The Utrecht Finite Volume Ice-Sheet Model (UFEMISM version 2.0) – part 1: description and idealised experiments

Constantijn J. Berends[1], Victor Azizi[2], Jorge A. Bernales[1], Roderik S. W. van de Wal[1,3]

[1]Institute for Marine and Atmospheric research Utrecht, Utrecht University, Utrecht, the Netherlands
5 [2]Netherlands eScience Center, Amsterdam, the Netherlands
[3]Faculty of Geosciences, Department of Physical Geography, Utrecht University, Utrecht, the Netherlands

Correspondence to: Constantijn J. Berends (c.j.berends@uu.nl)

Abstract. Projecting the man-made climate-change-caused mass loss of the Greenland and Antarctic ice sheets requires models that can accurately describe the physics of flowing ice, and its interactions with the atmosphere, the ocean, and the solid Earth.

10 As the irreducible uncertainty in many of these processes can only be explored by running large numbers of simulations to sample the phase-space of possible physical parameters, the computational efficiency and user-friendliness of such a model are just as relevant to its applicability as is its physical accuracy. Here, we present and verify version 2.0 of the Utrecht Finite Volume Ice-Sheet Model (UFEMISM). UFEMISM is a state-of-the-art finite-volume model which applies an adaptive grid in both space and time. Since the first version was published two years ago, v2.0 has added more accurate approximations to the

15 Stokes flow, more sliding laws, different schemes for calculating the ice thickness rates of change, a more numerically stable time-stepping scheme, more flexible and powerful mesh generation code, and a more generally applicable discretisation scheme. The parallelisation scheme has changed from a shared-memory architecture to distributed memory, enabling the user to utilise more computational resources. The version control system includes automated unit tests and benchmark experiments, to aid with model development, as well as automated installation of the required libraries, improving both user comfort and

20 reproducibility of results. The i/o now follows the NetCDF-4 standard, including automated remapping between regular grids and irregular meshes, reducing user workload for pre- and post-processing. These additions and improvements make UFEMISM v2.0 a powerful, flexible ice-sheet model, that can be used for long palaeoglaciological applications, as well as large ensemble simulations for future projections of ice-sheet retreat, and which is ready to be used for coupling within earth system models.

25 **1 Introduction**

One of the most worrisome, and at the same time most uncertain, possible long-term consequences of man-made climate change is mass loss of the Greenland and Antarctic ice sheets, leading to global sea-level rise (Oppenheimer et al., 2019; van de Wal et al., 2019; Fox-Kemper et al., 2021). Projections for the Antarctic contribution to sea-level rise in 2100 under RCP8.5, which were studied in the Ice-Sheet Model Intercomparison for CMIP6 (ISMIP6) range from −2.5 cm to +17 cm (Seroussi et

30 al., 2020), with a possible high-end value of +59 cm (van de Wal et al., 2022) and consequently much more on longer time

scales. Part of this large uncertainty stems from poorly constrained physical properties and processes in the Antarctic ice sheet system, including subglacial conditions (e.g. Kazmierczak et al., 2022; Berends et al., 2023a), basal sliding (Sun et al., 2020), interactions between the ice shelf and the ocean in the sub-shelf cavity (e.g. Burgard et al., 2022; Berends et al., 2023b), and ice-dynamical processes (e.g. Rückamp et al., 2022). However, even in idealised experiments where all these quantities are

35 known perfectly, different ice-sheet models can predict rates of sea-level rise that differ by a factor of three (Cornford et al., 2020). This represents the uncertainty arising from the numerical models themselves, which disagree on how to translate the physical equations to computer code. These model differences include the way the momentum balance (typically represented by the Stokes equations) is approximated, the choice of grid, the numerical treatment of discontinuities in basal friction and melt rates at the grounding line, and the way the model is initialised. Sampling both this model-intrinsic uncertainty, and the

40 uncertainty in the physical properties and processes of the actual ice sheet, require ice-sheet models that have the computational power, and the flexibility, to perform large numbers of simulations, at an adequate resolution to capture these processes. To meet this challenge, many research groups working on ice-sheet modelling have recently directed their efforts at creating new, more powerful ice-sheet models (e.g. Pattyn, 2017; Hoffman et al., 2018; Quiquet et al., 2018; Lipscomb et al., 2019; Robinson et al., 2020; Berends et al., 2022).

45

Here, we present version 2.0 of the Utrecht Finite Volume Ice-Sheet Model (UFEMISM). Version 1.0 (Berends et al., 2021) was the second ice-sheet model to use a dynamic adaptive mesh, the first being BISICLES (Cornford et al., 2013). This structure allows the model to resolve the grounding line at high (< 5 km) resolutions during multi-millennial simulations. Since the publication of v1.0, many new features have been added to UFEMISM, and many existing features have been improved in

50 terms of power, flexibility, and user-friendliness. In Sect. 2, we provide the physical equations for ice flow that are solved by the model. This includes several approximations to the Stokes equations for the momentum balance (Sect. 2.2), several sliding laws (Sect. 2.3), a new numerical scheme for treating basal friction at the grounding line (Sect. 2.4), different temporal discretisation schemes to calculate the ice geometry rates of change (Sect. 2.5), and a new adaptive time-stepping scheme (Sect. 2.6). In Sect. 3, we describe several improvements that were made to the model code. This includes a change from a

55 shared-memory to a distributed-memory architecture (Sect. 3.1), and a thoroughly reworked module for input/output (i/o) that now follows the NetCDF-4 standard (Unidata, 2023) and is much more flexible and user-friendly (Sect. 3.2). It also includes a version control system that includes automated unit tests and benchmark experiments to aid in developing robust code, and automated installation of external libraries to improve user-friendliness and reproducibility of results (Sect. 3.3). In Sect. 4, we present results of a number of idealised-geometry experiments to verify the new model physics and numerics.

60

This paper, part 1, focuses on the basic mathematics and physics of the model, and their verification in idealised benchmark experiments. Part 2, which is submitted for review and publication separately (Bernales et al, *in prep.*), focuses on model additions required for the application of UFEMISM to realistic ice sheets such as those in Greenland and Antarctica. It includes

descriptions of the routines for inverting for subglacial bed roughness and for ocean temperatures in shelf cavities, different

65 sub-shelf melt parameterisations, initialisation approaches, and future projections of mass loss.

2 Model description

2.1 General

UFEMISM is a large-scale ice-sheet model. It solves an approximation of the Stokes equations to calculate the flow velocities of the ice. These are combined with the ice accumulation/loss rates at the surface, basal, and lateral boundaries of the ice sheet

70 to find the thinning/thickening rates of the ice, which are integrated through time to find the evolution of the ice sheet. Note that hereafter, we will refer to UFEMISM version 1.0 as "v1.0", and to the version 2.0 presented here as "v2.0".

The main distinguishing feature of UFEMISM compared to many other ice-sheet models is the use of a dynamic adaptive grid. The two-dimensional plane on which the model operates is discretised as an irregular triangular mesh, an example of which is

75 shown in Fig. 1.

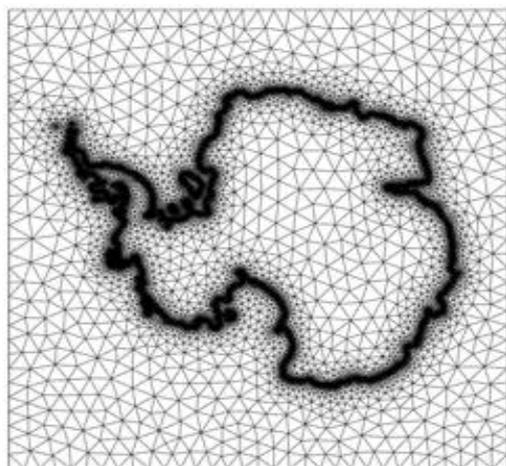



Fig. 1: A demo mesh generated by UFEMISM for the Antarctic ice sheet, using a 10 km grounding-line resolution, and up to 200 km for the ice-sheet interior and the open ocean.

Earlier research in ice-sheet modelling has shown that the accuracy of a numerical model is particularly sensitive to the spatial

80 resolution of the grid around the grounding line (Gladstone et al., 2012; Pattyn et al., 2012). There, the discontinuous basal friction, which is non-zero underneath the grounded ice but zero underneath the floating ice, causes strong gradients in englacial stresses and therefore the ice geometry. Although different solutions have been presented in earlier literature to produce accurate results at coarser resolutions (see Sect. 2.4), the resolutions required at the grounding line are still much higher than those needed in the slow-moving interior of the ice sheet, in order to achieve the same level of accuracy in the ice

85 thickness evolution. As the demand for both the temporal coverage and number of ice-sheet simulations increases,

computational efficiency becomes a more important property of ice-sheet models. Using a uniform high resolution over the entire ice sheet, while it is only needed in the relatively small area around the grounding line, is therefore undesirable. UFEMISM solves this problem by using a mesh that has a high resolution only where needed, and a low resolution where possible. This is the "adaptive" part of the mesh. However, as the ice-sheet geometry changes over the course of a model simulation, the location of the grounding line changes as well. This means that, after a while, the grounding line might no longer be located within the high-resolution area of the mesh. Some models solve this problem by using a mesh with a high resolution over a wider area, enveloping the expected future migration of the grounding line. While this is a feasible approach for century-scale simulations, doing this for the multi-millennial applications for which UFEMISM is also intended would mean creating a mesh with a very large high-resolution area, thus defeating the purpose of the adaptive mesh. Instead, UFEMISM periodically checks the mesh fitness to the modelled ice-sheet geometry and, if needed, automatically creates a new mesh that conforms to the new ice-sheet geometry (with a high-resolution area around the new grounding-line position), remapping the model data from the old mesh to the new one. This is the "dynamic" part of the mesh. Berends et al. (2021) showed that this approach results in good computational performance, with no significant loss of accuracy.

While the general principles of the dynamic adaptive mesh have not changed significantly in v2.0 with respect to v1.0, the way these principles are adapted into computer code has changed in several ways. The new mesh generation code, the scheme used to discretise the partial differential equations of the model on the mesh, and the scheme used to remap data from one mesh to another, are presented in Appendices A, B, and C, respectively.

2.2 Momentum balance

UFEMISM v2.0 includes solvers for several different approximations to the Stokes equations. The most complete is the Blatter-Pattyn approximation (BPA; Pattyn, 2003), which is described in Sect. 2.2.1. The depth-integrated viscosity approximation (DIVA; Goldberg, 2011; Sect. 2.2.2), the shallow shelf approximation (SSA; Morland, 1987; Sect. 2.2.3), the shallow ice approximation (SIA; Morland and Johnson, 1980; Sect. 2.2.4), and the hybrid SIA/SSA (Bueler and Brown, 2009; Sect. 2.2.5) can all be derived by neglecting increasingly more terms in the BPA.

Table 1: Model symbols, units, and default values where applicable

Symbol	Description	Units	Value
A	Glen's flow law factor	$Pa^{-n}\ yr^{-1}$	
b	Bedrock elevation	m	
β	Basal friction coefficient	$Pa\ m^{-1}\ yr$	
$\dot{\varepsilon}_e$	Effective strain rate	yr^{-1}	
g	Gravitational acceleration	$m\ s^{-2}$	9.81
H	Ice thickness	m	

η	Effective viscosity	Pa yr	
n	Glen's flow law exponent		3
ρ	Density of ice	kg m^{-3}	910
s	Surface elevation	m	
$\boldsymbol{u}$	Horizontal ice velocity vector	m yr^{-1}	
u	Horizontal ice velocity in x-direction	m yr^{-1}	
v	Horizontal ice velocity in y-direction	m yr^{-1}	
w	Vertical ice velocity	m yr^{-1}	
ζ	Scaled vertical coordinate		$0 - 1$

2.2.1 Blatter-Pattyn Approximation

The BPA arises from the Stokes equations by assuming hydrostatic equilibrium and neglecting the stresses arising from horizontal variations in the vertical velocity (i.e. $\frac{\partial w}{\partial x} \ll \frac{\partial u}{\partial z}, \frac{\partial w}{\partial y} \ll \frac{\partial v}{\partial z}$; Pattyn, 2003). This means that the pressure p and the vertical velocity w disappear as degrees of freedom from the momentum balance, so that only the horizontal velocities u, v

115 remain to be solved for. The BPA produces ice velocities that are generally very close to those from the Stokes equations (Pattyn et al., 2008). While less computationally expensive to solve than the Stokes equations, it is still significantly slower than the DIVA or the hybrid SIA/SSA, owing to the fact that, where those approximations either parameterise or neglect vertical variations in velocities or strain rates, the BPA solves for such variations explicitly. This requires the model to discretise the vertical dimension as well, whereas the DIVA and the hybrid SIA/SSA operate in the two-dimensional plane,

120 yielding a system of linear equations that is larger by a factor of the number of vertical layers in the model.

The set of partial different equations that must be solved in order to find the 3-D horizontal ice velocities u, v reads:

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

$$\frac{\partial}{\partial y}\left[2\eta\left(2\frac{\partial v}{\partial y} + \frac{\partial u}{\partial x}\right)\right] + \frac{\partial}{\partial x}\left[\eta\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right] + \frac{\partial}{\partial z}\left[\eta\frac{\partial v}{\partial z}\right] = \rho g \frac{\partial s}{\partial y}. \tag{1b}$$

125 The effective viscosity η is related to the effective strain rate $\dot{\varepsilon}$ by Glen's flow law (Glen, 1955):

$$\eta = \frac{1}{2}A^{-1/n}\dot{\varepsilon}_e^{\frac{1-n}{n}}. \tag{2}$$

The effective strain rate $\dot{\varepsilon}_e$ is given by:

$$\dot{\varepsilon}_e = \left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \frac{\partial u}{\partial x}\frac{\partial v}{\partial y} + \frac{1}{4}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \frac{1}{4}\left(\frac{\partial u}{\partial z}\right)^2 + \frac{1}{4}\left(\frac{\partial v}{\partial z}\right)^2\right]^{1/2}. \tag{3}$$

At the ice surface, the zero-stress boundary condition reads:

130
$$2\frac{\partial s}{\partial x}\left(2\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + \frac{\partial s}{\partial y}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) - \frac{\partial u}{\partial z} = 0, \tag{4a}$$

$$2\frac{\partial s}{\partial y}\left(2\frac{\partial v}{\partial y}+\frac{\partial u}{\partial x}\right)+\frac{\partial s}{\partial x}\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)-\frac{\partial v}{\partial z}=0.$$ (4b)

The similar zero-stress boundary condition at the ice base includes a basal friction term:

$$2\frac{\partial b}{\partial x}\left(2\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)+\frac{\partial b}{\partial y}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)-\frac{\partial u}{\partial z}+\frac{\beta}{\eta}u=0,$$ (5a)

$$2\frac{\partial b}{\partial y}\left(2\frac{\partial v}{\partial y}+\frac{\partial u}{\partial x}\right)+\frac{\partial b}{\partial x}\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)-\frac{\partial v}{\partial z}+\frac{\beta}{\eta}v=0.$$ (5b)

135 In order to solve the BPA, the vertical dimension must be discretised as well. This is not straightforward, as the surface and base of the ice are generally not flat, and evolve over time, so that these surfaces will generally move in between grid points in the vertical. In UFEMISM, as in most other ice-sheet models that solve a three-dimensional version of the momentum balance, this problem is solved by introducing a terrain-following coordinate transformation, which is described in Appendix D.

140 **2.2.2 Depth-integrated viscosity approximation**

The DIVA, which is the default option for the momentum balance approximation in v2.0, arises by neglecting vertical variations in the membrane stresses in the BPA (i.e. $\frac{\partial}{\partial z}\left(\frac{\partial u}{\partial x},\frac{\partial u}{\partial y},\frac{\partial v}{\partial x},\frac{\partial v}{\partial y}\right)\approx 0$), and integrating the resulting equations vertically. This means that, whereas the BPA is solved in three dimensions, the DIVA operates in the two-dimensional plane, greatly reducing the computational expense of solving it. In the Ice-Sheet Model Intercomparison Project for Higher-Order Models

145 (ISMIP-HOM; Pattyn et al., 2008) experiments, the DIVA produces velocities that agree well with the Stokes solution down to horizontal scales for basal topographical features of about 20 km (Berends et al., 2022; Robinson et al., 2022; this study, Sect. 4.1).

The partial differential equations of the DIVA read:

150
$$\frac{\partial}{\partial x}\left[2\eta H\left(2\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right]+\frac{\partial}{\partial y}\left[\eta H\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right]-\beta_{\text{eff}}u=\rho gH\frac{\partial s}{\partial x},$$ (6a)

$$\frac{\partial}{\partial y}\left[2\eta H\left(2\frac{\partial v}{\partial y}+\frac{\partial u}{\partial x}\right)\right]+\frac{\partial}{\partial x}\left[\eta H\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)\right]-\beta_{\text{eff}}v=\rho gH\frac{\partial s}{\partial y}.$$ (6b)

Here, β_{eff} is a term describing both basal friction and vertical shear stress:

$$\beta_{\text{eff}}=\frac{\beta}{1+\beta F_2}.$$ (7)

The integral term F_2, which can be thought of as the depth-integral of the square of the inverse viscosity, is defined as:

155
$$F_n=\int_b^h\frac{1}{\eta}\left(\frac{s-z}{H}\right)^n dz.$$ (8)

Note that, in Eq. 7, $n=2$; Eq. 8 lists the general form because elsewhere in the DIVA, F_1 appears as well. A comprehensive derivation of these and other required equations, including a step-by-step approach for how to solve them numerically, can be found in Lipscomb et al. (2019).

2.2.3 Shallow shelf approximation

160 The SSA arises by neglecting all vertical variations in the BPA (i.e. $\frac{\partial u}{\partial z}, \frac{\partial v}{\partial z} \approx 0$), leaving only the membrane stresses, and then vertically integrating the result. This is generally accepted to be a valid approximation in areas of negligible basal shear stress, such as ice shelves, as well as well-lubricated, fast-flowing ice streams.

The partial differential equations of the SSA read:

165
$$\frac{\partial}{\partial x}\left[2\eta H\left(2\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right]+\frac{\partial}{\partial y}\left[\eta H\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)\right]-\beta u=\rho g H\frac{\partial s}{\partial x}, \tag{9a}$$

$$\frac{\partial}{\partial y}\left[2\eta H\left(2\frac{\partial v}{\partial y}+\frac{\partial u}{\partial x}\right)\right]+\frac{\partial}{\partial x}\left[\eta H\left(\frac{\partial v}{\partial x}+\frac{\partial u}{\partial y}\right)\right]-\beta v=\rho g H\frac{\partial s}{\partial y}. \tag{9b}$$

Neglecting the same strain rates reduces the expression for the effective strain rate that is used in Glen's flow law to:

$$\dot{\varepsilon}_e=\left[\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2+\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}+\frac{1}{4}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2\right]^{1/2}. \tag{10}$$

2.2.4 Shallow ice approximation

170 The SIA arises by neglecting the membrane stresses in the BPA (i.e. $\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \approx 0$), leaving only the vertical shear strain rates. This is generally accepted to be a valid approximation for the thick, slow-moving ice in the interior of the Greenland and Antarctic ice sheets, where the flow is dominated by deformation due to vertical shearing, rather than by basal sliding. These assumptions simplify the Stokes equations to:

$$\frac{\partial}{\partial z}\left(\eta\frac{\partial u}{\partial z}\right)=\rho g\frac{\partial s}{\partial x}. \tag{11a}$$

175
$$\frac{\partial}{\partial z}\left(\eta\frac{\partial v}{\partial z}\right)=\rho g\frac{\partial s}{\partial y}. \tag{11a}$$

Similarly, the effective strain rate that is used in Glen's flow law reduces to:

$$\dot{\varepsilon}_e=\left[\frac{1}{4}\left(\frac{\partial u}{\partial z}\right)^2+\frac{1}{4}\left(\frac{\partial v}{\partial z}\right)^2\right]^{1/2}. \tag{12}$$

Substituting Eq. 12 into Eq. 11, and assuming a stress-free boundary condition at the ice surface and a no-slip boundary condition at the ice base, leads to the following analytical solution for the vertical profile of the horizontal ice velocity:

180
$$\boldsymbol{u}(z)=-2(\rho g)^n|\nabla s|^{n-1}\nabla s\int_b^z A\big(T(\zeta)\big)(s-\zeta)^n d\zeta. \tag{13}$$

2.2.5 Hybrid shallow ice / shallow shelf approximation

In this approach, the SIA and SSA are solved separately, following the approach proposed by Bueler and Brown (2009). Based on the observation that the flow regime in most areas of an ice sheet is generally dominated by either vertical shear (described by the SIA) or by basal sliding (described by the SSA), the two solutions are then simply added together to find an
185 approximation of the flow of the entire ice sheet. This approach produces accurate results in terms of large-scale ice flow (e.g.

Bueler and Brown, 2009; Berends et al., 2022), but starts to deviate significantly from the Stokes solution earlier than the DIVA as the length scale decreases (Berends et al., 2022; this study).

2.2.6 Vertical velocities

The assumption that glacial ice is incompressible is expressed mathematically as:

190
$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \tag{14}$$

The BPA, the DIVA, and the (hybrid) SIA/SSA only solve for the horizontal velocities u, v. From those, the horizontal divergence $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ can be calculated. Integrating Eq. 14 in the vertical dimension then yields the vertical velocity w:

$$w(z) = w(z = b) - \int_b^z \left(\frac{\partial u}{\partial x}(\zeta) + \frac{\partial v}{\partial y}(\zeta)\right) d\zeta. \tag{15}$$

Here too, the terrain-following coordinate transformation must be applied before evaluating the vertical integral. The way this
195 is done in UFEMISM is described in Appendix E.

2.3 Sliding laws

UFEMISM v2.0 includes a number of different sliding laws for the user to choose from, which relate the basal shear stress $\boldsymbol{\tau}_b$ to the basal velocity $\boldsymbol{u}_b$ through the basal friction coefficient $\beta = \frac{|\boldsymbol{\tau}_b|}{|\boldsymbol{u}_b|}$. All sliding laws are presented here as they are coded in the model, with the basal friction coefficient β expressed as a function of the basal speed $u_b = |\boldsymbol{u}_b|$. The first option is a
200 Weertman-type sliding law (Weertman, 1957):

$$\beta = C_w u_b^{\frac{1}{m}-1}. \tag{16}$$

Here, C_w represents the (spatially variable) subglacial bed roughness.

The second option is a Coulomb-type sliding law (Iverson et al., 1998):

$$\beta = N \tan\varphi\, u_b^{-1}. \tag{17}$$

205 Here, N is the effective pressure between the ice and the bedrock, which is equal to the ice overburden pressure minus the subglacial water pressure. The (spatially variable) till friction angle φ is a measure for the subglacial bed roughness.

The third option is a Budd-type sliding law, proposed by Bueler and van Pelt (2015):

$$\beta = N \tan\varphi \frac{u_b^{q-1}}{u_0^q}. \tag{18}$$

Here, u_0 is a transition velocity, with a default (configurable) value of 100 m yr^{-1}. Note that this is a Budd-type sliding law
210 (i.e. a power-law dependence on velocity, scaled with the effective pressure) for the current choice of exponent $q = 0.3$; for $q = 1$, this becomes a regularised Coulomb sliding law, with no dependence on velocity. This sliding law was the only option in UFEMISM 1.0 (Berends et al., 2021).

The fourth option is the hybrid sliding law proposed by Tsai et al. (2015), as formulated by Asay-Davis et al. (2016):

$$\beta = \min\left(\alpha^2 N, \beta^2 u_b^{\frac{1}{m}}\right) u_b^{-1}. \tag{19}$$

215   Note that here, the (spatially variable) subglacial bed roughness is described by two separate parameters: α^2 for the Coulomb-type part of the friction, and β^2 (which is not the square of the basal friction coefficient β, but a confusingly named separate entity, which we maintain for the sake of consistency with earlier literature) for the Weertman-type part.

The final option is the hybrid sliding law proposed by Schoof (2005), as formulated by Asay-Davis et al. (2016):

$$\beta = \frac{\beta^2 u_b^{\frac{1}{m}} \alpha^2 N}{[\beta^{2m} u_b + (\alpha^2 N)^m]^{\frac{1}{m}}} u_b^{-1}. \tag{20}$$

220   Note that the terms on the right-hand side of Eq. 20 are again the β^2 term from Eq. 19. In the idealised-geometry experiments presented here, the bed roughness is spatially uniform. For applications to realistic ice sheets, UFEMISM v2.0 includes routines for inverting the bed roughness by nudging. These are presented in part 2 of this work (Bernales et al., *in prep.*).

2.4 Sub-grid friction scaling

UFEMISM v1.0 used a grounding-line flux condition (Schoof, 2007; Pollard and DeConto, 2012) to improve grounding-line
225   migration. However, the flux condition was derived by Schoof (2007) for the zero-buttressing case. There is not yet a clear solution for including buttressing in this solution (Reese et al., 2018). Furthermore, while the implementation of this scheme in v1.0 performed well in idealised geometries, it frequently resulted in numerical instability in the more complex geometries encountered in e.g. the Antarctic ice sheet. Therefore, in UFEMISM v2.0 the flux condition has been replaced by a sub-grid friction scaling scheme, following the approach used in PISM (Feldmann et al., 2014), CISM (Leguy et al., 2021), and IMAU-
230   ICE (Berends et al., 2022). Here, the area fraction of each mesh triangle (where the velocities are defined) that is covered by grounded ice, is calculated by bilinearly interpolating the thickness above floatation on the three vertices spanning the triangle. The basal friction coefficient β that is calculated using the sliding law, is then multiplied with this grounded fraction, before being used to solve the momentum balance. This approach is much more numerically stable, does not require any special treatment to include buttressing, and works well in both idealised and realistic geometries.

2.5 Conservation of mass

After the momentum balance has been solved to find the ice velocities, the condition of conservation of mass can be used to find the rates of change of the ice geometry. Conservation of ice mass for a shallow layer of incompressible ice in the 2-D plane is expressed mathematically as:

$$\frac{\partial H}{\partial t} = -\nabla \cdot (\boldsymbol{u} H) + m. \tag{21}$$

240   Here, m is the net mass balance, including terms at the ice base, the ice surface, and the lateral boundaries. Eq. 21 is discretised spatially using the finite volume scheme that lent UFEMISM its name, which is derived in Appendix F, resulting in the following equation:

$$\frac{\partial H^i}{\partial t} = -M_{\text{divQ}}H^i + m^i. \tag{22}$$

Here, the ice thickness vector H^i contains the values of H on all the vertices i. M_{divQ} is a matrix whose coefficients depend on

245 the mesh geometry and the ice velocities, which can be multiplied with the ice thickness vector H^i to find the ice flux divergence $\nabla \cdot (\boldsymbol{u}H)^i = M_{\text{divQ}}H^i$. UFEMISM v2.0 offers three different options to discretise Eq. 22 temporally: an explicit scheme, an implicit scheme, and a semi-implicit scheme. In all three cases, the thickness rate of change $\frac{\partial H}{\partial t}$ is discretised using a simple first-order scheme. In the explicit scheme, all terms on the right-hand side of Eq. 22 are defined at time t:

$$\frac{H^{i,t+\Delta t}-H^{i,t}}{\Delta t} = -M_{\text{divQ}}H^{i,t} + m^{i,t}. \tag{23}$$

250 Rearranging the terms yields the following expression, which can be evaluated to find $H^{i,t+\Delta t}$:

$$H^{i,t+\Delta t} = \left(I - M_{\text{divQ}}\Delta t\right)H^{i,t} + m^{i,t}\Delta t. \tag{24}$$

In the implicit scheme, the ice thickness on the right-hand side of Eq. 22 is defined at time $t + \Delta t$:

$$\frac{H^{i,t+\Delta t}-H^{i,t}}{\Delta t} = -M_{\text{divQ}}H^{i,t+\Delta t} + m^{i,t}. \tag{25}$$

Rearranging the terms yields the following matrix equation that can be solved for $H^{i,t+\Delta t}$:

255 $$\left(I + M_{\text{divQ}}\Delta t\right)H^{i,t+\Delta t} = H^{i,t} + m^{i,t}\Delta t. \tag{26}$$

Lastly, the semi-implicit scheme is derived by defining the ice thickness on the right-hand side of Eq. 22 as the weighted average of $H^{i,t}$ and $H^{i,t+\Delta t}$:

$$\frac{H^{i,t+\Delta t}-H^{i,t}}{\Delta t} = -M_{\text{divQ}}(f_s H^{i,t+\Delta t} + (1-f_s)H^{i,t}) + m^{i,t}. \tag{27}$$

Here, using a coefficient $f_s = 0$ implies a fully explicit scheme, $f_s = 1$ implies a fully implicit scheme, $0 < f_s < 1$ implies a

260 semi-implicit scheme, and $f_s > 1$ implies an over-implicit scheme. Rearranging the terms yields the following matrix equation that can be solved for $H^{i,t+\Delta t}$:

$$\left(I + f_s M_{\text{divQ}}\Delta t\right)H^{i,t+\Delta t} = \left(I - \Delta t(1-f_s)M_{\text{divQ}}\right)H^{i,t} + m^{i,t}\Delta t. \tag{28}$$

2.6 Time stepping

v2.0 uses the predictor/corrector (PC) time-stepping scheme by Robinson et al. (2020). Whereas the SIA and SSA both have

265 well-defined critical time steps, no such condition has yet been derived for the DIVA or the BPA. The predictor/corrector scheme essentially operates by calculating two solutions of $H^{i,t+\Delta t}$: one with an explicit solution of the ice velocity u, and one with a pseudo-implicit solution. The difference between the two solutions of $H^{i,t+\Delta t}$ is a measure for the temporal discretisation error, which can be used to adapt the time step; if the error is found to be increasing, the time step is decreased, and vice versa. Robinson et al. (2022) showed that this scheme is particularly suitable to the DIVA (and, by extension, the BPA), where the

270 error is less sensitive to larger time steps than in the hybrid SIA/SSA, due to the weaker dependence of the velocity on the local surface slope.

A time step in the PC scheme consists of three parts: the predictor step, the update step, and the corrector step. First, in the predictor step, the "predicted" ice thickness is calculated, based on the current ice thickness and the current velocity solution:

275
$$H_{\text{pred}}^{t+\Delta t} = H^t + \Delta t^t \left[\left(1 + \frac{\zeta_t}{2}\right) \frac{\partial H}{\partial t}(H^t, \boldsymbol{u}^t) - \frac{\zeta_t}{2} \frac{\partial H}{\partial t}(H^{t-\Delta t}, \boldsymbol{u}^{t-\Delta t}) \right].$$
(29)

Here, $\zeta_t = \frac{\Delta t^t}{\Delta t^{t-\Delta t}}$ is the ratio between the current and the previous time steps.

Then, in the update step, a new ice velocity solution is calculated for the predicted ice thickness:

$$\boldsymbol{u}^{t+\Delta t} = \boldsymbol{u}\left(H_{\text{pred}}^{t+\Delta t}\right).$$
(30)

Lastly, in the corrector step, the "corrected" ice thickness is calculated, based on the current ice thickness and the new velocity

280 solution:

$$H_{\text{corr}}^{t+\Delta t} = H^t + \frac{\Delta t^t}{2} \left[\frac{\partial H}{\partial t}(H^t, \boldsymbol{u}^t) + \frac{\partial H}{\partial t}\left(H_{\text{pred}}^{t+\Delta t}, \boldsymbol{u}^{t+\Delta t}\right) \right].$$
(31)

The discretisation error τ in the ice thickness is estimated based on the difference between the predicted and the corrected ice thicknesses:

$$\tau^{t+\Delta t} = \frac{\zeta_t \left(H_{\text{corr}}^{t+\Delta t} - H_{\text{pred}}^{t+\Delta t}\right)}{(3\zeta_t + 3)\Delta t^t}.$$
(32)

285 The time step is then adapted based on the maximum discretisation error:

$$\Delta t^{t+\Delta t} = \left(\frac{\epsilon}{\max|\tau^{t+\Delta t}|}\right)^{(k_I + k_p)} \left(\frac{\epsilon}{\max|\tau^t|}\right)^{-k_p} \Delta t^t.$$
(33)

Here, ϵ is the target discretisation error (configurable, default value 3 m), and $k_I = 0.2$ and $k_p = 0.1$ are tuning parameters (values taken from Robinson et al., 2020).

3 Code

290 ### 3.1 Parallelisation

A major change in v2.0 with respect to v1.0 is the switch from a shared-memory architecture, where all data is stored in the same memory chip which all processors can access, to a distributed-memory architecture, where the data is distributed over many memory chips, so that each processor can only access part of the data. The distributed-memory architecture is slower than a shared-memory program running on the same number of processors, as data frequently needs to be exchanged between 295 the different processors. However, the shared-memory architecture can only run on as many processors as can access the same memory chip (typically 32 or 64 on many high-performance scientific computing systems). The distributed-memory architecture is not limited in this way, allowing the user to scale up to far larger numbers of processors if necessary. Solving the matrix equation representing the momentum balance is currently the most computationally demanding part of the model by far, often accounting for more than 80 % of the total computation time of a simulation. UFEMISM uses the PETSc library 300 (Balay et al., 2021) for this. Most of the other operations that require data exchange between processes (e.g. remapping,

calculating gradients, etc.) are represented by matrix operations, which are also handled by PETSc. In cases where the user requires a process to access data from another process, UFEMISM offers a set of standardised routines that interface with the OpenMPI library (Gabriel et al., 2004) to facilitate this.

305 Preliminary benchmark experiments showed that v2.0 does not yet scale well when using more than 32 processors, as shown in Fig. 2. This likely has to do with the way data communication between processes is handled by PETSc, which could be improved by paying more attention to the way the model domain is partitioned over the processes, and the way PETSc decides which data should be communicated. These improvements are reserved for future work. However, it should be noted that v2.0 in its current form is capable of performing multi-millennial simulations of the Antarctic ice sheet, using a grounding-line

310 resolution of < 5 km across selected basin-scale regions, on a dual-core, consumer-grade laptop (Bernales et al., *in prep.*). Large-scale practical applications of the model are therefore already feasible even without these future improvements.

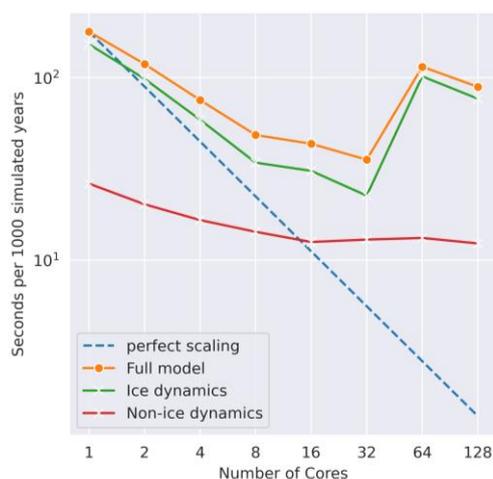



Figure 2: Strong scaling for UFEMISM v2.0 with the 10,000-yr initialisation phase of the MISMIP experiment (see also Sect. 4.2). The domain consists of approximately 13,000 triangles. The full model is the sum of the ice dynamics and non-ice dynamics
315 **components. With more than 32 cores a slowdown instead of a speedup is visible**

3.2 I/O

All input and output files of v2.0 follow the NetCDF-4 standard (Unidata, 2023). UFEMISM's NetCDF input routines automatically interact with the routines for remapping data between square grids (typical of ice-sheet-specific data, e.g.
320 BedMachine; Morlighem et al., 2019), lon/lat-grids (typical of global climate model output), and triangular meshes (e.g. output from other UFEMISM simulations). The user can provide input data in any of those formats, and UFEMISM will automatically detect the type of grid from the dimensions of the NetCDF file, choose the appropriate remapping function, and remap the data to the model mesh. The sparse matrices representing the remapping operators (see Appendix C) are stored in memory, so that

if more data is read from the same input file later on, the matrix is reused instead of needing to calculate it again. All of this is done automatically, requiring no user intervention.

UFEMISM produces output on both the model mesh, and on a square grid (with a used-defined resolution). The former is useful for detailed post-processing or visualisation, while the latter can be conveniently used for cursory inspection of model output using any NetCDF viewing software. The user can specify in the model configuration files which data fields should be included in the output files; the full list of the 100+ fields that the user can choose from, can be found in the NetCDF-output module. Adding a new field requires about 10 lines of new code.

3.3 Version control

UFEMISM is maintained on Github (https://github.com/IMAU-paleo/UFEMISM2.0). Github Actions have been set up to automatically perform all the unit tests that have been built in for the routines interfacing with OpenMPI and PETSc, the NetCDF i/o routines, mesh generation, remapping, and PDE discretisation. This enables the user to quickly diagnose any problems occurring in the model. A number of benchmark experiments have been set up similarly, which are automatically run when Git branches are merged. Figures for these experiments, following the style of the publications where these benchmark experiments were first presented (e.g. Pattyn et al., 2008 for the ISMIP-HOM experiments) are created automatically. The UFEMISM Github repository also features integration with the nix package manager (https://nixos.org/). This should allow the user to install all the required libraries (OpenMPI, PETSc, NetCDF) with their transient dependencies, using the exact version numbers for each of them, with a single command.

4 Idealised-geometry experiments

4.1 ISMIP-HOM

The Ice-Sheet Model Intercomparison Project for Higher-Order Models (ISMIP-HOM; Pattyn et al., 2008) contains several experiments to benchmark the velocities produced by the momentum balance in an idealised geometry. These experiments describe a slab of ice on a flat, sloping bed. In experiments A and B, no sliding is allowed, and periodic undulations are superimposed on the bed slope, either in both the along-slope and cross-slope directions (experiment A), or in only the along-slope direction (experiment B). In experiments B and C, the bedrock remains a flat slope, but sliding is now allowed, with the basal friction coefficient varying periodically in both the along-slope and cross-slope directions (experiment C), or in only the along-slope direction (experiment D). Six different versions of each experiment exist, differing in the wavelength of the bedrock undulations or the friction variations, with values ranging between 160 km and 5 km. Decreasing the wavelength increases the aspect ratio of the ice geometry, making the more simplified momentum balance approximations such as the SIA and SSA less accurate. The experimental setup is described in full by Pattyn et al. (2008).

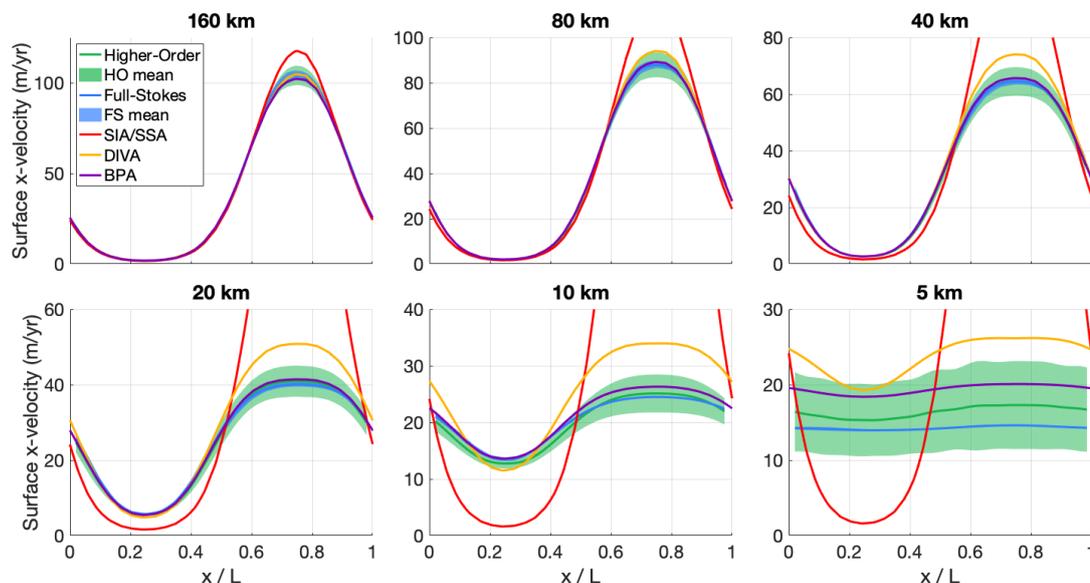
The velocities calculated by UFEMISM v2.0 for ISMIP-HOM experiments A and C using the hybrid SIA/SSA, the DIVA, and the BPA are compared to the ISMIP-HOM model ensemble by Pattyn et al. (2008) in Figs. 3 and 4.

355

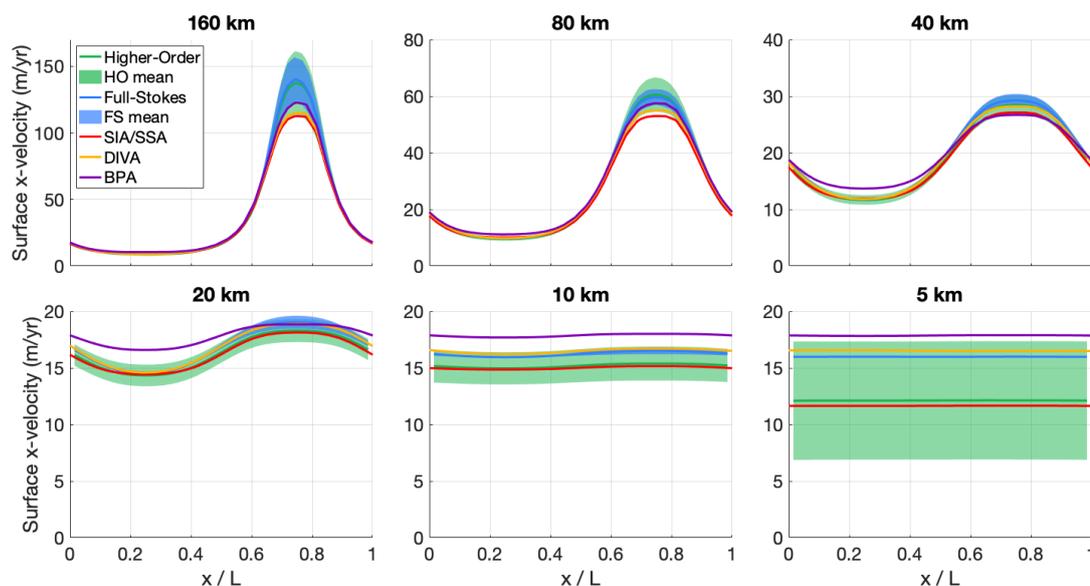
Figure 3: Ice surface velocities calculated by UFEMISM with the hybrid SIA/SSA (red), the DIVA (yellow), and the BPA (purple) in the six different versions of ISMIP-HOM experiment A (periodic bedrock undulations in both directions), compared to the model ensemble by Pattyn et al. (2008), which is divided into the higher-order model ensemble (green), and the full-Stokes model ensemble (blue).

360

Figure 4: Ice surface velocities calculated by UFEMISM with the hybrid SIA/SSA (red), the DIVA (yellow), and the BPA (purple) in the six different versions of ISMIP-HOM experiment C (flat sloping bed, periodic variations in friction in both directions), compared to the model ensemble by Pattyn et al. (2008), which is divided into the higher-order model ensemble (green), and the full-Stokes model ensemble (blue).

365 In experiment C (Fig. 4), which concerns sliding over a bed with spatially varying roughness, all three approximations result

in velocities that agree well with the ensemble. In experiment A (Fig. 3), which concerns viscous flow over an undulating bed,

the hybrid SIA/SSA starts to diverge from the ensemble with the increasing aspect ratio of the geometry at spatial scales of about 80 km. The DIVA remains accurate to spatial scales of about 20 km, whereas UFEMISM's solution to the BPA lies within the ensemble for all spatial scales.

4.2 MISMIP

To demonstrate the effectiveness of our sub-grid basal friction scaling scheme at resolving grounding-line migration, we performed an experiment along the lines of the Marine Ice-Sheet Intercomparison Project (MISMIP; Pattyn et al., 2012). The experiment describes a circular, cone-shaped island, subjected to a spatially uniform positive mass balance. This results in the formation of a circular, dome-shaped ice sheet, which flows radially outward, feeding into an ice shelf that extends outward to infinity. The experimental protocol consists of step-wise decreasing/increasing the flow parameter A in Glen's flow law, resulting in an advance/retreat of the grounding line. After being spun up to a steady state, a single advance-retreat cycle should, physically, result in the same grounding-line position as before. Any remaining difference in position is called grounding-line hysteresis, which the original MISMIP study showed could be significant (up to several hundred kilometres) in models that do not pay special attention to the way the discontinuous friction at the grounding line is handled (Pattyn et al., 2012).

We performed simulations with grounding-line resolutions of 10, 8, 5, and 4 km. We start with a 10,000-yr spin-up phase, with a uniform flow factor of $A = 10^{-16}\text{Pa}^{-3}\text{yr}^{-1}$. We then decrease the flow factor to $A = 10^{-17}\text{Pa}^{-3}\text{yr}^{-1}$ for a period of 10,000 years, resulting in an advance of the grounding line by about 200 km. Finally, we revert the flow factor back to its original value, causing the grounding line to retreat again. The results of this experiment are shown in Fig. 5; panel A shows transects of the ice sheet at the end of each of the three phases (spin-up, advance, retreat) for the 10 km simulation, while panel B shows the position of the grounding line over time for all three resolutions. The difference in grounding-line position between the end of the spin-up phase at 10 kyr, and the end of the retreat phase at 30 kyr is smaller than twice the grounding-line resolution in all simulations. Note that all these simulations were performed with the dynamic adaptive mesh; whereas in v1.0, a mesh update would result in a small but noticeable "jump" in the grounding-line position (Berends et al., 2021, their Fig. 10b), improvements to the remapping scheme in v2.0 have greatly reduced this problem.

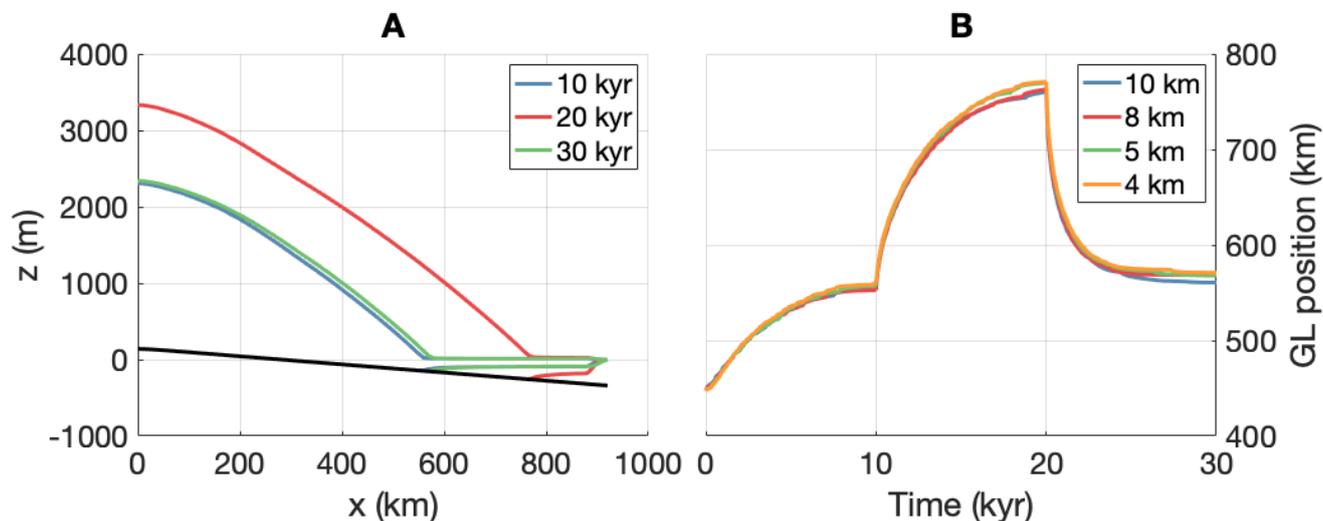



Figure 5: A) Transects of the ice sheet at the end of each of the three phases (spin-up [blue], advance [red], retreat [green]) for the 10 km simulation. B) Grounding-line position over time, using grounding-line resolutions of 10 km (blue), 8 km (red), 5 km (green), and 4 km (orange).

4.3 MISMIP+

395

The third Marine Ice-Sheet Model Intercomparison Project (MISMIP+; Asay-Davis et al., 2016) investigates the retreat of an ice stream feeding into a buttressed shelf. In the steady state, the ice stream flows down an 80-km wide, ~500 km long fjord. The grounding line rests on a retrograde slope, which is kept stable by the strongly buttressed ice shelf. In the experiment, the ice sheet starts from a steady state, and is subjected to a strong sub-shelf melt forcing. The resulting loss of buttressing causes

400 the grounding line to retreat by about 50 km over the course of the 100-yr simulation. The experimental set-up is described by Asay-Davis et al. (2016), while the results of the intercomparison are presented by Cornford et al. (2020). The resulting grounding-line retreat was found to vary by about a factor 3 between different models. A large part of this spread was attributed to (small) differences in initial conditions, as well as the choice of sliding law (the experimental protocol allows one to choose between three different sliding laws).

405 We have performed MISMIP+ experiment "ice1r" (100 years of increased-melt forcing) with UFEMISM v2.0, using the Schoof sliding law (Eq. 20), at grounding-line resolutions ranging from 5 km to 500 m. Glen's flow law parameter A has been tuned separately for each simulation to achieve a stable mid-stream grounding-line position at x = 450 km. The results of these simulations are compared to the model ensemble results by Cornford et al. (2020) in Fig. 6. The UFEMISM results lie well within the Cornford et al. (2020) ensemble range. Note that these simulations were all performed with the dynamic adaptive

410 mesh. In the 500 m simulation, the mesh was updated about once every model year on average, at no significant computational expense (as the computation time is dominated by solving the momentum balance).

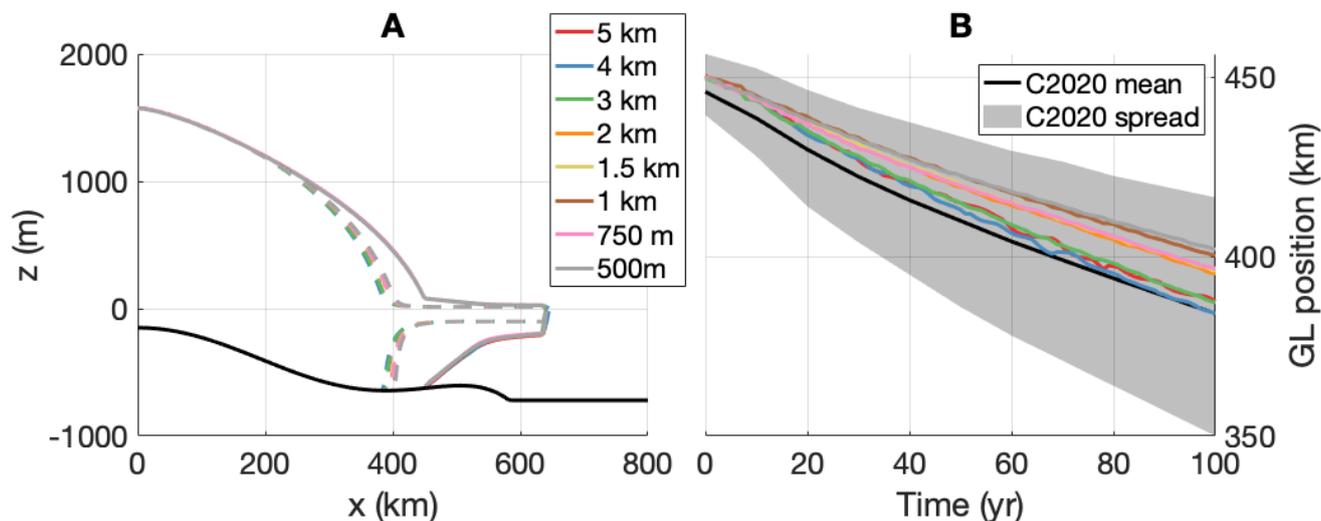



Figure 6: A) Mid-stream transects of the ice sheet at the beginning (solid lines) and the end (dashed lines) of the 100-yr retreat simulation at different resolutions (see legend). B) Mid-stream grounding-line position over time at different resolutions (see legend), compared to the Cornford et al. (2020) model ensemble (mean shown by solid black line, spread shown by grey shaded area).

5 Discussion and conclusions

We have presented version 2.0 of UFEMISM and verified its performance in a number of benchmark experiments with idealised geometries. We have shown that the model is able to solve different approximations to the Stokes equations, and to integrate the resulting thinning rates through time to model the evolving ice geometry on a dynamic adaptive mesh. The results lie within the published model ensembles for all these experiments. These verified model capabilities provide the groundwork for the realistic applications presented in part 2 of this work (Bernales et al., *in prep.*).

The numerical stability and computational performance of the model have been greatly improved. This includes the new time-stepping scheme, as well as the switch from a simple successive over-relaxation scheme to PETSc for solving the matrix equations. As a result, v2.0 is much faster than v1.0, capable of either running the same simulations in a fraction of the required computation time, or running a simulation in the same amount of time as before, but at a much higher resolution.

The ISMIP-HOM experiments presented here, as well as the work by Rückamp et al. (2022), demonstrate the importance of considering the model's approximation to the Stokes equations when moving to high resolutions. At the high resolutions that UFEMISM can now achieve, topographical features can be resolved that would invalidate the underlying assumptions of the DIVA. However, solving the BPA can easily require 50 times more computation time than solving the DIVA, which would be unfeasible for many practical applications. Improving the model's performance when using large numbers of processors, as mentioned before, could be a way to solve this problem. Another approach could be to reduce the size of the physical

problem by moving to regional ice-sheet modelling, limiting the model domain to a single drainage basin. In preparation for

435   such an approach, the code of UFEMISM's routines for solving the ice thickness equation has been written in such a way as to easily allow the user to define regions where the ice thickness should not change.

The current version of the model does not yet scale well, which is a major remaining point of improvement. While the model is already capable of performing high-resolution (< 5 km), multi-millennial simulations of the Antarctic ice sheet (Bernales et

440   al., *in prep.*), moving to even higher resolutions would currently still require the user to wait for several days for the simulation to complete. Improving this part of the model's performance, probably by revising the way PETSc is instructed to handle data communication between the processes, should be the focus of future work.

Appendix A: Mesh generation

UFEMISM uses an extended version of Ruppert's algorithm (Ruppert, 1995) to iteratively refine a simple initial mesh until it

445   meets the requirements of the ice-sheet geometry. In Ruppert's original algorithm, the mesh is inspected to find "bad" triangles, which are triangles whose smallest internal angle lies below a certain threshold value (typically 25°). These triangles are then "split", meaning that a new vertex is added at that triangle's circumcentre, and the Delaunay triangulation is updated to include the new vertex. In UFEMISM, Ruppert's algorithm is extended to additionally mark as "bad" those triangles whose longest leg exceeds the maximum resolution for the area of the domain where that triangle lies. For example, if the grounding line

450   passes through a triangle whose longest leg exceeds the user-defined maximum grounding-line resolution, that triangle is marked as "bad", even if it meets Ruppert's original smallest-angle criterion.

While the general functionality of the mesh generation code has not fundamentally changed from v1.0, the way meshes are refined is quite different now. In v2.0, the mesh generation code is provided with data fields of bedrock elevation and ice thickness, which can be defined either on a square grid or on a mesh. This geometry is then "reduced" to obtain a list of [x,y]

455   points that together span the grounding line (and similarly for the calving front, etc.). This is illustrated in Fig. A1.

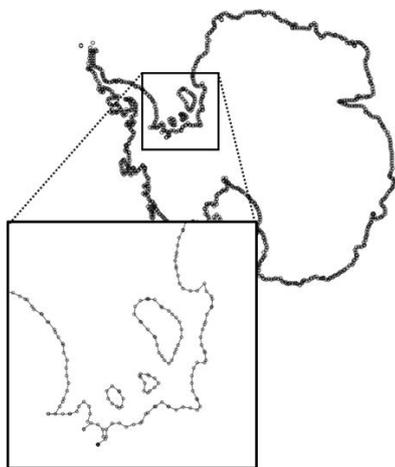



Figure A1: The grounding line of the Antarctic ice sheet can be represented by a series of short line segments.

This line is provided as input to the mesh generation code, which simply checks which triangles cross with any section of the line, and splits them if necessary. An advantage of this approach is that the code paths for generating a mesh based on an ice-

460 sheet geometry that is provided on a square grid (e.g. BedMachine; Morlighem et al., 2019), and for a geometry provided on a mesh (e.g. during a mesh update in a UFEMISM simulation) are identical from the point where these geometries are reduced to lines.

In addition to the line-based mesh refinement code, v2.0 also contains point-based and polygon-based refinement routines. The point-based routine can be used to obtain a high-resolution at a certain location of interest, for example an ice-core site.

465 The polygon-based routine can be used to increase the mesh resolution over a certain ice-sheet section, e.g. the Pine Island Glacier drainage basin. This is illustrated in Fig. A2.

Input mesh	Refinement	Output mesh

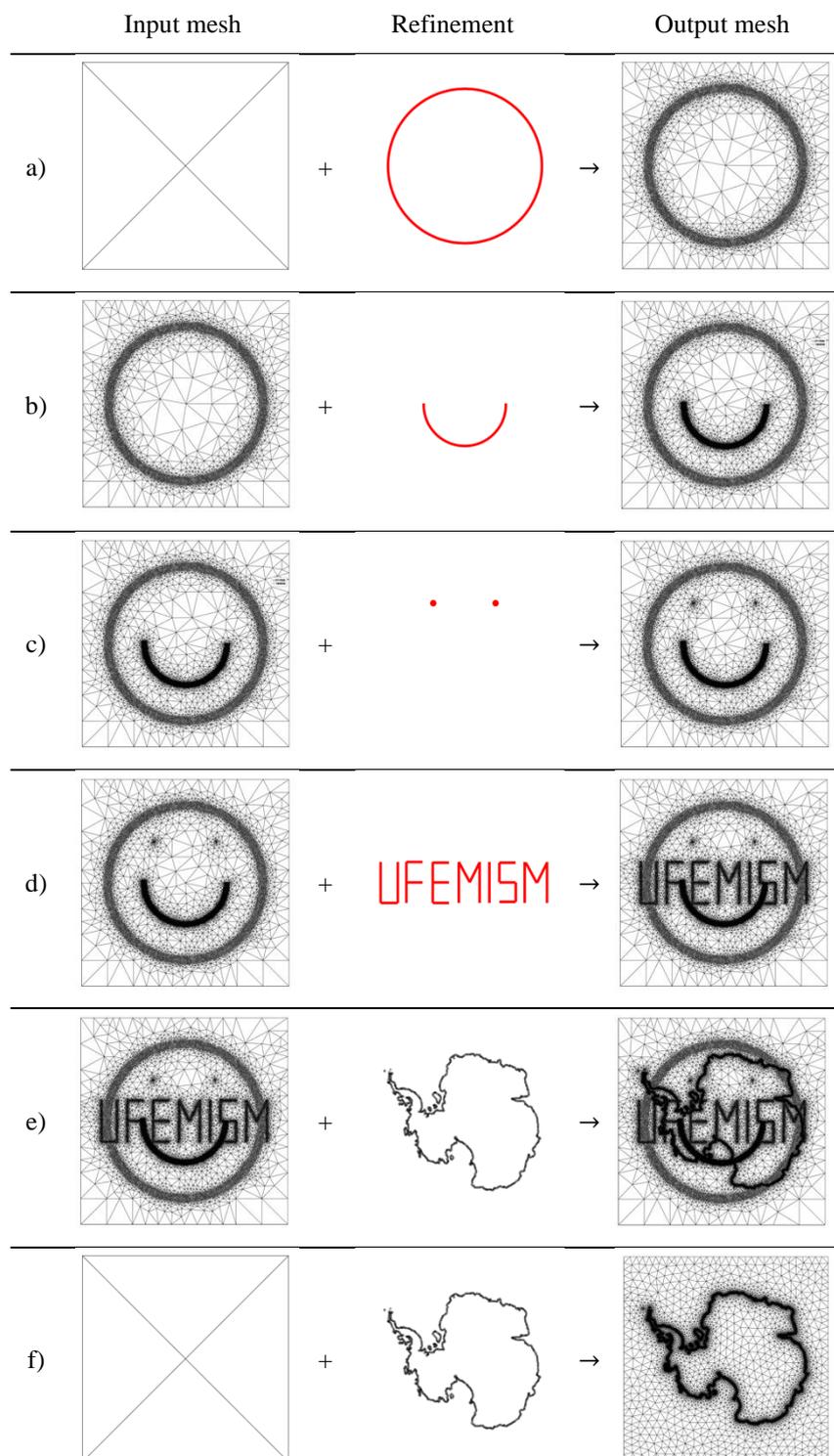
a) + →

b) + →

c) + →

d) + →

e) + →

f) + →

470 **Figure A2: Each row shows how the mesh refinement algorithm refines an existing mesh (first column) with a refinement forcing (second column) to produce a new mesh (third column). a) Starting with the 5-vertex, 4-triangle "dummy" mesh, the line-refinement algorithm is provided with a series of short line segments spanning a simple circle. b) The mesh is further refined over a series of short line segments spanning a half-circle. c) The mesh is further refined over two points. d) The mesh is further refined over a series of line segments spanning the letters of UFEMISM. e) The mesh is further refined over a series of line segments spanning the Antarctic grounding line. f) Starting with the dummy mesh and only refining over the Antarctic grounding line yields a mesh that**
475 **would be more suitable for the ice-sheet model.**

Through the config file, the user can set separate maximum resolutions for the entire domain, for grounded ice, floating ice, for (a band of specified width around) the grounding line, the calving front, the grounded ice margin, and the coastline.

Appendix B: Discretisation

The discretisation scheme used in v1.0, described in Berends et al. (2021), which was based on neighbour functions, has been
480 replaced by a least squares-based scheme based on Syrakos et al. (2017). The advantage of this new scheme is that it is easily extended to work on different Arakawa grids and to higher orders of accuracy, and that it can be coded much more elegantly.

Let $f : \mathbb{R}^2 \to \mathbb{R}$ be a function defined on the model domain, and let f_a, f_b, f_c be its discretised approximations on respectively the mesh vertices (equivalent to the Arakawa-A grid), triangles (B-grid), and edges (C-grid). For convenience, the discretised
485 approximations to the partial derivatives of f on the different grids are written as $f_{x,a} = \left(\frac{\partial f}{\partial x}\right)_a, f_{yy,c} = \left(\frac{\partial^2 f}{\partial x^2}\right)_c$, etc. These partial derivatives can be expressed as linear combinations of f_a, f_b, f_c, e.g.:

$$f_{x,a} = M_{x,a,a} f_a. \tag{B1}$$

Here, $M_{x,a,a}$ is an nV-by-nV matrix. In the notation convention used here, M has three subscript indices. The first indicates the operation represented by M: x for $\frac{\partial}{\partial x}$, yy for $\frac{\partial^2}{\partial y^2}$, etc., and m for mapping f between the different Arakawa grids. The second
490 and third indices represent the source and destination Arakawa grids, respectively. E.g., $M_{m,a,b}$ maps a data field from the vertices to the triangles.

B1: First-order, regular grid

Syrakos et al. (2017) describe a (weighted) least-squares scheme for discretising the gradient operator on an unstructured grid. Let $f_a^i, f_{x,a}^i, f_{y,a}^i$ be the values of the function f and its first partial derivatives on vertex i. The value f_a^j of f on vertex j, which
495 neighbours vertex i, can then be expressed as a Taylor expansion of f around i:

$$f_a^j = f_a^i + \Delta x_j f_{x,a}^i + \Delta y_j f_{y,a}^i + \mathcal{O}\left(\Delta x_j^2, \Delta y_j^2\right). \tag{B2}$$

Here, $\Delta x_j, \Delta y_j$ is the displacement between vertices j and i. If i has n neighbours, this results in the following system of n linear equations (defining $\Delta f_a^j \equiv f_a^j - f_a^i$, dropping the truncation error $\mathcal{O}(\Delta x_j^2, \Delta y_j^2)$, and introducing the vertex weights w_j for the weighted least-squares approximation):

500

$$\underbrace{\begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} \Delta f_a^1 \\ \Delta f_a^2 \\ \vdots \\ \Delta f_a^n \end{bmatrix}}_{b} = \underbrace{\begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} \Delta x_1 & \Delta y_1 \\ \Delta x_2 & \Delta y_2 \\ \vdots & \vdots \\ \Delta x_n & \Delta y_n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} f_{x,a}^i \\ f_{y,a}^i \end{bmatrix}}_{z}. \tag{B3}$$

Using matrix notation, this equation reads $Wb = WAz$, which can be solved for z:

$$z = (A^T W^T W A)^{-1} A^T W^T W b = Q\beta_b. \tag{B4}$$

Here, we have grouped the A and W terms into $Q = (A^T W^T W A)^{-1}$ and $\beta_b = A^T W^T W b$. The symmetric 2-by-2 matrix $A^T W^T W A$, which needs to be inverted to find Q, is expressed as:

505

$$A^T W^T W A = \sum_{c=1}^n w_c^2 \begin{bmatrix} \Delta x_c^2 & \Delta x_c \Delta y_c \\ & \Delta y_c^2 \end{bmatrix}. \tag{B5}$$

Here, c loops over all vertices that are connected to i. The second term, β_b, is expressed as:

$$\beta_b = \sum_{c=1}^n w_c^2 \begin{bmatrix} \Delta x_c \Delta f_a^c \\ \Delta y_c \Delta f_a^c \end{bmatrix}. \tag{B6}$$

Once Q has been calculated by inverting $A^T W^T W A$, the first partial derivative $f_{x,a}^i$ of f on i can be expressed as:

$$f_{x,a}^i = Q(1,1) \sum_{c=1}^n (w_c^2 \Delta x_c \Delta f_a^c) + Q(1,2) \sum_{c=1}^n (w_c^2 \Delta y_c \Delta f_a^c). \tag{B7}$$

510 Since we defined $\Delta f_a^j \equiv f_a^j - f_a^i$, this can be rewritten to read:

$$f_{x,a}^i = -f_a^i \sum_{c=1}^n [w_c^2 (Q(1,1)\Delta x_c + Q(1,2)\Delta y_c)] + \sum_{c=1}^n f_a^c [w_c^2 (Q(1,1)\Delta x_c + Q(1,2)\Delta y_c)]. \tag{B8}$$

This means that the coefficients of the operator matrix $M_{x,a,a}$ are given by:

$$M_{x,a,a}^{i,j} = \begin{cases} -\sum_{c=1}^n [w_c^2 (Q(1,1)\Delta x_c + Q(1,2)\Delta y_c)] & \text{if } i = j, \\ w_c^2 (Q(1,1)\Delta x_c + Q(1,2)\Delta y_c) & \text{if } j \text{ is connected to } i, \\ 0 & \text{otherwise.} \end{cases} \tag{B9}$$

Similarly, the coefficients for $M_{y,a,a}$ are given by:

515

$$M_{y,a,a}^{i,j} = \begin{cases} -\sum_{c=1}^n [w_c^2 (Q(2,1)\Delta x_c + Q(2,2)\Delta y_c)] & \text{if } i = j, \\ w_c^2 (Q(2,1)\Delta x_c + Q(2,2)\Delta y_c) & \text{if } j \text{ is connected to } i, \\ 0 & \text{otherwise.} \end{cases} \tag{B10}$$

The weights w_j depend on the distance between j and i:

$$w_j = \frac{1}{|r_j - r_i|^q}. \tag{B11}$$

Following Syrakos et al. (2017), we choose $q = \frac{3}{2}$.

B2: First-order, staggered grid

520 The derivation in section B1 holds for the case where both the function f and its gradients f_x, f_y are defined on the same grid, so that f_i is known. However, if for example we want to calculate the first partial derivative of f on the mesh triangles $f_{x,b}$ when f itself is defined on the mesh vertices (f_a), then this condition does not hold, and a slightly different derivation is needed.

Consider the Taylor series described by Eq. B2. We once again write out the system of linear equations for f on the collection

525 of neighbouring points, but this time we do not introduce Δf, so that we obtain the following expression:

$$
\underbrace{\begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} f_a^1 \\ f_a^2 \\ \vdots \\ f_a^n \end{bmatrix}}_{b} = \underbrace{\begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} 1 & \Delta x_1 & \Delta y_1 \\ 1 & \Delta x_2 & \Delta y_2 \\ \vdots & \vdots & \vdots \\ 1 & \Delta x_n & \Delta y_n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} f_b^i \\ f_{x,b}^i \\ f_{y,b}^i \end{bmatrix}}_{z}. \tag{B12}
$$

Following the same derivation as before, the symmetric 3-by-3 matrix $A^T W^T W A$ that needs to be inverted to find Q is now given by:

$$
A^T W^T W A = \sum_{c=1}^{n} w_c^2 \begin{bmatrix} 1 & \Delta x_c & \Delta y_c \\ & \Delta x_c^2 & \Delta x_c \Delta y_c \\ & & \Delta y_c^2 \end{bmatrix}. \tag{B13}
$$

530 Similarly, β_b is now given by:

$$
\beta_b = \sum_{c=1}^{n} w_c^2 \begin{bmatrix} f_a^c \\ \Delta x_c \Delta f_a^c \\ \Delta y_c \Delta f_a^c \end{bmatrix}. \tag{B14}
$$

This leads to the following expression for the coefficients of the matrices $M_{m,a,b}, M_{x,a,b}, M_{y,a,b}$:

$$
M_{m,a,b}^{i,j} = \begin{cases} -\sum_{c=1}^{n}[w_c^2(Q(1,1) + Q(1,2)\Delta x_c + Q(1,3)\Delta y_c)] & \text{if } i = j, \\ w_c^2(Q(1,1) + Q(1,2)\Delta x_c + Q(1,3)\Delta y_c) & \text{if } j \text{ is connected to } i, \\ 0 & \text{otherwise,} \end{cases} \tag{B15}
$$

$$
M_{x,a,b}^{i,j} = \begin{cases} -\sum_{c=1}^{n}[w_c^2(Q(2,1) + Q(2,2)\Delta x_c + Q(2,3)\Delta y_c)] & \text{if } i = j, \\ w_c^2(Q(2,1) + Q(2,2)\Delta x_c + Q(2,3)\Delta y_c) & \text{if } j \text{ is connected to } i, \\ 0 & \text{otherwise,} \end{cases} \tag{B16}
$$

535
$$
M_{y,a,b}^{i,j} = \begin{cases} -\sum_{c=1}^{n}[w_c^2(Q(3,1) + Q(3,2)\Delta x_c + Q(3,3)\Delta y_c)] & \text{if } i = j, \\ w_c^2(Q(3,1) + Q(3,2)\Delta x_c + Q(3,3)\Delta y_c) & \text{if } j \text{ is connected to } i, \\ 0 & \text{otherwise.} \end{cases} \tag{B17}
$$

B3: Second-order, regular grid

Here, we extend the discretisation scheme by Syrakos et al. (2017) to include the second-order partial derivatives f_{xx}, f_{xy}, f_{yy}.

First, we extend the Taylor expansion of f around i to include the second-order terms:

$$
f_a^j = f_a^i + \Delta x_j f_{x,a}^i + \Delta y_j f_{y,a}^i + \tfrac{1}{2}\Delta x_j^2 f_{xx,a}^i + \Delta x_j \Delta y_j f_{xy,a}^i + \tfrac{1}{2}\Delta y_j^2 f_{yy,a}^i + \mathcal{O}(\Delta x_j^3, \Delta y_j^3). \tag{B18}
$$

Writing out the system of linear equations for all neighbours of i now yields the following expression:

$$\underbrace{\begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} f_a^1 \\ f_a^2 \\ \vdots \\ f_a^n \end{bmatrix}}_{b} = \underbrace{\begin{bmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} \Delta x_1 & \Delta y_1 & \frac{1}{2}\Delta x_1^2 & \Delta x_1 \Delta y_1 & \frac{1}{2}\Delta y_1^2 \\ \Delta x_2 & \Delta y_2 & \frac{1}{2}\Delta x_2^2 & \Delta x_2 \Delta y_2 & \frac{1}{2}\Delta y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \Delta x_n & \Delta y_n & \frac{1}{2}\Delta x_n^2 & \Delta x_n \Delta y_n & \frac{1}{2}\Delta y_n^2 \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} f_{x,a}^i \\ f_{y,a}^i \\ f_{xx,a}^i \\ f_{xy,a}^i \\ f_{yy,a}^i \end{bmatrix}}_{z}. \tag{B19}$$

The symmetric 5-by-5 matrix $A^T W^T W A$ that needs to be inverted to find Q is now given by:

$$A^T W^T W A = \sum_{c=1}^n w_c^2 \begin{bmatrix} \Delta x_c^2 & \Delta x_c \Delta y_c & \frac{1}{2}\Delta x_c^3 & \Delta x_c^2 \Delta y_c & \frac{1}{2}\Delta x_c \Delta y_c^2 \\ & \Delta y_c^2 & \frac{1}{2}\Delta x_c^2 \Delta y_c & \Delta x_c \Delta y_c^2 & \frac{1}{2}\Delta y_c^3 \\ & & \frac{1}{4}\Delta x_c^4 & \frac{1}{2}\Delta x_c^3 \Delta y_c & \frac{1}{4}\Delta x_c^2 \Delta y_c^2 \\ & & & \Delta x_c^2 \Delta y_c^2 & \frac{1}{2}\Delta x_c \Delta y_c^3 \\ & & & & \frac{1}{4}\Delta y_c^4 \end{bmatrix}. \tag{B20}$$

Similarly, β_b is now given by:

$$\beta_b = \sum_{c=1}^n w_c^2 \begin{bmatrix} \Delta x_c \Delta f_a^c \\ \Delta y_c \Delta f_a^c \\ \frac{1}{2}\Delta x_c^2 \Delta f_a^c \\ \Delta x_c \Delta y_c \Delta f_a^c \\ \frac{1}{2}\Delta y_c^2 \Delta f_a^c \end{bmatrix}. \tag{B21}$$

Expressions for the coefficients of $M_{x,a,a}, M_{y,a,a}, M_{xx,a,a}, M_{xy,a,a}, M_{yy,a,a}$ (which are now fourth-order accurate operators) can be derived similar as before.

Appendix C: Remapping

Because of the dynamic adaptive grid, data fields must often be remapped between square grids and (different) irregular triangular meshes. Extensive preliminary experiments have shown that only second-order conservative remapping results in accurate model results (e.g., ice thickness over time that matches the analytical solution in the Halfar dome experiment). Less accurate remapping schemes (nearest-neighbour, bilinear, biquadratic, binning, Gaussian interpolation) all result in much more diffusion during each remapping operation, and additionally violate conservation of mass and energy when remapping ice thickness and temperature, as these schemes are generally not conservative.

The mathematical theory behind conservative remapping is described by Jones (1999), and is relatively straightforward. However, Jones (1999) derived the equations in spherical coordinates, whereas UFEMISM uses Cartesian coordinates. Furthermore, UFEMISM uses a slightly different scheme, which conserves both local and global integrated values (the definition of "conservative" used by Jones), as well as extreme values (an important property, as we do not want to end up with negative ice thickness after remapping). We will therefore provide a full derivation here.

C1: Theory

560

Let there exist two meshes that both cover the same domain Ω: a source mesh (indicated from here by the subscript s) and a destination mesh (subscript d). Suppose the source mesh is the one that existed before a mesh update, and the destination mesh is the newly generated mesh. Let f_{s^a} be a discrete function defined on the vertices of the source mesh. The remapping problem then consists of finding a new discrete function f_{d^a}, defined on the vertices of the destination mesh, such that:

565

$$\iint_\Omega f_{d^a} dA = \iint_\Omega f_{s^a} dA, \tag{C1}$$

$$\iint_{A_d^i} f_{d^a} dA = \iint_{A_d^i} f_{s^a} dA, \text{ where } A_d^i \text{ are the Voronoi cells of the vertices of the destination mesh,} \tag{C2}$$

$$\min(f_{s^a}) \le f_{d^a} \le \max(f_{s^a}), \tag{C3}$$

Here, Eq. C1 implies conservation of the global integrated value, Eq. C2 implies conservation of local integrated values, and Eq. C3 implies conservation of extreme values.

570

Let $f(x, y)$ be a piecewise bilinear function, which is obtained from the discrete source function on the source mesh triangles f_{s^b} by bilinearly interpolating inside the triangles:

$$f(x, y) = f_{s^b} + \left(x - x_{s^b}\right)\left(\frac{\partial f}{\partial x}\right)_{s^b} + \left(y - y_{s^b}\right)\left(\frac{\partial f}{\partial y}\right)_{s^b}. \tag{C4}$$

Here, x_{s^b}, y_{s^b} are the coordinates of the geometric centre of source mesh triangle s^b. Note that f_{s^b} can be obtained from f_{s^a}

575 using the operator matrices derived in Appendix B:

$$f_{s^b} = M_{m,s^a,s^b} f_{s^a}, \tag{C5}$$

$$\left(\frac{\partial f}{\partial x}\right)_{s^b} = M_{x,s^a,s^b} f_{s^a}, \tag{C6}$$

$$\left(\frac{\partial f}{\partial y}\right)_{s^b} = M_{y,s^a,s^b} f_{s^a}. \tag{C7}$$

The discrete function f_{d^a} on the vertices of the destination mesh is found by simply averaging $f(x, y)$ over the Voronoi cells

580 A_{d^a} of the vertices of the destination mesh:

$$f_{d^a} = \frac{1}{A_{d^a}} \iint_{A_{d^a}} f(x, y) dA. \tag{C8}$$

Note that, as Eq. C8 implies that $\min(f(x, y)) \le f_{d^a} \le \max(f(x, y))$, and Eq. C4 implies that $\min(f_{s^a}) \le f(x, y) \le \max(f_{s^a})$, this implies that $\min(f_{s^a}) \le f_{d^a} \le \max(f_{s^a})$, thus satisfying the conservation of extreme values required by Eq. C3. Substituting Eq. C4 into Eq. C8 yields:

585

$$f_{d_a} = \frac{1}{A_{d^a}} \sum_{s^b} \left[\iint_{A_{s^b d^a}} \left(f_{s^b} + \left(x - x_{s^b}\right)\left(\frac{\partial f}{\partial x}\right)_{s^b} + \left(y - y_{s^b}\right)\left(\frac{\partial f}{\partial y}\right)_{s^b} \right) dA \right]. \tag{C9}$$

Here, $A_{s^b d^a}$ indicates the area of overlap between the source mesh triangles s^b and the destination mesh Voronoi cells d^a. Eq. C9 can be rearranged to read:

$$f_{d_a} = \frac{1}{A_{d^a}} \sum_{s^b} \left[f_{s^b} \iint_{A_{s^b d^a}} dA + \left(\frac{\partial f}{\partial x}\right)_{s^b} \iint_{A_{s^b d^a}} (x - x_{s^b}) dA + \left(\frac{\partial f}{\partial y}\right)_{s^b} \iint_{A_{s^b d^a}} (y - y_{s^b}) dA \right],$$

$$= \frac{1}{A_{d^a}} \sum_{s^b} \left[\left(f_{s^b} - x_{s^b} \left(\frac{\partial f}{\partial x}\right)_{s^b} - y_{s^b} \left(\frac{\partial f}{\partial y}\right)_{s^b} \right) \iint_{A_{s^b d^a}} dA + \left(\frac{\partial f}{\partial x}\right)_{s^b} \iint_{A_{s^b d^a}} x dA + \left(\frac{\partial f}{\partial y}\right)_{s^b} \iint_{A_{s^b d^a}} y dA \right]. \quad (C10)$$

590 Since the area of overlap $A_{s^b d^a}$ between a triangle of the source mesh and a Voronoi cell of the destination mesh will generally be an irregularly-shaped polygon, Eq. C10 is generally not easy to evaluate. However, the problem can be simplified by applying the divergence theorem, rewriting the three surface integrals in Eq. C10 into line integrals:

$$\iint_A dA = \oint_{\partial A} x dy, \quad (C11)$$

$$\iint_A x dA = -\oint_{\partial A} x y dx, \quad (C12)$$

595
$$\iint_A y dA = \oint_{\partial A} x y dy. \quad (C13)$$

Note that, as the perimeters of both the source mesh triangles and the destination mesh Voronoi cells are piecewise linear curves, the perimeter of the area of overlap $A_{s^b d^a}$ must therefore also be a piecewise linear curve. The expressions for the three line integrals along a straight line from $\boldsymbol{p} = [x_p, y_p]$ to $\boldsymbol{q} = [x_q, y_q]$ are given by:

$$\int_{\boldsymbol{p}}^{\boldsymbol{q}} x dy = x_p \Delta y - y_p \Delta x + \frac{\Delta x}{2 \Delta y} (y_q^2 - y_p^2), \quad (C14)$$

600
$$\int_{\boldsymbol{p}}^{\boldsymbol{q}} -x y dx = \frac{1}{2} \left(x_p \frac{\Delta y}{\Delta x} - y_p \right) (x_q^2 - x_p^2) - \frac{1}{3} \frac{\Delta y}{\Delta x} (x_q^3 - x_p^3), \quad (C15)$$

$$\int_{\boldsymbol{p}}^{\boldsymbol{q}} x y dy = \frac{1}{2} \left(x_p - y_p \frac{\Delta x}{\Delta y} \right) (y_q^2 - y_p^2) + \frac{1}{3} \frac{\Delta x}{\Delta y} (y_q^3 - y_p^3). \quad (C16)$$

Here, $\Delta x = x_q - x_p, \Delta y = y_q - q_p$. Substituting Eqs. C11 – 13 into Eq. C10 yields:

$$f_{d^a} = \frac{1}{A_{d^a}} \sum_{s^b} \left[\left(f_{s^b} - x_{s^b} \left(\frac{\partial f}{\partial x}\right)_{s^b} - y_{s^b} \left(\frac{\partial f}{\partial y}\right)_{s^b} \right) \oint_{\partial A_{s^b d^a}} x dy - \left(\frac{\partial f}{\partial x}\right)_{s^b} \oint_{\partial A_{s^b d^a}} x y dx + \left(\frac{\partial f}{\partial y}\right)_{s^b} \oint_{\partial A_{s^b d^a}} x y dy \right]. (C17)$$

This implies that, in order to find the remapped value of f on a destination vertex, we need to find all the source triangles
605 overlapping with that vertex' Voronoi cell, and calculate the three line integrals around the perimeter of the area of overlap between that source triangle and the destination Voronoi cell.

As can be seen from Eq. C17, the remapped function f_{d^a} is a linear combination of the triangle source function values f_{s^b} and its gradients $\left(\frac{\partial f}{\partial x}\right)_{s^b}, \left(\frac{\partial f}{\partial y}\right)_{s^b}$, which are in turn linear combinations of the vertex source function values f_{s^a}. We can therefore rewrite Eq. C17 as a matrix equation. First, we define the three matrices B_{xdy}, B_{-xydx}, and B_{xydy}, which contain the
610 line integrals around the areas of overlap between the source triangles s^b and the destination Voronoi cells d^a:

$$B_{xdy}^{ij} = \oint_{\partial A_{s^{bj} d^{ai}}} x dy, \quad (C18)$$

$$B_{-xydx}^{ij} = -\oint_{\partial A_{s^{bj} d^{ai}}} x y dx, \quad (C19)$$

$$B_{xydy}^{ij} = \oint_{\partial A_{s^{bj} d^{ai}}} x y dy. \quad (C20)$$

Note that B_{xdy}^{ij}, B_{-xydx}^{ij}, and B_{xydy}^{ij} are non-zero if and only if source triangle j and destination Voronoi cell i overlap.
615 These three matrices can be combined to yield the three remapping weights matrices W_0, $W_{1,x}$, and $W_{1,y}$:

$$W_0^{ij} = \frac{B_{xdy}^{ij}}{A_{d^{ai}}}, \tag{C21}$$

$$W_{1,x}^{ij} = \frac{B_{-xydx}^{ij}}{A_{d^{ai}}} - W_0^{ij} x_{s^{bj}}, \tag{C22}$$

$$W_{1,y}^{ij} = \frac{B_{xydx}^{ij}}{A_{d^{ai}}} - W_0^{ij} y_{s^{bj}}. \tag{C23}$$

Substituting Eqs. C21 – 23 into Eq. C17 yields:

$$f_{d^a} = W_0 f_{s^b} + W_{1,x} \left(\frac{\partial f}{\partial x}\right)_{s^b} + W_{1,y} \left(\frac{\partial f}{\partial y}\right)_{s^b}. \tag{C24}$$

Substituting Eqs. C5 – 7 into Eq. C24 yields:

$$f_{d^a} = \left(W_0 M_{m,s^a,s^b} + W_{1,x} M_{x,s^a,s^b} + W_{1,y} M_{y,s^a,s^b}\right) f_{s^a} = M_{s^a,d^a} f_{s^a}. \tag{C25}$$

Here, $M_{s^a,d^a} = W_0 M_{m,s^a,s^b} + W_{1,x} M_{x,s^a,s^b} + W_{1,y} M_{y,s^a,s^b}$ is an nV_d-by-nV_s matrix that represents the second-order conservative remapping operation from the source mesh vertices to the destination mesh vertices.

C2: Implementation

In order to calculate the remapping matrix M_{s^a,d^a}, the three line integrals in Eqs. C11 – 13 need to be calculated around the areas of overlap between all source mesh triangles and destination mesh Voronoi cells. While the line integrals themselves are simple enough (Eqs. C14 – 16), determining which sources triangles overlap with which destination Voronoi cells is not straightforward. Given the large numbers of vertices and triangles involved in high-resolution meshes (easily several tens of thousands of both), it is necessary to pay attention to computational efficiency.

The perimeter $\partial A_{s^{bj}d^{ai}}$ of the area of overlap $A_{s^{bj}d^{ai}}$ between source triangle s^{bj} and destination Voronoi cell d^{ai} consists of part of the perimeter $\partial A_{s^{bj}}$ of source triangle s^{bj}, and part of the perimeter $\partial A_{d^{ai}}$ of destination Voronoi cell d^{ai}. This means that, in order calculate the coefficients of the three matrices in Eqs. C18 – 20, it suffices to integrate once around every source triangle and around every destination Voronoi cell, carefully keeping track of the triangle or Voronoi cell of the opposite mesh with which it overlaps.

In UFEMISM, this is done using a collection of "line tracing" subroutines. Given a line $[\boldsymbol{p}, \boldsymbol{q}]$, the model "traces" that line through a mesh, and returns a list of all the Voronoi cells or triangles through which that line passes, and the line integrals for all the individual line segments lying within them. Great care is taken to detect cases where the perimeters of source triangles and destination Voronoi cells coincide, to prevent double-counting. By actively "tracing" the line, finding the index of the next triangle or Voronoi cell it crosses into from the connectivity lists of the triangle or cell it departs, instead of performing a mesh-wide search operation every time, computational expense is greatly reduced. Thus, calculating the remapping matrix only takes a fraction of the computation time required to create a new mesh.

Appendix D: Terrain-following coordinate transformation

In order to solve the BPA, the heat equation, and conservation of mass, the vertical dimension must be discretised as well. In
645 UFEMISM, this is done by introducing a terrain-following coordinate transformation:

$$\hat{x}(x, y, z, t) = x, \tag{D1a}$$

$$\hat{y}(x, y, z, t) = y, \tag{D1b}$$

$$\zeta(x, y, z, t) = \frac{s(x,y,t)-z}{H(x,y,t)}, \tag{D1c}$$

$$\hat{t}(x, y, z, t) = t. \tag{D1d}$$

650 Eq. D1c implies that $\zeta = 0$ at the ice surface, and $\zeta = 1$ at the ice base. Note that, in order to transform the heat equation, the
time dimension is transformed as well. Applying this coordinate transformation results in the following expressions for the
gradient operators:

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial \hat{x}} + \frac{\partial \zeta}{\partial x} \frac{\partial}{\partial \zeta}, \tag{D2a}$$

$$\frac{\partial}{\partial y} = \frac{\partial}{\partial \hat{y}} + \frac{\partial \zeta}{\partial y} \frac{\partial}{\partial \zeta}, \tag{D2b}$$

655
$$\frac{\partial}{\partial z} = \frac{\partial \zeta}{\partial z} \frac{\partial}{\partial \zeta}, \tag{D2c}$$

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \hat{t}} + \frac{\partial \zeta}{\partial t} \frac{\partial}{\partial \zeta}. \tag{D2d}$$

Applying the chain rule to Eq. D1c yields the following expressions for the gradients of ζ:

$$\frac{\partial \zeta}{\partial x} = \frac{1}{H}\left(\frac{\partial s}{\partial x} - \zeta \frac{\partial H}{\partial x}\right), \tag{D3a}$$

$$\frac{\partial \zeta}{\partial y} = \frac{1}{H}\left(\frac{\partial s}{\partial y} - \zeta \frac{\partial H}{\partial y}\right), \tag{D3b}$$

660
$$\frac{\partial \zeta}{\partial z} = \frac{-1}{H}. \tag{D3c}$$

$$\frac{\partial \zeta}{\partial t} = \frac{1}{H}\left(\frac{\partial s}{\partial t} - \zeta \frac{\partial H}{\partial t}\right), \tag{D3b}$$

The gradient operators in Eqs. D2a – d can be represented by matrices as derived in Appendix B, by multiplying their
untransformed equivalents with the gradients of ζ, e.g.:

$$M_{x,a,b} = M_{\hat{x},a,b} + D\left(\frac{\partial \zeta}{\partial x}\right) M_{\zeta,a,b}. \tag{D4}$$

665 Here, $D(f)$ represents a diagonal matrix with the elements of the vector f on the diagonal, i.e. $D^{ij} = \begin{cases} f^i & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$ By
thus calculating the matrices for all the gradient operators, the stiffness matrix representing the momentum balance can be
assembled.

The scaled vertical coordinate ζ is discretised using an irregular, log-linear grid:

670
$$\zeta^k = 1 - \frac{R^{\left(\frac{n-k}{n-1}\right)} - 1}{R-1}, k \in [1, n]. \tag{D5}$$

This implies that the ratio between the grid spacings at the ice surface and ice base is approximately equal to R, which is a configurable number with a default value of $R = 10$. This scheme results in improved accuracy of the solution near the ice base, where the strain rates (in the BPA) and the temperature gradients (in the heat equation) are highest, without requiring additional vertical grid points. The number of vertical layers is configurable, and is by default set to 12.

Appendix E: vertical ice velocities

Applying the terrain-following coordinate transformation from Appendix D to the expression for conservation of mass in Eq. 14 yields:

$$\frac{\partial u}{\partial \hat{x}} + \frac{\partial \zeta}{\partial x}\frac{\partial u}{\partial \zeta} + \frac{\partial v}{\partial \hat{y}} + \frac{\partial \zeta}{\partial y}\frac{\partial v}{\partial \zeta} + \frac{\partial \zeta}{\partial z}\frac{\partial w}{\partial \zeta} = 0. \tag{E1}$$

The terms $\frac{\partial u}{\partial \hat{x}} + \frac{\partial v}{\partial \hat{y}}$ describe the divergence in the two-dimensional plane, in scaled coordinates:

$$\frac{\partial u}{\partial \hat{x}} + \frac{\partial v}{\partial \hat{y}} = \widehat{\nabla}\boldsymbol{u}. \tag{E2}$$

Averaging this divergence over the Voronoi cell of a mesh vertex yields:

$$\overline{\widehat{\nabla}\boldsymbol{u}} = \frac{1}{A}\iint_A \left(\widehat{\nabla}\boldsymbol{u}\right)dA. \tag{E3}$$

By applying the divergence theorem, this integral can be transformed to a loop integral around the boundary of the Voronoi cell:

$$\overline{\widehat{\nabla}\boldsymbol{u}} = \frac{1}{A}\oint_{\partial A}(\boldsymbol{u} \cdot \widehat{\boldsymbol{n}})dS. \tag{E4}$$

Here, $\widehat{\boldsymbol{n}}$ is the outward normal vector to the Voronoi cell boundary. Substituting this expression into Eq. 15 yields:

$$\frac{\partial w}{\partial \zeta} = \frac{-1}{\partial \zeta/\partial z}\left[\frac{1}{A}\oint_{\partial A}(\boldsymbol{u} \cdot \widehat{\boldsymbol{n}})dS + \frac{\partial \zeta}{\partial x}\frac{\partial u}{\partial \zeta} + \frac{\partial \zeta}{\partial y}\frac{\partial v}{\partial \zeta}\right]. \tag{E5}$$

This expression can then be integrated over the transformed vertical dimension to find w:

$$w(\zeta) = w(\zeta = 1) - \int_1^\zeta \frac{\partial w}{\partial \zeta}d\hat{\zeta}. \tag{E6}$$

Note that the minus sign in Eq. E6 arises from the fact that ζ runs from 0 at the ice surface, to 1 at the ice base, meaning that integrating upwards from the ice base means integrating in the negative ζ direction. The vertical velocity at the base is given by:

$$w(\zeta - 1) = w_b = u_b\left(\frac{\partial s}{\partial x} - \frac{\partial H}{\partial x}\right) + v_b\left(\frac{\partial s}{\partial y} - \frac{\partial H}{\partial y}\right) + \frac{\partial s}{\partial t} - \frac{\partial H}{\partial t}. \tag{E7}$$

Appendix F: calculating the ice flux divergence operator

Conservation of ice mass for a shallow layer of ice in the 2-D plane is expressed mathematically as:

$$\frac{\partial H}{\partial t} = -\nabla \cdot (\boldsymbol{u}H) + m. \tag{F1}$$

Here, m is the net mass balance, including terms at the ice base, the ice surface, and the lateral boundaries. This equation is discretised spatially using the finite volume scheme that lent UFEMISM its name. Averaging Eq. F1 over the Voronoi cell of vertex i (the control volume of the finite volume scheme) yields:

700

$$\frac{\partial H^i}{\partial t} = \frac{-1}{A^i} \iint_{A^i} \nabla \cdot (\boldsymbol{u}H) dA + m^i. \tag{F2}$$

Using the divergence theorem, the double integral in Eq. F2 can be transformed:

$$\frac{\partial H^i}{\partial t} = \frac{-1}{A^i} \oint_{\partial A^i} (\boldsymbol{u}H) \cdot \hat{\boldsymbol{n}} dS + m^i. \tag{F3}$$

Here, $\hat{\boldsymbol{n}}$ is the outward unit normal to the boundary ∂A^i of the Voronoi cell of vertex i. Let $(\boldsymbol{u}H)^{ij}$ be average ice flux on the shared Voronoi cell boundary of vertices i and j. Then the loop integral Eq. F3 can be transformed to a sum:

705

$$\frac{\partial H^i}{\partial t} = \frac{-1}{A^i} \sum_{j=1}^{n} [(\boldsymbol{u}H)^{ij} \cdot \hat{\boldsymbol{n}}^{ij} L^{ij}] + m^i. \tag{F4}$$

Here, $\hat{\boldsymbol{n}}^{ij}$ is the unit normal vector pointing from vertex i to vertex j, L^{ij} is the length of their shared Voronoi cell boundary, and $\sum_{j=1}^{n}$ sums over only those vertices j that are connected to i. We then introduce an upwind scheme for the ice flux $\boldsymbol{u}H$:

$$(\boldsymbol{u}H)^{ij} = \begin{cases} \boldsymbol{u}^{ij}H^i & \text{if } \boldsymbol{u}^{ij} \cdot \hat{\boldsymbol{n}}^{ij} > 0, \\ \boldsymbol{u}^{ij}H^j & \text{otherwise.} \end{cases} \tag{F5}$$

This implies that, if the ice flows from vertex i to vertex j, the ice thickness in vertex i determines the flux, and vice versa.

710 This scheme offers better numerical stability than using the average ice thickness of i and j regardless of the flow direction.

Eqs. F4 and F5 imply that $\frac{\partial H^i}{\partial t}$ is a linear combination of the ice thicknesses H^i. Eq. F4 can therefore be represented by a matrix equation:

$$\frac{\partial H^i}{\partial t} = -M_{\text{divQ}} H^i + m^i. \tag{F6}$$

Here, M_{divQ} is a matrix whose coefficients depend on the mesh geometry and the ice velocities, which can be multiplied with

715 the ice thickness vector H^i to find the ice flux divergence $\nabla \cdot (\boldsymbol{u}H)$. The coefficients of M_{divQ} are given by:

$$M_{\text{divQ}}^{ij} = \begin{cases} \frac{1}{A^i} \sum_{j=1}^{n} [L^{ij} \max(\boldsymbol{u}^{ij} \cdot \hat{\boldsymbol{n}}^{ij}, 0)] & \text{if } i = j, \\ \frac{L^{ij}}{A^i} \min(\boldsymbol{u}^{ij} \cdot \hat{\boldsymbol{n}}^{ij}, 0) & \text{if } i \text{ is connected to } j, \\ 0 & \text{otherwise.} \end{cases} \tag{F7}$$

Acknowledgements. We will thank any reviewers who are kind enough to provide constructive criticism of the manuscript.

720 *Author contributions.* CJB wrote the new model code except for the distributed-memory code, which was developed by VA. CJB performed the experiments. VA and JB set up the new version control system. CJB wrote the draft of the manuscript; all authors contributed to the final version.

Code and data availability. The source code of UFEMISM v2.0, scripts for compiling and running the model on a variety of
725 computer systems, and the configuration files for all simulations presented here, are freely available on Github:
https://github.com/IMAU-paleo/UFEMISM2.0. The exact version of the code that was used to produce the results presented
here is archived at zenodo.org (Berends et al., 2023c), though aspiring users are advised to check out the latest version from
Github.

730 *Competing interests*. The authors declare that they have no competing interests.

Financial support. CJB was supported by PROTECT. This project has received funding from the European Union's Horizon
2020 research and innovation programme under grant agreement no. 869304 (PROTECT; [PROTECT article number will be
assigned upon acceptance for publication!]). JB and VA were supported by NWO, grant no. OCENW.KLEIN.515.

735 **References**

Asay-Davis, X. S., Cornford, S. L., Durand, G., Galton-Fenzi, B. K., Gladstone, R. M., Gudmunsson, 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), Geoscientific Model Development 9, 2471-2497,
740 2016.

Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V.,
Gropp, W. D., Kaushik, D., Knepley, M. G., May, D. A., Curfman McInnes, L., Tran Mills, R., Munson, T., Rupp, K., Sanan,
P., Smith, B. F., Zampini, S., and Zhang, H.: PETSc Users Manual, ANL-95/11 – Revision 3.15, Argonne National Laboratory,
2021.

745 Berends, C. J., Goelzer, H., and van de Wal, R. S. W.: The Utrecht Finite Volume Ice-Sheet Model: UFEMISM
(version 1.0), Geoscientific Model Development 14, 2443-2470, 2021.

Berends, C. J., Goelzer, H., Reerink, T. J., Stap, L. B., and van de Wal, R. S. W.: Benchmarking the vertically
integrated ice-sheet model IMAU-ICE (version 2.0), Geoscientific Model Development 15, 5667-5688, 2022.

Berends, C. J., van de Wal, R. S. W., van den Akker, T., and Lipscomb, W. H.: Compensating errors in inversions
750 for subglacial bed roughness: same steady state, different dynamic response, The Cryosphere 17, 1585-1600, 2023a.

Berends, C. J., Stap, L. B., and van de Wal, R. S. W.: Strong impact of sub-shelf melt parameterisation on ice-sheet
retreat in idealised and realistic Antarctic topography, Journal of Glaciology 69, 1434-1448, doi: 10.1017/jog.2023.33, 2023b.

Berends, C. J., Bernales, J. A., Azizi, V., and van de Wal, R. S. W.: UFEMISM2.0, doi: 10.5281/zenodo.8434469,
2023c.

755   Bernales, J.A., Berends, C.J., and van de Wal., R. S. W.: The Utrecht Finite Volume Ice-Sheet Model (UFEMISM version 2.0) – part 2: application to the Greenland and Antarctic ice sheets, *in prep.*

Bueler, E. and Brown, J.: Shallow shelf approximation as a "sliding law" in a thermomechanically coupled ice sheet model, Journal of Geophysical Research 114, F03008, https://doi.org/10.1029/2008JF001179, 2009.

Bueler, E. and van Pelt, W.: Mass-conserving subglacial hydrology in the Parallel Ice Sheet Model version 0.6,
760   Geoscientific Model Development 8, 1613-1635, 2015.

Burgard, C., Jourdain, N. C., Reese, R., Jenkins, A., and Mathiot, P.: An assessment of basal melt parameterisations for Antarctic ice shelves, The Cryosphere 16, 4931-4975, 2022.

Cornford, S. L., Martin, D. F., Graves, D. T., Ranken, D. F., 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, Journal of Computational Physics 232,
765   529-549, 2013.

Cornford, S. L., Seroussi, H., Asay-Davis, X. S., Gudmunsson, G. H., Arthern, R., Borstad, C., Christmann, J., Dias dos Santos, T., Feldmann, J., Goldberg, D. N., Hoffman, M. J., Humbert, A., Kleiner, T., Leguy, G. R., Lipscomb, W. H., Merino, N., Durand, G., Morlighem, M., Pollard, D., Rückamp, M., Williams, C. R., and Yu, H.: Results of the third Marine Ice Sheet Model Intercomparison Project (MISMIP+), The Cryosphere 14, 2283-2301, 2020.

770   Feldmann, J., Albrecht, T., Khroulev, C., Pattyn, F., and Levermann, A.: Resolution-dependent performance of grounding line motion in a shallow model compared with a full-Stokes model according to the MISMIP3d intercomparison, Journal of Glaciology 60, 353-360, 2014.

Fox-Kemper, B., Hewitt, H. T., Xiao, C., Aðalgeirsdóttir, G., Drijfhout, S. S., Edwards, T. L., Golledge, N. R., Hemer, M., Kopp, R. E., Krinner, G., Mix, A., Notz, D., Nowicki, S., Nurhati, I. S., Ruiz, L., Sallée, J.-B., Slangen, A. B. A., and Yu,
775   Y.: Ocean, Cryosphere and Sea Level Change, in: Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change, edited by: Masson-Delmotte, V., Zhai, P., Pi- rani, A., Connors, S. L., Péan, C., Berger, S., Caud, N., Chen, Y., Goldfarb, L., Gomis, M. I., Huang, M., Leitzell, K., Lonnoy, E., Matthews, J. B. R., Maycock, T. K., Waterfield, T., Yelekçi, O., Yu, R., and Zhou, B., Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 1211–1362, https://doi.org/10.1017/9781009157896.011, 2021.

780   Gabriel, E., Fagg, G. E., Bosilca, G., Angskun, T., Dongarra, J. J., Squyres, J. M., Sahay, V., Kambadur, P., Barrett, B., Lumsdaine, A., Castain, R. H., Daniel, D. J., Graham, R., L., and Woodall, T. S.: Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation, Proceedings of the 11th European PVM/MPI Users' Group Meeting, 2004.

Gladstone, R. M., Payne, A. J., and Cornford, S. L.: Resolution requirements for grounding-line modelling: sensitivity to basal drag and ice-shelf buttressing, Annals of Glaciology 53, 97-105, 2012.

785   Glen, J. W.: The creep of polycrystalline ice, Proceedings of the Royal Society A 228, 519-538, 1955.

Goldberg, D. N.: A variationally derived, depth-integrated approximation to a higher-order glaciological flow model, Journal of Glaciology 57, 157-170, 2011.

Halfar, P.: On the Dynamics of Ice Sheets, Journal of Geophysical Research 86, 11065-11072, 1981.

Hoffman, M. J., Perego, M., Price, S. F., Lipscomb, W. H., Zhang, T., Jacobsen, D., Tezaur, I., Salinger, A. G.,

790 Tuminaro, R., and Bertagna, L.: MPAS-Albany Land Ice (MALI): a variable-resolution ice sheet model for Earth system modeling using Voronoi grids, Geoscientific Model Development 11, 3747-3780, 2018.

Iverson, N. R., Hoover, T. S., and Baker, R. W.: Ring-shear studies of till deformation: Coulomb-plastic behavior and distributed strain in glacier beds, Journal of Glaciology 44, 634-642, 1998.

Jones, P. W.: First- and Second-Order Conservative Remapping Schemes for Grids in Spherical Coordinates, Monthly

795 Weather Review 127, 2204-2210, 1999.

Kazmierczak, E., Sun, S., Coulon, V., and Pattyn, F.: Subglacial hydrology modulates basal sliding response of the Antarctic ice sheet to climate forcing, The Cryosphere 16, 4537-4552, 2022.

Leguy, G. R., Lipscomb, W. H., and Asay-Davis, X. S.: Marine ice sheet experiments with the Community Ice Sheet Model, The Cryosphere 15, 3229-3253, 2021.

800 Lipscomb, W. H., Price, S. F., Hoffman, M. J., Leguy, G. R., Bennett, A. R., Bradley, S. L., Evans, K. J., Fyke, J. G., Kennedy, J. H., Perego, M., Ranken, D. M., Sacks, W. J., Salinger, A. G., Vargo, L. J., and Worley, P. H.: Description and evaluation of the Community Ice Sheet Model (CISM) v2.1, Geoscientific Model Development 12, 387-424, 2019.

Morland, L. W.: Unconfined ice-shelf flow, in: Dynamics of the West Antarctic Ice Sheet, edited by: van der Veen, C. J. and Oerlemans, J., 99–116, Kluwer Acad., Dordrecht, the Netherlands, 1987.

805 Morland, L. W. and Johnson, I. R.: Steady motion of ice sheets, Journal of Glaciology 25, 229-246, 1980.

Morlighem, M., Rignot, E., Binder, T., Blankenship, D. D., Drews, R., Eagles, G., Eisen, O., Ferraccioli, F., Forsberg, R., Fretwell, P., Goel, V., Greenbaum, J. S., Gudmundsson, H., Guo, J., Helm, V., Hofstede, C., Howat, I., Humbert, A., Jokat, W., Karlsson, N. B., Lee, W. S., Matsuoka, K., Millan, R., Mouginot, J., Paden, J., Pattyn, F., Roberts, J., Rosier, S., Ruppel, A., Seroussi, H., Smith, E. C., Steinhage, D., Sun, B., van den Broeke, M. R., Van Ommen, T. D., van Wessem, M., and

810 Young, D. A.: Deep glacial troughs and stabilizing ridges unveiled beneath the margins of the Antarctic ice sheet, Nature Geoscience 13, 132-138, 2019.

Oppenheimer, M., Glavovic, B. C., Hinkel, J., van de Wal, R., Magnan, A. K., Abd-Elgawad, A., Cai, R., Cifuentes-Jara, M., DeConto, R. M., Ghosh, T., Hay, J., Isla, F., Marzeion, B., Meyssignac, B., and Sebesvari, Z.: Sea Level Rise and Im- plications for Low-Lying Islands, Coasts and Communities, in: IPCC Special Report on the Ocean and Cryosphere in a

815 Changing Climate, edited by: Pörtner, H.-O., Roberts, D. C., Masson-Delmotte, V., Zhai, P., Tignor, M., Poloczanska, E., Mintenbeck, K., Alegría, A., Nicolai, M., Okem, A., Petzold, J., Rama, B., and Weyer, N. M., Cambridge University Press, Cambridge, UK and New York, NY, USA, 321–445, https://doi.org/10.1017/9781009157964.006, 2019.

Pattyn, F.: A new three-dimensional higher-order thermomechanical ice sheet model: Basic sensitivity, ice stream development, and ice flow across subglacial lakes, Journal of Geophysical Research 108, doi:10.1029/2002JB002329, 2003.

820 Pattyn, F.: Sea-level response to melting of Antarctic ice shelves on multi-centennial timescales with the fast Elementary Thermomechanical Ice Sheet model (f.ETISh v1.0), The Cryosphere 11, 1851-1878, 2017.

Pattyn, F., Perichon, L., Aschwanden, A., Breuer, B., de Smedt, B., Gagliardini, O., Gudmunsson, G. H., Hindmarsh, R. C. A., Hubbard, A., Johnson, J. V., Kleiner, T., Konovalov, Y., Martin, C., Payne, A. J., Pollard, D., Price, S. F., 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, 2008.

Pattyn, F., Schoof, C., Perichon, L., Hindmarsh, R. C. A., Bueler, E., de Fleurian, B., Durand, G., Gagliardini, O., Gladstone, R. M., Goldberg, D., Gudmunsson, 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, 2012.

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

Quiquet, A., Dumas, C., Ritz, C., Peyaud, V., and Roche, D. M.: The GRISLI ice sheet model (version 2.0): calibration and validation for multi-millennial changes of the Antarctic ice sheet, Geoscientific Model Development 11, 5003-5025, 2018.

Reese, R., Winkelmann, R., and Gudmundsson, G. H.: Grounding-line flux formula applied as a flux condition in numerical simulations fails for buttressed Antarctic ice streams, The Cryosphere 12, 3229-3242, 2018.

Robinson, A., Alvarez-Solas, J., Montoya, M., Goelzer, H., Greve, R., and Ritz, C.: Description and validation of the ice-sheet model Yelmo (version 1.0), Geoscientific Model Development 13, 2805-2823, 2020.

Robinson, A., Goldberg, D., and Lipscomb, W. H.: A comparison of the stability and performance of depth-integrated ice-dynamics solvers, The Cryosphere 16, 689-709, 2022.

Rückamp, M., Kleiner, T., and Humbert, A.: Comparison of ice dynamics using full-Stokes and Blatter-Pattyn approximation: application to the Northeast Greenland Ice Stream, The Cryosphere 16, 1675-1696, 2022.

Ruppert, J.: A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation, Journal of Algorithms 18, 548-585, 1995.

Schoof, C.: The effect of cavitation on glacier sliding, Proceedings of the Royal Society A 461, 609-627, 2005.

Schoof, C.: Ice sheet grounding line dynamics: Steady states, stability, and hysteresis, J. Geophys. Res., 112, F03S28, https://doi.org/10.1029/2006JF000664, 2007.

Sun, S., Pattyn, F., Simon, E. G., Albrecht, T., Cornford, S. L., Calov, R., Dumas, C., Gillet-Chaulet, F., Goelzer, H., Golledge, N. R., Greve, R., Hoffman, M. J., Humbert, A., Kazmierczak, E., Kleiner, T., Leguy, G. R., Lipscomb, W. H., Martin, D., Morlighem, M., Nowicki, S., Pollard, D., Price, S. F., Quiquet, A., Seroussi, H., Schlemm, T., Sutter, J., van de Wal, R. S. W., Winkelmann, R., and Zhang, T.: Antarctic ice sheet response to sudden and sustained ice-shelf collapse (ABUMIP), Journal of Glaciology 66, 891-904, 2020.

Syrakos, A., Varchanis, S., Dimakopoulos, Y., Goulas, A., and Tsamopoulos, J.: A critical analysis of some popular methods for the discretisation of the gradient operator in finite volume methods, Physics of Fluids 29, 127103, doi:10.1063/1.4997682, 2017.

855   Tsai, V. C., Stewart, A. L., and Thompson, A. F.: Marine ice-sheet profiles and stability under Coulomb basal conditions, Journal of Glaciology 61, 205-215, 2015.

Unidata: NetCDF 4 [software]. Boulder, CO: UCAR/Unidata Program Center, doi: 10.5065/D6H70CW6, 2023

van de Wal, R. S. W., Zhang, X., Minobe, S., Jevrejeva, S., Riva, R. E. M., Little, C., Richter, K., and Palmer, M. D.: Uncertainties in Long-Term Twenty-First Century Process-Based Coastal Sea-Level Projections, Surveys in Geophysics 40,
860   1655-1671, 2019.

van de Wal, R. S. W., Nicholls, R. J., Behar, D., McInnes, K., Stammer, D., Lowe, J. A., Church, J. A., DeConto, R. M., Fettweis, X., Goelzer, H., Haasnoot, M., Haigh, I. D., Hinkel, J., Horton, B. P., James, T. S., Jenkins, A., LeCozannet, G., Levermann, A., Lipscomb, W. H., Marzeion, B., Pattyn, F., Payne, A. J., Pfeffer, W. T., Price, S. F., Seroussi, H., Sun, S., Veatch, W., and White, K.: A High-End Estimate of Sea Level RIse for Practitioners, Earth's Future 10,
865   https://doi.org/10.1029/2022EF002751, 2022.

Weertman, J.: On the sliding of glaciers, Journal of Glaciology 3, 33-38, 1957.