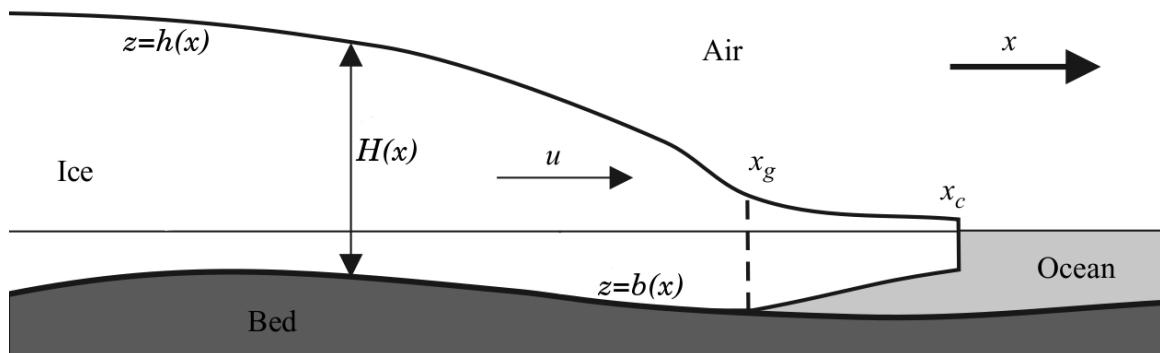


# NUMERICAL MODELLING OF ICE SHEETS, STREAMS, AND SHELVES

Ed Bueler

Karthaus, September 2014



Illustrates the notation used in these notes. Figure modified from [50].

## 1. INTRODUCTION

The most common use of numerical models in glaciology may be to help you ask: When I put together my incomplete understanding of glacier processes into a mathematical model, does the combination behave as I expect? Numerical models can at least demonstrate flaws in our understanding of glacier processes, and they can show us how these processes combine to give overall behavior, but they should be built with care. The worst outcome is to spend time (and perhaps reputation) explaining, through physical argumentation and perhaps using observational evidence, numerical model behavior that is an artifact of poor computer programming or numerical analysis.

So the reader of these notes may be surprised that a continuum model, and not a computer code, seems to be our focus much of the time. While all codes produce some numbers, we want numbers that actually come from our continuum model. We will therefore analyse numerical implementations to see if they match the continuum model and its solutions.

These notes have a limited scope:

*shallow approximations of ice flow.*

They adopt a constructive approach; we provide:

*example 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. We recall the Stokes model, but we do not address its numerical solution. Our numerical concepts include finite difference schemes, solving algebraic systems from stress balances, and the verification of codes using exact solutions.

Our notation, which generally follows [19], is common in the glaciological literature, but see Table 1. Cartesian coordinates  $x, y, z$  have  $z$  perpendicular to the geoid and positive-upward. If these coordinates or “ $t$ ” appear as subscripts then they denote partial derivatives:  $u_x = \partial u / \partial x$ . Tensor notation uses subscripts from the list  $\{1, 2, 3, i, j\}$ . For example, “ $\tau_{ij}$ ” or “ $\tau_{13}$ ” denote entries of the deviatoric stress tensor.

These notes are based on eighteen Matlab codes, each about one-half page. All have been tested in Matlab and Octave. They are distributed in .zip form at

<https://github.com/bueler/karthaus/releases>

Though only five of the codes are printed here, with their comments stripped for compactness and clarity, the electronic versions have generous comments and help files.

## 2. ICE FLOW EQUATIONS

My first goal in these notes is to get to an equation for which I can say:

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

A “usable” model tends to be *understood* as much as it is *correct*. Also, this first model will not be complete by any modern standard. To get to my goal I first (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

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; $B = A^{-1/n}$	$\text{Pas}^{1/n}$	
$b$	bedrock elevation	m	
$c$	specific heat in general	$\text{J kg}^{-1} \text{K}^{-1}$	
$\nabla$	(spatial) gradient	$\text{m}^{-1}$	
$\nabla \cdot$	(spatial) divergence	$\text{m}^{-1}$	
$\mathbf{g}$	gravity	$\text{m s}^{-2}$	9.81
$H$	ice thickness	m	
$h$	ice surface elevation	m	
$\kappa$	conductivity in general	$\text{J s}^{-1} \text{m}^{-1} \text{K}^{-1}$	
$M$	climatic mass balance	$\text{m s}^{-1}$	
$n$	exponent in Glen flow law		3
$\nu$	viscosity	$\text{Pa s}$	
$p$	pressure	Pa	
$\mathbf{q}$	map-plane ice flux: $\mathbf{q} = \int_b^h \mathbf{U} dx = \bar{\mathbf{U}} H$	$\text{m}^2 \text{s}^{-1}$	
$\rho$	(1) density in general	$\text{kg m}^{-3}$	
	(2) density of ice	$\text{kg m}^{-3}$	910
$\rho_w$	density of sea water	$\text{kg m}^{-3}$	1028
$T$	temperature	K	
$\tau$	magnitude of $\tau_{ij}$ : $2\tau^2 = \sum_{ij} \tau_{ij}^2$	Pa	
$\tau_{ij}$	deviatoric stress tensor	Pa	
$Du_{ij}$	strain rate tensor	$\text{s}^{-1}$	
$\mathbf{U}$	$= (u, v)$ horizontal ice velocity	$\text{m s}^{-1}$	
$\mathbf{u}$	$= (u, v, w)$ 3D ice velocity	$\text{m s}^{-1}$	

liquid water, then it would be a “typical” incompressible fluid. We would use the Navier-Stokes equations as the model:

$$\nabla \cdot \mathbf{u} = 0 \quad \text{incompressibility} \quad (1)$$

$$\rho (\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \nabla \cdot (\nu \nabla \mathbf{u}) + \rho \mathbf{g} \quad \text{stress balance} \quad (2)$$

In equation (2) the term  $\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}$  is an acceleration. The right-hand side of (2) is the total stress, and so equation (2) says “ $ma = F$ ”, i.e. it is Newton’s second law. Much time has been spent to get partial understanding of the rich solutions of these Navier-Stokes equations; a book-length introduction like [1] is recommended. The numerical solution of these equations is *computational fluid dynamics* (CFD).

But, is numerical ice flow modelling a part of CFD? Does a well-written general-purpose CFD text like [60] help the glaciers student? Ice sheet flow is a large-scale fluid problem like atmosphere and ocean circulation in climate systems, but it is an odd one. Consider some topics which might make ocean circulation modelling exciting, for example:

turbulence      convection      coriolis force      density stratification

None of these topics are relevant to ice flow. What could be interesting about the flow of slow and old, and surely boring, ice?

First observe that ice is indeed a slow fluid. In terms of equation (2), “slow” means  $\rho(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) \approx 0$ , which says 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  $\nu$  in (2) is therefore not constant, and so we separately state an empirically-based flow law below.

**Stokes equations.** So now the standard model for isothermal flow is this set of Stokes equations:

$$\nabla \cdot \mathbf{u} = 0 \quad \text{incompressibility} \quad (3)$$

$$0 = -\nabla p + \nabla \cdot \tau_{ij} + \rho \mathbf{g} \quad \text{stress balance} \quad (4)$$

$$Du_{ij} = A\tau^2 \tau_{ij} \quad n=3 \text{ Glen flow law} \quad (5)$$

In the flow law (5), the deviatoric stress tensor  $\tau_{ij}$  and the strain rate tensor  $Du_{ij}$  appear; previous lectures cover these. Here we merely note that:  $Du_{ij} = (1/2)((u_i)_{x_j} + (u_j)_{x_i})$  if we index coordinates by  $x_1, x_2, x_3 = x, y, z$ , each tensor in (5) is symmetric and has trace zero, and  $\tau^2 = (1/2)\tau_{ij}\tau_{ij}$  defines “ $\tau^2$ ” in (5) if we use the summation convention.

The Stokes equations do not contain a time derivative. Thus boundary stresses, the force of gravity  $\rho \mathbf{g}$ , and ice softness  $A$  together determine the velocity and stress fields (i.e.  $\mathbf{u}, p, \tau_{ij}$ ) 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 codes, because it is not needed for (re)starting a simulation.

**Plane-flow Stokes equations.** Consider now the  $x, z$ -plane case of equations (3), (4), and (5). “Plane-flow” means that velocity component  $v$  is zero and that all derivatives with respect to  $y$  are zero:

$$u_x + w_z = 0 \quad \text{incompressibility} \quad (6)$$

$$p_x = \tau_{11,x} + \tau_{13,z} \quad \text{stress balance } (x) \quad (7)$$

$$p_z = \tau_{13,x} - \tau_{11,z} - \rho g \quad \text{stress balance } (z) \quad (8)$$

$$u_x = A\tau^2 \tau_{11} \quad \text{flow law (diagonal)} \quad (9)$$

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

Note that  $\tau_{13}$  is a shear stress while  $\tau_{11}$  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, but we can handle a simplified situation first. A uniform slab of ice, or a “slab-on-a-slope”, is both a case in which we actually solve the Stokes equations, and a motivation for the shallow model in the next subsection.

We rotate our coordinates only for this example and not elsewhere in these notes. The two-dimensional axes  $(x, z)$  shown in Figure 1 are rotated downward (clockwise) at angle  $\alpha > 0$  so that the gravity vector has components  $\mathbf{g} = (g \sin \alpha, -g \cos \alpha)$ . Equations (7)

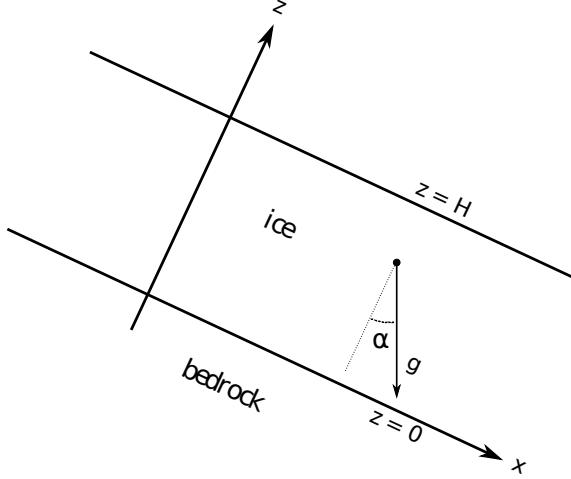


FIGURE 1. Rotated axes for a slab-on-a-slope flow calculation.

and (8) in these rotated coordinates are

$$p_x = \tau_{11,x} + \tau_{13,z} + \rho g \sin \alpha, \quad (11)$$

$$p_z = \tau_{13,x} - \tau_{11,z} - \rho g \cos \alpha. \quad (12)$$

Assuming also that there is no variation with  $x$ , the whole set of Stokes equations (6), (9), (10), (11), (12) simplifies greatly:

$$w_z = 0 \quad 0 = \tau_{11} \quad (13)$$

$$\tau_{13,z} = -\rho g \sin \alpha \quad u_z = 2A\tau^2 \tau_{13}$$

$$p_z = -\tau_{11,z} = -\rho g \cos \alpha$$

We apply boundary conditions for these functions of  $z$ :  $w(0) = 0$ ,  $p(H) = 0$ ,  $u(0) = u_0$ . The basal velocity  $u_0$  will remain undetermined for now.

By integrating equations (13) vertically and using  $\tau_{11} = 0$ , 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, so both the pressure  $p$  and shear stress  $\tau_{13}$  are proportional to depth. Because  $u_z = 2A\tau^2 \tau_{13}$ , by integrating vertically again we find the horizontal velocity:

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

Do we believe formula (14)? Figure 2 compares to observations of a mountain glacier. This comparison shows we have at least done a credible job of capturing deformation flow velocity, though we do not yet have a model for the sliding velocity  $u_0$  (i.e. basal motion).

**Plane-flow mass-continuity equation.** Observe that the equations so far do not address the change in shape of the glacier or ice sheet. For this we need another equation, the *mass continuity equation*. First, define the vertical average of velocity:

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

The flux  $q = \bar{U} H$  is the rate of flow input into the side of the area in Figure 3.

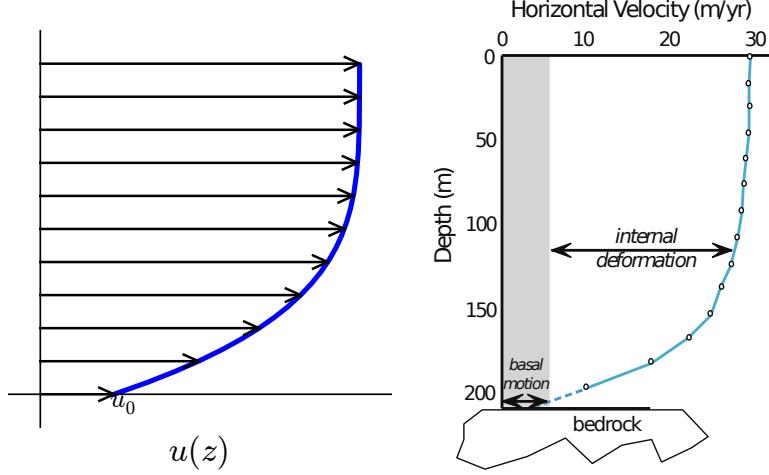


FIGURE 2. Left: Velocity from slab-on-a-slope formula (14). Right: Inclinometry-measured velocity in a glacier (Athabasca Glacier [48]).

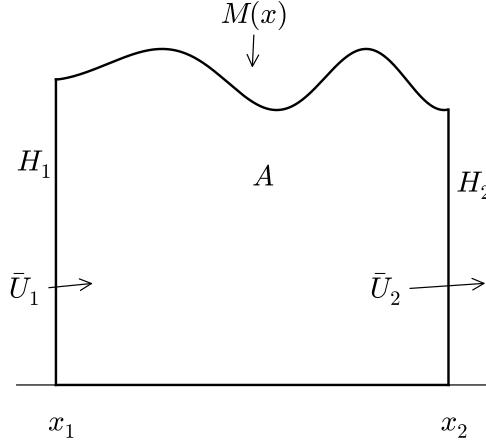


FIGURE 3. Mass continuity equation (16) follows from considering the changing area  $A$  of ice in a planar flow. Ice can be added by surface mass balance  $M$  or a difference of flux  $q = \bar{u}H$  into the left and right sides.

The ice area  $A$  in Figure 3 changes by adding all the boundary contributions,

$$\frac{dA}{dt} = \int_{x_1}^{x_2} M(x) dx + \bar{U}_1 H_1 - \bar{U}_2 H_2 : \quad (15)$$

Here  $M(x)$  is the climatic mass balance at the ice surface. (In three-dimensions, equation (15) would be an equation for  $dV/dt$ , the rate of change of ice volume.)

If the width  $\Delta x = x_2 - x_1$  is small then  $A \approx \Delta x H$ . So we divide by  $\Delta x$  and take  $\Delta x \rightarrow 0$  in (15) and get

$$H_t = M - (\bar{U}H)_x \quad (16)$$

This mass continuity equation describes change in the ice thickness in terms of surface mass balance and ice velocity. It is a major “use” of the velocity in ice flow simulations.

**Viscosity form of the flow law.** The flow law (5) has another form which we will use next, and later in describing ice shelf and stream flow. Recall  $\tau^2 = (1/2)\tau_{ij}\tau_{ij}$ , which

uses the summation convention. Also define  $|Du|^2 = (1/2)Du_{ij}Du_{ij}$ . The scalars  $\tau$  and  $|Du|$  are “norms” (also “second invariants”) of the tensors  $\tau_{ij}$  and  $Du_{ij}$ , respectively. By taking these norms of both sides of (5) we get  $|Du| = A\tau^3$ . But then  $\tau = A^{-1/3}|Du|^{1/3}$ , so (5) can be rewritten

$$\tau_{ij} = 2\nu Du_{ij} \quad \text{flow law (viscosity form)} \quad (17)$$

where  $\nu = (1/2)A^{-1/3}|Du|^{-2/3}$  is the nonlinear viscosity. Often  $B = A^{-1/3}$  is called the ice “hardness”. The derivation of (17) is worth knowing in detail; see the Exercises.

Form (17) of the flow law allows us to eliminate stresses  $\tau_{ij}$  from the Stokes equations by replacing them with formulas depending on derivatives of the velocity, that is, on the strain rates only. The next two approximate models also use this idea.

**The hydrostatic and Blatter-Pattyn approximations.** We return now briefly to plane-flow Stokes equations (6)–(10), and reconsider how to simplify them. One simplification step, present in all shallow models, is the “hydrostatic” approximation. It drops the single term  $\tau_{13,x}$  from the  $z$ -component of the stress balance (8). That is, it assumes that horizontal variation in the vertical shear stress is small compared to the other terms:

$$p_z = -\tau_{11,z} - \rho g. \quad (18)$$

Because the (Cauchy) stress tensor  $\sigma_{ij}$  is related to the deviatoric stress tensor by  $\sigma_{ij} = \tau_{ij} - p\delta_{ij}$ , and thus  $p + \tau_{11} = p - \tau_{33} = -\sigma_{33}$ , equation (18) says that the vertical normal stress  $\sigma_{33}$  is linear in depth. Taking it to have surface value zero we get

$$p + \tau_{11} = \rho g(h - z). \quad (19)$$

Equation (19) is the major hydrostatic statement, and it allows one to eliminate  $p$  from the model equations. Furthermore, taking the  $x$ -derivative of (19) and substituting into (7), then using the viscosity form (17), leads to this equation:

$$(4\nu u_x)_x + (\nu(u_z + w_x))_z = \rho gh_x \quad \text{hydrostatic stress balance} \quad (20)$$

The hydrostatic stress balance equation (20) is nontrivially-coupled to incompressibility (6) because derivatives of the vertical velocity  $w$  appear in both equations, though  $p$  is gone. Nonetheless coupled equations (6) and (20), along with the formula  $\nu = (1/2)A^{-1/3}|Du|^{-2/3}$  and appropriate boundary conditions, determine  $u$  and  $w$ .

If we drop  $w_x$  from equation (20) then we get the Blatter-Pattyn model

$$(4\nu u_x)_x + (\nu u_z)_z = \rho gh_x \quad \text{Blatter-Pattyn stress balance} \quad (21)$$

Using this equation one can solve first for the horizontal velocity  $u$  and then afterward recover  $w$  from (6); stress balance and incompressibility are decoupled.

### 3. SHALLOW ICE SHEETS

Ice sheets have four outstanding properties as fluids. They are (i) slow, (ii) shallow, (iii) non-Newtonian, and (iv) they experience some contact slip (basal sliding). The first ice flow model we consider, the non-sliding, isothermal *shallow ice approximation* (SIA), accounts for (i)–(iii).

Regarding the property of shallowness, Figure 4 shows both a no-vertical-exaggeration cross-section of Greenland at  $71^\circ$ , as well as the standard vertically-exaggerated version

which is more familiar in the glaciological literature. Ice sheets are shallow, though of course the portion of an ice sheet which you want to model may not be.

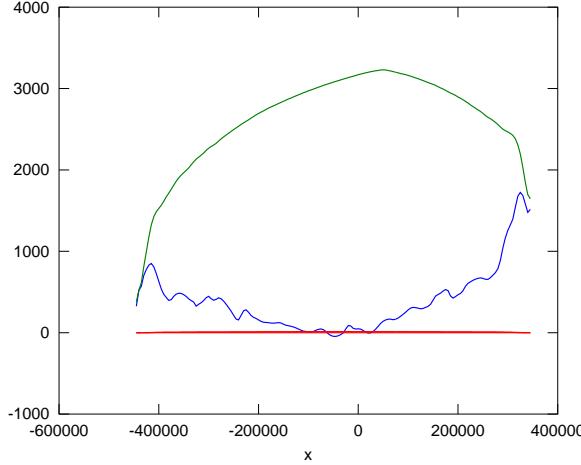


FIGURE 4. A vertically-exaggerated cross-section of the Greenland ice sheet (71° N) is shown by the green and blue curves. Without exaggeration it appears as nearly a horizontal line (red).

Our slab-on-a-slope example gives us a rough explanation of the SIA. To show the SIA in its plane-flow form, we vertically integrate velocity formula (14) in the  $u_0 = 0$  (non-sliding) case to get

$$\bar{u}H = \int_0^H \frac{1}{2}A(\rho g \sin \alpha)^3 (H^4 - (H-z)^4) dz = \frac{2}{5}A(\rho g \sin \alpha)^3 H^5. \quad (22)$$

Note  $\sin \alpha \approx \tan \alpha = -h_x$ . Combining these statements with mass continuity (16) gives

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

Equation (23) is the SIA equation for nonsliding plane flow. It is a model for the evolution of an ice sheet's thickness  $H$  in terms of surface mass balance  $M$ , ice softness  $A$ , and bed elevation  $b$  (because  $h = H + b$ ).

Additional arguments are needed to demonstrate that the SIA is more general-purpose than the special case of a simple slab; see Notes and References. Such arguments reduce the Stokes equations under the assumption that the surface and bed slopes, and the depth-to-width ratio, are small.

We will numerically solve the SIA in section 5, but first we state it in two horizontal dimensions. Let  $\mathbf{U} = (u, v)$  be the vector horizontal velocity. The shear stress approximation is  $(\tau_{13}, \tau_{23}) \approx -\rho g(h - z)\nabla h$ , which appeared as “ $\tau_{13} = \rho g \sin \alpha(h - z)$  and  $\sin \alpha \approx -h_x$ ” in the previous section, becomes an equality in the SIA. 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,

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

Mass continuity in two horizontal dimensions, which generalizes the 1D version (16), also applies:

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

Equation (25) may be written  $H_t = M - \nabla \cdot \mathbf{q}$  in terms of the map-plane flux  $\mathbf{q} = \int_b^h \mathbf{U} dz = \bar{\mathbf{U}} H$ .

Combining Equations (24) and (25), we get an equation for the rate of thickness change in terms of mass balance  $M$ , thickness, and surface slope  $\nabla h$ :

$$H_t = M + \nabla \cdot (\Gamma H^{n+2} |\nabla h|^{n-1} \nabla h), \quad (26)$$

where we have defined the positive constant  $\Gamma = 2A(\rho g)^n / (n+2)$ . Equation (26) is the SIA in two dimensions. Recalling our earlier promise, if we can solve (26) numerically then we have, following Mahaffy [34], a usable model for the Barnes ice cap in Canada, a particularly simple ice sheet on a rather flat bed.

**Analogy with the heat equation.** The SIA model is easy to compare with the better-known heat equation. All numerical methods for solving (26) can be understood as modifications of well-known heat equation methods.

In the simplest one-dimensional (1D) case, the heat equation for the temperature  $T(t, x)$  of a conducting rod is

$$T_t = DT_{xx}. \quad (27)$$

This form applies when material properties are constant and there are no heat sources. The positive constant  $D$  is the “diffusivity,” with units which can be read from comparing sides of the equation:  $D \sim \text{m}^2\text{s}^{-1}$ . Observe that equation (27) has a smoothing effect on the solution  $T$  as it evolves in time, because any local maximum in the temperature is flattened (i.e.  $T_{xx} < 0$  implies  $T_t < 0$  so  $T$  decreases), while any local minimum is also flattened (i.e.  $T_{xx} > 0$  implies  $T_t > 0$  so  $T$  increases).

The 2D heat equation, analogous to equation (26), describes the temperature  $T(t, x, y)$  at position  $x, y$  and time  $t$ . Recall that Fourier’s law for conduction is the formula  $\mathbf{Q} = -\kappa \nabla T$  for heat flux  $\mathbf{Q}$ , where  $\kappa$  is conductivity. We will assume, for the purposes of our analogy, that  $\kappa(x, y)$  may vary in space. Also suppose a variable heat source  $f(t, x, y)$ , with units of Watts per cubic meter. Then conservation of internal energy says

$$\rho c T_t = f + \nabla \cdot (\kappa \nabla T). \quad (28)$$

Here  $\rho$  is density and  $c$  is specific heat capacity. Assuming  $\rho c$  is constant, define the “diffusivity”  $D = \kappa / (\rho c)$  and the rescaled source term  $F = f / (\rho c)$ . The revised 2D heat equation is

$$T_t = F + \nabla \cdot (D \nabla T). \quad (29)$$

Figure 5 shows a solution of this heat equation, where the initial condition is a localized “hot spot”. Solutions of equation (29) always involve the spreading, in all directions, of any local heat maxima or minima, that is, diffusion.

The SIA equation (26) and the heat equation (29) are each diffusive, time-evolving partial differential equations (PDEs). A side-by-side comparison is illuminating:

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

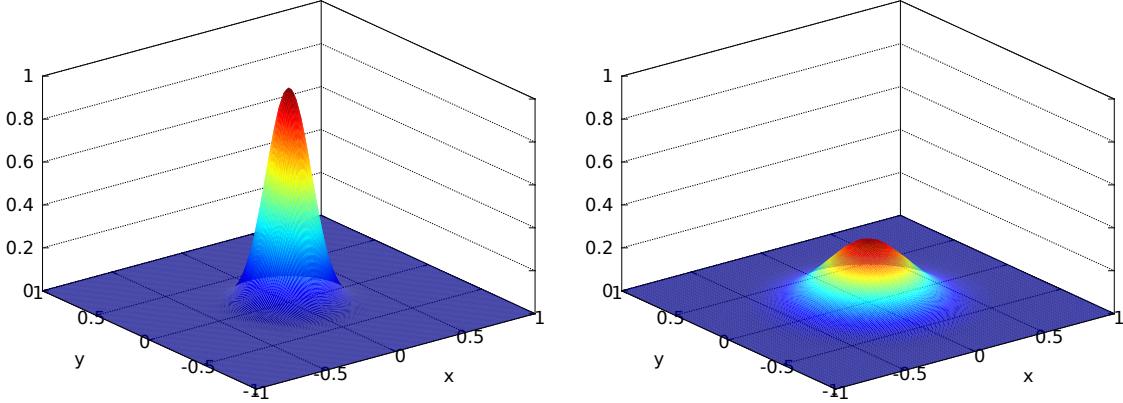


FIGURE 5. A solution of heat equation (29) with  $D = 1$  and  $F = 0$ . Left: Initial condition  $T(0, x, y)$ . Right: Solution  $T(t, x, y)$  at  $t = 0.02$ .

Notice that the number of derivatives (one time and two space derivatives) and the signs are the same. Surface mass balance  $M$  is analogous to heat source  $F$ .

The analogy suggests that we identify the *diffusivity in the SIA* as:

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

A non-sliding SIA flow diffuses the thickness of the ice sheet. When this  $D$ , a product of  $\Gamma$  and the powers of  $H$  and  $|\nabla h|$ , comes out large then the diffusion acts most quickly.

This diffusion equation analogy explains generally why the surfaces of ice sheets are smooth, at least if we overlook non-flow processes like crevassing and wind-driven (snow) dunes. There are, however, some issues with the analogy:

- The diffusivity  $D$  depends on the solution, both the thickness  $H$  and surface elevation  $h$ .
- The diffusivity  $D$  goes to zero at margins, where  $H \rightarrow 0$ , and at divides and domes, where  $|\nabla h| \rightarrow 0$ .

More important is a deficiency of the SIA model and not *per se* the analogy, namely

- Ice flow is much less diffusive when longitudinal (membrane) stresses are important, as when ice is floating or sliding or when the flow is confined by terrain.

But we will continue with the SIA, working toward a verified numerical scheme for (26) in Section 5.

#### 4. FINITE DIFFERENCE NUMERICS

Numerical schemes for the heat equation are a good starting place for solving the SIA equation (26). Here we demonstrate only finite difference (FD) schemes. These schemes replace derivatives in a differential equation by mere arithmetic.

The basic fact on which FD schemes are based is *Taylor's theorem*, which says that for a smooth function  $f(x)$ ,

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

You can replace “ $\Delta$ ” by its multiples, for example:

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

The idea for constructing FD schemes is to combine expressions like these to give approximations of derivatives. Thereby function values on a grid combine to approximate the differential equation.

Here we want partial derivative approximations, so we apply the Taylor’s expansions one variable at a time. For example, with a general function  $u = u(t, x)$ ,

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

Note that if  $\Delta$  is a small number then “ $+O(\Delta^2)$ ” is smaller than “ $+O(\Delta)$ ”, so the approximation is closer when you drop it.

**Explicit scheme for the heat equation.** We can build the simplest “explicit” scheme which approximates the 1D heat equation (27) by observing that these two FD expressions are nearly equal:

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

The FD scheme itself is not just an approximation of a PDE, like (30), but an actual method for computing numbers on a grid. Let  $(t_n, x_j)$  denote the time-space grid points. Denote our approximation of the solution value  $T(t_n, x_j)$  by  $T_j^n$ . Then the finite difference scheme is

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

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

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

FD scheme (31) is *explicit* because it directly computes  $T_j^{n+1}$  in terms of values at time  $t_n$ . Figure 6 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  $t_{n+1}$ .

Before moving on, notice that evaluating a heat equation solution at a grid point (i.e. the expression “ $T(t_n, x_j)$ ”) generally gives a different value from the value  $T_j^n$  computed by a scheme like (31). Of course we plan that these numbers will be close, but that needs checking (“verification”) or an *a priori* proof. Specifically we intend that the numbers  $T(t_n, x_j)$  and  $T_j^n$  become close to each other when the grid is made finer (i.e.  $\Delta t \rightarrow 0$  and  $\Delta x \rightarrow 0$ ), as the FD expressions become closer to the derivatives they approximate. That is, we intend our FD scheme to *converge* under *grid refinement*.

How accurate is scheme (31)? Its construction tells us that the difference between the scheme (31) and the PDE (27) is  $O(\Delta t + (\Delta x)^2)$ , so this difference goes to zero as we refine the grid in space and time, a property called *consistency*. With care about the smoothness of boundary conditions, and using mathematical facts about the heat equation itself, one can show that the difference between  $T_j^n$  and  $T(t_n, x_j)$  is also  $O(\Delta t + (\Delta x)^2)$ , which is thus the *convergence rate*; see Notes and References.

To get convergence the PDE problem must generate adequately smooth solutions, and also scheme (31) must be *stable*, which we address below. (The main theorem for numerical PDE schemes is “consistency plus stability implies convergence”; see Notes and References.) In these notes we do something rather practical, namely verification. We find problems for which we already know an exact solution  $T(t, x)$ , and then we compute the differences  $|T_j^n - T(t_n, x_j)|$ . This determines directly whether the *implementation* (i.e. computer code form) of our FD scheme actually converges, not just in theory.

**A first implemented scheme.** Our first Matlab implementation we consider the two spatial dimension Equation (29) with  $D$  constant and  $F = 0$ :

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

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

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

The stencil for the right-hand side of (33) is in Figure 6.

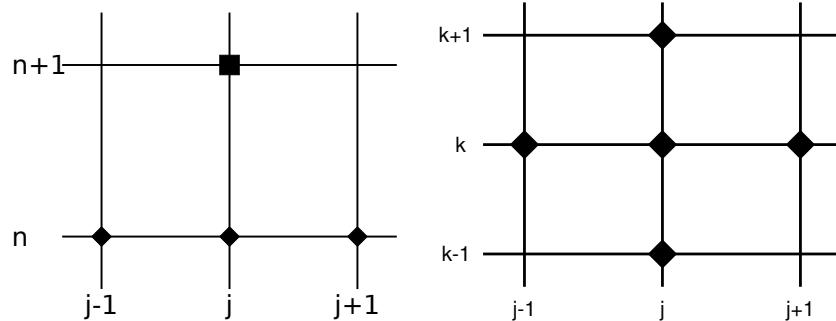


FIGURE 6. Left: Space-time stencil for the explicit scheme (31) for the 1D heat equation. Right: Spatial-only stencil for scheme (33).

Scheme (33) has implementation `heat.m` below. For simplicity we set  $T = 0$  on the boundary of the square  $-1 < x < 1$ ,  $-1 < y < 1$ . The initial condition is gaussian:  $T(0, x, y) = \exp(-30(x^2 + y^2))$ . The code uses Matlab “colon” notation to remove loops over spatial variables. Here is an example run:

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

This sets  $D = 1.0$  and uses a  $30 \times 30$  spatial grid. We take  $N = 20$  time steps of  $\Delta t = 0.001$ . The result is shown in Figure 5, right. This is the look of success.

heat.m

```

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

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

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

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

```

However, very similar runs seem to succeed or fail according to some yet unclear circumstance. For example, results from these calls are shown in Figure 7:

```

>> heat(1.0,40,40,0.0005,100); % Figure 7, left
>> heat(1.0,40,40,0.001,50); % Figure 7, 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. The second run clearly shows instability.

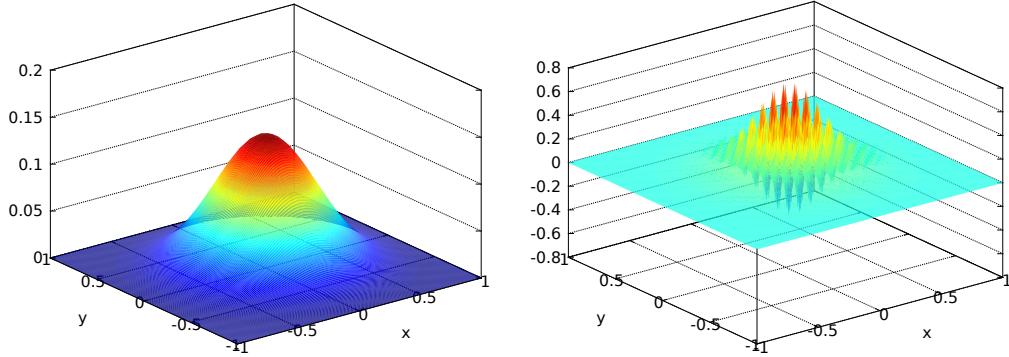


FIGURE 7. Numerically-computed temperature on  $40 \times 40$  grids. The two runs are the same except that the left has  $\Delta t = 0.0005$  so  $D\Delta t/(\Delta x)^2 = 0.2$ , while the right has  $\Delta t = 0.001$  so  $D\Delta t/(\Delta x)^2 = 0.4$ . Compare (34).

**Stability criteria and adaptive time stepping.** To avoid the instability shown at right in Figure 7, we need to understand the scheme better. It turns out we have not made an implementation error, but more care is required when choosing space and time steps.

Recall the 1D explicit scheme in form (31):  $T_j^{n+1} = \mu T_{j+1}^n + (1 - 2\mu)T_j^n + \mu T_{j-1}^n$ . The new value  $T_j^{n+1}$  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. Actually, however, the scheme is only an average *if* the middle coefficient is positive, as a

linear combination with coefficients which add to one is not an average if any coefficients are negative. (For example, we would not accept 15 as an “average” of 5 and 7, but we can write  $15 = -4 \times 5 + 5 \times 7$ , 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* follows, with these equivalent forms:

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

This condition on the size of  $\Delta t$  is a *sufficient* stability criterion; it is enough to guarantee stability, though something weaker might do. In summary, for given  $\Delta x$ , shortening the time steps  $\Delta t$  so that (34) holds will make FD scheme (31) into an 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 like those shown in Figure 7 with  $\Delta x = \Delta y$ , this condition requires  $D\Delta t/(\Delta x^2) \leq 0.25$ , which precisely distinguishes between the two parts of the Figure. Runs of `heat.m` are unstable if the time step  $\Delta t$  is too big relative to the spacing  $\Delta x$ .

The stability criterion above is easily satisfied by making each time step shorter. Doing so at each time step makes an *adaptive* implementation which can be stable even if the diffusivity  $D$  is changing in time. To show how easy this is to implement, `heatadapt.m` (not shown) is the same as `heat.m` except that the time step comes from the stability criterion, so it cannot generate the instability seen in Figure 7. However, if the diffusivity  $D$  is large or the grid spacings  $\Delta x, \Delta y$  are small, then adaptive explicit implementations must take many short time steps to assure stability.

**Implicit schemes.** There is an alternative stability fix instead of adaptivity, namely “implicitness.” For example, the finite difference scheme

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

is an  $O(\Delta t + (\Delta x)^2)$  implicit scheme for Equation (27). Such implicit schemes for the heat equation are 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 values at time step  $t_{n+1}$  are treated as a vector in a large system of equations which must be formed and solved at each time step. If the PDE is nonlinear then the system of equations may be hard to solve. Of course, the SIA is a highly nonlinear diffusion equation.

Generally, there is a tradeoff between the easy implementability of adaptive explicit schemes and the better stability of implicit schemes. For these notes we stay with adaptive explicit FD schemes.

**Numerical solution of diffusion equations.** We are trying to numerically model ice flows, not heat conduction. We have an analogy, however, which says that the SIA is diffusive like the heat equation. In this section, because we wish to solve the SIA on real bedrock, we construct a numerical scheme for a more general diffusion equation which has an extra “shift” inside the gradient, namely

$$T_t = F + \nabla \cdot (D\nabla(T + b)). \quad (36)$$

In equation (36), the source term  $F(x, y)$ , the diffusivity  $D(x, y)$ , and the “shift”  $b(x, y)$  may all vary in space.

The following code solves (36). It is called by the SIA-specific schemes we build next.

```

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;
```

This adaptive explicit method for the diffusion equation 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. That is, we use this expression for the second derivative:

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

where  $X = T + b$ . The left part of Figure 8 shows the stencil.

The user supplies the diffusivity  $D(x, y)$  to `diffusion.m` on the staggered grid. The initial temperature  $T(0, x, y)$ , source term  $F(x, y)$ , and “shift”  $b(x, y)$  are also supplied on the regular grid. When using this code for standard diffusions, or for the flat-bed case of the SIA, we would take  $b = 0$ .

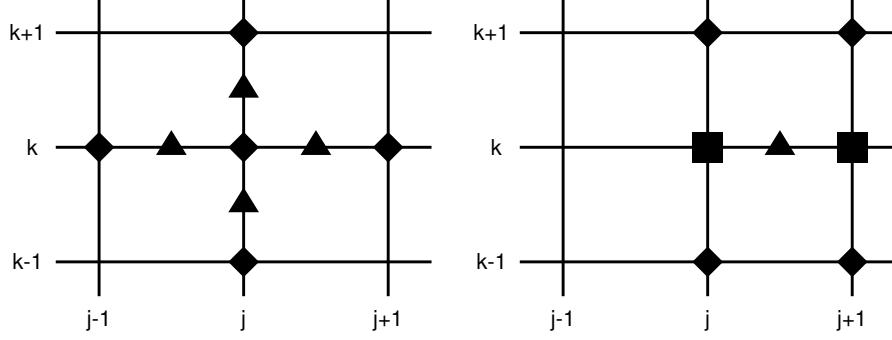


FIGURE 8. Left: Spatial stencil for staggered grid evaluation of diffusivity (at triangles) in the diffusion equation (36). Right: Stencil showing how the staggered-grid diffusivity (triangle) can be evaluated in the SIA, from surface elevation (diamonds) and thicknesses (squares).

## 5. NUMERICALLY SOLVING THE SIA

In SIA equation (26) we have diffusivity  $D = \Gamma H^{n+2} |\nabla h|^{n-1}$ . There are two interesting aspects of this formula. First, as already noted,  $D$  goes to zero, i.e. it “degenerates,” when either  $H \rightarrow 0$  or  $\nabla h \rightarrow 0$ . Degenerate diffusion equations are automatically free boundary problems, but this aspect of the thickness evolution problem is no surprise to a glaciologist. Determining the location of the margin is an obvious part of modelling a glacier or ice sheet. To address this free boundary issue in our explicit time-stepping code it suffices to numerically compute new thicknesses and then set them to zero if they come out negative.

Second, for numerical stability and mass conservation we should compute  $D$  on a “staggered” grid. Various finite difference schemes for computing it have been proposed. All of these schemes involve averaging  $H$  and differencing  $h$  in a “balanced” way onto the staggered grid. In the code `siaflat.m` below we use the Mahaffy [34] method, with the stencil for computing  $D$  shown in Figure 8. This code only works for the flat bed, zero surface mass balance case, but we will correct these deficiencies later.

*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) );
    % ... (rest of the code)
end

```

```

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 fairly complicated code. How do we make sure that such an implemented numerical scheme is correct? Here are three proposed techniques:

- (1) don't make any mistakes, or
- (2) compare your numerical results with results from other researchers, and hope that the outliers are in error, or
- (3) compare your numerical results to an exact solution.

The last one of these, which we prefer to the first two when possible, is called “verification.” That is, when we build a new computer code we should test it in cases where we know the right answer. To do so we need to return to the PDE itself, to get useful exact solutions.

**Exact solution of heat equation.** First we consider the simpler case of the 1D heat equation with constant  $D$ , namely  $T_t = DT_{xx}$ . Many exact solutions  $T(t, x)$  to this heat equation are known, but let's consider the time-dependent “Green's function,” 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  $x = 0$ . Then it spreads out over time. It is a solution of the heat equation on the whole line  $-\infty < x < \infty$  and for all  $t > 0$ .

We will calculate this exact solution by a method which generalizes to the SIA. The Green's function of the heat equation 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, as shown in Figure 9. These scalings are related to each other by conservation of energy, which says that the total heat energy is independent of time.

In particular, the Green's function of the 1D heat equation is

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

“Similarity” variables for this solution, the above-mentioned scalings, involving multiplying the input and output of an invariant shape function  $\phi(s) = (4\pi D)^{-1/2} e^{-s^2/(4D)}$  by the same power of  $t$ :

<i>input scaling</i>	<i>output scaling</i>	
$s = t^{-1/2}x$	$T(t, x) = t^{-1/2}\phi(s)$	(37)

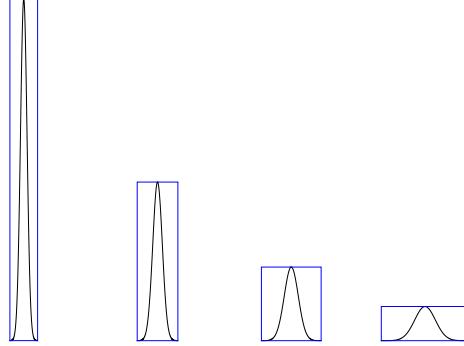


FIGURE 9. The heat equation Green's function in 1D has the same shape at each time, but with time-dependent input- and output-scalings.

Note that all time-dependence is from the input and output scalings.

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

**Halfar's similarity solution to the SIA.** Now we jump from Green's idea in about 1830 to the year 1981. That is when P. Halfar published the similarity solution of the SIA in the case of flat bed and zero surface mass balance. Halfar's solution to the 2D SIA model (26), using a Glen exponent  $n = 3$ , has scalings which are powers of  $t$  like (37) above:

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

Here  $r = (x^2 + y^2)^{1/2}$  is the distance from the origin. These scalings are related to each other by conservation of mass, because no mass is gained or lost through the surface. Scalings (38) imply that, quite differently from heat, the diffusion of ice slows down severely as the shape flattens out. The powers  $t^{-1/9}$  and  $t^{-1/18}$  change very slowly for large times  $t$ .

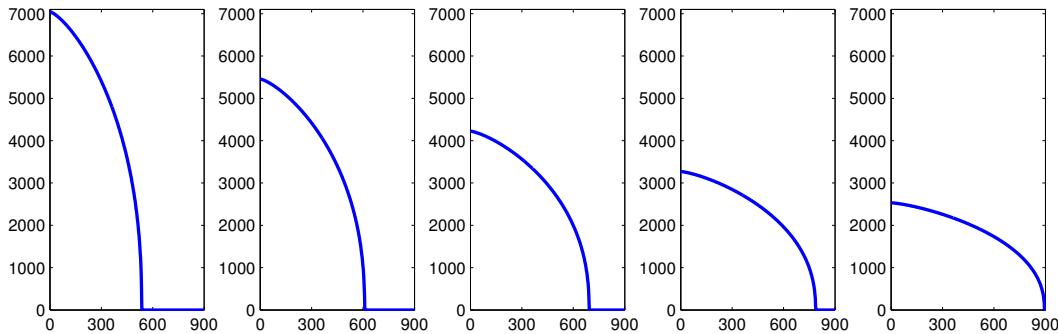


FIGURE 10. A case of Halfar's solution (39) of the SIA equation (26) on a flat bed with zero mass balance. The solution is shown on  $H$  (m) versus  $r$  (km) axes for times  $t = 1, 10, 100, 1000, 10000$  years.

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

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

Here the “characteristic time”  $t_0 = (18\Gamma)^{-1}(7/4)^3 R_0^4 H_0^{-7}$  is a parameter which can be determined by choosing center height  $H_0$  and radius  $R_0$ .

Formula (39) is plotted in Figure 10. We see that for times significantly greater than  $t_0$  (i.e.  $t/t_0 \gg 1$ ) the solution changes very slowly. For example, the change between years 1 and 100 is larger than that between years 1000 and 10000. The *volume* of ice in this Halfar ice cap is, however, constant with  $t$ ; see the Exercises.

**Using Halfar’s solution.** Formula (39) is simple enough 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` above to a final time 20000 a, and then compares to the Halfar formula for that time. By “compares” we mean it computes the thickness error, the absolute values of the differences between the numerical and exact thickness solutions 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. This is as expected for a correctly-implemented code. What is less obvious, perhaps, is that almost any numerical implementation mistake—almost any bug—will break this property, and these errors will not shrink.

You might ask, is the Halfar solution ever useful for modelling real ice masses? The answer is yes. In fact, J. Nye and others (2000; [38]) compared the long-time consequences of different flow laws for the south polar cap on Mars. In particular, they evaluated CO<sub>2</sub> ice and H<sub>2</sub>O 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<sub>2</sub>] flow laws will allow a 3000-m cap, the thickness suggested by stereogrammetry, to survive for 10<sup>7</sup> years, indicating that the south polar ice cap is probably not composed of pure CO<sub>2</sub> ice [but rather] water ice, with an unknown admixture of dust.

This theoretical result has since been confirmed by the observation and sampling of the polar geology of Mars.

Are exact solutions like Halfar’s always available when needed? The answer is “no”, of course, though many ice flow models do have exact solutions which are relevant to verification; see the Notes and References. For example, we will use van der Veen’s solution

for ice shelves in a later section. On the other hand, the absence of exact solutions may show that not enough thought has gone into the continuum model itself.

**A test of robustness.** Verification is an ideal way to start testing a code. Another kind of test is for “robustness”. One asks: Does the model break when you ask it to do hard things? Unlike for verification, we might not have precise knowledge of what it should do, but a well-implemented model should act in a “reasonable” way.

The robustness test in the program `roughice.m` (not shown) demonstrates that `siaflat.m` can handle an ice sheet with extraordinarily large “driving stresses.” Recall that glaciological driving stress is  $\tau_d = -\rho g H \nabla h$ . This quantity appears in the slab-on-a-slope example, and thus in the SIA model, as the value of the shear stress  $(\tau_{13}, \tau_{23})$  at the base of the ice. The driving stress is, obviously, large when the ice is both thick and has steep surface slope  $|\nabla h|$ .

In `roughice.m` we give `siaflat.m` a randomly-generated initial ice sheet which is of the worst possible sort because it is both thick (average of 3000 m) and it has large surface slopes. The initial shape is shown in the left side of Figure 11. During the run of 50 model years, the time step is determined adaptively from (34), increasing from 0.0002 years to about 0.2 years as the maximum diffusivity  $D$  decreases correspondingly. The maximum value of the driving stress decreases from 57 bar ( $= 5.7 \times 10^6$  Pa) to 3.6 bar. At the end the ice cap has the shape shown at right in Figure 11.

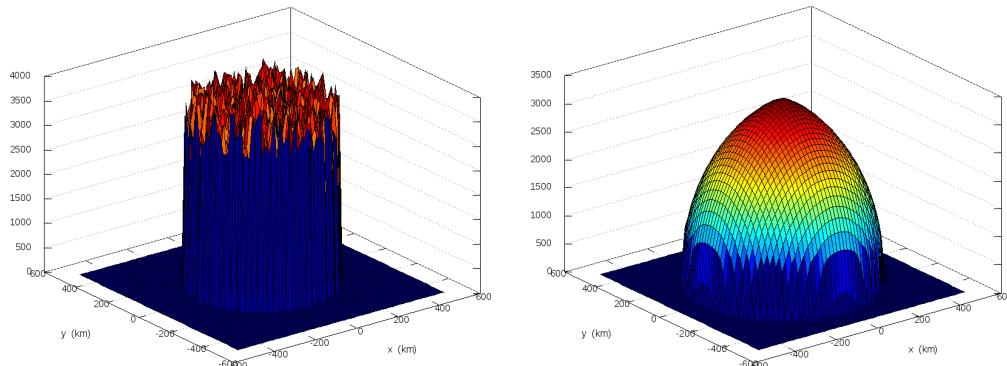


FIGURE 11. The SIA model evolves the huge-driving-stress initial ice sheet at left to the ice cap at right in only 50 model years.

The shape at right in Figure 11 is rather close to a Halfar solution. Indeed Halfar proved that all solutions of the zero-mass-balance SIA on a flat bed asymptotically approach the Halfar solution.

## 7. APPLYING OUR NUMERICAL ICE SHEET MODEL

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 surface mass balance  $M(x, y)$ . Also we calve floating ice, and we enforce non-negative thickness at each timestep. The result is `siageneral.m` (not shown), a code only ten lines longer than `siaflat.m`.

We use measured accumulation, bedrock elevation, and surface elevation from ALBMAPv1 data [28]. Melt is not modelled so the surface mass balance is the accumulation rate.

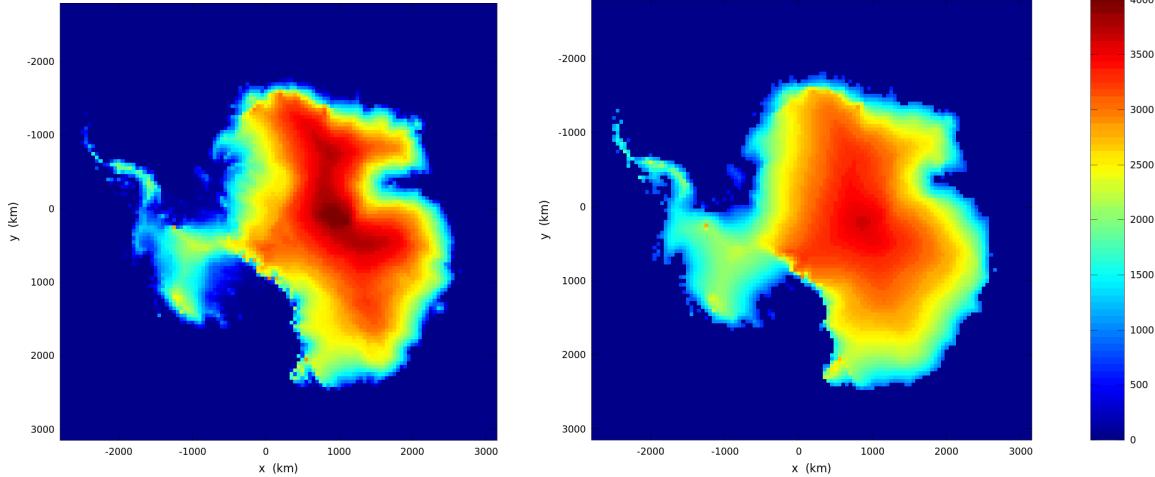


FIGURE 12. Left: Initial surface elevation (m) of Antarctic ice sheet. Right: Final surface elevation at end of 40 ka model run on 50 km grid.

These input data are read from a NetCDF file and preprocessed by an additional code `buildant.m` (not shown).

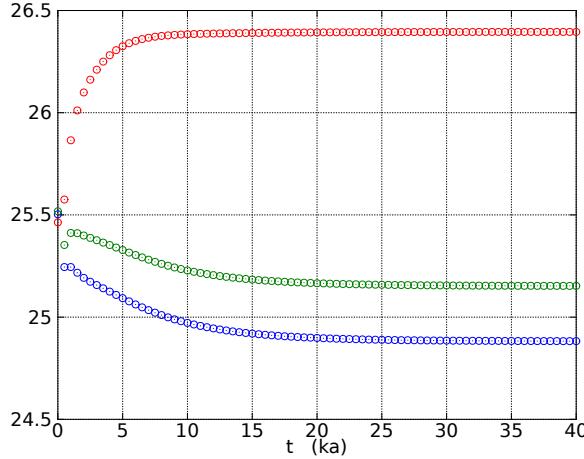


FIGURE 13. Ice volume of the modeled Antarctic ice sheet, in units of  $10^6 \text{ km}^3$ , from runs on 50 km (red), 25 km (green), and 20 km (blue) grids.

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 12 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 Antarctic ice sheet with low-slope and (actual) 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 near steady state, these most-obvious 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, explains the thinning near the divide. And of course we should be modeling floating ice too, but the SIA is completely inappropriate to that

purpose. See section 9 and Notes and References on modelling techniques which address these inadequacies.

Figure 13 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, and this is true to differing degrees at these coarse resolutions. Mainly Figure 13 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. INTERLUDE: MASS CONTINUITY AND KINEMATICAL EQUATIONS

Recall that in the SIA the “stress balance” is essentially formula (24) for the velocity. It combines with the mass continuity equation (25) to give model (26) for the ice sheet thickness. The major SIA equation (26) thus combines two concepts which we will now think about separately, and in greater generality, in the remainder of these notes.

The basic shallow assumption made by most ice flow theories<sup>1</sup> is that the surface and base of the ice are differentiable functions  $z = h(t, x, y)$  and  $z = b(t, x, y)$ . Thus surface overhang is not allowed, though, by contrast, the Stokes theory of slow viscous fluids only needs a closed surface in three-dimensional space as a boundary surface for the fluid. Most ice sheet and glacier models take a map-plane perspective, however, and they have a well-defined ice thickness:  $H = h - b$ .

To pursue such ideas a bit further, let us state the “kinematical equations” which apply at upper and lower surfaces of the ice sheet. Let  $a$  be the upper surface (climatic) mass balance function ( $a > 0$  is accumulation) and  $s$  be the basal melt rate function ( $s > 0$  is basal melting). In the equations which follow these are measured in thickness-per-time units, but they could be in mass-per-area-per-time units also. The net map-plane mass balance  $M = a - s$ , which already appears in the mass continuity equation (25), is the difference of these surface fluxes.

The *(upper) surface kinematical equation* is

$$h_t = a - \mathbf{U}|_h \cdot \nabla h + w|_h, \quad (40)$$

and the *base kinematical equation* is

$$b_t = s - \mathbf{U}|_b \cdot \nabla b + w|_b. \quad (41)$$

(Recall  $\mathbf{U}$  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.

We can now state an important mathematical fact which follows merely from the assumption of well-defined upper and basal surface elevations. Namely, that the surface kinematical and mass continuity equations are closely-related. More precisely, any pair of these equations implies the third:

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

---

<sup>1</sup>There are several inequivalent shallow theories: SIA, SSA, hybrids, Blatter-Pattyn, ...

One proves these facts by using the incompressibility of ice (1) and the Leibniz rule for differentiating integrals. The details are left for exercises.

The bedrock is often regarded as fixed (i.e.  $b_t = 0$ ), and in fact the basal kinematical equation is often not explicitly mentioned. Instead one gets a simplified view. In the case of non-deformable bedrock and no sliding, for example, the basal value of the vertical velocity equals the basal melt rate. This simplification corresponds to  $b_t = 0$  and  $\mathbf{U}|_b = 0$  so that  $w|_b = -s$  from (41).

**Prognostic models.** We can now sketch the structure of a general “prognostic,” i.e. ice geometry evolving, isothermal ice sheet model. Each time step follows this recipe:

- numerically solve a stress balance, which gives velocity  $\mathbf{u} = (u, v, w)$ ,
  - if the stress balance only gives  $\mathbf{U} = (u, v)$ , get  $w$  from incompressibility (1),
- decide on a time step  $\Delta t$  for (25) based on velocities and/or diffusivities,
- from the horizontal velocity  $\mathbf{U} = (u, v)$ , compute the flux  $\mathbf{q} = \bar{\mathbf{U}}H$ ,
- 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.

Like most ice sheet models, we use the mass continuity equation (25) to describe changes in ice sheet geometry, but we could use the surface kinematical equation (40) instead.

The above “standard” ice sheet model has many variations. Some glaciological questions are answered just by solving the stress balance for the velocity. Sometimes the goal is the steady state configuration of the glacier, which might be computed more quickly by iteratively solving steady state equations than by time-stepping physical evolution equations to steady state. Other processes are usually 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 usually an important step in designing the coupled model.

When using the SIA equation (26), one can seemingly bypass the computation of the velocity. That is because we could write the mass continuity equation as a diffusion, with  $\mathbf{q} = -D\nabla h$  for the flux instead of the more general  $\mathbf{q} = \bar{\mathbf{U}}H$ . 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, and so only “ $\mathbf{q} = \bar{\mathbf{U}}H$ ” applies. Solving the stress balance for the velocity field is then an obligatory, and usually nontrivial, step. We consider such a stress balance next.

## 9. SHELVES AND STREAMS

The shallow shelf approximation (SSA) stress balance applies to ice shelves as its name suggests. The SSA also applies reasonably well to ice streams, like those in Figure 14 which have not-too-steep bed topography and low basal resistance.

But what is, and is not, an ice stream? Ice streams slide at 50 to 1000 m a<sup>-1</sup>, 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. There are other fast-flowing grounded parts of ice sheets, however, called “outlet glaciers”. They can have even faster surface speed (up to 10 km a<sup>-1</sup>), but it is typically uncertain how much of this speed is from sliding at the base. In an outlet glacier there is substantial vertical shear “up” in the ice column, sometimes caused by soft temperate

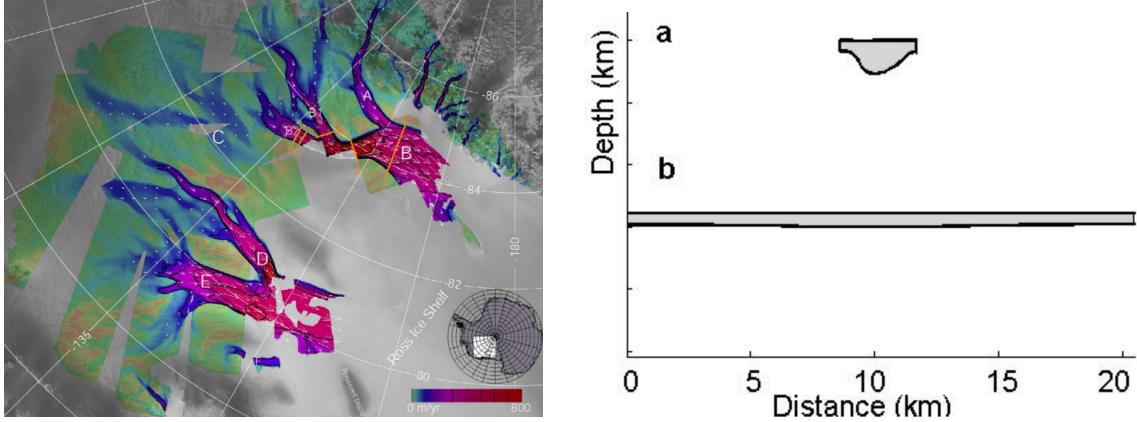


FIGURE 14. Left: The SSA model applies to ice streams like these on the Siple Coast in Antarctica. Color shows radar-derived surface speed. Right: Cross sections, *without* vertical exaggeration, of the Jakobshavn Isbrae outlet glacier in Greenland (a) and the Whillans Ice Stream on the Siple Coast (b); this is Figure 1 in [55].

ice in a significant fraction of the thickness. Furthermore, outlet glaciers are strongly controlled by fjord-like, high slope bedrock topography. Figure 14 (right) compares the shallowness and bedrock topography of an outlet glacier and an ice stream. Thus, few simplifying assumptions are appropriate for outlet glaciers, and the SSA may not be a sufficient model.

**The shallow shelf approximation (SSA).** We state this stress balance equation only in the plane flow (“flow-line”) case:

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

The term in parentheses is the vertically-integrated longitudinal stress, also called the “membrane” stress when there are two horizontal variables. The second term  $\tau_b = -C|u|^{m-1}u$  is the basal resistance, which is zero (i.e.  $C = 0$ ) in an ice shelf. The term on the right is the driving stress ( $\tau_d = -\rho g H h_x$ ). Thus the SSA equation is a balance wherein longitudinal strain rates are determined by the integrated ice hardness (i.e. the coefficient  $BH$ ), the slipperiness of the bed (i.e. by the coefficient  $C$  and the power  $m$ ) and the geometry of the ice sheet (i.e. the thickness  $H$  and the surface slope  $h_x$ ).

In (42) the velocity  $u$  is independent of the vertical coordinate  $z$ . We assume that the ice hardness  $B = A^{-1/n}$  is also independent of depth. Models which are not isothermal compute the vertical average of the temperature-dependent hardness. The formula for the basal resistance  $\tau_b$  is often called a “sliding law” in power law form.

The coefficient  $\bar{\nu} = B|u_x|^{1/n-1}$  in (42) is called the “effective viscosity”, so that (42) can be written

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

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

The inequality “ $\rho H < -\rho_w b$ ” is sometimes called the *floatation criterion*. For grounded ice we know  $\rho H > -\rho_w b$  and the driving stress  $\tau_d = -\rho g H h_x$  uses  $h = H + b$ . On the

floating side we know  $\rho H < -\rho_w b$  and, by Archimedes principle, we use  $h = (1 - \rho/\rho_w)H$  in the driving stress.

Equation (43) simplifies if the ice is floating. The ice surface elevation is proportional to the thickness if the ice is floating. Also we assume zero resistance ( $C = 0$ ) is applied by the ocean. Thus the SSA becomes

$$(2\bar{\nu} Hu_x)_x = \rho g(1 - \rho/\rho_w)HH_x \quad (44)$$

for floating ice. A useful observation about flow line equation (44) is that both left- and right-hand expressions are derivatives; this can be used to build a 1D exact solution.

**Steady ice shelf exact solution.** For a steady 1D ice shelf, in which  $H_t = 0$ , the mass continuity equation (25) reduces to  $M = (uH)_x$ . Because of the relative simplicity of the SSA equation (44) and the steady mass continuity equation for 1D floating ice, the exact velocity and thickness for a steady ice shelf can be computed [57]. This exact solution depends on the ice thickness  $H_g$  and velocity  $u_g$  at the grounding line. For the surface mass balance  $M$  we choose a positive constant  $M_0$ . These choices determine a unique solution, the derivation of which is left to the exercises.

Supposing  $H_g = 500$  m,  $u_g = 50$  m a $^{-1}$ , and  $M_0 = 30$  cm a $^{-1}$  we get the results in Figure 15, which are from code `exactshelf.m` (not shown). We will use this exact solution to verify a numerical SSA code. Note that driving stresses are much higher near the grounding line than away from it, and thus the highest longitudinal stresses, strain rates, and thinning rates occur near the grounding line.

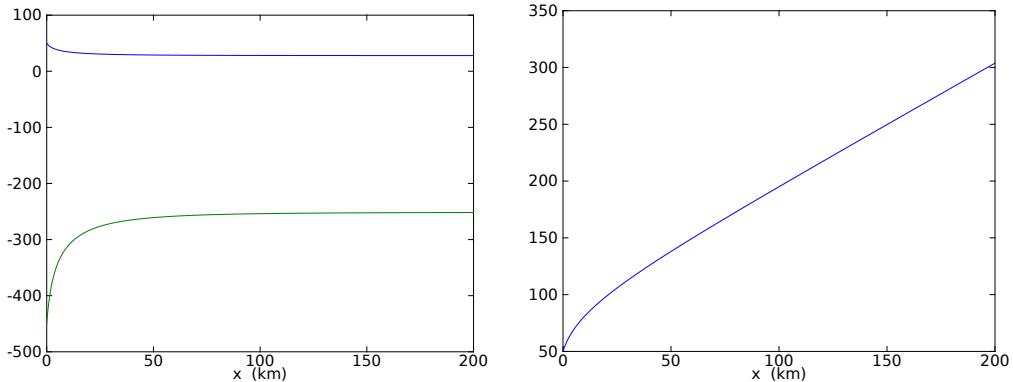


FIGURE 15. The upper and lower surface elevation (m; left) of the exact ice shelf solution and its velocity (m/a; right);  $x = 0$  is the grounding line.

**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 the unknown  $u(x)$ . When we do this numerically an iteration is needed because of the nonlinearity. The simplest iteration idea is to use an initial guess at the velocity, which allows us to compute an effective viscosity and then get a new velocity solution from a linear PDE problem. Then we recompute the effective viscosity, solve for a new velocity, and repeat until things stop changing. This is often called a “Picard” iteration, in contrast to a “Newton” iteration which should converge faster.

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 this linear elliptic PDE for the unknown  $u^{(k)}$  is a Picard iteration for (43):

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

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

Equation (45) is a linear boundary value problem. We can write it abstractly

$$(W(x)u_x)_x - \alpha(x)u = \beta(x) \quad (46)$$

where the functions  $W(x)$ ,  $\alpha(x)$ ,  $\beta(x)$  are known. Equation (46) applies on an interval of the  $x$ -axis. For one boundary condition we will suppose that  $x = x_g$  is a location where the velocity is known,  $u(x_g) = u_g$ , as in Figure 15. In the ice shelf case we also have the calving front condition (see Notes and References)

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

at the end of the ice shelf  $x = x_c$ . Boundary condition (47) can be solved for  $u_x(x_c) = \gamma$  in terms of known quantities including the thickness at the calving front.

Where to get an initial guess  $u^{(0)}$ ? Generally this may require effort, but we will use these choices for our 1D case. For floating ice, an initial velocity comes from assuming a uniform strain rate provided by the calving front condition:  $u^{(0)}(x) = \gamma(x - x_g) + u_g$ . For grounded ice, we may assume ice is held by basal resistance only:  $u^{(0)}(x) = (-C^{-1}\rho g H h_x)^{1/m}$ .

**Numerics of the linear boundary value problem.** Suppose equation (46) applies on  $[x_g, x_c] = [0, L]$ . We choose a grid with equal spacing  $\Delta x$  and index  $j = 1, 2, \dots, J+1$ , so that  $x_1 = 0$  and  $x_{J+1} = L$  are endpoints. The coefficient  $W(x)$  is needed on a staggered grid, for stability and accuracy reasons similar to those for the SIA diffusivity. Our finite difference approximation of (46) is, therefore,

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

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 “ $u_x(x_c) = \gamma$ .” Using equation (48) in  $j = J+1$  case, eliminate the  $u_{J+2}$  variable “by-hand”. This determines the form of the last equation in our linear system.

Now observe that each iteration to solve the SSA stress balance has the form

$$A\mathbf{v} = \mathbf{b}. \quad (49)$$

Indeed, at each location  $x_1, \dots, x_{J+1}$  we can write an equation, including a row of the matrix  $A$  in (49), involving the unknown velocities. It is this linear system of  $J+1$

equations:

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

The diagonal entries “ $A_{ij}$ ” are

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

except for special cases for the coefficients in the last equation,

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

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

System (50) is a tridiagonal linear system. But don’t bother looking up how to solve such a linear system unless you really need to! It is fully appropriate to give system (50) to Matlab’s linear solver, the “backslash” operator  $\mathbf{v} = A \setminus \mathbf{b}$ , especially at this initial implementation stage. In these notes we will not worry further about solving finite linear systems. We now have a code to solve (46) by finite differences and linear algebra, namely `flowline.m` below.

`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. In fact, results from `testflowline.m` (not shown) demonstrate that our implemented numerical scheme converges at the optimal rate  $O(\Delta x^2)$ .

**Solving the stress balance for an ice shelf.** The code `ssaflowline.m` (below) numerically computes the velocity for an ice shelf. The thickness is assumed to be given, so we are not yet addressing the full, “coupled” ice shelf problem, simultaneously solving the applicable mass continuity (16) and stress balance (44) equations. We are only solving the latter.

This code implements Picard iteration (45), in the floating case, to solve the nonlinear equation (44). It calls `ssainit.m` (not shown) to get the initial iterate  $u^{(0)}(x)$ , as already described, and it calls `flowline.m` at each iteration. It also calls small helper functions `stagav()`, `regslope()`, `stagslope()`, at the end of the code, to compute certain gridded values.

<i>ssaflowline.m</i>
----------------------

```

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

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

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

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

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

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

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

function fav = stagav(f)
fav = 0.5 * (f(1:end-1) + f(2:end));

```

```

function slope = regslope(dx,f)
J = length(f) - 1;
slope = [(f(2)-f(1))/dx; (f(3:J+1)-f(1:J-1))/(2*dx); (f(J+1)-f(J))/dx];

function slope = stagslope(dx,f)
slope = (f(2:end) - f(1:end-1)) / dx;

```

Now we can ask precisely: Does `ssaflowline.m` work correctly? The exact velocity solution shown in Figure 15, computed by `exactshelf.m`, allows us to compare the numerical to the exact velocities by finding the maximum difference between them. For this to work we take the exact thickness shown in Figure 15, also from `exactshelf.m`. A convergence comparison, shown in Figure 16, is done by codes `testshelf.m` and `shelfconv.m` (not shown). Each circle in the Figure gives the maximum velocity error on a given grid.

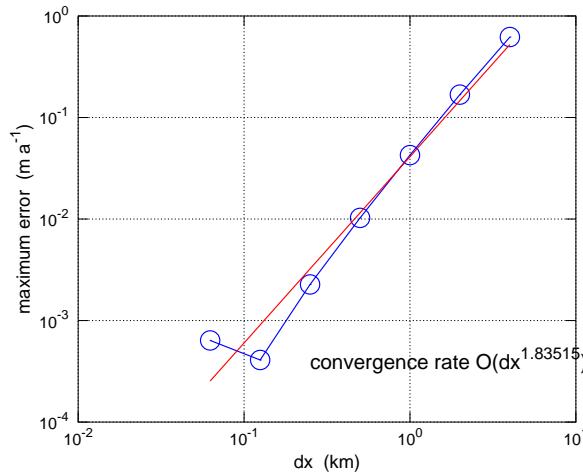


FIGURE 16. The numerical SSA velocity solution from `ssaflowline.m` converges to the exact solution, at nearly the optimal rate  $O(\Delta x^2)$ , as the grid is refined from spacing  $\Delta x = 4$  km to  $\Delta x = 62$  m.

Even on the coarsest  $\Delta x = 4$  km grid we see in Figure 16 that the maximum velocity error (i.e. difference) is less than 1 m/a, while the maximum velocity itself is  $\sim 300$  m/a. We can conclude from this comparison that, at screen resolution, our numerical velocity solutions are essentially identical to that shown in the right part of Figure 15. There is not even a reason to show the numerical solutions!

**Realistic ice shelf modelling.** Real ice shelves have two horizontal variables. They are frequently confined in bays, and thus they experience “side drag”. Their velocities vary spatially and temporally along their grounding lines, which are the curves where the flotation criterion is an equality. Furthermore real ice shelves have interesting boundary processes, including high basal melt near grounding lines, marine ice basal freeze-on, and fracturing which nears full thickness at the calving front. It is a bit complicated.

Nonetheless “diagnostic” (i.e. fixed geometry) ice shelf modelling in two horizontal variables, done like the above example where the velocity is unknown but the thickness is known and fixed, is quite successful using only the isothermal SSA model. For example,

Figure 17 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  $B$ . In this run, observed velocities for grounded ice were applied as boundary conditions. Many current ice shelf models yield comparable match [33].

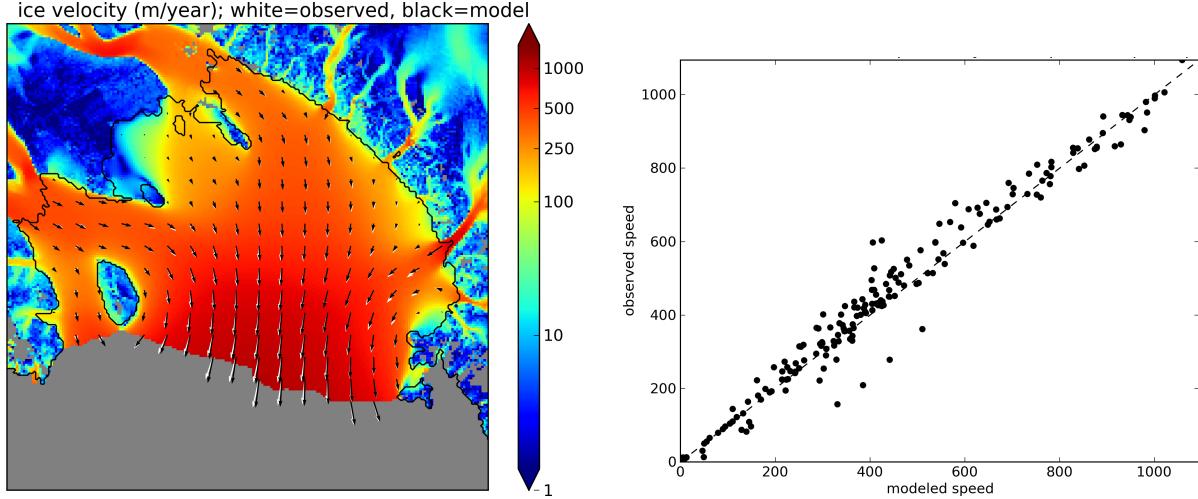


FIGURE 17. Results from PISM. Left: Observed (white) and modeled (black) ice velocities are nearly coincident across the whole Ross ice shelf. The grounding line is the thin black curve. Right: In this scatter plot there is one point for each arrow at left.

## 10. A SUMMARY OF NUMERICAL ICE SHEET MODELLING

These notes are brief, and so they give a very incomplete view of numerical models for glaciers and ice sheets. They do, however, illustrate some general principles about numerical modelling. One should:

- Return often to the continuum model.
- Modularize codes.
- Test the parts: Is the component robust? Does it show convergence?

Regarding the specific ice flow models covered in these notes, here are three high-level points, as a meager conclusion:

- The mass continuity equation is the part of an ice sheet model which describes how the ice geometry evolves. It is a kind of transport equation in the map-plane, but with diffusive character at larger spatial scales. The numerical approach to this equation depends on which is the stress balance which supplies the ice velocity or ice flux. Mass continuity is a diffusion for frozen bed, large scale flows, and in that case the SIA is a good choice. Mass continuity is *not* very diffusive for membrane stresses (e.g. SSA), especially with no basal resistance as in ice shelves. It has some diffusiveness for ice streams, though how much is hard to quantify.
- The SIA stress balance is exceptional because it is not horizontally-distributed. In the SIA, velocity follows immediately by vertical integration of the driving stress.
- Membrane stress balance equations like the SSA (and the Blatter-Pattyn, hydrostatic, and Stokes models also) determine horizontal velocity from geometry and

boundary conditions. Because of the Glen law these equations are nonlinear, so iteration is necessary. At each iteration a sparse matrix “inner” problem is solved; non-experts should give this matrix problem to a solver package.

## 11. NOTES

Recent and recommended books and reviews which extend the continuum modeling content of these notes include [12, 19, 52, 58].

The SIA model, which was derived by several authors [14, 24, 36], 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  $\epsilon$  is small [13, 24]; this is a “small-parameter argument”. In one scaling there are no  $O(\epsilon)$  terms in the scaled equations so one only drops  $O(\epsilon^2)$  terms [13]. The SIA is re-formulated as a well-posed free boundary problem in [25], which provides the correct boundary condition at grounded margins. The Mahaffy [34] scheme for diffusivity used here is not the only one [22].

The SSA model [59] was derived in [35] for ice shelves and in [32] for ice streams. In deriving the SSA, the aspect ratio  $\epsilon$  above is one small parameter but additionally a second parameter describing the magnitude of surface undulations must be assumed to be small [49, 53]. A well-posed model for the emergence of ice streams though till failure, using only the SSA, is in [49].

A key modelling issue omitted in these notes is thermomechanical coupling. Temperature is important because the ice softness varies by three orders of magnitude in the temperature range relevant to ice sheet modelling. Ice temperature therefore gives ice sheet dynamics a long memory of past climate, and because the geothermal flux is a significant input in slow-flowing parts of ice sheets. Equally important, dissipation of gravitational potential energy is a major part of the energy balance, and basal melt in particular. For example, each year the ice in the Jakobshavn drainage basin in Greenland dissipates enough gravitational potential energy to fully melt more than  $1 \text{ km}^3$  of ice [2]. Beautiful evidence that, as a result, outlet glaciers have thick temperate ice is in [31]. These physical effects motivate modelers to solve the conservation of energy equation simultaneously with the mass conservation (continuity) and momentum conservation (stress balance) equations. Traditionally the conservation of energy equation uses only temperature as the state variable [9], and this may be suitable for cold ice sheets, but ice sheets are generically polythermal. Enthalpy methods are a good way to track the energy content of polythermal ice sheets and glaciers [2], though one can also have a separate water-content equation for temperate ice [18]. In any case, the conservation of energy equation is strongly advection-dominated in general [9].

Pressurized basal water is required for most ice sliding. To model the production of such water in ice sheets one must at least compute the ice base temperature and the basal melt rate through the energy conservation equation [8, 11, 44, 56].

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, by whatever criterion that switch might be applied, though this has been attempted [23, 45]. However, “hybrid” schemes exist which solve the

SIA and SSA everywhere in the ice sheet [8, 61], or solve a related vertically-integrated model [16, 43], then combining the stresses or velocities according to different schemes.

“Higher-order” three-dimensional approximations of the Stokes stress balance, such as the Blatter-Pattyn model [4, 39], also use shallow approximations, at minimum including both the most-basic shallow assumption of well-defined thickness (see main text) *and* an assumption of hydrostatic normal stress [19]. 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. Vertically-integrated hybrids can generally be used at higher spatial resolution and longer time scales than higher-order models because the 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 [53] answer that Blatter-Pattyn is one.

Solving the Stokes stress balance itself [26, 29, 42] requires explicit accounting for incompressibility through a pressure variable. Numerical approximations of this stress balance are indefinite, thus harder to solve, essentially because incompressibility is an equality constraint. In plane flow one can 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 [42] or hybrid models.

The finite difference material in these notes should probably be read with reference [37] or similar in hand. The “main theorem for numerical PDE schemes” mentioned in the text is the Lax equivalence theorem [37]. Alternative numerical discretization techniques include the finite element [6], finite volume [30], and spectral [54] 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 implementation care is needed [27].

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 [17, 40, 41, 50]. The physics requires at least that the quantities  $H$  and  $u$  are continuous there, but several stress balance regimes exist near the grounding line, with increasing complexity as one focusses-in on the line [51].

Where to find exact solutions for ice flow models? The textbook Greve and Blatter [19] has a few. Halfar’s similarity solution to the SIA [20, 21] has been generalized to non-zero mass balance [10]. There are flow-line [5, 57] and cross-flow [49] solutions to the SSA model, and one can even construct an exact, steady marine ice sheet in the flow-line case [7]. For the Stokes equations themselves there are plane flow solutions for constant viscosity [3].

As a last resort for numerical verification, 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 [46]. There are such manufactured solutions to the thermomechanically-coupled SIA [9], plane flow Blatter-Pattyn model [15], and Glen-law Stokes equations [26, 29, 47].

## 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] G. BODVARSSON, *On the flow of ice-sheets and glaciers*, Jökull, 5 (1955), pp. 1–8.
- [6] D. BRAESS, *Finite Elements*, Cambridge University Press, Cambridge, third ed., 2007. Theory, fast solvers, and applications in elasticity theory, Translated from the German by Larry L. Schumaker.
- [7] E. BUELER, *An exact solution for a steady, flow-line marine ice sheet*, J. Glaciol., (to appear).
- [8] 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.
- [9] 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.
- [10] 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.
- [11] G. K. C. CLARKE, *Subglacial processes*, Annu. Rev. Earth Planet. Sci., 33 (2005), pp. 247–276.
- [12] K. M. CUFFEY AND W. S. B. PATERSON, *The Physics of Glaciers*, Elsevier, 4th ed., 2010.
- [13] A. C. FOWLER, *Mathematical Models in the Applied Sciences*, Cambridge Univ. Press, 1997.
- [14] 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.
- [15] R. GLOWINSKI AND J. RAPPASZ, *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.
- [16] D. GOLDBERG, *A variationally derived, depth-integrated approximation to a higher-order glaciological flow model*, J. Glaciol., 57 (2011), pp. 157–170.
- [17] D. GOLDBERG, D. M. HOLLAND, AND C. SCHOOF, *Grounding line movement and ice shelf buttressing in marine ice sheets*, J. Geophys. Res., 114 (2009).
- [18] R. GREVE, *A continuum-mechanical formulation for shallow polythermal ice sheets*, Phil. Trans. Royal Soc. London A, 355 (1997), pp. 921–974.
- [19] R. GREVE AND H. BLATTER, *Dynamics of Ice Sheets and Glaciers*, Advances in Geophysical and Environmental Mechanics and Mathematics, Springer, 2009.
- [20] P. HALFAR, *On the dynamics of the ice sheets*, J. Geophys. Res., 86 (1981), pp. 11065–11072.
- [21] ———, *On the dynamics of the ice sheets 2*, J. Geophys. Res., 88 (1983), pp. 6043–6051.
- [22] 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.
- [23] 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.
- [24] K. HUTTER, *Theoretical Glaciology*, D. Reidel, 1983.
- [25] 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.
- [26] G. JOUVET AND J. RAPPASZ, *Analysis and finite element approximation of a nonlinear stationary Stokes problem arising in glaciology*, Advances in Numerical Analysis, 2011 (2011), p. 24 pages.
- [27] C. T. KELLEY, *Solving Nonlinear Equations with Newton’s Method*, Fundamentals of Algorithms, SIAM Press, 1987.
- [28] 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.
- [29] 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).

- [30] R. J. LEVEQUE, *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] 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.
- [35] 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.
- [36] L. W. MORLAND AND I. R. JOHNSON, *Steady motion of ice sheets*, J. Glaciol., 25 (1980), pp. 229–246.
- [37] K. W. MORTON AND D. F. MAYERS, *Numerical Solutions of Partial Differential Equations: An Introduction*, Cambridge University Press, 2nd ed., 2005.
- [38] 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.
- [39] 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).
- [40] 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.
- [41] 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.
- [42] 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.
- [43] D. POLLARD AND R. M. DECONTTO, *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.
- [44] C. F. RAYMOND, *Energy balance of ice streams*, J. Glaciol., 46 (2000), pp. 665–647.
- [45] 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.
- [46] P. ROACHE, *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque, New Mexico, 1998.
- [47] A. SARGENT AND J. L. FASTOOK, *Manufactured analytical solutions for isothermal full-stokes ice sheet models*, The Cryosphere, 4 (2010), pp. 285–311.
- [48] J. C. SAVAGE AND W. S. B. PATERSON, *Borehole measurements in the Athabasca Glacier*, J. Geophys. Res., 68 (1963), pp. 4521–4536.
- [49] C. SCHOOF, *A variational approach to ice stream flow*, J. Fluid Mech., 556 (2006), pp. 227–251.
- [50] ———, *Marine ice-sheet dynamics. Part 1. The case of rapid sliding*, J. Fluid Mech., 573 (2007), pp. 27–55.
- [51] C. SCHOOF, *Marine ice sheet dynamics. Part 2: A Stokes Flow contact problem*, J. Fluid Mech., 679 (2011), pp. 122–255.
- [52] C. SCHOOF AND I. J. HEWITT, *Ice-sheet dynamics*, Annu. Rev. Fluid Mech., 45 (2013), pp. 217–239.
- [53] 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.
- [54] L. N. TREFETHEN, *Spectral Methods in MATLAB*, SIAM Press, 2000.
- [55] M. TRUFFER AND K. ECHELMAYER, *Of isbrae and ice streams*, Ann. Glaciol., 36 (2003), pp. 66–72.

- [56] 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.
- [57] 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.
- [58] ———, *Fundamentals of Glacier Dynamics*, CRC Press, 2nd ed., 2013.
- [59] M. WEIS, R. GREVE, AND K. HUTTER, *Theory of shallow ice shelves*, Continuum Mech. Thermodyn., 11 (1999), pp. 15–50.
- [60] P. WESSELING, *Principles of Computational Fluid Dynamics*, Springer-Verlag, 2001.
- [61] 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  $T_j^{n+1}$ , scheme (35) has only positive coefficients. Using [37] or other source, as needed, explain why this shows it is unconditionally stable.

6. *This multi-part exercise concerns the numerical treatment of “ $\nabla \cdot (D \nabla)$ ”.*

- (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 derived in (a) will occur if the  $u_j^{n+1}$  is computed from a linear combination of the older values which has all positive coefficients. If  $|E_0| \gg D_0$ , what does this stability assertion say about  $\Delta x$ ?

- (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)?

7. Derive the Green's function of the 1D heat equation, namely  $T = (4\pi Dt)^{-1/2} e^{-x^2/(4Dt)}$ , 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 in 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), namely  $T(t, x, y) = (4\pi Dt)^{-1} e^{-(x^2+y^2)/(4Dt)}$ , 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 formula (39) solves (26) in the case of flat bed ( $h = H$ ) and zero climatic mass balance ( $M = 0$ ). You will want to express divergence and gradient in polar coordinates.

**11.** Show, by appropriate integration of 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? Would a higher-order or Stokes model for an ice sheet with an advancing margin have significantly smaller thickness error, on the same grid and supposing we knew a relevant exact solution?

**14.** Derive the surface kinematical equation (40) from the mass continuity (25) and basal kinematical (41) equations. 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  $C_s = A(\rho g(1 - r)/4)^n$  and 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

$$u = [u_g^{n+1} + (C_s/M_0)((M_0x + u_gH_g)^{n+1} - (u_gH_g)^{n+1})]^{1/(n+1)}, \quad H = (M_0x + u_gH_g)/u,$$

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). On the other hand, note that the mass continuity equation  $M_0 = (uH)_x$  can be integrated to give the formula  $uH = M_0(x - x_g) + u_gH_g$  for the flux. Find  $u(x)$  first, then  $H(x)$  is in terms of  $u(x)$  as above. These exact solutions are from [57]. 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. I.e. consider the case where the calving front is at the point where the ice stream reaches flotation. (See also [5, 7].)

**17.** Derive the viscosity form of the flow law (17) from (5), in detail.

**18.** Derive the hydrostatic (20) and Blatter-Pattyn (21) equations, in detail.