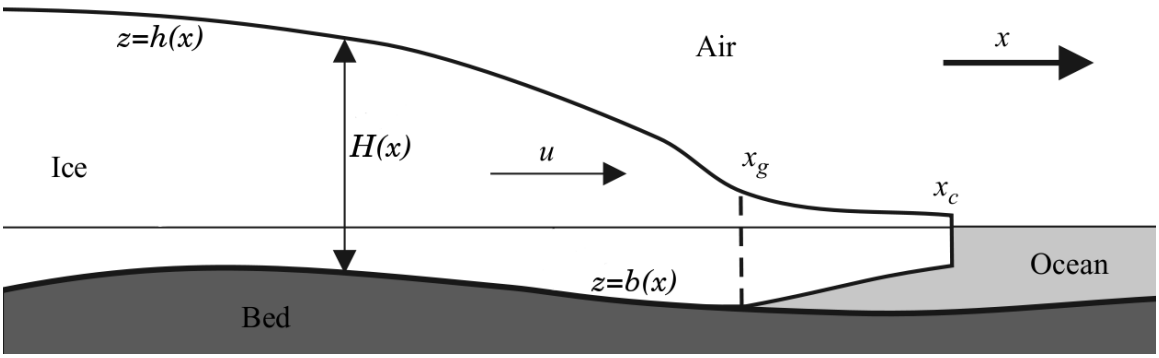
Numerical modelling

of glaciers, ice sheets, and ice shelves

Ed Bueler

International Summer School in Glaciology
McCarthy Alaska, June 2018



Illustrates the notation used in these notes. Figure modified from [58].

1. Introduction

The greatest importance of mathematical and numerical models in science, and glaciology in particular, is a kind of 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? When you put the mathematical model on a computer it will not nod and say "yes, I understand" like a person would—who might not actually understand. The dumb computer can only tell you consequences of your model, inadequacies and all.

Thus numerical models can show how processes interact to give overall behavior, and they can demonstrate flaws in understanding of those processes. But they should 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.

The reader may be surprised that continuum models, and not computer codes, seem to be the focus of these notes much of the time. 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 numerical codes that actually work.

Within our scope are 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, but we do not address its numerical solution except in a separate Appendix A.

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 [25], is covered by Table 1. Cartesian coordinates are x, y, z with z positive-upward. If these coordinates, or the letter t, appear as subscripts then they denote partial derivatives: $u_x = \partial u/\partial x$. Tensor notation also uses subscripts, but from the list $\{1, 2, 3, i, j\}$. Thus "τ_{ij}" and "τ_{13}" denote entries of the deviatoric stress tensor.

Download 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, but with their comments stripped for compactness and clarity. The electronic versions all have generous comments and help files.

TABLE 1. Notation used in these notes, with values for some constants.

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

2. Ice flow equations

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

by numerically solving this equation, we have a usable model for an ice sheet.

This initial model will not be complete by any modern standard. However, a "usable" model tends to be *understood* as much as it is *correct* in a physical sense.

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 it actually is, 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:

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

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

The term $\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u}$ in (2) is an acceleration and the right-hand side is the net force per unit volume $\nabla\cdot\sigma + \rho\mathbf{g}$. Thus equation (2) says "$ma = F$"; it is Newton's second law.

Investigation has yielded only partial understanding of the rich solutions of equations (1) and (2). A book-length introduction like [1] is recommended, but not needed here.

The numerical solution of the Navier-Stokes equations is *computational fluid dynamics* (CFD). One might ask is ice flow modelling a part of CFD? (Or: Does a good general-purpose CFD text like [68] help a glaciers student?) Glacier flow is a large-scale fluid problem like atmosphere and ocean circulation, but, from the point of view of general CFD, it is odd. Topics which might make ocean circulation exciting, for example turbulence, convection, coriolis force, and density stratification, are irrelevant 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 McCarthy 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}), i.e. the solution to (3)–(5), instantaneously. Thus ice flow simulation codes have 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:

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

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

$$-\tau_{13,x} + \tau_{11,z} + p_z = -\rho g \qquad \textit{stress balance } (z) \tag{8}$$

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

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

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 to make us pause before jumping in to numerical solution methods. 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.

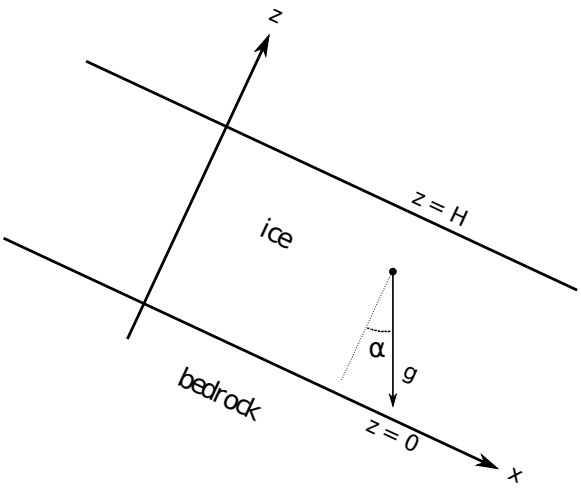


FIGURE 1. Rotated axes for a slab-on-a-slope flow calculation.

We rotate our coordinates only for this example (and not elsewhere in these notes, except in the separate Appendix A). 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), (10), (11), (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 surface of the ice is moving parallel to the x-axis, and that it is stress free: $w(0) = 0$, $p(H) = 0$, $\tau_{13}(H) = 0$. Then, by integrating some of 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 (14) to observations of a mountain glacier. Our model for velocity indeed does a credible job of capturing deformation flow in this case. However, 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.

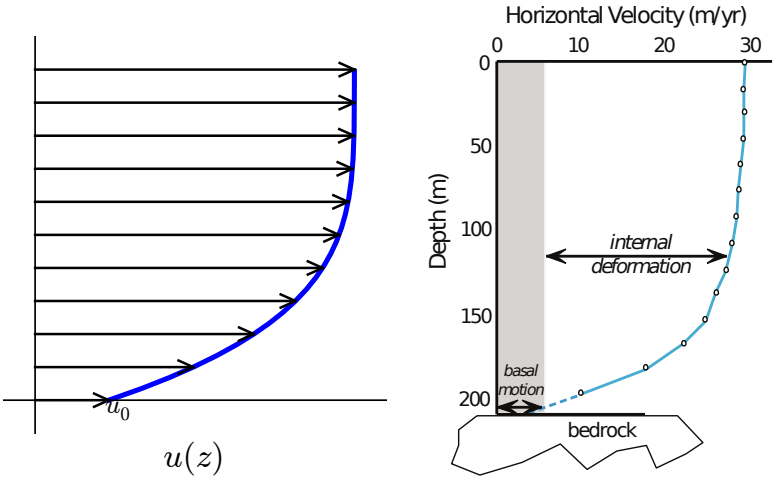


FIGURE 2. Left: Velocity from formula (14). Right: Measured velocity in a glacier (Athabasca Glacier [56]).

Viscosity form of the flow law. Flow law (5) has another form which we will use next, and also later in describing ice shelf and stream flow. 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*.) By taking norms of both sides of (5) we get $|Du| = A\tau^3$ so $\tau = A^{-1/3}|Du|^{1/3}$. Thus (5) can be rewritten

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

where

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

defines the nonlinear *viscosity* ν. Often $B = A^{-1/3}$ is called the *ice hardness*. Flow law (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. (This is similar, but not identical, to the statement that the pressure p 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 [25]:

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

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

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

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

Using this equation one can solve first for the horizontal velocity u and then afterward recover w from (6). In other words the stress balance and incompressibility are coupled only in one direction. Also, because it is similar to the well-understood Poisson equation $u_{xx} + u_{zz} = f$ [45], solving equation (19) is more familiar to numerical analysts than are solutions to (18).

Plane-flow mass-continuity equation. The equations so far do not address the change in shape of the glacier or ice sheet. For this we need the *mass continuity equation*. We derive it first in an informal way, but we also return 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 flux $q = \bar{U}\, H = \int_0^H u \, dz$ 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.

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

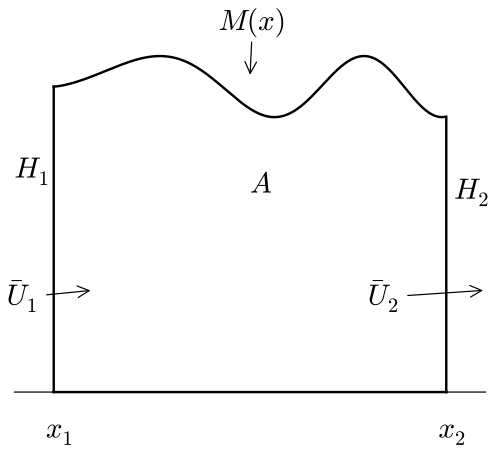


FIGURE 3. Mass continuity equation (21) follows from considering this domain of ice, which has time-dependent area A.

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. In other words, ice sheets *are* shallow. (Note, however, that the portion of an ice sheet which you want to model may not be.)

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 primary 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, 32].

Additional arguments are needed to derive the SIA in the general case. Such arguments reduce the Stokes equations under the assumption of small surface and bed

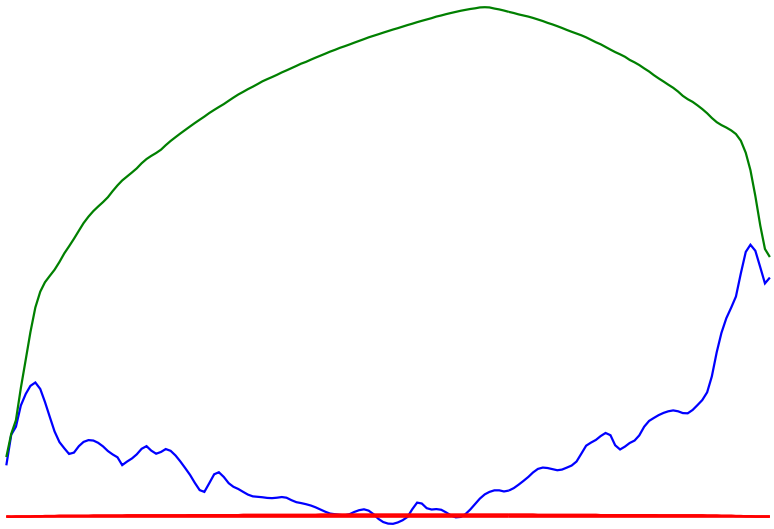


FIGURE 4. A cross-section of the Greenland ice sheet (71° N), with width about 760 km and maximum thickness 3200 m, is shown by the upper two curves. The thickened horizontal line (bottom) shows the same cross-section without exaggeration.

slopes, and small depth-to-width ratio; see Notes and References. Here we state the 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 funciton of 3D coordinates (x, y, z). The shear stress approximation is $(\tau_{13}, \tau_{23}) \approx -\rho g(h-z)\nabla h$. Equation (5) then gives the SIA 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), applies:

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

Equation (25) may also 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. For example, Mahaffy [42] used it in a 1976 numerical model of the Barnes ice cap in Canada, a particularly-simple ice sheet on a rather flat bed.

Analogy with diffusion equations. Numerical methods for solving (26) are modifications of methods for the better-known heat equation, which, in its simplest 1D form, is the equation $T_t = DT_{xx}$. This model of the temperature $T(t,x)$ in a conducting rod, for example, 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: $D \sim \mathrm{m}^2\mathrm{s}^{-1}$. The heat equation has a smoothing effect on the solution T as it evolves in time, because any local maximum or minimum in the temperature is flattened. In fact, $T_{xx} < 0$ implies $T_t < 0$, so T decreases near a maximum, while $T_{xx} > 0$ implies $T_t > 0$ so T increases near a minimum.

The 2D heat equation, which describes the temperature $T(t,x,y)$ of a planar object, is a better analogy to (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 there is a variable heat source $f(t,x,y)$. Then conservation of internal energy says

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

Here ρ is density and c is specific heat capacity. Assuming ρc is constant, define the "diffusivity" $D = \kappa/(\rho c)$. Let $F = f/(\rho c)$, a rescaled source term. Then the revised 2D heat equation is

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

which is similar to the above 1D equation, but 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 maxima or minima, that is, diffusion.

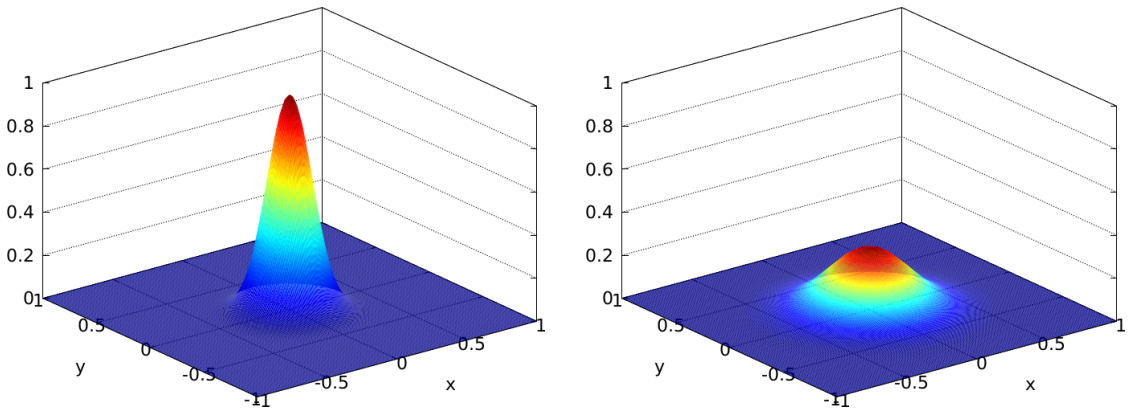


FIGURE 5. Left: Initial condition $T(0,x,y)$. Right: A solution $T(t,x,y)$ of (28), with $D = 1$ and $F = 0$, at $t = 0.02$.

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:

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

When D is large then the diffusion acts most quickly. As a product of powers of H and $|\nabla h|$, D is large if the ice is both thick and steep, because then ice flows downhill most strongly. Our analogy explains why the surfaces of ice sheets are smooth, at least overlooking non-flow processes like crevassing.

There are, however, conceptual and practical concerns about the analogy:

- The letters "H" and "h" denote different quantities in (26). Are we allowed to regard (26) as a diffusion anyway, 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 both more difficult to implement and to analyze.
- The diffusivity D in (29) goes to zero at margins, where $H \to 0$, and at divides and domes, where $|\nabla h| \to 0$. This ultimately means that the solution h is not smooth, even though the thickness is continuous everywhere. The solution h is not smooth even if the bed b and surface mass balance M *are* smooth. A non-smooth solution inevitably generates larger numerical errors.

Just as important is a 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 confined by terrain.

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

All of these concerns are areas of active research in mathematical glaciology; see the Notes and References. For now we continue toward a verified numerical scheme for (26). The next Section introduces numerical PDE methods.

4. Finite difference numerics

The above diffusion analogy suggests that numerical schemes for the heat equation are a good starting point for solving the SIA equation (26). Here we demonstrate only finite difference (FD) schemes. These schemes replace derivatives by mere arithmetic by using Taylor's theorem which says that for a smooth function $f(x)$,

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

In Taylor's theorem 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 solution methods for differential equations the idea is to combine expressions like these to give approximations of the derivatives in the equation. Thereby

an algebraic equation relating grid values will approximate the differential equation. Here we want partial derivative approximations, so we apply the Taylor's expansions one variable at a time. 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 the remainder $O(\Delta^2)$ is smaller than $O(\Delta)$, so the approximation is closer.

Explicit scheme for the heat equation. We first build the simplest "explicit" scheme which approximates the 1D heat equation $T_t = DT_{xx}$. 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) approximates the PDE, making it into 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. 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}:

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

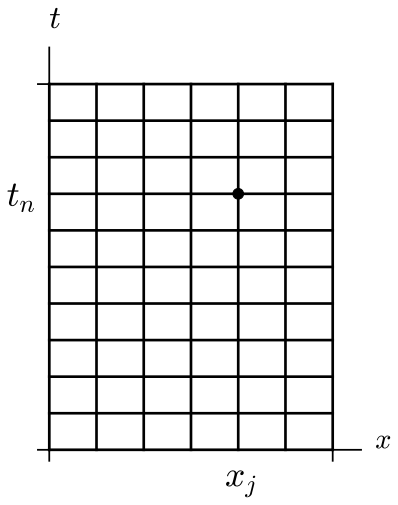


FIGURE 6. A grid for a finite difference solution to the 1D heat equation.

FD scheme (31) is *explicit* because it directly 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 the one value at the next time t_{n+1}.

The expression "$T(t_n, x_j)$" evaluates a heat equation solution at a grid point. This is generally a different number from the value T_j^n computed by a scheme like (31). We intend that the numbers $T(t_n, x_j)$ and T_j^n become closer together as the grid is made finer (i.e. $\Delta t \to 0$ and $\Delta x \to 0$), because the FD expressions become closer to the derivatives they approximate. That is, we intend our FD scheme to *converge* under *grid refinement*. Even if we expect that convergence will happen, this needs checking ("verification"; Section 6).

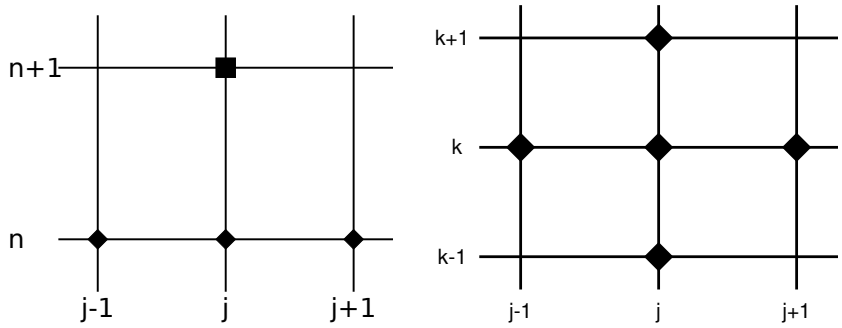


FIGURE 7. Left: Space-time stencil for the explicit scheme (31) for the 1D heat equation. Right: Spatial-only stencil for scheme (33).

How accurate is scheme (31)? The construction of the scheme 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 property of the scheme 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 $O(\Delta t + (\Delta x)^2)$, which is thus the *convergence rate*; see Notes and References. 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 Notes and References.

Instead of pursuing theory, in these notes we do something practical, namely verification using exact solutions. That is, we find problems for which we already know an exact solution $T(t, x)$, and then compute the differences $|T_j^n - T(t_n, x_j)|$. This determines directly whether our actual implementation converges, not just whether it should in theory. Then the verified code can be more 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..

The implementation of (33) is `heat.m` below. It uses Matlab "colon" notation to remove loops over spatial variables. The following example run

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

sets $D = 1.0$, uses a 30×30 spatial grid, and takes $N = 20$ time steps of $\Delta t = 0.001$. The result is shown in Figure 5 (right), 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 cases where the exact solution is bounded.

Stability criteria and adaptive time stepping. To avoid the instability shown above, we need to understand the scheme better. We have not made an implementation error, but we must be more careful when choosing 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 add to one. Averaging is stable because averaged wiggles are smaller than the wiggles themselves. However, the scheme is only an average *if* the middle coefficient is positive. (A linear combination with coefficients which add to one is not an average if any

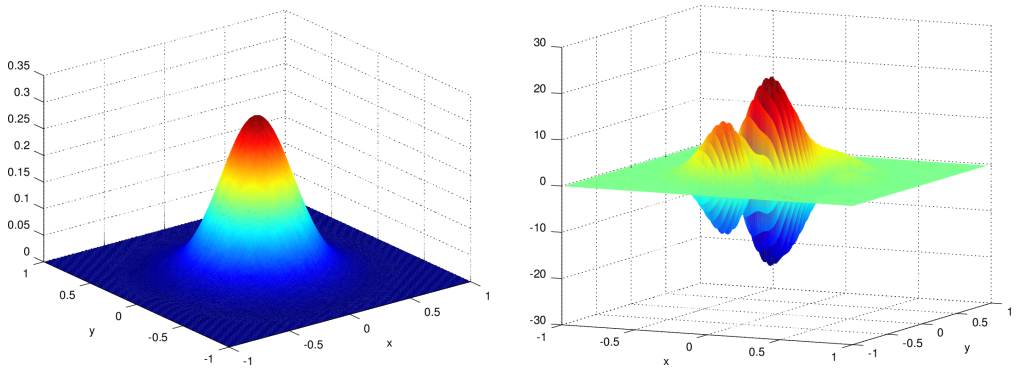


FIGURE 8. Numerically-computed temperature on 40×40 grids. The two runs are the same except that the left has $\Delta t = 0.0005$ and $D\Delta t/(\Delta x)^2 = 0.2$, while the right has $\Delta t = 0.001$ and $D\Delta t/(\Delta x)^2 = 0.4$. Compare (34).

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 $-4 + 5 = 1$.)

So, what follows from requiring the middle coefficient in (31) to be positive so it computes such an average? A *stability criterion*, with these equivalent forms:

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

The third form states the condition as a limitation on the size of Δt. It is a *sufficient* stability criterion; it is enough to guarantee stability, though something weaker might do. In summary, for given Δx, shortening the time steps Δt so that (34) holds will make FD scheme (31) into a stable averaging process.

Applying this same idea to the 2D heat equation (32) leads to the stability condition that $1 - 2\mu^x - 2\mu^y \geq 0$ 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 the two parts of Figure 8.

In summary, runs of `heat.m` are unstable if the time step Δt is too big relative to the spacing Δx. The stability criterion can always be satisfied by making each time step shorter, and enforcing it inside the code is an *adaptive* implementation. We illustrate this idea with small changes to `heat.m` which yield `heatadapt.m` (not shown). Adaptive codes can be stable even if the diffusivity D is changing in time. However, if the diffusivity D is large or the grid spacings Δx, Δy are small, then adaptive explicit implementations must take many short time steps to assure stability; adaptive explicit schemes can be slow.

Implicit schemes. There is an alternative stability fix instead of adaptivity, namely "implicitness." 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 scheme is stable for *any* positive time step $\Delta t > 0$ ("unconditionally stable"); see

Notes and References. Another well-known implicit scheme is *Crank-Nicolson*, which is unconditionally stable for the heat equation, but with smaller error $O((\Delta t)^2+(\Delta x)^2)$.

Implicit schemes are harder 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 and effective when the right tools are used [11]. In the tradeoff between the easy implementability of adaptive explicit schemes and the better stability of implicit schemes, for these notes we choose the former.

Numerical solution of generalized diffusion equations. While we have an analogy which says that the SIA is diffusive, we are trying to numerically model flowing ice not heat conduction. In this section, as a step toward an SIA code which works on nonflat beds, we construct a numerical scheme for a generalized diffusion equation which has an extra "tilt" inside the gradient, namely

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

In (36) the source term $F(x,y)$, diffusivity $D(x,y)$, and tilt $b(x,y)$ may all vary in space. The following code is called by the SIA-specific schemes we build next, with $b(x,y)$ equal to the bedrock elevation.

diffusion.m

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

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

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

```
        mu_x * Dleft  .* ( Tb(2:J,2:K)   - Tb(1:J-1,2:K) );
   T = T + F * dt;
   t = t + dt;     count = count + 1;
end
dtav = tf / count;
```

The adaptive explicit method in `diffusion.m` is conditionally stable, with the same essential time step restriction as for the constant diffusivity case, as long as we evaluate $D(x,y)$ at *staggered* grid points and we the following approximation:

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

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

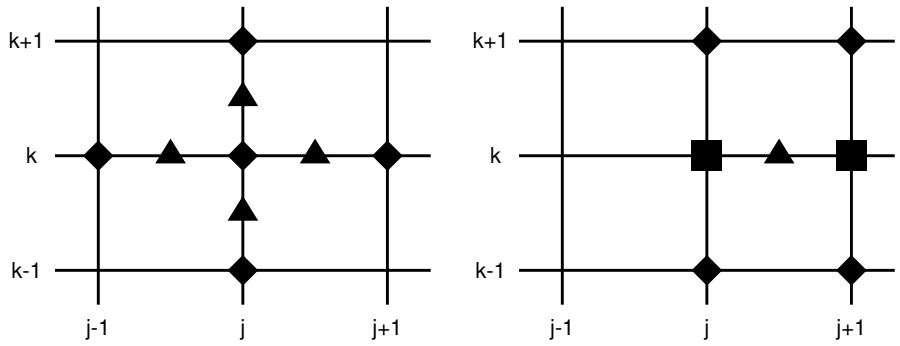


FIGURE 9. Left: Spatial stencil for staggered-grid (triangles) evaluation of diffusivity D in (36). Right: Stencil showing how D is evaluated in the SIA, from surface elevation (diamonds) and thicknesses (squares).

5. Numerically solving the SIA

In SIA equation (26) we have diffusivity $D = \Gamma H^{n+2}|\nabla h|^{n-1}$. As already noted, D goes to zero, i.e. the diffusion "degenerates," when either $H \to 0$ or $\nabla h \to 0$. Degenerate diffusion equations are automatically free boundary problems, but this aspect of the thickness evolution problem should be no surprise to a glaciologist because the determination of the location of the ice margin is a standard goal when modelling a glacier. To address this free boundary problem in an explicit time-stepping scheme it suffices to numerically compute new thicknesses and then set them to zero if they come out negative.

For numerical stability and mass conservation we compute D on a "staggered" grid. Various finite difference schemes 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 we use the Mahaffy [42] method, with the stencil for computing D shown in Figure 9 (right). This code only works for a flat bed and zero surface mass balance, deficiences which will be corrected shortly in an extended code `siageneral.m`.

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;
H = H0;

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

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

6. Exact solutions and verification

In `siaflat.m`, which calls `diffusion.m`, we already have a complicated code. How do we make sure that such an implemented numerical scheme is correct? Here are three proposed techniques:

(i) don't make any mistakes, or

(*ii*) compare your numerical results with results from other researchers, and assume the outliers are in error, or

(*iii*) compare your numerical results to an exact solution.

The first two approaches, which one might call "infallibility" and "intercomparison," respectively, should be less common than they are. The last one, preferred by the CFD community generally [54, 68], is called "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 seeking useful exact solutions.

Exact solution of heat equation. First we return to the simpler 1D heat equation $T_t = DT_{xx}$ with constant D. Many exact solutions are known, but consider 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 and 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. The 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 function $\phi(s)$ by the same power of t:

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

One may then show that $\phi(s) = (4\pi D)^{-1/2}\, e^{-s^2/(4D)}$ so the Green's function is

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

A numerical solver for the 1D heat equation which starts with initial values $T(t_0, x)$, from this exact solution at $t_0 > 0$ should, at a later time $t > t_0$, produce numbers which are close to the exact solution $T(t, x)$; see the Exercises.

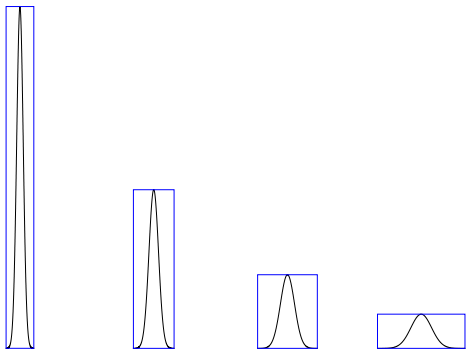


FIGURE 10. The heat equation Green's function in 1D has the same shape at each time, but with time-dependent input- and output-scalings.

Halfar's similarity solution to the SIA. Now we jump to the year 1981. P. Halfar [27] published the similarity solution of the SIA in the case of flat bed and zero surface mass balance. For the 2D SIA model (26) using a Glen exponent $n = 3$ this solution [28] 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 heat, the diffusion of ice slows down dramatically as the shape flattens out because the powers $t^{-1/9}$ and $t^{-1/18}$ change very slowly for large times t.

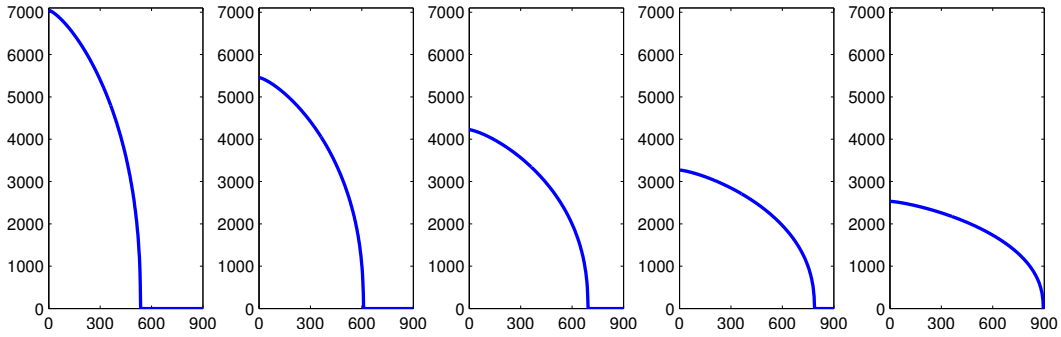


FIGURE 11. Radial sections of Halfar's solution (39) of (26), shown on H (m) versus r (km) axes at $t = 1, 10, 100, 1000, 10000$ years.

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

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

Here the characteristic time satisfies $t_0 = (18\Gamma)^{-1}(7/4)^3 R_0^4 H_0^{-7}$ for center height H_0 and radius R_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. By "compares" we mean it computes the thickness *numerical error*, the absolute differences between the numerical and exact thicknesses at the final time:

```
>> verifysia(20)
average thickness error     = 22.310
>> verifysia(40)
average thickness error     = 9.490
>> verifysia(80)
```

```
average thickness error     = 2.800
>> verifysia(160)
average thickness error     = 1.059
```

We see that the average 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 essential benefit of verification against an exact solution is thus that it is *fragile*! 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. J. Nye and others [46] 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, with an unknown admixture of dust.

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

Are exact solutions like Halfar's always available when needed? The answer is "no." 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 the code `roughice.m` (not shown) demonstrates that `siaflat.m` can handle an ice sheet with extraordinarily large driving stresses. Recall that the driving stress $\tau_d = -\rho g H \nabla h$ appears in the SIA model as the modeled value of the shear stress (τ_{13}, τ_{23}) at the base of the ice. The driving stress is large when the ice is thick and has steep surface slope $|\nabla h|$. In our test we give `siaflat.m` a randomly-generated initial ice sheet which has extraordinarily large driving stresses. It is thick (average of 3000 m), and it has huge surface slopes because of the random surface elevation values, so the product $H|\nabla h|$ is large everywhere. An initial shape is shown in the left side of Figure 12. 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 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 at right in Figure 12. This shape is rather close to a Halfar solution (39), and it illustrates what Halfar proved [27, 28], namely that *all* solutions of the zero-mass-balance SIA on a flat bed will asymptotically approach (39). The Halfar solution is generic, attracting, and worth knowing.

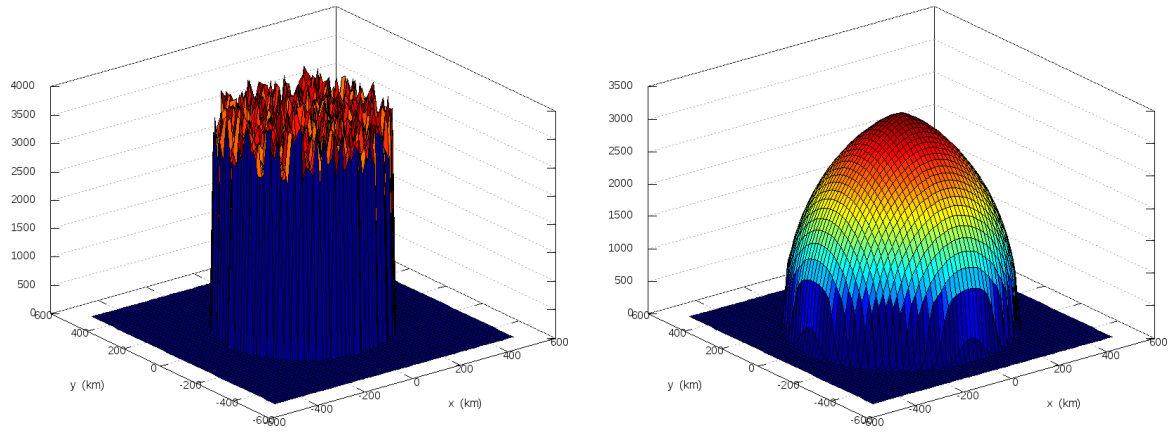


FIGURE 12. Our numerical model evolves the huge-driving-stress initial ice sheet at left to the ice cap at right in only 50 model years.

7. Application to the Antarctic ice sheet

Finally we apply the 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 time-independent climatic mass balance $M(x, y)$, and we enforce non-negative thickness at each timestep. Also we add a minimal model of interaction with the ocean, namely 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`.

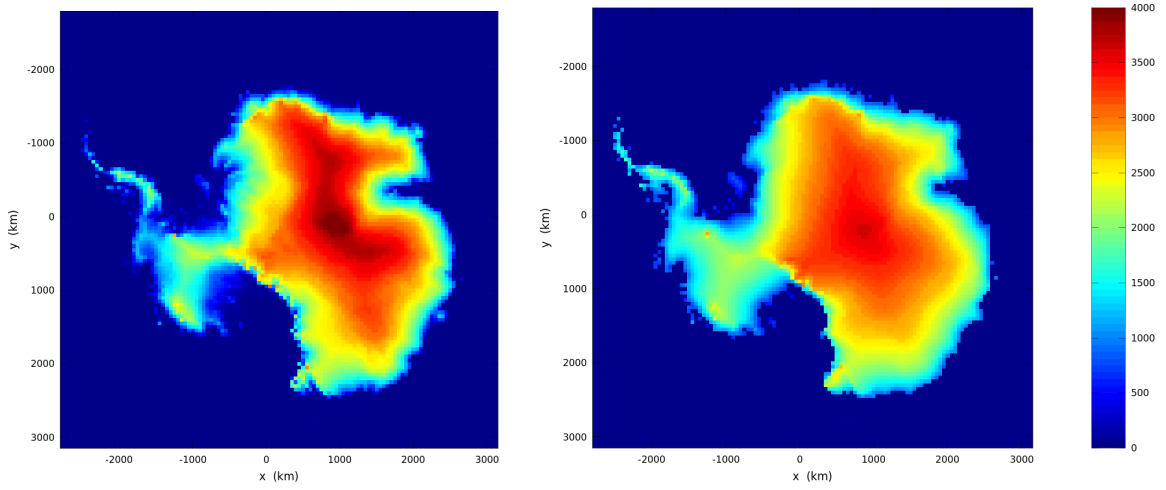


FIGURE 13. Left: Initial surface elevation (m) of Antarctic ice sheet. Right: Final surface elevation at end of 40 ka model run on 50 km grid.

We use measured accumulation, bedrock elevation, and surface elevation from ALBMAPv1 data [35]. Melt is not modelled so the climatic mass balance is equal to the precipitation rate. (This is a more-suitable approximation for Antarctica than for Greenland, but still a rough approximation.) These input data are read from a NetCDF file and preprocessed by an additional code `buildant.m` (not shown).

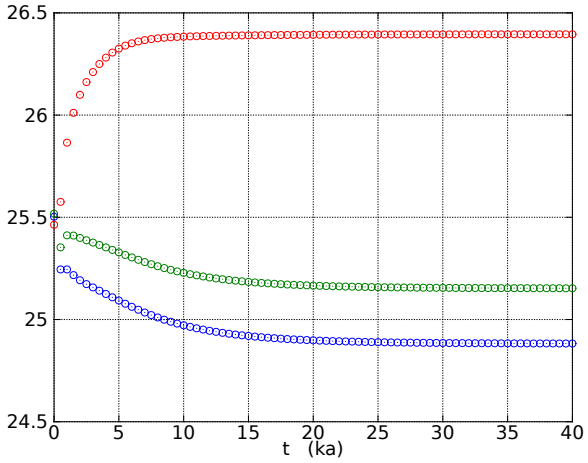


FIGURE 14. Ice volume of the modeled Antarctic ice sheet, in units of $10^6\,\text{km}^3$, from 50 km (top), 25 km (middle), and 20 km (bottom) grids.

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, we can see that 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. (See the Notes and References on widely-used modelling techniques which address these inadequacies.) We should be modelling floating ice too, but the SIA is completely inappropriate to that purpose; compare Section 9.

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 most steep gradients near the ice margin are poorly resolved, but to differing degrees at such coarse resolutions. 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 resolutions to evaluate the robustness of the model results.

8. Mass continuity and kinematical equations

In the shallow ice approximation the stress balance becomes formula (24) for the velocity. This combines with the mass continuity equation (25) to give model (26) for

evolution of the ice sheet thickness. Equation (26) therefore combines two concepts which we now think about separately, and in greater generality.

The most basic shallow assumption made by most ice flow models[1] is that the surface and base of the ice are differentiable functions $z = h(t, x, y)$ and $z = b(t, x, y)$. 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). Here these are measured in thickness-per-time units, which are typically converted from mass-per-area-per-time 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's upper surface and lower surfaces, respectively, from the velocity of the ice and the mass balance functions at the respective surfaces. Note that in the important (idealized) case of non-deformable bedrock and no sliding, (41) says that the basal value of the vertical velocity equals the basal melt rate: $w\big|_b = -s$.

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

- 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. (The details are left for exercises.) Thus the surface kinematical and mass continuity equations are closely-related.

Prognostic models. Consider a general, explicit, "prognostic," i.e. ice geometry evolving, isothermal ice sheet model. Each time step follows this recipe:

- numerically solve an incompressible stress balance, yielding velocity $\mathbf{u} = (u, v, w)$,
  - the geometry of the ice mass and the stress boundary conditions determine $\mathbf{u}$
  - if the stress balance only gives $\mathbf{U} = (u, v)$ then w comes from equation (1)
- decide on a time step Δt for (25) based on velocities and/or diffusivities,

[1] There are several inequivalent shallow stress balance models: SIA, SSA, hybrids, Blatter-Pattyn, ...

- 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 (e.g. $h \mapsto h + H_t \Delta t$), and repeat

Most ice sheet models use the mass continuity equation (25) to describe changes in ice sheet geometry, but they can use the surface kinematical equation (40) instead.

The above "standard ice sheet model" has many variations. First of all, some glaciological questions are answered just by solving the stress balance for the velocity, and sometimes the goal is the steady state configuration of the glacier. The constraint that $H \geq 0$ is important to correctly computing the steady or time-dependent thickness H [11, 32]. 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 is more complete than the isothermal SIA (26).

When using equation (26) one apparently 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. The general formula $\mathbf{q} = \bar{\mathbf{U}} H$ applies both for the SIA and fast flow. An ice flow model is generally neither diffusive nor wave-like (hyperbolic). The ice velocity is always related in some way to the local surface gradient, and generally solving the stress balance for the velocity field is both a nontrivial and an 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 ice shelves. The SSA also applies reasonably well to 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 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 km a^{-1}), but only 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 this stress balance equation only in the plane flow, isothermal case:

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

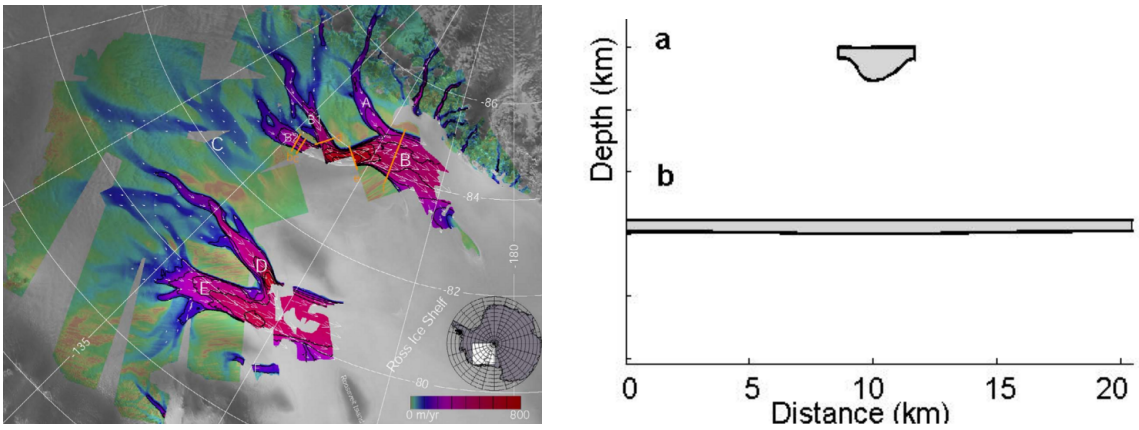


FIGURE 15. Left: The SSA model applies to ice streams like those on the Siple Coast in Antarctica, here colored by radar-derived surface speed. Right: Cross sections, *without* vertical exaggeration, of (**a**) Jakobshavns Isbrae outlet glacier, Greenland and (**b**) Whillans Ice Stream, Siple Coast [63].

The term in parentheses is the vertically-integrated longitudinal stress, also called the "membrane" stress (especially 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 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 (the coefficient BH), the slipperyness of the bed (the coefficient C and the power m) and the geometry of the ice sheet (the 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. (Models which are not isothermal must compute the vertical average of the temperature-dependent hardness.) The coefficient $\bar\nu = B|u_x|^{1/n-1}$ is the "effective viscosity", and (42) can be written

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

In form (43) it is understood that the viscosity $\bar\nu$ depends on u.

The inequality "$\rho H < -\rho_w b$" is called the *flotation criterion*. For grounded ice ($\rho H > -\rho_w b$) the driving stress uses $h = H + b$. For floating ice ($\rho H < -\rho_w b$) one uses $h = (1 - \rho/\rho_w)H$ in the driving stress so that equation (43) simplifies to

$$(2\,\bar\nu\, H u_x)_x = \rho g(1 - \rho/\rho_w) H H_x. \tag{44}$$

Steady ice shelf exact solution. For a steady 1D ice shelf, in which $H_t = 0$, van der Veen [65] built an exact solution by noting that both sides in equation (44) are derivatives. 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 (44) can be found when M is constant ($M = M_0$). This exact solution depends on the ice thickness H_g and velocity u_g at the grounding line (i.e. at 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.

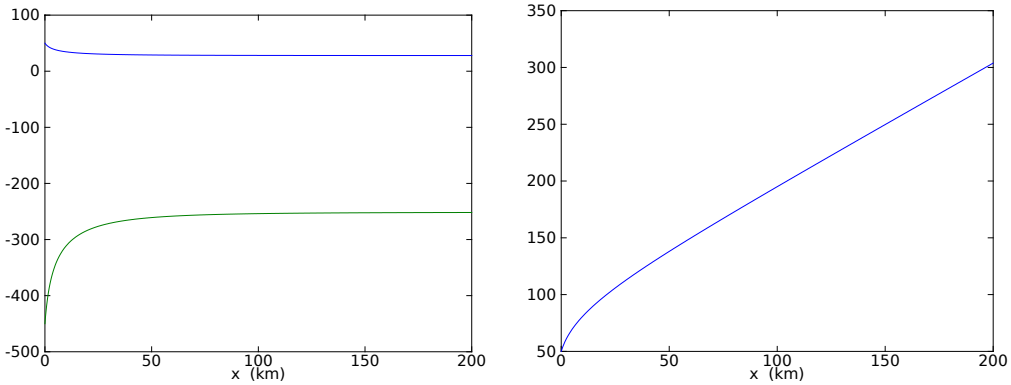


FIGURE 16. The upper and lower surface elevation (m; left) and velocity (m/a; right) of the exact ice shelf solution; $x = 0$ is the grounding line.

Numerical solution of the SSA. Suppose the ice thickness is a fixed function $H(x)$. To find the velocity we must solve the nonlinear PDE (43) or (44) for $u(x)$. When we do this numerically an iteration is needed because of the nonlinearity. 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. This is often called a "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):

$$\left(W^{(k-1)}u_x^{(k)}\right)_x - C|u^{(k-1)}|^{m-1}u^{(k)} = \rho g H h_x. \tag{45}$$

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

$$(W(x)\,u_x)_x - \alpha(x)\,u = \beta(x) \tag{46}$$

where the functions $W(x)$, $\alpha(x)$, $\beta(x)$ are known. Equation (46) applies for x in an interval $[x_g, x_c]$ where x_g, the "grounding line", is a location where the velocity is known, and x_c is the calving front. At x_g we have $u(x_g) = u_g$. In the ice shelf case we also have the calving front condition (see Notes and References)

$$2BH|u_x|^{1/n-1}u_x = \frac{1}{2}\rho(1-\rho/\rho_w)gH^2 \tag{47}$$

at $x = x_c$. Notice that (47) 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 our 1D case. For grounded ice, we may assume ice is

held by basal resistance only: $u^{(0)}(x) = (-C^{-1}\rho g H h_x)^{1/m}$. 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 (46) applies on $[x_g, x_c] = [0, L]$ and choose a grid with equal spacing Δx. For $j = 1, 2, \ldots, J+1$ 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 (46) is, therefore,

$$\frac{W_{j+1/2}(u_{j+1} - u_j) - W_{j-1/2}(u_j - u_{j-1})}{\Delta x^2} - \alpha_j u_j = \beta_j \tag{48}$$

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}. Now require $(u_{J+2} - u_J)/(2\Delta x) = \gamma$, which is a centered approximation to "$u_x(x_c) = \gamma$." Then, using equation (48) 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 $A\mathbf{v} = \mathbf{b}$ or

$$\begin{bmatrix} 1 & & & & & \\ W_{3/2} & A_{22} & W_{5/2} & & & \\ & W_{5/2} & A_{33} & & & \\ & & \ddots & \ddots & & \\ & & W_{J-1/2} & A_{JJ} & W_{J+1/2} \\ & & & A_{J+1,J} & A_{J+1,J+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_J \\ u_{J+1} \end{bmatrix} = \begin{bmatrix} u_g \\ \beta_2 \Delta x^2 \\ \beta_3 \Delta x^2 \\ \vdots \\ \beta_J \Delta x^2 \\ b_{J+1} \end{bmatrix} \tag{49}$$

The diagonal entries A_{jj} are

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

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 (49) is a *tridiagonal* linear system. There is no need to look up how to solve such a linear system because it is fully-appropriate to hand it off to Matlab's linear solver, the "backslash" operator ($\mathbf{v} = A \setminus \mathbf{b}$), especially at this initial implementation stage. Matlab can identify a tridiagonal system and solve it efficiently so these notes we will not worry further about solving finite linear systems. Collecting these ideas yields the following code to solve the abstracted problem (46) using only finite differences and linear algebra:

flowline.m

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

dx = L / J;
rhs = dx^2 * beta(:);
```

```
rhs(1) = V0;
rhs(J+1) = rhs(J+1) - 2 * gamma * dx * W(J+1);

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

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

u = A \ rhs;
```

By "manufacturing" exact solutions to (46)—see Notes and References—we can test this first piece of our SSA-solving codes 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. The thickness $H(x)$ is assumed given, so we are not yet addressing the full, "coupled" ice shelf problem in which one would simultaneously solve the mass continuity (21) and stress balance (44) equations. This code implements Picard iteration (45) to solve (44). It calls `ssainit.m` (not shown) to get an initial iterate $u^{(0)}(x)$ and it calls `flowline.m` at each iteration.

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 velocity solution shown in Figure 16, as computed by `exactshelf.m` (not shown), allows us to evaluate the numerical error. (For this we use the exact thickness also shown in Figure 16.) The convergence result in Figure 17 is from codes `testshelf.m` and `shelfconv.m` (not shown). Even on the coarsest 4 km grid the maximum velocity error is less than 1 m/a, while the maximum velocity itself is ~ 300 m/a. We can conclude that at screen resolution our numerical velocity solutions, even on coarse grids, are the same as shown in Figure 16.

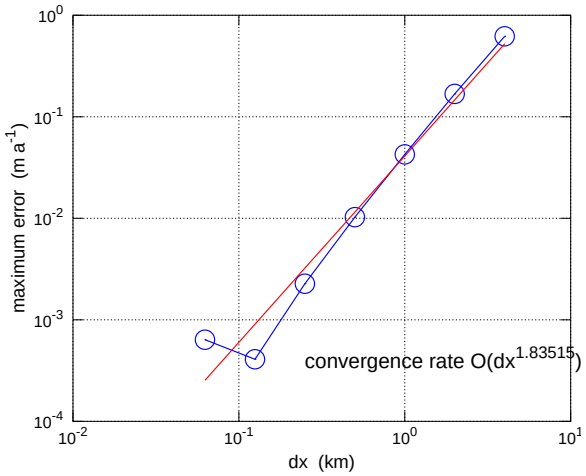


FIGURE 17. The numerical SSA velocity solution from `ssaflowline.m` converges to the exact solution, at nearly the optimal rate $O(\Delta x^2)$, as the grid is refined from spacing $\Delta x = 4$ km to $\Delta x = 62$ m.

Realistic ice shelf modelling. Real ice shelves have two horizontal variables, are frequently confined in bays, and experience "side drag". Their velocities vary spatially and temporally along their grounding lines, namely the curves along which the flotation criterion is an equality. 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. It's 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) 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 [41].

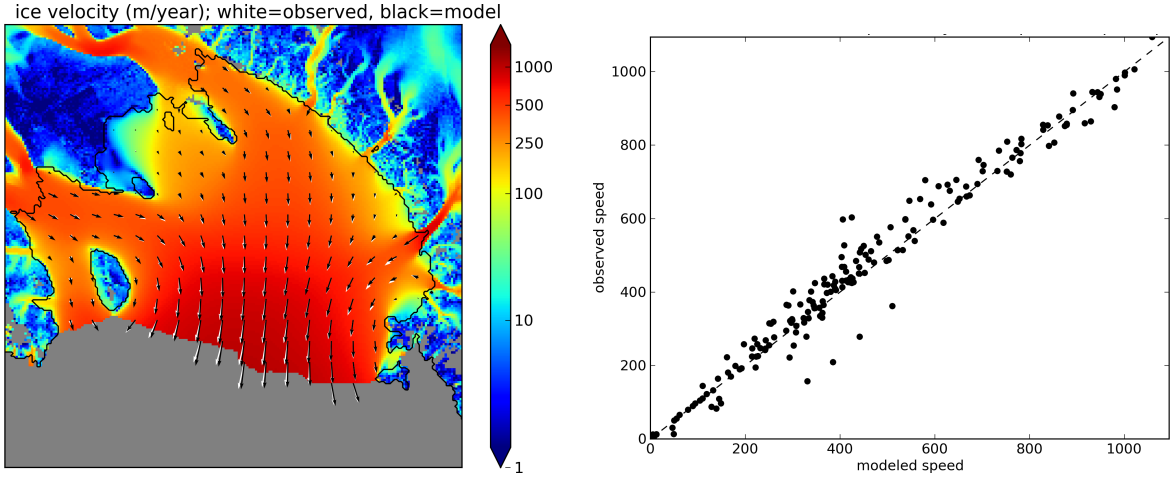


FIGURE 18. Left: Observed (white) and PISM-modeled (black) ice velocities are nearly the same across the whole Ross ice shelf. Right: This scatter plot shows one dot for each arrow at left.

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 with the thickness as the unknown, is the part of an ice sheet model which describes how the ice 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 (in finite difference thinking) or on the boundaries of cells (in finite volume thinking).
- The mass continuity equation has diffusive character (SIA-like) at larger spatial scales and in areas with significant basal resistance. The equation is *not* very diffusive for membrane-stress dominated areas, e.g. those typically modeled by the SSA, especially where there is zero basal resistance as in ice shelves.
- 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).

- Membrane-stress balance equations like the SSA, Blatter-Pattyn, hydrostatic, and Stokes models 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 should be solved using a linear-algebra solver package.

Also, we hope to have illustrated some best practices for numerical modelling:

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

11. Notes

Continuum modelling of ice flows is covered in textbooks [17, 25, 66] and the excellent review [60] among other sources.

The SIA model, which was originally derived by several authors [20, 31, 44], follows by 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 the terms that are small if ϵ is small [19, 31], a "small-parameter argument". In Fowler's form of the scaling argument [19] there are no $O(\epsilon)$ terms in the scaled equations so one only drops $O(\epsilon^2)$ terms. The SIA is re-formulated in [32] 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 [42] scheme for diffusivity used here is not the only one [29]. It can be incorporated into an unconditionally-stable implicit scheme [11].

The SSA model [67] was derived in [43] for ice shelves and in [40] for ice streams. In deriving the SSA, the aspect ratio ϵ is one of two small parameters, where the other describes the magnitude of surface undulations [57, 61]. A well-posed model for the emergence of ice streams though till failure, using only the SSA, is in [57]. Little is known, in theory or practice, about the well-posedness of the SSA+(mass continuity) combination.

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 is a key part of an ice sheet's 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 for melting basal ice and facilitating 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, plus the heating of large volumes of ice up to the pressure-melting temperature, thereby generating thick layers of temperate ice in outlet glaciers. Beautiful evidence of this is shown in [39]. 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 models can use only temperature as the state variable [13], and this may be suitable for their coldest parts, ice sheets are generally polythermal [20, 24]. Enthalpy methods are a good way to track the energy content of polythermal ice sheets and glaciers [2, 26], though one may also solve a separate water-content equation in temperate ice [24].

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 [12, 15, 16, 52, 64]. 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. There is much speculation and experimentation.

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 [30, 53]. However, "hybrid" schemes exist which solve the SIA and SSA everywhere in the ice sheet [12, 69], and then combining the stresses or velocities according to different schemes, or which solve a related vertically-integrated model [8, 22, 51].

"Higher-order" three-dimensional approximations of the Stokes stress balance, such as the Blatter-Pattyn model (19) [4, 47], are also shallow approximations. At minimum their shallowness includes both the most-basic shallow assumption of well-defined thickness (see main text) *and* an assumption of hydrostatic normal stress [25]. Computational limitations generally restrict either the spatial extent, the spatial resolution, or the run duration of these more complete models, primarily because 3D stress balances involve more memory. In the future, careful numerical analysis will generate more computationally-efficient solutions [9]. It will always be the case that vertically-integrated hybrid models will allow higher map-plane resolution and longer time scales than higher-order models because 2D stress balance equations are easier to solve.

As both the SIA and the SSA are derived by small-parameter arguments from the Stokes equations, one might ask whether there is a common shallow antecedent model of both SIA and SSA? Schoof and Hindmarsh [61] answer that Blatter-Pattyn is one.

Solving the Stokes stress balance itself essentially requires a solving the incompressibility using pressure as an unknown [33, 36, 50]. (The separate Appendix A shows a 2D example.) Numerical approximations of this stress balance are indefinite, thus harder to solve, essentially because incompressibility is a constraint on the remaining equations. In plane flow one can instead address the incompressibility constraint by using stream functions [3]. Questions remain about what are the most important deficiencies, relative to the Stokes model, when using either higher-order [50] or hybrid models.

The finite difference material in these notes should probably be read with a reference like [37] or [45] 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 [38], and spectral [62] 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 [34].

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 [18, 23, 48, 49, 58]. The physics requires at least that the quantities H and u are continuous there. Several stress balance regimes exist in the vicinity of the grounding line [59].

Where to find exact solutions for ice flow models? Textbooks by van der Veen [66] and Greve and Blatter [25] have a few. Halfar's similarity solution to the SIA [27, 28] has been generalized to non-zero mass balance [14]. There are flow-line [6, 65] and cross-flow [57] 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]. 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 [54]. The literature contains such manufactured solutions to the thermomechanically-coupled SIA [13], plane flow Blatter-Pattyn model [21], and Glen-law Stokes equations [33, 36, 55].

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 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.

[13] 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.

[14] 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.

[15] 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.
[16] G. K. C. Clarke, *Subglacial processes*, Annu. Rev. Earth Planet. Sci., 33 (2005), pp. 247–276.
[17] K. M. Cuffey and W. S. B. Paterson, *The Physics of Glaciers*, Elsevier, 4th ed., 2010.
[18] 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.
[19] A. C. Fowler, *Mathematical Models in the Applied Sciences*, Cambridge Univ. Press, 1997.
[20] 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.
[21] 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.
[22] D. Goldberg, *A variationally derived, depth-integrated approximation to a higher-order glaciological flow model*, J. Glaciol., 57 (2011), pp. 157–170.
[23] D. Goldberg, D. M. Holland, and C. Schoof, *Grounding line movement and ice shelf buttressing in marine ice sheets*, J. Geophys. Res., 114 (2009).
[24] R. Greve, *A continuum–mechanical formulation for shallow polythermal ice sheets*, Phil. Trans. Royal Soc. London A, 355 (1997), pp. 921–974.
[25] R. Greve and H. Blatter, *Dynamics of Ice Sheets and Glaciers*, Advances in Geophysical and Environmental Mechanics and Mathematics, Springer, 2009.
[26] R. Greve and H. Blatter, *Comparison of thermodynamics solvers in the polythermal ice sheet model SICOPOLIS*, Polar Science, 10 (2016), pp. 11–23.
[27] P. Halfar, *On the dynamics of the ice sheets*, J. Geophys. Res., 86 (1981), pp. 11065–11072.
[28] ——, *On the dynamics of the ice sheets 2*, J. Geophys. Res., 88 (1983), pp. 6043–6051.
[29] 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.
[30] 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.
[31] K. Hutter, *Theoretical Glaciology*, D. Reidel, 1983.
[32] 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.
[33] G. Jouvet and J. Rappaz, *Analysis and finite element approximation of a nonlinear stationary Stokes problem arising in glaciology*, Advances in Numerical Analysis, (2011).
[34] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, Fundamentals of Algorithms, SIAM Press, 1987.
[35] 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.
[36] 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).
[37] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*, SIAM Press, 2007.
[38] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.
[39] 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.
[40] 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.

[41] 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.
[42] 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.
[43] 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.
[44] L. W. Morland and I. R. Johnson, *Steady motion of ice sheets*, J. Glaciol., 25 (1980), pp. 229–246.
[45] K. W. Morton and D. F. Mayers, *Numerical Solutions of Partial Differential Equations: An Introduction*, Cambridge University Press, 2nd ed., 2005.
[46] 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.
[47] 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).
[48] 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.
[49] 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.
[50] 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.
[51] 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.
[52] C. F. Raymond, *Energy balance of ice streams*, J. Glaciol., 46 (2000), pp. 665–647.
[53] 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.
[54] P. Roache, *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque, New Mexico, 1998.
[55] A. Sargent and J. L. Fastook, *Manufactured analytical solutions for isothermal full-stokes ice sheet models*, The Cryosphere, 4 (2010), pp. 285–311.
[56] J. C. Savage and W. S. B. Paterson, *Borehole measurements in the Athabasca Glacier*, J. Geophys. Res., 68 (1963), pp. 4521–4536.
[57] C. Schoof, *A variational approach to ice stream flow*, J. Fluid Mech., 556 (2006), pp. 227–251.
[58] ——, *Marine ice-sheet dynamics. Part 1. The case of rapid sliding*, J. Fluid Mech., 573 (2007), pp. 27–55.
[59] C. Schoof, *Marine ice sheet dynamics. Part 2: A Stokes Flow contact problem*, J. Fluid Mech., 679 (2011), pp. 122–255.
[60] C. Schoof and I. J. Hewitt, *Ice-sheet dynamics*, Annu. Rev. Fluid Mech., 45 (2013), pp. 217–239.
[61] 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.
[62] L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM Press, 2000.
[63] M. Truffer and K. Echelmeyer, *Of isbrae and ice streams*, Ann. Glaciol., 36 (2003), pp. 66–72.
[64] 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.

[65] 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.
[66] ——, *Fundamentals of Glacier Dynamics*, CRC Press, 2nd ed., 2013.
[67] M. Weis, R. Greve, and K. Hutter, *Theory of shallow ice shelves*, Continuum Mech. Thermodyn., 11 (1999), pp. 15–50.
[68] P. Wesseling, *Principles of Computational Fluid Dynamics*, Springer-Verlag, 2001.
[69] 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.

Exercises

1. Assume f has continuous derivatives of all orders. Show using Taylor's theorem:

$$f'(x) = \frac{f(x+\Delta) - f(x-\Delta)}{2\Delta} + O(\Delta^2) \quad \text{and} \quad f''(x) = \frac{f(x+\Delta) - 2f(x) + f(x-\Delta)}{\Delta^2} + O(\Delta^2).$$

2. Sometimes we want finite difference approximations for derivatives in-between grid points. Continuing exercise **1**, show $f'(x + (\Delta/2)) = (f(x+\Delta) - f(x))/\Delta + O(\Delta^2)$.

3. Rewrite `heat.m` using `for` loops instead of colon notation. (The only purpose here is to help understand colon notation.)

4. The 1D explicit scheme (31) for the heat equation, namely $T_j^{n+1} = \mu T_{j+1}^n + (1-2\mu)T_j^n + \mu T_{j-1}^n$, is averaging if stability criterion (34) holds. But of course we must be stepping *forward* in time. Show that the scheme is not averaging for any values of $\Delta t < 0$. Try running `heat.m` backward in time to see what happens. Moral: *there are no consistent and stable numerical schemes for unstable PDE problems.*

5. Show that when written as a formula for T_j^{n+1}, implicit scheme (35) has only positive coefficients. Using [45] etc., as needed, explain why this shows it is unconditionally stable.

6. *This multi-part exercise concerns the numerical treatment of* $\nabla \cdot (D\nabla \dots)$ *in equation* (28).

(a) Write down the obvious centered $O(\Delta t) + O(\Delta x^2)$ explicit finite difference method for the equation $u_t = D_0 u_{xx} + E_0 u_x$, assuming $D_0 > 0$ and E_0 are constant. Solve the scheme for the unknown u_j^{n+1}.

(b) Stability for the method in (a) occurs if u_j^{n+1} is computed from a linear combination of time t_n values which has all nonnegative coefficients. If $|E_0| \gg D_0$, what does this stability assertion say about Δx?

(c) Show that if $D = D(x,y)$ and $u = u(x,y)$ then $\nabla \cdot (D\nabla u) = D\nabla^2 u + \nabla D \cdot \nabla u$.

(d) Why do we use the staggered grid for "$\nabla \cdot (D\nabla)$," instead of expanding by the product rule as in part (c)? References [37, 45] may help you formulate your answer.

7. Derive the Green's function of the 1D heat equation, namely $T = (4\pi D t)^{-1/2} e^{-x^2/(4Dt)}$, which is a solution to $T_t = DT_{xx}$. Start by supposing there is a solution of the form $T(t,x) = t^{-1/2}\phi(s)$ where $s = t^{-1/2}x$ is the similarity variable. Thereby write down an ordinary differential equation in s for $\phi(s)$, and solve it.

8. In the text, and the code `verifysia.m`, we used Halfar's solution to verify our numerical scheme `siaflat.m`. Create the analogous code `verifyheat.m` to use the Green's function of the 2D heat equation (32)—see the last exercise—to verify `heatadapt.m`. You can use the

high quality approximation $e^{-A^2} \approx 0$ for $|A| > 10$ to choose a rectangular domain in space for which you may use $T = 0$ Dirichlet boundary conditions.

9. Is P. Halfar male or female, and what does the first initial "P." stand for? (*I don't know the answers to these questions.*)

10. Show that the Halfar solution formula (39) solves the SIA equation (26) in the case of flat bed ($h = H$), zero climatic mass balance ($M = 0$), and $n = 3$. You will want to express divergence and gradient in polar coordinates.

11. Show, by changing variables in integrating formula (39), that the volume of ice in the Halfar solution is independent of t.

12. In the text it is claimed that any modification of `siaflat.m` will make the output of `verifysia.m` show non-convergence, e.g. the reported average thickness error will not go to zero as the grid is refined. By randomly altering lines of `siaflat.m`, or by other methods of your choice, evaluate this claim.

13. Some output from `verifysia.m` has been suppressed in the text, including a map-plane view of the numerical ice thickness error. Run the program to see this error map. Near the grounded margin of an ice sheet this error is much larger than elsewhere. Why? On the same grids, would a higher-order or Stokes model for an ice sheet with an advancing margin have significantly smaller thickness error, supposing we knew a relevant exact solution?

14. Derive the surface kinematical equation (40) from the mass continuity (25) and base kinematical (41) equations. You will use the incompressibility of ice, equation (1). Note that the Leibniz rule for differentiating integrals, mentioned in the text, is

$$\frac{d}{dx}\left(\int_{g(x)}^{f(x)} h(x,y)\,dy\right) = f'(x)h(x,f(x)) - g'(x)h(x,g(x)) + \int_{g(x)}^{f(x)} h_x(x,y)\,dy.$$

15. Let $C_s = A(\rho g(1-r)/4)^n$ where $r = \rho/\rho_w$. Assume $x_g = 0$ is the location of the grounding line. Derive the two parts of the van der Veen exact ice shelf solution, namely

$$u = \left[u_g^{n+1} + (C_s/M_0)\left((M_0 x + u_g H_g)^{n+1} - (u_g H_g)^{n+1}\right)\right]^{1/(n+1)}, \quad H = (M_0 x + u_g H_g)/u,$$

Start from equation (44), and use the fact $(H^2)_x = 2HH_x$ to generate the first integral of (44). Also use boundary condition (47). Get an equation $u_x = C_s H^n$ or equivalent. On the other hand, note that the mass continuity equation $M_0 = (uH)_x$ can be integrated to give the formula $uH = M_0(x - x_g) + u_g H_g$ for the flux. Use this expression for uH to find $u(x)$ first, and then write $H(x)$ in terms of $u(x)$ as above. These exact solutions are from [65]. They are used in code `exactshelf.m`.

16. Modify the code `ssaflowline.m` to solve the ice stream SSA equation (43). For boundary conditions it would be reasonable to have fixed velocity at the upstream end, but keep the calving front boundary condition at the downstream end. For instance, consider the case where the calving front is at the point where the ice stream reaches flotation. (See also [6, 10].)

17. Derive the hydrostatic (18) equation, in detail.

18. Build a better ice sheet model.