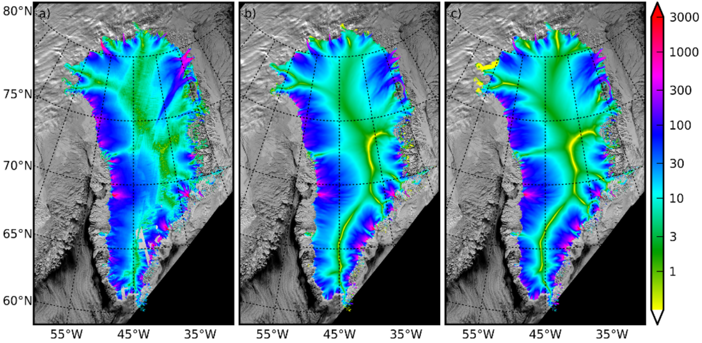
- both the SIA and the SSA are derived by small-parameter arguments from the Stokes equations
- so: is there a *common* shallow antecedent model?
- Schoof and Hindmarsh (2009) answer: *yes*, the model derived by Blatter (1995) is one



- my advice: don't use the crossed-out one
  - sliding is not *locally* controlled by driving stress alone

shallow hybrid schemes

- so how about “gluing together” SIA and SSA in “hybrid”?
- there are multiple schemes (Pollard and deConto, 2007; Bueler and Brown, 2009; Goldberg 2011)
- shallow hybrids can be used *now* at high spatial resolution and paleoglacial time scales
- for example, we can compare observed (left) and 1km grid modeled (middle and right; PISM) surface velocity maps for the Greenland ice sheet



the mass continuity equation: a summary

- the *mass continuity equation* is
$$H_t = M - \nabla \cdot (\mathbf{u} H)$$
- the numerical nature of this equation depends on the stress balance:
  - the equation is a diffusion for frozen bed, large scale flows (i.e. SIA)
  - it is *not* very diffusive for membrane stresses and no basal resistance (e.g. SSA for ice shelves)
  - transitional for ice streams and outlet glaciers
    - that is: how much role for longitudinal stresses?
  - not much helpful theory on this transport problem
  - maybe *you* will help find this theory!

The End

- that's the end
- questions?

POSTSCRIPT on technical skills

- next two slides are free advice on technical skills needed for numerical ice sheet modelling
  - you get what you pay for

technical skills for numerical ice sheet modelling

- you need comfort in a technical computing environment, including
  - an editor,
  - a scripting/prototyping language (Matlab, Python, etc.),
  - a compiled language (C or Fortran),
  - a version control system (Subversion, git, etc.), and
  - some tools for NetCDF files
- you need willingness to read physics, numerical analysis, computer science, etc. books

technical skills for numerical ice sheet modelling 2

- you should *never* re-invent the wheel for basic numerics like these, *except* to write throw-away codes to help you understand them:
  - numerical linear algebra
  - mesh generation
  - finite element assembly and solve
- try existing open source ice flow models:
  - shallow comprehensive models:
    - GLIMMER
    - SICOPOLIS
    - PISM
  - open source higher-order/Stokes models:
    - Elmer/Ice
    - ISSM

POSTSCRIPT have I oversold diffusivity?

- I have asserted that the default model for ice sheets, the SIA, is a *diffusion* like the heat equation
- recall the analogy:

heat eqn	$\leftrightarrow$	*SIA*
$T_t = \nabla \cdot (D \nabla T)$	$\leftrightarrow$	$H_t = M + \nabla \cdot (D \nabla h)$
$D = D(x, y)$	$\leftrightarrow$	$D = \Gamma H^{n+2} \lvert\nabla h\rvert^{n-1}$

- have I oversold this diffusivity analogy?
  - possibly
  - I've acknowledged there are "issues"
- but it turns out there is some robustness to the analogy

have I oversold diffusivity? 2

DIFFUSIVE IDEA 1: rough beds have the effect of *reducing* diffusivity

- define the local bed topography by removing the local mean over some range $\lambda \approx 10$ km:

$$\tilde{b}(x,\xi) = b(x+\xi) - \fint_{-\lambda}^{\lambda} b(x+\xi)\, d\xi$$

- define this average of the local bed:

$$\theta(x) = \left(\fint_{-\lambda}^{\lambda} \left(1 - \frac{\tilde{b}(x,\xi)}{H(x)}\right)^{-(n+2)/n} d\xi \right)^{-1/n}$$

- using a multiple-scales analysis, Schoof [2003] says you will get closer to solving the Stokes equations by making these two modifications of the SIA:
  - smooth the bed
  - but don't lose track of the smoothed-away local bed roughness; use it to reduce the diffusivity:

$$D_{\text{new}} = \theta D \qquad \text{where } 0 \le \theta \le 1$$

have I oversold diffusivity? 3

DIFFUSIVE IDEA 2: the large-scale effect of *sliding* in ice streams (*addressed in next section*), is also diffusive

- suppose that, for an ice stream modeled by the SSA equation

$$\left(2A^{-1/n}H|u_x|^{1/n-1}u_x\right)_x - C|u|^{m-1}u = \rho g H h_x$$

  we assume that the basal resistance term balances the driving stress:

$$-C|u|^{m-1}u = \rho g H h_x$$

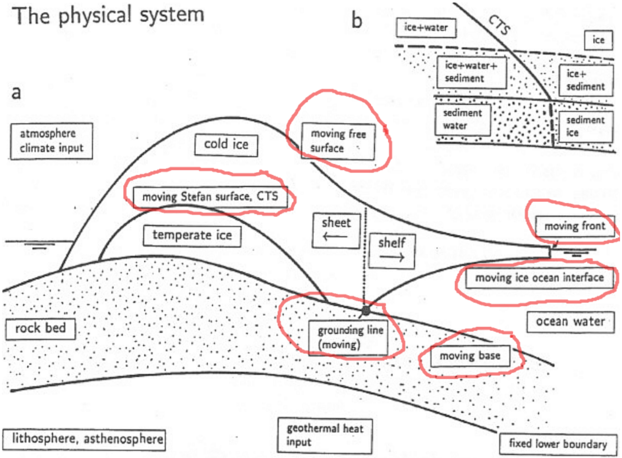
- then the ice stream geometry evolves by a non-SIA diffusion,

$$H_t = M + \nabla \cdot (D\nabla h) \qquad \text{where } D = C' H^{(1/m)+1}|\nabla h|^{(1/m)-1}$$

- this "outer problem" is part of a matched asymptotic expansion that *does a good job of tracking the grounding line in a marine ice sheet* [Schoof, 2007]

POSTSCRIPT on free boundary value problems

- ice sheet/shelf modelling means free boundary problems
- Hutter [1999] identifies some:



what is a "free boundary"?

- a *free boundary* for a PDE is an unknown location at which there is a boundary condition
  - the location of the free boundary must be found at the same time as one solves the PDE problem
  - there must be enough additional information at a free boundary to determine its location
  - all free boundary problems are nonlinear, even if the PDE is linear
- classic *example*: consider an elastic membrane attached to a wire frame and stretched over an obstacle:

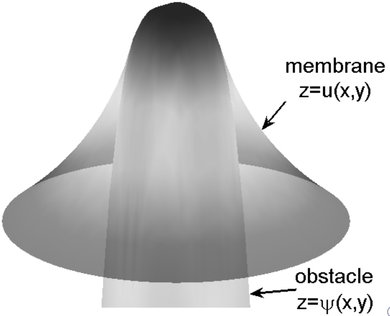
constraint:

$$u \geq \psi$$

in locations where $u > \psi$, solve:

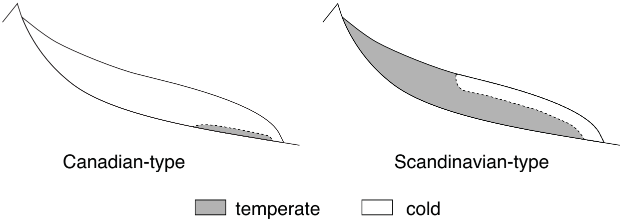
$$\nabla^2 u = 0$$

where is the free boundary, and what facts about u are true there?



free boundary value problem 1: polythermal ice

- by volume, majority of ice sheet is *cold* ($T < 0\ ^\circ$C)
- ...but there is some ice which is *temperate*, where $T = 0\ ^\circ$C *and* there is a positive liquid fraction within the ice matrix
- ice sheets are *polythermal*
- boundary between cold and temperate ice is "CTS" (= cold-temperate transition surface):
  - must be found, as free boundary, when solving conservation of energy equation ("Stefan problem")
  - can be tracked explicitly [Greve, 1997]
  - or treated as a level surface of the enthalpy variable [Aschwanden et al, 2012]

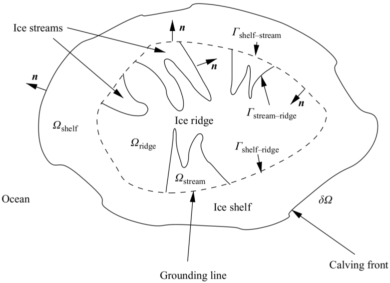


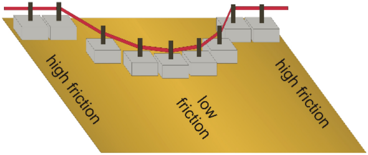
free boundary value problem 2: for ice streaming

- Schoof's [2006] insight, for diagnostic case

$$\text{SSA} + (\text{plastic till}) = \begin{pmatrix} \text{well-posed free boundary problem} \\ \text{for location } \textit{and} \text{ velocity of sliding} \end{pmatrix}$$

- "plastic till" means the basal strength (resistance) is given by a yield stress τ_c: $\vec{\tau}_b = \tau_c \mathbf{v}_b / |\mathbf{v}_b|$
- Schoof's scheme is a *whole ice sheet form* of MacAyeal's [1989] individual ice stream model





free boundary problem 3: for grounded ice sheet margin

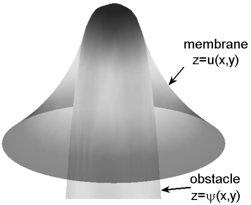
- side-by-side comparison, *classical elastic membrane problem* versus *steady ice sheet problem*
- Jouvet & Bueler [2012] show this is well-posed

constraint:

$$u \geq \psi$$

where $u > \psi$, solve:

$$\nabla^2 u = 0$$

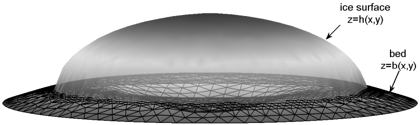


constraint:

$$h \geq b \qquad \Longleftrightarrow \qquad H \geq 0$$

where $h > b$, solve steady SIA:

$$0 = M + \nabla \cdot \left(\Gamma H^{n+2} |\nabla h|^{n-1} \nabla h \right)$$



free boundary problems: why do they matter?

- location of boundary may *be* the glaciological question
- free boundaries are always locations of *loss of smoothness* relative to fixed boundary solutions
  - numerical errors dominated by errors near free boundaries
  - are model results at free boundaries poor because of numerical problems or because of missing physical processes?