Adaptive mesh, finite volume modeling of marine ice sheets

Stephen L. Cornford[a], Daniel F. Martin[b], Daniel T. Graves[b], Douglas F. Ranken[c], Anne M. Le Brocq[d], Rupert M. Gladstone[a], Antony J. Payne[a], Esmond G. Ng[b], William H. Lipscomb[c]

[a]*School of Geographical Sciences, University of Bristol, UK*
[b]*Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, Berkeley, California*
[c]*Los Alamos National Laboratory, New Mexico*
[d]*Geography, College of Life and Environmental Sciences, University of Exeter, UK*

Abstract

Continental scale marine ice sheets such as the present day West Antarctic Ice Sheet are strongly affected by highly localized features, presenting a challenge to numerical models. Perhaps the best known phenomenon of this kind is the migration of the grounding line — the division between ice in contact with bedrock and floating ice shelves — which needs to be treated at sub-kilometer resolution. We implement a block-structured finite volume method with adaptive mesh refinement (AMR) for three dimensional ice sheets, which allows us to discretize a narrow region around the grounding line at high resolution and the remainder of the ice sheet at low resolution. We demonstrate AMR simulations that are in agreement with uniform mesh simulations, but are computationally far cheaper, appropriately and efficiently evolving the mesh as the grounding line moves over significant distances. As an example application, we model rapid deglaciation of Pine Island Glacier in West Antarctica caused by melting beneath its ice shelf.

1. Introduction

The West Antarctic Ice Sheet (WAIS) is the world's major present-day continent-sized marine ice sheet [1]. In general, a marine ice sheet consists of an expanse of grounded ice in contact with a bed that is largely below sea level, joined at some (or all) of its edge to floating ice shelves across a contour known as the grounding line. In their turn, the ice shelves end at calving fronts [2], which then form part or all of the ice sheet's horizontal boundary. Ice flowing across the grounding line into the ice shelves is ultimately lost to the ocean, either through melting, departing across the calving front as icebergs, or even through large-scale disintegration of an ice shelf [3]. In 1979, Mercer proposed that the ice shelves surrounding WAIS may be vulnerable to additional melting due to anthropogenic global warming, which may in turn make the ice sheet unstable [4], a proposal that is still considered credible [5].

Theoretical and numerical studies have shown that marine ice sheets can be unstable, and so might undergo rapid change, if they lie on beds that slope upward toward the sea. The ice shelves which surround WAIS are all fed by narrow, fast flowing ice streams, whose velocity is dominated by sliding at the base. Schoof proved that for an idealized ice stream flowing in the $xz-$plane that does not vary in y the flux of ice across the grounding line increases with its thickness there and hence the bedrock depth [6, 7]. Thus, the grounding line cannot come to stable equilibrium anywhere that the bedrock slopes up toward the sea, something also seen in numerical studies which permit both sliding and vertical shear [8]. Now, the flow of marine ice sheets which vary in y is buttressed, that is, stabilized by additional stresses arising from the motion of ice shelf within a bay [9] or across obstructions [10]. Provided that the regions of up-sloping bed are wide enough at the grounding line, removal (or reduction) of the ice shelves can lead to an unstable marine ice sheet [11].

Email addresses: s.l.cornford@bristol.ac.uk (Stephen L. Cornford), dfmartin@lbl.gov (Daniel F. Martin)

Rapid retreat of the grounding line in the Amundsen sea sector of the West Antarctic Ice Sheet has indeed been observed since the early 1990s [1, 12–14], leading to a net loss of 105 Gt of ice from the region in 2007 [15]. In particular, the grounding line of Pine Island Glacier has retreated at a rate of more than 1 km per year between 1996 and 2007 [15] leading to considerable interest in its future [16–20]. Over longer timescales, the majority of WAIS is thought to deglaciate periodically, according to, for example, numerical modeling [9], and observations of the close genetic relationship between marine fauna in the Ross and Weddell seas [21]. Looking to the future, observations of its bowl-shaped bedrock topography suggest that the deglaciation of WAIS due to marine ice sheet instability would result in sea level rise of around 3 m [5].

Modeling marine ice sheets, at least using conventional finite difference, finite volume or finite element methods[1], requires fine spatial resolution, of the order of 100 m in the region of the grounding line [8, 10, 22, 23]. At the root of this is an assumption that resistance to sliding changes abruptly at the grounding line from a finite value to zero; lower resolution might be sufficient whenever resistance decays gradually [24]. In this paper, we take discontinuous change for granted, and describe and test a finite volume ice sheet model which employs adaptive mesh refinement (AMR) to obtain fine spatial resolution only where it is required. We take special care to demonstrate that:

- our AMR results converge with spatial resolution,
- AMR results obtained on a non-uniform mesh (or sequence of such meshes) whose finest resolution is Δx agree with results obtained on a uniform mesh of resolution Δx, and
- that we can apply the model to realistic problems.

To this end, after presenting the model's governing equations and numerical methods in sections 2 and 3, we describe a number of tests in section 4. We compare AMR and uniform mesh results that exhibit spatially second-order accuracy when there is no grounding line in 4.1, and first-order accuracy when a grounding line is present in 4.2. Finally, we consider the retreat of Pine Island Glacier's grounding line driven by extremely rapid melting of its ice shelf in 4.3.

2. Governing equations

We consider a volume of ice whose upper surface is given at any instant by $z = s(x, y)$ and whose lower surface is $z = b(x, y)$, as shown in figure 1. We neglect time evolution of the underlying bedrock, whose elevation is $r(x, y)$. Sea level, assumed to be spatially constant, is at $z = 0$. Parts of the ice sheet are grounded — the lower surface is in contact with the bedrock — and the remainder is floating in ocean water. The grounding line (or its projection onto the xy-plane, Γ_{gl}), marks the boundary between these regions. Lateral boundaries, which we assume to be constant in time, are made up by vertically extruding a closed contour Γ partitioned into ice divides Γ_d and calving fronts Γ_{cf}.

The evolution of $s(x, y)$ and $b(x, y)$ with time may be treated as a thin-film free-surface problem, subject to incompressible Stokes flow with shear-thinning, non-linear rheology and sliding at the base [25–27]. At any instant, the components of the deviatoric stress tensor, $\tau_{ij}(x, y, z)$ are related to the components of the rate-of-strain tensor, $\dot{\epsilon}_{ij}(x, y, z)$ through Glen's law,

$$\dot{\epsilon}_{ij}(x, y, z) = A \left(\tau^2\right)^{(n-1)/2} \tau_{ij}(x, y, z) \qquad : \qquad n \geq 1. \tag{1}$$

Here, the second invariant of the stress tensor is, making use of the Einstein summation convention,

$$\tau^2 = \frac{1}{2}\tau_{ij}\tau_{ji}, \tag{2}$$

[1] An unconventional method was described by Pollard and DeConto, who treated their grounding line by applying Schoof's expression to avoid the need for high resolution, and were able to run simulations spanning millions of years [9]. It is not straightforward to quantify the validity of this treatment, though it may well be a small source of error among the myriad uncertainties that face ice sheet models.

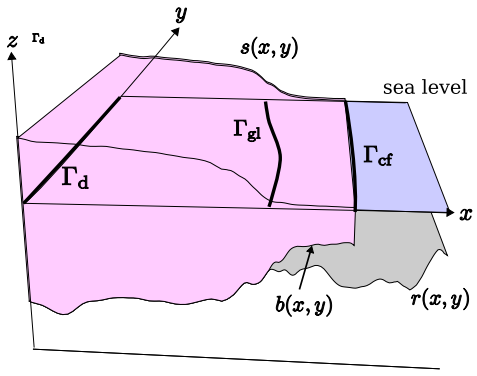


Figure 1: Schematic marine ice sheet geometry.

and the rate-of-strain is expressed in terms of the velocity field $v_i(x, y, z)$ through

$$\dot{\epsilon}_{ij} = \frac{1}{2}\left(\partial_j v_i + \partial_i v_j\right). \tag{3}$$

Commonly, and throughout this paper, the power-law exponent in Glen's law is taken to be $n = 3$. The rate factor A depends strongly on temperature, but we will consider only isothermal ice.

The upper surface of the ice is stress-free everywhere, but boundary conditions at the base of the ice vary between grounded and floating regions. Where the ice is floating, the normal stress at the lower surface of the ice matches the hydrostatic water pressure. Elsewhere, the ice is in contact with bedrock and there can be no flow of ice normal to it. Sliding generates resistance according to a power law, so that if $\vec{n}$ is a unit normal to the lower surface, and $\vec{t}$ is any unit vector tangential to it

$$n_i t_j \tau_{ij}(x, y, z = b) = -C(x, y)\left(v_i v_i\right)^{(m-1)/2} t_j v_j(x, y, z = b), \tag{4}$$

where the basal traction coefficient $C(x, y) \geq 0$ and the exponent $m \geq 0$.

A number of approaches to this free-surface problem exist in the literature, the principal distinction between them being the choice of stress-balance equation. The most general models solve the full Stokes problem across the whole domain [8, 28, 29] but, as these can be computationally expensive, approximate models which assume a small aspect ratio of vertical to horizontal length scale, ϵ, are widespread. The simplest such approximation, the shallow-ice approximation [11, 30], also assumes a large ratio of vertical to horizontal stress, λ, and is not at all applicable to floating ice. Intermediate approaches which are valid for both grounded and floating ice include Blatter's model [25, 31, 32] (also known as the Blatter-Pattyn model) and the shelfy-stream approximation [10, 17, 33]. Blatter's model reduces the Stokes problem to a three-dimensional stress-balance equation in the two horizontal velocity components v_x and v_y, incurring an error of order $O(\epsilon^2)$ to do so [27]. The shelfy-stream approximation further assumes that λ is small, so that vertical strains are neglected, an $O(\epsilon\lambda)$ error is introduced, and only a two-dimensional stress-balance equation needs to be solved. As well as these, a hybrid category of models has been reported, which also only require the solution of a two-dimensional stress-balance equation, but take some account of vertical strains [34, 35]. We employ a model devised along those lines by Schoof and Hindmarsh [27].

2.1. Schoof-Hindmarsh model

The Schoof-Hindmarsh model [27] is an approximation to the full Stokes model comprised of an elliptic problem to be solved for the $x-$ and $y-$ components of the velocity at the base of the ice, $\mathrm{v}_i(x, y) = v_i(x, y, z = b)$, and a simple formula for $v_i(x, y, z)$ that is applied once $\mathrm{v}_i(x, y)$ has been found. The resulting

$v_i(x,y,z)$ deviates from solutions to the full Stokes problem by $O(\epsilon^2)$ in moderate or fast sliding regimes and $O(\epsilon^{1+2/n})$ in slow sliding regimes. Originally, this model was derived for geometries which do not vary in y but adapting it to a three-dimensional ice sheet is straightforward.

At low aspect ratios the full Stokes model can be approximated by the Blatter-Pattyn model. In this case, the pressure field is hydrostatic, and the $z-$component of the velocity field is eliminated from the $x-$ and $y-$components of the stress balance, which take the form

$$\partial_j \left(\tau_{ij}(x,y,z) + \delta_{ij}\tau_{kk}(x,y,z)\right) = \rho g \partial_i s(x,y) \qquad \text{for } i,k \in x,y \qquad j \in x,y,z. \tag{5}$$

Here, ρ is the density of ice, g is the gravitational acceleration, and δ_{ij} is the Kronecker delta. As the pressure field is hydrostatic, the grounding line is determined by a simple flotation criterion, and the upper surface elevation is related to the thickness through

$$s = \begin{cases} r + H & \text{if } \frac{\rho}{\rho_w} H > -r \\ (1 - \frac{\rho}{\rho_w}) H & \text{otherwise,} \end{cases} \tag{6}$$

where ρ_w is the density of ocean water. This allows the incompressibility equation to be integrated vertically to provide a horizontal, two-dimensional thickness transport equation. If we denote the vertical average of v_i by $\bar{v}_i$, then

$$\partial_t H(x,y) + \partial_i \left(\bar{v}_i(x,y) H(x,y)\right) = M_s(x,y) + M_b(x,y) \qquad \text{for } i \in x,y. \tag{7}$$

M_s and M_b are the rates of accumulation (or loss) of ice at the upper and lower surfaces respectively. In section 4.3, we will consider a case where M_b is large and negative underneath the ice shelf due to contact with warm water.

By making one further approximation (5) can integrated vertically, leading to the the final, two-dimensional elliptic equations. First, the deviatoric stress is split into horizontal and normal components ϑ_{ij} and vertical shear components φ_{ij}, such that

$$\boldsymbol{\tau}(x,y,z) = \begin{pmatrix} \vartheta_{xx} & \vartheta_{xy} & 0 \\ \vartheta_{xy} & \vartheta_{yy} & 0 \\ 0 & 0 & \vartheta_{zz} \end{pmatrix} + \begin{pmatrix} 0 & 0 & \varphi_{xz} \\ 0 & 0 & \varphi_{yz} \\ \varphi_{xz} & \varphi_{yz} & 0 \end{pmatrix}. \tag{8}$$

Then, (1) is replaced with

$$\dot{\zeta}_{ij}(x,y) = A \left(\vartheta^2(x,y,z) + \tilde{\varphi}^2(x,y,z)\right)^{(n-1)/2} \vartheta_{ij}(x,y,z) + O(\lambda\epsilon), \tag{9}$$

neglecting the $O(\lambda\epsilon)$ remainder, where

$$\dot{\boldsymbol{\zeta}} = \begin{pmatrix} \partial_x \mathrm{v}_x & \frac{1}{2}(\partial_y \mathrm{v}_x + \partial_x \mathrm{v}_y) & 0 \\ \frac{1}{2}(\partial_y \mathrm{v}_x + \partial_x \mathrm{v}_y) & \partial_y \mathrm{v}_y & 0 \\ 0 & 0 & -\partial_x \mathrm{v}_x - \partial_y \mathrm{v}_y \end{pmatrix} \tag{10}$$

and the $\tilde{\varphi}_{iz} \approx \varphi_{iz}$ are given by the shallow ice approximation

$$\tilde{\varphi}_{iz}(x,y,z) = -\rho g (s - z) \partial_i s(x,y) \qquad \text{for } i \in x,y. \tag{11}$$

Since $\dot{\zeta}$ depends only on the velocity field at the base of the ice, (5) can now be integrated with respect to z. Doing so, we arrive at the two-dimensional elliptic equation

$$\partial_j \left(H\bar{\mu}\, F_{ij}\right) + \tau_i^b = \rho g H \partial_i s \qquad i,j,k \in x,y, \tag{12}$$

where $F_{ij} = \partial_j \mathrm{v}_i + \partial_i \mathrm{v}_j + 2\delta_{ij}\partial_k \mathrm{v}_k$, $H\bar{\mu} = \int_b^s \mu(x,y,z)\mathrm{d}z$, and the effective viscosity μ satisfies

$$2\mu(x,y,z) A (4\mu^2(x,y,z)\dot{\zeta}^2(x,y) + \tilde{\varphi}^2(x,y,z))^{(n-1)/2} = 1. \tag{13}$$

The basal traction term is

$$\tau_i^b = \begin{cases} -C\left|\mathrm{v}_k\mathrm{v}_k\right|^{(m-1)/2}\mathrm{v}_i & \text{if } \frac{\rho}{\rho_w}H > -r \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Although (13) has an analytic solution when $n = 3$, it is somewhat cumbersome and we generally solve it numerically, using the secant method. Equations (12) are similar to the stress-balance equations which appear in the shelfy-stream model, reducing to just those if $\bar{\varphi} = 0$.

To form a closed problem, we need to complement (12) and (7) with the conditions at the lateral boundary of the ice sheet. At the ice divides, we impose a Dirichlet condition on the velocity normal to the divide

$$n_i\mathrm{v}_i(x,y) = 0 \qquad \text{for } i \in x,y \text{ and } x,y \in \Gamma_\mathrm{d} \tag{15}$$

and Neumann boundary conditions on the tangential velocity

$$n_i\partial_i\mathrm{v}_j(x,y)t_j = 0 \tag{16}$$

where $\vec{t}$ is a unit vector tangent to the boundary. A Neumann condition also applies to the thickness:

$$n_i\partial_i H(x,y) = 0. \tag{17}$$

Normal stress across the calving front is equal to the hydrostatic water pressure there, leading to a Neumann condition:

$$n_i H(x,y)\bar{\mu}(x,y)F_{ij}(x,y) = \frac{1}{2}\rho g\left(1-\frac{\rho}{\rho_w}\right)H^2(x,y)n_j \qquad \text{for } i \in x,y \text{ and } x,y \in \Gamma_{\mathrm{cf}}. \tag{18}$$

These boundary conditions, and indeed, (18) alone for a problem whose entire boundary is a calving front, are sufficient provided that $H(x,y,t=0)$ is given and that the basal friction coefficient C is non-zero in at least part of the ice sheet.

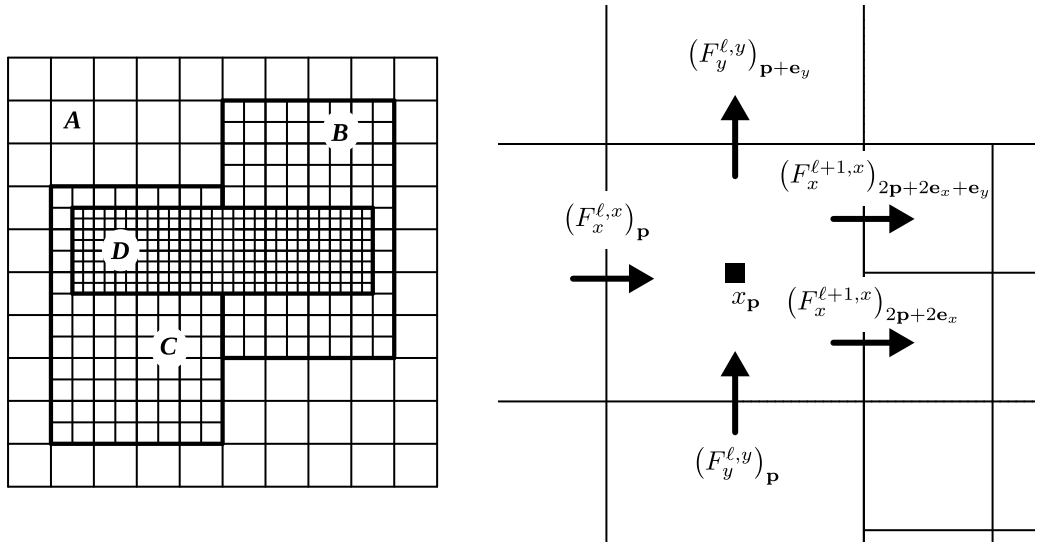
Having solved (12)-(18) for $\mathrm{v}_i(x,y)$, one needs to find $v_i(x,y,z)$ (and hence $\bar{v}_i(x,y)$). Properly, we should use the formula

$$\begin{aligned} v_i(x,y,z) &= \mathrm{v}_i(x,y) \\ &\quad +2\int_b^z A(4\mu^2(x,y,z)\dot{\zeta}^2(x,y)+\varphi^2(x,y,z))^{(n-1)/2}\varphi_{iz}(x,y,z)\,\mathrm{d}z' \end{aligned} \tag{19}$$

where

$$\begin{aligned} \varphi_{xz}(x,y,z) &= \int_z^s \partial_x\left[2\mu(x,y,z)(2\dot{\zeta}_{xx}(x,y)+2\dot{\zeta}_{yy}(x,y))\right]dz' \\ &\quad +\int_z^s \partial_y\left[2\mu(x,y,z)\dot{\zeta}_{xy}(x,y)\right]dz' \\ &\quad -\rho g(s-z)\partial_x s(x,y) \end{aligned} \tag{20}$$

and similarly for φ_{yz}. However, this introduces a diffusion-like flux into (7) so that the stable timestep decreases with the square of mesh resolution. While we made some progress toward stable time evolution by splitting the fluxes and evolving some parts implicitly, we could not stably treat grounding line evolution at the resolutions required. For the remainder of this paper we neglect this $O(\epsilon\lambda)$ additional velocity, so that we are limited to considering fast flowing ice streams, and set $v_i = \mathrm{v}_i$. The resulting scheme is still distinct from the shelfy-stream approximation due to the modified viscosity calculation. For example, the steady state grounding line we compute in section 4.2 is around 5 km short of that computed by the full Stokes model Elmer/Ice [36]: using the shelfy-stream model, it is more than 50 km in advance.



(a) Example block-structured mesh

(b) Fluxes at a coarse-fine interface

Figure 2: Block structured meshes. (a) Example block structured mesh. Discrete level domain Ω^0 comprises the cell centers of the coarsest grid, marked 'A', while while $\Omega^{0,x}$ and $\Omega^{0,y}$ are made up from the corresponding cell faces. Ω^1, $\Omega^{1,x}$ and $\Omega^{1,y}$ are built from two rectangular blocks, marked 'B' and 'C'; Ω^2, $\Omega^{2,x}$ and $\Omega^{2,y}$ from the block marked 'D'. The valid domains $\Omega^\ell_{\text{valid}}$ are the subsets of Ω^ℓ which are not overlaid by the cells of $\Omega^{\ell+1}$. (b) Fluxes at a coarse-fine interface. The discrete cell-centered divergence $\partial_i F_i$ integrated over a level ℓ cell adjacent to a coarse-fine boundary is expressed in terms of the valid face-centered vectors $F_i^{\ell,i}(\Omega^{\ell,i}_{\text{valid}})$ and $F_i^{\ell+1,i}(\Omega^{\ell,i}_{\text{valid}})$ only.

3. Numerical treatment

3.1. Discretization on block structured grids

It seems clear from the outset that we should make use of a spatially varying mesh resolution and also that the mesh might need to evolve over time [8, 10, 22, 23]. If we assume that a mesh spacing $\Delta x \sim 250$ m is needed at the grounding line, and that we want to model regions with surface areas of 10^{11} m^2 (the area of Pine Island Glacier's catchment) or more, then a uniform mesh would contain at least $\sim 10^6$ cells . If we can restrict such high resolution to only a small fraction of the domain, say, within a few kilometers of the grounding line, then we expect to reduce the number of degrees of freedom significantly. That said, the grounding line might migrate to sweep out a major proportion of the problem domain, so we will need to modify the mesh over time. To these ends, we discretize equations (12) and (7) on block-structured, non-uniform meshes using a finite volume method, supported by the Chombo AMR framework. [37]

As in [38], our meshes, illustrated in figure 2(a) for 2 dimensions, are composed of a hierarchy of cell-centered *level domains* $\Omega^0, \Omega^1, \ldots, \Omega^L$ on which the discrete thickness and velocity fields are located . Certain derived quantities, such as mass fluxes, are located on two supplementary hierarchies of face-centered level domains, $\Omega^{0,x}, \Omega^{1,x}, \ldots, \Omega^{L,x}$ and $\Omega^{0,y}, \Omega^{1,y}, \ldots, \Omega^{L,y}$. At each level ℓ, the problem domain is discretized on a grid Γ^ℓ with spacing Δx^ℓ chosen such that the *refinement ratio* $r^\ell = \frac{\Delta x^{\ell+1}}{\Delta x^\ell}$ is an even integer. Each Ω^ℓ is then constructed from one or more rectangular subsets of Γ^ℓ (and each $\Omega^{\ell,i}$ from the corresponding faces) such that Ω^ℓ is *properly nested*; that is, interfaces only exist between Ω^ℓ and $\Omega^{\ell\pm1}$, and never, say, Ω^ℓ and $\Omega^{\ell\pm2}$.

Having defined the level domains, we need to define notation for cell- and face-centered level variables. We define a grid vector, $\boldsymbol{p} \in \mathbb{Z}^2$, choosing to number cells starting at $\boldsymbol{p}_0 = (0,0)$, and grid basis vectors $\boldsymbol{e}_x = (1,0)$ and $\boldsymbol{e}_y = (0,1)$. Cell centers within Ω^ℓ are located at $\boldsymbol{x}^\ell_{\boldsymbol{p}} = \Delta x^\ell(\boldsymbol{p} + \frac{1}{2}\boldsymbol{e}_x + \frac{1}{2}\boldsymbol{e}_y)$, and the midpoints of cell faces within $\Omega^{\ell,i}$ at $\boldsymbol{x}^\ell_{\boldsymbol{p}} \pm \frac{\Delta x^\ell}{2}\boldsymbol{e}_i$. Cell-centered level variables $\phi^\ell_{\boldsymbol{p}} = \phi^\ell(\boldsymbol{x}^\ell_{\boldsymbol{p}})$ and face-centered level

variables $F^{\ell,i}_{\boldsymbol{p}} = F^{\ell,i}(\boldsymbol{x}^\ell_{\boldsymbol{p}} - \frac{\Delta x^\ell}{2}\boldsymbol{e}_i)$ follow naturally. Notice that $F^{\ell,x}_{\boldsymbol{p}}$ is located on the 'west' face of the cell $\boldsymbol{p}$, and $F^{\ell,y}_{\boldsymbol{p}}$ is on the 'south' face.

Following [38], the discrete representations of continuous fields are cell-centered composite variables ϕ^{comp} and face-centered composite variables F^{comp}, which are composed from portions of the level variables. First, level domains Ω^l are then divided into valid and invalid regions $\Omega^\ell_{\text{valid}}$ and $\Omega^\ell_{\text{invalid}}$, such that $\Omega^\ell_{\text{valid}}$ is the part of Ω^l not covered by $\Omega^{\ell+1}$. Then,

$$\phi^{\text{comp}}(\Omega^0_{\text{valid}} \cup \Omega^1_{\text{valid}} \cup \ldots \cup \Omega^L_{\text{valid}}) = \phi^0(\Omega^0_{\text{valid}}) \cup \phi^1(\Omega^1_{\text{valid}}) \cup \ldots \cup \phi^L(\Omega^L_{\text{valid}}), \tag{21}$$

and $F^{\text{comp,i}}$ is defined similarly. It is important to note that while we make use of $\phi^\ell(\Omega^\ell_{\text{invalid}})$ for computational convenience, our solutions depend on ϕ^{comp} and nothing else.

Given the valid and invalid regions, we now have two natural measures of the size of a given mesh: the valid cell count includes only the valid regions, while the total cell count includes the invalid regions too. While the valid cell count is related to the number of degrees of freedom, the total cell count is a better measure of computational cost, certainly in terms of storage, but also in terms of cost in time because we carry out most calculations on both the valid and invalid regions, replacing data in the invalid regions afterward.

We also construct *ghost regions* $\Omega^\ell_{\text{ghost}}$ that surround Ω^ℓ. They are used to store either data calculated by interpolation with $\Omega^{\ell-1}$ data, at coarse-fine interfaces between levels ℓ and $(\ell-1)$ or extrapolated data at the problem boundaries. As with the invalid regions, the ghost regions do not contribute any degrees of freedom to the problem and exist purely for numerical convenience.

Finite volume discretization of (7) and (12) on the meshes described above is a conventional affair, except at the coarse-fine boundaries separating $\Omega^\ell_{\text{valid}}$ and $\Omega^{\ell+1}_{\text{valid}}$. Modifications to the usual discrete operators are required for three terms: the flux divergence on the left hand side of (7), the driving stress on the right hand side of (12), and the viscous term on the left of (12). Such modifications are discussed elsewhere [39], but we describe them briefly here for clarity. Consider a level ℓ cell with its center at $\boldsymbol{x}^\ell_{\boldsymbol{p}}$, which is adjacent to a coarse-fine boundary along its $x-$face, such that the cell face $\boldsymbol{p}+\boldsymbol{e}_x \in \Omega^{\ell,x}_{\text{invalid}}$, as in figure 2(b). Introducing a shorthand notation for the cell average

$$\langle f \rangle_{\boldsymbol{p}} = \frac{1}{(\Delta x^\ell)^2} \int_{y^\ell_{\boldsymbol{p}}-\Delta x^\ell/2}^{y^\ell_{\boldsymbol{p}}+\Delta x^\ell/2} \int_{x^\ell_{\boldsymbol{p}}-\Delta x^\ell/2}^{x^\ell_{\boldsymbol{p}}+\Delta x^\ell/2} f(x,y)\, \mathrm{d}x\mathrm{d}y, \tag{22}$$

the divergence of a vector field can be approximated by applying the usual formula

$$\langle \partial_i F_i \rangle_{\boldsymbol{p}} = \frac{1}{\Delta x^\ell} \left\{ \left(F^{\ell,x}_x\right)_{\boldsymbol{p}+\boldsymbol{e}_x} - \left(F^{\ell,x}_x\right)_{\boldsymbol{p}} + \left(F^{\ell,y}_y\right)_{\boldsymbol{p}+\boldsymbol{e}_y} - \left(F^{\ell,y}_y\right)_{\boldsymbol{p}} \right\} + O\left((\Delta x^\ell)^2\right) \tag{23}$$

if and only if $(F^{\ell,x}_x)_{\boldsymbol{p}+\boldsymbol{e}_x}$ is expressed in terms of fluxes defined on $\Omega^{\ell+1,x}_{\text{valid}}$. Assuming also that the refinement ratio $r^\ell = 2$, the appropriate relation is:

$$\left(F^{\ell,x}_x\right)_{\boldsymbol{p}+\boldsymbol{e}_x} = \frac{1}{2} \left\{ \left(F^{\ell+1,x}_x\right)_{2\boldsymbol{p}+2\boldsymbol{e}_x+\boldsymbol{e}_y} + \left(F^{\ell+1,x}_x\right)_{2\boldsymbol{p}+2\boldsymbol{e}_x} \right\} + O\left((\Delta x^\ell)^2\right). \tag{24}$$

Note that (24) relies on our starting $\boldsymbol{p}$ at $(0,0)$.

As for the driving stress, an expression of the form

$$\langle \rho g H \partial_i s \rangle_{\boldsymbol{p}} = \rho g H_{\boldsymbol{p}} \frac{s^\ell_{\boldsymbol{p}+\boldsymbol{e}_i} - s^\ell_{\boldsymbol{p}-\boldsymbol{e}_i}}{2\Delta x^\ell} + O\left((\Delta x^\ell)^2\right) \tag{25}$$

applies once $s^\ell(\Omega^\ell_{\text{invalid}})$ has been calculated by averaging $s^{\ell+1}(\Omega^{\ell+1}_{\text{valid}})$, and $s^\ell(\Omega^\ell_{\text{ghost}})$ has been found by linear interpolation of $s^{\ell-1}(\Omega^{\ell-1}_{\text{valid}})$. Finally, the viscous term can be approximated using (22) provided that the face-centered gradients $\partial_j u^\ell_i(\Omega^{\ell,k}_{\text{ghost}})$ in both normal $(k=j)$ and transverse $(k \neq j)$ directions are determined from u^{comp}_i to second order accuracy, which is achieved by quadratic interpolation across the coarse-fine boundaries using the approach described in the appendix.

3.2. Grounding line

We replace (25) in the region of the grounding line with a formula that uses one-sided differences to evaluate the driving stress. In preliminary work, we treated the grounding line conventionally and encountered a step change in the discretized basal traction there,

$$\langle \tau_i^b \rangle_{\boldsymbol{p}} = \begin{cases} -C_{\boldsymbol{p}}^{\ell} \left(\mathrm{v}_j^{\ell} \mathrm{v}_j^{\ell} \right)_{\boldsymbol{p}}^{(m-1)/2} \left(\mathrm{v}_i^{\ell} \right)_{\boldsymbol{p}} & \rho H_{\boldsymbol{p}}^{\ell} > -\rho_w r \\ 0 & \text{otherwise} \end{cases} + O\left((\Delta x^{\ell})^2 \right), \tag{26}$$

but a smoothly changing driving stress term. The result was an unphysical combination of large driving stress and no basal stress in the floating ice adjacent to the grounding line, leading to unphysical thinning rates there especially at low resolution.

We retain (26), but alter (25) wherever cell centers at $\boldsymbol{p}$ and $\boldsymbol{p} - \boldsymbol{e}_i$ are grounded and the cell centers at $\boldsymbol{p} + \boldsymbol{e}_i$ and $\boldsymbol{p} + 2\boldsymbol{e}_i$ are floating. If the face-centered driving stress on the grounded side, located midway between $\boldsymbol{p}$ and $\boldsymbol{p} - \boldsymbol{e}_i$, is

$$R_{\boldsymbol{p}}^{\ell,i} = \rho g \frac{H_{\boldsymbol{p}}^{\ell} + H_{\boldsymbol{p}-\boldsymbol{e}_i}^{\ell}}{2} \frac{s_{\boldsymbol{p}}^{\ell} - s_{\boldsymbol{p}-\boldsymbol{e}_i}^{\ell}}{\Delta x^{\ell}} \tag{27}$$

then we modify the cell-centered stress to be

$$\langle \rho g H \partial_i s \rangle_{\boldsymbol{p}} = R_{\boldsymbol{p}}^{\ell,i} + O(\Delta x^{\ell}). \tag{28}$$

Likewise, we replace the cell-centered driving stress in the first floating cell with the face-centered driving stress on its floating side,

$$\langle \rho g H \partial_i s \rangle_{\boldsymbol{p}+\boldsymbol{e}_i} = R_{\boldsymbol{p}+2\boldsymbol{e}_i}^{\ell,i} + O(\Delta x^{\ell}). \tag{29}$$

We carry out a similar replacement when the cells $\boldsymbol{p}$ and $\boldsymbol{p}+\boldsymbol{e}_i$ are grounded and the cells $\boldsymbol{p}-\boldsymbol{e}_i$ and $\boldsymbol{p}-2\boldsymbol{e}_i$ are floating. Others have considered special treatment of the grounding line, imposing a known mass flux [6, 7, 9], softening the step change over an extended region [24, 32], or attempting to locate the grounding line with sub-grid resolution and modifying both basal traction and driving stress terms [23].

3.3. Calving front

Simply including appropriate ice free cells in our domains is enough to ensure that the calving front boundary condition (18) is satisfied. Consider a one-dimensional ice shelf with a calving front located at x_{cf}. The stress-balance equation (18) reduces to

$$\partial_x \left(H(x) \bar{\mu}(x) \, F_{xx}(x) \right) = \rho g \left(1 - \frac{\rho}{\rho_w} \right) H(x) \partial_i H(x) \tag{30}$$

and the boundary condition is

$$H(x_{\mathrm{cf}}) \bar{\mu}(x_{\mathrm{cf}}) \, F_{xx}(x_{\mathrm{cf}}) = \frac{1}{2} \rho g \left(1 - \frac{\rho}{\rho_w} \right) H^2(x_{\mathrm{cf}}) \tag{31}$$

and the cells up to $\boldsymbol{p}$ contain ice, while cells $\boldsymbol{p} + \boldsymbol{e}_x$ and onward are ice-free. If we integrate (30) over the first ice free cell $\boldsymbol{p} + \boldsymbol{e}_x$, we have

$$- \left(H \bar{\mu} \, F_{xx} \right)_{\boldsymbol{p}+\boldsymbol{e}_x}^{\ell,x} = O\left((\Delta x^{\ell})^2 \right) \tag{32}$$

because H is certainly zero beyond the face $\boldsymbol{p} + \boldsymbol{e}_x$, if undetermined at the face itself. Now, integrating over the last ice filled cell gives us

$$\left(H \bar{\mu} \, F_{xx} \right)_{\boldsymbol{p}+\boldsymbol{e}_x}^{\ell,x} - \left(H \bar{\mu} \, F_{xx} \right)_{\boldsymbol{p}}^{\ell,x} = \rho g H_{\boldsymbol{p}} \frac{H_{\boldsymbol{p}+\boldsymbol{e}_i}^{\ell} - H_{\boldsymbol{p}-\boldsymbol{e}_i}^{\ell}}{2} + O\left((\Delta x^{\ell})^2 \right). \tag{33}$$

so we can satisfy both (32) and (33) if we assume the calving front is just short of the cell face so that $H^{\ell,x}_{\boldsymbol{p}+\boldsymbol{e}_x} = 0$. That means that (33) does not retain any terms in $(\mathrm{v}^{\ell}_x)_{\boldsymbol{p}+\boldsymbol{e}_x}$ and the system of equations is closed. Finally, if we integrate (30) from x_{cf} to the cell face $\boldsymbol{p}+\boldsymbol{e}_x$,

$$(H\bar{\mu}\, F_{xx})^{\ell,x}_{\boldsymbol{p}+\boldsymbol{e}_x} - H(x_{\mathrm{cf}})\mu(x_{\mathrm{cf}})F_{xx}(x_{\mathrm{cf}}) = \frac{1}{2}\rho g\left(1-\frac{\rho}{\rho_w}\right)\left(H^{\ell,x}_{\boldsymbol{p}+\boldsymbol{e}_x} - H^2(x_{\mathrm{cf}})\right), \tag{34}$$

and since $H^{\ell,x}_{\boldsymbol{p}+\boldsymbol{e}_x} = 0$, the boundary condition (31) is satisfied. The same idea applies in two dimensions. It is hardly surprising that the calving front condition is equivalent to having an adjacent ice-free region. However, imposing the boundary condition in this way allows us to avoid explicit calculation of velocity gradients at the calving front.

3.4. Solution of the stress-balance equations

To solve the nonlinear force balance equations, we implemented both a Jacobian-free Newton-Krylov (JFNK) method [40, 41], and a simpler Picard iteration method. As others have noted [42], it is possible to share a great deal between the two methods. Most importantly, we were able to use the same multigrid preconditioner when solving the different linearized force balance equations which arise.

We need to introduce some notation just for the purposes of this section. First we subsume a composite vector field $\mathrm{v}^{\mathrm{comp}}_i$ into a single state vector, $\mathbf{v}$. Then, we can write the discretized stress-balance equation, including boundary conditions as

$$L[\mathbf{v}]\mathbf{v} = \boldsymbol{r}, \tag{35}$$

where $L[\mathbf{v}]$ is a linear operator defined by a particular $\mathbf{v}$. $L[\mathbf{v}]$ is linear in the usual sense, that is,

$$L[\mathbf{v}](\boldsymbol{p}+\boldsymbol{q}) = L[\mathbf{v}]\boldsymbol{p} + L[\mathbf{v}]\boldsymbol{q}. \tag{36}$$

Note that $L[\mathbf{v}]$ is not symmetric, due to its formulation at coarse-fine interfaces. We describe the multilevel discretization of $L[\mathbf{v}]$ in the appendix.

Equation (35) can be solved by means of Picard iterations, as follows:

1. choose an initial $\mathbf{v}^k = \mathbf{v}^0$ and a tolerance ϵ;
2. compute the effective viscosity and basal traction coefficients given $\mathbf{v}^k$;
3. if $\left|\left|L[\mathbf{v}^k]\mathbf{v}^k - \boldsymbol{r}\right|\right|_\infty < \epsilon \left|\left|L[\mathbf{v}^0]\mathbf{v}^0 - \boldsymbol{r}\right|\right|_\infty$ stop, else
4. solve the linear system $L[\mathbf{v}^k]\mathbf{v}^{k+1} = \boldsymbol{r}$, set $k \leftarrow k+1$, and repeat 2-4

Step 4 requires solution of the linear system $L[\mathbf{v}^k]\mathbf{v}^{k+1} = \boldsymbol{r}$, and we employ a either a multigrid method or a multigrid-preconditioned BiCGStab [43] method to do so. The multigrid method is straightforward; we simply compute a sequence of approximations to $\mathbf{v}^{k+1}$,

$$\mathbf{v}^{k+1}_{j+1} = \mathbf{v}^{k+1}_j + M^{-1}[\mathbf{v}^k](\boldsymbol{r} - L[\mathbf{v}^k]\mathbf{v}^{k+1}_j) \tag{37}$$

starting from $\mathbf{v}^{k+1}_0 = \mathbf{v}^k$, and stopping once $\left|\left|\boldsymbol{r} - L[\mathbf{v}^k]\mathbf{v}^{k+1}_j\right|\right|_\infty$ drops below some chosen tolerance . Of course, we do not form a matrix $M^{-1}[\mathbf{v}^k]$ explicitly; rather it represents the execution of one multigrid V-cycle given the viscosity and basal friction coefficients implied by $\mathbf{v}^k$. As for BiCGStab , if we wish to solve a preconditioned linear system

$$L[\mathbf{v}^k]P^{-1}[\mathbf{v}^k]P[\mathbf{v}^k]\mathbf{v}^{k+1} = \boldsymbol{r} \tag{38}$$

we need to compute a vector $L[\mathbf{v}^k]P^{-1}[\mathbf{v}^k]\boldsymbol{q}$ twice per iteration. To do so, we calculate a vector $\tilde{\boldsymbol{q}} = P^{-1}[\mathbf{v}^k]\boldsymbol{q}$ by carrying out one or more iterations of (37), that is, one or more V-cycles, and then calculate $L[\mathbf{v}^k]\tilde{\boldsymbol{q}}$.

JFNK methods rely on a well-known property of Krylov subspace projection methods (including BiCGStab): they require only matrix-vector products $J\boldsymbol{y}$ to be computed in order to find an approximate solution to the linear system $J\boldsymbol{x} = \boldsymbol{b}$. The same is true of (37). If $J[\mathbf{v}^k]$ is the Jacobian operator which appears in Newton's method applied to (35), then

$$J[\mathbf{v}^k]\boldsymbol{y} \approx \frac{1}{h}\left(L[\mathbf{v}^k + h\boldsymbol{y}](\mathbf{v}^k + h\boldsymbol{y}) - L[\mathbf{v}^k]\mathbf{v}^k\right) \tag{39}$$

where h is small but finite, and the elements of J itself need never be computed. We can then use the same methods and the same multigrid operator $M^{-1}[\mathbf{v}^k]$ to solve the resulting linear system as we did for the Picard methods. We therefore have a multigrid method

$$\mathbf{v}_{j+1}^{k+1} = \mathbf{v}_j^{k+1} + M^{-1}[\mathbf{v}^k](\boldsymbol{r} - J[\mathbf{v}^k]\mathbf{v}_j^{k+1}) \tag{40}$$

and a BiCGStab method, where we calculate $\tilde{\boldsymbol{q}} = P^{-1}[\mathbf{v}^k]\boldsymbol{q}$ in the same way as before, and then apply (39) to compute $J[\mathbf{v}^k]P^{-1}\tilde{\boldsymbol{q}}$. In essence, we are assuming that approximations to $L[\mathbf{v}^k]$ are a good choice of preconditioner for $J[\mathbf{v}^k]$.

Although $L[\mathbf{v}^k]$ was chosen to precondition the JFNK linear system for pragmatic reasons, in some circumstances we can see that it is the ideal choice, and furthermore, that the Picard iterations are bound to converge more slowly. Consider a one-dimensional ice stream or shelf where both v_x and $\partial_x \mathrm{v}_x$ are positive and $\partial_x s$, v_y and all $y-$derivatives are negligible. Given these assumptions,

$$L[\mathrm{v}_x]w \approx \partial_x\left(A^{-1/n}\left(\partial_x \mathrm{v}_x\right)^{1/n-1} w\right) - C\left(\mathrm{v}_x\right)^{m-1} w \tag{41}$$

while

$$J[\mathrm{v}_x]w \approx \frac{1}{n}\partial_x\left(A^{-1/n}\left(\partial_x \mathrm{v}_x\right)^{1/n-1} w\right) - mC\left(\mathrm{v}_x\right)^{m-1} w. \tag{42}$$

If $C = 0$, or $C(x, y)$ is constant and, as is common, $m = 1$ or $m = 1/n$, L and J have the same eigenvectors, and, if $C = 0$ or $m = 1/n$, $L^{-1}J$ is diagonal. On top of that, provided that $n \geq 1$ and $m \leq 1$ — which is usually the case — each eigenvalue of J is smaller than the corresponding eigenvalue of L, so the steps taken in the Picard iterations are shorter than the Newton steps, constituting a form of under-relaxation.

Both the Picard and JFNK solvers require an initial guess for the velocity fields $\mathbf{v}^0$. In idealized geometries without floating ice shelves we find that the obvious initial guess, $\mathbf{v}^0 = 0$, works well enough but in other cases, especially in realistic problems we see only slow or erratic reduction of the residual. Instead, we choose a viscosity coefficient that is constant over the domain, and, when the basal friction law is not linear, an initial non-zero constant velocity field. That leads to a linear system, which we solve to give $\mathbf{v}^0$. We also find that the radius of convergence of the JFNK solvers are somewhat smaller than the Picard solvers, so we initialize those by carrying out a few Picard iterations.

3.5. Explicit treatment of the thickness evolution equation

To compute a conservative update of the ice thickness from time t^n to time $t^{n+1} = t^n + \Delta t$, we follow the approach used to advect passive scalars in [39]. Given $H^\ell(t^n)$ and $\bar{v}_i^l(t^n)$, the cell-centered ice thickness and vertically averaged velocity at time t^n, we require face-centered evaluations of the thickness fluxes $Q_i^{\ell,i}(t^n + \frac{1}{2}\Delta t)$. We first make use of the piecewise parabolic method (PPM)[44], an upwind scheme that uses a Taylor expansion of (7) in time and space to provide explicit predictions of the face-centered thickness values $H^{\ell,i}(t^n + \frac{1}{2}\Delta t)$. We then multiply these face-centered thickness values by a face-centered velocity $V_i^{\ell,i}$ to compute the requisite fluxes, $Q_i^{\ell,i}(t^n + \frac{1}{2}\Delta t) = H^{\ell,i}(t^n + \frac{1}{2}\Delta t)V_i^{\ell,i}$. Last of all, we compute the multilevel divergence of these fluxes and update the ice thickness according to (7). Now, if $V_i^{\ell,i}$ was centered at $t^{n+\frac{1}{2}}$, we would have a temporally second-order scheme, but in our initial implementation we derive $V_i^{\ell,i}$ by averaging the cell-centered $\bar{v}_i^\ell(t^n)$ to cell faces. So we expect our scheme, even without the difficulties introduced by the grounding line, to only be temporally first-order. It is still spatially second order for problems without grounding lines.

Our AMR discretization is simplified by the fact that unlike the scheme described in [38], we do not refine temporally but instead all AMR levels are advanced at the same timestep: $\Delta t^\ell = \Delta t$. While this results in some inefficiency (all AMR levels are advanced at a timestep dictated by the stability requirements of the finest level), temporal refinement (also known as sub-cycling) has proved most useful for problems which are hyperbolic in nature, and those with deep AMR hierarchies which span large ranges of spatial

scales. Because this problem is dominated by the elliptic nature of the ice velocity equation, and because our refinement hierarchies tend to be only moderately deep, we have not found sub-cycling to be necessary for the problems described here. Unlike the relatively simple sub-cycled AMR schemes for fully hyperbolic systems like gas dynamics [45], a subcycled scheme which correctly respects the elliptic nature of this problem would result in a significantly more complex algorithm along the lines of that in [38], without obviously improving the actual efficiency of the code due to the increased computational overhead inherent in such an algorithm.

3.6. Mesh refinement and coarsening

To generate an initial mesh, we begin with single level mesh with a single domain, Ω^0, that spans the problem domain at the coarsest resolution Δx^0. We find a solution to the stress balance equations there, and then apply refinement criteria to tag regions of the base level domain for refinement to level 1. Next, the Berger-Rigoutsos point-clustering algorithm[46] is employed to form the refined level 1 domains, which will cover all of the level 0 tags as well as some of the surrounding untagged region. We then need to compute a velocity field on the new mesh, which we do by re-solving the stress-balance equations starting from the level 0 solution. That completed, regions of Ω^1 are tagged according to the same criteria, leading to a yet more refined mesh and solution, and so on until no further refinement is indicated or until the finest level, L, is reached.

A similar procedure is then used to generate a new mesh every n_{regrid} timesteps, refining some regions and coarsening others. First, the velocity and thickness fields are approximated in the invalid regions of $\Omega^0 \ldots \Omega^{L-1}$ by averaging the fields from Ω^L onto Ω^{L-1} and so on down. Next, the cells in both the valid and invalid regions of each level domain are tagged according to the chosen refinement criteria, and the tagged cells are used to generate a new mesh. Any regions of $\Omega^\ell_{\text{valid}}$ that are tagged will certainly be refined relative to the older mesh. On the other hand, if the approximation to the velocity and thickness fields in a given region of $\Omega^\ell_{\text{invalid}}$ is such that the refinement criteria are no longer met the new mesh may — depending on the behavior of the Berger-Rigoutsos algorithm — be coarsened there.

Clearly, the success or failure of our methods will depend in large part on the refinement criteria we choose. We will investigate two physically motivated criteria in section 4: in 4.1 we refine the mesh in the region of a fast-flowing, narrow ice stream by considering the Laplacian of the velocity field, and in 4.2 and 4.3 we choose meshes that are refined in the region of the grounding line by tagging cells adjacent to it. In other problems, such as in our submission to the MISMIP3D inter-comparison exercise[36], we use both criteria.

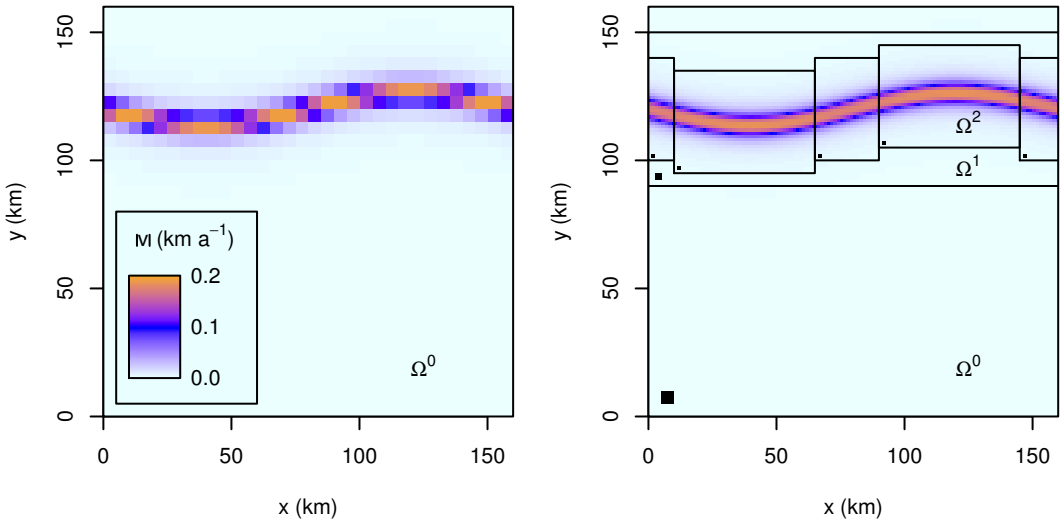
Block-structured meshes are well suited to problems, like that of grounding line evolution, where spatial resolution needs to be time-dependent. Mesh generation itself is inexpensive, but more importantly, it is straightforward to transfer the ice sheet state from one mesh to another. Imagine that at time t we need to replace a union of valid sub-domains $\Omega^{\ell,\text{valid,t}-}$ with a new set $\Omega^{\ell,\text{valid,t}+}$. Wherever $\Omega^{\ell,\text{valid,t}+}$ intersects $\Omega^{\ell,\text{valid,t}-}$, we simply set $H(\Omega^{\ell,\text{valid,t}+}) = H(\Omega^{\ell,\text{valid,t}-})$ to obtain the thickness on the new mesh. Linear interpolation is used to determine $H(\Omega^{\ell,\text{valid,t}+})$ where the mesh is newly refined, that is, where $\Omega^{\ell,\text{valid,t}+}$ replaces regions of $\Omega^{\ell-1,\text{valid,t}-}$. The slope-limiting scheme due to van Leer [47] prevents the introduction of spurious extrema. Likewise, arithmetic averaging is used in regions where the mesh is coarsened. In principle, we could follow the same procedure with the velocity components, resulting in a field $\mathrm{v}_i^*(\Omega^{\ell,\text{valid,t}+})$ but we obtain better accuracy by solving the stress-balance equations as described in 3.4 using $\mathrm{v}_i^*(\Omega^{\ell,\text{valid,t}+})$ as an initial guess, which is potentially as expensive as solving the stress-balance equations starting from $\mathrm{v}_i = 0$ (or some other guess far from the solution), but is usually less so.

4. Numerical tests

4.1. AMR ice stream.

To demonstrate the effectiveness of our AMR approach on a simple fully-grounded ice sheet, we look at a simple ice stream in a doubly-periodic domain of size 160km × 160km. In this example, the ice has constant thickness $H = 1$ km on a sloping bed,

$$r = -x\cos(\theta) \tag{43}$$



(a) Single level solution with $\Delta x = 5$ km

(b) AMR solution with $L = 2$ and $\Delta x^L = 1.25$ km

Figure 3: Ice stream velocity magnitude, computed on a single mesh with $\Delta x = 5$ km (a) and on a refined mesh with $L = 2$ and $\Delta x^L = 1.25$ km. Finer level domains Ω^1 and Ω^2 are centered on the fast flowing ice, with Ω^2 made up from several rectangular patches. Each patch includes a dark cell at the bottom left indicating the mesh spacing.

where $\theta = 0.5°$. The basal friction law is linear, with a coefficient given by

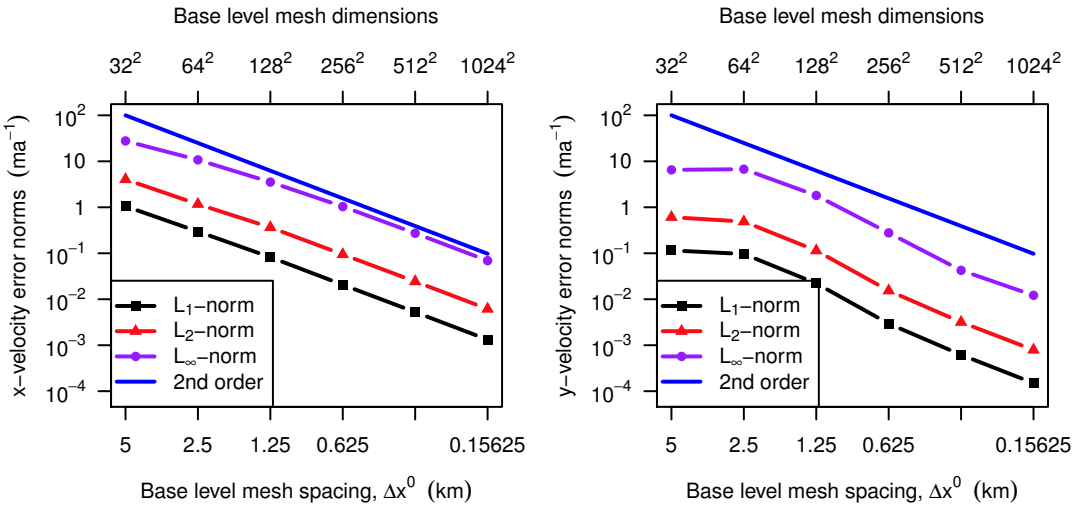
$$C(x, y) = C_0 \left\{ 1 + \epsilon + \sin \left[\frac{2\pi y}{R} + m \sin \left(\frac{2\pi x}{R} \right) \right] \right\} \tag{44}$$

where $C_0 = 1.0 \times 10^5$ Pa m^{-1} a, $\epsilon = 5.0 \times 10^{-5}$, $R = 160$ km and $m = 0.25$. We compute the velocity field for both single-level and AMR solutions, with a refinement criterion based on the Laplacian of the velocity field. Figure 3(a) depicts a single level solution with mesh spacing $\Delta x = 5$ km. Starting from there, cells on level ℓ are tagged wherever the size of the undivided Laplacian, $\left| 4\phi^\ell_{\boldsymbol{p}} - \phi^\ell_{\boldsymbol{p}+\boldsymbol{e}_x} - \phi^\ell_{\boldsymbol{p}+\boldsymbol{e}_x} - \phi^\ell_{\boldsymbol{p}+\boldsymbol{e}_y} - \phi^\ell_{\boldsymbol{p}+\boldsymbol{e}_y} \right|$, exceeds a threshold value for either $\phi^\ell = \mathrm{v}^\ell_x$ or $\phi^\ell = \mathrm{v}^\ell_y$, leading to mesh refinement around the ice stream in the upper half of the figure. After two rounds of refinement, we arrive at the solution shown in figure 3(a), which has finer level domains centered on the fast flowing region, and coarse resolution elsewhere.

We first use Richardson extrapolation to calculate the convergence rate of the velocity field on uniform meshes, which involves pairwise comparison between solutions with grid spacing Δx and $2\Delta x$. Convergence results are shown in figure 4; both velocity components exhibit second-order convergence.

To be effective, our AMR scheme should produce solutions with comparable accuracy to a uniform mesh solution with the same resolution. To demonstrate this, we compute a uniform-mesh solution on a very fine mesh (in this case 2048×2048), which we take as the "exact" solution, against which we can compute "error" for single-level and AMR solutions. This will give a slightly different convergence rate from Richardson extrapolation, but still indicates that the error in our AMR solutions is comparable to that of the single-level solution with the same effective resolution. Figure 5 shows that for two-level solutions with refinement ratios of $r^0 = 2$, $r^0 = 4$, and for three-level solutions with $r^0 = r^1 = 2$ we see the same error as the equivalent-resolution single-level solutions for the L_1- and L_∞-norms (the L_2-norm is similar) – a horizontal line drawn from a point on the "uniform mesh" line should intersect the corresponding points on the other lines, as is generally the case.

The multigrid-based methods that we use to solve the stress balance equations perform well enough that a total cell count, including both $\Omega^\ell_{\text{valid}}$ and $\Omega^\ell_{\text{invalid}}$, is a sensible indicator of the computational expense of a given mesh. Figure 6 plots the residual norm of the stress balance equations, $\left|\left| L[\mathbf{v}^k]\mathbf{v}^k - \boldsymbol{r} \right|\right|_\infty$, as a



(a) $x-$velocity (b) $y-$velocity

Figure 4: Richardson convergence of velocity

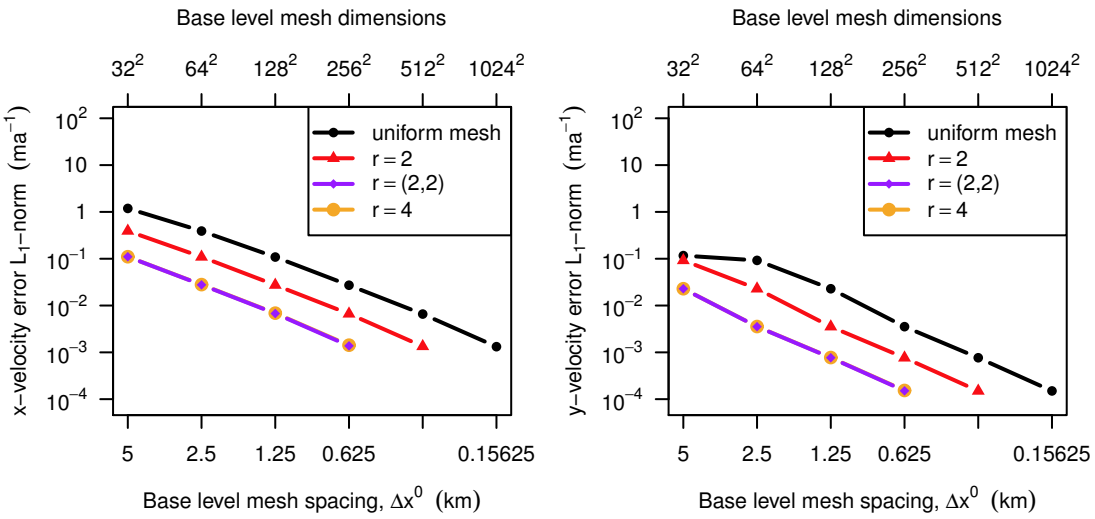
(a) Comparison of solver methods

Mesh details	Solution method	CPU time (s)
AMR, $L = 2$, $\Delta x^0 = 2.5$ km, $\Delta x^L = 625$ m, 16,096 valid cells, 20,096 total cells	Picard/MG	133
	Picard/BiCGStab	189
	JFNK/MG	66
	JFNK/BiCGStab	93
uniform, $\Delta x = 625$ m, 65,536 cells	Picard/MG	572
	Picard/BiCGStab	1730
	JFNK/MG	203
	JFNK/BiCGStab	210

(b) Comparison of AMR and uniform mesh performance

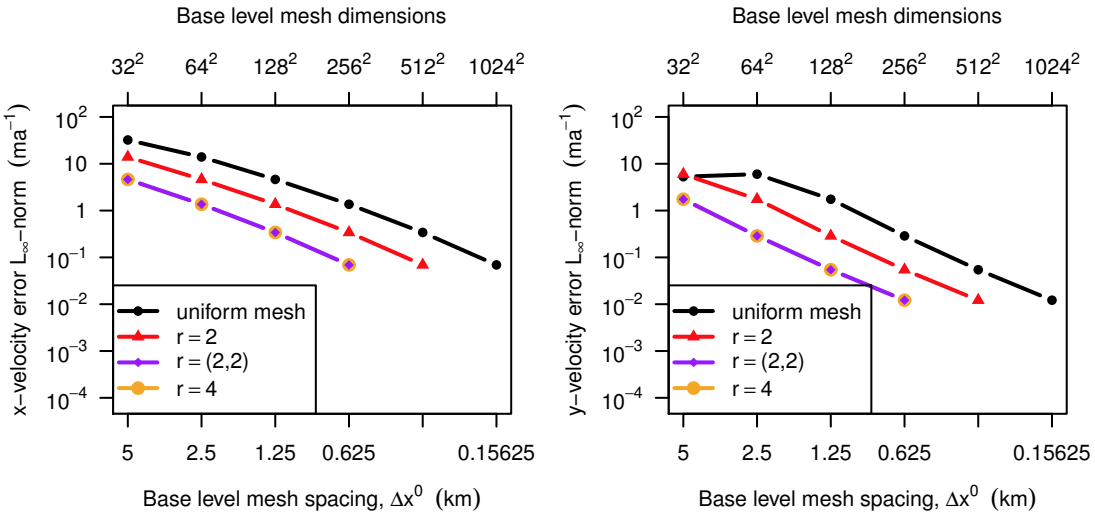
Δx^L	L	Δx^0	valid cells	total cells	CPU time (s)
1.25 km	0	1.25 km	16,384	16,384	52
1.25 km	2	5 km	5,248	6,656	17
625 m	0	625 m	65,536	65,536	203
625 m	2	2.5 km	16,096	20,096	66
312.5 m	0	312.5 m	262,144	262,144	957
312.5 m	2	1.25 km	70,372	88,368	318
156.25 m	0	156.25 m	1,048,576	1,048,576	3,810
156.25 m	2	625 m	276,064	346,240	1,460

Table 1: Solver performance for the the ice stream problem. (a) Comparison of solver methods for AMR and uniform meshes with $\Delta x^L = 625$ m. Complementary plots are given in figure 6. (b) Comparison of AMR and uniform mesh performance, for the JFNK/MG method only. This table corresponds to the uniform mesh and $r = (2, 2)$ curve in figure 5. CPU times given for the AMR runs include the entire calculation, starting from a coarse mesh. Calculations were carried out in serial on a 2.26GHz Intel Core 2 Duo P7550 CPU.



(a) x-velocity error L_1-norm

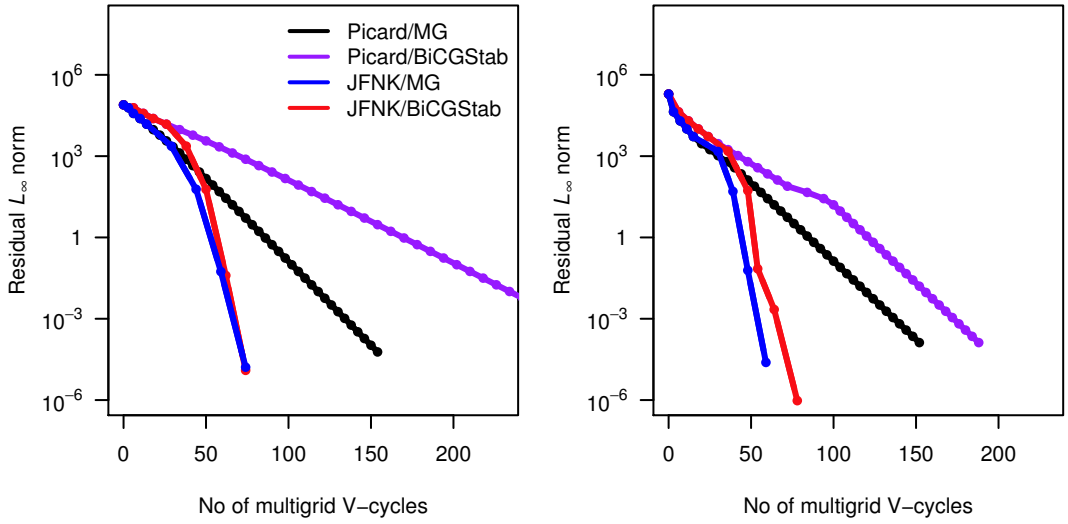
(b) y-velocity error L_1-norm



(c) x-velocity error L_∞-norm

(d) y-velocity error L_∞-norm

Figure 5: Error estimates computed by comparing AMR solutions with a fine mesh reference solution.



(a) AMR, $L = 0, \Delta x^L = 2.5$ km

(b) AMR, $L = 1, \Delta x^L = 1.25$ m

(c) AMR, $L = 2, \Delta x^L = 625$ m

(d) uniform mesh,$\Delta x = 625$ m

Figure 6: Performance of the Picard and JFNK solvers in AMR and uniform mesh ice stream experiments. The outer residual norm $\left|\left|L[\mathbf{v}^k]\mathbf{v}^k - \boldsymbol{r}\right|\right|_\infty$ is plotted against the number of multigrid V-cycles, and each outer iteration k is marked with a solid circle. The residual is reduced at a similar rate on coarse (a) and fine (d) uniform meshes, and as the AMR calculation progresses through additional levels of refinement (b) and (c). Results are shown for the Picard methods, using the simple multigrid method (Picard/MG) and the multigrid-preconditioned BiCGStab method (Picard/BiCGStab), and for the corresponding JFNK methods (JFNK/MG and JFNK/BiCGStab). Note that the JFNK methods are identical to the Picard methods at first, because we initialize them with a few Picard iterations. CPU times and cell counts are given in table 1.

function of the number of multigrid V-cycles taken, for the four solution methods available to us. Plots 6(a)–(c) track the solution of an AMR problem, starting from a coarse uniform mesh with $\Delta x^0 = 2.5$ km, and progressing through one level of refinement (b, $L = 1$) and then two (c, $L = 2$), to a final solution with $\Delta x^L = 625$ km. Figure 6(d) shows a similar plot for a uniformly fine mesh with $\Delta x = 625$ km.

Most importantly we see that the number of multigrid V-cycles needed to progress from level ℓ to level $\ell + l$ does not grow as we add levels of refinement. Our program spends most of its time carrying out V-cycles, and since they have a cost in time proportional to the total cell count, the AMR solution should be far less costly than the uniform mesh solution. Measuring the CPU time for the uniform and AMR mesh calculations supports this, as shown in table 1, where the uniform meshes typically have three times as many cells as the AMR meshes, and the runs take three times as long.

As expected, and as others have noted for related problems [42, 48], the Newton methods outperform the Picard methods by some margin. Comparing the simple multigrid methods first, in figures 6, the Newton method (JFNK/MG) residual drops at an increasing rate, outstripping the corresponding Picard/MG method. That results in calculations that are around twice as fast in the AMR case and three times as fast in the uniform mesh case, according to table 1. The BiCGStab calculations are somewhat more mercurial, and although JFNK/BiCGStab can deliver similar results to JFNK/MG, it is often slower, while Picard/BiCGStab performs worst of all. Later on we will see that in the Pine Island Glacier example, JFNK/BiCGStab outperforms JFNK/MG by some margin, but overall we find JFNK/MG to be the most reliable method.

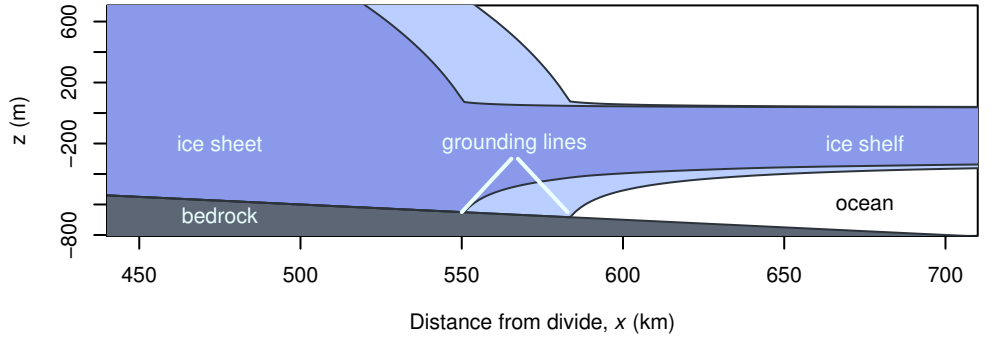
4.2. Grounding line steady states

We considered the evolution of an ice stream towards steady state, in tests similar to the forthcoming marine ice sheet inter-comparison 3D (MISMIP3D) standard experiment. An ice stream sits on an inclined plane sloping down to the sea such that $r = -100 - x/1000$ m. There is an ice divide at $x = 0$ km; a marine boundary at $x = 800$ km; and, for our simulations, periodic conditions at $y = 0$ km and $y = 100$ km. Values for the parameters in (7) and (12) were the same as those prescribed for MISMIP3D: $n = 3$, $m = 1/3$, $A = 3.1536 \times 10^{-18}$ Pa^{-3} a^{-1}, $C = 31651.76$ Pa m$^{-1/3}$ a$^{1/3}$, $\rho = 900$ kg m^{-3}, $\rho_w = 1000$ kg m^{-3}, $g = 9.8$ m s^{-2} and $M_s = 500$ mm a^{-1}. Given these conditions, one expects a unique steady state solution which does not vary in y, consisting of an ice sheet joined to an ice shelf across a grounding line at $x = x_{gl}$, much as in the earlier flowline marine ice sheet inter-comparison (MISMIP) exercise [49].

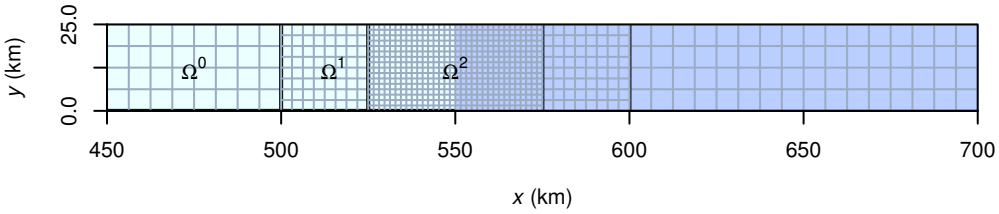
As this problem has no variation in y we were able to carry out both one-dimensional uniform mesh simulations, and two-dimensional AMR simulations over a range of resolutions, which allowed us to both study the convergence of the basic discretization scheme with mesh spacing, and compare AMR and uniform mesh solutions. The uniform mesh simulations were run at resolutions ranging from $\Delta x = 12.5$ km (128 cells) to $\Delta x = 24$ m (32768 cells), starting from a 150 m thick slab of ice and then running for 50,000 years — sufficient time to reach a steady-state. As for the AMR calculations, the finest resolution was varied from $\Delta x^L = 6.25$ km to $\Delta x^L = 195$ m. These were also started from a 150 m thick slab of ice and run for 50,000 years, with the first AMR level added after 10,000 years and another every 1,000 years after that. New meshes were generated at every timestep by tagging 4 cells for refinement on either side of the grounding line at every level $\ell < L$. We arrived at the number 4 by experimentation : tagging fewer cells leads to meshes with a similar number of degrees of freedom but with the grounding line much closer to one coarse-fine interface than the other, while tagging many more makes little difference to the results but does increase the computational cost. Figure 7 show an ice sheet with two levels of refinement at a point during its evolution and at steady state along with the corresponding meshes.

Convergence with mesh resolution is at best first-order in Δx, and that modest target is only achieved at very fine resolutions.[2] Figure 8(a) plots the final grounding line positions attained in the flow-line shelfy-stream and Schoof-Hindmarsh simulations against mesh resolution. In the shelfy-stream calculations, the

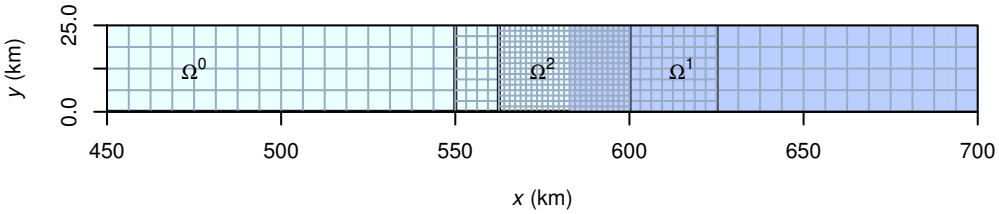
[2]We place the grounding line at cell faces between grounded and floating cells having computed $H(x)$, so that a first-order error in x_{gl} of at least $\Delta x/2$ would be present even if we knew $H(x)$ exactly. The error in the calculation of $H(x)$ is much larger.



(a) Ice sheet geometry

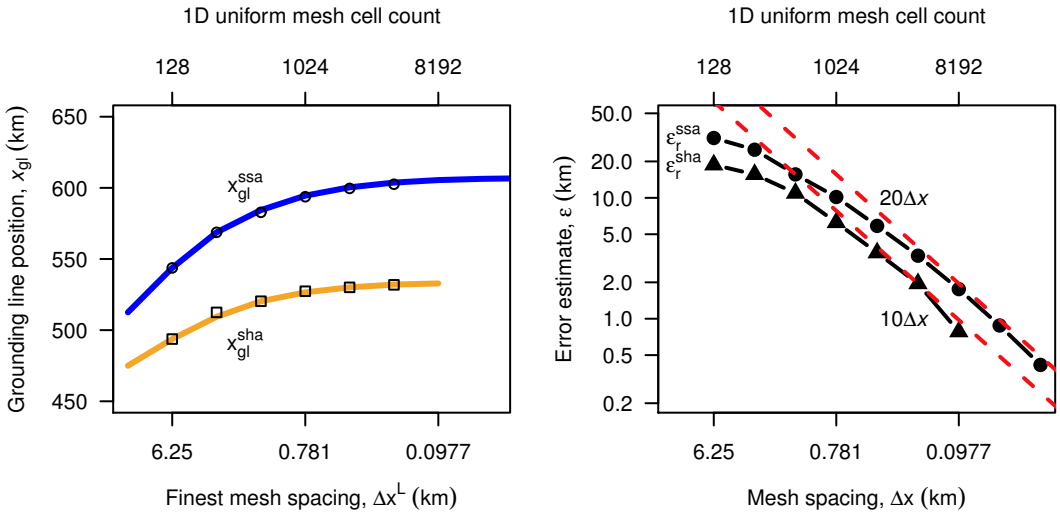


(b) Mesh during evolution



(c) Mesh at steady state

Figure 7: Cross-section through part of an AMR ice sheet geometry close to the grounding line during evolution and at steady state (a) with corresponding meshes (b) and (c). The figures show a model with $L = 2$ levels of refinement in addition to the base level. Starting from a slab of uniform thickness, the ice sheet is first evolved for ten thousand years at low resolution, then progressively finer AMR levels are added every thousand years until there are L. At that time the ice sheet, shown as a dark shaded region in panel (a) is still far from equilibrium. From then the model is run to steady state, corresponding to the light shaded region in panel (a), over tens of thousand of years. As the grounding line migrates, so does the finely resolved portion of the mesh. In both (b) and (c) the unshaded area is filled with grounded ice and the shaded area with floating ice.



(a) AMR and uniform mesh grounding line positions (b) Uniform mesh grounding line error estimates

Figure 8: (a) Steady-state grounding line positions computed with the shelfy-stream ($x_{\mathrm{gl}}^{\mathrm{ssa}}$) and Schoof-Hindmarsh ($x_{\mathrm{gl}}^{\mathrm{sha}}$) approximations plotted against grid resolution. Results from the 1D uniform mesh calculations are plotted with solid lines and results from the 2D AMR calculations are plotted with symbols. (b) Estimates of the error in grounding line position x_{gl} plotted against grid resolution. For the shelfy-stream calculations, the Richardson estimates $\epsilon_{\mathrm{r}}^{\mathrm{ssa}}$ are close to $20\Delta x$ (the upper dashed line) once $\Delta x \lessapprox 500$ m. For calculations carried out using the Schoof-Hindmarsh approximation, the Richardson estimates ultimately satisfy $\epsilon_{\mathrm{r}}^{\mathrm{sha}} \approx 10\Delta x$ (the lower dashed line).

grounding line position $x_{\mathrm{gl}}^{\mathrm{ssa}}$ changes by around 100 km (one eighth of the problem domain's length) from the coarsest to the finest mesh: from $x_{\mathrm{gl}}^{\mathrm{ssa}} = 513$ km when $\Delta x = 12.5$ km to $x_{\mathrm{gl}}^{\mathrm{ssa}} = 606.8$ km when $\Delta x = 0.0244$km. In comparison, an analytic approximation to the shelfy-stream model gives $x_{\mathrm{gl}}^{\mathrm{a}} = 606.8$ km [6, 7]. The Schoof-Hindmarsh grounding line position $x_{\mathrm{gl}}^{\mathrm{sha}}$ is somewhat further upstream, varying over around 50 km from $x_{\mathrm{gl}}^{\mathrm{sha}} = 475$ km when $\Delta x = 12.5$ km to $x_{\mathrm{gl}}^{\mathrm{sha}} = 532.8$ km when $\Delta x = 0.0977$ km . Both sets of results do appear to converge on a limit as Δx decreases, and the nature of this convergence is evident in figure 8b. Two error estimates are plotted there: a Richardson estimate of the error in $x_{\mathrm{gl}}^{\mathrm{ssa}}$,

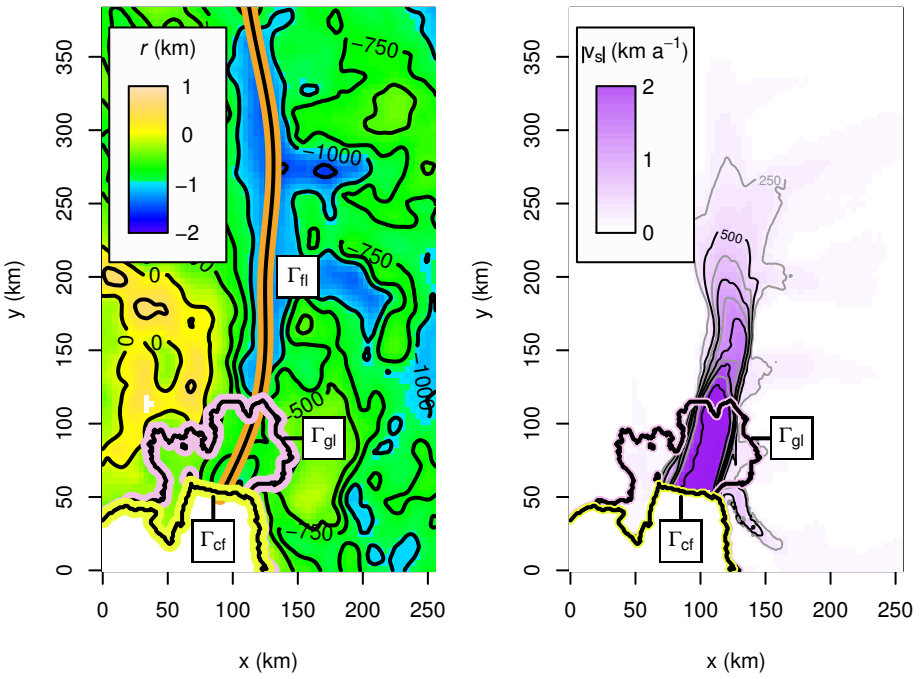
$$\epsilon_{\mathrm{r}}^{\mathrm{ssa}}(\Delta x) = \left| x_{\mathrm{gl}}^{\mathrm{ssa}}(\Delta x) - x_{\mathrm{gl}}^{\mathrm{ssa}}(2\Delta x) \right|, \tag{45}$$

and a Richardson estimate, $\epsilon_{\mathrm{r}}^{\mathrm{sha}}$ of the error in $x_{\mathrm{gl}}^{\mathrm{sha}}$. For each, the rate of convergence approaches first-order with $\epsilon_{\mathrm{r}}^{\mathrm{ssa}} \approx 20\Delta x$ and $\epsilon_{\mathrm{r}}^{\mathrm{sha}} \approx 10\Delta x$ only when $\Delta x \lessapprox 500$ m.

Our 2D AMR simulations result in grounding line positions in close agreement with the 1D uniform mesh simulations at a far lower computational cost than would be required to perform 2D uniform mesh simulations. Figure 8a also plots the (constant in y) distance of the grounding line from $x = 0$ computed in these simulations as a function of the finest resolution $(\Delta x^L)^{-1}$: they are no further than Δx^L from the positions attained in the uniform mesh flow-line calculations. Table 2 shows the valid and total cell counts on the final meshes of each of the Schoof-Hindmarsh calculations, which increase at a rate roughly proportional to their finest resolution $(\Delta x^L)^{-1}$; and the cell counts on a mesh with a uniform resolution Δx^L, which are of course proportional to $(\Delta x^L)^{-2}$. At the finest resolution, where $\Delta x^L = 195$ m, the uniform mesh problem would have more than one hundred times as many valid cells, and hence degrees of freedom, as the adaptive mesh. Storage requirements and calculation time grow with the total cell count rather than the valid cell count, but this is still eighty times larger for the uniform mesh.

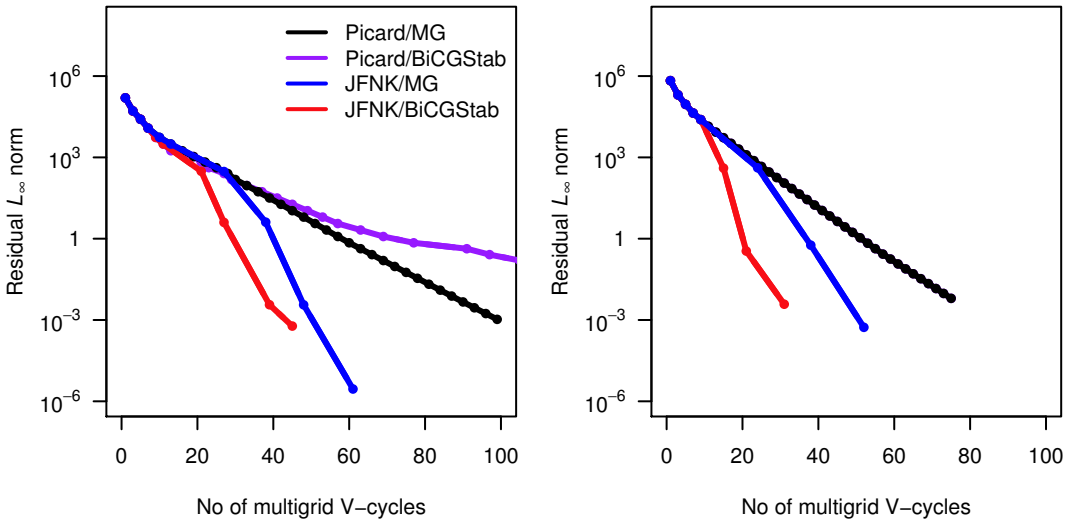
L	Δx^L	valid cell count	total cell count	uniform mesh cell count
0	6.25 km	2,048	2,048	2,048
1	3.12 km	2,624	2,816	8,192
2	1.56 km	3,776	4,352	32,768
3	781 m	6,468	8,488	131,072
4	391 m	13,568	17,408	524,288
5	195 m	20,096	26,112	2,097,152

Table 2: Cell counts on the final adaptive meshes of the Schoof-Hindmarsh simulations. The valid cell counts include only the valid regions $\Omega^\ell_{\text{valid}}$, while the total cell count also includes $\Omega^\ell_{\text{invalid}}$ and is thus a better indicator of computational cost. The uniform mesh cell count is simply that of a uniform mesh with spacing Δx^L.



(a) Bedrock elevation, $r(x, y)$ (m) (b) Observed upper surface speed $|\vec{v}_s|$, (m a^{-1})

Figure 9: Pine Island Glacier, West Antarctica. (a) Bedrock topography with the initial grounding line Γ_{gl} and calving front Γ_{cf} marked. Ice flows along the deep channel, marked by Γ_{fl} and its branches. The channel floor rises toward the calving front and attains its maximum beyond the present day grounding line. (b) Observed upper surface speed $|\vec{v}_s|$, (m a^{-1}) at the upper surface of the ice [50].



(a) $L = 1, \Delta x^L = 2$ km (b) $L = 4, \Delta x^L = 250$ m

Figure 10: Performance of the Picard and JFNK solvers at the start of the Pine Island Glacier calculations. The outer residual norm $||L[\mathbf{v}^k]\mathbf{v}^k - \boldsymbol{r}||_\infty$ is reduced at similar rates in a low resolution calculation (a) and in a high resolution calculation (b). We plot the number of multigrid V-cycles taken on the x-axis (rather than the outer iteration number k) as this operation occupies most of the program's time. Results are shown for both Picard and JFNK methods, each with simple multigrid iterations and with multigrid preconditioned BiCGStab iterations used to solve the inner linear systems. The JFNK method begins with a few Picard iterations, so the curves are identical to begin with, but the JFNK methods are ultimately quicker. The Picard/MG and Picard/BiCGStab results are identical in (b).

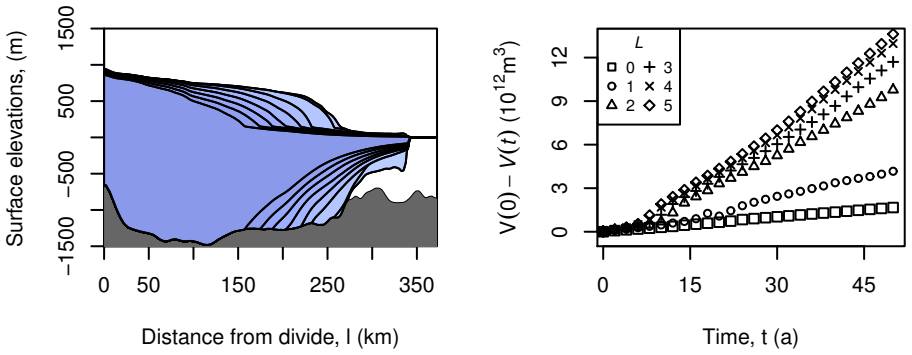
4.3. Grounding line migration at Pine Island Glacier

We considered grounding line retreat due to ice shelf melting at Pine Island Glacier (PIG), West Antarctica as a typical application of the model. We chose a 256 km × 384 km region surrounding PIG; its bedrock topography is shown in figure 9(a) together with the present day grounding line and calving front. The topography, and the ice thickness set at the start of our simulation, are derived from a number of sources including airborne survey [51], satellite observation [52, 53] and submarine exploration [16, 54], and then interpolated onto a 1 km grid following the approach of [55]. Recent measurements show ice streams draining the region (figure 9(b)), converging in the deep central channel and crossing the grounding line at speeds of around 2 km a^{-1}, to be lost to the ocean at a rate of approximately 70 $\text{km}^3\,\text{a}^{-1}$ [12, 18].

A series of simulations were carried out using the Schoof-Hindmarsh model and varying the finest mesh spacing Δx^L from 4 km down to 125 m. Much as before, meshes were generated at each timestep by selecting 4 cells for refinement on either side of the grounding line at every level ℓ, though cells close to the domain boundaries were not chosen. In each calculation the flow law rate factor was $A = 4.0 \times 10^{-17}$ $\text{Pa}^{-3}\,\text{a}^{-1}$, the exponents in the flow and friction laws were $n = 3$ and $m = 1$ respectively, and a uniform rate of accumulation $M_s = 300$ mm a^{-1} was imposed. The friction law coefficient C varies across the domain and was chosen so that the initial velocity roughly matched the observations of [50] shown in figure 9(b) : it is lowest, with values of around 50 Pa m^{-1}a, in the central channel and up to 10^5 Pa m^{-1}a elsewhere[3]. Boundary conditions at all four of the domain edges are given by the divide condition (15), while the calving front Γ_{cf} is kept constant throughout.

Before we turn to time-dependent calculations, we note that the performance of our multigrid Picard and JFNK solvers remains adequate for this much more realistic problem. We will consider just the initial velocity calculation, since performance is similar at subsequent time steps. Figure 10 plots the outer residual

[3]More sophisticated methods which can be used to match C to observations exist in the literature [17, 28, 29, 50, 56]



(a) Pine Island Glacier as it retreats

(b) Ice loss over time

Figure 11: (a) Sections through Pine Island Glacier as it retreats. The sections are parallel to the contour Γ_{fl} and are computed at 10 year intervals. Later results are in the foreground. (b) Cumulative volume of ice loss $V(0) - V(t)$ plotted against time for simulations with finest resolution Δx^L, $0 \leq L \leq 5$.

norm, $\left|\left|L[\mathbf{v}^k]\mathbf{v}^k - \boldsymbol{r}\right|\right|_\infty$ against the number of multigrid V-cycles taken for two cases, one with a coarsely resolved grounding line having $L = 1$ and $\Delta x^L = 2$ km and one with a finely resolved grounding line with $L = 4$ and $\Delta x^L = 250$ m. These start from the $L = 0$ and $L = 3$ solutions respectively. Just as in the idealized problem of section 4.1, solver performance does not worsen with improving resolution, and here we even see that the residual is reduced more quickly for the case $\Delta x^L = 250$ m than for $\Delta x^L = 2$ km. While the Picard method needs only one or two multigrid V-cycles per outer iteration, they reduce the outer residual slowly. In contrast, the JFNK solvers require more V-cycles per outer iteration, but the outer residual is reduced more quickly overall. There is little advantage to BiCGStab over simple multigrid iterations when using the Picard method; indeed, because it requires at least two V-cycles its overall performance is sometimes worse, as in figure 10 (a), and is sometimes identical, as in figure 10 (b). On the other hand, it can deliver significantly better JFNK performance.

If we run a simulation with just the parameters above the ice sheet tends to advance, so for it to retreat (or even remain stationary) we need to specify some additional loss of ice. We drove the glacier into retreat by specifying a simple melting rate under the floating ice shelf:

$$M_b = \begin{cases} 0 & H < 50 \\ (H - 50)/9 & 50 \leq H \leq 500 \\ 50 & H > 500 \end{cases} \text{ ma}^{-1}, \tag{46}$$

which results in the floating ice shelf thinning, eventually reaching 50 m at the calving front, so that the lateral stresses which buttress the glacier against retreat are reduced. This thinning and subsequent retreat is shown in cross-section in figure 11(a). We should point out that expression (46) was chosen because we found that it led to the kind of calculations we wished to test: rapid and significant retreat of the grounding line. It is not intended to be realistic, but does have some plausible features: melting is greatest near the grounding line and at greater depth where one might expect warmer water and a melt rate of 50 ma^{-1} is comparable to values in the literature [17, 19, 57]. However, we have no good reason to suggest that warm water ought to follow the retreating glacier, as (46) implies.

Figures 11(b) and 12 illustrate some more detail of the glacier's behavior over time and with mesh resolution. In figure 11(b), we plot the net volume of grounded ice lost through the grounding line, $V(t = 0) - V(t)$, where

$$V(t) = \int_\Omega \left\{ \begin{array}{ll} H & s = r + H \\ 0 & s > r + H \end{array} \right\} \mathrm{d}\Omega. \tag{47}$$

After an initial period of little change, the glacier retreats steadily, diminishing at a rate which varies with resolution, from as little as 35 km^3 each year when $\Delta x^L = 4$ km, to as much as 230 km^3 each year when

(a) Cell counts used to simulate 30 years of Pine Island Glacier evolution.

L	Δx^L	minimum	mean $\sum N_{\mathrm{A}}^{\ell}(t)$	maximum	$\sum N_{\mathrm{S}}^{\ell}$	uniform mesh cell count
1	2 km	8,832	9,088	9,728	9,728	24,576
2	1 km	14,208	15,849	18,560	19,968	98,304
3	500 m	26,816	30,711	36,544	48,896	393,216
4	250 m	50,240	60,205	72,576	139,712	1,572,864
5	125 m	97,344	118,156	140,480	429,440	6,291,456

(b) Cell counts used to simulate 50 years of Pine Island Glacier evolution.

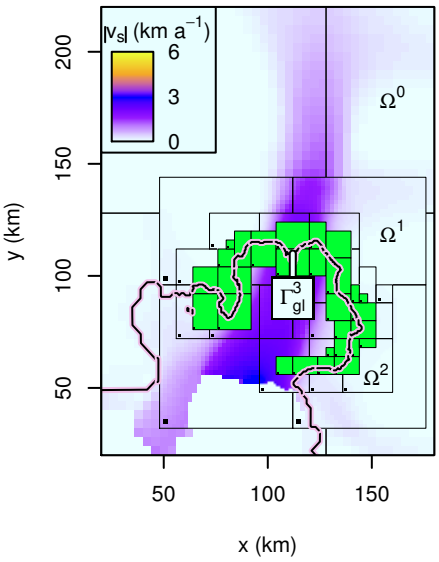
L	Δx^L	minimum	mean $\sum N_{\mathrm{A}}^{\ell}(t)$	maximum	$\sum N_{\mathrm{S}}^{\ell}$	uniform mesh cell count
1	2 km	8,832	9,338	9,728	9,728	24,576
2	1 km	14,208	17,292	20,864	23,808	98,304
3	500 m	26,816	34,522	44,800	65,472	393,216
4	250 m	50,240	68,818	91,328	203,264	1,572,864
5	125 m	97,344	136,652	184,256	678,592	6,291,456

Table 3: Cell counts used to simulate 30 years (a) and 50 years (b) of Pine Island Glacier evolution. Our adaptive meshes evolve in time, so the total cell count $\sum N_{\mathrm{A}}^{\ell}(t)$, summed over each level domain $\Omega_{\mathrm{A}}^{\ell}(t)$ varies with time; its minimum, mean, and maximum values are given above. Compare these with the cell count $\sum N_{\mathrm{S}}^{\ell}$ on a mesh which could be used for all 30 (or 50) years, defined such that each Ω_S^{ℓ} is the union of all the appropriate $\Omega_{\mathrm{A}}^{\ell}(t)$, and the cell count on a uniform grid with spacing Δx^L.

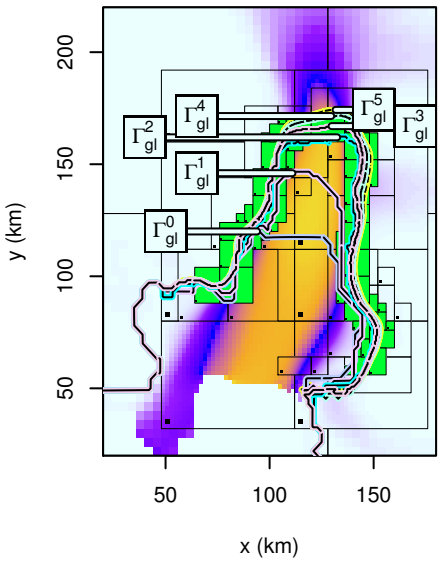
$\Delta x^L = 125$ m. Both the onset and rate of retreat converge with resolution, albeit slowly. The shape of the grounding line itself, shown after 30 years of evolution in figure 12(b) and after 50 years in figure 12(c), exhibits a similar, first-order rate of convergence, suggesting that a resolution near the grounding line of $\Delta x^L < 500$ m is necessary to achieve accuracy on the order of a few kilometers.

Evolving our meshes in time is computationally cheaper than defining a uniform mesh which is adequately resolved to treat the grounding line migration. Figures 12(a),12(b) and 12(c) show ice velocity magnitudes, at the beginning of a $\Delta x^L = 500$ m calculation, 30 years later, and 50 years later. During this period, the grounding line has retreated by up to 100 km so that the most finely resolved portions of the mesh have changed. If we denote the level domains used in the simulation by $\Omega_A^{\ell}(t)$ and the total number of cells in these as $\sum N_{\mathrm{A}}^{\ell}(t)$, then we can regard the maximum value of $\sum N_{\mathrm{A}}^{\ell}(t)$ over a given time period as a measure of the storage requirement of the calculation and the mean value as a measure of its cost in time. As in the previous example, these measures are approximately doubled for every additional level of refinement, as shown in table 3, and once again the cell count on AMR meshes with minimum mesh spacing Δx^L is dwarfed by the cell count on a uniform mesh with mesh spacing Δx^L.

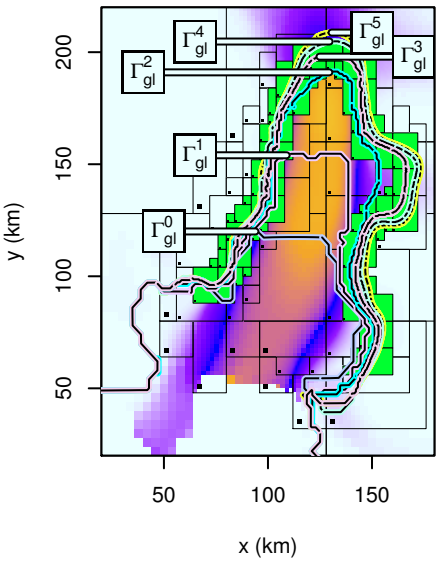
Few would consider a uniform mesh scheme to be suited to this problem, but an obvious alternative to a time evolving mesh is a non-uniform mesh that is adequately resolved in the area swept out by the grounding line. One such mesh has level domains $\Omega_{\mathrm{S}}^{\ell}$ defined to be the union of every $\Omega_{\mathrm{A}}^{\ell}(t)$ used over the appropriate time period, and although we would not be able to determine this mesh *a priori*, we can regard it as close to ideal. Since the Ω_{S}^{L} must cover the whole of the ice stream between the initial and final grounding lines, rather than a region of finite width around the grounding line, we expect the cell count $\sum N_{\mathrm{S}}^{\ell}$ to grow more quickly than the mean and maximum values of $\sum N_{\mathrm{A}}^{\ell}(t)$, and we show this in table 3 . Even if this mesh could be identified and generated at the the start of the calculations, our choice of time-evolving meshes is still more efficient. Provided that $\Delta x^L < 500$ m, $\sum N_{\mathrm{S}}^{\ell}$ is so much larger than the mean value of $\sum N_{\mathrm{A}}^{\ell}(t)$ that we can afford to generate a new mesh (recall that this means computing an additional solution to the stress-balance equations, potentially doubling the time required) at every timestep, while at $\Delta x^L > 500$ m,



(a) Pine Island Glacier at $t = 0$ a



(b) Pine Island Glacier at $t = 30$ a



(c) Pine Island Glacier at $t = 50$ a

Figure 12: Pine Island Glacier velocity magnitude $|\vec{v}|$, grounding line Γ^3_{gl} and mesh computed in the $L = 3$ simulations (with finest resolution 500 m). (a) At $t = 0$ a. (b) At $t = 30$ a and (c) $t = 50$ a , together with the grounding lines Γ^L_{gl} computed in every simulation. The rectangular patches that make up Ω^1 and Ω^2 are shown with a dark cell at the bottom left to indicate the mesh spacing. Ω^3 is shaded to distinguish it from Ω^2.

		30 years			50 years		
L	Δx^L	N_{time}	N_{mesh}	C_{mesh}	N_{time}	N_{mesh}	C_{mesh}
2	1 km	629	44	0.08	1,298	67	0.07
3	500 m	1,450	244	0.12	2,780	393	0.09
4	250 m	3,064	985	0.16	5,688	1,913	0.13
5	125 m	6,299	3,649	0.22	11,518	6,788	0.20

Table 4: The cost of evolving the mesh in time for 30 and 50 year simulations of Pine Island Glacier. As the number of levels of refinement (L) increases we need to take more time steps, N_{time}. We tag cells for refinement on every time-step, but only need a new mesh on some, for a total of N_{mesh} new meshes generated. Finer meshes need to be replaced more often, and so the proportion of CPU time spent on re-meshing operations, C_{mesh}, increases. C_{mesh} is dominated by the time spent re-solving the stress balance equations.

we can afford to do so every few timesteps.

Although we tag cells for refinement on every timestep, the resulting mesh does not necessarily change that often. Table 4 sets out the number of new meshes generated over the thirty and fifty year simulations, together with the number of time-steps taken and the proportion of CPU time spent on mesh generation and re-solving the stress balance equations. At low levels of refinement, new meshes are only rarely needed, and so the associated CPU cost is tiny. As finer meshes are brought into play, the number of new meshes needed rises rapidly, so for example at $\Delta x^L = 125$ m around a fifth of the CPU time is expended on re-meshing operations. Nonetheless, the overall efficiency grows: for the 30 year runs, at $\Delta x^L = 250$ m our scheme would seem to be around half as costly as the putative unchanging mesh, and at $\Delta x^L = 125$ m it is one third as expensive.

5. Conclusions

We have developed a block-structured AMR treatment of isothermal marine ice sheets. The resulting code [4] is built upon the Chombo[5] framework, which implements MPI-based parallel computing in addition to support for the AMR capabilities discussed in the paper. At each timestep, we must solve a nonlinear, elliptic stress balance equation to find a cell-centered velocity field . To that end, we utilize a combination of Picard and Jacobian-free Newton methods, solving the resulting linear systems with either a straightforward multigrid method, or with multigrid-preconditioned BiCGStab iterations. In turn, the cell-centered ice thickness is evolved conservatively using the piecewise parabolic method.

Our use of block structured AMR allows us to efficiently model ice sheets which feature a grounding line. Grounding line problems are particularly difficult because they exhibit degraded convergence with mesh spacing. In theory, and in our tests, both uniform mesh and AMR solutions to the stress balance equation converge at second-order rates when all the ice is grounded. On the other hand, when we include a grounding line we experience only first-order rates of convergence, and so require much higher resolution. If we evolve the mesh over time such that the mesh spacing is smallest at the grounding line but increases quickly with distance from it we can avoid much of the computational cost we might otherwise incur. We demonstrated just such a saving in both an idealized case and in a simulation of the retreat of Pine Island Glacier, in which use of AMR resulted in a more than forty-fold reduction in the number of unknowns compared to a uniform mesh, and a three to five-fold reduction compared to a mesh with fine resolution in the fast-flowing ice stream but coarse resolution elsewhere.

In future work, we will address vertical shearing (and the associated temporal stability issues), and non-isothermal ice.

[4] Our ice sheet model is known as 'BISICLES' and is freely available at http://bisicles.lbl.gov

[5] Chombo is freely available at http://chombo.lbl.gov

6. Acknowledgments

We thank Jed Brown and Daniel Goldberg for their careful and insightful reviews. Research at the Universities of Bristol and Exeter was supported financially by the UK Natural Environment Research Council. Research at LBNL was supported financially by the Office of Advanced Scientific Computing Research of the US Department of Energy under contract number DE-AC02-05CH11231. Rupert Gladstone was supported financially at the University of Bristol by the UK National Centre for Earth Observation. We thank Phil Colella and the Chombo developers at Lawrence Berkeley National Laboratory, Mark Adams of Columbia University, Vicky Lee and Gethin Williams of the University of Bristol, and Stephen Price of Los Alamos National Laboratory for the many useful discussions we have held with them.

Appendix A. Multilevel Tensor Operator Discretization

The elliptic operator we employ for the basal velocity field in (12) may be generalized as the viscous tensor operator L. L acts on a vector field to give

$$L\begin{pmatrix} \mathrm{v}_x \\ \mathrm{v}_y \end{pmatrix} = \begin{pmatrix} \alpha(x,y)\mathrm{v}_x - \partial_j(F_{xj}) \\ \alpha(x,y)\mathrm{v}_y - \partial_j(F_{yj}) \end{pmatrix}, \tag{A.1}$$

where the viscous flux term is defined by

$$F_{ij} = \mu(x,y,z)(\partial_j \mathrm{v}_i + \partial_i \mathrm{v}_j) + \delta_{ij}\upsilon(x,y,z)\partial_k \mathrm{v}_k, \tag{A.2}$$

and in our case α is the basal friction coefficient C, μ is the viscosity defined in (13), and $\upsilon = 2\mu$.

We discretize the divergence in (A.1) using the finite-volume formulation of eqn (22), which requires the flux term (A.2) at cell faces, in turn requiring partial derivatives at cell faces except at the ice divides and calving fronts, where the flux is provided directly by (15) or (18). We determine the face centered derivatives using a centered-difference stencil in the normal direction, and the average of cell-centered derivatives in the tangential direction:

$$(\partial_j \phi)^{\ell,i}_{\boldsymbol{p}} = \begin{cases} (\phi^\ell_{\boldsymbol{p}} - \phi^\ell_{\boldsymbol{p}-\mathbf{e}_i})/(\Delta x), & \text{if } i = j \\ \frac{1}{2}\Big((\partial_j\phi)^\ell_{\boldsymbol{p}} + (\partial_j\phi)^\ell_{\boldsymbol{p}-\mathbf{e}_i}\Big), & \text{if } i \neq j \end{cases} \tag{A.3}$$

and the cell-centered derivatives in the usual way,

$$(\partial_j \phi)^\ell_{\boldsymbol{p}} = (\phi^\ell_{\boldsymbol{p}+\mathbf{e}_j} - \phi^\ell_{\boldsymbol{p}-\mathbf{e}_j})/(2\Delta x). \tag{A.4}$$

At coarse-fine interfaces, we interpolate values for ϕ^ℓ into ghost cells using the quadratic interpolation described in [39]. For example, figure A.13 illustrates how ghost-cell-centered values for the tangential y-direction gradients are computed by computing an estimate for the coarse-level tangential gradient, and then quadratically interpolating in the normal direction with the fine-level gradients to compute a ghost-cell-centered value for the gradient, which is then used in (A.3).

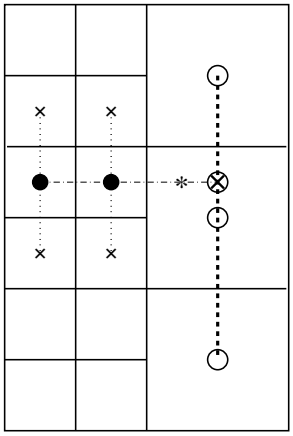


Figure A.13: Interpolating gradients tangential to coarse-fine interfaces. Compute fine-level centered gradient (using "X" values to compute gradient at filled circle). Compute interpolated coarse-level gradient at crossed circle by passing a quadratic through three coarse-level points (open circles). Then, compute tangential gradient in fine-level ghost cell (asterisk) by quadratically interpolating coarse-level and fine-level tangential gradients.

[1] D. G. Vaughan, West Antarctic Ice Sheet collapse the fall and rise of a paradigm, Climatic Change 91 (2008) 65, doi:10.1007/s10584-008-9448-3.

[2] D. I. Benn, C. R. Warren, R. H. Mottram, Calving processes and the dynamics of calving glaciers, Earth-Science Reviews 82 (2007) 143, doi:10.1016/j.earscirev.2007.02.002.

[3] E. Rignot, et al., Accelerated ice discharge from the Antarctic Peninsula following the collapse of Larsen B ice shelf, Geophysical Resarch Letters 31 (2004) L18401, doi:10.1029/2004GL020697.

[4] J. H. Mercer, West Antarctic ice sheet and CO2 greenhouse effect: a threat of disaster, Nature 271 (1978) 321, doi:10.1038/271321a0.

[5] J. L. Bamber, R. E. M. Riva, B. L. A. Vermeersen, A. M. Le Brocq, Reassessment of the Potential Sea-Level Rise from a Collapse of the West Antarctic Ice Sheet, Science 324 (2009) 901, doi:10.1126/science.1169335.

[6] C. Schoof, Ice sheet grounding line dynamics: Steady states, stability, and hysteresis, Journal of Geophysical Research 112 (2007) F03S28, doi:10.1029/2006JF000664.

[7] C. Schoof, Marine ice-sheet dynamics. Part 1. The case of rapid sliding, Journal of Fluid Mechanics 573 (2007) 27, doi:10.1017/S0022112006003570.

[8] G. Durand, et al., Full Stokes modeling of marine ice sheets: influence of the grid size, Annals of Glaciology 50 (2009) 109, doi:10.3189/172756409789624283.

[9] D. Pollard, R. M. DeConto, Modelling West Antarctic ice sheet growth and collapse through the past five million years, Nature 458 (2009) 329, doi:10.1038/nature07809.

[10] D. Goldberg, D. M. Holland, C. Schoof, Grounding line movement and ice shelf buttressing in marine ice sheets, Journal of Geophysical Research 114 (2009) F04026, doi:10.1029/2008JF001227.

[11] R. F. Katz, M. G. Worster, Stability of ice-sheet grounding lines, Proceedings of the Royal Society A 466 (2010) 1597, doi:10.1098/rspa.2009.0434.

[12] E. J. Rignot, Fast Recession of a West Antarctic Glacier, Science 281 (1998) 549, doi:10.1126/science.281.5376.549.

[13] E. Rignot, et al., Acceleration of Pine Island and Thwaites Glaciers, West Antarctica, Annals of Glaciology 34 (2002) 189, doi:10.3189/172756402781817950.

[14] R. Thomas, Accelerated Sea-Level Rise from West Antarctica, Science 306 (2004) 255, doi:10.1126/science.1099650.

[15] E. Rignot, Changes in West Antarctic ice stream dynamics observed with ALOS PALSAR data, Geophysical Research Letters 35 (2008) L12505, doi:10.1029/2008GL033365.

[16] A. Jenkins, et al., Observations beneath Pine Island Glacier in West Antarctica and implications for its retreat, Nature Geoscience 3 (2010) 468, doi:10.1038/NGEO890.

[17] I. Joughin, B. E. Smith, D. M. Holland, Sensitivity of 21st century sea level to ocean-induced thinning of Pine Island Glacier, Antarctica, Geophysical Research Letters 37, doi:10.1029/2010GL044819.

[18] A. J. Payne, A. P. Shepherd, D. J. Wingham, E. Rignot, Recent dramatic thinning of largest West Antarctic ice stream triggered by oceans, Geophysical Resarch Letters 31 (2004) L23401, doi:10.1029/2004GL021284.

[19] A. J. Payne, et al., Numerical modeling of ocean-ice interactions under Pine Island Bays ice shelf, Journal of Geophysical Research 112 (2007) C10019, doi:10.1029/2006JC003733.

[20] J. W. Park, N. Gourmelen, A. Shepherd, S. Kim, D. Vaughan, D. J. Wingham, Sustained retreat of the Pine Island Glacier through accelerated ocean melting, submitted to Geophysical Review Letters, 2011.

[21] D. K. A. Barnes, C. D. Hillenbrand, Faunal evidence for a late quaternary trans-Antarctic seaway, Global Change Biology 16 (2010) 3297, doi:10.1111/j.1365-2486.2010.02198.x.

[22] A. Vieli, A. J. Payne, Assessing the ability of numerical ice sheet models to simulate grounding line migration, Journal of Geophysical Research 110 (2005) F01003, doi:10.1029/2004JF000202.
[23] R. M. Gladstone, V. Lee, A. Vieli, A. J. Payne, Grounding line migration in an adaptive mesh ice sheet model, Journal of Geophysical Research 115, doi:10.1029/2009JF001615.
[24] F. Pattyn, A. Huyghe, S. D. Brabander, B. D. Smedt, Role of transition zones in marine ice sheet dynamics, Journal of Geophysical Research 111, doi:10.1029/2005JF000394.
[25] R. Greve, H. Blatter, Dynamics of ice sheets and glaciers, Springer-Verlag, 2009.
[26] W. Paterson, The Physics of Glaciers, Butterworth-Heinemann, third edn., 2000.
[27] C. Schoof, R. C. A. Hindmarsh, Thin-Film Flows with Wall Slip: An Asymptotic Analysis of Higher Order Glacier Flow Models, Quarterly Journal of Mechanics and Applied Mathematics 63 (2010) 73, doi:10.1093/qjmam/hbp025.
[28] M. Morlighem, et al., Spatial patterns of basal drag inferred using control methods from a full-Stokes and simpler models for Pine Island Glacier, West Antarctica, Geophysical Resarch Letters 37 (2010) L14502, doi:10.1029/2010GL043853.
[29] R. J. Arthern, G. H. Gudmundsson, Initialization of ice-sheet forecasts viewed as an inverse Robin problem, Journal of Glaciology 56 (2010) 527, doi:10.3189/002214310792447699.
[30] I. C. Rutt, M. Hagdorn, N. R. J. Hulton, A. J. Payne, The Glimmer community ice sheet model, Journal of Geophysical Research 114 (2009) F02004, doi:10.1029/2008JF001015.
[31] H. Blatter, Velocity and stress fields in grounded glaciers: a simple algorithm for including deviatoric stress gradients, Journal of Glaciology 41 (1995) 333.
[32] F. Pattyn, 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 (2003) 2382, doi:0.1029/2002JB002329.
[33] D. R. MacAyeal, Large-Scale Ice Flow Over a Viscous Basal Sediment: Theory and Application to Ice Stream B, Antarctica, Journal of Geophysical Research 94 (1989) 4071, doi:10.1029/JB094iB04p04071.
[34] E. Bueler, J. Brown, Shallow shelf approximation as a sliding law in a thermomechanically coupled ice sheet model, Journal of Geophysical Research 114, doi:10.1029/2008JF001179.
[35] D. N. Goldberg, A variationally derived, depth-integrated approximation to a higher-order glaciological flow model, Journal of Glaciology 57 (2011) 157, doi:10.3189/002214311795306763.
[36] F. Pattyn, O. Gagliardini, G. Durand, R. C. A. Hindmarsh, T. Zwinger, ice2sea MISMIP3D: Marine Ice Sheet Model Intercomparison Project for planview models (3D), URL `http://homepages.ulb.ac.be/~fpattyn/mismip3d/`, 2011.
[37] P. Colella, D. T. Graves, N. Keen, T. J. Ligocki, D. F. Martin, P. McCorquodale, D. Modiano, P. Schwartz, T. Sternberg, B. V. Straalen, Chombo Software Package for AMR Applications - Design Document, unpublished, 2000.
[38] D. F. Martin, P. Colella, D. Graves, A cell-centered adaptive projection method for the incompressible NavierStokes equations in three dimensions, Journal of Computational Physics 227 (2007) 1863, doi:10.1016/j.jcp.2007.09.032.
[39] D. F. Martin, P. Colella, A Cell-Centered Adaptive Projection Method for the Incompressible Euler Equations, Journal of Computational Physics 163 (2000) 271, doi:10.1006/jcph.2000.6575.
[40] P. N. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM Journal on Scientific and Statistical Computing 11 (1990) 450, doi:10.1137/0911026.
[41] D. A. Knoll, D. E. Keyes, Jacobian-free Newton-Krylov methods: a survey of approaches and applications, Journal of Computational Physics 193 (2004) 357, doi:10.1016/j.jcp.2003.08.010.
[42] J. F. Lemieux, et al., Implementation of the Jacobian-free Newton-Krylov method for solving the first-order ice sheet momentum balance, Journal of Computational Physics 230 (2011) 6531, doi:10.1016/j.jcp.2011.04.037.
[43] H. A. van der Vorst, Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems, SIAM Journal on Scientific and Statistical Computing 13 (1992) 631, doi:10.1137/0913035.
[44] P. Colella, P. R. Woodward, The Piecewise Parabolic Method (PPM) for gas-dynamical simulations, Journal of Computational Physics 54 (1984) 174, doi:10.1016/0021-9991(84)90143-8.
[45] M. J. Berger, P. Colella, Local Adaptive Mesh Refinement for Shock Hydrodynamics, Journal of Computational Physics 82 (1) (1989) 64–84.
[46] M. Berger, I. Rigoutsos, An algorithm for point clustering and grid generation, IEEE Transactions on Systems, Man, and Cybernetics 21 (1991) 1278, doi:10.1109/21.120081.
[47] B. van Leer, Towards the Ultimate Conservative Differences Scheme V: A Second Order Sequel to Godunov's Methods, Journal of Computational Physics 32 (1979) 101–136.
[48] J. Brown, B. Smith, A. Ahmadia, Achieving textbook multigrid efficiency for hydrostatic ice sheet flow, Submitted to SIAM J. Scientific Computing, 2011, preprint at http://59a2.org/research/publications/ .
[49] F. Pattyn, et al., Results of the Marine Ice Sheet Model Intercomparison Project, MISMIP, The Cryosphere 6 (2012) 573, doi:10.5194/tc-6-573-2012.
[50] I. Joughin, et al., Basal conditions for Pine Island and Thwaites Glaciers, West Antarctica, determined using satellite and airborne data, Journal of Glaciology 55 (2009) 245, doi:10.3189/002214309788608705.
[51] D. G. Vaughan, et al., New boundary conditions for the West Antarctic ice sheet: Subglacial topography beneath Pine Island Glacier, Geophysical Research Letters 33, doi:10.1029/2005GL025588.
[52] J. L. Bamber, J. L. Gomez-Dans, J. A. Griggs, A new 1 km digital elevation model of the Antarctic derived from combined satellite radar and laser data Part 1: Data and methods, The Cryosphere 3 (2009) 101, doi:10.5194/tc-3-101-2009.
[53] J. A. Griggs, J. L. Bamber, A new 1 km digital elevation model of Antarctica derived from combined radar and laser data Part 2: Validation and error estimates, The Cryosphere 3 (2009) 113, doi:10.5194/tc-3-113-2009.
[54] R. Timmermann, et al., A consistent data set of Antarctic ice sheet topography, cavity geometry, and global bathymetry, Earth System Science Data 2 (2010) 261, doi:10.5194/essd-2-261-2010.
[55] A. M. Le Brocq, A. J. Payne, A. Vieli, An improved Antarctic dataset for high resolution numerical ice sheet models

(ALBMAP v1), Earth System Science Data 2 (2010) 247, doi:10.5194/essd-2-247-2010.
[56] I. Joughin, D. R. MacAyeal, S. Tulaczyk, Basal shear stress of the Ross ice streams from control method inversions, Journal of Geophysical Research 109 (2004) B09405, doi:10.1029/2003JB002960.
[57] S. S. Jacobs, A. Jenkins, C. F. Giulivi, P. Dutrieux, Stronger ocean circulation and increased melting under Pine Island Glacier ice shelf, Nature Geoscience 4 (2011) 519, doi:10.1038/ngeo1188.