Description and Evaluation of the Community Ice Sheet Model (CISM) v2.1

William H. Lipscomb[1,2], Stephen F. Price[1], Matthew J. Hoffman[1], Gunter R. Leguy[2], Andrew R. Bennett[3,4], Sarah L. Bradley[5], Katherine J. Evans[3], Jeremy G. Fyke[1,6], Joseph H. Kennedy[3], Mauro Perego[7], Douglas M. Ranken[1], William J. Sacks[2], Andrew G. Salinger[7], Lauren J. Vargo[1,8], and Patrick H. Worley[3]

[1]Los Alamos National Laboratory, Los Alamos, NM 87544 USA
[2]Climate and Global Dynamics Laboratory, National Center for Atmospheric Research, Boulder, CO 80305 USA
[3]Computational Earth Sciences Group, Oak Ridge National Laboratory, Oak Ridge, TN 37830 USA
[4]University of Washington, Seattle, WA 98195 USA
[5]Delft University of Technology, Delft, 2600 AA, the Netherlands
[6]Associated Engineering Group, Ltd., Vernon, BC V1T 9P9 Canada
[7]Center for Computing Research, Sandia National Laboratories, Albuquerque NM 87185 USA
[8]Victoria University of Wellington, Wellington 6140, New Zealand

Correspondence to: William H. Lipscomb (Lipscomb@ucar.edu)

Abstract. We describe and evaluate version 2.1 of the Community Ice Sheet Model (CISM). CISM is a parallel, 3D thermomechanical model, written mainly in Fortran, that solves equations for the momentum balance and the thickness and temperature evolution of ice sheets. CISM's velocity solver incorporates a hierarchy of Stokes-flow approximations, including shallow-shelf, depth-integrated higher-order, and 3D higher-order. CISM also includes a suite of test cases, links to third-party solver libraries, and parameterizations of physical processes such as basal sliding, iceberg calving, and sub-ice-shelf melting. The model has been verified for standard test problems, including the ISMIP-HOM experiments for higher-order models, and has participated in the initMIP–Greenland initialization experiment. In multi-millennial simulations with modern climate forcing on a 4-km grid, CISM reaches a steady state that is broadly consistent with observed flow patterns of the Greenland ice sheet. CISM has been integrated into version 2.0 of the Community Earth System Model, where it is being used for Greenland simulations under past, present and future climates. The code is open-source with extensive documentation, and remains under active development.

1 Introduction

As mass loss from the Greenland and Antarctic ice sheets has accelerated (Shepherd et al., 2012; Church et al., 2013; Hanna et al., 2013; Shepherd et al., 2017), climate modelers have recognized the importance of dynamic ice sheet models (ISMs) for predicting future mass loss and sea level rise (Vizcaino, 2014). Meanwhile, ISMs have become more accurate and complex in their representation of ice flow dynamics. Early ISMs used either the shallow-ice approximation (SIA; Hutter 1983) or the shallow-shelf approximation (SSA; MacAyeal 1989). The SIA, which assumes that vertical shear stresses are dominant, is valid for slow-moving ice-sheet interiors, whereas the SSA, which assumes that flow is dominated by lateral and longitudinal

stresses in the horizontal plane, is valid for floating ice shelves. Neither approximation is valid for ice streams and outlet glaciers where both vertical-shear and horizontal-plane stresses are important. Advanced ISMs developed in recent years solve the Stokes equations or various higher-order approximations (Pattyn et al., 2008; Schoof and Hindmarsh, 2010). Among the models that solve Stokes or higher-order equations, or otherwise combine features of the SIA and SSA, are the Parallel Ice

5 Sheet Model (PISM; Bueler and Brown 2009; Winkelmann et al. 2011), the Ice Sheet System Model (ISSM; Larour et al. 2012), the Penn State Model (Pollard and DeConto, 2012), BISICLES (Cornford et al., 2013), Elmer-Ice (Gagliardini et al., 2013), and MPAS-Albany Land Ice (MALI; Tezaur et al. 2015; Hoffman et al. 2018). Higher-order models have performed well for standard test cases (Pattyn et al., 2008, 2012) and have been applied to many scientific problems.

Here we describe and evaluate the Community Ice Sheet Model (CISM) version 2.1, a higher-order model that evolved from

10 the Glimmer model (Rutt et al., 2009). The current name reflects the model's evolution as a component of the Community Earth System Model (CESM; Hurrell et al. 2013). Like Glimmer, CISM is written mainly in Fortran 90 and its extensions, to maximize efficiency and to simplify coupling to climate models. Glimmer, however, is a serial SIA model, whereas CISM is a parallel model that solves not only the SIA but also higher-order Stokes approximations.

CISM development was guided by the following goals:

15 – The model should be well documented and easy to install and run on a variety of platforms, ranging from laptops to local clusters to high-performance supercomputers.

– It should solve a range of Stokes approximations, including the SIA, SSA, and higher-order approximations. Velocity solvers for these approximations are included in a new dynamical core called Glissade. (The dynamical core, or "dycore", is the part of the model that solves equations for conservation of mass, energy and momentum.)

20 – It should remain backward-compatible with Glimmer, allowing continued use of the older Glide SIA dycore.

– It should be well verified for standard test cases such as ISMIP-HOM (Pattyn et al., 2008), with a user-friendly verification framework.

– It should run efficiently—supporting whole-ice-sheet applications even on small platforms—and should scale to hundreds of processor cores, enabling century-to-millennial-scale simulations with higher-order solvers at grid resolutions

25 of ~ 5 km or finer.

– It should support not only standalone ice sheet simulations, but also coupled applications in which fields are exchanged with a global climate or Earth system model (CESM in particular).

– It should support simulations of the Greenland ice sheet, a scientific focus of CESM. Support for Antarctic applications is deferred to future model releases and publications.

30 – The code should be open-source, with periodic public releases.

Many of these features were present in CISM v.2.0, which was released in 2014 (Price et al., 2015). Changes between versions 2.0 and 2.1 have been made primarily to support robust, accurate, and efficient Greenland ice sheet simulations, both as a standalone model and in CESM. These changes include a depth-integrated higher-order velocity solver (Section 3.1.4), new parameterizations of basal sliding (Section 3.4), iceberg calving (Section 3.5), and sub-ice-shelf melting (Section 3.6), a new build-and-test structure (Section 4.6), and many improvements in model numerics.

We begin with an overview of CISM, including the core model and its testing and coupling infrastructure (Section 2). We then describe the model dynamics and physics, focusing on the new Glissade dycore (Section 3). To verify the model, we present results from standard test cases (Section 4). We then present results from long spin-ups of the Greenland ice sheet, with and without floating ice shelves (Section 5). Finally, we summarize the model results and suggest directions for future work (Section 6).

2 Model overview

CISM is a numerical model—a collection of software libraries, utilities and drivers—used to simulate ice sheet evolution. It is modular in design and is coded mainly in standard-compliant Fortran 90/95. CISM consists of several components:

– **cism_driver:** the high-level driver (i.e., the executable) that is used to run the standalone model in all configurations, including idealized test cases with simplified climate forcing, as well as model runs with realistic geometry and climate forcing data.

– **Glide:** the serial dycore based on shallow-ice dynamics. Glide solves the governing conservation equations and computes ice velocities, internal ice temperature, and ice geometry evolution. Apart from minor changes, this is the same dycore used in Glimmer and described by Rutt et al. (2009). It will not be discussed further here.

– **Glissade:** the dycore that solves higher-order approximations of the Stokes equations for ice flow. Glissade, unlike Glide, is fully parallel in order to take advantage of multi-processor, high-performance architectures. It is described in detail in Section 3.

– **Glint:** the original climate model interface for Glimmer. Glint allows the core ice sheet model to be coupled to a global climate model, or any other source of time-varying climate data on a lat-lon grid. Glint computes the surface mass balance (SMB) on the ice sheet grid using a positive-degree-day (PDD) scheme.

– **Glad:** a lightweight climate model interface that has replaced Glint in CESM. The CESM coupler supports remapping and downscaling between general land-surface grids and ice-sheet grids, and thus is able to send CISM an SMB that is already downscaled. The Glad interface simply sends and receives fields on the ice sheet grid, accumulating and averaging as needed based on the ice dynamic time step and the coupling interval.

- **Test cases:** idealized test cases for the Glide and Glissade dynamical cores. These are used to confirm that the model is working as expected and to provide a range of simple model configurations from which new users can learn about model options and create their own configurations. CISM test cases are described in Section 4.

- **Shared code:** a number of modules shared by different parts of the code. Examples include modules for defining derived types, physical constants, and model parameters, and modules that parse CISM configuration files and handle data input/output (I/O).

In order to reduce development effort, CISM runs on a structured rectangular grid and thus lacks the flexibility of models that run on variable-resolution or adaptive meshes (e.g., ISSM and BISICLES). Although CISM includes several common Stokes approximations, it does not solve the more complex "full Stokes" equations (Pattyn et al., 2008).

CISM is distributed as source code and therefore requires a reasonably complete build environment to compile the model. For UNIX and LINUX based systems, the CMake system is used to build the model. Sample build scripts for a number of standard architectures are included, as are working build scripts for a number of high-performance-computing architectures including Cheyenne (NCAR Computational and Information Systems Laboratory), Titan (Oak Ridge Leadership Computing Facility), Edison (National Energy Research Scientific Computing Center), and Cartesius (the Dutch national supercomputer).

The source code can be obtained by downloading a released version from the CISM website or by cloning the code from a public git repository; see Section 7. In either case, a Fortran90 compiler is required. Other software dependencies include the netCDF library (used for data I/O) and a Python distribution (used to analyze dependencies and to automatically generate parts of the code) with several specific Python modules. Parallel builds require MPI, and users desiring access to Trilinos packages (Heroux et al., 2005) will need to build Trilinos and then link it to CISM. Finally, CMake and Gnu Make are needed to compile the code and link to the various third-party libraries. The CISM documentation contains detailed instructions for downloading and building the code.

CISM is run by specifying the names of the executable (usually *cism_driver*) and a configuration file. Typically, the configuration file includes the input and output filenames, the grid dimensions, the time step and length of the run, the dycore (Glide or Glissade), and various options and parameter values appropriate for a given application. If not set in the config file, each option or parameter takes a default value. Supported options are described in the model documentation.

Multiple input, forcing and output files can be specified, containing any subset of a large number of global scalars and fields (1D, 2D and 3D). If a given field is "loadable" and is present in the input file, it is read automatically at start-up; otherwise it is set to a default value. Loadable fields include the initial ice thickness and temperature, bedrock topography, surface mass balance, surface air temperature, and geothermal heat flux. Forcing files are input files that are read at every time step (not just at initialization) so that time-dependent forcing can be applied during a simulation. Each output file includes a user-chosen set of variables (listed in the configuration file) and can contain multiple time slices, written at any frequency. A special kind of I/O file is the restart file, which includes all the fields needed to restart the model exactly. Whatever configuration options are chosen, model results are exactly reproducible (i.e., bit-for-bit) for a given computer platform and processor count, regardless of how many times a simulation is stopped and restarted.

Some basic information is sent to standard output during the run, and more verbose output is written at regular intervals to a log file. The log file lists the options and parameter values chosen for the run, and also notes the simulation time when CISM reads or writes I/O. In addition, the log file includes diagnostic information about the global state of the model (e.g., the total ice area and volume, total surface and basal mass balance, and maximum surface and basal ice speeds), along with vertical profiles of ice speed and temperature for a user-chosen grid point.

CISM2.1 has been implemented in CESM version 2.0, released in June 2018. Earlier versions of CESM supported one-way forcing of the Greenland ice sheet (using SIA dynamics) by the SMB computed in CESM's land model (Lipscomb et al., 2013). CESM2.0 extends this capability by supporting conservative, interactive coupling between ice sheets and the land and atmosphere. Coupling of ice sheets with the ocean is not yet supported, but is under development. CESM2 does not have an interactive Antarctic ice sheet, in part because of the many scientific and technical issues associated with ice sheet–ocean coupling. However, CISM2.1 includes many of the features needed to simulate marine ice sheets, and a developmental model version (including a grounding-line parameterization to be included in a near-future code release) has been used for Antarctic simulations.

3 Model dynamics and physics

CISM includes a parallel, higher-order dynamical core called Glissade, which solves equations for conservation of momentum (i.e., an appropriate approximation of Stokes flow), mass, and internal energy. Glissade numerics differ substantially from Glide numerics:

- *Velocity:* Glide solves the SIA only, but Glissade can solve several Stokes approximations, including the SIA, SSA, a depth-integrated higher-order approximation based on Goldberg (2011), and a 3D higher-order approximation based on Blatter (1995) and Pattyn (2003). Glide uses finite differences, whereas the Glissade velocity solvers use finite-element methods.

- *Temperature:* To evolve the ice temperature, Glide solves a prognostic equation that incorporates horizontal advection as well as vertical heat diffusion and internal dissipation. In Glissade, temperature advection is handled by the transport scheme, and a separate module solves for vertical diffusion and internal dissipation in each column.

- *Mass and tracer transport:* Glide solves an implicit diffusion equation for mass tranport, incorporating shallow-ice velocities. Glissade solves explicit equations for horizontal transport of mass (i.e., ice thickness) and tracers (e.g., ice temperature) using either an incremental remapping scheme (Lipscomb and Hunke, 2004) or a simpler first-order upwind scheme. Horizontal transport is followed by vertical remapping to terrain-following sigma coordinates.

Glissade numerics are described in detail below.

3.1 Velocity solvers

Glissade computes the ice velocity by solving approximate Stokes equations, given the surface elevation, ice thickness, ice temperature, and relevant boundary conditions. Section 3.1.1 describes the solution method for the Blatter-Pattyn (BP; Blatter 1995; Pattyn 2003) approximation, which is the most sophisticated and accurate solver in CISM. Subsequent sections discuss
5 simpler approximations.

3.1.1 Blatter-Pattyn approximation

The basic equations of the Blatter-Pattyn approximation are

$$\frac{\partial}{\partial x}\left[2\eta\left(2\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right]+\frac{\partial}{\partial y}\left[\eta\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right]+\frac{\partial}{\partial z}\left(\eta\frac{\partial u}{\partial z}\right)=\rho_i g\frac{\partial s}{\partial x},$$
$$\frac{\partial}{\partial y}\left[2\eta\left(2\frac{\partial v}{\partial y}+\frac{\partial u}{\partial x}\right)\right]+\frac{\partial}{\partial x}\left[\eta\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right]+\frac{\partial}{\partial z}\left(\eta\frac{\partial v}{\partial z}\right)=\rho_i g\frac{\partial s}{\partial y},$$

(1)

where u and v are the components of horizontal velocity, η is the effective viscosity, ρ_i is the density of ice (assumed constant),
10 g is gravitational acceleration, s is the surface elevation, and x, y, z are 3D Cartesian coordinates. These and other variables and parameters used in CISM are listed in Tables 1 and 2 for reference. In each equation the three terms on the LHS describe gradients of longitudinal stress, lateral shear stress, and vertical shear stress, respectively, and the RHS gives the gravitational driving force. The longitudinal and lateral shear stresses together are sometimes called membrane stresses (Hindmarsh, 2006). Neglecting membrane stress gradients leads to the much simpler SIA, and neglecting vertical shear stress gradients leads to the
15 SSA.

The equations are discretized on a structured 3D mesh. In the map plane the mesh consists of rectangular cells, each with 4 vertices. The vertical levels of the mesh are based on a terrain-following sigma coordinate system, with $\sigma = (s - z)/H$, where H is the ice thickness. Each cell layer is treated as a 3D hexahedral element with 8 nodes. (In other words, cells and vertices are defined to lie in the map plane, whereas elements and nodes live in 3D space.) Scalar 2D fields such as H and s are defined
20 at cell centers, and 3D scalars such as the ice temperature T lie at the centers of elements. Gradients of 2D scalars (e.g., the surface slope ∇s) live at vertices. The velocity components u and v are 3D fields defined at nodes.

For problems on multiple processors, a cell or vertex (and the associated elements and nodes in its column) may be either locally owned, or part of a computational halo. Each processor is responsible for computing u and v at its locally owned nodes. Any cell that contains one or more locally owned nodes and has H exceeding a threshold thickness (typically 1 m) is considered
25 dynamically active, as are the elements in its column. Likewise, any vertex of an active cell is active, as are the nodes in its column.

The effective viscosity η is defined in each active element by

$$\eta \equiv \frac{1}{2}A^{\frac{-1}{n}}\dot{\varepsilon}_e^{\frac{1-n}{n}},$$

(2)

where A is the temperature-dependent rate factor in Glen's flow law (Glen, 1955), and $\dot{\varepsilon}_e$ is the effective strain rate, given in the BP approximation by

$$\dot{\varepsilon}_e^2 = \dot{\varepsilon}_{xx}^2 + \dot{\varepsilon}_{yy}^2 + \dot{\varepsilon}_{xx}\dot{\varepsilon}_{yy} + \dot{\varepsilon}_{xy}^2 + \dot{\varepsilon}_{xz}^2 + \dot{\varepsilon}_{yz}^2, \tag{3}$$

where the components of the symmetric strain rate tensor are

$$\dot{\varepsilon}_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{4}$$

The rate factor A is given by an Arrhenius relationship:

$$A(T^*) = a e^{-Q/RT^*}, \tag{5}$$

where T^* is the absolute temperature corrected for the dependence of the melting point on pressure ($T^* = T + 8.7 \cdot 10^{-4}(s-z)$, with T in Kelvin), a is a temperature–independent material constant from Paterson and Budd (1982), Q is the activation energy for creep, and R is the universal gas constant.

The coupled PDEs (1) are discretized using the finite-element method (e.g., Hughes 2000; Huebner et al. 2001). This method is more often applied to unstructured grids, but was chosen for CISM's rectangular grid because of its robustness and natural treatment of boundary conditions (Dukowicz et al., 2010). The PDEs, with appropriate boundary conditions, are converted to a system of algebraic equations by dividing the full domain into subdomains (i.e., hexahedral elements), representing the velocity solution on each element, and integrating over elements. The solution is approximated as a sum over trilinear basis functions φ. Each active node is associated with a basis function whose value is $\varphi = 1$ at that node, with $\varphi = 0$ at all other nodes. The solution at a point within an element can be expanded in terms of basis functions and nodal values:

$$u(x,y,z) = \sum_n \varphi_n(x,y,z) u_n, \quad v(x,y,z) = \sum_n \varphi_n(x,y,z) v_n, \tag{6}$$

where the sum is over the nodes of the element; u_n and v_n are nodal values of the solution; and φ_n varies smoothly between 0 and 1 within the element.

Glissade's finite-element scheme is formally equivalent to that described by Perego et al. (2012). Eq. (1) can be written as

$$
\begin{aligned}
-\boldsymbol{\nabla}\cdot(2\eta\dot{\boldsymbol{\varepsilon}}_1) &= -\rho_i g\frac{\partial s}{\partial x}, \\
-\boldsymbol{\nabla}\cdot(2\eta\dot{\boldsymbol{\varepsilon}}_2) &= -\rho_i g\frac{\partial s}{\partial y},
\end{aligned}
\tag{7}
$$

where

$$
\dot{\boldsymbol{\varepsilon}}_1 = \begin{bmatrix} 2\dot{\varepsilon}_{xx} + \dot{\varepsilon}_{yy} \\ \dot{\varepsilon}_{xy} \\ \dot{\varepsilon}_{xz} \end{bmatrix}, \quad
\dot{\boldsymbol{\varepsilon}}_2 = \begin{bmatrix} \dot{\varepsilon}_{xy} \\ \dot{\varepsilon}_{xx} + 2\dot{\varepsilon}_{yy} \\ \dot{\varepsilon}_{yz} \end{bmatrix}. \tag{8}
$$

Following Perego et al. (2012), these equations can be rewritten in weak form. This is done by multiplying (7) by the basis functions and integrating over the domain, using integration by parts to eliminate the second derivative:

$$\int_\Omega 2\eta\dot{\varepsilon}_1(u,v)\cdot\nabla\varphi_1 \, d\Omega + \int_{\Gamma_B} \beta u\varphi_1 \, d\Gamma + \int_{\Gamma_L} pn_1\varphi_1 \, d\Gamma + \int_\Omega \rho_i g\frac{\partial s}{\partial x}\varphi_1 \, d\Omega = 0,$$

$$\int_\Omega 2\eta\dot{\varepsilon}_2(u,v)\cdot\nabla\varphi_2 \, d\Omega + \int_{\Gamma_B} \beta u\varphi_2 \, d\Gamma + \int_{\Gamma_L} pn_2\varphi_2 \, d\Gamma + \int_\Omega \rho_i g\frac{\partial s}{\partial y}\varphi_2 \, d\Omega = 0,$$

$$(9)$$

where Ω represents the domain volume, Γ_B denotes the lower boundary, Γ_L denotes the lateral boundary (e.g., the calving front of an ice shelf), β is a basal traction parameter, p is the pressure at the lateral boundary, and n_1 and n_2 are components of the normal to Γ_L. These equations can also be obtained from a variational principle as described by Dukowicz et al. (2010). The four terms in (9) represent internal ice stresses, basal friction, lateral pressure, and the gravitational driving force, respectively.

At the basal boundary we assume a friction law of the form

$$\boldsymbol{\tau_b} = \beta\boldsymbol{u_b}, \tag{10}$$

where $\boldsymbol{\tau_b}$ is the basal shear stress, $\boldsymbol{u_b} = (u_b, v_b)$ is the basal velocity, and β is a non-negative friction parameter that is defined at each vertex and can vary spatially. For some basal sliding laws (see Section 3.4), β depends on the basal velocity.

The lateral pressure p applies at marine-terminating boundaries. The net pressure is equal to the pressure directed outward from the ice toward the ocean by the ice, minus the (smaller) pressure directed inward from the ocean by the hydrostatic water pressure. The outward pressure is found by integrating $\rho_i g(s-z)dz$ from $(s-H)$ to s and then dividing by H; it is given by

$$p_{\text{out}} = \frac{\rho_i gH}{2}. \tag{11}$$

The inward pressure is found by integrating $\rho_o gzdz$ (where ρ_o is the seawater pressure) from $(s-H)$ to 0 and then dividing by $(H-s)$; it is given by

$$p_{\text{in}} = \frac{\rho_o g(s-H)^2}{2H} \tag{12}$$

Assuming hydrostatic balance (i.e., a floating margin), we have $s-H = (\rho_i/\rho_o)H$, in which case (11) and (12) can be combined to give the net pressure

$$p = \frac{\rho_i gH}{2}\left(1 - \frac{\rho_i}{\rho_w}\right), \tag{13}$$

directed from the ice to the ocean. However, (11) and (12) are used in the code because they are valid at both grounded and floating marine margins. They are combined to give the net pressure $p = p_{\text{out}} - p_{\text{in}}$ that is integrated over vertical cliff faces Γ_L in (9).

The gravitational forcing terms require evaluating the gradients $\partial s/\partial x$ and $\partial s/\partial y$ at each active vertex. For vertex (i,j) (which lies at the upper right corner of cell (i,j)), these are computed using a second-order-accurate centered approximation:

$$\frac{\partial s}{\partial x}(i,j) = \frac{s(i+1,j+1) + s(i+1,j) - s(i,j+1) - s(i,j)}{2\Delta x}, \tag{14}$$

and similarly for $\partial s / \partial y$. This is equivalent to computing $\partial s / \partial x$ at the edge midpoints adjacent to the vertex in the y direction, and then averaging the two edge gradients to the vertex. In some cases (e.g., near steep margins), (14) can lead to checkerboard noise (i.e., 2D patterns of alternating positive and negative deviations in s at the grid scale), because centered averaging of s permits a computational mode. To damp this noise, Glissade also supports upstream gradient calculations that do not have this

5 computational mode but are formally less accurate.

Eq. (14) is ambiguous at the ice margin, where at least one of the four cells neighboring a vertex is ice-free. One option is to include all cells, including ice-free cells, in the gradient. This approach works reasonably well (albeit with numerical errors; see, e.g., Van den Berg et al. (2006)) for land-based ice, but can give large gradients and excessive ice speeds at floating shelf margins. A second option is to include only ice-covered cells in the gradient. For example, suppose cells (i, j)

10 and $(i, j+1)$ have ice, but cells $(i+1, j)$ and $(i+1, j+1)$ are ice-free. Then, lacking the required information to compute x gradients at the adjacent edges, we would set $\partial s / \partial x = 0$. With a y gradient available at one adjacent edge, we would have $\partial s / \partial y = (s(i, j+1) - s(i, j)) / (2\Delta y)$. This option works well at shelf margins but tends to underestimate gradients at land margins. A third, hybrid option is to compute a gradient at edges where either (1) both adjacent cells are ice-covered, or (2) one cell is ice-covered and lies higher in elevation than a neighboring ice-free land cell. Thus, gradients are set to zero at edges

15 where an ice-covered cell (either grounded or floating) lies above ice-free ocean, or where an ice-covered land cell lies below ice-free land (i.e., a nunatak). Since this option works well for both land and shelf margins, it is the default.

Given T, s, H, and an initial guess for u and v, the problem is to solve (1) for u and v at each active node. This problem can be written as

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{b}, \tag{15}$$

20 or more fully,

$$\begin{bmatrix} \mathbf{A_{uu}} & \mathbf{A_{uv}} \\ \mathbf{A_{vu}} & \mathbf{A_{vv}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b_u} \\ \boldsymbol{b_v} \end{bmatrix}. \tag{16}$$

In Glissade, $\mathbf{A}$ is always symmetric and positive-definite.

Since $\mathbf{A}$ depends (through η and possibly β) on u and v, the problem is nonlinear and must be solved iteratively. For each nonlinear iteration, Glissade computes the 3D η field based on the current guess for the velocity field and solves a linear

25 problem of the form (16). Then η is updated and the process is repeated until the solution converges to within a given tolerance. This procedure is known as Picard iteration.

Appendix A describes how the terms in (9) are summed over elements and assembled into the matrix $\mathbf{A}$ and vector $\boldsymbol{b}$. Section 3.1.5 discusses solution methods for the inner linear problem and the outer nonlinear problem.

3.1.2 Shallow-ice approximation

The shallow-ice equations follow from the Blatter-Pattyn equations if membrane stresses are neglected. The SIA analogs of (1) are

$$
\frac{\partial}{\partial z}\left(\eta\frac{\partial u}{\partial z}\right) = \rho_i g \frac{\partial s}{\partial x},
$$
$$
\frac{\partial}{\partial z}\left(\eta\frac{\partial v}{\partial z}\right) = \rho_i g \frac{\partial s}{\partial y}. \tag{17}
$$

5   The SIA could be considered a special case of the Blatter-Pattyn equations and solved using the same finite-element methods, ignoring the horizontal stresses. With these methods, however, each ice column cannot be solved independently, because each node is linked to its horizontal neighbors by terms that arise during element assembly. As a result, a finite-element-based SIA solver is not very efficient.

     Instead, Glissade has an efficient local SIA solver. The solver is local in the sense that u and v in each column are found

10   independently of u and v in other columns. It resembles Glide's SIA solver as described by Payne and Dongelmans (1997) and Rutt et al. (2009), except that Glide incorporates the velocity solution in a diffusion equation for ice thickness, whereas Glissade's SIA solver computes u and v only, with thickness evolution handled separately as described in Section 3.3. For small problems that can run on one processor, there is no particular advantage to using Glissade's local SIA solver in place of Glide. Glide's implicit thickness solver permits a longer time step and thus is more efficient for problems run in serial. For

15   whole-ice-sheet problems, however, Glissade's parallel solver can hold more data in memory and may have faster throughput, simply because it can run on tens to hundreds of processors.

3.1.3 Shallow-shelf approximation

The SSA equations can be derived by vertically integrating the BP equations, given the assumption of small basal shear stress and vertically uniform velocity. The shallow-shelf analog of (1) is

$$
\frac{\partial}{\partial x}\left[2\bar{\eta}H\left(2\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right] + \frac{\partial}{\partial y}\left[\bar{\eta}H\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right] = \rho_i g H \frac{\partial s}{\partial x},
$$
$$
\frac{\partial}{\partial y}\left[2\bar{\eta}H\left(2\frac{\partial v}{\partial y}+\frac{\partial u}{\partial x}\right)\right] + \frac{\partial}{\partial x}\left[\bar{\eta}H\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right] = \rho_i g H \frac{\partial s}{\partial y}, \tag{18}
$$

where $\bar{\eta}$ is the vertically averaged effective viscosity, and u and v are vertically averaged velocity components. The SSA equations in weak form resemble (8) and (9), but without the vertical shear terms. Thus the SSA matrix and RHS can be assembled using the methods described in Section 3.1.1 and Appendix A, using 2D rectangular (instead of 3D hexahedral) elements and omitting the vertical-shear terms. The effective viscosity is defined as in (2), but with a vertically averaged flow

25   factor and with (3) replaced by

$$
\dot{\varepsilon}_e^2 = \dot{\varepsilon}_{xx}^2 + \dot{\varepsilon}_{yy}^2 + \dot{\varepsilon}_{xx}\dot{\varepsilon}_{yy} + \dot{\varepsilon}_{xy}^2. \tag{19}
$$

The element matrices are 2D analogs of (A3)–(A6), with vertical derivatives excluded.

3.1.4 Depth-integrated-viscosity approximation

Goldberg (2011) derived a higher-order stress approximation that in most cases is similar in accuracy to BP but is much cheaper to solve, because (as with the SSA) the matrix system is assembled and solved in two dimensions. Since the stress balance equations use a depth-integrated effective viscosity in place of a vertically varying viscosity, we refer to this scheme as a depth-integrated-viscosity approximation, or DIVA. Arthern and Williams (2017) used a similar model, also based on Goldberg (2011), to simulate ice flow in the Amundsen Sea sector of West Antarctica. To our knowledge, CISM is the first model to adapt this scheme for multi-millennial whole-ice-sheet simulations. Here we summarize the DIVA stress balance and describe Glissade's solution method. Section 4.2 compares DIVA results to BP results for the ISMIP-HOM experiments (Pattyn et al., 2008).

Dukowicz et al. (2010) showed that the Blatter-Pattyn equations (1) and associated boundary conditions can be derived by taking the first variation of a functional and setting it to zero. The functional includes terms that depend on the effective strain rate (3), which can be written as

$$\dot{\varepsilon}_{\text{BP}}^2 = u_x^2 + v_y^2 + u_x v_y + \frac{1}{4}(u_y + v_x)^2 + \frac{1}{4}u_z^2 + \frac{1}{4}v_z^2, \tag{20}$$

where u_x denotes the partial derivative $\partial u/\partial x$, and similarly for other derivatives. Goldberg (2011) derived an alternate set of equations using a similar functional, but with the horizontal velocity gradients in the effective strain rate replaced by their vertical averages:

$$\dot{\varepsilon}_{\text{DIVA}}^2 = \bar{u}_x^2 + \bar{v}_y^2 + \bar{u}_x \bar{v}_y + \frac{1}{4}(\bar{u}_y + \bar{v}_x)^2 + \frac{1}{4}u_z^2 + \frac{1}{4}v_z^2, \tag{21}$$

where

$$\bar{u} = \frac{1}{H}\int_b^s u(z)dz, \quad \bar{v} = \frac{1}{H}\int_b^s v(z)dz. \tag{22}$$

The resulting equations of motion, analogous to (1), are

$$\frac{1}{H}\frac{\partial}{\partial x}\left[2\bar{\eta}H\left(2\frac{\partial\bar{u}}{\partial x} + \frac{\partial\bar{v}}{\partial y}\right)\right] + \frac{1}{H}\frac{\partial}{\partial y}\left[\bar{\eta}H\left(\frac{\partial\bar{u}}{\partial y} + \frac{\partial\bar{v}}{\partial x}\right)\right] + \frac{\partial}{\partial z}\left(\eta\frac{\partial u}{\partial z}\right) = \rho_i g\frac{\partial s}{\partial x},$$

$$\frac{1}{H}\frac{\partial}{\partial y}\left[2\bar{\eta}H\left(2\frac{\partial\bar{v}}{\partial y} + \frac{\partial\bar{u}}{\partial x}\right)\right] + \frac{1}{H}\frac{\partial}{\partial x}\left[\bar{\eta}H\left(\frac{\partial\bar{u}}{\partial y} + \frac{\partial\bar{v}}{\partial x}\right)\right] + \frac{\partial}{\partial z}\left(\eta\frac{\partial v}{\partial z}\right) = \rho_i g\frac{\partial s}{\partial y}, \tag{23}$$

where the depth-integrated effective viscosity is given by

$$\bar{\eta} = \frac{1}{H}\int_b^s \eta(z)dz. \tag{24}$$

The vertical shear terms still contain $\eta(z)$.

Since the horizontal stress terms are depth-independent, (23) can be integrated in the vertical to give

$$\frac{\partial}{\partial x}\left[2\bar{\eta}H\left(2\frac{\partial\bar{u}}{\partial x}+\frac{\partial\bar{v}}{\partial y}\right)\right]+\frac{\partial}{\partial y}\left[\bar{\eta}H\left(\frac{\partial\bar{u}}{\partial y}+\frac{\partial\bar{v}}{\partial x}\right)\right]-\beta u_b=\rho_i gH\frac{\partial s}{\partial x},$$

$$\frac{\partial}{\partial y}\left[2\bar{\eta}H\left(2\frac{\partial\bar{v}}{\partial y}+\frac{\partial\bar{u}}{\partial x}\right)\right]+\frac{\partial}{\partial x}\left[\bar{\eta}H\left(\frac{\partial\bar{u}}{\partial y}+\frac{\partial\bar{v}}{\partial x}\right)\right]-\beta v_b=\rho_i gH\frac{\partial s}{\partial y},$$

(25)

where the boundary conditions at b and s have been used to evaluate the vertical stress terms, with a sliding law of the form (10). Eq. (25) is equivalent to the SSA stress balance (18), except for the addition of basal stress terms. Thus the methods used

5 to assemble and solve the SSA equations can be applied to the DIVA equations to find the mean velocity components $\bar{u}$ and $\bar{v}$.

In order to solve (25), however, the basal stress terms must be rewritten in terms of $\bar{u}$ and $\bar{v}$. Following Goldberg (2011), we show how this is done for the x component of velocity; the y component is analogous. The x component of (25) can be rearranged and divided by H:

$$\frac{\beta u_b}{H}=\frac{1}{H}\frac{\partial}{\partial x}\left(2H\bar{\eta}\left(2\frac{\partial\bar{u}}{\partial x}+\frac{\partial\bar{v}}{\partial y}\right)\right)+\frac{1}{H}\frac{\partial}{\partial y}\left(H\bar{\eta}\left(\frac{\partial\bar{u}}{\partial y}+\frac{\partial\bar{v}}{\partial x}\right)\right)-\rho_i g\frac{\partial s}{\partial x}.$$

(26)

10 From (23), the RHS of (26) is just $-(\eta u_z)_z$, giving

$$\frac{\beta u_b}{H}=-\frac{\partial}{\partial z}\left(\eta\frac{\partial u}{\partial z}\right).$$

(27)

Integrating (27) from z to the surface s gives

$$\eta\frac{\partial u}{\partial z}=\frac{\beta u_b\left(s-z\right)}{H}.$$

(28)

Dividing by η and integrating from b to z gives $u(z)$ in terms of u_b and $\eta(z)$:

15 $$u(z)=u_b+\frac{\beta u_b}{H}\int_b^z\frac{(s-z')}{\eta(z')}dz'.$$

(29)

Following Arthern et al. (2015), we can define some useful integrals F_n as

$$F_n\equiv\int_b^s\frac{1}{\eta}\left(\frac{s-z}{H}\right)^n dz.$$

(30)

In this notation the surface velocity in related to the bed velocity by

$$u_s=u_b\left(1+\beta F_1\right).$$

(31)

20 Integrating $u(z)$ from the bed to the surface gives the depth-averaged mean velocity $\bar{u}$:

$$\bar{u}=u_b\left(1+\beta F_2\right).$$

(32)

We can think of F_2 as a depth-integrated inverse viscosity. The less viscous the ice, the greater the value of F_2, and the greater the difference between $\bar{u}$ and u_b.

Given (32), we can replace βu_b and βv_b in (25) with $\beta_{\text{eff}}\bar{u}$ and $\beta_{\text{eff}}\bar{v}$, respectively, where

$$\beta_{\text{eff}} = \frac{\beta}{1 + \beta F_2}. \tag{33}$$

For a frozen bed ($u_b = v_b = 0$, with nonzero basal stress τ_{bx}), the βu_b term on the RHS of (29) is replaced by τ_{bx}, leading to

$$\bar{u} = \tau_{bx} F_2. \tag{34}$$

5   Then the basal stress terms in (25) can be replaced by $\beta_{\text{eff}}^{\text{frz}}\bar{u}$ and $\beta_{\text{eff}}^{\text{frz}}\bar{v}$, where

$$\beta_{\text{eff}}^{\text{frz}} = \frac{1}{F_2} \tag{35}$$

With these substitutions, (25) can be written in terms of mean velocities $\bar{u}$ and $\bar{v}$ and is fully analogous to the SSA.

In order to compute $\bar{\eta}$, the effective viscosity $\eta(z)$ must be evaluated at each level. We use (2) with the effective strain rate (21). The horizontal terms in $\dot{\varepsilon}_{\text{DIVA}}^2$ are found using the mean velocities from the previous iteration. The vertical shear terms

10   are computed using (28):

$$u_z(z) = \frac{\tau_{bx}(s - z)}{\eta(z)H}, \quad v_z(z) = \frac{\tau_{by}(s - z)}{\eta(z)H}, \tag{36}$$

where both $\eta(z)$ and $\boldsymbol{\tau}_b$ are from the previous iteration.

The DIVA solution procedure can be summarized as follows:

1. Starting with the current guess for the velocity field (u, v), assemble the DIVA solution matrix in analogy to the SSA

15   solution matrix, with the following DIVA-specific computations:

   – Compute $\eta(z)$ using (2), (21) and (36), and integrate to find the depth-averaged $\bar{\eta}$.

   – Evaluate numerically the vertical integrals in (29) and (30).

   – Interpolate F_2 to vertices and compute β_{eff}.

2. Solve the matrix system for $\bar{u}$ and $\bar{v}$.

20   3. Given $(\bar{u}, \bar{v})$ at each vertex, use (32) to find (u_b, v_b), and then use (29) to find $u(z)$ at each level.

4. Iterate to convergence.

The code is initialized with $(u, v) = 0$ everywhere, and thereafter each iteration starts with (u, v) from the previous time step.

DIVA assumes that horizontal gradients of membrane stresses are vertically uniform. It is less accurate than the BP approximation in regions of slow sliding and rough bed topography, where the horizontal gradients of membrane stresses vary strongly

25   with height, and may also be less accurate where vertical temperature gradients are large. The accuracy of DIVA compared to BP is discussed in Section 4.2 with reference to the ISMIP-HOM benchmark experiments.

3.1.5 Solving the matrix system

After assembling the matrices and right-hand side vectors for the chosen approximation (SSA, DIVA or BP), we solve the linear problem (16). Glissade supports three kinds of solvers: (1) a native Fortran preconditioned conjugate gradient (PCG) solver, (2) links to Trilinos solver libraries, and (3) links to the Sparse Linear Algebra Package (SLAP).

5   The native PCG solver works directly with the assembled matrices $\mathbf{A_{uu}}$, $\mathbf{A_{uv}}$, $\mathbf{A_{vu}}$, and $\mathbf{A_{vv}}$, and the right-hand-side vectors $\boldsymbol{b_u}$ and $\boldsymbol{b_v}$. Instead of converting to a sparse matrix format such as compressed row storage, Glissade maintains these matrices and vectors as structured rectangular arrays with i and j indices, so that they can be handled by CISM's halo update routines. Each node location (i,j) (in 2D) or (i,j,k) (in 3D) corresponds to a row of the matrix, and an additional index m ranges over the neighboring nodes that can have nonzero terms in the columns of that row. The maximum number of nonzero terms per row is 9 in 2D and 27 in 3D.

The matrices and RHS vectors are passed to either a "standard" PCG solver (Shewchuk, 1994) or a Chronopoulos-Gear PCG solver (Chronopoulos, 1986; Chronopoulos and Gear, 1989). The PCG algorithm includes two dot products (each requiring a global sum), one matrix-vector product, and one preconditioning step per iteration. For small problems, the dominant computational cost is the matrix-vector product. For large problems, however, the global sums become increasingly expensive. In this case the Chronopoulos-Gear solver is more efficient than the standard solver, because it rearranges operations such that both global sums are done with a single MPI call.

As described by Shewchuk (1994), the convergence rate of the PCG method depends on the effectiveness of the preconditioner (i.e., a matrix $\mathbf{M}$ which approximates $\mathbf{A}$ and can be inverted efficiently). Glissade's native PCG solver has two preconditioning options:

20   – Diagonal preconditioning, in which $\mathbf{M}$ consists of the diagonal terms of $\mathbf{A}$ and is trivial to invert.

– Shallow-ice-based preconditioning (for the BP approximation only). In this case $\mathbf{M}$ includes only the terms in $\mathbf{A}$ that link a given node to itself and its neighbors above and below. Thus $\mathbf{M}$ is tridiagonal and can be inverted efficiently. This preconditioner works well when the physics is dominated by vertical shear, but not as well when membrane stresses are important. Since the preconditioner is local (it consists of independent column solves), it scales well for large problems.

25   The Trilinos solver (Heroux et al., 2005) uses C++ packages developed at Sandia National Laboratories. As described by Evans et al. (2012), an earlier version of CISM's higher-order dycore, known as Glam, was parallelized and linked to Trilinos. The Trilinos links developed for Glam were then adapted for Glissade's SSA, DIVA, and BP solvers. The CISM documentation gives instructions for building and linking to Trilinos and choosing appropriate solver settings.

SLAP is a set of Fortran routines for solving sparse systems of linear equations. SLAP was part of Glimmer and is used to solve for thickness evolution and temperature advection in Glide. It can also be used to solve higher-order systems in Glissade, using either GMRES or a biconjugate gradient solver. The SLAP solver, however, is a serial code unsuited for large problems.

The default linear solver is native PCG using the Chronopoulos-Gear algorithm. Although it has been found to run efficiently on platforms ranging from laptops to supercomputers, its preconditioning options are limited, so convergence can be slow for

problems with dominant membrane stresses (e.g., large marine ice sheets). SLAP solvers are generally robust and efficient, but are limited to one processor. Trilinos contains many solver options, but in tests to date (using a GMRES solver with ILU preconditioning) it has been found to be slower than the native PCG solver because of the extra cost of setting up Trilinos data structures, and possibly the slower performance of C++ compared to Fortran for the solver.

5   The linear solver is wrapped by a nonlinear solver that does Picard iteration. Following each iteration, a new global matrix is assembled, using the latest velocity solution to compute the effective viscosity. Glissade then computes the new residual $r = b - \mathbf{A}x$. If the l_2-norm of the residual, defined as $\sqrt{(r, r)}$, is smaller than a prescribed tolerance threshold, the nonlinear system of equations is considered solved. Otherwise the linear solver is called again, until the solution converges or the maximum number of nonlinear iterations (typically 100) is reached. The number of nonlinear iterations per solve—and optionally, the

10   number of linear iterations per nonlinear iteration—are written to standard output. In the event of non-convergence, a warning message is written to output, but the run continues. In some cases, convergence may improve later in the simulation, or solutions may be deemed sufficiently accurate even when not fully converged.

3.2   Temperature solver

The thermal evolution of the ice sheet is given by

$$\frac{\partial T}{\partial t} = \frac{k_i}{\rho_i c_i} \nabla^2 T - \mathbf{u} \cdot \boldsymbol{\nabla} T - w \frac{\partial T}{\partial z} + \frac{\Phi}{\rho_i c_i}, \tag{37}$$

where T is the temperature in °C, $\boldsymbol{u} = (u, v)$ is the horizontal velocity, w is the vertical velocity, k_i is the thermal conductivity of ice, c_i is the specific heat of ice, ρ_i is the ice density, and Φ is the rate of heating due to internal deformation and dissipation. This equation describes the conservation of internal energy under horizontal and vertical diffusion (the first term on the RHS), horizontal and vertical advection (the second and third terms, respectively), and internal heat dissipation (the last term). Unlike

20   Glide, which solves this equation in a single calculation, Glissade divides the temperature evolution into separate advection and diffusion/dissipation components. The temperature module uses finite-difference methods to solve for vertical diffusion and internal dissipation in each ice column:

$$\frac{\partial T}{\partial t} = \frac{k_i}{\rho_i c} \frac{\partial^2 T}{\partial z^2} + \frac{\Phi}{\rho_i c_i}, \tag{38}$$

as described in this section. The advective part of (37) is described in Section 3.3. Horizontal diffusion is assumed to be

25   negligible compared to vertical diffusion, giving $\nabla^2 T \simeq \frac{\partial^2 T}{\partial z^2}$.

    In Glissade's vertical discretization of temperature, T is located at the midpoints of the $(nz - 1)$ layers, staggered relative to the velocity. This staggering makes it easier to conserve internal energy under transport, where the internal energy in an ice column is equal to the sum over layers of $\rho_i c_i T \Delta z$, with layer thickness Δz. The upper surface skin temperature is denoted by T_0 and the lower surface skin temperature by T_{nz}, giving a total of $nz + 1$ temperature values in each column.

30   The following sections describe how the terms in (38) are computed, how the boundary conditions are specified, and how the equation is solved.

3.2.1 Vertical diffusion

Glissade uses a vertical σ-coordinate, with $\sigma \equiv (s - z)/H$. Thus the vertical diffusion terms can be written as

$$\frac{\partial^2 T}{\partial z^2} = \frac{1}{H^2} \frac{\partial^2 T}{\partial \sigma^2}. \tag{39}$$

The central difference formulas for first derivatives at the upper and lower interfaces of layer k are

$$\left.\frac{\partial T}{\partial \sigma}\right|_{\sigma_k} = \frac{T_k - T_{k-1}}{\tilde{\sigma}_k - \tilde{\sigma}_{k-1}},$$

$$\left.\frac{\partial T}{\partial \sigma}\right|_{\sigma_{k+1}} = \frac{T_{k+1} - T_k}{\tilde{\sigma}_{k+1} - \tilde{\sigma}_k}, \tag{40}$$

where $\tilde{\sigma}_k$ is the value of σ at the midpoint of layer k, halfway between σ_k and σ_{k+1}. The second partial derivative, defined at the midpoint of layer k, is

$$\left.\frac{\partial^2 T}{\partial \sigma^2}\right|_{\tilde{\sigma}_k} = \frac{\left.\frac{\partial T}{\partial \sigma}\right|_{\sigma_{k+1}} - \left.\frac{\partial T}{\partial \sigma}\right|_{\sigma_k}}{\sigma_{k+1} - \sigma_k} \tag{41}$$

Inserting (40) in (41), we obtain the vertical diffusion term:

$$\left.\frac{\partial^2 T}{\partial \sigma^2}\right|_{\tilde{\sigma}_k} = \frac{T_{k-1}}{(\tilde{\sigma}_k - \tilde{\sigma}_{k-1})(\sigma_{k+1} - \sigma_k)} - T_k \left(\frac{1}{(\tilde{\sigma}_k - \tilde{\sigma}_{k-1})(\sigma_{k+1} - \sigma_k)} + \frac{1}{(\tilde{\sigma}_{k+1} - \tilde{\sigma}_k)(\sigma_{k+1} - \sigma_k)} \right)$$
$$+ \frac{T_{k+1}}{(\tilde{\sigma}_{k+1} - \tilde{\sigma}_k)(\sigma_{k+1} - \sigma_k)}. \tag{42}$$

3.2.2 Heat dissipation

In higher-order models the internal heating rate Φ in (38) is given by the tensor product of strain rate and stress:

$$\Phi = \dot{\varepsilon}_{ij} \tau_{ij}, \tag{43}$$

where τ_{ij} is the deviatoric stress tensor. The effective stress (cf. (3)) is defined by

$$\tau_e^2 = \frac{1}{2} \tau_{ij} \tau_{ij}. \tag{44}$$

The stress tensor is related to the strain rate and the effective viscosity by

$$\tau_{ij} = 2\eta \dot{\varepsilon}_{ij}. \tag{45}$$

20 Dividing each side of (45) by 2η, substituting in (43), and using (44) gives

$$\Phi = \frac{\tau_e^2}{\eta}. \tag{46}$$

Both terms on the RHS of (46) are available to the temperature solver, since the higher-order velocity solver computes η during matrix assembly and diagnoses τ_e from η and $\dot{\varepsilon}_{ij}$ at the end of the calculation.

3.2.3 Boundary conditions

The temperature T_0 at the upper boundary is set to $\min(T_{\text{air}}, 0)$, where the surface air temperature T_{air} is usually specified from data or passed from a climate model. The diffusive heat flux at the upper boundary (defined as positive up) is

$$F_d^{\text{top}} = \frac{k_i}{H} \frac{T_1 - T_0}{\tilde{\sigma}_1},$$ (47)

where the denominator contains just one term because $\sigma_0 = 0$.

The lower ice boundary is more complex. For grounded ice there are three heat sources and sinks. First, the diffusive flux from the bottom surface to the ice interior (positive up) is

$$F_d^{\text{bot}} = \frac{k_i}{H} \frac{T_{nz} - T_{nz-1}}{1 - \tilde{\sigma}_{nz-1}}.$$ (48)

Second, there is a geothermal heat flux F_g which can be prescribed as a constant (~ 0.05 W m^{-2}) or read from an input file. Finally, there is a frictional heat flux associated with basal sliding, given by

$$F_f = \boldsymbol{\tau_b} \cdot \boldsymbol{u_b},$$ (49)

where $\boldsymbol{\tau_b}$ and $\boldsymbol{u_b}$ are 2D vectors of basal shear stress and basal velocity, respectively. With a friction law of the form (10), this becomes

$$F_f = \beta(u_b^2 + v_b^2).$$ (50)

If the basal temperature $T_{nz} < T_{\text{pmp}}$ (where T_{pmp} is the pressure melting point temperature), then the fluxes at the lower boundary must balance:

$$F_g + F_f = F_d^{\text{bot}}.$$ (51)

In other words, the energy supplied by geothermal heating and sliding friction is equal to the energy removed by vertical diffusion. If, on the other hand, $T_{nz} = T_{\text{pmp}}$, then the net flux is nonzero and is used to melt or freeze ice at the boundary:

$$M_b = \frac{F_g + F_f - F_d^{\text{bot}}}{\rho_i L},$$ (52)

where M_b is the melt rate and L is the latent heat of melting. Melting generates basal water, which may either stay in place, flow downstream (possibly replaced by water from upstream), or simply disappear from the system, depending on the basal water parameterization. While basal water is present, T_{nz} is held at T_{pmp}.

For floating ice the basal boundary condition is simpler; T_{nz} is set to the freezing temperature T_f of seawater, taken as a linear function of depth based on the pressure-dependent melting point of seawater. Optionally, a melt rate can be prescribed at the lower surface (Section 3.6).

3.2.4 Vertical temperature solution

Eq. (38) can be discretized for layer k as

$$\frac{T_k^{n+1} - T_k^n}{\Delta t} = \frac{k_i}{\rho_i c_i H^2} \left[a_k T_{k-1}^{n+1} - (a_k + b_k) T_k^{n+1} + b_k T_{k+1}^{n+1} \right] + \frac{\Phi_k}{\rho_i c_i}, \tag{53}$$

where the coefficients a_k and b_k are inferred from (42), and n is the current time level. The vertical diffusion terms are evaluated at the new time level, making the discretization backward-Euler (i.e., fully implicit) in time. Eq. (53) can be rewritten as

$$-\alpha_k T_{k-1}^{n+1} + (1 + \alpha_k + \beta_k) T_k^{n+1} - \beta_k T_{k+1}^{n+1} = T_k^n + \frac{\Phi_k \Delta t}{\rho_i c_i}, \tag{54}$$

where

$$\alpha_k = \frac{k_i a_k \Delta t}{\rho_i c_i H^2}, \quad \beta_k = \frac{k_i b_k \Delta t}{\rho_i c_i H^2}. \tag{55}$$

At the lower surface we have either a temperature boundary condition ($T_{nz} = T_{\mathrm{pmp}}$ for grounded ice, or $T_{nz} = T_f$ for floating ice) or a flux boundary condition:

$$F_f + F_g - \frac{k_i}{H} \frac{T_{nz}^{n+1} - T_{nz-1}^{n+1}}{1 - \tilde{\sigma}_{nz-1}} = 0, \tag{56}$$

which can be rearranged to give

$$-T_{nz-1}^{n+1} + T_{nz}^{n+1} = \frac{(F_f + F_g) H (1 - \tilde{\sigma}_{nz-1})}{k_i}. \tag{57}$$

In each ice column the above equations form a tridiagonal system that is solved for T_k in each layer.

Occasionally, the solution T_k can exceed T_{pmp} for a given layer. If so, we set $T_k = T_{\mathrm{pmp}}$ and use the extra energy to melt ice internally. This melt is assumed to drain immediately to the bed. If (52) applies, we compute M_b and adjust the basal water depth. When the basal water goes to zero, T_{nz} is set to the temperature of the lowest layer (less than T_{pmp} at the bed), so that the flux boundary condition will apply during the next time step.

3.3 Mass and tracer transport

The transport equation for ice thickness H can be written as

$$\frac{\partial H}{\partial t} + \boldsymbol{\nabla} \cdot (H \boldsymbol{U}) = B, \tag{58}$$

where $\boldsymbol{U}$ is the vertically averaged 2D velocity and B is the total surface mass balance. This equation describes the conservation of ice volume under horizontal transport. With the assumption of uniform density, volume conservation is equivalent to mass conservation. There is a similar conservation equation for the internal energy in each ice layer:

$$\frac{\partial (hT)}{\partial t} + \boldsymbol{\nabla} \cdot (hT \boldsymbol{u}) = 0, \tag{59}$$

where h is the layer thickness, T is the temperature (located at the layer midpoint), and $\boldsymbol{u}$ is the layer-average horizontal velocity. (Vertical diffusion and internal dissipation are handled separately as described in Section 3.2.) If other tracers are present, their transport is described by conservation equations of the same form as (59). Glissade solves (58) and (59) in a coordinated way, one layer at a time. All tracers, including temperature, are advected using the same algorithm.

5   Glissade has two horizontal transport schemes: a first-order upwind scheme and a more accurate incremental remapping (IR) scheme (Dukowicz and Baumgardner, 2000; Lipscomb and Hunke, 2004). The IR scheme is the default. This scheme was first implemented in the Los Alamos sea ice model, CICE, and has been adapted for CISM. Since the scheme is fairly complex, we give a summary in Section 3.3.1 and refer the reader to the earlier publications for details.

After horizontal transport, the mass balance is applied at the top and bottom ice surfaces. The new vertical layers generally 10   do not have the desired spacing in σ coordinates. For this reason, a vertical remapping scheme is applied to transfer ice volume, internal energy, and other conserved quantities between adjacent layers, thus restoring each column to the desired σ coordinates while conserving mass and energy. (This is a common feature of arbitrary Lagrangian-Eulerian (ALE) methods.) Internal energy divided by mass gives the new layer temperatures, and similarly for other tracers.

3.3.1   Incremental remapping

15   The incremental remapping scheme has several desirable features:

- It conserves the quantities being transported (including mass and internal energy).

- It is non-oscillatory; that is, it does not create spurious ripples in the transported fields.

- It preserves tracer monotonicity. That is, it does not create new extrema in tracers such as temperature; the values at time $n+1$ are bounded by the values at time n. Thus, T never rises above the melting point under advection.

20   - It is second-order accurate in space and therefore is less diffusive than first-order schemes. The accuracy may be reduced locally to first order to preserve monotonicity.

- It is efficient for large numbers of tracers. Much of the work is geometrical and is performed only once per cell edge instead of being repeated for each field being transported.

The model's upwind scheme, like IR, is conservative, non-oscillatory, and monotonicity-preserving, but because it is first-order 25   it is highly diffusive.

The IR time step is limited by the requirement that trajectories projected backward from grid cell corners are confined to the four surrounding cells; this is what is meant by incremental as opposed to general remapping. This requirement leads to an advective Courant–Friedrichs–Lewy (CFL) condition,

$$\frac{\max |\mathbf{u}| \Delta t}{\Delta x} \leq 1 \tag{60}$$

30   The remapping algorithm can be summarized as follows:

1. Given mean values of the ice thickness and tracer fields in each grid cell, construct linear approximations of these fields that preserve the mean. Limit the field gradients to preserve monotonicity.

2. Given ice velocities at grid cell vertices, identify departure regions for the fluxes across each cell edge. Divide these departure regions into triangles and compute the coordinates of the triangle vertices.

3. Integrate the thickness and tracer fields over the departure triangles to obtain the mass and energy transported across each cell edge.

4. Given these transports, update the state variables.

These steps are carried out for each of $nz - 1$ ice layers, where nz is the number of velocity levels.

3.3.2 CFL checks

As mentioned above, the time step for explicit mass transport is limited by the advective CFL condition (60). For ice flow parallel to ∇s, ice thickness evolution is diffusive, giving rise to a diffusive CFL condition (Bueler, 2009):

$$(\max D)\Delta t \leq 0.5\Delta x^2, \tag{61}$$

where D is the ice flow diffusivity. Flow governed by the shallow-ice approximation is subject to this diffusive CFL. The stability of Glissade's BP, DIVA, and SSA solvers, however, is limited by (60) in practice; (61) is too restrictive. For this reason, Glissade checks for advective CFL violations at each time step. Optionally, the transport equation can be adaptively subcycled within a time step to satisfy advective CFL. This can prevent the model from crashing, though possibly with a loss of accuracy.

3.4 Basal sliding

Glissade assumes a basal friction law of the form (10). If β were independent of velocity, then (10) would be a simple linear sliding law. Allowing β to depend on velocity allows more complex and physically realistic sliding laws. The following options are supported in CISM:

– Spatially uniform β, possibly a large value that makes sliding negligible.

– No sliding, enforced by a Dirichlet boundary condition $\boldsymbol{u_b} = 0$ during finite-element assembly.

– A 2D β field specified from an external file. Typically, β is chosen at each vertex to optimize the fit to observed surface velocity.

– β is set to a large value where the bed is frozen ($T_b < T_{pmp}$) and a lower value where the bed is thawed ($T_b = T_{pmp}$).

- Power-law sliding, based on the sliding relation of Weertman (1957). Basal velocity is given by

$$\boldsymbol{u_b} = k\boldsymbol{\tau_b^p}N^{-q}, \tag{62}$$

where N is the effective pressure (the difference between the ice overburden pressure $P_0 = \rho_i gH$ and the basal water pressure p_w) and k is an empirical constant. This can be rearranged to give

$$\boldsymbol{\tau_b} = k^{-1/p}N^{q/p}\boldsymbol{u_b}^{1/p}. \tag{63}$$

- Plastic sliding, in which the bed deforms with a specified till yield stress.

- Coulomb friction as described by Schoof (2005), with notation and default values from Pimentel et al. (2010). The form of the sliding law is

$$\boldsymbol{\tau_b} = C|\boldsymbol{u}|^{\frac{1}{n}-1}\boldsymbol{u}\left(\frac{N^n}{\kappa|\boldsymbol{u}|+N^n}\right)^{\frac{1}{n}} \tag{64}$$

where C is a constant, $\kappa = m_{\max}/(\lambda_{\max}A_b)$, $m_{\max}$ is the maximum bed obstacle slope, $\lambda_{\max}$ is the wavelength of bedrock bumps, and A_b is a basal flow-law parameter. Eq. (64) has two asymptotic behaviors. In the interior, where the ice is thick and slow-moving, $\kappa|\mathbf{u}| \ll N^n$ and the basal friction is independent of N:

$$\boldsymbol{\tau_b} \approx C|\boldsymbol{u}|^{\frac{1}{n}-1}\boldsymbol{u}, \tag{65}$$

In the Coulomb-friction limit, where the ice is thin and fast-moving, $\kappa|\boldsymbol{u}| \gg N^n$, giving

$$\boldsymbol{\tau_b} \approx \frac{C}{\kappa^{\frac{1}{n}}}N\frac{\boldsymbol{u}}{|\boldsymbol{u}|}. \tag{66}$$

- Pseudo-plastic sliding, as described by Schoof and Hindmarsh (2010) and Aschwanden et al. (2013) and implemented in PISM. The basal friction law is

$$\boldsymbol{\tau_b} = -\tau_c\frac{\boldsymbol{u_b}}{|\boldsymbol{u_b}|^{1-q}u_0^q}, \tag{67}$$

where τ_c is the yield stress, q is a pseudo-plastic exponent, and u_0 is a threshold speed. This law incorporates linear ($q = 1$), plastic ($q = 0$) and power-law ($0 < q < 1$) behavior in a single expression. The yield stress is given by $\tau_c = \tan(\phi)N$, where ϕ is a friction angle that can vary with bed elevation, resulting in a lower yield stress at lower elevations. This scheme has been shown to give a realistic velocity field for most of the Greenland ice sheet with tuning of just a few parameters, instead of adjusting a basal friction parameter in every grid cell (Aschwanden et al., 2016).

The power-law, Coulomb-friction and pseudo-plastic sliding laws give lower basal friction and faster sliding as the effective pressure N decreases from the overburden pressure. CISM offers several options for computing N at the bed:

- $N = \rho_i gH$, the full overburden pressure. That is, the water pressure $p_w = 0$ at the bed.

- N is reduced as the basal water depth increases, reaching a small fraction of overburden pressure (typically $\delta = 0.02$) when the water depth reaches a prescribed threshold.

- Following Leguy et al. (2014), N is reduced where the bed is below sea level, to account for partial or full connectivity of the basal water system to the ocean. The effective pressure is given by

$$N = \rho_i gH \left(1 - \frac{H_f}{H}\right)^p, \tag{68}$$

where $H_f = \max(0, -\rho_o b/\rho_i)$ is the flotation thickness, ρ_o is seawater density, b is the bed elevation (negative below sea level), and $0 \leq p \leq 1$. For $p = 0$ there is no connectivity and N is the full overburden pressure. For $p = 1$ there is full connectivity, and the basal water pressure is equal to the ocean pressure at the same depth.

Glissade does not yet support sophisticated subglacial hydrology. (The model of Hoffman and Price (2014) was implemented in serial in an earlier version of CISM, but is not currently supported.) The following (non-conserving) options are available for handling basal meltwater:

- All basal water immediately drains.

- Basal water depth is set to a constant everywhere, to force $T = T_{\text{pmp}}$.

- Basal water depth is computed using a local till model, as described by Bueler and van Pelt (2015). In this model, water depth W evolves according to

$$\frac{\partial W}{\partial t} = \frac{-B_b}{\rho_w} - C_d, \tag{69}$$

where W is the water depth (capped at $W_{\max}$), B_b is the basal mass balance (negative for melting), ρ_w is the density of water, and C_d is a fixed drainage rate. The effective pressure N is related to water depth by

$$N = \min\left[P_0, N_0\left(\frac{\delta P_0}{N_0}\right)^{W/W_{\max}} 10^{(e_0/C_c)(1 - W/W_{\max})}\right], \tag{70}$$

where e_0 is the void ratio at reference effective pressure N_0, C_c is the till compressibility, and δ is a scalar between 0 and 1. Default values of these terms are taken from Bueler and van Pelt (2015). The effect of (70) is to drive N from P_0 down to δP_0 as W increases from 0 to $W_{\max}$.

3.5 Iceberg calving

Glissade supports several schemes for calving ice at marine margins. Two of these are very simple: (1) Calve all floating ice, and (2) Calve ice where the bedrock (either the current bedrock, or the bedrock toward which a viscous asthenosphere is

relaxing) lies below a prescribed elevation. In addition, Glissade includes several new calving schemes: mask-based calving, thickness-based calving, and eigencalving.

With mask-based calving, any floating ice that forms in cells defined by a calving mask is assumed to calve instantly. By default, the mask is based on the initial ice extent. Floating ice calves in all cells that are initially ice-free ocean, and thus the calving front cannot advance (but it can retreat). Alternatively, users can define a calving mask that is read at initialization.

Thickness-based calving is designed to remove floating ice that is thinner than a user-defined thickness, $H_c^{\min}$. As discussed by Albrecht et al. (2011), accurate thickness-based calving requires a subgrid-scale parameterization of the calving front. Suppose, for example, that $H_c^{\min} = 100$ m, and an ice shelf with a calving front thicker than 100 m is advancing. During the next time step, ice is transported to grid cells just downstream of the initial calving front. Typically, these cells have $H < H_c^{\min}$. If this thin ice immediately calves, the calving front cannot advance.

Glissade avoids this problem using an approach similar to that of Albrecht et al. (2011). Floating cells adjacent to ice-free ocean are identified as calving-front (CF) cells. For each CF cell, an effective thickness H_{eff} is determined as the minimum thickness of adjacent ice-filled cells not at the CF. CF cells with $H < H_{\text{eff}}$ are deemed to be partially filled. For example, a cell with $H = 50$ m and $H_{\text{eff}} = 100$ m is considered to be half-filled with 100-m ice. It is dynamically inactive and thus cannot transport ice downstream. It can thicken, however, as ice is transported from active cells upstream. Once the ice in this cell thickens to $H \geq H_{\text{eff}}$, it becomes dynamically active. The downstream cells, previously ice-free, then become inactive CF cells and can thicken in turn. In this way the calving front can advance. Similarly, the calving front can retreat when an inactive CF cell becomes ice-free; its upstream neighbor, formerly an active cell in the shelf interior, becomes an inactive CF cell.

Thickness-based calving is applied not to CF cells with $H < H_c^{\min}$, but rather to cells with $H_{\text{eff}} < H_c^{\min}$. In other words, the CF thickness is inferred from the active cells just interior to the CF. Where $H_{\text{eff}} < H_c^{\min}$, the effective rate of thickness change is given by

$$\frac{dH_{\text{eff}}}{dt} = -\frac{\left(H_c^{\min} - H_{\text{eff}}\right)}{t_c}, \tag{71}$$

where t_c is a calving time scale. This effective rate is converted to dH/dt (i.e., the rate of ice volume change per unit cell area) as

$$\frac{dH}{dt} = \min\left(H/H_{\text{eff}}, 1\right)\frac{dH_{\text{eff}}}{dt}. \tag{72}$$

Glissade's eigencalving scheme is related to the method described by Levermann et al. (2012), but differs in the details. In the method of Levermann et al. (2012), the lateral calving rate is proportional to the product of the two eigenvalues of the horizontal strain rate tensor, provided that both eigenvalues are positive. This scheme was tested in CISM but found to give noisy, erratic results. Instead, we compute the lateral calving rate c as

$$c = k_\tau \tau_{ec}, \tag{73}$$

where k_τ is an empirical constant with units of m yr^{-1} Pa^{-1}, and τ_{ec} is an effective calving stress defined by

$$\tau_{ec}^2 = \max\left(\tau_1, 0\right)^2 + w_2 \max\left(\tau_2, 0\right)^2. \tag{74}$$

Here, τ_1 and τ_2 are the eigenvalues of the 2D horizontal deviatoric stress tensor, and w_2 is an empirical constant. The stresses τ_1 and τ_2 are positive when the ice is in tension; τ_1 corresponds to along-flow tension and τ_2 to across-flow tension. These stresses are computed for dynamically active cells and then are extrapolated to inactive neighbors. When $w_2 > 1$, across-flow tension drives calving more effectively than does along-flow tension, as is the case for the calving law of Levermann et al. (2012). Eq. (74) is analogous to eq. 6 in Morlighem et al. (2016) (in a calving law for Greenland's Store Gletscher), but with stress replacing strain rate and with a weighting term added.

Using (73) and (74), we can compute a lateral calving rate c for each CF cell. The lateral calving rate is converted to a rate of volume change using

$$\frac{dH}{dt} = -\frac{H_{\text{eff}}c}{\sqrt{\Delta x \Delta y}}, \tag{75}$$

where Δx and Δy are the grid cell dimensions. Typically, CISM is run with $\Delta x = \Delta y$, but this is not a model requirement.

Eigencalving, when applied on its own, can allow very thin ice to persist at the calving front where stresses are small. For this reason, the eigencalving algorithm is automatically followed by thickness-based calving as described above, with H_c^{min} set to a value large enough to remove thin ice, where present, but not so large as to be the dominant driver of calving.

An additional option related to calving is iceberg removal. An iceberg is defined as a connected region where every cell is floating and has no connection to grounded ice. Icebergs can be present in initial data sets and also can arise during simulations, depending on the calving scheme. Finding the velocity for such a region is an ill-posed problem, so it is best to remove icebergs as soon as they occur. This is done by first marking all the grounded ice cells, and then using a parallel flood-fill algorithm to mark every floating cell that is connected to a grounded cell. (That is, one could travel from a grounded cell to a given floating cell along a path at least one cell wide, without leaving the ice sheet.) Any unmarked floating cells then calve immediately.

Another option is to limit the thickness of ice cliffs, defined as grounded marine-based cells adjacent to ice-free ocean. Bassis and Walker (2012) pointed out that there is an upper thickness limit for marine-terminating cliffs of outlet glaciers. If the ice cliff sits more than ~ 100 m out of the water, the longitudinal stresses exceed the yield strength, and the ice will fail. The maximum stable thickness at the terminus is given by

$$H_{\text{max}} = \frac{\tau_c}{\rho_i g} + \sqrt{\left(\frac{\tau_c}{\rho_i g}\right)^2 + \frac{\rho_o}{\rho_i}d^2}, \tag{76}$$

where $\tau_c \sim 1$ MPa is the depth-averaged shear strength, d is the ocean depth, and ρ_i and ρ_o are densities of ice and seawater. When cliff limiting is enabled in CISM, marine-grounded cells adjacent to ice-free ocean are limited to $H \leq H_{\text{max}}$, with any excess thickness added to the calving flux. This thinning mechanism does not trigger the rapid ice sheet retreat seen by Pollard et al. (2015) in Antarctic simulations that combined cliff failure with hydrofracture. CISM2.1 does not simulate hydrofracture or lateral cliff retreat.

3.6 Sub-shelf melting

By default, melting beneath ice shelves is set to zero. The following simple options may be enabled for simulations with marine ice:

- Sub-shelf melting is set to a constant value for all floating ice.

- CISM reads in a 2D field of basal melt rates and applies the prescribed rates to floating ice.

- Sub-shelf melting is prescribed as for the MISMIP+ Ice1 experiments described by Asay-Davis et al. (2016). In this parameterization, the melt rate is set to zero above a reference elevation z_0, then increases linearly as a function of depth. In shallow sub-shelf cavities, the melt rate is reduced to zero over a characteristic length scale H_0. The depth dependence is motivated by observations of temperature increasing with depth in polar regions, but is not necessarily a realistic treatment of melting near grounding lines.

- The sub-shelf melt rate is a piecewise linear function of depth. Above a reference elevation z_0 there is freezing, and below z_0 there is melting. Above z_0 the freezing rate increases linearly from 0 to B_{frz}^{max} at z_{frz}^{max}; above z_{frz}^{max} it is capped at B_{frz}^{max}. Similarly, below z_0 the melt rate increases linearly from 0 to B_{mlt}^{max} at z_{mlt}^{max}; below z_{mlt}^{max} it is capped at B_{mlt}^{max}. In shallow cavities, the melt rate is reduced to zero over a scale H_0 as for MISMIP+. Like the MISMIP+ parameterization, this is an ad hoc scheme that should be used with caution. More realistic treatments of sub-shelf melting are being actively developed in the ocean and ice modeling communities (Asay-Davis et al., 2017).

Sub-shelf melting is applied only to cells that are floating based on the criterion $b < -(\rho_i/\rho_o)H$, where b is bed elevation (negative below sea level) and H is ice thickness. In partly filled cells at the calving front, basal melt is applied to the effective thickness H_{eff} rather than the grid cell mean thickness H (see Section 3.5). For example, a melt rate of 10 m/yr applied to a cell that is 10% full would reduce H at a rate of only 1 m/yr. Basal melting for grounded ice, including marine-based ice, is described in Section 3.2.3.

3.7 Isostasy

Isostastic adjustment is handled as in Rutt et al. (2009), with several options for the lithosphere and underlying asthenosphere. The lithosphere can be described as either local (i.e., floating on the asthenosphere) or elastic (taking flexural rigidity into account). The asthenosphere is either fluid (reaching isostatic equilibrium instantaneously) or relaxing (responding to the load on an exponential time scale of several thousand years). The default, when isostasy is active, is an elastic lithosphere and relaxing asthenosphere.

In the elastic lithosphere calculation, data from many grid cells must be summed to compute the load in each location. For parallel runs, this is done by gathering data to one processor to compute the load and then distributing the result to the local processors. This calculation scales poorly, but it is done infrequently (every ~ 100 model years) and therefore has a minimal computational cost at grid resolutions of ~ 5 km.

4 Model results: Standard test cases

Test cases for CISM include (1) problems with analytic solutions, (2) standard experiments without analytic solutions but for which community benchmarks are available, and (3) some CISM-specific experiments used for code development and testing.

These tests are organized in directories, each of which includes a README file with instructions on running the test. Most tests have Python scripts that are used to set up the initial conditions and run the model, and some tests have an additional Python script to analyze and plot the output. Each test has a default configuration file, whose settings can be adjusted manually to make changes (e.g., swapping velocity solvers). Several tests are described below, with plots generated by Python scripts included in the code release.

4.1 Halfar dome test

The Halfar test case describes the time evolution of a parabolic dome of ice (Halfar, 1983; Bueler et al., 2005). For a flat-bedded SIA problem without accumulation, this case has an analytic solution for the time-varying ice thickness. The SIA ice evolution equation can be written as

$$\frac{\partial H}{\partial t} = \boldsymbol{\nabla} \cdot (\Gamma H^{n+2} |\boldsymbol{\nabla} H|^{n-1} \boldsymbol{\nabla} H), \tag{77}$$

where n is the exponent in the Glen flow law, commonly taken as 3, and $\Gamma = 2A(\rho g)^n/(n+2)$ is a positive constant. For $n = 3$, the time-dependent solution is

$$H(t,r) = H_0 \left(\frac{t_0}{t}\right)^{\frac{1}{9}} \left[1 - \left(\left(\frac{t_0}{t}\right)^{\frac{1}{18}} \frac{r}{R_0} \right)^{\frac{4}{3}} \right]^{\frac{3}{7}}, \tag{78}$$

where

$$t_0 = \frac{1}{18\Gamma} \left(\frac{7}{4}\right)^3 \frac{R_0^4}{H_0^7}, \tag{79}$$

and H_0 and R_0 are the central dome height and dome radius, respectively, at time $t = t_0$. As the dome evolves, the ice margin advances and the thickness decreases. The test can be run using either Glide or Glissade solvers.

The top panels of Fig. 1 show Halfar dome results for the Glide solver with $H_0 = 500\sqrt{2} \approx 700$ m, $R_0 = 15\sqrt{2} \approx 21$ km, a grid resolution of 2 km, and a time step of 5 yr. The three panels show the modeled thickness, analytic thickness, and thickness difference, respectively, at $t = 200$ yr. Differences are largest near the ice margin. The rms error in the solution is 6.43 m, with a maximum absolute error of 32.8 m.

The bottom panels of Fig. 1 show Halfar results for the Glissade SIA solver described in Section 3.1.2, with the same grid and initial conditions and a time step of 1 yr. (Glissade's explicit transport solver requires a shorter time step than Glide for numerical stability.) The errors are larger than for Glide; the rms error is 9.06 m and the maximum error is 34.2 m. Since Glissade's SIA velocity solutions are very similar to Glide's for a given ice sheet geometry, the larger errors can be attributed to the explicit transport solver (Section 3.3), which is less accurate for this problem than Glide's implicit diffusion-based solver.

4.2 ISMIP-HOM tests

The Ice Sheet Model Intercomparison Project for Higher-Order Models (ISMIP-HOM), described by Pattyn et al. (2008), consists of six experiments, labeled A through F, for higher-order velocity solvers. CISM supports all six experiments, and

here we show results for Experiments A and C. These two tests are particularly useful for benchmarking higher-order models, since they gauge the accuracy of simulated 3D flow over a bed with large- and small-scale variations in basal topography and friction.

Experiment A is a test for ice flow over a bumpy bed. The domain is doubly periodic, and the bed topography is sinusoidal in both the x and y directions, with an amplitude of 500 m and wavelength $L =$ 5, 10, 20, 40, 80 or 160 km. The mean ice thickness is 1000 m, and the surface slopes smoothly downward from left to right. The velocity field is diagnosed from this geometry given a no-slip basal boundary condition. Fig 2 shows the surface ice speed as a function of x along the bump at $y = L/4$, computed using Glissade's BP solver. For $L = 10$ km or more, the solution agrees well with the mean from the Stokes models in Pattyn et al. (2008). At $L = 5$ km the amplitude is too large (as is typical of BP models), but the magnitude is close to the Stokes solution.

Fig. 3 shows test A results using Glissade's DIVA solver. (Cf. Fig. 1 in Goldberg (2011), which shows similar results from a depth-integrated flowline solver.) The DIVA solution is much less accurate than the BP solution; the modeled ice speed is greater than for the Stokes solution, with the relative error increasing as L decreases. These errors are expected, since a bumpy, frozen bed implies that the horizontal velocity gradients are not well approximated (as DIVA assumes) by their vertical averages. The DIVA solution is more accurate, however, than the L1L2 scheme implemented by Perego et al. (2012). The latter scheme does a 2D matrix solve for the basal velocity and then integrates upward, reducing to the very inaccurate SIA when the basal velocity is zero. DIVA, by solving for the 2D mean velocity instead of the basal velocity, is more accurate than the SIA.

Experiment C is a test for flow over a bed with variable friction. The ice surface and mean thickness are as in Experiment A, but the bed is flat, with sinusoidal variations in the basal friction parameter β. Fig. 4 shows the surface speed at $y = L/4$ for each of six wavelengths using the BP solver, and Fig. 5 shows the surface speed using the DIVA solver. (Cf. Fig. 2 from the depth-integrated flowline solver in Goldberg (2011).) For both BP and DIVA, the CISM results are similar to the Stokes results at all wavelengths. There is a modest difference between BP and DIVA at $L = 5$ km, where the DIVA velocity has a small variation across the domain, whereas the BP and Stokes velocities are nearly uniform.

For both Experiments A and C, the Glissade BP results are nearly identical to those shown by Tezaur et al. (2015), who used similar finite-element methods to assemble and solve the BP equations.

4.3 Stream tests

CISM's stream tests simulate flow over an idealized ice stream underlain by subglacial till with a specified yield stress distribution. For the two yield stress distributions specified in this test case, analytical solutions are available from Raymond (2000) and Schoof (2006). For the Raymond test case, the yield stress is given within the ice stream by a uniform value of 5.2 kPa (below the gravitational driving stress of about 10 kPa), and outside the ice stream by a uniform value of 70 kPa (much larger than the driving stress). That is, the yield stress distribution is approximated by a step function. For the Schoof test case, the yield stress across the ice stream varies continuously between about 45 kPa and 0. In both cases, the basal properties and resulting velocity solutions vary in the across-flow direction only and are symmetric about the ice stream centerline.

Figs. 6 and 7 compare model results to the analytic Raymond and Schoof solutions, respectively. CISM is run using the BP velocity solver at a grid resolution of 2 km. (DIVA results, not shown, are nearly identical.) The top panels show across-flow velocity profiles, and the bottom panels show the prescribed yield stress and driving stress. In both cases there is excellent agreement between the modeled and analytic solutions. For the Raymond test, this excellent agreement requires a modified assembly procedure for basal friction terms, as described in Section A2, to resolve the step change in yield stress. With the standard finite-element procedure, there is some smoothing of the friction parameter β over neighboring vertices, and the results at a given grid resolution are less accurate. For the Schoof test with a smooth transition in yield stress, either assembly procedure works well.

4.4 Shelf tests

CISM includes three shelf tests:

- a confined-shelf test based on tests 3 and 4 from the Ice Shelf Model Intercomparison exercise (Rommelaere, 1996). This test simulates the flow of a 500-m-thick ice shelf within a rectangular embayment that is confined on three sides. Fig. 8 shows the 2D ice speed using Glissade's SSA solver (Section 3.1.3) on a 5-km mesh. As expected, the solution is nearly identical to those obtained with the DIVA and BP solvers (not shown). No benchmark is available, but the Glissade solutions are similar to the first-order (BP) solutions found by Perego et al. (2012) and Tezaur et al. (2015) for the same test.

- a circular-shelf test that simulates the radially symmetric flow of a 1000-m-thick ice shelf that is pinned at one point in the center. The SSA solution (not shown) is similar but not identical to the DIVA and BP solutions, since the latter allow vertical shear above the pinning point.

- a test that simulates the flow of the Ross Ice Shelf in Antarctica under idealized conditions (e.g., a spatially uniform flow-rate factor), as described by MacAyeal et al. (1996). Fig. 9, which is based on Figs. 1 and 2 of MacAyeal et al. (1996), shows the ice speed computed by Glissade's SSA solver on the 6.8-km finite-difference grid used in that paper, compared to observed ice speeds. The model velocities agree well with the published velocities from the finite-element model in Fig. 3A of MacAyeal et al. (1996) and, like the published values, tend to be biased fast compared to observations. As noted by MacAyeal et al. (1996), the model–data agreement could be improved by allowing the flow-rate factor to vary spatially. Results from the DIVA and BP solvers (not shown) are nearly identical.

4.5 Dome test

CISM's dome test has a simple configuration with a parabolic, radially symmetric dome on a flat bed. It is a general-purpose time-dependent problem that can be used to test not just the velocity solver, but also the transport and temperature solvers and various physics options. There is no analytic solution, but the test has proven useful for day-to-day testing.

4.6 Build and test structure

To facilitate testing, CISM includes a build and test structure (BATS) that can automatically build the model and then run a set of regression tests, including the tests discussed above. BATS allows users and developers to quickly generate a set of regression tests for use with the Land Ice Verification and Validation toolkit (LIVVkit; Kennedy et al. 2017). LIVVkit is designed to provide robust and automated numerical verification, software verification, performance validation, and physical validation analyses for ice sheet models. Instructions on using using BATS and LIVV with CISM can be found on the LIVV website.

5 Model results: Greenland ice sheet simulations

The CISM v2.0 release included the tests described in Section 4 but did not support robust multi-millennial simulations of whole ice sheets. Recent work, included in CISM v2.1, has made the model more efficient, reliable, and realistic for Greenland ice sheet simulations. These improvements support the use of CISM in CESM2 for century-to-millennial-scale Greenland simulations under paleoclimate (e.g., Pliocene and Eemian), present-day, and future climate conditions.

Coupled ice-sheet/climate simulations can be problematic, because ice sheets require 10^4 years or longer to equilibrate with a given climate (Vizcaino, 2014), and because climate is never truly constant on these time scales. If the initial ice sheet conditions in a coupled simulation are not consistent with the climate, then the transient adjustment can swamp any climate change signal. One way to minimize the adjustment is to adjust selected model parameters (e.g., basal traction coefficients or bed topography at each grid point) to satisfy an optimization problem, reducing the mismatch between model variables and observations. In this way, one can generate ice sheet thickness and velocity fields consistent with the SMB from a climate model (Perego et al., 2014). The risk of this approach is that the model can be over-tuned to present-day conditions that might not hold on long time scales or in different climates. An alternate approach is to spin up the ice sheet to equilibrium under forcing from a climate model: e.g., steady pre-industrial forcing, or forcing spliced together from two or more climate time slices (Fyke et al., 2014). Because of climate model biases, however, the spun-up ice thickness and velocity can differ substantially from modern observations.

Here we take the second approach, spinning up the Greenland ice sheet with surface forcing from the regional climate model RACMO2.3p1 statistically downscaled to 1 km resolution (Noël et al., 2016). Since RACMO2 is run at high resolution, is constrained by reanalysis at model boundaries, and is well validated against observations, its SMB is more realistic than that of a global climate model. RACMO2 provides the SMB only for the region included within its ice sheet mask; outside this region, we prescribe a negative SMB. The ice sheet is initialized with present-day extent and thickness, and then is spun up for 50,000 years (long enough to reach quasi-equilibrium) on a 4-km grid (the standard grid for CESM ice-sheet simulations).

We analyze two experiments. The first experiment uses a "no-float" calving scheme, in which floating ice is assumed to calve immediately. A similar configuration was used to initialize CISM for the initMIP-Greenland experiment (Goelzer et al., 2018). The no-float assumption simplifies model set-up and analysis, while still simulating ice flow for the vast majority of the Greenland ice sheet. The second experiment uses the eigencalving scheme described in Section 3.5, along with the depth-

dependent sub-shelf melting scheme of Section 3.6. This simulation tests the model's ability to generate robust, stable ice shelves that at least roughly resemble Greenland's present-day ice shelves, including the floating shelf of Petermann Glacier in northern Greenland and two floating termini of the Northeast Greenland Ice Stream (NEGIS): Nioghalvfjerdsbræ (79 North) and Zachariae Isstrom.

5   Although the model has been configured to give results that are reasonably realistic, these spin-ups should be viewed primarily as a demonstration of CISM capabilities, not as scientifically validated simulations. For example, model tuning has likely compensated for biases in the forcing data. Generally speaking, model parameters should always be tested and reviewed depending on the science application.

5.1   Simulation without ice shelves

10   The simulation without floating ice shelves is configured as follows:

- The model is initialized with present-day ice sheet geometry. The ice thickness and bed topography are based on the mass-conserving-bed method of Morlighem et al. (2011, 2014).

- The SMB forcing over the Greenland ice sheet is a 1958–2015 climatology from RACMO2.3p1 (Noël et al., 2016). For ice-free regions where RACMO2 does not compute an SMB, the SMB is arbitrarily set to -4 m/yr, thus limiting
15   deviations from present-day ice extent. The surface temperature is from the 20th century RACMO2 climatology of Ettema et al. (2009).

- The basal geothermal heat flux is prescribed from Shapiro and Ritzwoller (2004).

- Where the SMB is negative, the initial temperature profile in each column is linear, with $T = \min(T_{air}, 0)$ at the surface and $T = T_{pmp} - 5°$ at the bed. Where the SMB is positive, the temperature is initialized to an analytic profile based
20   on a balance between vertical conduction and cold advection (Cuffey and Paterson 2010, Section 9.5.1). This profile is relatively cold in the upper part of the ice sheet.

- The 3D velocity field is computed at 11 vertical levels using the DIVA solver described in Section 3.1.4. Thickness and tracer evolution are given by the incremental remapping scheme of Section 3.3, with a time step $dt = 0.2$ yr.

- The basal friction coefficient β is computed at each vertex using a pseudo-plastic sliding law as described in Section 3.4.
25   This scheme has several parameters that can be tuned to reduce effective pressure and increase sliding in warm, low-elevation regions. Parameter values, mostly following Aschwanden et al. (2016), are $q = 0.5$ and $u_0 = 100$ m/yr, with ϕ decreasing from $40°$ to $5°$ as the bed elevation decreases from 700 m to -700 m.

- Effective pressure is a function of basal water depth as computed by the local till model described in Section 3.4 with till parameters from Bueler and van Pelt (2015). In particular, $W_{\text{max}} = 2$ m, $C_d = 1$ mm/yr, and $\delta = 0.02$.

30   - Floating ice calves immediately, and marine cliff heights are limited as described in Section 3.5.

– Some of the model physics is constrained to make the simulation more robust. The surface gradients $\partial s/\partial x$ and $\partial s/\partial y$ in the gravitational driving stress are limited to a magnitude of 0.10 to prevent unrealistically large ice speeds in coastal regions with steep topography, and β is held to a minimum of $100\ \mathrm{Pa\ m^{-1}\ yr}$ for grounded ice to prevent very fast sliding.

5 After 50,000 model years the ice sheet extent and volume have equilibrated to $1.63 \times 10^6\ \mathrm{km^2}$ and $2.97 \times 10^6\ \mathrm{km^3}$, respectively, in close agreement with values of $1.67 \times 10^6\ \mathrm{km^2}$ and $2.95 \times 10^6\ \mathrm{km^3}$ in the observational data set (Morlighem et al., 2014). The close agreement between observed and modeled extent and volume depends, in part, on the negative SMB applied outside the RACMO2 ice-sheet mask to inhibit ice advance. We checked for equilibrium by comparing thickness snapshots at 500-year intervals near the end of the run. Thickness variations over the last 1,000 years (not shown) are less than 50 m

10 everywhere on the ice sheet.

Figure 10 shows the surface ice speed at the end of the run, compared to observations. In most of the ice sheet the simulated ice flow is in good agreement with observations. The model captures slow-flowing regions in the interior where the bed is frozen, as well as fast-flowing outlet glaciers along the margins. A major shortcoming of the simulation is its failure to capture NEGIS, which in reality extends far into the northeast interior but is simulated to be slower, shorter and more diffuse.

15 Aschwanden et al. (2016), who ran PISM using the pseudo-plastic basal sliding scheme adopted by CISM, obtained similar results. While capturing complex outlet glacier flow, both models have a poor representation of NEGIS, possibly due to a missing geothermal heat source or simplified subglacial hydrology.

Figure 11 shows the difference between the final ice thickness and the observed initial thickness. In most of the interior, the thickness errors are ~ 100 m or less, with the main exception being the upper part of NEGIS, where slow flow correlates with a

20 positive thickness bias of ~ 200 m. Errors are larger along the margins, with positive biases of up to ~ 500 m in the southeast, where accumulation is large, and negative biases of up to ~ 500 m in the north and northwest, where accumulation is lower and there is significant ablation at the margins. Low thicknesses along the coast could result from excessive marginal ablation in the SMB dataset, insufficient basal friction, or a combination. High thickness in the southeast could be attributed to positive accumulation biases in the dataset or too-slow flow in outlet glaciers, among other factors.

25 Fig. 12 illustrates the basal water state. The ice sheet is shaded to indicate three regions: a frozen central region (extending to parts of the northern and eastern margin) with basal water depth $W = 0$; thawed coastal regions, especially along the western margin and in major outlet glacier basins where W is capped at 2 m; and intermediate regions where $0 < W < 2$ m. The color scheme in Fig. 12 was chosen for comparison to Fig. 11 from MacGregor et al. (2015), who presented a synthesis of Greenland's basal state from observations and models. The agreement is generally excellent (though the two results are not

30 fully independent, since models similar to CISM were part of the synthesis). The largest area of discrepancy is the upstream part of NEGIS, which is thawed or uncertain in the synthesis but frozen in the model.

The intermediate regions in Fig. 12 have striped patterns with century-scale temporal variability, consistent with the band-like patterns in driving and basal shear stresses found by Sergienko et al. (2014) in regions with rapid basal sliding. Sergienko et al. (2014) attributed these patterns to instabilities related to subglacial water. A more detailed investigation would be needed

to determine how robust these features are in the model, and to what extent they are real features as opposed to numerical artifacts.

As shown by Schoof and Hindmarsh (2010), depth-integrated approximations like DIVA are computationally much cheaper than BP while incurring only a small loss of accuracy, mainly in slow-sliding regions. However, their analysis did not consider the effects of vertically varying temperature. In CISM simulations of real ice sheets, the depth-integrated viscosity depends on the vertical temperature profile (cf. (2) and (24)), with dynamic effects that are not included in idealized problems with a uniform temperature or flow factor. To assess possible errors associated with the DIVA solver (taking BP to be "truth"), we ran CISM for 10,000 years using the BP solver, with settings and forcing otherwise identical to the DIVA run. After 10 kyr, the ice sheet area and volume are 1.635×10^6 km^2 and 2.991×10^6 km^3, respectively, for the BP run, compared to 1.634×10^6 km^2 and 2.993×10^6 km^3 for DIVA. Figs. 13 and 14 show differences in thickness and surface ice speed, respectively, between the two runs. In the vast majority of the ice sheet, thickness differences are less than 50 m, and velocity differences are less than 20 m/yr. In several outlet glaciers, however—notably Humboldt Glacier in the northwest—thickness differences exceed 100 m, and velocity differences are greater than 100 m/yr. These differences might merit further study. Some, although not all, of the larger differences can be attributed to transient behavior. For example, Humboldt Glacier is not fully equilibrated at 10 kyr, and the two simulations could capture the glacier in different phases.

Table 3 shows the cost of these runs on 144 and 288 cores of NCAR's Cheyenne supercomputer. On 144 cores, the DIVA run without ice shelves completes 1350 model yr/wall-clock-hr, or 9.41 model yr/core-hr. Running on 288 cores, the throughput increases to 2480 model yr/wall-clock-hr, at slightly reduced efficiency of 8.61 model yr/core-hr. With the BP solver and an otherwise identical configuration (including 11 vertical velocity levels), the throughput decreases by nearly a factor of 50. Table 3 also shows the cost of DIVA runs with ice shelves, which are described next. Although we have found that solver convergence can be slower when floating ice is present, the runs presented here have a similar cost with or without ice shelves.

5.2 Simulation with ice shelves

The simulation with ice shelves is configured like the no-shelf simulation, but with these differences:

- No-float calving is replaced by the eigencalving scheme of Section 3.5, with $k_\tau = 0.0025$ m yr^{-1} Pa^{-1} and $w_2 = 25$. These parameters give a low calving rate for along-flow tensile stresses and a much higher rate for across-flow stresses.

- Beneath floating ice we apply the depth-dependent melt rate of Section 3.6, with a basal mass balance $b = 0$ at $z = -200$ m. The freezing rate is 3 m/yr above $z = -100$ m, with a linear ramp between -200 m and -100 m. The melt rate is 100 m/yr below $z = -500$ m, with a linear ramp between -200 m and -500 m. The scale for reducing melting in shallow cavities is $H_0 = 20$ m.

Choi et al. (2017) used a similar combination of stress-based calving and depth-dependent sub-shelf melting in ISSM to simulate glacier evolution in the NEGIS region. For the CISM simulations, calving and melting parameters were tuned to improve agreement with Greenland's observed ice shelves (or lack thereof). For example, higher values of k_τ were found to remove existing ice shelves in northern Greenland. With the chosen value of k_τ, lower values of w_2 are insufficient to

calve long, unrealistic ice tongues along the southeast coast. Similarly, sub-shelf freezing is needed near the surface to prevent substantial calving-front retreat for Petermann and 79 North. While near-surface freezing is physically plausible, basal freezing in the simulation might be compensating for a negative SMB bias (Noël et al., 2018) or excessive calving. A maximum melt rate of 100 m/yr exceeds inferred melt rates of ~ 30 m/yr near the grounding line (GL) of Petermann Glacier (Cai et al., 2017),

5   but smaller rates in the model permit unrealistic GL advance. Simulated GL locations could differ, however, if the model were run at different resolution or with a grounding-line parameterization (Gladstone et al., 2010; Leguy et al., 2014).

Fig. 15 shows (for grounded ice) the thickness difference between the simulation and observations. Simulated floating ice is shaded light green, and observed floating ice is enclosed by black contours. The pattern of thickness errors is generally similar to the run without ice shelves (Fig. 11), with positive biases in the northeast interior and along the southeast coast, and negative biases along the northern and northwest margins. The run with shelves, however, has additional positive biases in east central

15   and west central Greenland, upstream of large outlet glaciers such as Jakobshavn in the west and Helheim and Kangerlussuaq in the southeast. The basins that have thickened relative to the no-shelf simulation generally have ice shelves at their termini, where no shelves exist in reality. Although these shelves are small, they appear to provide buttressing that leads to thickening upstream. Fig. 16 shows the difference in surface ice speed between the simulations with and without ice shelves. Differences are largest in outlet glaciers with unrealistic floating termini. The general pattern is of faster speeds near glacier termini in the

20   run with shelves (where ice can be much thicker than in the no-shelf simulation), with slower speeds upstream.

Fig. 17 shows closeups of three regions enclosed by boxes in Fig. 15. These regions illustrate floating ice simulated in the vicinity of Petermann Glacier in the northwest, 79 North/Zachariae Isstrom in the northeast, and Kangerlussuaq Glacier in the southeast. The Petermann simulation (top panel) is the most realistic, with modest retreat of both the grounding line and calving front, and a total area similar to observations. As mentioned above, however, both the CF and GL locations are

25   sensitive to model parameters. The 79 North ice shelf (upper left of center panel) shows modest GL retreat with somewhat larger CF retreat. The simulated Zachariae shelf (lower right of center panel) substantially retreats; we found no combination of calving and basal melting/freezing parameters that maintains Zachariae in its observed location. The bottom panel, showing Kangerlussuaq Glacier, illustrates the model's tendency to grow unobserved shelves. While there is little floating ice in this region in reality, there is a moderate-sized simulated shelf at the fjord mouth, buttressing the glacier upstream.

30   These results demonstrate a modeling challenge for Greenland: how to simulate existing ice shelves in northern Greenland without permitting unrealistic shelves in other regions. An eigencalving parameterization, suitably tuned, can prevent shelves from extending into the open ocean, but in enclosed bays there are unrealistic shelves with non-negligible dynamic effects. We were unable to eliminate the unobserved shelves in CISM without also removing observed shelves. This result suggests that important calving or melting mechanisms could be missing in the model. For example, Bassis and Ma (2015) suggested that

basal melting, which is spatially variable, is a major driver of crevassing and calving. Lacking ocean forcing, the model does not simulate spatially dependent basal melt rates.

Despite these biases, the results show that CISM is capable of simulating ice velocities broadly consistent with observations for both slow-flowing and fast-flowing parts of the Greenland ice sheet, with reasonable computational costs for multi-millennial simulations with a higher-order Stokes approximation.

6 Conclusions

We have described CISM v2.1, which includes many innovations to support robust, accurate, and efficient ice sheet simulations in both idealized and real-world applications. The model incorporates a hierarchy of Stokes approximations, including SIA, SSA, depth-integrated higher-order, and 3D higher-order. To solve the large elliptic systems associated with membrane stresses, CISM has an efficient native-Fortran preconditioned conjugate gradient solver, along with links to third-party solver libraries (Trilinos and SLAP). CISM also adds test cases for higher-order models, including ISMIP-HOM and various shelf and stream cases. For each velocity solver and test case, we have verified that CISM solutions are consistent with community benchmarks. That is, the solutions are as accurate as expected given the simplifications in the various approximations.

CISM's structured rectangular grid has limitations; in particular, the grid cannot be refined selectively near grounding lines and other regions requiring high resolution. On the other hand, the structured grid lends itself to straightforward algorithm development, debugging, creation of forcing data sets, and analysis of results. CISM runs efficiently on a 4-km Greenland grid, with throughput of ~ 2000 model years per wall-clock hour on NCAR's Cheyenne supercomputer using the depth-integrated higher-order solver. Performance improvements might be needed, however, to support whole-ice-sheet simulations on grids as fine as 1 or 2 km.

CISM has participated in the initMIP–Greenland experiments (Goelzer et al., 2018) for model initialization. When spun up for 50 kyr with modern climate forcing and without floating ice, CISM gives a steady-state simulation in good agreement with the observed ice extent, volume, and velocity structure of the Greenland ice sheet. The quality of the simulation can be attributed, in part, to a higher-order solver that simulates a realistic stress state in fast outlet glaciers, along with a pseudo-plastic basal sliding law that enhances sliding on thawed beds at low elevation. The simulation also benefits from an accurate SMB (from the regional climate model RACMO2), adjusted to inhibit ice sheet advance beyond modern boundaries. When floating ice is allowed, CISM can maintain ice shelves that resemble observed shelves in northern Greenland, but the model also simulates many small shelves that do not exist in reality.

CISM can be used for both standalone and coupled ice sheet simulations. CISM v2.1 is the dynamic ice sheet component of CESM2.0, released in June 2018. Compared to earlier CESM versions, CESM2.0 has more sophisticated ice sheet dynamics and physics (as described here), along with interactive coupling capabilities. These improvements will support increasingly realistic simulations of coupled ice sheet evolution.

7   Code availability

Source code for CISM v2.1 can be obtained by downloading a released version from *https://cism.github.io/download.html* or by cloning the code from the public git repository at *https://github.com/CISM/cism*. The CISM2.1 documentation, which includes detailed instructions for downloading and building the code, can be found at *https://cism.github.io/documentation.html*.

Following the release of CESM2.0 in June 2018, new CISM development was moved to the Earth System Modeling Community Portal (ESCOMP). The latest code is available on the public git repository at *https://github.com/escomp/cism*. Current documentation for CISM and for the land-ice implementation in CESM can be found at *http://www.cesm.ucar.edu/models/cesm2/land-ice/*. New developers are welcome.

Appendix A: Matrix assembly for the Blatter-Pattyn approximation

Here we describe the assembly of the terms in (9) into a global matrix $\mathbf{A}$ and vector $\mathbf{b}$. The four terms in (9) describe internal ice stresses, basal friction, lateral pressure, and the gravitational driving force, respectively. We begin with the internal stress term, which is the most complex.

A1   Internal ice stresses

The first term on the LHS of each component of (9) can be rewritten in terms of velocity components:

$$
x : \int_{\Omega} 2\eta \left[2\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \quad \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \quad \frac{1}{2}\frac{\partial u}{\partial z} \right] \left\{ \begin{array}{c} \frac{\partial \varphi}{\partial x} \\ \frac{\partial \varphi}{\partial y} \\ \frac{\partial \varphi}{\partial z} \end{array} \right\},
$$

$$
y : \int_{\Omega} 2\eta \left[\frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \quad 2\frac{\partial v}{\partial y} + \frac{\partial u}{\partial x} \quad \frac{1}{2}\frac{\partial v}{\partial z} \right] \left\{ \begin{array}{c} \frac{\partial \varphi}{\partial x} \\ \frac{\partial \varphi}{\partial y} \\ \frac{\partial \varphi}{\partial z} \end{array} \right\},
$$

$$(A1)$$

where brackets denote row vectors and braces denote column vectors. Glissade evaluates (A1) for each active element. Hexahedral elements have eight nodes, with u and v to be determined at each active node. Inserting the velocity expressions (6) in (A1), we obtain

$$
x : \int_{\Omega} 2\eta \left(\frac{\partial \varphi_i}{\partial x} \left(2\left[\frac{\partial \varphi_j}{\partial x}\right]\{u_j\} + \left[\frac{\partial \varphi_j}{\partial y}\right]\{v_j\} \right) + \frac{\partial \varphi_i}{\partial y}\left(\frac{1}{2}\left[\frac{\partial \varphi_j}{\partial y}\right]\{u_j\} + \frac{1}{2}\left[\frac{\partial \varphi_j}{\partial x}\right]\{v_j\}\right) + \frac{\partial \varphi_i}{\partial z}\left(\frac{1}{2}\left[\frac{\partial \varphi_j}{\partial z}\right]\{u_j\}\right)\right),
$$

$$
y : \int_{\Omega} 2\eta \left(\frac{\partial \varphi_i}{\partial y} \left(2\left[\frac{\partial \varphi_j}{\partial y}\right]\{v_j\} + \left[\frac{\partial \varphi_j}{\partial x}\right]\{u_j\} \right) + \frac{\partial \varphi_i}{\partial x}\left(\frac{1}{2}\left[\frac{\partial \varphi_j}{\partial x}\right]\{v_j\} + \frac{1}{2}\left[\frac{\partial \varphi_j}{\partial y}\right]\{u_j\}\right) + \frac{\partial \varphi_i}{\partial z}\left(\frac{1}{2}\left[\frac{\partial \varphi_j}{\partial z}\right]\{v_j\}\right)\right).
$$

$$(A2)$$

Each row or column vector has eight terms, one for each node of the element. These terms can be evaluated to form a set of four 8×8 element matrices, denoted as $\mathbf{K_{uu}}$, $\mathbf{K_{uv}}$, $\mathbf{K_{vu}}$, and $\mathbf{K_{vv}}$. Each row of an element matrix is associated with u or v at a given node. The columns in this row contain terms linking the node to u or v at the other nodes of the same element (with the diagonal term linking the node to itself).

5   In the x component of (A2), the terms that multiply u_j are given by

$$\int_\Omega \eta \left(4 \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} + \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_j}{\partial y} + \frac{\partial \varphi_i}{\partial z} \frac{\partial \varphi_j}{\partial z} \right) d\Omega. \tag{A3}$$

Letting i and j range from 1 to 8, (A3) gives the 64 terms of the 8×8 element matrix $\mathbf{K_{uu}}$, which links the u value at each node to the u values at all eight nodes. Similarly, the 64 terms of element matrix $\mathbf{K_{uv}}$, which links u at each node to v at each of the eight nodes, are given by

$$\int_\Omega \eta \left(2 \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial y} + \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_j}{\partial x} \right) d\Omega. \tag{A4}$$

Likewise, two 8×8 matrices are associated with the y component of (A2). The terms of $\mathbf{K_{vu}}$ are

$$\int_\Omega \eta \left(2 \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_j}{\partial x} + \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial y} \right) d\Omega, \tag{A5}$$

and the terms of $\mathbf{K_{vv}}$ are

$$\int_\Omega \eta \left(4 \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_j}{\partial y} + \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} + \frac{\partial \varphi_i}{\partial z} \frac{\partial \varphi_j}{\partial z} \right) d\Omega. \tag{A6}$$

15 Because of the symmetry of the underlying PDEs, $\mathbf{K_{uu}}$ and $\mathbf{K_{vv}}$ are symmetric, and $\mathbf{K_{uv}} = \mathbf{K_{vu}}^T$. The terms containing z (i.e., the vertical shear stresses) appear only in $\mathbf{K_{uu}}$ and $\mathbf{K_{vv}}$, whereas the terms containing x and y (i.e., the membrane stresses) appear in all four element matrices.

    In the weak form of the equations, each of the 64 coefficients in each element matrix must be integrated over the element. (Since φ varies over the element, the integrands in (A3)–(A6) have a different value at each point.) The integrals can be 20 computed exactly by evaluating the integrand at each of eight quadrature points and summing over quadrature points.

    We now specify the form of the basis functions and transform these functions to the geometry of the element (which is irregular in the vertical direction because of the sigma coordinate). We can then evaluate the basis function derivatives at quadrature points. Glissade uses trilinear basis functions defined on a reference cube. This cube is centered at the origin $(0,0,0)$ in local reference coordinates $(\hat{x}, \hat{y}, \hat{z})$. The eight nodes of the reference cube are located at $(\hat{x}, \hat{y}, \hat{z}) = (\pm 1, \pm 1, \pm 1)$.

25 By convention, nodes 1–4 are the nodes of the lower face, proceeding counterclockwise from the southwest corner $(\hat{x}, \hat{y}) =$

$(-1, -1)$. Nodes 5–8 are the nodes of the upper face, also moving counterclockwise from the southwest corner. Thus we have

$$\varphi_1 = (1 - \hat{x})(1 - \hat{y})(1 - \hat{z})/8,$$

$$\varphi_2 = (1 + \hat{x})(1 - \hat{y})(1 - \hat{z})/8,$$

$$\varphi_3 = (1 + \hat{x})(1 + \hat{y})(1 - \hat{z})/8,$$

$$\varphi_4 = (1 - \hat{x})(1 + \hat{y})(1 - \hat{z})/8,$$

$$\varphi_5 = (1 - \hat{x})(1 - \hat{y})(1 + \hat{z})/8,$$

$$\varphi_6 = (1 + \hat{x})(1 - \hat{y})(1 + \hat{z})/8,$$

$$\varphi_7 = (1 + \hat{x})(1 + \hat{y})(1 + \hat{z})/8,$$

$$\varphi_8 = (1 - \hat{x})(1 + \hat{y})(1 + \hat{z})/8.$$

$(A7)$

For each n we have $\varphi_n = 1$ at node n, with $\varphi_n = 0$ at the other nodes.

The integrands in (A3)–(A6) are written in terms of real Cartesian coordinates (x, y, z) rather than reference coordinates

5 $(\hat{x}, \hat{y}, \hat{z})$. Spatial derivatives in real coordinates are related to derivatives in reference coordinates by

$$\begin{Bmatrix} \frac{\partial \varphi_n}{\partial \hat{x}} \\ \frac{\partial \varphi_n}{\partial \hat{y}} \\ \frac{\partial \varphi_n}{\partial \hat{z}} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{x}} & \frac{\partial z}{\partial \hat{x}} \\ \frac{\partial x}{\partial \hat{y}} & \frac{\partial y}{\partial \hat{y}} & \frac{\partial z}{\partial \hat{y}} \\ \frac{\partial x}{\partial \hat{z}} & \frac{\partial y}{\partial \hat{z}} & \frac{\partial z}{\partial \hat{z}} \end{bmatrix} \begin{Bmatrix} \frac{\partial \varphi_n}{\partial x} \\ \frac{\partial \varphi_n}{\partial y} \\ \frac{\partial \varphi_n}{\partial z} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial \varphi_n}{\partial x} \\ \frac{\partial \varphi_n}{\partial y} \\ \frac{\partial \varphi_n}{\partial z} \end{Bmatrix},$$

$(A8)$

where $[J]$ is the Jacobian of the transformation between coordinate systems. Given the finite-element expansion

$$x = \sum_n \varphi_n x_n,$$

$(A9)$

along with the spatial derivatives of φ at $(\hat{x}, \hat{y}, \hat{z})$ (which are easily derived from (A7)), we can compute $[J(\hat{x}, \hat{y}, \hat{z})]$ as

10 $$[J] = \begin{bmatrix} \sum_{n=1}^{8} \frac{\partial \varphi_n}{\partial \hat{x}} x_n & \sum_{n=1}^{8} \frac{\partial \varphi_n}{\partial \hat{x}} y_n & \sum_{n=1}^{8} \frac{\partial \varphi_n}{\partial \hat{x}} z_n \\ \sum_{n=1}^{8} \frac{\partial \varphi_n}{\partial \hat{y}} x_n & \sum_{n=1}^{8} \frac{\partial \varphi_n}{\partial \hat{y}} y_n & \sum_{n=1}^{8} \frac{\partial \varphi_n}{\partial \hat{y}} z_n \\ \sum_{n=1}^{8} \frac{\partial \varphi_n}{\partial \hat{z}} x_n & \sum_{n=1}^{8} \frac{\partial \varphi_n}{\partial \hat{z}} y_n & \sum_{n=1}^{8} \frac{\partial \varphi_n}{\partial \hat{z}} z_n \end{bmatrix}.$$

$(A10)$

We then invert (A8) to obtain the basis function derivatives in terms of (x, y, z):

$$\begin{Bmatrix} \frac{\partial \varphi_n}{\partial x} \\ \frac{\partial \varphi_n}{\partial y} \\ \frac{\partial \varphi_n}{\partial z} \end{Bmatrix} = [J^{-1}] \begin{Bmatrix} \frac{\partial \varphi_n}{\partial \hat{x}} \\ \frac{\partial \varphi_n}{\partial \hat{y}} \\ \frac{\partial \varphi_n}{\partial \hat{z}} \end{Bmatrix}.$$

$(A11)$

The LHS of (A11) contains the spatial derivatives needed to evaluate (A3)–(A6).

Eqs. (A3)–(A6) also contain the viscosity η, which is computed at each quadrature point. In the BP approximation, η is given by (2); it is a function of the flow factor A and the effective strain rate defined by (3). We approximate A by its value at the element center. The (squared) effective strain rate $\dot{\varepsilon}_e^2$ is evaluated at each quadrature point by summing over strain-rate
5 components. The x components are given by

$$\frac{\partial u}{\partial x} = \sum_{n=1}^{8} \frac{\partial \varphi_n}{\partial x} u_n, \quad \frac{\partial v}{\partial x} = \sum_{n=1}^{8} \frac{\partial \varphi_n}{\partial x} v_n, \tag{A12}$$

and similarly for the y and z components. The nodal velocities in (A12) are values from the previous iteration.

We now have the information needed to compute the integrands (A3)–(A6) at quadrature points. To integrate over a hexahedron, we take a weighted sum of the values at each of eight quadrature points. These points are located at reference coordinates
10 $(\hat{x}, \hat{y}, \hat{z}) = (\pm 1/\sqrt{3}, \pm 1/\sqrt{3}, \pm 1/\sqrt{3})$. To evaluate an integral of the form

$$\int_{\Omega} \eta \left(\frac{\partial \varphi_i}{\partial z} \frac{\partial \varphi_j}{\partial z} \right) d\Omega \tag{A13}$$

over element volume Ω, we compute the sum over quadrature points

$$\sum_{p=1}^{8} w_p \eta_p \left(\frac{\partial \varphi_i}{\partial z} \frac{\partial \varphi_j}{\partial z} \right)_p |J_p|, \tag{A14}$$

where $|J|$ is the determinant of the Jacobian (A10). For this choice of quadrature points, each point has a weight $w_p = 1$.

15 The terms of the element matrices $\mathbf{K_{uu}}, \mathbf{K_{uv}}, \mathbf{K_{vu}}$, and $\mathbf{K_{vv}}$ are then inserted into the corresponding global matrices $\mathbf{A_{uu}}, \mathbf{A_{uv}}, \mathbf{A_{vu}}$, and $\mathbf{A_{vv}}$. This is mostly a matter of bookkeeping. For example, the first row of $\mathbf{K_{uu}}$ corresponds to a particular node of element (k, i, j) (specifically, the node with indices $(k-1, i-1, j-1)$, given the convention for numbering nodes within elements). This row corresponds to a row of the global matrix $\mathbf{A_{uu}}$, and each of the eight terms in the row of $\mathbf{K_{uu}}$ is associated with a column of $\mathbf{A_{uu}}$. Glissade determines the correct column and adds the $\mathbf{K_{uu}}$ term to the corresponding
20 term in $\mathbf{A_{uu}}$. This process proceeds until the code has looped over all the active elements and filled the global matrices.

If written in full, each global matrix would have as many rows and columns as there are active nodes. These matrices, however, are sparse, with a maximum of 27 nonzero terms per row (corresponding to a node and its 26 nearest neighbors in a hexahedral lattice). Glissade therefore assembles and stores arrays of dimension $(27, nz, nx-1, ny-1)$, where nx, ny, and nz are the grid dimensions. The 27 terms of the first array dimension are arranged according to a geometric convention. For
25 example, suppose we are filling columns for the matrix row corresponding to node (k, i, j). Then, by convention, index 1 refers to the node with coordinates $(k-1, i-1, j-1)$, index 14 refers to the node itself (i.e., the diagonal term of the row), and index 27 refers to the node at $(k+1, i+1, j+1)$ (and similarly for the other indices). After assembly, these arrays can be converted to the form required by a particular linear solver.

The remaining assembly consists of evaluating the other terms in (9) (i.e., the basal and lateral boundary conditions and the
30 gravitational forcing) and implementing Dirichlet boundary conditions, if applicable. We consider these in turn.

A2 Basal boundary conditions

The basal boundary terms in (9) are

$$x : \int_{\Gamma_B} \beta u \varphi_1 d\Gamma,$$

$$y : \int_{\Gamma_B} \beta v \varphi_2 d\Gamma. \tag{A15}$$

The basal face of each cell is a rectangle. To integrate over a rectangle, we sum over four quadrature points lying at $(\hat{x}, \hat{y}) = (\pm 1/\sqrt{3}, \pm 1/\sqrt{3})$ in a reference square with center $(0, 0)$ and vertices $(\pm 1, \pm 1)$. This reference square is the 2D analog of the reference cube discussed above. We define four bilinear basis functions on the square (cf. (A7)):

$$\varphi_1 = (1 - \hat{x})(1 - \hat{y})/4,$$
$$\varphi_2 = (1 + \hat{x})(1 - \hat{y})/4,$$
$$\varphi_3 = (1 + \hat{x})(1 + \hat{y})/4,$$
$$\varphi_4 = (1 - \hat{x})(1 + \hat{y})/4. \tag{A16}$$

Given these basis functions and their spatial derivatives, we can compute the Jacobian for the transformation between the reference square and the rectangular cell face, using the 2D versions of (A10) and (A11):

$$[J] = \begin{bmatrix} \sum_{n=1}^{4} \frac{\partial \varphi_n}{\partial \hat{x}} x_n & \sum_{n=1}^{4} \frac{\partial \varphi_n}{\partial \hat{x}} y_n \\ \sum_{n=1}^{4} \frac{\partial \varphi_n}{\partial \hat{y}} x_n & \sum_{n=1}^{4} \frac{\partial \varphi_n}{\partial \hat{y}} y_n \end{bmatrix}, \tag{A17}$$

$$\left\{ \begin{array}{c} \frac{\partial \varphi_n}{\partial x} \\ \frac{\partial \varphi_n}{\partial y} \end{array} \right\} = [J^{-1}] \left\{ \begin{array}{c} \frac{\partial \varphi_n}{\partial \hat{x}} \\ \frac{\partial \varphi_n}{\partial \hat{y}} \end{array} \right\}. \tag{A18}$$

The integrand at a quadrature point has the form $\beta \varphi_i \varphi_j$ where the second φ term arises from the finite-element expansion of u at a quadrature point:

$$u = \sum_{n=1}^{4} u_n \varphi_n. \tag{A19}$$

We determine β at quadrature points from the values at cell vertices:

$$\beta = \sum_{n=1}^{4} \beta_n \varphi_n. \tag{A20}$$

The integral over a cell is then computed as a sum over quadrature points:

$$\sum_{p=1}^{4} w_p \beta_p (\varphi_i \varphi_j)_p |J_p|, \tag{A21}$$

where $w_p = 1$ for each point. This procedure yields a 4×4 matrix describing the connections between each vertex and its neighbors in the cell. Since the x term in (A15) contains u but not v, and the y term contains v but not u, we form 2D matrices $\mathbf{K_{uu}}$ and $\mathbf{K_{vv}}$, but not $\mathbf{K_{uv}}$ and $\mathbf{K_{vu}}$. Each term of $\mathbf{K_{uu}}$ is then inserted into the global matrix $\mathbf{A_{uu}}$, and similarly for $\mathbf{K_{vv}}$ into $\mathbf{A_{vv}}$.

This assembly method tends to smooth the β field. If it is desired to resolve sharp discontinuities in β, as in the stream test problem of Section 4.3, Glissade supports a local assembly method in which the basal friction at a particular vertex depends on the value of β at that vertex alone.

A3 Lateral boundary conditions

The lateral boundary terms in (9) are

$$x : \int_{\Gamma_L} p n_1 \varphi_1 \, d\Gamma,$$

$$y : \int_{\Gamma_L} p n_2 \varphi_2 \, d\Gamma. \tag{A22}$$

Since these terms are independent of u and v, they contribute to the load vectors $\boldsymbol{b_u}$ and $\boldsymbol{b_v}$ on the right-hand side of (16). They are integrated over the lateral faces of cells (either grounded or floating) that border the ocean.

The lateral faces bordering the ocean are quadrilaterals that can be mapped to a reference square. The integral over each face is found by summing over four quadrature points. Basis functions are given by (A16), and the Jacobian of the reference square is found using (A17). The ice thickness H at each quadrature point is evaluated using

$$H = \sum_{n=1}^{4} H_n \varphi_n, \tag{A23}$$

where the H_n are nodal values interpolated from cell centers.

The integrands have the form $p\varphi$, where p is the vertically averaged net pressure normal to the ice edge, given by (13). The integral of the pressure terms over a lateral face is computed as a sum over quadrature points:

$$\sum_{p=1}^{4} \pm w_p p_p (\varphi_i)_p |J_p|, \tag{A24}$$

where the sign depends on the orientation of the face. The resulting pressure terms are inserted into the load vector (either $\boldsymbol{b_u}$ or $\boldsymbol{b_v}$, depending on the orientation) in the rows associated with each of the four nodes of the face.

A4 Gravitational driving stress

The gravitational forcing terms in (9) are

$$x : \int_{\Omega} \rho g \frac{\partial s}{\partial x} \varphi_1 \, d\Omega,$$

$$y : \int_{\Omega} \rho g \frac{\partial s}{\partial y} \varphi_2 \, d\Omega. \tag{A25}$$

To compute these terms we evaluate $\partial s/\partial x$ and $\partial s/\partial y$ at each active vertex, typically using (14) or its y analog.

The integrals in (A25) are over 3D elements. Each hexahedral element is mapped to a reference cube as described above. Given $\partial s/\partial x$ at the vertices of a cell, the surface slope terms at quadrature points are

$$\frac{\partial s}{\partial x} = \sum_{n=1}^{8} \varphi_n \left(\frac{\partial s}{\partial x} \right)_n, \quad \frac{\partial s}{\partial y} = \sum_{n=1}^{8} \varphi_n \left(\frac{\partial s}{\partial y} \right)_n, \tag{A26}$$

where the basis functions φ are given by (A7) and the spatial derivatives are derived from (A10) and (A11). The integral of $\rho g \frac{\partial s}{\partial x} \varphi$ over an element is evaluated as a sum over quadrature points:

$$\sum_{p=1}^{8} w_p \rho g \left(\frac{\partial s}{\partial x} \right)_p (\varphi_i)_p |J_p|, \tag{A27}$$

and similarly for the $\partial s/\partial y$ term. Glissade inserts these terms into the load vectors $\boldsymbol{b_u}$ and $\boldsymbol{b_v}$.

As described above for the assembly of β terms, this method tends to smooth the surface elevation gradient field, averaging neighbor values of $\partial s/\partial x$ and $\partial s/\partial y$ into the driving stress at each vertex. For problems with sharp variations in surface gradients, we have found the solver to be more robust when the driving stress at each vertex depends on $\partial s/\partial x$ and $\partial s/\partial y$ at that vertex alone. Thus, Glissade also supports a local assembly method for the driving stress.

A5 Dirichlet boundary conditions

Once the matrix has been assembled, it may need to be adjusted for Dirichlet boundary conditions (i.e., prescribed velocity values at certain nodes). A common Dirichlet condition is to set $u = v = 0$ at the bed to enforce a no-slip boundary condition. (A no-slip condition can also be enforced by setting the basal traction coefficient β to a large value, but formally this is not a Dirichlet condition.) Also, it may be desirable to set u and v to observed values at certain locations, as in the Ross Ice Shelf test case (Section 4.4).

Suppose that at node (k, i, j) we have $u = u_c$ and $v = v_c$, where u_c and v_c are prescribed values. Let nr be the row of $\mathbf{A_{uu}}$ associated with this node, and let nc range over the columns with nonzero entries in this row. To enforce the Dirichlet condition, we set $\mathbf{A_{uu}}(nr, nc) = \mathbf{A_{vv}}(nr, nc) = 0$ for all values of nc except $nc = nr$ (the diagonal term); we set $\mathbf{A_{uu}}(nr, nr) = \mathbf{A_{vv}}(nr, nr) = 1$. In addition, we set $\mathbf{A_{uv}}(nr, nc) = \mathbf{A_{vu}}(nr, nc) = 0$ for all nc, since these two matrices do not contain any terms on the diagonal of the full global matrix (i.e., $\mathbf{A}$ in (15)). On the right-hand side, we set $\mathbf{b_u}(nr) = u_c$

and $\mathbf{b_v}(nr) = v_c$. These operations convert the matrix rows associated with node (k, i, j) to the equations $1 \cdot u = u_c, 1 \cdot v = v_c$, which clearly have the desired solutions u_c and v_c.

A further step is needed to maintain matrix symmetry, as required for the PCG solver. Consider the term $\mathbf{A_{uu}}(nr, nc)$, where nc is a column associated with a neighboring node. We have already set $\mathbf{A_{uu}}(nr, nc) = 0$, so we need to set $\mathbf{A_{uu}}(nc, nr) = 0$ to maintain symmetry. The Dirichlet condition is $u(nr) = u_c$. Thus we can replace $\boldsymbol{b_u}(nc)$ with $\boldsymbol{b_u}(nc) - \boldsymbol{A_{uu}}(nc, nr)u_c$ and set $\mathbf{A_{uu}}(nc, nr) = 0$ without altering the problem. We do this for all the terms in the columns associated with node (k, i, j) (i.e., all the terms multiplied by u_c or v_c in the matrix-vector product). Thus, both the rows and the columns associated with node (k, i, j) are filled with zeros, except for the diagonal term, and the full global matrix remains symmetric.

Author contributions. WHL, SFP and MJH were the primary developers of CISM; WHL led development of the Glissade dycore. KJE, GRL, MP, DMR, WJS, AGS and PHW contributed to various parts of the code, including parallel infrastructure, performance metrics, links to Trilinos, and links to CESM. JHK, KJE and ARB integrated CISM verification and validation tests into the LIVVkit package. ARB, SLB, JGF, JHK, GRL, and LV helped set up and analyze Greenland experiments. WHL wrote the paper with contributions from SFP, MJH, and GRL, and with input from all authors.

Acknowledgements. CISM development was supported primarily by the Earth System Modeling program, Office of Biological and Environmental Research (BER) of the U.S. Department of Energy's Office of Science. Additional support was provided by the DOE's Office of Advanced Scientific Computing Research (ASCR), by BER's Regional and Global Climate Modeling Program, and by the National Science Foundation. This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977. Computing resources (doi:10.5065/D6RX99HX) were provided by the Climate Simulation Laboratory at NCAR's Computational and Information Systems Laboratory, sponsored by the National Science Foundation and other agencies. LV was supported by NSF grant ANT-0424589 to the Center for Remote Sensing of Ice Sheets (CReSIS). This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

We thank Jan Lenaerts and Brice Noël for providing surface forcing data sets from RACMO2. We are also grateful to Rob Arthern, Xylar Asay-Davis, Andy Aschwanden, Jeremy Bassis, Jed Brown, Ed Bueler, Steph Cornford, John Dukowicz, Heiko Goelzer, Dan Goldberg, Jesse Johnson, Eric Larour, Marcus Löfverström, Dan Martin, Mathieu Morlighem, Sophie Nowicki, Frank Pattyn, Tony Payne, David Pollard, Hélène Seroussi, Slawek Tulaczyk, Miren Vizcaíno and Morgan Whitcomb for helpful discussions. We thank Johannes Sutter and one anonymous referee for constructive comments.

References

Albrecht, T., Martin, M., Haseloff, M., Winkelmann, R., and Levermann, A.: Parameterization for subgrid-scale motion of ice-shelf calving fronts, The Cryosphere, 5, 35–44, doi:10.5194/tc-5-35-2011, 2011.

Arthern, R. J. and Williams, C. R.: The sensitivity of West Antarctica to the submarine melting feedback, Geophys. Res. Lett., 44, 2352–2359, doi:10.1002/2017GL072514, 2017.

Arthern, R. J., Hindmarsh, R. C. A., and Williams, C. R.: Flow speed within the Antarctic ice sheet and its controls inferred from satellite observations, J. Geophys. Res. Earth Surf., 120, doi:10.1002/2014JF003239, 2015.

Asay-Davis, X. S., Cornford, S. L., Durand, G., Galton-Fenzi, B. K., Gladstone, R. M., Gudmundsson, G. H., Hattermann, T., Holland, D. M., Holland, D., Holland, P. R., Martin, D. F., Mathiot, P., Pattyn, F., and Seroussi, H.: Experimental design for three interrelated marine ice sheet and ocean model intercomparison projects: MISMIP v. 3 (MISMIP+), ISOMIP v. 2 (ISOMIP+) and MISOMIP v. 1 (MISOMIP1), Geosci. Model Dev., 9, 2471–2497, doi:10.5194/gmdd-9-2471-2016, 2016.

Asay-Davis, X. S., Jourdain, N. C., and Nakayama, Y.: Developments in simulating and parameterizing interactions between the Southern Ocean and the Antarctic Ice Sheet, Curr. Clim. Change Rep., 3, 316–329, doi:10.1007/s40641-017-0071-0, 2017.

Aschwanden, A., Adalgeirsdóttir, G., and Khroulev, C.: Hindcasting to measure ice sheet model sensitivity to initial states, The Cryosphere, 7, 1083–1093, doi:10.5194/tc-7-1083-2013, 2013.

Aschwanden, A., Fahnestock, M. A., and Truffer, M.: Complex Greenland outlet glacier flow captured, Nat. Commun., 7:10524, doi:10.1038/ncomms10524, 2016.

Bassis, J. N. and Ma, Y.: Evolution of basal crevasses links ice shelf stability to ocean forcing, Earth Planet. Sci. Lett., 409, 203–211, doi:10.1016/j.epsl.2014.11.003, 2015.

Bassis, J. N. and Walker, C. C.: Upper and lower limits on the stability of calving glaciers from the yield strength envelope of ice, Proc. Roy. Soc. A, 468, 913–931, doi:10.1098/rspa.2011.0422, 2012.

Blatter, H.: Velocity and stress fields in grounded glaciers - a simple algorithm for including deviatoric stress gradients, J. Glaciol., 41, 333–344, 1995.

Bueler, E.: Lectures at Karthaus: Numerical modelling of ice sheets and ice shelves, https://www.projects.science.uu.nl/iceclimate/karthaus/archive/lecturenotes/2009/bueler/EdBueler.pdf, last accessed 27 May 2018, 2009.

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

Bueler, E. and van Pelt, W.: Mass-conserving subglacial hydrology in the Parallel Ice Sheet Model version 0.6, Geosci. Model Dev., 8, 1613–1635, doi:10.5194/gmd-8-1613-2015, 2015.

Bueler, E., Lingle, C. S., Kallen-Brown, J. A., Covey, D. N., and Bowman, L. N.: Exact solutions and verification of numerical models for isothermal ice sheets, J. Glaciol., 51, 291–306, doi:10.3189/172756505781829449, 2005.

Cai, C., Rignot, E., Menemenlis, D., and Nakayama, Y.: Observations and modeling of ocean-induced melt beneath Petermann Glacier Ice Shelf in northwestern Greenland, Geophys. Res. Lett., 44, 8396–8403, doi:10.1002/2017GL073711, 2017.

Choi, Y., Morlighem, M., Rignot, E., Mouginot, J., and Wood, M.: Modeling the response of Nioghalvjerdsjorden and Zachariae Isstrom Glaciers, Greenland, to ocean forcing over the next century, Geophys. Res. Lett., 44, 11,071–11,079, doi:10.1002/2017GL075174, 2017.

Chronopoulos, A. T.: A class of parallel iterative methods implemented on multiprocessors, Ph.D. thesis, Department of Computer Science, University of Illinois, 1986.

Chronopoulos, A. T. and Gear, C. W.: *s*-step iterative methods for symmetric linear systems, J. Comput. Appl. Math., 25, 153–168, 1989.

Church, J., Clark, P., Cazenave, A., Gregory, J., Jevrejeva, S., Levermann, A., Merrifield, M., Milne, G., Nerem, R., Nunn, P., Payne, A., Pfeffer, W., Stammer, D., and Unnikrishnan, A.: Sea Level Change, in: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, edited by Stocker, T., Qin, D., Plattner, G.-K., Tignor, M., Allen, S., Boschung, J., Nauels, A., Xia, Y., Bex, V., and Midgley, P., p. 1137–1216, Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, doi:10.1017/CBO9781107415324.026, 2013.

Cornford, S. L., Martin, D. F., Graves, D. T., Ranken, D. R., Le Brocq, A. M., Gladstone, R. M., Payne, A. J., Ng, E. G., and Lipscomb, W. H.: Adaptive mesh, finite volume modeling of marine ice sheets, J. Comput. Phys., 232, 529–549, 2013.

Cuffey, K. and Paterson, W. S. B.: The Physics of Glaciers, Butterworth-Heinneman, Amsterdam, 4th edn., 2010.

Dukowicz, J. K. and Baumgardner, J. R.: Incremental remapping as a transport/advection algorithm, J. Comput. Phys., 160, 318–335, 2000.

Dukowicz, J. K., Price, S. F., and Lipscomb, W. H.: Consistent approximations and boundary conditions for ice-sheet dynamics from a principle of least action, J. Glaciol., 56, 480–496, 2010.

Ettema, J., van den Broeke, M. R., van Meijgaard, E., van de Berg, W. J., Bamber, J. L., Box, J. E., and Bales, R. C.: Higher surface mass balance of the Greenland ice sheet revealed by high-resolution climate modeling, Geophys. Res. Lett., 36, L12501, doi:10.1029/2009GL038110, 2009.

Evans, K. J., Salinger, A. G., Worley, P. H., Price, S. F., Lipscomb, W. H., Nichols, J. A., III, J. B. W., Perego, M., Vertenstein, M., Edwards, J., and Lemieux, J.-F.: A modern solver interface to manage solution algorithms in the Community Earth System Model, Int. J. High Perform. Comput. Appl., 26, 54–62, doi:10.1177/1094342011435159, 2012.

Fyke, J. G., Sacks, W. J., and Lipscomb, W. H.: A technique for generating consistent ice sheet initial conditions for coupled ice sheet/climate models, Geosci. Model Dev., 7, 1183–1195, doi:10.5194/gmd-7-1183-2014, 2014.

Gagliardini, O., Zwinger, T., Gillet-Chaulet, F., Durand, G., Favier, L., De Fleurian, B., Greve, R., Malinen, M., Martin, C., Råback, P., Ruokolainen, J., Sacchettini, M., Schäfer, M., Seddik, H., and Thies, J.: Capabilities and performance of Elmer/Ice, a new-generation ice sheet model, Geosci. Model Dev., 6, 1299–1318, 2013.

Gladstone, R. M., Payne, A. J., and Cornford, S. L.: Parameterising the grounding line in flow-line ice sheet models, The Cryosphere, 4, 605–619, 2010.

Glen, J. W.: The creep of polycrystalline ice, Proc. R. Soc. Lond. A, 228, 519–538, 1955.

Goelzer, H., Nowicki, S., Edwards, T., Beckley, M., Abe-Ouchi, A., Aschwanden, A., Calov, R., Gagliardini, O., Gillet-Chaulet, F., Golledge, N. R., Gregory, J., Greve, R., Humbert, A., Huybrechts, P., Kennedy, J. H., Larour, E., Lipscomb, W. H., Le clec'h, S., Lee, V., Morlighem, M., Pattyn, F., Payne, A. J., Rodehacke, C., Rückamp, M., Saito, F., Schlegel, N., Seroussi, H., Shepherd, A., Sun, S., van de Wal, R., and Ziemen, F. A.: Design and results of the ice sheet model initialisation experiments initMIP-Greenland: an ISMIP6 intercomparison, The Cryosphere, 12, 1433–1460, doi:10.5194/tc-12-1433-2018, 2018.

Goldberg, D. N.: A variationally derived, depth-integrated approximation to a higher-order glaciological flow model, J. Glaciol., 57, 157–170, doi:10.3189/002214311795306763, 2011.

Halfar, P.: On the dynamics of the ice sheets 2, J. Geophys. Res., 88, 6043–6051, 1983.

Hanna, E., Navarro, F. J., Pattyn, F., Domingues, C. M., Fettweis, X., Ivins, E. R., Nicholls, R. J., Ritz, C., Smith, B., Tulaczyk, S., Whitehouse, P. L., and Zwally, H. J.: Ice-sheet mass balance and climate change, Nature, 498, 51–59, doi:10.1038/nature12238, 2013.

Heroux, M. A., Bartlett, R. A., Howle, V. E., Hoekstra, R. J., Hu, J. J., Kolda, T. G., Lehoucq, R. B., Long, K. R., Pawlowski, R. P., Phipps, E. T., Salinger, A. G., Thornquist, H. K., Tuminaro, R. S., Willenbring, J. M., Williams, A., and Stanley, K. S.: An overview of the Trilinos project, ACM Trans. Math. Softw., 31, 397–423, doi:http://doi.acm.org/10.1145/1089014.1089021, 2005.

Hindmarsh, R.: The role of membrane-like stresses in determining the stability and sensitivity of the Antarctic ice sheets: back pressure and grounding line motion, Phil. Trans. R. Soc. A, 364, 1733–1767, doi:10.1098/rsta.2006.1797, 2006.

Hoffman, M. J. and Price, S.: Feedbacks between coupled subglacial hydrology and glacier dynamics, J. Geophys. Res. Earth Surf., 119, doi:10.1002/2013JF002943, 2014.

Hoffman, M. J., Perego, M., Price, S. F., Lipscomb, W. H., Jacobsen, D., Tezaur, I., Salinger, A. G., Tuminaro, R., and Zhang, T.: MPAS-Albany Land Ice (MALI): A variable resolution ice sheet model for Earth system modeling using Voronoi grids, Geosci. Model Dev. Discuss., pp. 1–47, doi:10.5194/gmd-2018-78, in review, 2018.

Huebner, K. H., Dewhirst, D. L., Smith, D. E., and Byrom, T. G.: The Finite Element Method for Engineers, Wiley, New York, 4th edn., 2001.

Hughes, T.: The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Dover Civil and Mechanical Engineering, Dover, Mineola, New York, 1st edn., 2000.

Hurrell, J., Holland, M., Gent, P., Ghan, S., Kay, J., Kushner, P., Lamarque, J.-F., Large, W., Lawrence, D., Lindsay, K., Lipscomb, W., Long, M., Mahowald, N., Marsh, D., Neale, R., Rasch, P., Vavrus, S., Vertenstein, M., Bader, D., Collins, W., Hack, J., Kiehl, J., and Marshall, S.: The Community Earth System Model: A framework for collaborative research, B. Am. Meteorol. Soc., 94, 1339–1360, doi:10.1175/BAMS-D-12-00121.1, 2013.

Hutter, K.: Theoretical Glaciology, Mathematical Approaches to Geophysics, D. Reidel Publishing Company, Dordrecht, Boston, Lancaster, 1983.

Joughin, I., Smith, B., Howat, I., and Scambos, T.: MEaSUREs Greenland Ice Sheet Velocity Map from InSAR Data, National Snow and Ice Data Center, Boulder, Colorado, 2010.

Kennedy, J. H., Bennett, A. R., Evans, K. J., Price, S., Hoffman, M., Lipscomb, W. H., Fyke, J., Vargo, L., Boghozian, A., Norman, M., and Worley, P. H.: LIVVkit: An extensible, python-based, land ice verification and validation tool kit for ice sheet models, J. Adv. Model. Earth Syst., 9, 854–869, doi:10.1002/2017MS000916, 2017.

Larour, E., Seroussi, H., Morlighem, M., and Rignot, E.: Continental scale, high order, high spatial resolution, ice sheet modeling using the Ice Sheet System Model (ISSM), J. Geophys. Res., 117, F01022, doi:10.1029/2011JF002140, 2012.

Leguy, G. R., Asay-Davis, X. S., and Lipscomb, W. H.: Parameterization of basal friction near grounding lines in a one-dimensional ice sheet model, The Cryosphere, 8, 1239–1259, doi:10.5194/tc-8-1239-2014, 2014.

Levermann, A., Albrecht, T., Winkelmann, R.and Martin, M. A., Haseloff, M., and Joughin, I.: Kinematic first-order calving law implies potential for abrupt ice-shelf retreat, The Cryosphere, 6, 273–286, doi:10.5194/tc-6-273-2012, 2012.

Lipscomb, W. H. and Hunke, E. C.: Modeling sea ice transport using incremental remapping, Mon. Wea. Rev., 132, 1341–1354, 2004.

Lipscomb, W. H., Fyke, J. G., Vizcaino, M., Sacks, W. J., Wolfe, J., Vertenstein, M., Craig, A., Kluzek, E., and Lawrence, D. M.: Implementation and initial evaluation of the Glimmer Community Ice Sheet Model in the Community Earth System Model, J. Climate, 26, 7352–7371, doi:10.1175/JCLI-D-12-00557.1, 2013.

MacAyeal, D. R.: Large-scale ice flow over a viscous basal sediment - Theory and application to Ice Stream B, Antarctica, J. Geophys. Res., 94, 4071–4087, 1989.

MacAyeal, D. R., Rommelaere, V., Huybrechts, P., Hulbe, C. L., Determann, J., and Ritz, C.: An ice-shelf model test based on the Ross Ice Shelf, Antarctica, Ann. Glaciol., 23, 46–51, 1996.

MacGregor, J. A., Fahnestock, M. A., Catania, G. A., Aschwanden, A., Clow, G. D., Colgan, W. T., Gogineni, S. P., Morlighem, M., Nowicki, S. M. J., Paden, J. D., Price, S. F., and Seroussi, H.: A synthesis of the basal thermal state of the Greenland Ice Sheet, J. Geophys. Res. Earth Surf., 121, doi:10.1002/ 2015JF003803, 2015.

Morlighem, M., Rignot, E., Seroussi, H., Larour, E., Dhia, H. B., and Aubry, D.: A mass conservation approach for mapping glacier ice thickness, Geophys. Res. Lett., 38, L19503, doi:10.1029/2011GL048659, 2011.

Morlighem, M., Rignot, E., Mouginot, J., Seroussi, H., and Larour, E.: Deeply incised submarine glacial valleys beneath the Greenland Ice Sheet, Nat. Geosci., 7, 418–422, doi:10.1038/ngeo2167, 2014.

Morlighem, M., Bondzio, J., Seroussi, H., Rignot, E., Larour, E., Humbert, A., and Rebuffi, S.: Modeling of Store Gletscher's calving dynamics, West Greenland, in response to ocean thermal forcing, Geophys. Res. Lett., 43, doi:10.1002/2016GL067695, 2016.

Noël, B., van de Berg, W. J., Machguth, H., Lhermitte, S., Howat, I., Fettweis, X., and van den Broeke, M. R.: A daily, 1 km resolution data set of downscaled Greenland ice sheet surface mass balance (1958–2015), The Cryosphere, 10, 2361–2377, doi:10.5194/tc-10-2361-2016, 2016.

Noël, B., van de Berg, W. J., van Wessem, J. M., van Meijgaard, E., van As, D., Lenaerts, J. T. M., Lhermitte, S., Kuipers Munneke, P., Smeets, C. J. P. P., van Ulft, L. H., van de Wal, R. S. W., and van den Broeke, M. R.: Modelling the climate and surface mass balance of polar ice sheets using RACMO2 – Part 1: Greenland (1958–2016), The Cryosphere, 12, 811–831, doi:10.5194/tc-12-811-2018, 2018.

Paterson, W. and Budd, W. F.: Flow parameters for ice sheet modeling, Cold Reg. Sci. Technol., 6, 175–177, 1982.

Pattyn, F.: 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, 2382, doi:10.1029/2002JB002329, 2003.

Pattyn, F., Perichon, L., Aschwanden, A., Breuer, B., de Smedt, B., Gagliardini, O., Gudmundsson, G. H., Hindmarsh, R. C. A., Hubbard, A., Johnson, J. V., Kleiner, T., Konovalov, Y., Martin, C., Payne, A. J., Pollard, D., Price, S., Rückamp, M., Saito, F., Souček, O., Sugiyama, S., and Zwinger, T.: Benchmark experiments for higher-order and full-Stokes ice sheet models (ISMIP—HOM), The Cryosphere, 2, 95–108, doi:10.5194/tc-2-95-2008, 2008.

Pattyn, F., Schoof, C., Perichon, L., Hindmarsh, R. C. A., Bueler, E., de Fleurian, B., Durand, G., Gagliardini, O., Gladstone, R., Goldberg, D., Gudmundsson, G. H., Huybrechts, P., Lee, V., Nick, F. M., Payne, A. J., Pollard, D., Rybak, O., Saito, F., and Vieli, A.: Results of the Marine Ice Sheet Model Intercomparison Project, MISMIP, The Cryosphere, 6, 573–588, doi:10.5194/tc-6-573-2012, 2012.

Payne, A. J. and Dongelmans, P. W.: Self–organisation in the thermomechanical flow of ice sheets, J. Geophys. Res., 102, 12 219–12 233, 1997.

Perego, M., Gunzburger, M., and Burkardt, J.: Parallel finite-element implementation for higher-order ice sheet models, J. Glaciol., 58, 76–88, doi:10.3189/2012JoG11J063, 2012.

Perego, M., Price, S., and Stadler, G.: Optimal initial conditions for coupling ice sheet models to Earth system models, J. Geophys. Res., 119, 1894–1917, doi:10.1002/2014jf003181, 2014.

Pimentel, S., Flowers, G. E., and Schoof, C. G.: A hydrologically coupled higher-order flow-band model of ice dynamics with a Coulomb friction sliding law, J. Geophys. Res., 115, 1–16, doi:10.1029/2009JF001621, 2010.

Pollard, D. and DeConto, R. M.: Description of a hybrid ice sheet-shelf model, and application to Antarctica, Geoscientific Model Development, 5, 1273–1295, doi:10.5194/gmd-5-1273-2012, 2012.

Pollard, D., DeConto, R. M., and Alley, R. B.: Potential Antarctic Ice Sheet retreat driving by hydrofracturing and ice cliff failure, Earth Plan. Sci. Lett., 412, 112–121, doi:10.1016/j.epsl.2014.12.035, 2015.

Price, S., Lipscomb, W., Hoffman, M., Hagdorn, M., Rutt, I., Payne, T., Hebeler, F., and Kennedy, J. H.: CISM 2.0.5 Documentation, Tech. rep., Los Alamos National Laboratory, https://cism.github.io/data/cism_documentation_v2_0.pdf, last accessed 03 December 2018, 2015.

Raymond, C. F.: Energy balance of ice streams, J. Glaciol., 46, 665–674, 2000.

Rommelaere, V.: Ice Shelf Models Intercomparison: Setup of the experiments, http://homepages.vub.ac.be/~phuybrec/eismint/shelf-descr.pdf, last accessed 13 May 2016, 1996.

Rutt, I., Hagdorn, M., Hulton, N., and Payne, A.: The Glimmer community ice sheet model, J. Geophys. Res., 114, F02 004, 2009.

Schoof, C.: The effect of cavitation on glacier sliding, Proc. R. Soc. A, 461, 609–627, doi:10.1098/rspa.2004.1350, 2005.

Schoof, C.: A variational approach to ice stream flow, J. Fluid Mech., 556, 227–251, 2006.

Schoof, C. and Hindmarsh, R. C. A.: Thin-film flows with wall slip: an asymptotic analysis of higher order glacier flow models, Q. J. Mech. Appl. Math., 63, 73–114, 2010.

Sergienko, O. V., Creyts, T. T., and Hindmarsh, R. C. A.: Similarity of organized patterns in driving and basal stresses of Antarctic and Greenland ice sheets beneath extensive areas of basal sliding, Geophys. Res. Lett., 41, doi:10.1002/2014GL059976, 2014.

Shapiro, N. and Ritzwoller, M.: Inferring surface heat flux distributions guided by a global seismic model: particular application to Antarctica, Earth Planet. Sci. Lett., 223, 213–224, doi:10.1016/j.epsl.2004.04.011, 2004.

Shepherd, A., Ivins, E., A, G., Barletta, V., Bentley, M., Bettadpur, S., Briggs, K., Bromwich, D., Forsberg, R., Galin, N., Horwath, M., Jacobs, S., Joughin, I., King, M., Lenaerts, J., Li, J., Ligtenberg, S., Luckman, A., Luthcke, S., McMillan, M., Meister, R., Milne, G., Mouginot, J., Muir, A., Nicolas, J., Paden, J., Payne, A., Pritchard, H., Rignot, E., Rott, H., Sørensen, L., Scambos, T., Scheuchl, B., Schrama, E., Smith, B., Sundal, A., van Angelen, J., van de Berg, W., van den Broeke, M., Vaughan, D., Velicogna, I., Wahr, J., Whitehouse, P., Wingham, D., Yi, D., Young, D., and Zwally, H.: A reconciled estimate of ice-sheet mass balance, Science, 338, 1183–1189, doi:10.1126/science.1228102, 2012.

Shepherd, A., Ivins, E., Rignot, E., Smith, B., van den Broeke, M., Velicogna, I., Whitehouse, P., Briggs, K., Joughin, I., Krinner, G., Nowicki, S., Payne, T., Scambos, T., Schlegel, N., Geruo, A., Agosta, C., Ahlstrøm, A., Babonis, G., Barletta, V., Blazquez, A., Bonin, J., Csatho, B., Cullather, R., Felikson, D., Fettweis, X., Forsberg, R., Gallee, H., Gardner, A., Gilbert, L., Groh, A., Gunter, B., Hanna, E., Harig, C., Helm, V., Horvath, A., Horwath, M., Khan, S., Kjeldsen, K., Konrad, H., Langen, P., Lecavalier, B., Loomis, B., Luthcke, S., McMillan, M., Melini, D., Mernild, S., Mohajerani, Y., Moore, P., Mouginot, J., Moyano, G., Muir, A., Nagler, T., Nield, G., Nilsson, J., Noel, B., Otosaka, I., Pattle, M., Peltier, W., Nadege, P., Rietbroek, R., Rott, H., Sandberg-Sørensen, L., Sasgen, I., Save, H., Schrama, E., Schröder, L., Seo, K.-W., Simonsen, S., Slater, T., Spada, G., Sutterley, T., Talpe, M., Tarasov, L., van de Berg, W., van der Wal, W., van Wessem, M., Vishwakarma, B., Wiese, D., and Wouters, B.: Mass balance of the Antarctic ice sheet from 1992 to 2017, Nature, 558, 219–222, doi:10.1038/s41586-018-0179-y, 2017.

Shewchuk, J. R.: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, Tech. rep., Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

Tezaur, I. K., Perego, M., Salinger, A. G., Tuminaro, R. S., and Price, S. F.: Albany/FELIX: a parallel, scalable and robust, finite element, first-order Stokes approximation ice sheet solver built for advanced analysis, Geosci. Model Dev., 8, 1197–1220, doi:10.5194/gmd-8-1197-2015, 2015.

Van den Berg, J., Van de Wal, R., and Oerlemans, J.: Effects of spatial discretization in ice-sheet modelling using the shallow-ice approximation, J. Glaciol., 52, 89—-98, doi:10.3189/172756506781828935, 2006.

Vizcaino, M.: Ice sheets as interactive components of Earth System Models: progress and challenges, WIREs Clim. Change, 5, 557–568, doi:10.1002/wcc.285, 2014.

Weertman, J.: On the sliding of glaciers, J. Glaciol., 3, 33–38, 1957.

Winkelmann, R., Martin, M. A., Haseloff, M., Albrecht, T., Bueler, E., Khroulev, C., and Levermann, A.: The Potsdam Parallel Ice Sheet Model (PISM-PIK) Part 1: Model description, The Cryosphere, 5, 715–726, 2011.
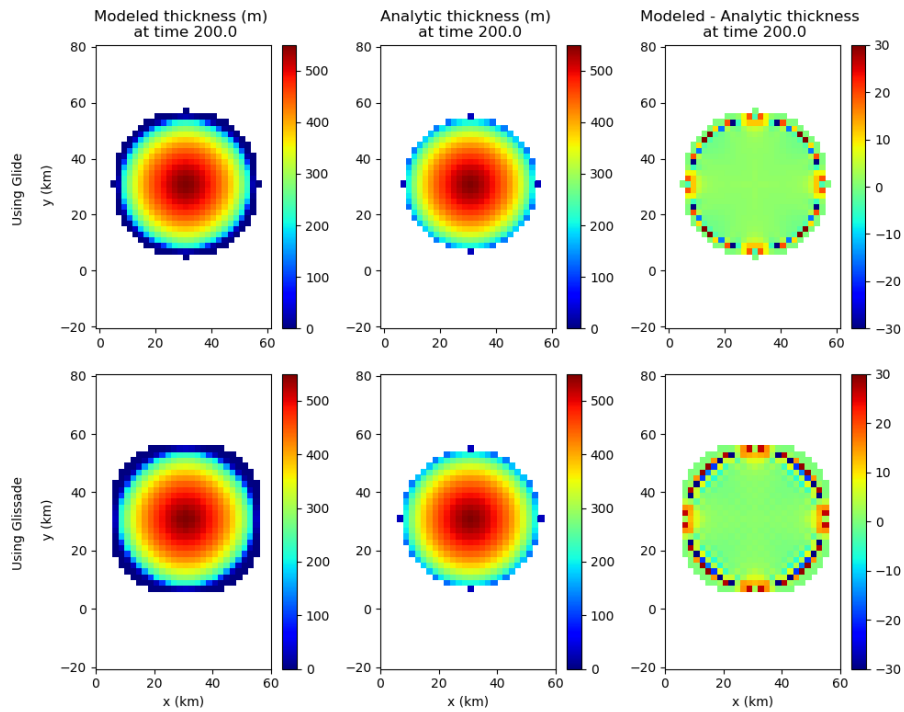
Figure 1. Results from the Halfar dome test using SIA solvers from Glide (top panels) and Glissade (bottom panels). (Left) Modeled ice thickness. (Center) Analytic ice thickness. (Right) Difference between modeled and analytic thickness.
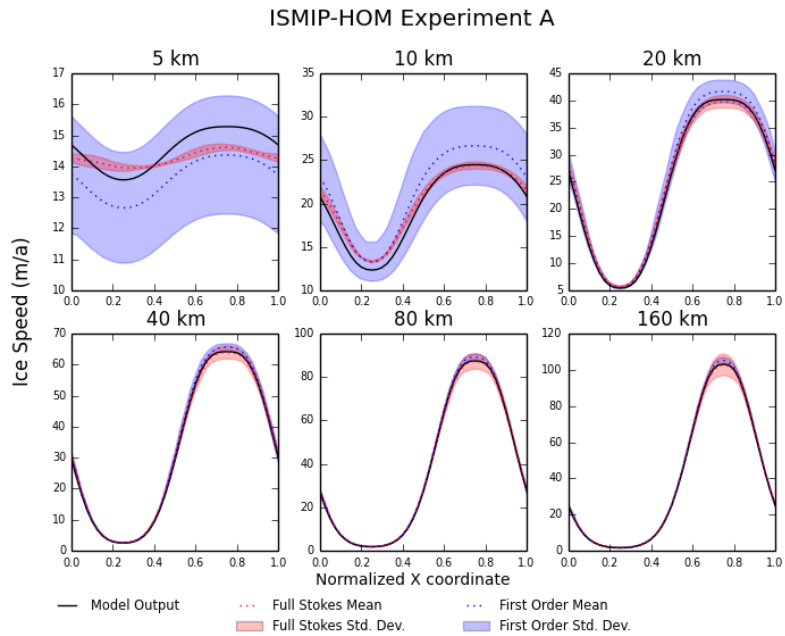
Figure 2. Results for ISMIP-HOM Experiment A (ice flow over a bumpy bed). Each plot shows the surface ice speed across the bump at $y = L/4$ for a different length scale L. The solid black line shows output from Glissade's BP velocity solver; the dotted red and blue lines show output from full-Stokes and first-order (i.e., higher-order) models, respectively, in Pattyn et al. (2008); and the red and blue shaded regions show the corresponding standard deviations.
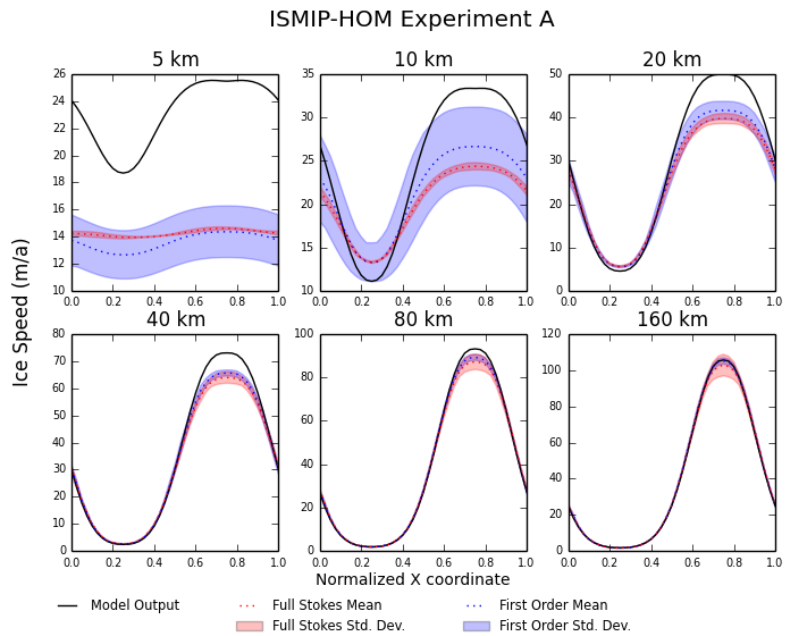
Figure 3. Same as Figure 2, except that the solid black line shows output from Glissade's DIVA velocity solver. Vertical scales are adjusted relative to Figure 2 to show the full range of the model solution.
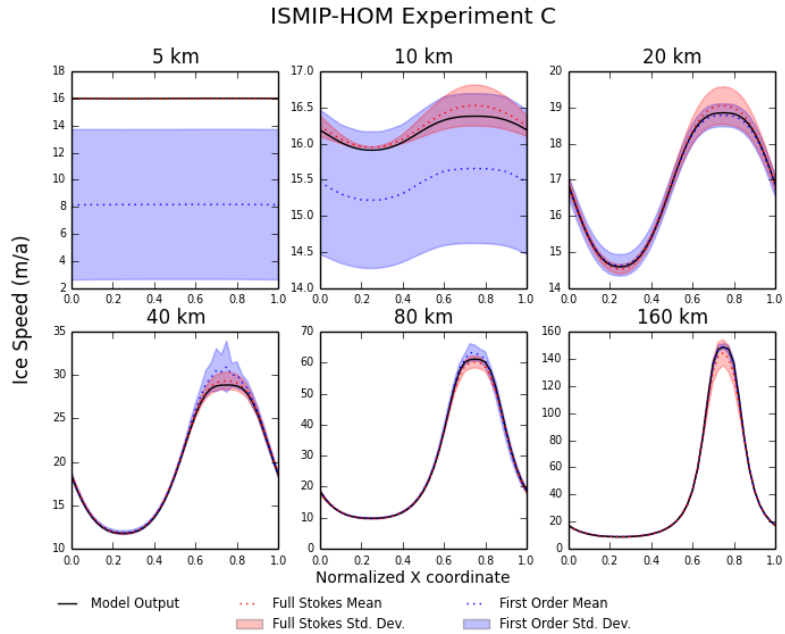
Figure 4. Results for ISMIP-HOM Experiment C (ice flow over a bed with variable basal friction). Each plot shows the surface ice speed across the bump at $y = L/4$ for a different length scale L. The solid black line shows output from Glissade's BP velocity solver; the dotted red and blue lines show output from Stokes and first-order models, respectively, in Pattyn et al. (2008); and the red and blue shaded regions show the corresponding standard deviations.
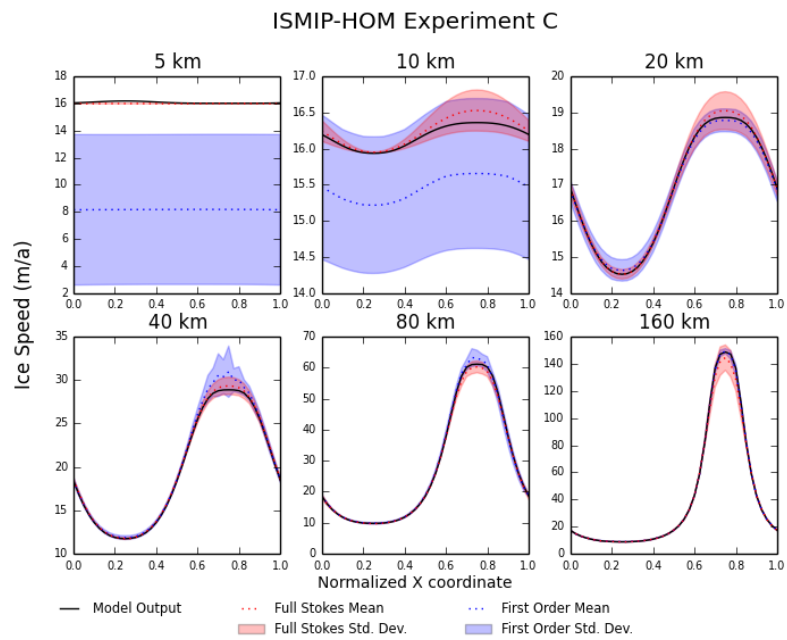
Figure 5. Same as Figure 4, except that the solid black line shows output from Glissade's DIVA velocity solver.
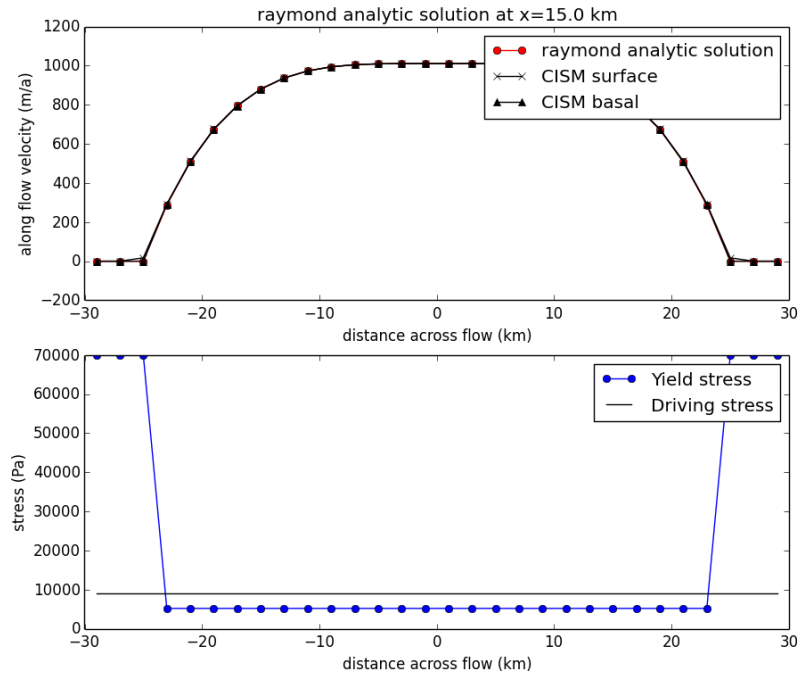
Figure 6. Results from the Raymond stream test using Glissade's BP velocity solver. (Top) Across-flow surface and basal velocity (m/yr) at $x = 15$ km compared to the analytic solution. At most points the analytic and simulated solutions are indistinguishable. (Bottom) Prescribed yield stress and gravitational driving stress (Pa).
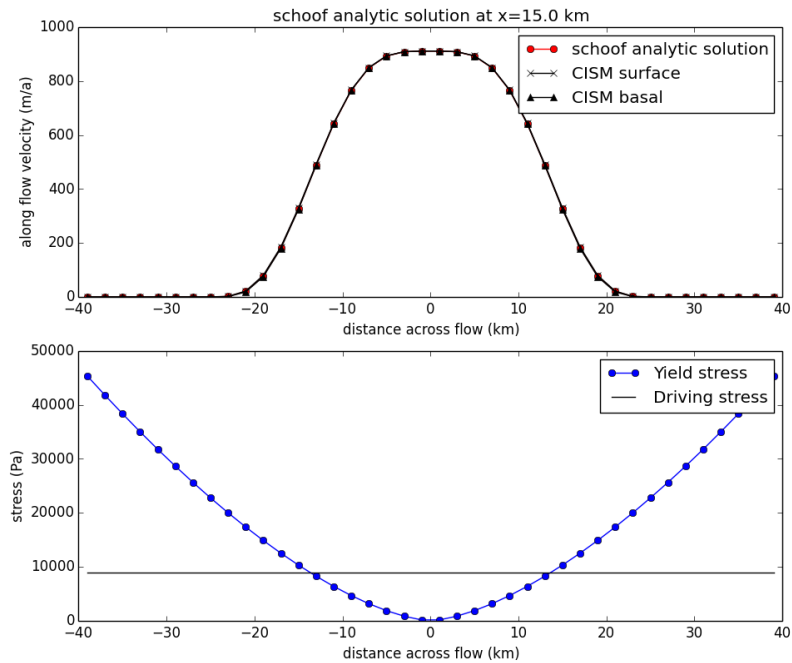
Figure 7. Same as Figure 6, but for the Schoof stream test with a smooth transition in yield stress. The analytic and simulated solutions are virtually indistinguishable.
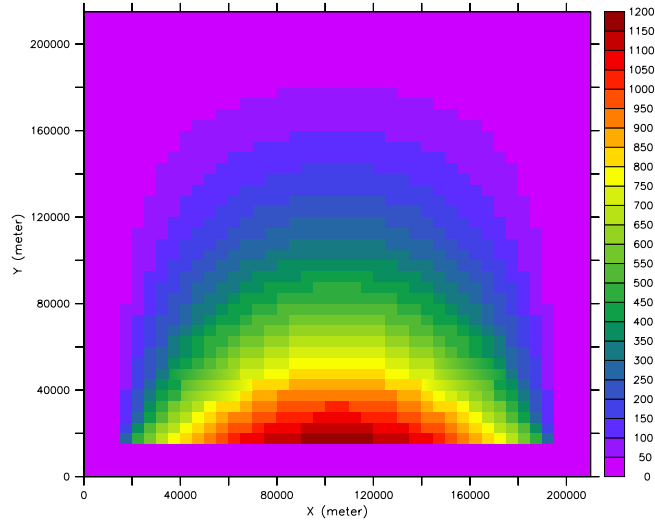
Figure 8. Ice speed (m/yr) for the confined-shelf test (based on tests 3 and 4 from the Ice Shelf Model Intercomparison exercise of Romme-laere (1996)) using Glissade's SSA velocity solver. The ice shelf is 500 m thick. The rectangular domain is open to the south and confined on the other three sides.
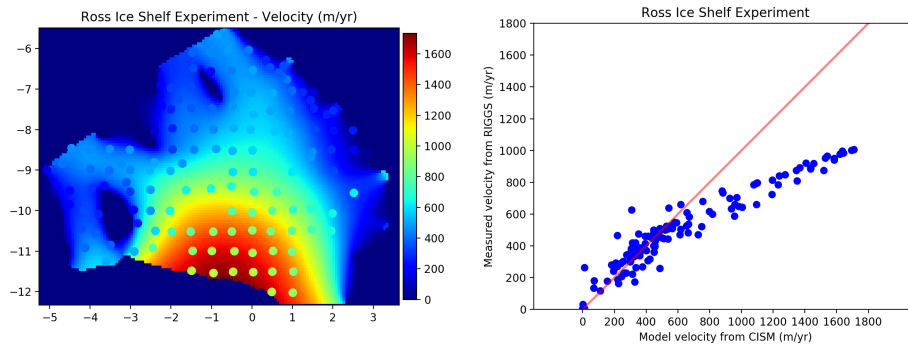
Figure 9. Results of the Ross Ice Shelf test described by MacAyeal et al. (1996), computed using Glissade's SSA solver. The left panel, based on Fig. 1 of MacAyeal et al. (1996), shows the vertically uniform ice speed (m/yr). The circles indicate locations where ice speed was measured by the Ross Ice Shelf Geophysical and Glaciological Survey (RIGGS); measured speeds are shaded with the same color scale as the observations. Axis coordinates match the coordinates in MacAyeal et al. (1996). The right panel, based on Fig. 2 of MacAyeal et al. (1996), compares simulated to measured ice speeds at the RIGGS locations.
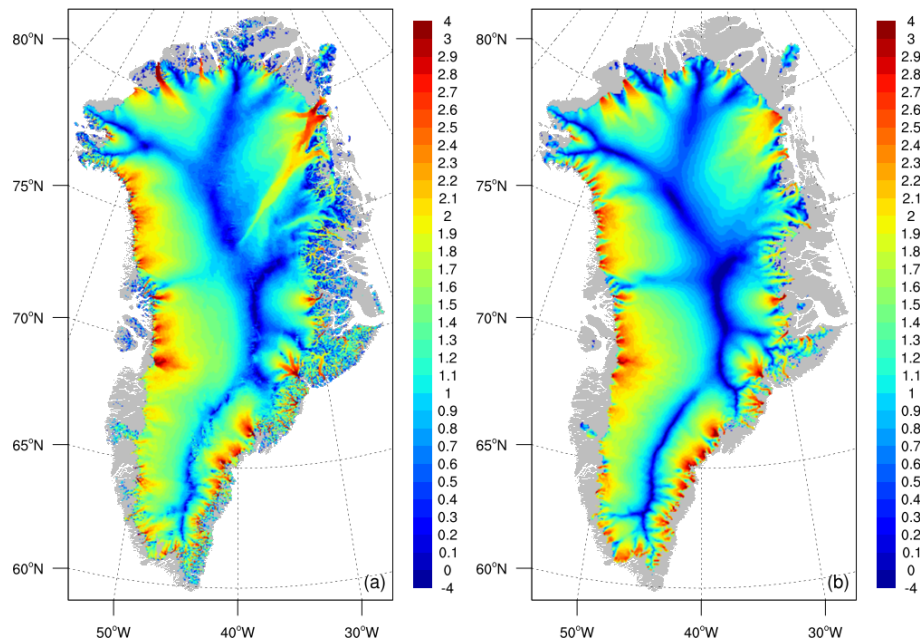
Figure 10. Surface ice speed (m/yr, log scale) for the Greenland ice sheet. (a) Observed speed (Joughin et al., 2010). (b) Simulated speed at the end of a 50 kyr Greenland spin-up without floating ice shelves. The gray background shows ice-free regions. Figs. 10–17 were created using the NCAR Command Language (Version 6.4.0) [Software]. (2017). Boulder, Colorado: UCAR/NCAR/CISL/VETS. http://dx.doi.org/10.5065/D6WD3XH5.
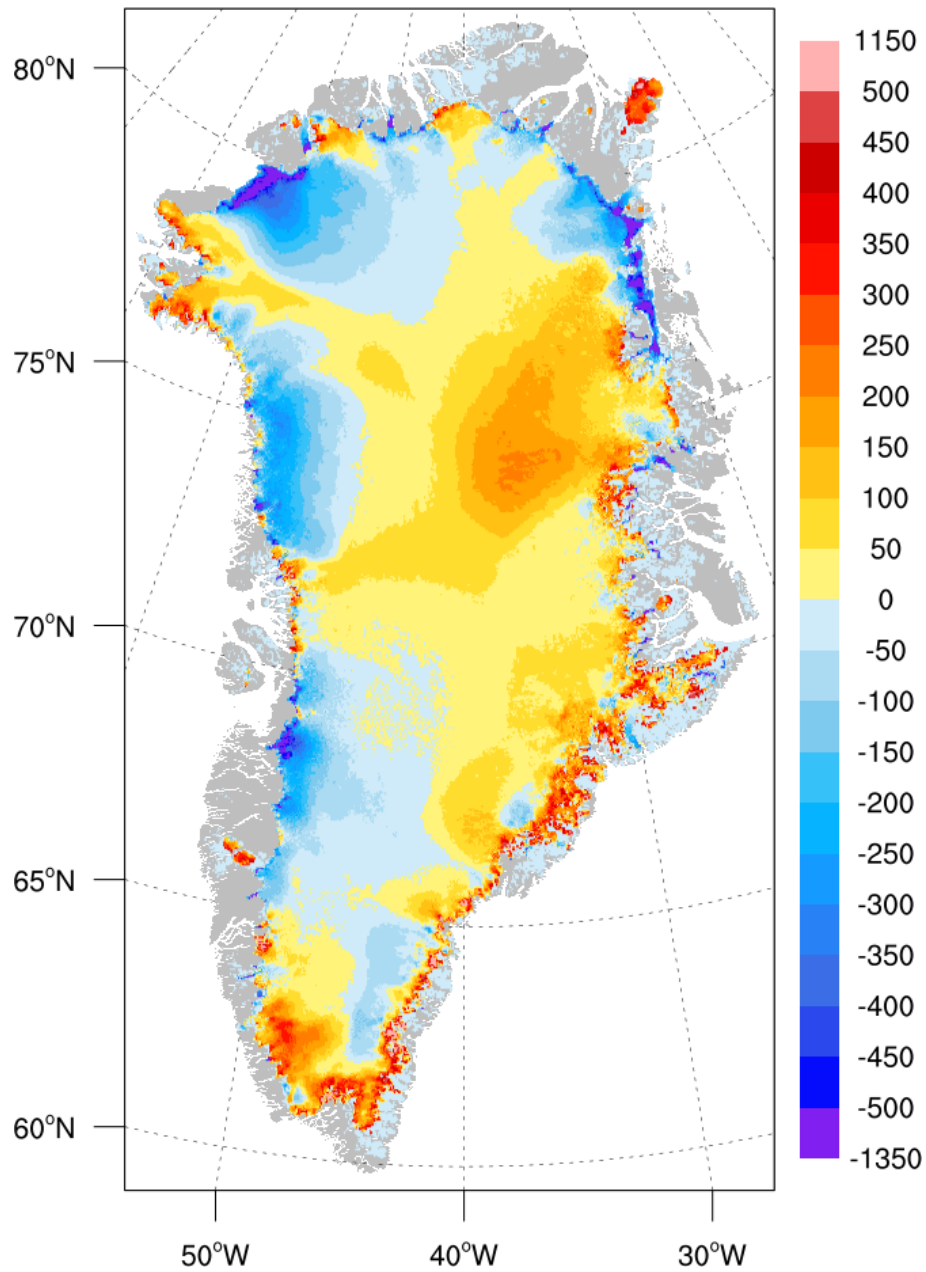
Figure 11. Difference (m) between (1) simulated ice thickness at the end of a 50 kyr Greenland spin-up without ice shelves and (2) observed thickness from Morlighem et al. (2014). The observed thickness is the model initial condition. The gray background shows regions that are ice-free at both the beginning and end of the simulation.
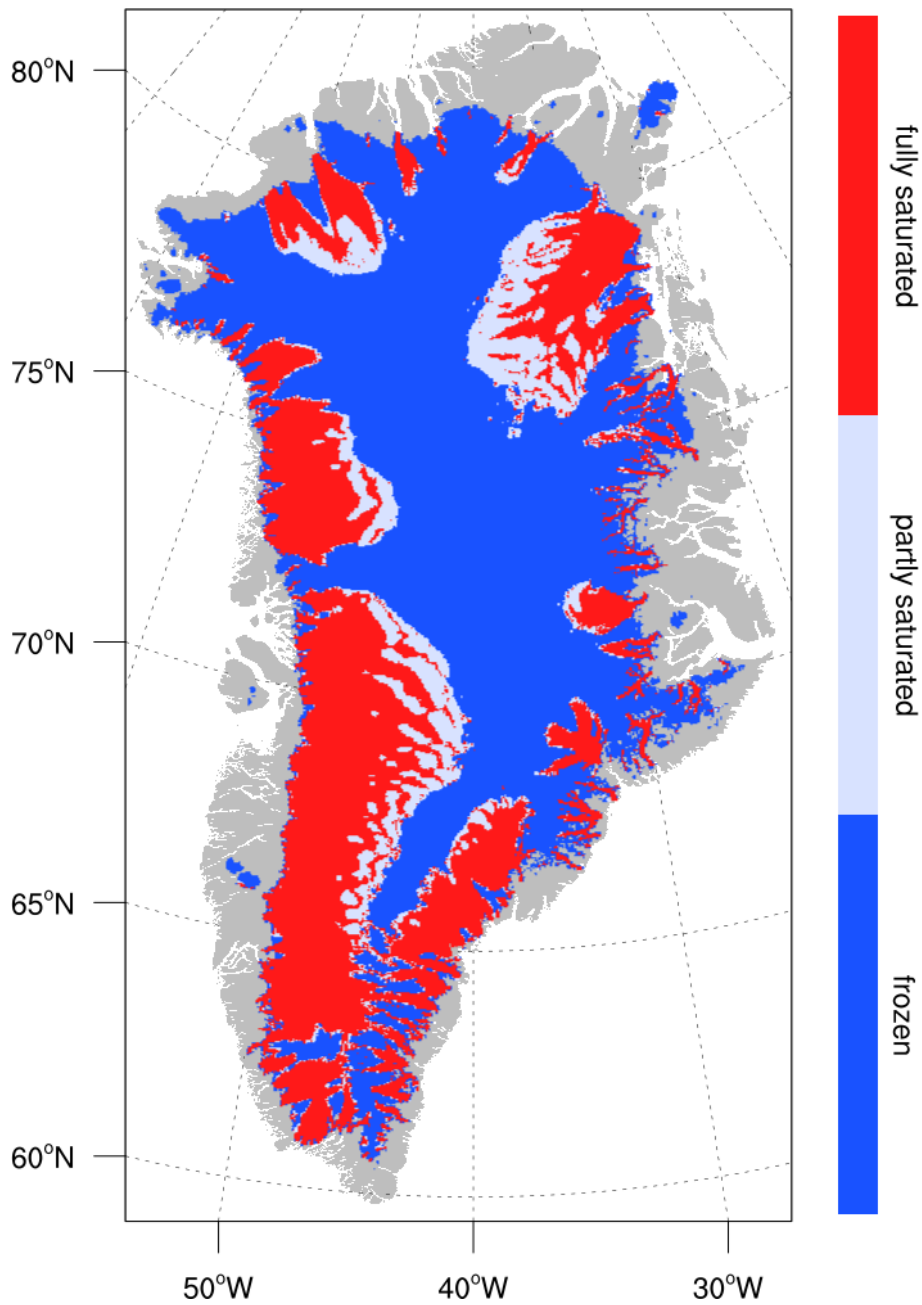
Figure 12. Basal water state at the end of a 50 kyr Greenland spin-up. Red: fully saturated bed with basal water depth W capped at the upper limit of 2 m. Light blue: partly saturated bed with $0 < W < 2$ m. Dark blue: frozen bed with $W = 0$. The color scale is based on Fig. 11 from MacGregor et al. (2015). The gray background shows ice-free regions.
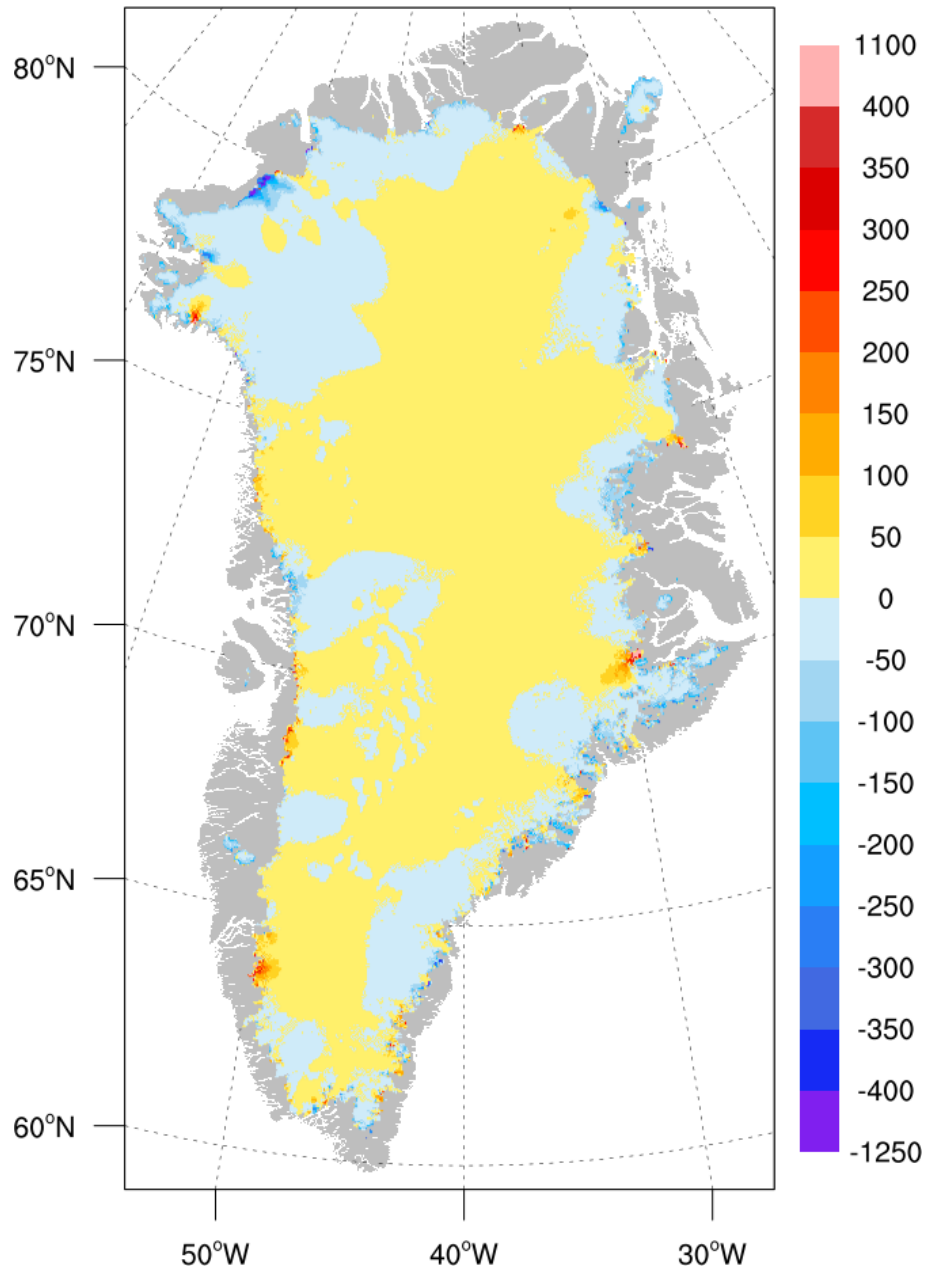
Figure 13. Difference (m) in simulated ice thickness after 10 kyr between (1) a run using the BP velocity solver and (2) a run using the DIVA velocity solver and otherwise configured identically. The gray background shows regions that are ice-free in both simulations.
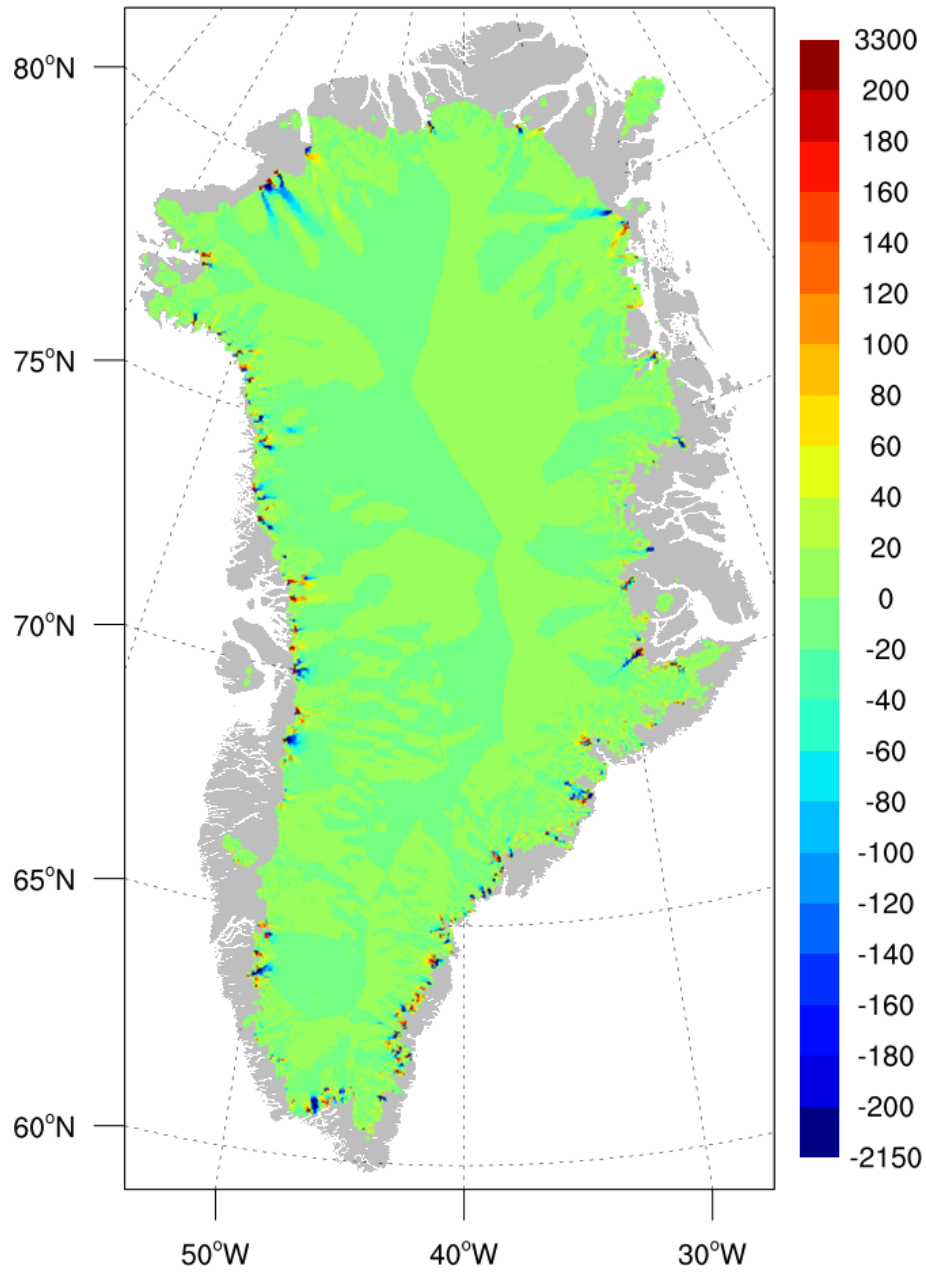
Figure 14. Difference (m/yr) in simulated ice speed after 10 kyr between (1) a run using the BP velocity solver and (2) a run using the DIVA velocity solver and otherwise configured identically. The gray background shows regions that are ice-free in both simulations.
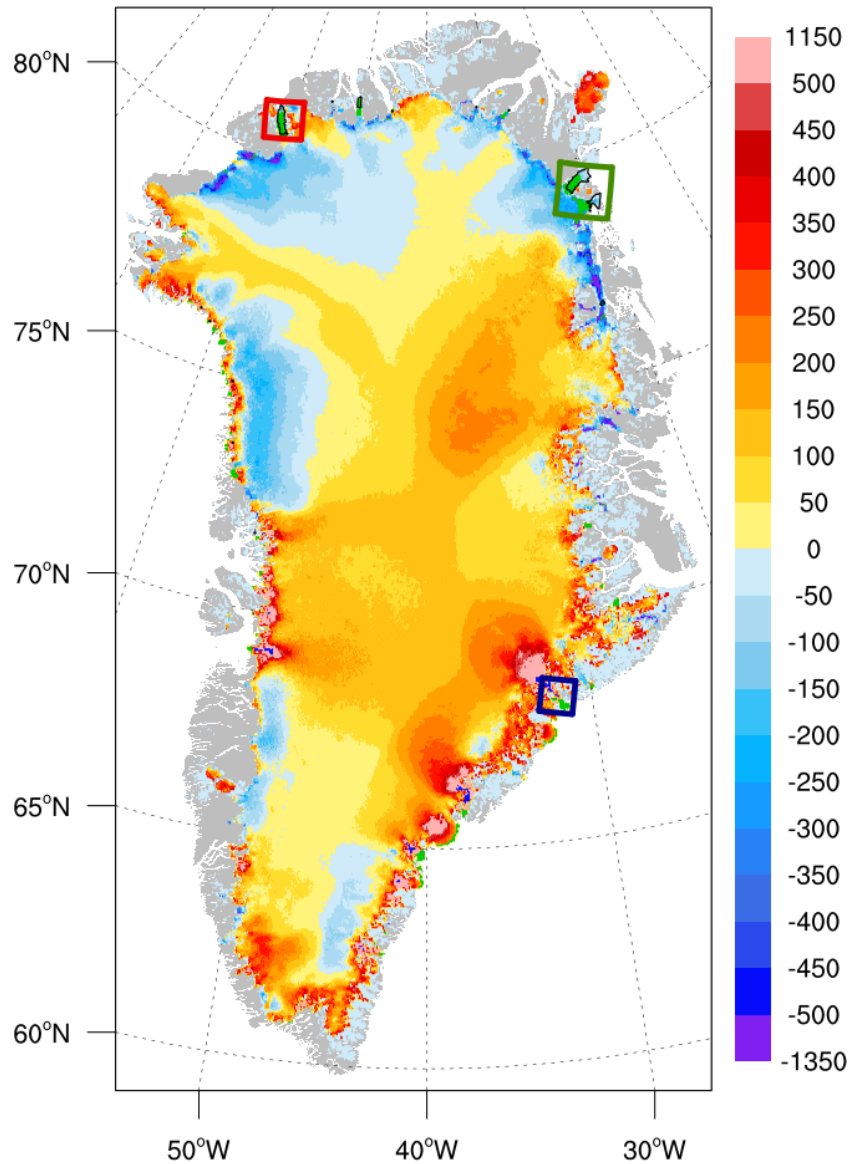
Figure 15. Difference (m) between (1) simulated ice thickness at the end of a 50 kyr Greenland spin-up with ice shelves and (2) observed thickness from Morlighem et al. (2014). The observed thickness is the model initial condition. The thickness difference is shown for grounded ice only. Regions shaded light green are floating at the end of the simulation, and black contours enclose regions that are floating in observations. The three boxes indicate floating regions highlighted in Fig. 17: Petermann Glacier (red), 79 North and Zachariae Isstrom (dark green), and Kangerlussuaq Glacier (dark blue). The gray background shows regions that are ice-free at both the beginning and end of the simulation.
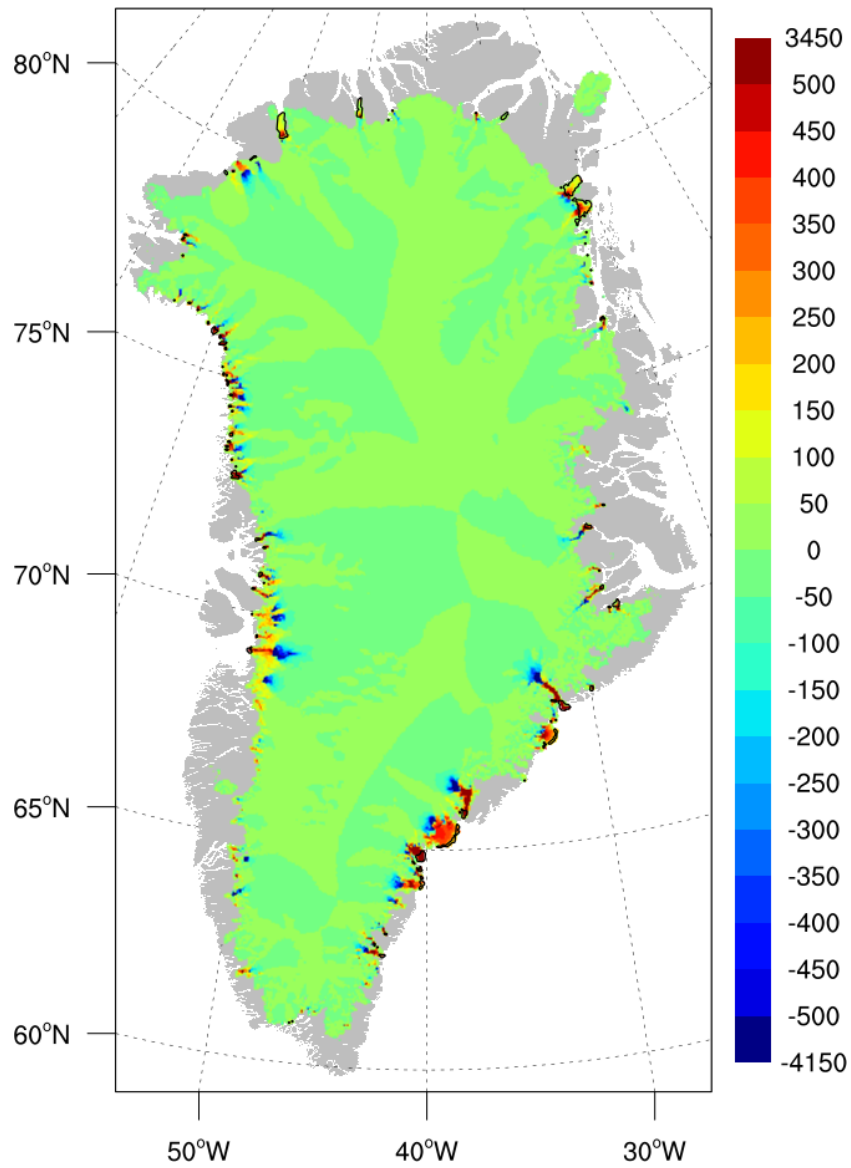
Figure 16. Difference (m/yr) in simulated surface ice speed at the end of 50 kyr Greenland spin-up runs with and without ice shelves. Black contours enclose regions that are floating in the run with ice shelves. The gray background shows regions that are ice-free in both simulations.
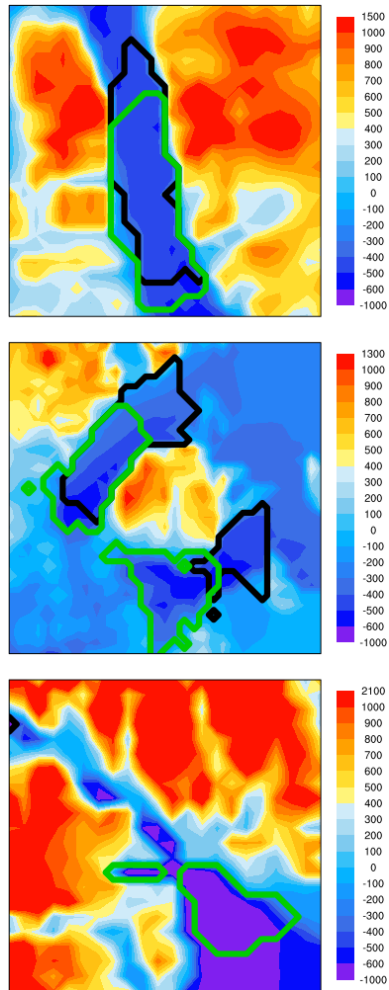
Figure 17. Close-ups of three regions with floating ice at the end of a 50 kyr Greenland spin-up with ice shelves: Petermann Glacier (top), 79 North and Zachariae Isstrom (middle), and Kangerlussuaq Glacier (bottom). These regions are enclosed by boxes in Fig. 15. The shaded color shows the bed topography (m). Black and green contours enclose regions that are floating at the start and end of the simulation, respectively. Calving fronts are located at the north, northeast, and southeast shelf boundaries, respectively from top to bottom; grounding lines are located at the south, southwest, and northwest shelf boundaries, respectively.

Variables	Definition
A	Temperature-dependent rate factor
B	Surface mass balance
B_b	Basal mass balance
b	Ice sheet bed elevation
c	Lateral calving rate
F_d	Diffusive heat flux
F_g	Geothermal heat flux
F_f	Frictional heat flux
H	Ice thickness
H_{eff}	Effective thickness at calving front
H_f	Flotation thickness
N	Effective pressure
P_0	Full overburden pressure
p	Pressure at lateral boundaries
s	Surface elevation
T	Ice temperature
T_f	Ice temperature at floating lower surface
T_{pmp}	Pressure melting point temperature
$T^{\star}$	Absolute temperature corrected for T_{pmp}
u	Horizontal ice velocity component
u_b	Basal u
v	Horizontal ice velocity component
v_b	Basal v
W	Water depth
β	Basal traction parameter
β_{eff}	Effective basal traction parameter (DIVA)
η	Effective viscosity
$\dot{\epsilon}_e$	Effective strain rate
$\dot{\epsilon}_{\mathbf{ij}}$	Strain rate tensor
σ	Vertical sigma coordinate
$\boldsymbol{\tau}_b$	Basal shear stress
$\tau_{\mathbf{ij}}$	Deviatoric stress tensor
τ_e	Effective stress
τ_c	Yield stress
τ_{ec}	Effective calving stress
Φ	Rate of internal heating
ϕ	Till friction angle

Table 1. Model variables defined in the text.

Parameters	Value	Units	Definition
g	9.81	$\mathrm{m\,s^{-2}}$	Gravitational acceleration
c_i	2009	$\mathrm{J\,kg^{-1}\,K^{-1}}$	Specific heat of ice
k_i	2.1	$\mathrm{W\,m^{-1}\,K^{-1}}$	Thermal conductivity of ice
L	335	$\mathrm{J\,kg^{-1}}$	Latent heat of melting
ρ_i	917	$\mathrm{kg\,m^{-3}}$	Ice density
ρ_o	1026	$\mathrm{kg\,m^{-3}}$	Ocean water density
ρ_w	1000	$\mathrm{kg\,m^{-3}}$	Fresh water density
n	3	-	Glen flaw low exponent
q	0.5	-	Exponent for pseudo-plastic sliding
u_0	100	$\mathrm{m\,yr^{-1}}$	Velocity scale for pseudo-plastic sliding
ϕ_{max}	40	deg	Maximum bed angle for pseudo-plastic sliding
ϕ_{min}	5	deg	Minimum bed angle for pseudo-plastic sliding
b_{max}	700	m	Upper bed limit for pseudo-plastic sliding
b_{min}	-700	m	Lower bed limit for pseudo-plastic sliding
δ	0.02	-	Minimum effective pressure relative to overburden
C_d	0.001	$\mathrm{m\,yr^{-1}}$	Till drainage rate
W_{max}	2	m	Maximum till water depth
N_0	1000	Pa	Reference effective pressure
e_0	0.69	-	Void ratio at reference effective pressure
C_C	0.12	-	Till compressibility
τ_c	10^6	Pa	Yield stress for cliff limiting
k_τ	0.0025	$\mathrm{m\,yr^{-1}\,Pa^{-1}}$	Empirical constant for calving
w_2	25	-	Empirical constant for calving
$H_c^{\min}$	75	m	Minimum thickness for calving
t_c	1	yr	Calving time scale
z_0	-200	m	Neutral elevation for sub-shelf melting
$B_{\mathrm{frz}}^{\max}$	3	m/yr	Maximum sub-shelf freezing rate
$z_{\mathrm{frz}}^{\max}$	-100	m/yr	Elevation for maximum sub-shelf freezing
$B_{\mathrm{mlt}}^{\max}$	100	m/yr	Maximum sub-shelf melting rate
$z_{\mathrm{frz}}^{\max}$	-500	m/yr	Elevation for maximum sub-shelf melting
H_0	20	m	Length scale for reducing melt in shallow cavities

Table 2. Model constant and parameter values used for simulations described in the text. The calving and sub-shelf melting parameters are used only for simulations with ice shelves; other parameters are used for all Greenland simulations.

Simulation	No. of cores	model yr/wall-clock-hr	model yr/core-hr
DIVA without shelves	144	1350	9.41
DIVA without shelves	288	2480	8.61
BP without shelves	144	28.4	0.197
BP without shelves	288	54.0	0.188
DIVA with shelves	144	1330	9.26
DIVA with shelves	288	2590	8.98

Table 3. Computational performance on NCAR's Cheyenne supercomputer for various simulations: DIVA vs. BP solvers, with and without floating ice shelves, and 144 vs. 288 processor cores.