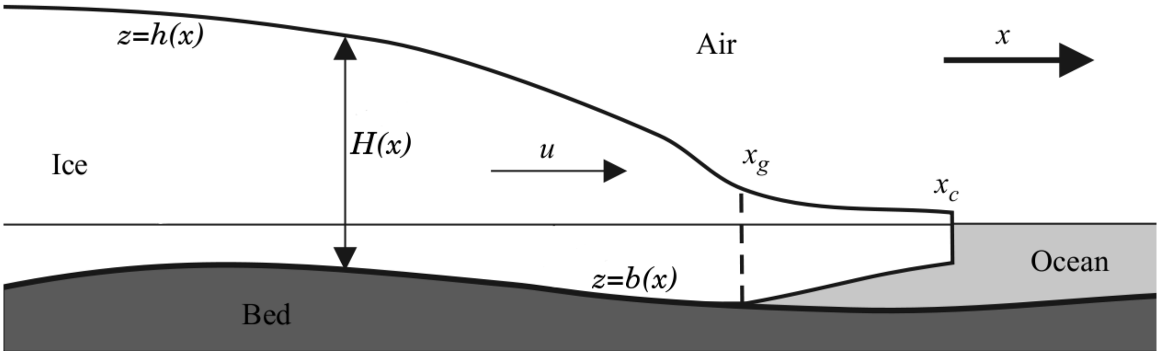
Numerical modelling

# of glaciers, ice sheets, and ice shelves

Ed Bueler

International Summer School in Glaciology
McCarthy Alaska
2010, 2012, 2014, 2016, 2018, 2022, 2024



Illustrates the notation used in these notes. Figure modified from [61].

1. Introduction

The greatest importance of mathematical and numerical models to science, and to glaciology in particular, is in *introspection*. In building and using a model you are asking yourself: When I combine my imperfect and incomplete understanding of glacier processes into this model does it behave as I expect? A person who does not actually understand your scientific explanation might nod and say "yes, I understand." The dumb computer, however, can only tell you consequences of your model, inadequacies and all. Numerical models can show how processes interact to give overall behavior, and they can demonstrate flaws in understanding of those processes.

But numerical models must be built with care. An avoidable bad outcome is to spend time—or worse, reputation—interpreting, explaining, or justifying model behavior that is actually an artifact of buggy computer programming or flawed numerical analysis.

Continuum models, and not computer codes, will be the primary focus of these notes. (This might be surprising!) While all codes produce numbers, we want numbers that actually come from the intended continuum model. Only careful analysis of a numerical implementation can confirm that it matches the model equations.

Scope. These notes have a limited scope:

shallow approximations of ice flow.

They adopt a constructive approach:

we provide understandable numerical codes that actually run.

Our scope includes the shallow ice approximation (SIA) in two horizontal dimensions (2D), the shallow shelf approximation (SSA) in 1D, and the mass continuity and surface kinematical equations. (These technical terms will be defined in due course.) We also recall the Stokes model here, and a separate Appendix addresses its numerical solution.

Numerical concepts within our scope include finite difference schemes, solving algebraic systems from stress balances, and the verification of codes using exact solutions.

Notation. Our notation, which generally follows [27], is covered by Table 1. Cartesian coordinates are x, y, z with z positive-upward, plus time t. These coordinates appear as subscripts to denote partial derivatives: $u_x = \partial u/\partial x$. Tensor notation uses subscripts from the list $\{1, 2, 3, i, j\}$. Thus τ_{ij}, τ_{13} denote entries of the deviatoric stress tensor.

Downloading codes. These notes are based on nineteen Matlab codes, each about one-half page. All have been tested in Matlab and Octave. They are distributed by cloning the `git` repository at

`https://github.com/bueler/mccarthy`

The codes are in the `mfiles/` subdirectory. Five of them are printed below with their comments stripped for compactness and clarity. The electronic versions have generous comments and help files.

2. Ice flow equations

Our initial goal is an equation for which one might say:

by numerically solving this equation one has a usable model for an ice sheet.

Table 1. Notation used in these notes.

variable	description	SI units	value
A	$A = A(T) =$ ice softness in Glen law	$\mathrm{Pa}^{-n}\,\mathrm{s}^{-1}$	
B	ice hardness; $B = A^{-1/n}$	$\mathrm{Pa}\,\mathrm{s}^{1/n}$	
b	bedrock elevation	m	
c	specific heat	$\mathrm{J\ kg}^{-1}\ \mathrm{K}^{-1}$	
∇	(spatial) gradient	m^{-1}	
$\nabla\cdot$	(spatial) divergence	m^{-1}	
$\mathbf{g}$	gravity	$\mathrm{m\ s}^{-2}$	9.81
H	ice thickness	m	
h	ice surface elevation	m	
κ	temperature conductivity	$\mathrm{J\ s}^{-1}\ \mathrm{m}^{-1}\ \mathrm{K}^{-1}$	
M	climatic mass balance	$\mathrm{m\ s}^{-1}$	
n	exponent in Glen flow law		3
ν	viscosity	Pa s	
p	pressure	Pa	
$\mathbf{q}$	map-plane ice flux: $\mathbf{q} = \int_b^h \mathbf{U}\,dx$	$\mathrm{m}^2\,\mathrm{s}^{-1}$	
ρ	density of ice	$\mathrm{kg\ m}^{-3}$	910
ρ_w	density of sea water	$\mathrm{kg\ m}^{-3}$	1028
σ_{ij}	Cauchy stress tensor: $\sigma_{ij} = \tau_{ij} - p\,\delta_{ij}$	Pa	
T	temperature	K	
τ_{ij}	deviatoric stress tensor	Pa	
Du_{ij}	strain rate tensor	s^{-1}	
$\mathbf{U}$	$= (u, v)$ horizontal ice velocity	$\mathrm{m\ s}^{-1}$	
$\mathbf{u}$	$(u_1, u_2, u_3) = (u, v, w)$ 3D ice velocity	$\mathrm{m\ s}^{-1}$	
x_i	$x_1, x_2, x_3 = x, y, z$ coordinates	m	

A usable model must be *understood* at least as much as it is *correct* in a physical sense. Our initial model will not be complete by any modern standard.

First we (briefly!) recall the continuum mechanical equations of ice flow. Ice in glaciers is a moving fluid so we describe its motion by a velocity field $\mathbf{u}(t, x, y, z)$. If the ice fluid were faster-moving than in reality, and if it were linearly-viscous like liquid water, then it would be a "typical" incompressible fluid and we would use the Navier-Stokes equations as the model:

$$\nabla \cdot \mathbf{u} = 0 \qquad \textit{incompressibility} \qquad (1)$$

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

Equation (2) says "$ma = F$"; it is Newton's second law. The term $\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}$ is an acceleration while the right-hand side is net force per unit volume, from viscous, pressure, and gravity forces respectively. If σ is the Cauchy stress tensor, the right side can also be written $\nabla\cdot\sigma + \rho\mathbf{g}$.

The space of solutions to (1) and (2) is rich. It includes turbulent flows, which we will not need here. Reading a book-length introduction to fluids such as [1] is overkill for glacier modeling, but it is recommended for broader education.

The numerical solution of the Navier-Stokes equations is *computational fluid dynamics* (CFD). One might ask: is ice flow modelling really a part of CFD? Does a good general-purpose CFD text like [71] help a glaciers student? While glacier flow is a large-scale fluid problem, comparable to atmosphere and ocean circulation, from the point of view of general CFD it is strange. Topics which might make ocean circulation exciting, for example turbulence, convection, coriolis force, and density stratification, are irelevant to ice flow. What could be interesting about the flow of slow, thick ice?

The Glen-Stokes equations. Glacier ice is a slow fluid. In terms of equation (2), "slow" is a reasonably-precise term which means that $\rho\left(\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}\right) \approx 0$, in other words that the forces (stresses) of inertia are negligible. However, ice is also a shear-thinning fluid with a specific kind of nonlinearly-viscous ("non-Newtonian") behavior in which larger strain rates imply smaller viscosity. The viscosity ν in (2) is therefore not constant, and one must add an empirically-based flow law. Combining these ideas gives the standard model for isothermal ice flow:

$$\nabla\cdot\mathbf{u} = 0 \qquad \textit{incompressibility} \tag{3}$$

$$-\nabla\cdot\tau_{ij} + \nabla p = \rho\mathbf{g} \qquad \textit{stress balance} \tag{4}$$

$$Du_{ij} = A\tau^2\tau_{ij} \qquad \textit{n=3 Glen flow law} \tag{5}$$

In (5) the deviatoric stress tensor τ_{ij} and the strain rate tensor Du_{ij} now appear. Other lectures cover these ideas, but recall that by definition

$$Du_{ij} = \frac{1}{2}\left((u_i)_{x_j} + (u_j)_{x_i}\right).$$

Both tensors in (5) are symmetric and have trace zero. Note that $\tau^2 = (1/2)\tau_{ij}\tau_{ij}$, using the summation convention, which defines the *effective stress* τ.

Because the Stokes equations do not contain a time derivative, the combined effect of boundary stresses, gravity $\rho\mathbf{g} = \langle 0, 0, -\rho g\rangle$, and a value for ice softness $A > 0$ will determine the velocity and stress fields ($\mathbf{u}$, p, τ_{ij}). Equations (3)–(5) apply at each instant, even as the glacier geometry and the solution fields change. Ice flow simulation codes need no memory of prior momentum or velocity. Said another way, velocity is a "diagnostic" output of ice flow models. Unlike in a Navier-Stokes-based simulation, velocity is not needed for (re)starting a simulation.

Consider now the x,z-plane case of equations (3)–(5). Such a planar flow has velocity component $v = 0$. All derivatives with respect to y are also zero so the equations are:

$$u_x + w_z = 0 \qquad \textit{incompressibility} \tag{6}$$

$$-\tau_{11,x} - \tau_{13,z} + p_x = 0 \qquad \textit{stress balance (x)} \tag{7}$$

$$-\tau_{13,x} - \tau_{33,z} + p_z = -\rho g \qquad \textit{stress balance (z)} \tag{8}$$

$$u_x = A\tau^2\tau_{11} \qquad \textit{flow law (diagonal)} \tag{9}$$

$$u_z + w_x = 2A\tau^2\tau_{13} \qquad \textit{flow law (off-diagonal)} \tag{10}$$

Note that τ_{13} is a shear stress while τ_{11} and $\tau_{33} = -\tau_{11}$ are deviatoric longitudinal stresses. Also $\tau^2 = \tau_{11}^2 + \tau_{13}^2$ in this case. Equations (6)–(10) form a system of five nonlinear equations in five scalar unknowns $(u, w, p, \tau_{11}, \tau_{13})$.

Slab-on-a-slope. Equations (6)–(10) are complicated enough already! They should make us pause before jumping into numerical solution methods. For now we start with a simplified situation, namely a uniform slab of ice, in which we can both solve the Stokes equations exactly and motivate the shallow model in the next subsection.

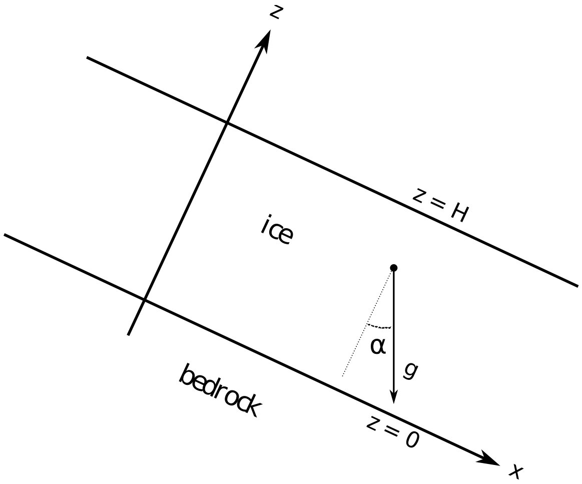


FIGURE 1. Rotated axes for a slab-on-a-slope flow calculation.

We rotate our coordinates only for this example. The two-dimensional axes (x,z) shown in Figure 1 are rotated clockwise at angle $\alpha > 0$ so that the gravity vector has components $\mathbf{g} = (g \sin\alpha, -g\cos\alpha)$ in the new axes. Equations (7) and (8) are now

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

$$-\tau_{13,x} + \tau_{11,z} + p_z = -\rho g \cos\alpha. \tag{12}$$

Assume further that there is no variation with x. Then the whole set of Stokes equations (6), (9)–(12) simplifies greatly:

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

Each unknown is now a function of z only. Boundary conditions come from assuming that the ice is not crossing the bed, that the ice surface is moving parallel to the x-axis, and that the ice surface is stress free: $w(0) = 0$, $p(H) = 0$, $\tau_{13}(H) = 0$. Then, by integrating equations (13) with respect to z, we get $w = 0$, $p = (\rho g\cos\alpha)(H - z)$, and $\tau_{13} = (\rho g \sin\alpha)(H - z)$. Note that $H - z$ is the depth below the ice surface; both pressure and shear stress are proportional to depth. Integrating the remaining equation in (13) yields horizontal velocity:

$$u = u_0 + \frac{1}{2}A(\rho g \sin\alpha)^3 \left(H^4 - (H - z)^4\right) \tag{14}$$

The basal velocity $u_0 = u(0)$ remains undetermined for now.

Do we believe formula (14), which makes a specific prediction about the profile of flow in a glacier? Figure 2 compares observations of a mountain glacier, showing that our model does a credible job of capturing deformation flow in this case.

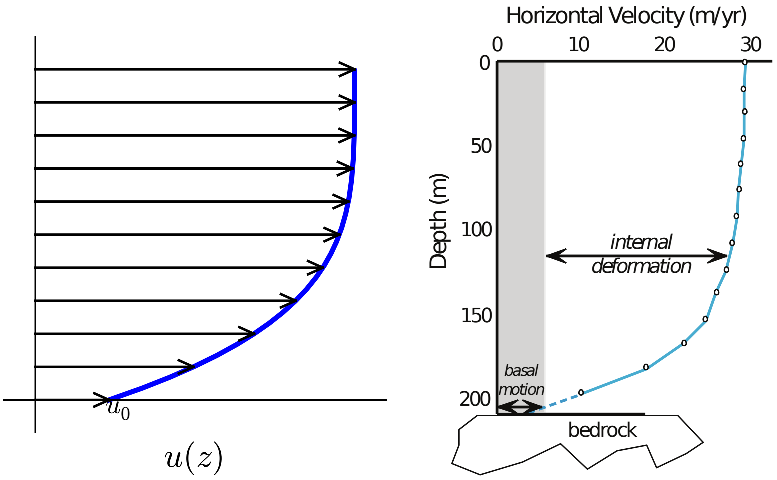


FIGURE 2. Left: Velocity from formula (14). Right: Measured velocity in a glacier (Athabasca Glacier [59]).

Until section 9 we will not have a reasonable dynamical model for the sliding velocity u_0. We assume until then that the ice is frozen to the bed and there is no sliding.

Viscosity form of the flow law. Flow law (5) has another form. Recall $\tau^2 = (1/2)\tau_{ij}\tau_{ij}$; also define $|Du|^2 = (1/2)Du_{ij}Du_{ij}$. (The scalars τ and $|Du|$ are tensor *norms*.) From (5) we get $|Du| = A\tau^3$ so $\tau = A^{-1/3}|Du|^{1/3}$. Thus (5) can be rewritten

$$\tau_{ij} = 2\nu\, Du_{ij} \qquad \textit{flow law} \tag{15}$$

where

$$2\nu = A^{-1/3}|Du|^{-2/3}$$

defines the nonlinear *viscosity* ν. Often $B = A^{-1/3}$ is called the *ice hardness*. Equation (15) allows us to eliminate stresses τ_{ij} from the Stokes equations, resulting in formulas depending only on the strain rates Du_{ij}.

The Blatter-Pattyn approximation. One may approximate the plane-flow Stokes equations (6)–(10) by assuming that horizontal variation in the vertical shear stress is small compared to the other terms. This removes the single term $\tau_{13,x}$ from the z-component of the stress balance (8):

$$p_z = -\tau_{11,z} - \rho g. \tag{16}$$

Because $\sigma_{ij} = \tau_{ij} - p\delta_{ij}$, thus $p + \tau_{11} = p - \tau_{33} = -\sigma_{33}$, equation (16) says that the vertical normal stress σ_{33} is linear in depth. (A slightly-different model assumes that the pressure p itself is hydrostatic.) Taking σ_{33} to have surface value zero we get

$$p + \tau_{11} = \rho g(h - z). \tag{17}$$

Equation (17) allows elimination of the pressure p from the model equations. Furthermore, taking the x-derivative of (17), then substituting into (7) to eliminate p_x, and then using the viscosity form (15) leads to this equation [27]:

$$(4\nu u_x)_x + (\nu(u_z + w_x))_z = \rho g h_x \qquad \textit{hydrostatic stress balance} \tag{18}$$

Equations (6), (15), (18), plus appropriate boundary conditions, determine u and w and then all remaining unknowns. However, although p is gone, equation (18) is nontrivially-coupled to incompressibility (6) because the vertical velocity w appears in both places.

If we additionally drop w_x from equation (18) then we get the Blatter-Pattyn model:

$$(4\nu u_x)_x + (\nu u_z)_z = \rho g h_x \qquad \textit{Blatter-Pattyn stress balance} \tag{19}$$

Solving (19) as a boundary value problem yields the horizontal velocity u. Then one may recover w from incompressibility (6). Because it is somewhat similar to the well-understood Poisson equation $u_{xx} + u_{zz} = f$, as described in [40, 48] for example, the process of solving equation (19) is more familiar to many professionals than (18).

Plane-flow mass-continuity equation. The equations so far do not address how a glacier or ice sheet changes shape. For this we need to add a mass conservation equation. We derive it here in an informal way, returning to the topic in section 8.

Consider any x, z-plane flow with variable thickness and velocity, not just slab-on-a-slope. Define the vertical average of the horizontal velocity:

$$\bar U = \frac{1}{H}\int_0^H u\, dz.$$

The ice flux $q = \int_0^H u\, dz$ is the rate of flow input into the side of the area in Figure 3. Ice can be added by climatic (surface) mass balance M or by a difference in the flux $q = \bar U H$ between the left and right sides.

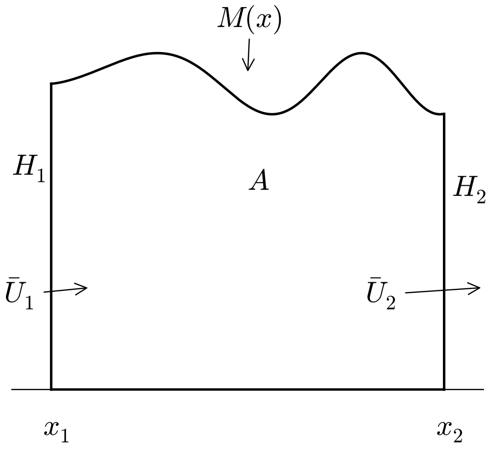


FIGURE 3. Mass continuity equation (21) follows from considering this domain of ice, which has time-dependent area A.

The area A changes according to the sum of all the boundary contributions:

$$\frac{dA}{dt} = \int_{x_1}^{x_2} M(x,t)\,dx + \bar{U}_1 H_1 - \bar{U}_2 H_2. \tag{20}$$

In three-dimensions, (20) becomes an equation for dV/dt, the ice volume rate of change.

If the width $\Delta x = x_2 - x_1$ is small then $A \approx \Delta x\, H$. So we divide by Δx and take $\Delta x \to 0$ in (20) to get

$$H_t = M - \left(\bar{U}H\right)_x \tag{21}$$

This *mass continuity* differential equation describes change in the ice thickness in terms of surface mass balance and ice velocity. Ice flow simulations can compute the velocity u and then use this equation for the changing glacier geometry.

3. Shallow ice sheets

Ice sheet flow has four outstanding properties as a fluids problem. Ice sheets are

(i) slow,
(ii) shallow,
(iii) non-Newtonian (shear-thinning), and
(iv) subject to widespread contact slip (basal sliding).

The first numerical ice flow model in these notes accounts for the first three properties only. It is the non-sliding, isothermal *shallow ice approximation* (SIA).

Regarding shallowness (ii), Figure 4 shows both a no-vertical-exaggeration cross-section of Greenland at 71° and the standard vertically-exaggerated version which is more familiar in the glaciological literature. Without vertical exaggeration we see the ice sheet as a thin blanket, barely three-dimensional. This illustrates that ice sheets are shallow. Note, however, that the portion of an ice sheet which you seek to model may not be shallow.

Our earlier slab-on-a-slope example can be converted into a rough derivation of the plane-flow SIA. We vertically integrate velocity formula (14) in the $u_0 = 0$ case to get

$$\bar{U}H = \int_0^H \frac{1}{2} A(\rho g \sin\alpha)^3 \left(H^4 - (H-z)^4\right)\, dz = \frac{2}{5} A(\rho g \sin\alpha)^3 H^5. \tag{22}$$

Regarding the surface slope, note $\sin\alpha \approx \tan\alpha = -h_x$ when α is a small angle. Combining these statements with mass continuity (21) gives

$$H_t = M + \left(\frac{2}{5}(\rho g)^3 A H^5 |h_x|^2 h_x\right)_x. \tag{23}$$

Equation (23) is the SIA equation for nonsliding plane flow. Noting $h = H + b$, it models the evolution of an ice sheet's thickness H. The model should, however, be solved subject to the constraint that the thickness is positive ($H \geq 0$) [11, 35]. For careful SIA derivations, which reduce the Stokes equations under a small depth-to-width ratio assumption, follow the Notes and References at the end.

We now state the complete SIA model in preparation for numerical solutions in section 5. Let $\mathbf{U} = (u, v)$ be the vector horizontal velocity; this is a 2D vector-valued function of 3D coordinates (x, y, z). The SIA shear stress approximation is

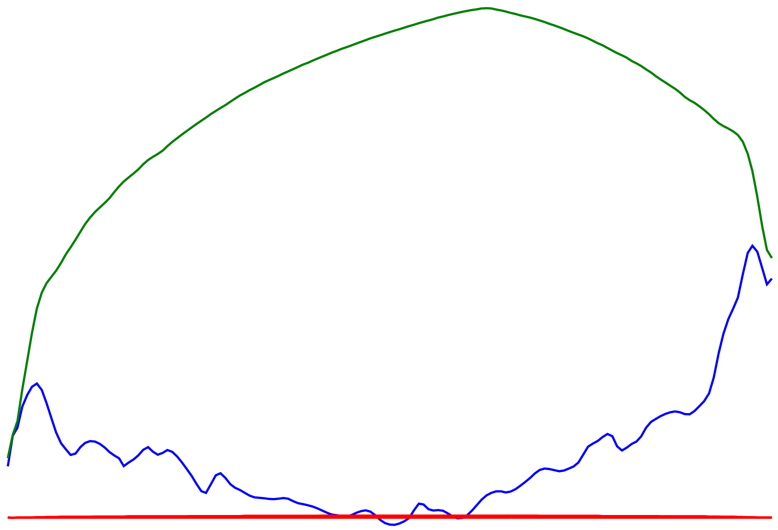


FIGURE 4. A cross-section of the Greenland ice sheet (71° N), with width about 760 km and maximum thickness 3200 m, is shown by the upper two curves. The thickened horizontal line (bottom) shows the same cross-section without vertical exaggeration.

$(\tau_{13}, \tau_{23}) \approx -\rho g(h-z)\nabla h$, and equation (5) gives a formula for shear strain rates

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

By integrating vertically we get, in the non-sliding case,

$$\mathbf{U} = -\frac{2A(\rho g)^n}{n+1}\left[H^{n+1} - (h-z)^{n+1}\right] |\nabla h|^{n-1}\nabla h. \tag{24}$$

Mass continuity in 2D, which generalizes the 1D version (21), says:

$$H_t = M - \nabla \cdot \left(\bar{\mathbf{U}} H\right) \tag{25}$$

Equation (25) may be written $H_t = M - \nabla \cdot \mathbf{q}$ in terms of the map-plane flux $\mathbf{q} = \int_b^h \mathbf{U}\,dz = \bar{\mathbf{U}}\,H$. Note $\bar{\mathbf{U}}$ and $\mathbf{q}$ are functions of 2D coordinates (x, y) only. Combining equations (24) and (25), we get an equation for the rate of thickness change in terms of mass balance M, thickness H, and surface gradient ∇h:

$$H_t = M + \nabla \cdot \left(\Gamma H^{n+2} |\nabla h|^{n-1} \nabla h\right). \tag{26}$$

We have defined the positive constant $\Gamma = 2A(\rho g)^n/(n+2)$ only to improve appearance.

Equation (26) is the SIA in 2D. Recalling our earlier promise, if we can solve (26) numerically then we have a usable model for certain ice sheets. Mary-Anne Mahaffy [45] used it in a 1976 numerical model of the Barnes ice cap in Canada, a particularly-simple ice sheet on a rather flat bed. This was the start of numerical ice sheet modeling.

Analogy with diffusion equations. Numerical methods for solving (26) are modifications of methods for the better-known heat equation, e.g. $T_t = DT_{xx}$ for the temperature of a conducting rod, its simplest 1D form. This model applies when material properties are constant and there are no heat sources. The positive constant D is the "diffusivity," with SI units which can be read from comparing sides of the equation:

$D \sim \mathrm{m}^2\mathrm{s}^{-1}$. This equation has a smoothing effect on the solution $T(t,x)$ as it evolves in time, because any local maximum or minimum in the temperature is flattened: $T_{xx} < 0$ implies $T_t < 0$, so T decreases near a maximum, while $T_{xx} > 0$ implies $T_t > 0$, so T increases near a minimum. All extrema of the temperature diffuse away.

The 2D heat equation, which describes the temperature $T(t,x,y)$ of a planar object, is a closer analog of (26). Recall that Fourier's law for conduction is the formula $\mathbf{Q} = -\kappa \nabla T$ for heat flux $\mathbf{Q}$, where κ is conductivity. We will assume, so as to help the analogy to the SIA, that $\kappa(x,y)$ may vary in space, and we suppose the heat source $f(t,x,y)$ is variable. Adding conservation of internal energy, we find the equation

$$\rho c T_t = f + \nabla \cdot (\kappa \nabla T). \tag{27}$$

Here ρ is density and c is specific heat capacity. Assuming ρc is constant, define the diffusivity $D = \kappa/(\rho c)$. Defining $F = f/(\rho c)$, a rescaled source term, (27) becomes

$$T_t = F + \nabla \cdot (D\, \nabla T). \tag{28}$$

This 2D heat equation is similar to the above 1D equation $T_t = DT_{xx}$, but now with variable diffusivity and an additional heat source.

Figure 5 shows a solution of (28) in which the initial condition is a localized "hot spot". Solutions of the heat equation always involve the spreading, in all directions, of any local concentration of heat; this is diffusion.

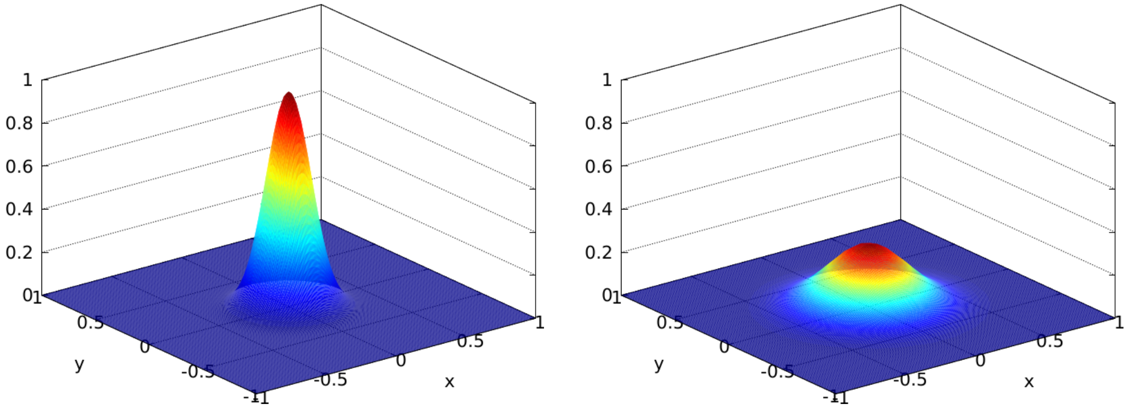


FIGURE 5. Left: Initial condition $T(0,x,y)$. Right: A solution $T(t,x,y)$ of (28), with $D = 1$ and $F = 0$, at $t = 0.02$.

The SIA equation (26) and the heat equation (28) are each diffusive, time-evolving partial differential equations (PDEs). A side-by-side comparison is appropriate:

SIA: H is ice thickness	heat: T is temperature
$H_t = M + \nabla \cdot (\Gamma H^{n+2} \lvert\nabla h\rvert^{n-1}\, \nabla h)$	$T_t = F + \nabla \cdot (D\, \nabla T)$

Notice that the number of derivatives (one time and two space) and the signs of the various terms are the same. Surface mass balance M is analogous to heat source F. The analogy suggests that we identify the *SIA diffusivity* as:

$$D = \Gamma H^{n+2} \lvert\nabla h\rvert^{n-1}. \tag{29}$$

Diffusion acts quickly where D is large. SIA diffusivity formula (29) gives a large value when the ice is thick and steep, because ice flows downhill most-strongly in that case. Our analogy also explains why the surfaces of ice sheets are smooth, at least once we overlook non-fluid processes like crevassing.

There are, however, conceptual and practical concerns about the (26)↔(28) analogy:

- The letters "H" and "h" generally denote different quantities in (26). Are we allowed to regard (26) as a diffusion anyway, even when the bed is not flat?
- The diffusivity D in (29) depends on the solution—both on thickness H and surface slope $|\nabla h|$—so equation (26) is a nonlinear PDE. Numerical solutions for nonlinear equations are, at least, more difficult to implement and analyze.
- The diffusivity D in (29) goes to zero at margins, where $H \to 0$, and at divides and domes, where $|\nabla h| \to 0$. Thus the solution (h or H) is not smooth at these locations, though it is continuous everywhere. The solution is not smooth even if the bed b and surface mass balance M *are* smooth. Large numerical errors will arise where the solution is not smooth.

Just as important is a physical deficiency of the SIA model, namely

- Ice flow is not very diffusive when significant longitudinal (membrane) stresses are present, as when ice is floating or sliding, or when the flow is significantly confined by terrain.

This concerns a deficiency of the SIA model, not just our analogy to explain it.

All four concerns are areas of active research in mathematical glaciology; see the Notes and References.

4. Finite difference numerics

Now we introduce numerical methods. The above diffusion analogy suggests that numerical schemes for the heat equation are a good starting point for solving the SIA. The analogy will lead us to a verified numerical scheme for (26). We demonstrate only finite difference (FD) schemes, which replace derivatives by difference quotients.

Taylor's theorem says that for a smooth function $f(x)$,

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

One may also replace "Δ" by its multiples, for example:

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

When constructing FD methods for differential equations the idea is to combine expressions like these to give approximations of the derivatives in the equation. The resulting algebraic equations, relating unknown gridded solution values via mere arithmetic, will approximate the differential equation.

We must apply Taylor expansions for each variable to approximate the partial derivatives appearing in the heat and SIA equations. For example, for $u = u(t,x)$ we have

$$\begin{aligned}
u_t(t,x) &= \frac{u(t+\Delta t,x) - u(t,x)}{\Delta t} + O(\Delta t), \\
u_t(t,x) &= \frac{u(t+\Delta t,x) - u(t-\Delta t,x)}{2\Delta t} + O((\Delta t)^2), \\
u_x(t,x) &= \frac{u(t,x+\Delta x) - u(t,x-\Delta x)}{2\Delta x} + O((\Delta x)^2), \\
u_{xx}(t,x) &= \frac{u(t,x+\Delta x) - 2\,u(t,x) + u(t,x-\Delta x)}{\Delta x^2} + O((\Delta x)^2)
\end{aligned}$$

Note that if Δ is a small number then $O(\Delta^2)$ is smaller than $O(\Delta)$; the first approximation is inferior to the latter three in this sense.

Explicit scheme for the heat equation. Our simplest scheme approximates the 1D heat equation $T_t = DT_{xx}$ *explicitly*. It is based on the fact that because T_t and DT_{xx} are equal, these two FD expressions are nearly equal:

$$\frac{T(t+\Delta t,x) - T(t,x)}{\Delta t} \approx D\,\frac{T(t,x+\Delta x) - 2\,T(t,x) + T(t,x-\Delta x)}{\Delta x^2}. \tag{30}$$

While (30) only approximates the PDE, forcing it to be an equality allows us to determine grid values. Let (t_n, x_j) denote the points of the time-space grid shown in Figure 6. It has spacing $\Delta x > 0$ and $\Delta t > 0$ in the two dimensions. Denote our approximation of the solution value $T(t_n, x_j)$ by T_j^n. The finite difference scheme is

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = D\,\frac{T_{j+1}^n - 2\,T_j^n + T_{j-1}^n}{\Delta x^2}.$$

To get a simplified formula, let $\mu = D\Delta t/(\Delta x)^2$ and solve for T_j^{n+1}:

$$T_j^{n+1} = \mu T_{j+1}^n + (1-2\mu)T_j^n + \mu T_{j-1}^n \tag{31}$$

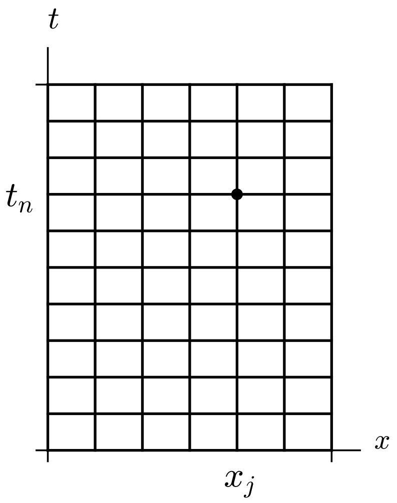


FIGURE 6. A grid for a finite difference solution to the 1D heat equation.

FD scheme (31) is *explicit* because it directly (explicitly) computes T_j^{n+1} in terms of values at time t_n. Figure 7 (left) shows the “stencil” for scheme (31): three values at the current time t_n are combined to update one value at the next time t_{n+1}.

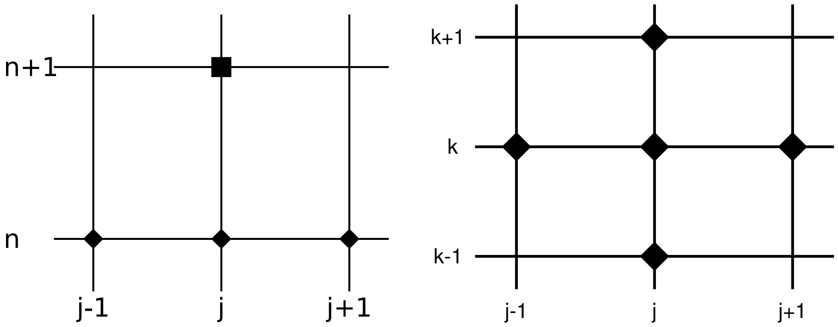


FIGURE 7. Left: Space-time stencil for the explicit scheme (31) for the 1D heat equation. Right: Spatial-only stencil for scheme (33).

The expression $T(t_n, x_j)$ is the value of a heat equation solution at a grid point. This is generally a different number from T_j^n, the value actually computed by a scheme like (31). Because the FD expressions become closer to the derivatives they approximate, we intend that the numbers $T(t_n, x_j)$ and T_j^n become closer together as the grid is made finer ($\Delta t \to 0$ and $\Delta x \to 0$). That is, we intend our FD scheme to *converge* under *grid refinement*. Proving that such convergence happens is a main topic in numerical textbooks [40, 48]. Even if we expect convergence to happen, this needs checking for the actual computer program which is our implementation (*verification*; Section 6).

How accurate is scheme (31)? Our construction tells us that the difference between the scheme (31) and the PDE it solves is $O(\Delta t + (\Delta x)^2)$. This difference goes to zero as we refine the grid in space and time, a scheme property called *consistency*. With care about the smoothness of boundary conditions, and using mathematical facts about the heat equation itself, one can also show that the difference between T_j^n and $T(t_n, x_j)$ is $O(\Delta t + (\Delta x)^2)$, the *convergence rate*. However, to get convergence the scheme must be *stable*, which we address below. The main theorem for numerical PDE schemes asserts that “consistency plus stability implies convergence”. See the Notes and References.

Instead of pursuing theory, in these notes we emphasize something practical, namely verification using exact solutions. We will find PDE problems for which we already know an exact solution $T(t, x)$, and then compute differences like $|T_j^n - T(t_n, x_j)|$ and confirm that they go to zero under grid refinement. This demonstrates directly that our actual implementation converges, not just that it should do so in theory. A verified code can be confidently applied to real situations.

A first implemented scheme. Our first Matlab implementation is for the 2D heat equation (28) with D constant and $F = 0$:

$$T_t = D(T_{xx} + T_{yy}). \tag{32}$$

We solve this equation on the square $-1 < x < 1$, $-1 < y < 1$. For a simple example problem we set $T = 0$ on the boundary of the square and choose a gaussian initial condition: $T(0, x, y) = \exp(-30(x^2 + y^2))$.

Writing $T^n_{jk} \approx T(t_n, x_j, y_k)$, the 2D explicit scheme is

$$\frac{T^{n+1}_{jk} - T^n_{jk}}{\Delta t} = D\left(\frac{T^n_{j+1,k} - 2\,T^n_{jk} + T^n_{j-1,k}}{\Delta x^2} + \frac{T^n_{j,k+1} - 2\,T^n_{jk} + T^n_{j,k-1}}{\Delta y^2}\right). \tag{33}$$

The stencil for the right side of (33) is in Figure 7. Using Matlab colon notation to implement loops over the spatial variables, our scheme becomes `heat.m` below.

Consider the example run:

```
>>  heat(1.0,30,30,0.001,20);
```

This sets $D = 1.0$, uses a 30×30 spatial grid, sets $\Delta t = 0.001$, and takes $N = 20$ time steps. The result was already shown in Figure 5 (right); this is the look of success.

heat.m

```
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
```

However, very similar runs seem to succeed or fail according to some as-yet unclear circumstances. For example, results from these calls are shown in Figure 8:

```
>> heat(1.0,40,40,0.0005,100);     % Figure 8, left
>> heat(1.0,40,40,0.001,50);       % Figure 8, right
```

Both runs compute temperature T on the same spatial grid, for the same final time $t_f = N\Delta t = 0.05$, but with different time steps. Noting the vertical axes, the second run clearly shows "instability," an extreme form of inaccuracy characterized in practice by exponentially-growing solution magnitude. (In fact the exact solution is bounded.)

Stability criteria and adaptive time stepping. To avoid such instabilities we need to understand the scheme better. While we have not made an implementation error, we must learn how to choose compatible space and time steps.

Recall the 1D explicit scheme in form (31): $T^{n+1}_j = \mu T^n_{j+1} + (1 - 2\mu)T^n_j + \mu T^n_{j-1}$. The new value T^{n+1}_j is an average of the old values, in the sense that the coefficients sum to one. Averaging is stable because averaged wiggles are smaller than the wiggles themselves.

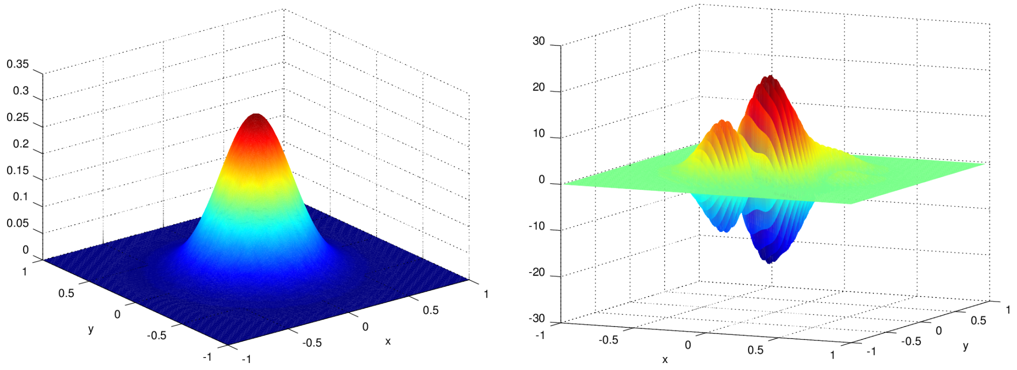


Figure 8. Numerically-computed temperature on 40 × 40 grids. The two runs are the same except that the left has $\Delta t = 0.0005$ and $D\Delta t/(\Delta x)^2 = 0.2$, while the right has $\Delta t = 0.001$ and $D\Delta t/(\Delta x)^2 = 0.4$. Compare (34).

However, the scheme is only an average *if* the coefficients, specifically the middle coefficient $1-2\mu$, are positive. A linear combination with coefficients which add to one is not an average if some coefficients are negative. For example, one would not accept 15 as an "average" of 5 and 7, but: $15 = -4 \times 5 + 5 \times 7$, and $-4 + 5 = 1$.

What consequences follow from requiring the middle coefficient in (31) to be positive? We find a *stability criterion*, with these equivalent forms:

$$1 - 2\mu \geq 0 \quad \iff \quad \frac{D\Delta t}{\Delta x^2} \leq \frac{1}{2} \quad \iff \quad \Delta t \leq \frac{\Delta x^2}{2D}. \tag{34}$$

The third form states the criterion as a limitation on the size of Δt. It is a *sufficient* stability condition; it guarantees stability though something weaker might do.

In summary, for a given value of Δx, shortening the time steps Δt until (34) holds will make FD scheme (31) for the 1D heat equation into a stable averaging process. Applying this same idea to the 2D heat equation (32) leads to the stability condition $1-2\mu^x-2\mu^y \geq 0$, where $\mu^x = D\Delta t/(\Delta x^2)$ and $\mu^y = D\Delta t/(\Delta y^2)$. In the cases shown in Figure 8, wherein $\Delta x = \Delta y$, this condition requires $D\Delta t/(\Delta x^2) \leq 0.25$. This inequality precisely distinguishes between these stable and unstable results, respectively.

Runs of `heat.m` are unstable if the time step Δt is too large relative to the spacing Δx. However, the stability criterion (34) can be enforced inside the code, and this is called an *adaptive* implementation. Changing `heat.m` in this way yields `heatadapt.m` (not shown). Furthermore, such adaptive codes can be stable even if the diffusivity D varies in space or time. On the other hand, if the diffusivity D is large, or if grid spacings Δx, Δy are small, then adaptive explicit implementations must take many short time steps to assure stability. Adaptive explicit schemes become inefficient on spatially-refined grids.

Implicit schemes. *Implicitness* is an alternative stability fix to adaptivity. For example, the finite difference scheme

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = D\,\frac{T_{j+1}^{n+1} - 2\,T_j^{n+1} + T_{j-1}^{n+1}}{\Delta x^2} \tag{35}$$

is an $O(\Delta t+(\Delta x)^2)$ implicit scheme for the 1D heat equation $T_t = DT_{xx}$. This implicit *backward Euler* scheme is stable for any positive time step $\Delta t > 0$; it is *unconditionally stable* [40, 48]. Another well-known implicit scheme for the heat equation is *Crank-Nicolson*, an unconditionally-stable scheme with smaller error $O((\Delta t)^2 + (\Delta x)^2)$.

Implicit schemes are more difficult to implement because the unknown solution at time step t_{n+1} must be treated as a vector, and a (large) linear or nonlinear system of equations must be formed and solved at each time step. Though the SIA is a highly-nonlinear diffusion equation, implicit schemes are indeed implementable when appropriate solver tools are used [11]. However, for these notes, in the tradeoff between the easy implementability of adaptive explicit schemes and the improved stability of implicit schemes, we choose the former.

Numerical solution of generalized diffusion equations. We are trying to numerically model flowing ice, not temperature as seen so far. In this section, as a step toward an explicit and adaptive SIA code which works on nonflat beds, `diffstag.m` below is an implemented numerical scheme for a generalized diffusion equation:

$$T_t = F + \nabla \cdot (D\nabla(T + b)) \,. \tag{36}$$

Equation (36) has an extra "tilt" b inside the gradient. The source term $F(x, y)$, diffusivity $D(x, y)$, and tilt $b(x, y)$ may all vary in space.

diffstag.m

```
function [T,dtav] = diffstag(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;
```

This function is called for SIA-specific solution schemes, which we build next, using $b(x, y)$ equal to the bedrock elevation. Its explicit time-stepping method is conditionally-stable, with the same time step restriction as for the constant diffusivity case. For stability to apply under the same criterion it is important that we evaluate $D(x, y)$ at *staggered* grid points. That is, where $X = T + b$, we will be using the approximation

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

Figure 9 (left) shows the stencil. The user supplies the diffusivity $D(x, y)$ on the staggered grid (triangles). The initial values $T(0, x, y)$, source term $F(x, y)$, and tilt $b(x, y)$ are supplied on the regular grid (diamonds). When using this code for the flat-bed case of the SIA we will take $b = 0$.

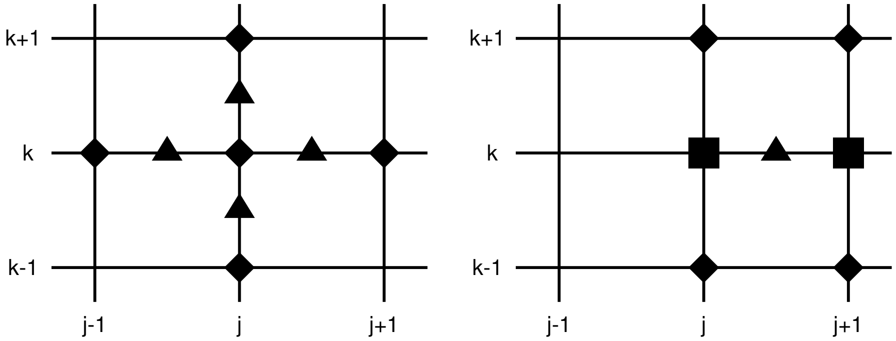


FIGURE 9. Left: Spatial stencil for staggered-grid (triangles) evaluation of diffusivity D in (36). Right: Stencil showing how D is evaluated in the SIA, from surface elevation (diamonds) and thicknesses (squares).

5. NUMERICALLY SOLVING THE SIA

As already noted, we compute the SIA diffusivity D on a staggered grid. Various schemes for staggered-grid diffusivity have been proposed, but all involve averaging H and differencing h in a "balanced" way onto the staggered grid. In the code `siaflat.m` below, which only works for a flat bed and zero surface mass balance, we use the Mahaffy [45] method. The Mahaffy stencil for computing D is shown in Figure 9 (right). Note that this SIA solver calls `diffstag.m` (above).

For non-flat beds and actual climates, namely when the surface mass balance term M can take either sign, `siaflat.m` does not suffice. Its deficiences are corrected in an extended code `siageneral.m` (not shown), which in fact only alters a few lines.

siaflat.m

```
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;

N = ceil(tf / deltat);   deltat = tf / N;
dx = 2 * Lx / J;   dy = 2 * Ly / K;   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 = [];
H = H0;
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] = diffstag(Lx,Ly,J,K,Dup,Ddn,Drt,Dlt,H,deltat);
  t = t + deltat;
  dtlist = [dtlist dtadapt];
end
```

Addressing ablation ($M < 0$) requires a small modification which deserves exposure. Note that the grounded ice margin occurs in areas of ablation where the SIA thickness solution H goes to zero. However, the thickness of ice must obviously be greater than or equal to zero everywhere: $H(t, x) \geq 0$. It follows that geometry-evolving numerical glacier and ice sheet models are solving a *free-boundary problem*. For an explicit time-stepping scheme it suffices to make the following rule:

> *If the numerical computation generates an updated thickness which is negative then it is reset to zero:* $H_{j,k}^{n+1} \leftarrow \max\{0, H_{j,k}^{n+1}\}$.

Determining glacier-covered area is a major glaciological purpose for modeling, thus this seemingly-small complication is ubiquitous and unavoidable in explicit time-stepping glacier models if the margin is allowed to move. Modelers often implement and then overlook the above rule; it violates conservation of mass [13].

6. Exact solutions and verification

Program `siaflat.m` calls `diffstag.m`, so we already have a complicated code. How can we make sure that such an implemented numerical scheme is correct? Here are three proposed techniques in the modeling community:

(i) don't make any mistakes, or
(ii) compare your numerical results with results from other researchers, and hope the outliers are in error, or
(iii) compare your numerical results to an exact solution.

The first two approaches, *infallibility* and *intercomparison* respectively, should be less common than they are. The last one, preferred by the CFD community generally [57, 71], is *verification*. It is a simple idea: A computer code should be tested in cases where we know the right answer. To do this we must return to the PDE itself. We must seek-out useful exact solutions.

Exact solutions of heat equation. Again consider the 1D heat equation $T_t = DT_{xx}$ with constant D. Many exact solutions are known, but we will use the time-dependent *Green's function*, circa 1830, also known as the *heat kernel*. It starts at time $t = 0$ with a delta function $T(0, x) = \delta(x)$ of heat at the origin. Then it spreads out over time, solving the equation on $-\infty < x < \infty$ and for $t > 0$. We calculate it by a method which generalizes to the SIA.

This Green's function is self-similar over time, in the sense that it changes shape *only* by shrinking the output (vertical) axis and lengthening the input (horizontal) axis (Figure 10). These scalings are related to each other by conservation of energy. Similarity variables for this solution involve multiplying the input and output of an invariant shape $\phi(s)$ by the same power of t:

$$s \overset{\textit{input scaling}}{=} t^{-1/2}x, \qquad T(t,x) \overset{\textit{output scaling}}{=} t^{-1/2}\phi(s). \tag{37}$$

One may use $T_t = DT_{xx}$ to show that $\phi(s) = (4\pi D)^{-1/2}\, e^{-s^2/(4D)}$, and thus

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

To verify a numerical solver for the 1D heat equation using this exact solution we may evaluate this exact solution at $t_0 > 0$ to provide initial values. At a later time $t_1 > t_0$ the numerical solution should be close to the exact solution $T(t_1, x)$ (see the Exercises).

Halfar's similarity solution to the SIA. Now we jump to the year 1981 in which Peter Halfar [29] published the corresponding similarity solution of the SIA, at least for flat bed and zero surface mass balance. For the 2D SIA model (26) using a Glen exponent $n = 3$ this solution [30] has scalings similar to (37) above:

$$s = t^{-1/18}r, \qquad H(t,r) = t^{-1/9}\phi(s), \tag{38}$$

where $r = (x^2 + y^2)^{1/2}$ is the distance from the origin. These scalings are related to each other by the constancy of total mass/volume, because no mass is gained or lost through the surface (see the Exercises). Scalings (38) imply that, differently from

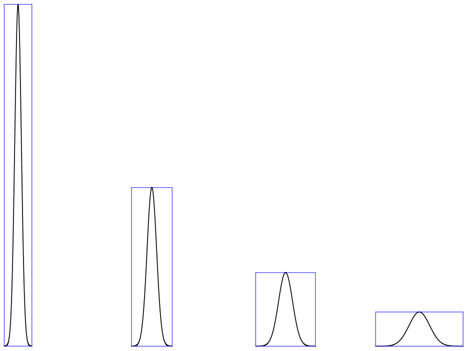


FIGURE 10. The Green's function of the heat equation has the same shape at each time, but with time-dependent input- and output-scalings.

heat, the diffusion of ice slows down dramatically as the shape flattens out. That is, the powers $t^{-1/9}$ and $t^{-1/18}$ change very slowly for large times t.

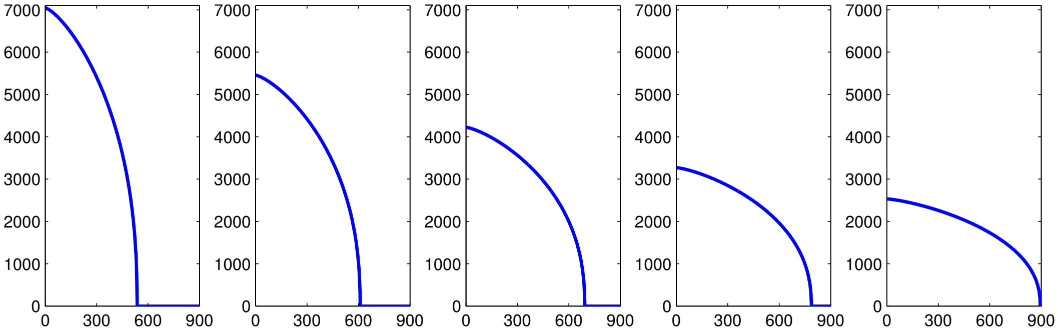


FIGURE 11. Radial sections of Halfar's solution (39) of (26), shown on H (m) versus r (km) axes at $t = 1, 10, 100, 1000, 10000$ years.

The Halfar solution formula is remarkably simple given all that it accomplishes:

$$H(t,r) = H_0 \left(\frac{t_0}{t}\right)^{1/9} \left[1 - \left(\left(\frac{t_0}{t}\right)^{1/18} \frac{r}{R_0}\right)^{4/3}\right]^{3/7}. \qquad (39)$$

Here the characteristic time $t_0 = (18\Gamma)^{-1}(7/4)^3 R_0^4 H_0^{-7}$ depends on the radius R_0 and the center height H_0. In Figure 11 we see that for times significantly greater than t_0 $(t/t_0 \gg 1)$ the solution changes very slowly. For example, the change between years 1 and 100 is larger than that between years 1000 and 10000.

Using Halfar's solution. Formula (39) is simple to use for verifying time-dependent SIA models. The code `verifysia.m` (not shown) takes as input the number of grid points in each (x, y) direction. It uses the Halfar solution at 200 a as the initial condition, does a numerical run of `siaflat.m` to a final time 20000 a, and then compares to the Halfar formula at that time. It computes the thickness *numerical error*, the absolute difference between the numerical and exact thicknesses at the final time:

```
>> verifysia(20)
average thickness error  = 22.310
```

```
>> verifysia(40)
average thickness error  = 9.459
>> verifysia(80)
average thickness error  = 2.771
>> verifysia(160)
average thickness error  = 1.085
```

The average numerical thickness error decreases with increasing grid resolution, as expected for a correctly-implemented code. What is less obvious is that almost any numerical implementation mistake will break such convergence. That is, the reported errors will not shrink if there is any meaningful bug (see the Exercises). The fragility of verification is its essential benefit! One must implement the model correctly, not just generate good-looking results, to get convergence as above.

The Halfar solution has also been used for modelling real ice masses. Nye and others [49] used it to compare the long-time consequences of different flow laws for the south polar cap on Mars. In particular, they evaluated CO_2 and H_2O ice softness parameters by comparing the long-time behavior of the corresponding Halfar solutions to the observed polar cap properties. Their 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 [but rather] water ice, . . .

This theoretical result has since been confirmed by sampling the surface of Mars.

Are exact solutions like Halfar's always available when needed? No, of course not. However, many ice flow models have exact solutions which are relevant to verification; see the Notes and References. For example, we will use an exact solution for ice shelves in a later section. An apparent absence of exact solutions to a given model may actually show that not enough thought has gone into the continuum model itself.

A test of robustness. Verification is an ideal way to test a code, but another worthwhile test is for "robustness" in unusual or difficult input cases. One asks: Does the model break? Here one may not have a precise expectation of what the code *should* do, but it should not produce obviously-unreasonable outputs.

The robustness test run by `roughice.m` (not shown) demonstrates that `siaflat.m` can handle an ice sheet with extraordinarily-large driving stresses. Recall that the glaciological driving stress $\tau_d = -\rho g H \nabla h$ appears in the SIA model as the shear stress (τ_{13}, τ_{23}) at the base of the ice. The driving stress is largest where the ice is thick and the surface slope $|\nabla h|$ is steep. In our test we give `siaflat.m` a randomly-generated initial ice sheet which has huge driving stresses. It is thick—average of 3000 m—with huge surface slopes because of random surface elevation values. An initial shape is shown in Figure 12 (left). During a run of just 50 model years on a 17 km grid, with time step determined adaptively from (34), the maximum diffusivity D decreases as the surface both lowers and becomes smooth through flow. The time-step consequently increases, in this case by three orders of magnitude from about 0.0002 years to 0.2 years. The maximum value of the driving stress decreases from 57 bar ($= 5.7 \times 10^6$ Pa) to 3.6 bar. At the end of the run the ice cap has the shape shown in Figure 12 (right).

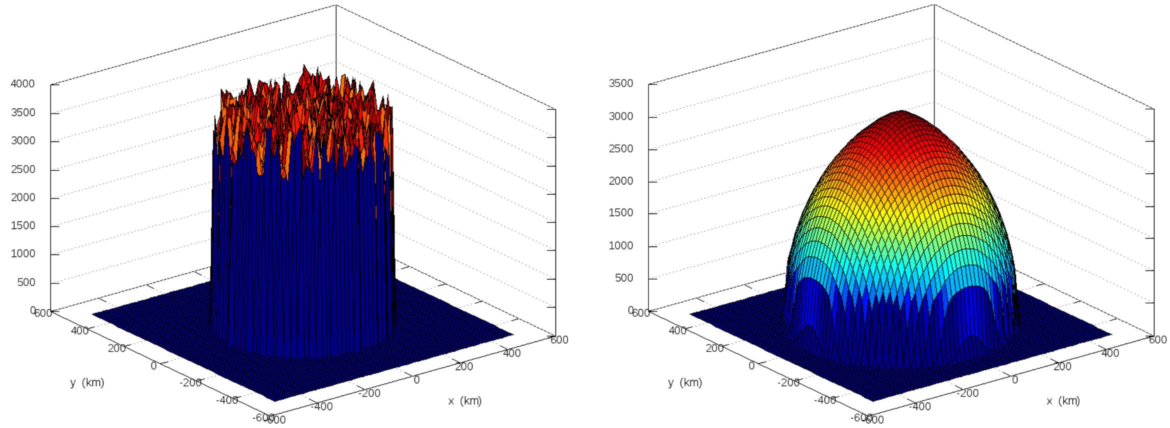


FIGURE 12. The initial ice sheet at left, with huge driving stresses, evolves to the ice cap at right in only 50 model years.

The final shape in Figure 12 is close to a Halfar solution, illustrating what he proved [29, 30], namely that *all* solutions of the zero-mass-balance SIA on a flat bed asymptotically approach his solution. The Halfar solution is attracting and thus worth knowing.

7. APPLICATION TO THE ANTARCTIC ICE SHEET

Let us apply the SIA model to the Antarctic ice sheet. To do this we must first modify `siaflat.m` to allow non-flat bedrock elevation $b(x, y)$ and arbitrary, though here time-independent, climatic mass balance $M(x, y)$. Also we enforce non-negative thickness at each timestep, and add a minimal model of interaction with the ocean. Specifically, we calve-off any ice that satisfies the flotation criterion (see Section 9). The result is `siageneral.m` (not shown), a code only ten lines longer than `siaflat.m`.

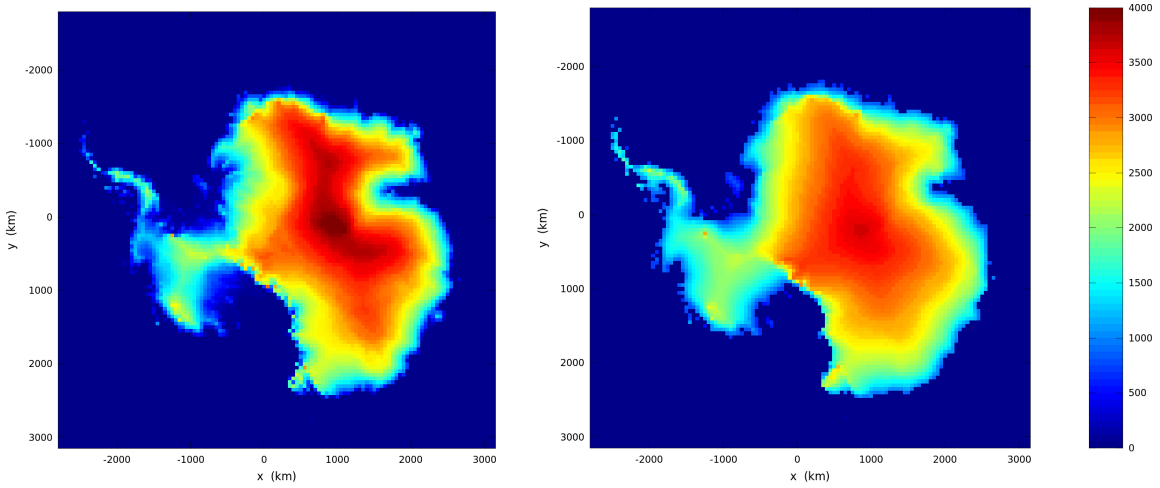


FIGURE 13. Left: Initial surface elevation (m) of Antarctic ice sheet. Right: Final surface elevation at end of 40 ka model run on 50 km grid.

We use measured accumulation, bedrock elevation, and surface elevation from the ALBMAPv1 data set [38]. Melt is not modelled so the climatic mass balance is equal to the precipitation rate. (This is a reasonable approximation for Antarctica, but not for Greenland.) These input data are read from a NetCDF file and preprocessed by an additional code `buildant.m` (not shown).

The code `ant.m` (not shown) calls `siageneral.m` to do the simulation in blocks of 500 model years. The volume is computed at the end of each block. Figure 13 shows the initial and final surface elevations from a run of 40,000 model years on a $\Delta x = \Delta y = 50$ km grid. (The runtime on a typical laptop is a few minutes.) Areas of the actual Antarctic ice sheet with low-slope and fast-flowing ice experience thickening in the model, while near-divide ice in East Antarctica, in particular, thins.

Assuming the present-day Antarctic ice sheet is somewhere near steady state, these thickness differences reflect model inadequacies. The lack of a sliding mechanism explains the thickening in low-slope areas. The lack of thermomechanical coupling, or equivalently the constancy of ice softness in the model, and the fact that our isothermal A value is quite "soft", explains the thinning near the divide. We should be modelling floating ice too, but the SIA is completely inappropriate to that purpose; compare Section 9. See the Notes and References for widely-used thermomechanical and membrane-stress modelling techniques which address these inadequacies.

Figure 14 compares the ice volume time series for 50 km, 25 km, and 20 km grids. This result, namely grid dependence of the ice volume, is typical. One cause is that steep gradients near the ice margin are poorly resolved, but to differing degrees, at such coarse resolutions. In any case, Figure 14 is a warning about the interpretation of model runs. Even if the data is available only on a fixed grid, the model should be run at different grid spacings to evaluate the dependence of model results on resolution.

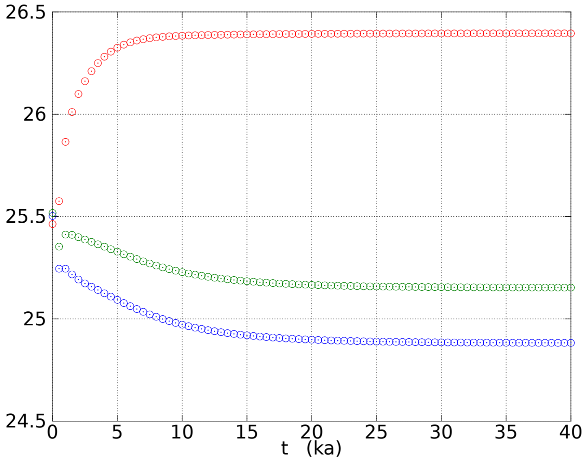


FIGURE 14. Ice volume of the modeled Antarctic ice sheet, in units of 10^6 km^3, from 50 km (top), 25 km (middle), and 20 km (bottom) grids.

8. Mass continuity and kinematical equations

In the SIA the stress balance becomes formula (24) for the velocity. Combined with the mass continuity equation (25), this gives model (26) for evolution of the ice sheet thickness. Equation (26) therefore combines two concepts which we now want to think about separately, and in greater generality.

The most basic shallow assumption made by most ice flow models is that the surface and base of the ice are differentiable functions $z = h(t,x,y)$ and $z = b(t,x,y)$. That is, though the Stokes theory allows the fluid boundary to be any closed surface in three-dimensional space, ice sheet and glacier models take a map-plane perspective and have a well-defined ice thickness: $H = h - b$. For example, surface overhang is rarely allowed in models.

Kinematical equations apply on the upper and lower surfaces of the ice sheet. Let a be the climatic mass balance function ($a > 0$ is accumulation) and s be the basal melt rate function ($s > 0$ is basal melting). We need these fields in thickness-per-time units; convert from mass flux units by assuming an ice density. The net map-plane mass balance $M = a - s$, which appears in the mass continuity equation (25), is the difference of these surface fluxes. The *(upper) surface kinematical equation* is

$$h_t = a - \mathbf{U}\big|_h \cdot \nabla h + w\big|_h, \tag{40}$$

and the *base kinematical equation* is

$$b_t = s - \mathbf{U}\big|_b \cdot \nabla b + w\big|_b. \tag{41}$$

In these equations $\mathbf{U}$ is the horizontal ice velocity and w the vertical ice velocity. Equations (40) and (41) describe the movement of the ice surfaces caused by the velocity of the ice and the mass balance functions.

An important mathematical fact follows from the assumption of well-defined surface elevations and incompressibility. Namely, that any *pair* of these three equations implies the third:

- the surface kinematical equation (40),
- the base kinematical equation (41), and
- the map-plane mass continuity equation (25).

One proves the any-two-implies-the-other implications by using equation (1) and the Leibniz rule for differentiating integrals (see the Exercises). Thus the surface kinematical and mass continuity equations are closely-related.

Prognostic models. Consider an explicit, geometry-evolving, isothermal ice sheet model, called a *prognostic* model in glaciology. Each time step follows this recipe:

- numerically solve an incompressible stress balance, yielding velocity $\mathbf{u} = (u,v,w)$,
  - ○ the ice geometry and the stress boundary conditions determine $\mathbf{u}$
  - ○ if the stress balance yields $\mathbf{U} = (u,v)$ then w comes from equation (1)
- decide on a time step Δt for (25) based on velocities and/or diffusivities,
- from the horizontal velocity $\mathbf{U} = (u,v)$, compute the flux $\mathbf{q} = \int_b^h \mathbf{U}\,dz = \bar{\mathbf{U}}H$
- update mass balance $M = a - s$ and do a time-step of (25) to get H_t
- update the upper surface elevation and thickness, and repeat.

The last step uses the rules $H \mapsto H + H_t\Delta t$ and $h \mapsto h + H_t\Delta t$, but it also enforces $H \geq 0$ and $h \geq b$ as mentioned above. Most ice sheet models use the mass continuity equation (25) to describe changes in ice sheet geometry, but (40) may be used instead.

The above "standard explicit ice sheet model" has many variations. Some glaciological questions are answered just by solving the stress balance for the velocity, so the geometry does not evolve. Sometimes the goal is the steady-state configuration of the glacier, both geometry and velocity, in which case the constraint $H \geq 0$ should be regarded as a part of the continuum formulation and the solver [11, 35].

Other processes are often simulated at each time step, such as the conservation of energy within the ice, or subglacial and supraglacial processes. Understanding the diverse time scales associated to these processes is always an important step in designing a model which extends our "introductory" isothermal SIA model.

It may seem that when using SIA equation (26) one bypasses the computation of the ice velocity. That is because the mass continuity equation can be written as a diffusion, with $\mathbf{q} = -D\nabla h$ for the flux. Fast flow in ice sheets is associated with sliding and floating ice, however, and for these flows the ice geometry evolution is not a diffusion. A complete ice flow model will be neither diffusive nor wave-like (hyperbolic), but the general formula $\mathbf{q} = \bar{\mathbf{U}} H$ will apply everywhere. The ice velocity is always related in some way to the local surface gradient, but usually one must solve membrane-stress-balancing equations. Solving the stress balance for the velocity field is both a generally-nontrivial and obligatory step in a numerical ice flow model. The next section explores one important case.

9. Shelves and streams

As its name suggests, the *shallow shelf approximation* (SSA) stress balance applies to floating *ice shelves*. It also applies reasonably well to grounded *ice streams* which have not-too-steep bed topography and low basal resistance (Figure 15). But what is, and is not, an ice stream? Ice streams slide at 50 to 1000 $\mathrm{m\,a^{-1}}$, they have a concentration of vertical shear in a thin layer near base, and typically they flow into ice shelves. Pressurized liquid water at their beds plays a critical role enabling their fast flow. However, there are other fast-flowing grounded parts of ice sheets, generally called *outlet glaciers*. They can have even faster surface speed (up to 10 $\mathrm{km\,a^{-1}}$), but an unknown fraction of this speed is from sliding at the base. In an outlet glacier there is substantial vertical shear within the ice column, sometimes caused by a thick layer of soft temperate ice above the bed. Outlet glaciers are typically controlled by fjord-like, large-slope bedrock topography. Few simplifying assumptions are appropriate for outlet glaciers, and the SSA is often not a sufficient model.

The shallow shelf approximation (SSA). We state the SSA stress balance equation only in the plane flow, isothermal case:

$$\left(2BH|u_x|^{1/n-1}u_x\right)_x - C|u|^{m-1}u = \rho g H h_x \tag{42}$$

The term in parentheses is the vertically-integrated longitudinal stress, also called the *membrane stress* in two horizontal dimensions. The second term $\tau_b = -C|u|^{m-1}u$ is the basal resistance, which is zero ($C = 0$) in an ice shelf. The term on the right is the

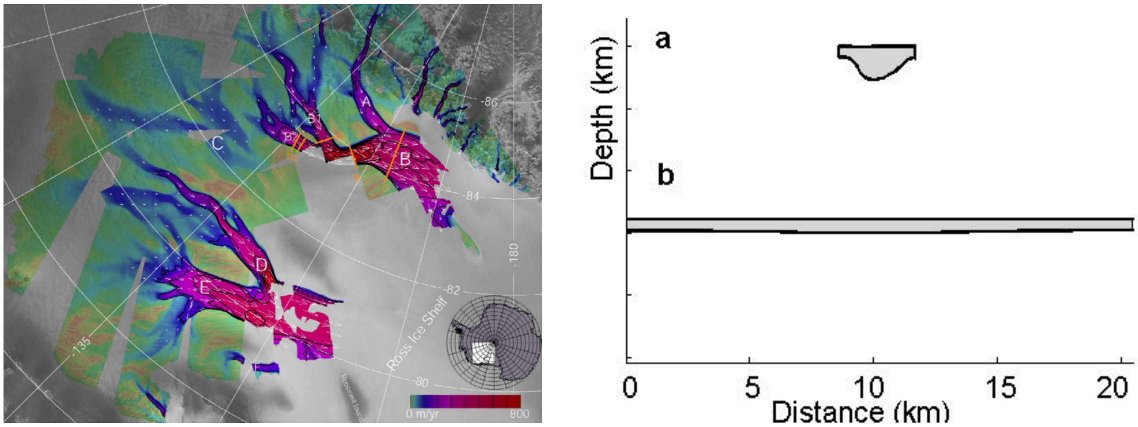


FIGURE 15. Left: The SSA model applies to these Siple Coast ice streams [34]; color is radar-derived surface speed. Right: Cross sections, with no vertical exaggeration, of (**a**) Jakobshavns Isbrae outlet glacier, Greenland and (**b**) Whillans Ice Stream, Siple Coast, Antarctica [66].

negative of the driving stress ($\tau_d = -\rho g H h_x$). Thus the longitudinal strain rates are determined by a balance involving the integrated ice hardness (coefficient BH), the slipperyness of the bed (coefficient C and the power m) and the geometry of the ice (thickness H and surface slope h_x).

In (42) the velocity u is independent of z. We assume that the ice hardness $B = A^{-1/n}$ is also independent of depth. (Non-isothermal models compute the vertical average of the temperature-dependent hardness.) The coefficient $\bar{\nu} = B|u_x|^{1/n-1}$ is the *effective viscosity*, which depends on u; (42) can be written

$$(2\,\bar{\nu}\,Hu_x)_x - C|u|^{m-1}u = \rho g H h_x. \tag{43}$$

The inequality

$$\rho H < -\rho_w b \tag{44}$$

is called the *flotation criterion*. In our simple mechanical model, a *grounding line* occurs where $\rho H = -\rho_w b$. For grounded ice ($\rho H \geq -\rho_w b$) the driving stress uses $h = H + b$. For floating ice one uses $h = (1 - \rho/\rho_w)H$ in the driving stress so that equation (43) simplifies to

$$(2\,\bar{\nu}\,Hu_x)_x = \rho g(1 - \rho/\rho_w)HH_x. \tag{45}$$

Steady ice shelf exact solution. For a steady 1D ice shelf, in which $H_t = 0$, van der Veen [68] built an exact solution by noting that both sides in equation (45) are derivatives, thus the equation is integrable. In fact, because the mass continuity equation (25) also reduces to $M = (uH)_x$ in the steady case, velocity and thickness simultaneously solving (25) and (45) can be found when M is constant ($M = M_0$). This exact solution is parameterized by M_0 and the ice thickness H_g and velocity u_g at the grounding line (i.e. the inflow).

Supposing $H_g = 500$ m, $u_g = 50\,\mathrm{m\,a^{-1}}$, and $M_0 = 30\,\mathrm{cm\,a^{-1}}$ we get the exact results shown in Figure 16. We will use this exact solution to verify a numerical SSA code.

Because driving stresses are highest near the grounding line, the highest longitudinal strain rates and thinning rates also occur there.

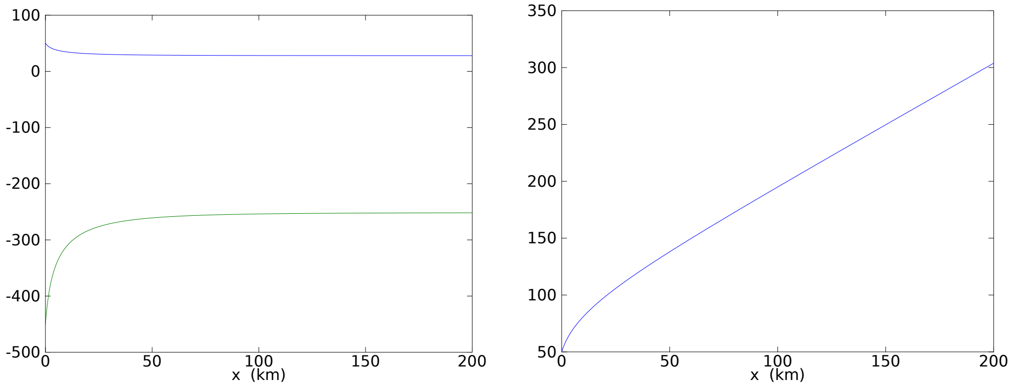


FIGURE 16. The upper and lower surface elevation (m; left) and velocity (m/a; right) of the exact ice shelf solution; $x = 0$ is the grounding line.

Numerical solution of the SSA. Suppose the ice thickness is a known function $H(x)$. To find the velocity we must solve the nonlinear PDE (43) or (45) for $u(x)$. When we do this numerically an iteration is needed because of the nonlinearity with respect to u. The simplest iterative approach is to use an initial guess at the velocity, then compute an effective viscosity, then get a new velocity solution from a linear PDE problem, and repeat until the change is as small as desired. We will demonstrate such a *fixed point* or *Picard iteration*. *Newton iteration*, which would converge faster, is more complicated to implement and may require a better initial guess to converge at all.

Denote the previous velocity iterate as $u^{(k-1)}$ and the current iterate as $u^{(k)}$. Compute $\bar{\nu}^{(k-1)} = B|u_x^{(k-1)}|^{1/n-1}$ and define $W^{(k-1)} = 2\bar{\nu}^{(k-1)}H$. Solving the following linear elliptic PDE for the unknown $u^{(k)}$ is a Picard iteration for (43):

$$\left(W^{(k-1)}u_x^{(k)}\right)_x - C|u^{(k-1)}|^{m-1}u^{(k)} = \rho g H h_x. \tag{46}$$

If the difference between $u^{(k-1)}$ and $u^{(k)}$ were zero then we would have a solution of (43). In practice we stop the iteration when the difference is smaller than some tolerance.

Equation (46) is a linear boundary value problem, written abstractly as

$$\left(W(x)\,u_x\right)_x - \alpha(x)\,u = \beta(x) \tag{47}$$

where the functions $W(x)$, $\alpha(x)$, $\beta(x)$ are known. Equation (47) applies for x in an interval $[x_g, x_c]$ where x_g, x_c are the locations of the grounding line and calving front, respectively. At x_g we have $u(x_g) = u_g$. In the ice shelf case we also have the calving front condition

$$2BH|u_x|^{1/n-1}u_x = \frac{1}{2}\rho(1-\rho/\rho_w)gH^2 \tag{48}$$

at $x = x_c$; see Notes and References. Notice that (48) can be solved for $u_x(x_c) = \gamma$ in terms of the thickness $H(x_c)$ at the calving front.

Where do we get an initial guess $u^{(0)}$? Generally this may require effort, but the choice is straightforward in the 1D case. For grounded ice, we may assume ice is held by basal resistance only: $u^{(0)}(x) = \left(-C^{-1}\rho g H h_x\right)^{1/m}$. For floating ice, an initial velocity iterate comes from assuming a uniform strain rate provided by the boundary conditions: $u^{(0)}(x) = u_g + \gamma(x - x_g)$.

Suppose equation (47) applies on $[x_g, x_c] = [0, L]$ and choose a grid with equal spacing Δx. For $j = 1, 2, \dots, J+1$ we let $x_j = (j-1)\Delta x$ so that $x_1 = 0$ and $x_{J+1} = L$ are the boundary points. The coefficient $W(x)$ is needed on a staggered grid, for stability and accuracy reasons similar to those for the SIA diffusivity: $W_{j+1/2}$ at $x_{j+1/2} = x_j + \Delta x/2$. Our finite difference approximation of (47) is, therefore,

$$\frac{W_{j+1/2}(u_{j+1} - u_j) - W_{j-1/2}(u_j - u_{j-1})}{\Delta x^2} - \alpha_j u_j = \beta_j \tag{49}$$

For the left end boundary condition we have $u_1 = u_g$ given, which is easy to include in the linear system (below). For the right end boundary condition we have $u_x(L) = \gamma$, which requires more thought. First introduce a notional point x_{J+2}, then require $(u_{J+2} - u_J)/(2\Delta x) = \gamma$, which is a centered approximation to "$u_x(x_c) = \gamma$," and then, using equation (49) in the $j = J+1$ case, eliminate the u_{J+2} variable. This procedure generates the last equation in our linear system.

Thus each iteration solves a linear system of $J+1$ equations of the form $A\mathbf{u} = \mathbf{b}$ or

$$\begin{bmatrix} 1 & & & & \\ W_{3/2} & a_{22} & W_{5/2} & & \\ & W_{5/2} & a_{33} & & \\ & & \ddots & \ddots & \\ & & W_{J-1/2} & a_{JJ} & W_{J+1/2} \\ & & & a_{J+1,J} & a_{J+1,J+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_J \\ u_{J+1} \end{bmatrix} = \begin{bmatrix} u_g \\ \beta_2 \Delta x^2 \\ \beta_3 \Delta x^2 \\ \vdots \\ \beta_J \Delta x^2 \\ b_{J+1} \end{bmatrix} \tag{50}$$

The diagonal entries a_{jj} are

$$a_{22} = -(W_{3/2} + W_{5/2} + \alpha_2 \Delta x^2), \quad \dots, \quad a_{JJ} = -(W_{J-1/2} + W_{J+1/2} + \alpha_J \Delta x^2).$$

There are special cases for the coefficients in the last equation:

$$a_{J+1,J} = 2W_{J+1/2}, \quad a_{J+1,J+1} = -(2W_{J+1/2} + \alpha_{J+1}\Delta x^2).$$

For the right side of the last equation, $b_{J+1} = -2\gamma \Delta x W_{J+3/2} + \beta_{J+1}\Delta x^2$.

System (50) is a *tridiagonal* linear system. It is fully-appropriate to hand this linear system to Matlab's linear solver, the "backslash" operator ($\mathbf{u} = A \setminus \mathbf{b}$), especially at this initial implementation stage. Matlab will identify it as a tridiagonal system and solve it efficiently. (These notes will not worry further about solving linear systems arising from discretization, but see [12].) Collecting these ideas yields the following function for the abstracted problem (47):

flowline.m

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

dx = L / J;
b = dx^2 * beta(:);
b(1) = ug;  b(J+1) = b(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
b = b ./ scale;

u = A \ b;
```

By *manufacturing* exact solutions to (47)—see Notes and References—we can test this first piece before proceeding to solve the actual nonlinear SSA problem. Results from **testflowline.m** (not shown) show that our implemented numerical scheme converges at the correct rate $O(\Delta x^2)$.

The case of an ice shelf. The code **ssaflowline.m** below numerically computes the velocity for an ice shelf. It calls **ssainit.m** (not shown) to get an initial iterate $u^{(0)}(x)$, and then **flowline.m** at each iteration, thereby implementing Picard iteration (46) to solve (45). The thickness $H(x)$ is assumed given, so we are not yet addressing the full, *coupled* ice shelf problem in which one simultaneously solves the mass continuity (21) and stress balance (45) equations for H and u.

ssaflowline.m

```
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 = [(h(2)-h(1))/dx; (h(3:J+1)-h(1:J-1))/(2*dx); (h(J+1)-h(J))/dx];

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 = 0.5 * (H(1:end-1) + H(2:end));
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 = (u(2:end) - u(1:end-1)) / dx;
    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
```

Does **ssaflowline.m** work correctly? The exact thickness and velocity solution shown in Figure 16, as evaluated by **exactshelf.m** (not shown), allows us to evaluate the numerical error. The convergence result in Figure 17 is from codes **testshelf.m** and **shelfconv.m** (not shown), which call **ssaflowline.m**. Even on a coarse 4 km grid the maximum velocity error is less than 1 m/a, while the maximum velocity itself is ~ 300 m/a. In particular, at screen resolution our numerical velocity solutions are the same as shown in Figure 16.

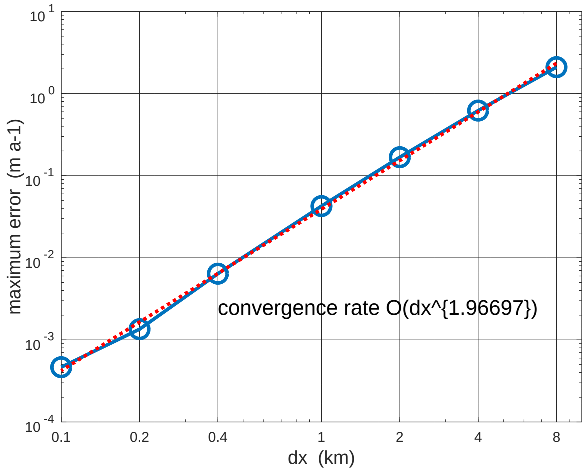


FIGURE 17. The numerical SSA velocity solution from **ssaflowline.m** converges to the exact solution, at nearly the optimal rate $O(\Delta x^2)$, as the grid is refined from spacing $\Delta x = 8000$ m to $\Delta x = 100$ m.

Realistic ice shelf modelling. Real ice shelves have two horizontal variables, are confined in bays, and experience *confining stresses* or side drag, which is to say membrane stresses in 2D. Their velocities vary spatially and temporally along their grounding lines, which are now complicated curves. Furthermore real ice shelves have nontrivial mass and energy processes on their surfaces, including high basal melt near grounding lines, marine ice basal freeze-on, and fracturing which nears full thickness at the calving front. So the modeling of real ice shelves is complicated.

Nonetheless *diagnostic* (fixed geometry) ice shelf modelling in two horizontal variables, where the velocity is unknown but the thickness is known and fixed, is quite successful using only the isothermal SSA model. For example, Figure 18 shows a Parallel Ice Sheet Model (PISM) [14, 72] result for the Ross ice shelf, compared to observed velocities. There is only one tuned parameter, the constant value of the ice hardness B. Note that observed velocities for grounded ice were applied as boundary conditions. Many current ice shelf models yield comparable match [44].

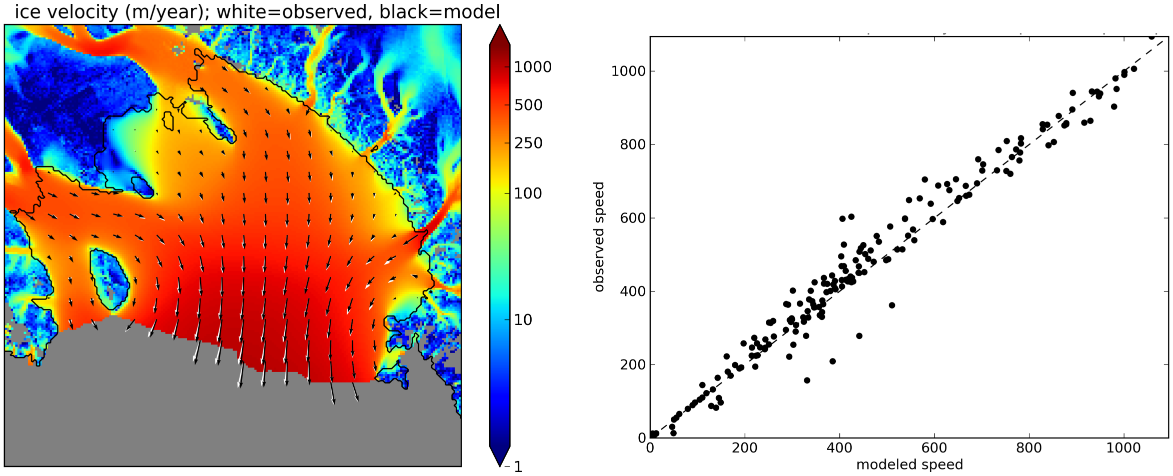


FIGURE 18. Left: Observed (white) and PISM-modeled (black) ice velocities are nearly the same across the whole Ross ice shelf. Right: A scatter plot shows the same points by observed-vs-modeled ice speed.

10. SUMMARY

These brief notes give an incomplete view of numerical ice flow models for glaciers and ice sheets. As a meager conclusion here are four high-level views of such models:

- The mass continuity equation, a conservation equation in the map-plane, describes ice thickness changes. In most ice sheet models it is the component which describes how the geometry evolves. The numerical approach to this equation depends on the form of the stress balance which supplies the ice velocity, but, since it is a conservation equation, it is common to evaluate the ice flux on the staggered grid (finite difference thinking) or on cell boundaries (finite volume thinking).
- The mass continuity equation has diffusive character (SIA-like) at larger spatial scales and in areas with significant basal resistance, but it is *not* very diffusive for membrane-stress dominated areas. Ice streams and shelves are often modeled by the membrane-stress-resolving SSA.
- The SIA stress balance is exceptional because it contains no horizontal derivatives of the velocities. It follows that in the SIA the velocity can be found simply by vertical integration of geometric quantities (e.g. the driving stress). No other stress balance is like this.
- Membrane-stress-resolving balance equations, such as the SSA, Blatter-Pattyn, hydrostatic, and Stokes models, all determine horizontal velocity from solving

spatial, elliptic-like PDEs using the ice geometry and stress boundary conditions. Because of the Glen law these PDEs are nonlinear, so iteration is necessary. At each iteration a sparse matrix "inner" problem is typically solved using a linear-algebra solver package, e.g. a high-performance library like PETSc [12].

We have illustrated some best practices for numerical modelling:

- Return often to the continuum model.
- Modularize your codes.
- Test your modules: Does it show convergence to exact solutions? Is it robust?

11. Notes

Continuum modelling of ice flows is covered in textbooks [19, 27, 69], and in the excellent review [63], among other sources.

The SIA model, which was originally derived by several authors [22, 33, 47], follows from scaling the Stokes equations using the aspect ratio $\epsilon = [H]/[L]$, where $[H]$ is a typical thickness of an ice sheet and $[L]$ is a typical horizontal dimension. After scaling one drops small terms in the $\epsilon \to 0$ limit [21, 33], a "small-parameter argument". In Fowler's form of the scaling argument [21] there are no $O(\epsilon)$ terms in the scaled equations so one only drops $O(\epsilon^2)$ terms. The SIA is re-formulated in [35] as a well-posed free boundary problem, by adding the constraint that the thickness is positive, thereby providing the correct boundary condition at grounded margins; a similar approach is in [11]. The Mahaffy [45] scheme for diffusivity used here is not the only one [31]. The Mahaffy scheme is the basis for an unconditionally-stable implicit scheme [11].

The SSA model [70] was first applied to ice shelves in [46] and ice streams in [43]. In deriving the SSA, the aspect ratio ϵ is one of two small parameters; the other describes the magnitude of surface undulations [60, 64]. A well-posed model for the emergence of ice streams though till failure, using only the SSA, is in [60]. Little theory is known about the well-posedness of the SSA+(mass continuity) coupled equations.

A major modelling issue omitted from these notes is thermomechanical coupling [5]. Temperature is important because the ice softness varies by three orders of magnitude in the temperature range relevant to ice sheet modelling, and because temperature gives ice sheets a long memory of past climate. While the geothermal flux is a significant input in slow-flowing parts of ice sheets, the dissipation of gravitational potential energy is often the dominant heat source, especially as it melts basal ice and facilitates sliding. For example, each year the ice in the Jakobshavn drainage basin in Greenland dissipates enough gravitational potential energy to fully melt more than $1\,\mathrm{km}^3$ of ice [2]. In practice this is reflected as high basal water production. Furthermore it heats large volumes of ice up to the pressure-melting temperature, thereby generating thick layers of temperate ice in outlet glaciers. (Beautiful observation evidence appears in [42].) These physical effects motivate modellers to solve the conservation of energy equation simultaneously with the mass conservation (mass continuity) and momentum conservation (stress balance) equations. The conservation of energy equation is strongly advection-dominated for ice flows. While some models only use temperature as the state variable [15], perhaps suitable for their coldest parts, ice sheets are generally polythermal [22, 26]. Enthalpy methods track the energy content of polythermal ice sheets and glaciers [2, 28], though one may also solve a separate water-content equation in temperate ice [26].

Pressurized basal water is usually required for ice to slide at significant speed. To model the production of such water one must at least compute the basal melt rate via the energy conservation equation [14, 17, 18, 55, 67]. Beyond this statement there is little consensus on what is essential, or even constructive, in terms of process submodels to improve the modeling of ice sheet sliding. Belief in one's preferred cartoon of subglacial morphology will lead to continuum equations for subglacial hydrology, and then numerical experiments will proceed.

One of the most significant issues in modelling ice sheets using shallow models is to describe the "switch", in space and time, between shear-dominated and membrane-stress-dominated flow. It is not a good idea to abruptly switch from the SIA model to the SSA model at the edge of an ice stream, though this was attempted in early models [32, 56]. Instead *hybrid* schemes exist which solve the SIA and SSA everywhere in the ice sheet [14, 54, 72] and then combine the stresses or velocities according to different schemes. Coherent vertically-integrated models have also arisen [8, 24].

Solving the Stokes stress balance normally accomodates the incompressibility constraint by using pressure as an unknown [12, 36, 39, 53]. (Separate Appendix A shows a 2D example.) Numerical approximations of this stress balance are indefinite, thus harder to solve. In plane flow one may address the incompressibility constraint by using stream functions [3].

"Higher-order" three-dimensional approximations of the Stokes stress balance, such as the Blatter-Pattyn model (19) [4, 50, 64], are also shallow approximations in fact. Even Stokes ice sheet models typically apply the most-basic shallow assumption of well-defined thickness (see main text). Higher-order models add an assumption such as hydrostatic normal stress [27]. Computational limitations generally restrict either the spatial extent, the spatial resolution, or the run duration of these models relative to SIA-SSA hybrids, but careful numerical analysis may generate more computationally-efficient solutions [9]. Vertically-integrated and hybrid models will always allow finer map-plane resolution and longer time scales than higher-order and Stokes models, simply because 2D stress balance equations are easier to solve. On the other hand, questions remain about what are the most important deficiencies, relative to the Stokes model, when using either higher-order [53] or hybrid models.

The finite difference material in these notes should probably be read with a reference like [40] or [48] in hand. The "main theorem for numerical PDE schemes" mentioned in the text is the Lax equivalence theorem. Alternative numerical discretization techniques include the finite element [7], finite volume [41], and spectral [65] methods. Newton iteration for the nonlinear discrete equations is superior to Picard iteration used here, in terms of rapid convergence once iterates are near the solution, but care is needed to globalize convergence [37]. High-performance Newton solvers for nonlinear PDEs are demonstrated in [12].

Which are the best numerical models for moving grounding lines? Even when the minimal SSA stress balance is used, this is still something of an open question [20, 25, 51, 52, 61]. The physics requires at least that the quantities H and u are continuous there. Multiple stress balance regimes exist in the vicinity of the grounding line [62].

Where to find exact solutions for ice flow models? Textbooks by van der Veen [69] and Greve and Blatter [27] have a few. Halfar's similarity solution to the SIA [29, 30] has been generalized to non-zero mass balance [16]. There are flow-line [6, 68] and cross-flow [60] solutions to the SSA model, and one can even construct an exact, steady marine ice sheet in the flow-line case [10]. For the Stokes equations themselves there are plane flow solutions for constant viscosity [3].

More generally, for numerical verification purposes one can *manufacture* exact solutions by starting with a specified solution and then deriving a source term so that the specified function is actually a solution [12, 57]. The literature contains such manufactured solutions to the thermomechanically-coupled SIA [15], plane flow Blatter-Pattyn model [23], and Glen-law Stokes equations [36, 39, 58].

References

[1] D. J. Acheson, *Elementary Fluid Dynamics*, Oxford University Press, Oxford, 1990.

[2] A. Aschwanden, E. Bueler, C. Khroulev, and H. Blatter, *An enthalpy formulation for glaciers and ice sheets*, J. Glaciol., 58 (2012), pp. 441–457.

[3] M. Balise and C. Raymond, *Transfer of basal sliding variations to the surface of a linearly-viscous glacier*, J. Glaciol., 31 (1985), pp. 308–318.

[4] H. Blatter, *Velocity and stress fields in grounded glaciers: a simple algorithm for including deviatoric stress gradients*, J. Glaciol., 41 (1995), pp. 333–344.

[5] H. Blatter, R. Greve, and A. Abe-Ouchi, *A short history of the thermomechanical theory and modelling of glaciers and ice sheets*, J. Glaciol., 56 (2010), pp. 1087–1094.

[6] G. Bodvardsson, *On the flow of ice-sheets and glaciers*, Jökull, 5 (1955), pp. 1–8.

[7] D. Braess, *Finite Elements: Theory, fast solvers, and applications in elasticity theory*, Cambridge University Press, 3rd ed., 2007.

[8] D. J. Brinkerhoff and J. Johnson, *Data assimilation and prognostic whole ice sheet modelling with the variationally derived, higher order, open source, and fully parallel ice sheet model VarGlaS*, The Cryosphere, 7 (2013), pp. 1161–1184.

[9] J. Brown, B. Smith, and A. Ahmadia, *Achieving textbook multigrid efficiency for hydrostatic ice sheet flow*, SIAM J. Sci. Computing, 35 (2013), pp. B359–B375.

[10] E. Bueler, *An exact solution for a steady, flow-line marine ice sheet*, J. Glaciol., 60 (2014), pp. 1117–1125.

[11] E. Bueler, *Stable finite volume element schemes for the shallow-ice approximation*, J. Glaciol., 62 (2016), pp. 230–242.

[12] E. Bueler, *PETSc for Partial Differential Equations: Numerical Solutions in C and Python*, SIAM Press, 2021.

[13] E. Bueler, *Conservation laws for free-boundary fluid layers*, SIAM J. Appl. Math., 81 (2021), pp. 2007–2032.

[14] E. Bueler and J. Brown, *Shallow shelf approximation as a "sliding law" in a thermodynamically coupled ice sheet model*, J. Geophys. Res., 114 (2009). F03008, doi:10.1029/2008JF001179.

[15] E. Bueler, J. Brown, and C. Lingle, *Exact solutions to the thermomechanically coupled shallow ice approximation: effective tools for verification*, J. Glaciol., 53 (2007), pp. 499–516.

[16] E. Bueler, C. S. Lingle, J. A. Kallen-Brown, D. N. Covey, and L. N. Bowman, *Exact solutions and numerical verification for isothermal ice sheets*, J. Glaciol., 51 (2005), pp. 291–306.

[17] E. Bueler and W. van Pelt, *Mass-conserving subglacial hydrology in the Parallel Ice Sheet Model version 0.6*, Geosci. Model Dev., 8 (2015), pp. 1613–1635.

[18] G. K. C. Clarke, *Subglacial processes*, Annu. Rev. Earth Planet. Sci., 33 (2005), pp. 247–276.

[19] K. M. Cuffey and W. S. B. Paterson, *The Physics of Glaciers*, Elsevier, 4th ed., 2010.

[20] J. Feldmann, T. Albrecht, C. Khroulev, F. Pattyn, and A. Levermann, *Resolution-dependent performance of grounding line motion in a shallow model compared to a full-Stokes model according to the MISMIP3d intercomparison*, J. Glaciol., 60 (2014), pp. 353–360.

[21] A. C. Fowler, *Mathematical Models in the Applied Sciences*, Cambridge Univ. Press, 1997.

[22] A. C. Fowler and D. A. Larson, *On the flow of polythermal glaciers. I. Model and preliminary analysis*, Proc. R. Soc. Lond. A, 363 (1978), pp. 217–242.

[23] R. Glowinski and J. Rappaz, *Approximation of a nonlinear elliptic problem arising in a non-Newtonian fluid flow model in glaciology*, M2AN Math. Model. Numer. Anal., 37 (2003), pp. 175–186.

[24] D. Goldberg, *A variationally derived, depth-integrated approximation to a higher-order glaciological flow model*, J. Glaciol., 57 (2011), pp. 157–170.

[25] D. Goldberg, D. M. Holland, and C. Schoof, *Grounding line movement and ice shelf buttressing in marine ice sheets*, J. Geophys. Res., 114 (2009).

[26] R. Greve, *A continuum–mechanical formulation for shallow polythermal ice sheets*, Phil. Trans. Royal Soc. London A, 355 (1997), pp. 921–974.

[27] R. Greve and H. Blatter, *Dynamics of Ice Sheets and Glaciers*, Advances in Geophysical and Environmental Mechanics and Mathematics, Springer, 2009.

[28] R. Greve and H. Blatter, *Comparison of thermodynamics solvers in the polythermal ice sheet model SICOPOLIS*, Polar Science, 10 (2016), pp. 11–23.

[29] P. Halfar, *On the dynamics of the ice sheets*, J. Geophys. Res., 86 (1981), pp. 11065–11072.

[30] ——, *On the dynamics of the ice sheets 2*, J. Geophys. Res., 88 (1983), pp. 6043–6051.

[31] R. C. A. Hindmarsh and A. J. Payne, *Time–step limits for stable solutions of the ice–sheet equation*, Ann. Glaciol., 23 (1996), pp. 74–85.

[32] C. L. Hulbe and D. R. MacAyeal, *A new numerical model of coupled inland ice sheet, ice stream, and ice shelf flow and its application to the West Antarctic Ice Sheet*, J. Geophys. Res., 104 (1999), pp. 25349–25366.

[33] K. Hutter, *Theoretical Glaciology*, D. Reidel, 1983.

[34] I. Joughin, S. Tulaczyk, R. Bindschadler, and S. F. Price, *Changes in West Antarctic ice stream velocities: observation and analysis*, J. Geophys. Res., 107 (2002).

[35] G. Jouvet and E. Bueler, *Steady, shallow ice sheets as obstacle problems: well-posedness and finite element approximation*, SIAM J. Appl. Math., 72 (2012), pp. 1292–1314.

[36] G. Jouvet and J. Rappaz, *Analysis and finite element approximation of a nonlinear stationary Stokes problem arising in glaciology*, Advances in Numerical Analysis, (2011).

[37] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, Fundamentals of Algorithms, SIAM Press, 1987.

[38] A. M. Le Brocq, A. J. Payne, and A. Vieli, *An improved Antarctic dataset for high resolution numerical ice sheet models (ALBMAP v1)*, Earth System Science Data, 2 (2010), pp. 247–260.

[39] W. Leng, L. Ju, M. Gunzburger, S. Price, and T. Ringler, *A parallel high-order accurate finite element nonlinear Stokes ice sheet model and benchmark experiments*, J. Geophys. Res., 117 (2012).

[40] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*, SIAM Press, 2007.

[41] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.

[42] M. Lüthi, M. Fahnestock, and M. Truffer, *Correspondence: Calving icebergs indicate a thick layer of temperate ice at the base of Jakobshavn Isbrae, Greenland*, J. Glaciol., 55 (2009), pp. 563 – 566.

[43] D. R. MacAyeal, *Large-scale ice flow over a viscous basal sediment: theory and application to ice stream B, Antarctica*, J. Geophys. Res., 94 (1989), pp. 4071–4087.

[44] D. R. MacAyeal, V. Rommelaere, P. Huybrechts, C. Hulbe, J. Determann, and C. Ritz, *An ice-shelf model test based on the Ross ice shelf*, Ann. Glaciol., 23 (1996), pp. 46–51.

[45] M. W. Mahaffy, *A three–dimensional numerical model of ice sheets: tests on the Barnes Ice Cap, Northwest Territories*, J. Geophys. Res., 81 (1976), pp. 1059–1066.

[46] L. W. Morland, *Unconfined ice-shelf flow*, in Dynamics of the West Antarctic ice sheet, C. J. van der Veen and J. Oerlemans, eds., Kluwer Academic Publishers, 1987, pp. 99–116.

[47] L. W. Morland and I. R. Johnson, *Steady motion of ice sheets*, J. Glaciol., 25 (1980), pp. 229–246.

[48] K. W. Morton and D. F. Mayers, *Numerical Solutions of Partial Differential Equations: An Introduction*, Cambridge University Press, 2nd ed., 2005.

[49] J. F. Nye, W. B. Durham, P. M. Schenk, and J. M. Moore, *The instability of a South Polar Cap on Mars composed of carbon dioxide*, Icarus, 144 (2000), pp. 449–455.

[50] F. Pattyn, *A new three-dimensional higher-order thermomechanical ice sheet model: Basic sensitivity, ice stream development, and ice flow across subglacial lakes*, J. Geophys. Res., 108 (2003).

[51] F. Pattyn, L. Perichon, G. Durand, and others, *Grounding-line migration in plan-view marine ice-sheet models: results of the ice2sea MISMIP3d intercomparison*, J. Glaciol., 59 (2013), pp. 410–422.

[52] F. Pattyn, C. Schoof, L. Perichon, and others, *Results of the Marine Ice Sheet Model Intercomparison Project, MISMIP*, The Cryosphere, 6 (2012), pp. 573–588.

[53] F. Pattyn and twenty others, *Benchmark experiments for higher-order and full Stokes ice sheet models (ISMIP-HOM)*, The Cryosphere, 2 (2008), pp. 95–108.

[54] D. Pollard and R. M. DeConto, *A coupled ice-sheet/ice-shelf/sediment model applied to a marine-margin flowline: Forced and unforced variations*, in Glacial Sedimentary Processes and Products, M. J. Hambrey et al., eds., Blackwell Publishing Ltd., 2007. Special Publication Number 39 of the International Association of Sedimentologists.

[55] C. F. Raymond, *Energy balance of ice streams*, J. Glaciol., 46 (2000), pp. 665–647.

[56] C. Ritz, V. Rommelaere, and C. Dumas, *Modeling the evolution of Antarctic ice sheet over the last 420,000 years: Implications for altitude changes in the Vostok region*, J. Geophys. Res., 106 (2001), pp. 31943–31964.

[57] P. Roache, *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque, New Mexico, 1998.

[58] A. Sargent and J. L. Fastook, *Manufactured analytical solutions for isothermal full-stokes ice sheet models*, The Cryosphere, 4 (2010), pp. 285–311.

[59] J. C. Savage and W. S. B. Paterson, *Borehole measurements in the Athabasca Glacier*, J. Geophys. Res., 68 (1963), pp. 4521–4536.

[60] C. Schoof, *A variational approach to ice stream flow*, J. Fluid Mech., 556 (2006), pp. 227–251.

[61] ——, *Marine ice-sheet dynamics. Part 1. The case of rapid sliding*, J. Fluid Mech., 573 (2007), pp. 27–55.

[62] C. Schoof, *Marine ice sheet dynamics. Part 2: A Stokes Flow contact problem*, J. Fluid Mech., 679 (2011), pp. 122–255.

[63] C. Schoof and I. J. Hewitt, *Ice-sheet dynamics*, Annu. Rev. Fluid Mech., 45 (2013), pp. 217–239.

[64] C. Schoof and R. Hindmarsh, *Thin-film flows with wall slip: an asymptotic analysis of higher order glacier flow models*, Quart. J. Mech. Appl. Math., 63 (2010), pp. 73–114.

[65] L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM Press, 2000.

[66] M. Truffer and K. Echelmeyer, *Of isbrae and ice streams*, Ann. Glaciol., 36 (2003), pp. 66–72.

[67] S. Tulaczyk, W. B. Kamb, and H. F. Engelhardt, *Basal mechanics of Ice Stream B, West Antarctica 2. Undrained plastic bed model*, J. Geophys. Res., 105 (2000), pp. 483–494.

[68] C. J. van der Veen, *A note on the equilibrium profile of a free floating ice shelf.* IMAU Report V83-15. State University Utrecht, Utrecht, 1983.

[69] ——, *Fundamentals of Glacier Dynamics*, CRC Press, 2nd ed., 2013.

[70] M. Weis, R. Greve, and K. Hutter, *Theory of shallow ice shelves*, Continuum Mech. Thermodyn., 11 (1999), pp. 15–50.

[71] P. Wesseling, *Principles of Computational Fluid Dynamics*, Springer-Verlag, 2001.

[72] R. Winkelmann, M. A. Martin, M. Haseloff, T. Albrecht, E. Bueler, C. Khroulev, and A. Levermann, *The Potsdam Parallel Ice Sheet Model (PISM-PIK) Part 1: Model description*, The Cryosphere, 5 (2011), pp. 715–726.