Title

A high order cut-cell method for solving the shallow-shelf equations

Authors

Thacher, Will
Johansen, Hans
Martin, Daniel

Publication Date

2024-08-01

DOI

10.1016/j.jocs.2024.102319

A High Order Cut-Cell Method for Solving the Shallow-Shelf Equations

Will Thacher[a,b,*], Hans Johansen[b], Daniel Martin[b]

[a] *Applied Science and Technology Group, University of California Berkeley, Berkeley, CA, 94720, United States*
[b] *Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, United States*

Abstract

In this paper we present a novel method for solving the shallow-shelf equations in the presence of grounding lines. The shallow-self equations are a two-dimensional system of nonlinear elliptic PDEs with variable coefficients that are discontinuous across the grounding line, which we treat as a sharp interface between grounded and floating ice. The grounding line is "reconstructed" from ice thickness and basal topography data to provide necessary geometric information for our cut-cell, finite volume discretization. Our discretization enforces jump conditions across the grounding line and achieves high-order accuracy using stencils constructed with a weighted least-squares method. We demonstrate second and fourth order convergence of the velocity field, driving stress, and reconstructed geometric information.

Keywords: Shallow-Shelf Equations, Ice Sheet Model, Jump Conditions, Grounding Line, Cut cell, Embedded Boundary

1. Introduction

Marine-terminating ice sheets exhibit complex behavior in the grounding zone, the region where seaward-flowing ice transitions from being grounded (in contact with bedrock) to floating in the ocean. This region is the focus of much glaciological research because the flux of ice through the grounding line can have a large impact on the total contribution of the changing ice sheet to global sea level rise [1]. From a mathematical perspective, this abrupt change may violate the underlying smoothness assumptions of the discretization scheme. This challenge has been dealt with in various ways including increased/adaptive resolution, an internal boundary condition, and sub-grid interpolation of basal friction and driving stress (see [2, 3]). In this work we propose a novel approach to this problem: we treat the grounding line as a sharp interface between grounded and floating ice, which are each represented as distinct fluids coupled by jump conditions at the grounding line. We use these jump conditions to create modified stencils near the grounding line, but away from the grounding line, in the bulk of the ice sheet or shelf, the discretization is unaffected.

The majority of the computational effort in an ice sheet model is devoted to solving a stress-balance equation, which relates the unknown velocity of the ice to the instantaneous ice thickness and basal topography

*Corresponding author: wthacher@lbl.gov

based on the bedrock elevation map. Many ice sheet models use an approximation to the three-dimensional equations for Stokes flow to reduce the stress-balance equation to a two-dimensional system of nonlinear elliptic PDEs [4]. By considering the grounding line to be an interface, we can solve this problem using methods for solving elliptic PDEs with discontinuous coefficients and source term across an interface. Numerous schemes have been proposed to solve this problem based on finite difference, finite volume, and finite element formulations. See [5] for a thorough review and list of references. There are many different finite element approaches that support sharp interfaces for elliptic problems, including XFEM [6] and CutFEM [7]. We only note these are based on modifications of standard conforming finite element methods and thus are not written in flux-divergence form (and in contrast to DG-FEM methods).

In general, a finite volume approach has a natural interpretation of the physical control volumes and their conservative flux balances, which hold regardless of discretization errors. For example, this is important for conservation of mass (ice) in the BISICLES model [3]. In the finite volume context, it is natural to represent the grounding line as a sharp interface, or "embedded boundary." Embedded boundary (EB) methods (also known as "cut-cell" methods) represent complex domains by intersecting a $(D-1)$-dimensional boundary with a regular D-dimensional Cartesian grid. [8], [9] have extended this method to represent the interface between fluids or materials as an EB for elliptic problems.

Advantages of these methods include discrete conservation, efficiency of generating new geometries, and the flexibility to create stable, geometry-dependent stencils. This problem is particularly well-suited for an EB method: Conservation is critical because one of the primary goals of an ice sheet model is to predict ice mass change over time. In addition, the grounding line may move in time, meaning new geometries and stencils will need to be re-calculated each time step. Although this paper focuses specifically on solving the elliptic stress-balance equation, in future research we will couple this solver to a time integrator.

This work builds off of our prior work [9], which develops a high-order EB method for solving 2D elliptic interface problems $-\beta\phi + \nabla \cdot \eta \nabla \phi = f$, where $\{\beta, \eta, f\}$ vary in space and may jump across an interface. There are several difficulties in extending that work to the present problem. First, we are dealing with a system of coupled elliptic PDEs rather than just one equation. Second, the stress-balance equation is nonlinear. Third, we do not assume that the geometric description of the interface is given; it must be generated from given physical data, and produces a higher-order accurate reconstruction for the spatial discretizations. These issues add complexity to the discretization, and significantly extend the purely linear scheme in [9].

The outline of this paper is as follows: in Section 2 we briefly introduce governing equations. In Section 3 we extend the analysis and methodology of [9] to the present problem and additionally propose an algorithm for grounding line "reconstruction." Lastly, in Section 4 we present convergence tests and analyze the results.

H	Ice thickness [m]
H_f	Ice thickness above flotation [m]
z_s	Sea level [m]
z_b	Bedrock elevation [m]
ρ	Ice density $[kg/m^3]$
ρ_w	Ocean water density $[kg/m^3]$
s	Ice surface elevation [m]
$\mathbf{u} = (u, v)$	Vertically-integrated velocity [m/s]
μ	Effective viscosity [Pa·s]
θ	Ice temperature [K]
β	Sliding coefficient $[Pa/m^2]$

Table 1: Physical variables

2. Governing Equations

In this work we use the 2D Shallow-Shelf Approximation (SSA) to the full 3D Stokes equations for large-scale ice sheet dynamics, ~~since these equations~~ which take advantage of the fact that ice sheets are dynamically thin ([10],[11],[12]). See Table 1 for physical variable definitions; bold lowercase variables represent vectors and bold uppercase variables represent matrices. We provide a quick mathematical summary here to put the solver and discretization in a physical context; for a derivation of the SSA from the Stokes equations see references above.

Ice is treated as a shear-thinning, non-Newtonian fluid that can deform or slide on top of bedrock to balance gravity-driven forces. The ice flow must satisfy a stress balance equation, which for the purpose of numerical modeling is often simplified based on scaling arguments (See [13], Chapter 6). In the SSA, the stress balance equation is integrated vertically, resulting in a 2D nonlinear, coupled system of elliptic PDEs relating the ice velocity $\mathbf{u}(x, y)$, the ice thickness $H(x, y)$:

$$-\beta(\mathbf{u})\,\mathbf{u} + \nabla \cdot (\mu(\mathbf{u})\ \ H\ \mathbf{F}(\mathbf{u})) = \rho g H \nabla s(H, z_b, z_s)\ , \tag{1}$$

where the linear part of the stress tensor $\mathbf{F}(\mathbf{u})$, is defined as:

$$\mathbf{F} = \begin{bmatrix} F_{xx} & F_{xy} \\ F_{yx} & F_{yy} \end{bmatrix} = \begin{bmatrix} 4\dfrac{\partial u}{\partial x} + 2\dfrac{\partial v}{\partial y} & \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} & 2\dfrac{\partial u}{\partial x} + 4\dfrac{\partial v}{\partial y} \end{bmatrix}\ . \tag{2}$$

There are ~~also~~ several nonlinear quantities in (1). First, the gradient of the upper surface of the ice $\nabla s(H, z_b, z_s)$ depends on the thickness H and the bedrock elevation, z_b, relative to the sea level, z_s, which together determine if the ice is resting on the bedrock ("grounded") or floating, and by how much. This is

expressed by the *thickness above flotation*,

$$H_f = H - \frac{\rho_w}{\rho}(z_s - z_b) \ , \tag{3}$$

where z_s is the sea level, and we assume $z_s = 0$ for simplicity. Note that if H_f is positive, the ice is grounded and if H_f is negative then the ice is floating. Therefore the grounding line is the zero contour of the H_f function. As a result, the upper surface of the ice s is given by:

$$s = \begin{cases} z_b + H & \text{for grounded ice: } \ H_f > 0 \\ \left(1 - \frac{\rho}{\rho_w}\right) H & \text{for floating ice: } \ H_f < 0 \end{cases} \ . \tag{4}$$

The effective viscosity $\mu(\mathbf{u})$ and friction coefficient $\beta(\mathbf{u})$ are both highly nonlinear functions of the velocity and its gradients:

$$\mu = \mu_0(\theta) \left(\dot{\epsilon}^2 + \dot{\epsilon}_0^2\right)^{\frac{1-n}{2n}} \ , \tag{5}$$

$$\dot{\epsilon}^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^2 + \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 \ , \tag{6}$$

$$\beta = C \left|u^2 + v^2 + \dot{u_0}^2\right|^{(m-1)/2} \ , \text{ for grounded ice.} \tag{7}$$

where $\dot{\epsilon}^2$ is the effective strain rate invariant, and $\mu_0(\theta)$ is a coefficient dependent on temperature (as in [14]). The sliding coefficient β follows the Weertmann sliding law, and the constant C can vary in space, but is zero for floating ice. For the simplest case of Glen's Flow law, $n = 3$ is commonly used [4]. The net result is that these exponential relationships are *negative*, that is as the velocity or shear become small, the sliding coefficient and viscosity become large, limited only by small regularization constants $\dot{\epsilon}_0^2, \dot{u_0}^2$ to prevent infinite viscosity and friction coefficient, respectively.

To understand the jump conditions at the grounding line, assume $\hat{\mathbf{n}}$ is the unit normal to it and define $\eta \equiv \mu(\mathbf{u})H$ to be the spatially-varying, nonlinear coefficient of the *linear* stress flux. Using arguments for conservation of mass and momentum across the grounding line [15], the following homogeneous jump conditions should be satisfied:

$$[\mathbf{u}] = 0 \ , \tag{8}$$

$$[\eta \mathbf{F} \cdot \hat{\mathbf{n}}] = 0 \ , \tag{9}$$

where $[\cdot]$ denotes a jump in that quantity across the grounding line. Along with problem-specific boundary conditions, these four jump conditions couple the elliptic equations (1) across the grounding line. Note that the coefficients and right-hand-side may jump at the grounding line; the sliding coefficient β is positive for grounded ice and discontinuously changes to zero for floating ice. Similarly, the piecewise definition of the upper surface of the ice means that the driving stress on the right-hand-side of (1) is also discontinuous at the grounding line. We assume ice thickness is continuous across the grounding line, meaning there are no

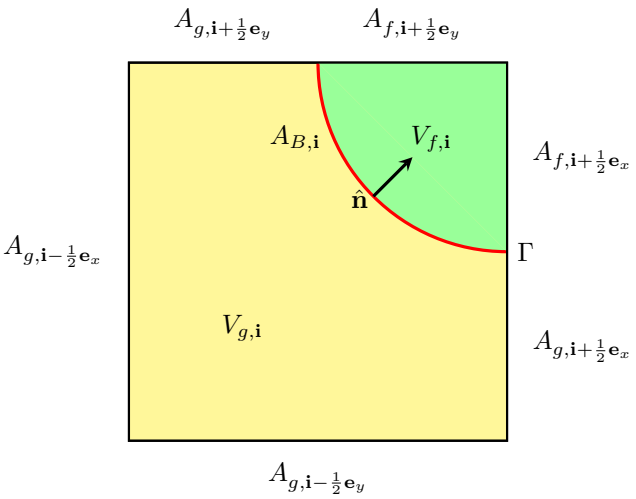


Figure 1: Cut cell geometric quantities that make up the finite volume notation.

cliffs there. However, we cannot assume that viscosity will be continuous across the grounding line because we are only enforcing continuity of $\mu H \mathbf{F} \cdot \hat{\mathbf{n}}$, so the partial derivatives of $\mathbf{u}$ may be discontinuous across the grounding line, caused by a jump in viscosity μ.

3. Finite Volume, Embedded Boundary Discretization

In this section we briefly review the notation and analysis in [9] and extend it to the present problem. Let Ω be the physical domain, which is divided into the subdomains Ω^g and Ω^f (where f, g denote a phase of floating or grounded ice, respectively) by the grounding line interface, Γ. The domain Ω is discretized into a Cartesian mesh of control volumes (or "cells") $V_{p,\mathbf{i}}$, where $\mathbf{i} \in \mathbb{Z}^2$ and $V_{p,\mathbf{i}}$ phase $p \in \{f, g\}$, and side lengths of scale h, the grid spacing (see Figure 1). Note that if these do not contain a portion of the grounding line, they are called "regular" cells in phase p on the Cartesian grid.

Any cell that is intersected by the grounding line Γ is treated as a "cut" cell containing an embedded boundary. We make the following assumptions to simplify the geometric considerations. First, a cut cell consists of only two control volumes $V_{f,\mathbf{i}}$ and $V_{g,\mathbf{i}}$ divided by a portion of the EB, denoted by $A_{B,\mathbf{i}}$, with a unit normal vector $\hat{\mathbf{n}}$ on Γ points from Ω^g to Ω^f. Cell $V_{p,\mathbf{i}}$ can have up to four grid-aligned faces, $A_{p,\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$ (where $\mathbf{e}_d$ is an index vector in direction d). So, along with these grid-aligned faces of each portion of the cut cell, each cut cell has at least 3, but at most 5, faces.

For the higher-order finite volume formulation, we define a geometric "moment," which is the integral of a centered monomial over some specified region. We use multi-index notation for this, where $\mathbf{q} = [q_x \; q_y]$ is a vector of non-negative integers, and $(\mathbf{x} - \mathbf{x}_c)^{\mathbf{q}} = (x - x_c)^{q_x}(y - y_c)^{q_y}$, where $\mathbf{x}_c$ is the local center of the moment calculation. Multi-indices are ordered lexicographically: $\{00, 10, \ldots, P0, 01, 11, \ldots, 1(P-1), 0P\}$,

which allows us to refer to $\mathbf{v}[\mathbf{q}]$ as the $\mathbf{q}^{th}$ entry of a vector $\mathbf{v}$ with maximum sum of exponents, P.

We can then define four moments corresponding to four components of the geometry:

$$m^{\mathbf{q}}_{p,\mathbf{i}} = \int_{V_{p,\mathbf{i}}} (\mathbf{x} - \mathbf{x}_c)^{\mathbf{q}}\, dV \qquad \text{“volume”} \tag{10}$$

$$m_{p,f} = \int_{A_{p,f}} (\mathbf{x} - \mathbf{x}_c)^{\mathbf{q}}\, dA\ , f = \mathbf{i}^{\mathbf{q}} \pm \frac{1}{2} e_d \qquad \text{“grid-aligned face”} \tag{11}$$

$$m^{\mathbf{q}}_{B,\mathbf{i}} = \int_{A_{B,\mathbf{i}}} (\mathbf{x} - \mathbf{x}_c)^{\mathbf{q}}\, dA \qquad \text{“boundary”} \tag{12}$$

$$m^{\mathbf{q}}_{B,\mathbf{i},d} = \int_{A_{B,\mathbf{i},d}} (\mathbf{x} - \mathbf{x}_c)^{\mathbf{q}}\ \hat{n}_d\, dA \qquad \text{“}d\text{-normal boundary”}. \tag{13}$$

Clearly, the volume of cell $\mathbf{i}$ is just $|V_{p,\mathbf{i}}| = m^{00}_{p,\mathbf{i}}$, and the centroid $\bar{\mathbf{x}}_{p,\mathbf{i}}$ of $V_{p,\mathbf{i}}$ is $[m^{10}_{p,\mathbf{i}}, m^{01}_{p,\mathbf{i}}]/|V_{p,\mathbf{i}}|$. Similarly, $m^{00}_{B,\mathbf{i}}$ is the area of the EB, and $m^{00}_{B,\mathbf{i},0}$ is its x normal component-weighted area, or x direction *cross-section*. As we will see, the overall accuracy of the scheme depends on the accuracy of these moments; lower-order moments should be very accurate, whereas higher-order ones can have lower accuracy. See §3.6 for a full description of how these moments can be calculated as a zero contour of the thickness above flotation, H_f, with convergence results in Fig. 4.

For ease of notation, throughout this paper we drop $\mathbf{x}_c$, since the polynomial interpolants based on moment equations are not significantly affected by their centering; in practice we use the cell-center of each full Cartesian cell or face.

For our discretization, variables are stored on the mesh as either cell-averaged quantities $\langle\phi\rangle_{p,\mathbf{i}} \equiv \frac{1}{|V_{p,\mathbf{i}}|}\int_{V_{p,\mathbf{i}}} \phi\ dV$, or centroid-centered (“pointwise”) quantities $\phi_{p,\mathbf{i}} \equiv \phi(\bar{\mathbf{x}}_{p,\mathbf{i}})$. For $\langle\mathbf{u}\rangle_{p,\mathbf{i}}$ in each volume $V_{p,\mathbf{i}}$ in the mesh, our finite volume system for (1), (8), and (9) becomes:

$$-\langle\beta\mathbf{u}\rangle_{p,\mathbf{i}} + \langle\nabla\cdot\eta\mathbf{F}\rangle_{p,\mathbf{i}} = \langle\rho g H\nabla s\rangle_{p,\mathbf{i}} \tag{14}$$

$$[\mathbf{u}]_{\mathbf{i}} = 0 \tag{15}$$

$$[\eta\mathbf{F}\cdot\hat{\mathbf{n}}]_{\mathbf{i}} = 0\,, \tag{16}$$

where $[\cdot]_{\mathbf{i}}$ denotes the integral of the jump of a quantity across the EB in cut cell $\mathbf{i}$:

$$[\phi]_{\mathbf{i}} = \int_{A_{B,\mathbf{i}}} \phi_g - \phi_f\, dA\ . \tag{17}$$

The system of equations we are solving has two degrees of freedom $(\langle u\rangle_{p,\mathbf{i}}, \langle v\rangle_{p,\mathbf{i}})$ in regular cells, and four degrees of freedom $(\langle u\rangle_{g,\mathbf{i}}, \langle v\rangle_{g,\mathbf{i}}, \langle u\rangle_{f,\mathbf{i}}, \langle v\rangle_{f,\mathbf{i}})$ in cut cells, separated by the grounding line. Note that so far, we have not introduced any approximations, we have simply defined the control volumes for each of the discrete variables in (14) - (16).

3.1. Taylor Series Error Analysis

The analysis in [9] showed that arbitrarily high order stencils can be generated from cell-centered Taylor expansions that are implicitly defined in terms of local solution information, such as cell averages and jump

conditions, as well as the geometry, including cut cells and curved boundaries. We now review this concept and extend it to the present problem.

Let ϕ_p be a function that is sufficiently smooth throughout phase p, but may experience a jump discontinuity at the EB. We can approximate ϕ_p with a Taylor series expansion:

$$\phi_p(\mathbf{x}) = \sum_{|\mathbf{q}|\leq P} \frac{1}{\mathbf{q}!}\phi_p^{(\mathbf{q})}(\mathbf{x}_c)(\mathbf{x}-\mathbf{x}_c)^{\mathbf{q}} + O\left(h^{P+1}\right) \tag{18}$$

where $\mathbf{x}^{\mathbf{q}} = x^{q_x}y^{q_y}$, $\mathbf{q}! = q_x!q_y!$, and $\phi_p^{(\mathbf{q})} = \frac{\partial^{q_x}\partial^{q_y}}{\partial x^{q_x}\partial y^{q_y}}\phi_p$. From this we see that the equivalent expressions for the Taylor polynomial coefficients are $c_{\phi,p}^{\mathbf{q}} = \frac{1}{\mathbf{q}!}\phi_p^{(\mathbf{q})}(\mathbf{x}_c)$.

We will now show how a local polynomial fit based on (18) can be approximated. Consider a cell-centered Taylor expansion of ϕ in cell $\mathbf{i}$, we need to interpolate local *cell-averages* of ϕ, denoted by $\langle\phi\rangle$. Note that this is in general *not* equivalent to interpolating point values of ϕ. The relationship is clearer for low-order methods, where a first-order ($P=0$) approximation for any $\langle\phi\rangle$ could be approximated by its value at $\phi(\mathbf{x}_c)$, using just the first term in (18)), or by its value at $\phi(\mathbf{x}_c = \bar{\mathbf{x}})$ for $P=1$, using both the first and second terms in (18). For higher-order approximation we use more terms in the series and, assuming $\mathbf{x}_c = 0$, we can write the average using the moment notation:

$$\langle\phi\rangle_{\mathbf{j}} = \frac{1}{|V_{\mathbf{j}}|}\int_{V_{\mathbf{j}}}\left(\sum_{|\mathbf{q}|\leq P} c_\phi^{\mathbf{q}}\mathbf{x}^{\mathbf{q}}\right) + O\left(h^{P+1}\right)dV \tag{19}$$

$$= \frac{1}{|V_{\mathbf{j}}|}\sum_{|\mathbf{q}|\leq P} m_{\mathbf{j}}^{\mathbf{q}}c_\phi^{\mathbf{q}} + O\left(h^{P+1}\right) \tag{20}$$

$$\equiv \mathbf{m}_{\mathbf{j}}^T\mathbf{c}_\phi + O\left(h^{P+1}\right)\ , \tag{21}$$

where $\mathbf{m}_{\mathbf{j}}$ is now defined as a vector of *cell-average* volume moments for cell $\mathbf{j}$, meaning the moments are divided by the (arbitrarily-small) cell volume to correspond to cell-average quantities. This is in contrast to the scalar quantity, $m_{\mathbf{j}}^{\mathbf{q}}$ which is defined in (10). From this point forward, we will use this vector notation almost exclusively to avoid implied subscripts and summations.

Suppose we now choose n arbitrary volumes in a nearby neighborhood, labeled $\mathbf{j}_1...\mathbf{j}_n$, that we will use for our interpolation of their corresponding values $\langle\phi\rangle_{\mathbf{j}}$ on each of volumes. Then we can write:

$$\begin{bmatrix}\mathbf{m}_{\mathbf{j}_1}^T\\ \mathbf{m}_{\mathbf{j}_2}^T\\ \cdots\\ \mathbf{m}_{\mathbf{j}_n}^T\end{bmatrix}\mathbf{c}_\phi \equiv \mathbf{M}_\phi\mathbf{c}_\phi\ \text{, and }\begin{bmatrix}\langle\phi\rangle_{\mathbf{j}_1}\\ \langle\phi\rangle_{\mathbf{j}_2}\\ \cdots\\ \langle\phi\rangle_{\mathbf{j}_n}\end{bmatrix} \equiv \mathbf{d}_\phi\,. \tag{22}$$

The matrix $\mathbf{M}_\phi$ is the "moment matrix" with rows made up of the vectors $\mathbf{m}_{\mathbf{j}_i}^T$. This matrix maps the coefficients of an order P polynomial the average values of that polynomials over the n volumes $\mathbf{j}_1...\mathbf{j}_n$. The

corresponding "data" vector $\mathbf{d}_\phi$ contains the actual cell-average value of ϕ in those volumes. If the moment matrix is full rank, then the Taylor series coefficients can be approximated by the least-squares solution:

$$\mathbf{c}_\phi = \mathbf{M}_\phi^\dagger \mathbf{d}_\phi \ , \tag{23}$$

where † represents the Moore-Penrose inverse, or pseudoinverse. Given this definition, we can see it is completely analogous to a Lagrange polynomial interpolation over arbitrary point values, resulting in a full-rank Vandermonde matrix system. In this case, the only difference is that we are dealing with cell-average quantities, and the cells included in the interpolation may be arbitrary in their shape and size. Given cell averages of some function, we are merely approximating a polynomial of a given order using least-squares.

If the values $\langle\phi\rangle_{\mathbf{j}_1} ... \langle\phi\rangle_{\mathbf{j}_n}$ are intended to be unknowns, then the Taylor coefficients $\mathbf{c}_\phi$ are defined implicitly as linear combinations of the unknown values. More generally, the $\mathbf{d}_\phi$ vector may also contain information about ϕ besides cell averages defined on arbitrary cell volumes; point values, boundary conditions, jump conditions, etc. can be used in the matrix system. The corresponding rows of $\mathbf{M}_\phi$ will represent the action on the Taylor polynomial that would produce that piece of data. The basic idea, detailed in [9], is that we are enforcing consistency: if ϕ is a P^{th} order polynomial, we must recover its coefficients exactly, with any additional errors coming from higher-order derivatives or errors in the geometric quantities. Throughout the paper we may add additional subscripts where necessary to indicate which phase and cell the Taylor expansion will be used in; for example the matrix $\mathbf{M}_{u,p,\mathbf{i}}$ is the moment matrix that interpolates u in phase p, centered around the cell $\mathbf{i}$.

This Taylor series formulation allows us to create stencils that approximate the terms in (14): if we write the variable coefficients β, η as well as the unknown $\mathbf{u}$ as Taylor polynomials, we can expand the terms in the stress balance equation to obtain expressions that are combinations of the Taylor series coefficients for $\mathbf{u}$. Since each Taylor series coefficient for $\mathbf{u}$ is a linear combination of unknown local cell averages of $\mathbf{u}$ in (23) and possibly other known information such as boundary and jump conditions, we can obtain an expression that is also a linear combination of local cell averages of $\mathbf{u}$, i.e., a *stencil* that acts on $\langle\mathbf{u}\rangle$. This process is shown in detail in below for each term in the stress balance equation.

We will first tackle the divergence of fluxes term, where the x-component (and similarly for the y-component) of the divergence over the (cut or regular) volume $V_{p,\mathbf{i}}$ can be written as a sum of fluxes over the grid-aligned and EB faces of the volume:

$$\langle \nabla \cdot \eta \mathbf{F}_x \rangle_{p,\mathbf{i}} = \frac{1}{|V_{p,\mathbf{i}}|} \int_V \nabla \cdot \eta \mathbf{F}_x \, dV \tag{24}$$

$$= \frac{1}{|V_{p,\mathbf{i}}|} \left(\sum_{\pm,d} \int_{A_{p,\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}} \eta \mathbf{F}_x \cdot \hat{\mathbf{n}} \, dA + \int_{A_{B,\mathbf{i}}} \eta \mathbf{F}_x \cdot \hat{\mathbf{n}} \, dA \right) . \tag{25}$$

Again, we must repeat that this has no approximations in it; it is an exact discretization given the control volume $V_{p,\mathbf{i}}$, and errors are only introduced through the approximations of fluxes.

For the x component of the flux $\mathbf{F}_x = (F_{xx}, F_{xy})$, we have for the surface integral along any face or boundary A with outward facing normal $\hat{\mathbf{n}}$:

$$\int_A \eta \mathbf{F}_x \cdot \hat{\mathbf{n}}\, dA = \int_A \eta \left[4\frac{\partial u}{\partial x} + 2\frac{\partial v}{\partial y}\,, \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right] \cdot \hat{\mathbf{n}}\, dA\,,$$

and expanding each variable in terms of its Taylor series expansion:

$$\begin{aligned}
&\int_A \eta \mathbf{F}_x \cdot \hat{\mathbf{n}}\, dA = \\
&\int_A \left(\sum_{|\mathbf{r}|\le P} c^{\mathbf{r}}_{\eta,p} \mathbf{x}^{\mathbf{r}} \right) \left(\sum_{|\mathbf{q}|\le P} \begin{bmatrix} c^{\mathbf{q}}_{u,p} \\ c^{\mathbf{q}}_{v,p} \end{bmatrix}^T \begin{bmatrix} 4q_x \mathbf{x}^{\mathbf{q}-\mathbf{e}_x} \hat{n}_x + q_y \mathbf{x}^{\mathbf{q}-\mathbf{e}_y} \hat{n}_y \\ 2q_y \mathbf{x}^{\mathbf{q}-\mathbf{e}_y} \hat{n}_x + q_x \mathbf{x}^{\mathbf{q}-\mathbf{e}_x} \hat{n}_y \end{bmatrix} \right) + O(h^P)\, dA \\
&= \sum_{|\mathbf{r}+\mathbf{q}|\le P} c^{\mathbf{r}}_{\eta,p} \begin{bmatrix} 4q_x m^{\mathbf{q}+\mathbf{r}-\mathbf{e}_x}_{A,x} + q_y m^{\mathbf{q}+\mathbf{r}-\mathbf{e}_y}_{A,y} \\ 2q_y m^{\mathbf{q}+\mathbf{r}-\mathbf{e}_y}_{A,x} + q_x m^{\mathbf{q}+\mathbf{r}-\mathbf{e}_x}_{A,y} \end{bmatrix}^T \begin{bmatrix} c^{\mathbf{q}}_{u,p} \\ c^{\mathbf{q}}_{v,p} \end{bmatrix} + O(h^{P+1}) + O(h^{R_A}) \qquad (26)
\end{aligned}$$

where $O(h^{R_A})$ is the accuracy of the area moments (11), (12), and (13), and $\mathbf{e}_x$ is the unit vector $(1,0)$, etc. Note that we have treated $\eta \equiv \mu H$ as a single variable coefficient; although μ depends on u and v, we ultimately need to build stencils for linear solvers in u and v, and μ and H can be reevaluated as needed within the nonlinear solver iterations.

Applying (26) to each surface integral for the cell averaged flux divergence term, we have a truncation error of $O(h^{R_A-2})$ $+O(h^{P-1})$ for (25) because we divide by the $O(h^2)$ volume $|V_{p,\mathbf{i}}|$. If we define $\mathbf{G}_{u,\eta,x,A}$ and $\mathbf{G}_{v,\eta,x,A}$ to be the matrices whose $\mathbf{r},\mathbf{q}$ entry are given by

$$\mathbf{G}_{u,\eta,x,A}\,[\mathbf{r},\mathbf{q}] \equiv 4q_x m^{\mathbf{q}+\mathbf{r}-\mathbf{e}_x}_{A,x} + q_y m^{\mathbf{q}+\mathbf{r}-\mathbf{e}_y}_{A,y} \qquad (27)$$

$$\mathbf{G}_{v,\eta,x,A}\,[\mathbf{r},\mathbf{q}] \equiv 2q_y m^{\mathbf{q}+\mathbf{r}-\mathbf{e}_y}_{A,x} + q_x m^{\mathbf{q}+\mathbf{r}-\mathbf{e}_x}_{A,y}\,, \qquad (28)$$

we can define the stress tensor derivative terms as operators involving the face moments, normals, and Taylor coefficients of η and $\mathbf{u}$ alone. Making the necessary substitutions, the flux component can be written as:

$$\begin{aligned}
\int_A \eta \mathbf{F}_x \cdot \hat{\mathbf{n}}\, dA &= \mathbf{c}^T_{\eta,p} \left[\mathbf{G}_{u,\eta,x,A} \mathbf{c}_{u,p} + \mathbf{G}_{v,\eta,x,A} \mathbf{c}_{v,p} \right] + O\left(h^{\min(P+1,R_A)} \right) \qquad (29) \\
&= \mathbf{d}^T_{\eta,p} \left(\mathbf{M}^\dagger_{\eta,p} \right)^T \left[\mathbf{G}_{u,\eta,x,A} \mathbf{M}^\dagger_{u,p} \mathbf{d}_{u,p} + \mathbf{G}_{v,\eta,x,A} \mathbf{M}^\dagger_{v,p} \mathbf{d}_{v,p} \right] + O\left(h^{\min(P+1,R_A)} \right)\,, \qquad (30)
\end{aligned}$$

where we have used the corresponding expressions for $\mathbf{c}_{\eta,p}$, $\mathbf{c}_{u,p}$, and $\mathbf{c}_{v,p}$ from (23). In summary, we have obtained a "bilinear" stencil for the surface integral of the flux. First, we multiply on the left by $\mathbf{d}^T_{\eta,p}$, the vector of values of η, and then on the right by $\mathbf{d}_{u,p}$ and $\mathbf{d}_{v,p}$, the vectors of cell averaged values of $\mathbf{u}$. Note that all the geometric information and averages have been collapsed into a single matrix expression that produces a single scalar quantity for the flux to be used in the divergence term. We do the same for all grid-aligned faces and boundary segments, using exactly the same formulas but with different normals and geometric moments.

For the x (or similarly y) component of the basal traction term in any grounded cell, we can define another bilinear stencil:

$$\langle \beta u \rangle_{g,\mathbf{i}} = \frac{1}{|V_{g,\mathbf{i}}|} \int_{V_{g,\mathbf{i}}} \left(\sum_{|\mathbf{r}| \le P} c_\beta^{\mathbf{r}} \mathbf{x}^{\mathbf{r}} \right) \left(\sum_{|\mathbf{q}| \le P} c_{u,g}^{\mathbf{q}} \mathbf{x}^{\mathbf{q}} \right) + O(h^{P+1}) dV \tag{31}$$

$$= \left[\frac{1}{|V_{g,\mathbf{i}}|} \sum_{|\mathbf{r}+\mathbf{q}| \le P} c_\beta^{\mathbf{r}} m_{g,\mathbf{i}}^{\mathbf{r}+\mathbf{q}} c_{u,g}^{\mathbf{q}} \right] + O\left(h^{\min(R_V - 2, P+1)} \right) \tag{32}$$

$$= \mathbf{d}_{\beta,p}^T \left(\mathbf{M}_{\beta,p}^\dagger \right)^T \mathbf{G}_{u,\beta,\mathbf{i}} \mathbf{M}_{u,p}^\dagger \mathbf{d}_{u,p} + O\left(h^{\min(R_V - 2, P+1)} \right) \tag{33}$$

where $O(h^{R_V})$ is the accuracy of the volume moments (10), and $\mathbf{G}_{u,\beta,\mathbf{i}}$ is the matrix whose $\mathbf{r}, \mathbf{q}$ entry is given by

$$\mathbf{G}_{u,\beta,\mathbf{i}}[\mathbf{r}, \mathbf{q}] \equiv \frac{m_{g,\mathbf{i}}^{\mathbf{r}+\mathbf{q}}}{V_{g,\mathbf{i}}} \ . \tag{34}$$

Again we treat β as a variable coefficient rather than a function of the Taylor expansion of u and v. Lastly, for the x-component (and likewise for the y-component) of the right-hand-side we have for grounded ice

$$\left\langle \rho g H \frac{\partial s}{\partial x} \right\rangle_{g,\mathbf{i}} = \frac{1}{|V_{g,\mathbf{i}}|} \int_{V_{g,\mathbf{i}}} \rho g H \frac{\partial (H + z_b)}{\partial x} dV \tag{35}$$

$$= \frac{1}{|V_{g,\mathbf{i}}|} \rho g \sum_{|\mathbf{r}+\mathbf{q}| \le P} c_{H,g}^{\mathbf{q}} (c_{H,g}^{\mathbf{r}} + c_{z_b,g}^{\mathbf{r}}) \ r_x m_{g,\mathbf{i}}^{\mathbf{q}+\mathbf{r}-\mathbf{e_x}} + O\left(h^{\min(P, R_V - 2)} \right) \tag{36}$$

and similarly for floating ice:

$$\left\langle \rho g H \frac{\partial s}{\partial x} \right\rangle_{f,\mathbf{i}} = \frac{1}{|V_{f,\mathbf{i}}|} \rho g \left(1 - \frac{\rho}{\rho_w} \right) \sum_{|\mathbf{q}+\mathbf{r}| \le \mathbf{P}} c_{H,f}^{\mathbf{q}} (c_{H,f}^{\mathbf{r}}) \ r_x m_{f,\mathbf{i}}^{\mathbf{q}+\mathbf{r}-\mathbf{e_x}} + O\left(h^{\min(P, R_V - 2)} \right) \tag{37}$$

Putting this all together, if we want to achieve a truncation error of order $P-1$ for the three terms in (14), we need to calculate geometric moments with errors of order $R_V, R_A = P+1$ and Taylor expansions of order P. We emphasize that this analysis did not depend in any way on the shape of the volume. However, as we will expand upon later in the paper, in cells that are away from the grounding line, due to symmetries in square-shaped volumes we can achieve an order P truncation error with an order P Taylor expansion. Since the grounding line is a codimension one smaller set, globally we expect order P truncation error in the L^1 norm from using order P Taylor expansions, due to the nature of the elliptic equations.

For linear problems, the analysis and results in [9] show that this cell-centered Taylor series formulation produces the expected level of truncation and solution error for the simpler equation $-\beta\phi + \nabla \cdot \eta \nabla \phi = f$ where β, η, f vary smoothly in space on either side of an interface but can jump across that interface. If we were to linearize the stress balance equation, the η and β fields would look like this simpler equation, and we would expect similar accuracy results in the linearized case. Because of the complexity of the full nonlinear relationships, we only empirically assess the effect of the nonlinearity on the error for numerical test problems.

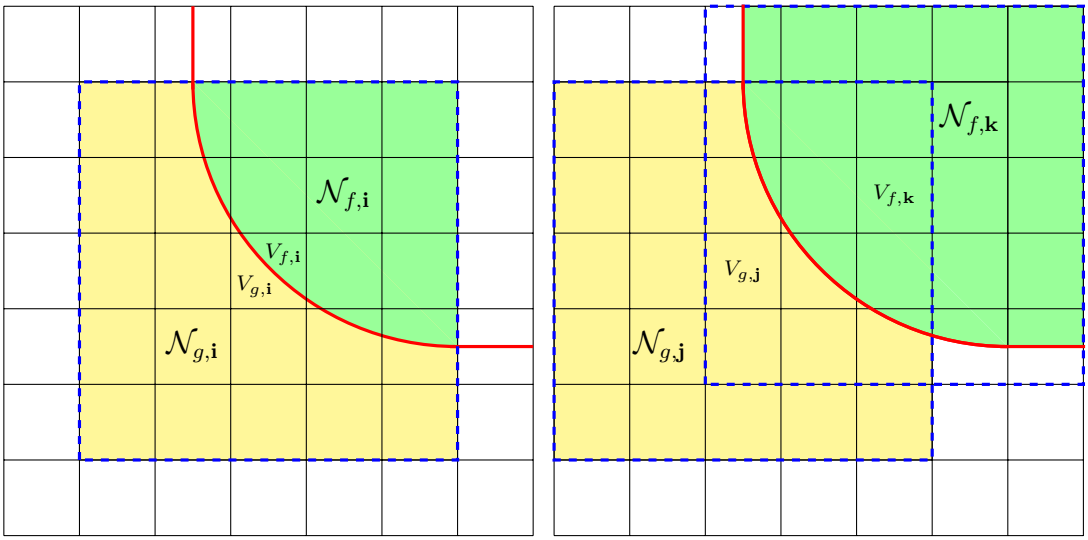


Figure 2: Figure (a) shows neighborhoods $\mathcal{N}_{p,\mathbf{i}}$ used to construct interpolation matrices for order $P = 2$, around cut cell $\mathbf{i}$, which contains two volumes $V_{p,\mathbf{i}}$ bordering the interface. Figure (b) shows neighborhoods surrounding full cells $\mathbf{j}$ and $\mathbf{k}$ that don't contain the interface, but are "irregular," meaning the regular stencil for order $P = 2$ would be inconsistent.

3.2. Stencil Construction

To complete our description of the discretization, we need to define the various moment and data matrices that appear in (30) and (33). As in [9], we partition our cells into three subsets: regular cells Ω_R, cut cells Ω_C, and irregular cells Ω_I. Regular cells use no geometric information beyond the selection of the (also regular) neighbor set, and can thus use a regular, unmodified finite volume stencil. Cut cells are intersected by the EB and must use geometric information. Irregular cells are not intersected by the EB, but at least one cell in the stencil footprint for a regular cell is intersected by the EB, thus making the regular cell stencil inaccurate.

Construction of the moment matrices is similar to what is done in [9], but there are additional complications that we must address for the stress-balance equation. First, fluxes in cut cells with jumps must couple both components *and* both phases of the velocity field. Second, the operator also contains cross-derivatives, which must be suitably discretized in regular cells. Lastly, the coefficients β, μ are functions of $\mathbf{u}$ and so must be calculated in a consistent way with other terms.

Starting with the Taylor coefficients in (23), these nonlinear coefficients are stored as point values at the *centroids* of cells, $\bar{\mathbf{x}}_\mathbf{i}$, so that the matrices $\mathbf{M}_{\eta,p,\mathbf{i}}$ and $\mathbf{M}_{\beta,p,\mathbf{i}}$ interpolate these in a neighborhood around each cell $\mathbf{i}$ (see Fig. 2). This is not done for accuracy reasons, but to make it easier to evaluate the nonlinear formulas (5)–(7); although it is possible to use cell average values with higher-order nonlinear correction terms, our ultimate goal is to create higher-order operator stencils for u, v, which are stored as cell averages.

Thus the regular and irregular cell moments are used in the matrices $\mathbf{M}_{u,p,\mathbf{i}}$ and $\mathbf{M}_{v,p,\mathbf{i}}$ to interpolate cell averages in the same neighborhood of cell $\mathbf{i}$. In cut cells that contain a portion of the EB, we additionally use the moment matrix to enforce jump conditions. Finally, in regular cells, we take advantage of symmetries to minimize the stencil footprint. We will go into the details of each of these steps next.

3.2.1. Regular Cells

The vast majority of cells will be regular and will all have the same bilinear stencil, meaning we only have to solve for this stencil once. Since regular cells are squares, the integral of any monomial error term with odd degree over a regular cell is 0. This means that for the the flux divergence term, in regular cells we can achieve a truncation error of order P using an order P polynomial. This logic only applies if the neighborhood for the moment matrix is symmetric about the center cell, similar to how an extra order of accuracy is gained from centered differences. We can achieve an order P cell averaged linear term with an order $P-2$ polynomial. See [9] for more details on these simplifications. The columns of the moment matrix $\mathbf{M}_{u,\mathbf{i}}$ correspond to all monomials with either $|\mathbf{q}| < P$ or $|\mathbf{q}| = P$ and q_x, q_y are both even. Each row contains the cell-averaged moments $\mathbf{m}_{\mathbf{j}}^T$ for each cell $\mathbf{j}$ in the stencil, and the resulting matrix is square.

For the matrix $\mathbf{M}_{\beta,\mathbf{i}}$ and $\mathbf{M}_{\eta,\mathbf{i}}$ we use the same footprint as $\mathbf{M}_{u,\mathbf{i}}$, but the entries are monomial terms evaluated at centroids rather than cell-averaged moments. Construction of the stencil for the linear term is then straightforward using (33). The flux divergence term is slightly more involved. As in [9], for each face of the cell we zero out rows and columns of $\mathbf{M}_{\eta,\mathbf{i}}$ to create the smaller matrices $\mathbf{M}_{\eta,A_{\mathbf{i}\pm e_d}}$. The neighborhoods for these matrices are symmetric about their respective faces and only contain moments that are necessary for the surface integral flux divergence term. We calculate the flux stencil for each face using (30), and average that with the stencil calculated from the neighboring cell. This symmetric averaging, along with the 0 odd moments for regular faces and volumes, eliminates several $O(h^P)$ error terms that were not accounted for in the Taylor expansions of u and v. This procedure produces a nine point stencil for $P = 2$ and a 21 point stencil for $P = 4$. The bilinear stencils are written out explicitly for $P = 2$ in the Appendix for clarity.

3.2.2. Irregular Cells

Let cell $\mathbf{i}$ be an irregular cell in phase p, meaning its regular cell footprint contains at least one cell which is intersected by the EB. Therefore we must use a more general method for construction of moment matrices for irregular cells. We note that while irregular cells are squares, we do not cancel odd order truncation error terms because our stencil are generally not symmetric about the center of the stencil. Let $\mathcal{N}_{p,\mathbf{i}}$ be a neighborhood of cells in phase p around cell $\mathbf{i}$. See Figure 2 (b). Our data vector $\mathbf{d}_{u,p,\mathbf{i}}$ will consist of cell averaged values $\langle u \rangle_{\mathbf{j}}$ for each cell $\mathbf{j} \in \mathcal{N}_{p,\mathbf{i}}$. As defined in (23), each row of the corresponding moment matrix $\mathbf{M}_{u,p,\mathbf{i}}$ will simply be $\mathbf{m}_{\mathbf{j}}^T$, the row of cell averaged volume moments for each cell $\mathbf{j}$ up to order P, and is used likewise for the v component of the velocity field. For irregular cells, the data vector $\mathbf{d}_{\beta,p,\mathbf{i}}$ will

consist of the point values of β at the centroid of each cell $\mathbf{j} \in \mathcal{N}_{p,\mathbf{i}}$, and the moment matrix $\mathbf{M}_{\beta,p,\mathbf{i}}$ will rows of monomials up to order P evaluated relative to the centroid, exactly like a Vandermonde interpolation matrix, but over-determined. The same approach is used for the η field. Then on each face of the irregular cell, we calculate the matrices $\mathbf{G}_{u,\eta,x,A}$ etc. to create a stencil for the flux integral along a face as in (30).

3.2.3. Cut Cells

In cut cells, we must enforce the jump conditions (8) and (9). Let $\mathcal{N}_\mathbf{i}$ be a neighborhood of cells about cut cell $\mathbf{i}$. See Figure 2 (a). Again, evaluating the jump in velocity u or v in terms of Taylor series and coefficients, we have in cut cell $\mathbf{j}$ in $\mathcal{N}_\mathbf{i}$:

$$\int_{A_{B,\mathbf{j}}} u_g - u_f \; dA = \mathbf{m}_{B,\mathbf{j}}^T (\mathbf{c}_{u,g} - \mathbf{c}_{u,f}) + O(h^{P+2}) \; , \tag{38}$$

where $\mathbf{m}_{B,\mathbf{j}}^T$ is the row vector of EB moments defined in (11). For the jump in flux, we have

$$\int_{A_{B,\mathbf{j}}} (\eta_g \mathbf{F}_{x,g} - \eta_f \mathbf{F}_{x,f}) \cdot \hat{n} \; dA = \\ \mathbf{c}_{\eta,g}^T \left[\mathbf{G}_{u,\eta,x,A} \mathbf{c}_{u,g} + \mathbf{G}_{v,\eta,x,A} \mathbf{c}_{v,g} \right] - \mathbf{c}_{\eta,f}^T \left[\mathbf{G}_{u,\eta,x,A} \mathbf{c}_{u,f} + \mathbf{G}_{v,\eta,x,A} \mathbf{c}_{v,f} \right] + O(h^{P+1}) \; , \tag{39}$$

with a similar expression for $\mathbf{F}_y$. Note that the flux jump condition couples the velocity components across the grounding line in one equation. The jump conditions are linear with respect to the velocity, so once $\mathbf{c}_{\eta,g}$ and $\mathbf{c}_{\eta,f}$ are approximated, these look like linear constraints on the four sets of coefficients $\mathbf{c}_{u,g}, \mathbf{c}_{v,g}, \mathbf{c}_{u,f},$ and $\mathbf{c}_{v,f}$.

To see this, let the matrix $\mathbf{M}_B$ have rows $\mathbf{m}_{B,\mathbf{j}}^T$ for each cut cell $\mathbf{j} \in \mathcal{N}_\mathbf{i}$. Similarly, let the matrix $\mathbf{M}_{\mathbf{F}_x,u,g}$, have rows $\mathbf{c}_{\eta,g} \mathbf{G}_{u,\eta,x,A}$ for each cut cell $\mathbf{j} \in \mathcal{N}_\mathbf{i}$. Again, $\mathbf{M}_p$ is the matrix that interpolates the cell averaged values of u in phase p in $\mathcal{N}_\mathbf{i}$. Then finally we can assemble these into a coupled moment matrix:

$$\mathbf{M}_{\mathbf{u},\mathbf{i}} = \begin{bmatrix} \mathbf{M}_g & 0 & 0 & 0 \\ 0 & \mathbf{M}_f & 0 & 0 \\ 0 & 0 & \mathbf{M}_g & 0 \\ 0 & 0 & 0 & \mathbf{M}_f \\ \mathbf{M}_B & -\mathbf{M}_B & 0 & 0 \\ 0 & 0 & \mathbf{M}_B & -\mathbf{M}_B \\ \mathbf{M}_{\mathbf{F}_x,u,g} & -\mathbf{M}_{\mathbf{F}_x,u,f} & \mathbf{M}_{\mathbf{F}_x,v,g} & -\mathbf{M}_{\mathbf{F}_x,v,f} \\ \mathbf{M}_{\mathbf{F}_y,u,g} & -\mathbf{M}_{\mathbf{F}_y,u,f} & \mathbf{M}_{\mathbf{F}_y,v,g} & -\mathbf{M}_{\mathbf{F}_y,v,f} \end{bmatrix} , \tag{40}$$

where the first 4 rows interpolate u_g, u_f, v_g and v_f. The next two rows enforce $[u]$ and $[v]$, respectively, while the last two are the flux jump conditions which couples all four components. Note that the jump conditions are enforced in a least-squares sense in this system as an average over the cut cell GL interface for a given

η ($= \mu(\mathbf{u})H$) field. However, because we are using a FV scheme, the resulting fluxes cancel and the jump conditions are exactly enforced in the polynomial reconstruction.

The data vector for cell averages and the jump constraints is then:

$$\mathbf{d_u} = \left[\mathbf{d}_{u,g}\ \mathbf{d}_{u,f}\ \mathbf{d}_{v,g}\ \mathbf{d}_{v,f}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\right]^T \ . \tag{41}$$

Because the jump conditions are homogeneous, the last rows in the data vector $\mathbf{d_u}$ corresponding to the jump conditions are all zeros. Thus, there is no contribution from cut cell stencils to the right-hand-side from the jump conditions. Also, we calculate the matrices $\mathbf{M}_{\eta,p,\mathbf{i}}$ only within each phase, to avoid nonlinear constraints on the coefficients, and we only need the grounded phase β matrix, $\mathbf{M}_{\beta,g,\mathbf{i}}$, because there is no basal friction term in the floating part of the cell. We note that the number of moment matrices that need to be pseudo-inverted to assemble the operator scales linearly with the number of volumes that intersect the grounding line, which is $O(n)$ for an $n \times n$ domain. The size of each moment matrix is $(4|\mathcal{N}|+4|\mathcal{N}_B|)\times 4|c_P|$ where $|\mathcal{N}|$ is the number of neighbors of a volume, $|\mathcal{N}_B|$ is the number of cut cells in the neighborhood, and $|c_P|$ is the number of polynomial coefficients up to order P in two dimensions, $\binom{P+2}{2}$. $|\mathcal{N}|$ is typically $(2P+1)^2$, so for the 4th order scheme a moment matrix might be of size 360×60. The pseudo-inverse is computed using an SVD routine. This computation is parallelized across multiple MPI processes, and it could be further parallelized if necessary using GPU operations. Although our code is not optimized, the cost of creating stencils, which includes these pseudo-inverses, is a less significant cost than solving the linearized system of equations.

Finally, if we set $\mathbf{d}_{u,p}$ to the cell averaged values of u in each phase, then our four coupled sets of velocity coefficients can be approximated by:

$$\begin{bmatrix} \mathbf{c}_{u,g} \\ \mathbf{c}_{u,f} \\ \mathbf{c}_{v,g} \\ \mathbf{c}_{v,f} \end{bmatrix} = \mathbf{M}^{\dagger}_{\mathbf{u},\mathbf{i}}\ \mathbf{d_u}\ . \tag{42}$$

This is the last piece that was needed: now that the interpolation matrices include jump conditions, grounding line geometry and partial cells, all parts of the domain are coupled and we can generate a nonlinear operator acting on $\langle \mathbf{u} \rangle_{\mathbf{i}}$ for all the terms in (14):

$$\mathbf{L}(\ \beta(\mathbf{u}),\ \mu(\mathbf{u}), H\ )\mathbf{u} = \mathbf{b}\,.$$

The matrix operator $\mathbf{L}$ is a nonlinear, $2N \times 2N$ sparse matrix, where N is the number of cell volumes. It has a typical banded structure of finite difference operators, with off-diagonal banded blocks that locally couples the two components of $\mathbf{u}$ with neighbors. The width of the bands changes depending on if it is "regular" away from the interface, with band width $P+1$, or near it with band width $2P+1$.

3.3. Neighborhood Selection, Weighting, Conservation

We choose each neighborhood of cells so that the moment matrix has more rows than columns, and the interpolation leads to a full rank, over-determined least-squares system. To avoid a prolonged search based on local geometry, we opt to make the neighborhood sufficiently large to accommodate a locally smooth geometry. For any irregular or cut cell in phase p, we let $\mathcal{N}_{p,\mathbf{i}}$ be those cells in phase p that lie in the the square of cells with side length $2P+1$ surrounding cell i. In cut and irregular cells we use a weighted least-squares approach and solve the least-squares system $\mathbf{W}\mathbf{M}_{u,\mathbf{i}}\mathbf{c}_{u,\mathbf{i}} \approx \mathbf{W}\mathbf{d}_{u,\mathbf{i}}$, where $\mathbf{W}$ is a diagonal matrix whose entries are weights which decay with distance from the center of the stencil. Let $\mathbf{x_j}$ be the centroid of a volume in the stencil. The weight corresponding to that row of the moment matrix is given by:

$$w_{\mathbf{j}} = \left(\|\mathbf{x_i} - \mathbf{x_j}\|_{\ell^s} + 1\right)^{-\alpha} \ . \tag{43}$$

Experiments have shown that using $s = 2, \alpha = P+1$ is an effective combination. This weighting is critical for controlling spectral properties of the global system that make it amenable to off-the-shelf preconditioners and iterative solvers. This technique is used in [9], [16], [17] and many other works. However, weighting does not affect accuracy. This follows from the fact that if $\mathbf{d}_{u,\mathbf{i}}$ is in the range of $\mathbf{M}_{u,\mathbf{i}}$, this system will be solved exactly regardless of the diagonal weight matrix used. In order for our discretization to be conservative, we need a single flux calculation for each face. This is accomplished by simply averaging the flux stencils from the two cells neighboring a face, and assembling the divergence stencils and other operators into a matrix coupling all the unknown velocity degrees of freedom.

3.4. Right-hand-side Discretization

We use the same moment matrix technique to calculate the right-hand-side driving stress from the thickness and topography fields. In (36) and (37), we need the Taylor coefficients of the H and z_b fields, which can be interpolated from cell-averaged values. We again expect that this will have order $P-1$ errors in cut cells and order P errors in regular cells.

3.5. Nonlinear Iteration

We use a simple Picard (fixed-point) iteration to solve the nonlinear system, meaning we repeatedly linearize the stress balance equation around the current guess for the velocity field $\mathbf{u}^k$, which determines the coefficient fields β^k, η^k, and then solve the linear system for $\mathbf{u}^{k+1}$. Algorithm 1 summarizes the algorithm details we present next.

Given a previous iteration of the velocity field $\mathbf{u}^k$, we can approximate the nonlinear coefficient fields β^k, η^k. These are functions of the values and gradients of $\mathbf{u}^k$ at cell centroids, as well values of H at cell centroids. All values and gradients are determined through interpolation using the moment matrix methodology. The moment matrix in cut cells depends on η^k through the nonlinear jump condition (39), so

Algorithm 1 Picard Iteration

Set tolerance ϵ, max iterations K
Set β^0, μ^0 to constants
Construct operator $\mathbf{L}^0$
Solve $\mathbf{L}^0\mathbf{u}^1 = \mathbf{b}$
Set $\mathbf{r}^0 = \mathbf{b} - \mathbf{L}^0\mathbf{u}^1$
while $(\epsilon \left\|r^0\right\|_{L^\infty} < \left\|\mathbf{r}^k\right\|_{L^\infty}$ AND $k < K)$ **do**
    Compute β^k, μ^k given $\mathbf{u}^k$
    Construct operator $\mathbf{L}^k(\beta^k, \mu^k, \mathbf{u}^k)$
    Solve $\mathbf{L}^k\mathbf{u}^{k+1} = \mathbf{b}$
    Set $\mathbf{r}^k = \mathbf{b} - \mathbf{L}^k\mathbf{u}^{k+1}$
end while

in cut cells we must rebuild the flux matrices $\mathbf{M_F}$, which are blocks of $\mathbf{M_u}$, each time the η field is updated. This means we must also rebuild the stencils in cut cells and neighboring irregular cells that share flux surfaces. However, in the remainder of irregular and regular cells, the bilinear stencils remain the same each iteration because they do not depend on satisfying jump conditions. Together, these stencils are assembled into a linear operator for that iteration, $\mathbf{L}^k$. The right-hand-side vector $\mathbf{b}$ does not change because it is a function of the ice thickness and basal topography, but not the velocity.

We then solve the linearized stress balance equation for a new velocity field $\mathbf{u}^{k+1}$ and repeat until convergence, as measured by the ratio of the initial residual norm to the current residual norm. The first iteration sets β^0 and η^0 to constants. The linearized equation is solved using PETSc Kryolv subspace methods and preconditioners [18]. We first multiply each row of $\mathbf{L}_k$ and $\mathbf{b}$ by the cell volume fraction (cell area divided by h^2) to reduce row-scaling issues associated with dividing by small cell volumes in the divergence operation. The linear system is non-symmetric, so we use either GMRES or BiCGStab for the Krylov method and algebraic multigrid as the preconditioner. In practice we have found it is unnecessary to fully solve the linear system each Picard iteration. It typically takes around 40 Picard iterations to reduce the residual norm by a factor of $1e10$. In future research we will pursue a geometric multigrid solver and a more sophisticated nonlinear solver such as the JFNK method, which is used successfully in time-dependent ice sheet problems [19].

3.6. Grounding Line Reconstruction Algorithm

The shape and position of the grounding line Γ must be "reconstructed" by interpolating the known values of $\langle H \rangle_{\mathbf{i}}$ and $\langle s \rangle_{\mathbf{i}}$. Therefore geometric moments are subject to error from the reconstruction, and will introduce additional numerical error into the discretization of (14) - (16).

Algorithm 2 Grounding Line Reconstruction

Set order P

Set $\langle H_f \rangle_{\mathbf{i}} = \langle H \rangle_{\mathbf{i}} + \frac{\rho_w}{\rho} \langle z_b \rangle_{\mathbf{i}}$ for each cell $\mathbf{i}$ in mesh

Compute partial derivatives of H_f on nodes

Tag the set $\mathbf{S} = \{\mathbf{i}_1, \mathbf{i}_2 ... \mathbf{i}_N\}$ of cut cells

for $(\mathbf{i}_k \in \mathbf{S})$ **do**

    Compute bi-polynomial interpolant of H_f in cell $\mathbf{i}_k$

    Compute gridline intersections

    Compute $m^{\mathbf{q}}_{p,\mathbf{i}}, m^{\mathbf{q}}_{B,\mathbf{i}}, m^{\mathbf{q}}_{B,\mathbf{i},d}$ for $|\mathbf{q}| \leq P$

end for

To complete our description of the discretization, we must calculate the geometric moments for each cell to use in the moment matrices and stencil expressions. Recall that moments are simply integrals of monomials over some region defined by the intersection of the grounding line with the Cartesian grid. For many numerical methods these integrals are computed using quadratures, either on the elements themselves or through a mapping from a reference element. However, the volumes in our method can have arbitrary shapes, so we rely on the integration technique detailed in [9]. This technique explicitly reconstructs the boundary of the volume as a polygon, and through an application of Green's theorem computes moments as integrals over the boundary of that polygon. The polygons are successively refined, and combined with an extrapolation technique, arbitrary accuracy can be achieved. One method for reconstructing the boundary, which is very natural for this problem, is to consider the boundary to be the zero level set of some implicit function, so that points on the boundary can be located by a root-finder.

The grounding line is simply the zero level set of the function $H_f = H + \frac{\rho_w}{\rho} z_b$, that is, where the "thickness above flotation" is exactly zero. Typically, cell-averaged ice thickness and basal topography data would be given by output from a previous timestep or initial conditions for an ice sheet evolution problem. The reconstructed zero level set should be 1) piecewise continuous across the grounding line and between cells, and 2) sufficiently accurate for the order of the scheme. We construct an interpolant of the H_f field in each cell whose zero level set satisfies the two necessary conditions. Geometric moments can quickly be calculated from this interpolant using the algorithm in [9] for computing monomial integrals from an implicit function description of an interface. Regarding condition 2, in the analysis section we determined that for an order P scheme, we need to calculate moments with errors of order at least $P+1$, but our reconstruction algorithm computes moments with errors of order $P+2$ so that the geometric error should not adversely affect the truncation error.

The order P algorithm proceeds as follows, and is summarized in Algorithm 2: For each node in the mesh, we calculate the value and partial derivatives of H_f using a polynomial interpolant of local cell averages of H_f

in the square of side length $P+2$ surrounding each node. Specifically, this is a "bi-polynomial" interpolant of order $P+1$, meaning a linear combination of all monomial terms $x^{q_x}y^{q_y}$ such that $q_x, q_y \leq P+1$. We expect errors of order $P+2$ for the nodal values. Now given the values of H_f on each node of the mesh, we can form the set of cut cells: a cell is intersected by the grounding line if the four values of H_f on the nodes of that cell all do not have the same sign.

For each cut cell $\mathbf{i}$, we use the values and partial derivatives of H_f on the four nodes of the cell to create an order $P+1$ bi-polynomial interpolant whose zero level set represents the portion of the grounding line in cell $\mathbf{i}$. In each coordinate direction, the interpolant is an order $P+1$ polynomial, so we expect errors in the location of the level set to be of order $P+2$. This construction ensures continuity of the grounding line because along the gridlines between neighboring cells the interpolant is defined uniquely by the shared nodal values and derivatives, which define the grid line intersections. For an order $\mathbf{p}$ area moment (defined in (11), (13), (12)) we expect order $|\mathbf{p}|+P+2$ error, and for an order $\mathbf{p}$ volume moment (defined in (10)) we expect an order $|\mathbf{p}|+P+3$ error.

3.7. Software

The method is implemented using the Chombo software library, which allows for straightforward parallelization of the algorithm [20]. As mentioned, we use the PETSc library [18] for a linear solver. The least-squares solvers in Chombo are built on top of the SVD least-squares routine in LAPACK [21].

4. Test Problems

~~We present two tests: the first verifies that the method converges at the expected order, and the second demonstrates how our method can be used on a realistic problem.~~ We present three tests: the first verifies that the method converges at the expected order, the second one examines the robustness of the method to small cells, and the third demonstrates how the method can be used on a realistic problem. For all tests, we use periodic boundary conditions in order to isolate our study of the grounding line reconstruction and modified discretization.

4.1. Numerical Convergence

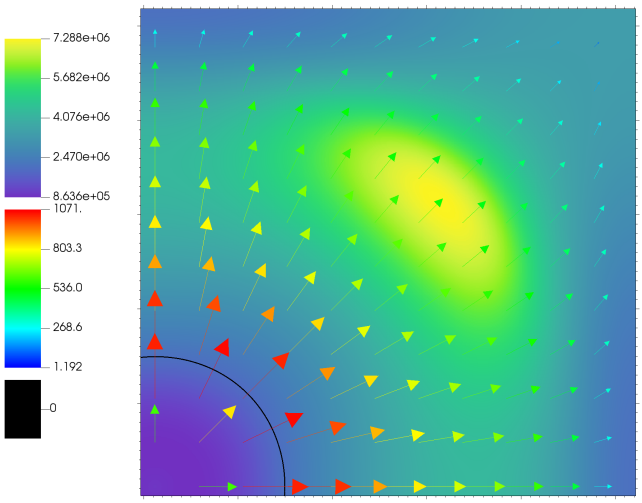
Let the domain $\Omega = [0, 130\,\text{km}]^2$. Let the function $B(x, y)$ be given by

$$B(x, y) = \cos(\omega x)^2 \cos(\omega y)^2 + c \tag{44}$$

with $\omega = \pi/130\,\text{km}, c = -5/6$. The ice thickness and topography are then given by:

$$H = 600B + 600 \tag{45}$$

$$z_b = 600B - \frac{\rho}{\rho_w} 600\ , \tag{46}$$



(a)

Figure 3: Upper right quadrant of domain for numerical convergence test. Color map of μ field with velocity field vectors scaled and colored by magnitude.

so that the grounding line is the zero level set of B. For the friction term we let $m = \frac{1}{3}$ [19] and let $C = 3000$ everywhere. We consider isothermal ice with a constant rate factor $A = 3e{-}17$. The regularization constants are $u_0^2 = 10^{-6}, \epsilon_0^2 = 10^{-12}$ The domain and grounding line are shown in Figure 3. For this test we are interested in measuring the numerical convergence of several quantities: the geometric moments from the grounding line reconstruction, the right-hand-side driving stress, and the velocity field. We use values computed at the finest level, $n = 1024$, as an "exact" solution. To demonstrate high-order accuracy, we run the convergence tests with both $P = 2$ and $P = 4$. We measure the error $\langle e \rangle_{\mathbf{i}}$ as discrete cell averages, and evaluate it using discrete ℓ^p norms:

$$\|e\|_1 = \frac{1}{|\Omega|} \sum_{p,\mathbf{i}} \left| \langle e \rangle_{p,\mathbf{i}} \right| |V_{p,\mathbf{i}}| \tag{47}$$

$$\|e\|_\infty = \max_{p,\mathbf{i}} \left| \langle e \rangle_{p,\mathbf{i}} \right| \ , \tag{48}$$

where $|\Omega|$ is the volume of the domain. Convergence results are shown in Figure 4. We are primarily interested in the rate of convergence of the solution error in the velocity field, as that is what we are solving for. Our analysis showed that there are several interconnected factors contributing to the truncation error: geometric information, right-hand-side discretization, and operator discretization. We saw that in order to achieve an order $P-1$ truncation error in regular cells and order P truncation error in cut and irregular cells, we required 1) order P polynomial interpolants of the $\mathbf{u}, \beta, \mu, H, z_b$ fields and 2) order $P+1$ geometric moments. Firstly, we confirm expected behavior from the reconstruction scheme in panels (c) and (d). Since

the moments are sufficiently accurate, we expect to be able to compute order P polynomial interpolants, from which we discretize the right-hand-side driving stress. The right-hand-side converges at the required order of P. This is an indication that our operator discretization will also have the order predicted by the analysis, because it uses the same interpolation and integration process. The ℓ^∞ norm is two orders of magnitude larger than the ℓ^1 norm because the driving stress is much larger in the grounded part of the ice sheet, where the slope of the topography contributes to the surface gradient. Given these plots, we are satisfied that discretization of the geometric information and the driving stress do not interfere with the truncation error of the operator.

P, s	2,1	2,2	2,∞	4,1	4,2	4,∞
Slope	2.26	2.26	2.27	3.83	3.86	3.03

Table 2: Slopes for least-squares fit to data points at $n = 128, 256, 512$ for error in X component of velocity for order P scheme measured in the ℓ^s norm. See Fig. 4.

To obtain the velocity field we must solve a nonlinear elliptic equation, so the relationship between truncation and solution error is highly complex and we choose to analyze it empirically. We see that for all norms except the ℓ^∞ norm for the $P = 4$ scheme we obtain roughly order P convergence. A least-squares fit to the convergence rate of the error in the x component of the velocity is shown in Table 2 (results are identical for the y component due to symmetry). Specifically, the slope β_1 is calculated from fitting the line:

$$\log_2(e) \approx \beta_1 \log_2(h) + \beta_0 \tag{49}$$

to the error data points at $n = 128, 256, 512$ for 3 different ℓ^s norms. A possible explanation for the slow convergence of the ℓ^∞ norm for the $P = 4$ may be due to the following: we can see from Figure 3 that the velocity field flows radially outward from the center of the domain, meaning the both components of the velocity field should pass through zero at that point. The form of the sliding law in (7) shows that the friction coefficient is highly sensitive near the regularization constant, i.e., when the velocity is small. Errors in the velocity field are amplified by the friction coefficient, possibly leading to further errors in the velocity field. Figure 5 shows that the highest error is concentrated at the center of the domain, and at the grounding line where the ice flow is fastest. Despite this singular error at the center, we note that the $P = 4$ scheme with $n = 128$ achieves approximately the same accuracy as the $P = 2$ scheme with $n = 512$.

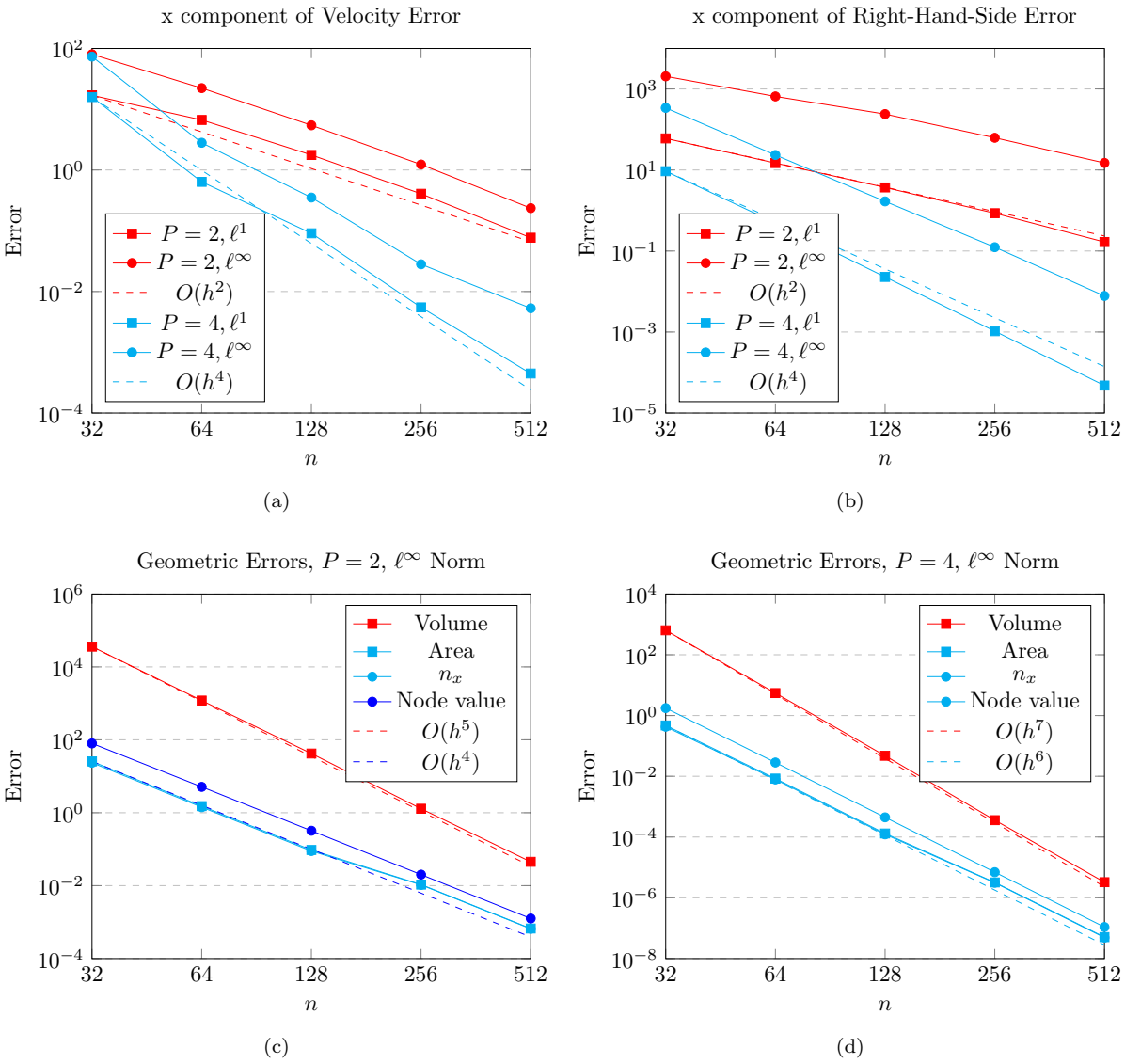


Figure 4: Errors for ice rise test. The number of cells (n) per domain side of length 130km. Figures (a) and (b) show roughly order P convergence for the x-component of the velocity field and right-hand-side. In (c) and (d) we plot the error in zeroth-order geometric moments as well as nodal values from the reconstruction.

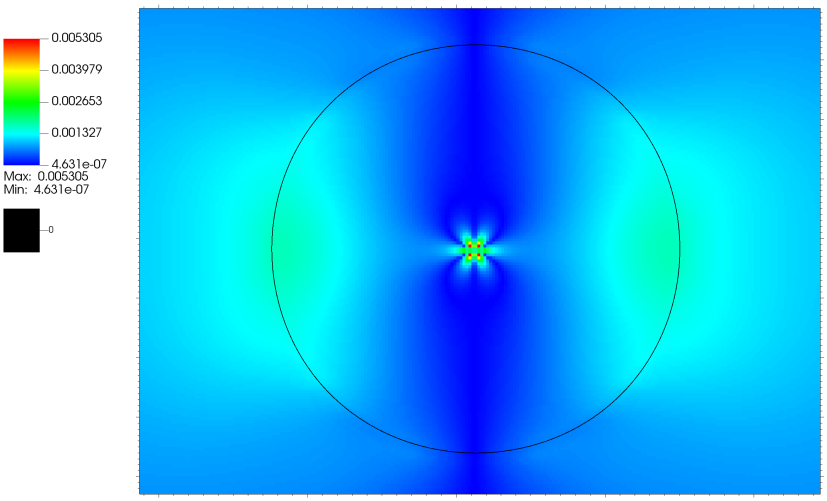


Figure 5: Absolute value of x component of velocity error for $n = 512$. The black contour is the grounding line. The error is concentrated near grounding line, where ice is flowing the fastest, and at the center of the domain where the basal sliding coefficient increases sharply.

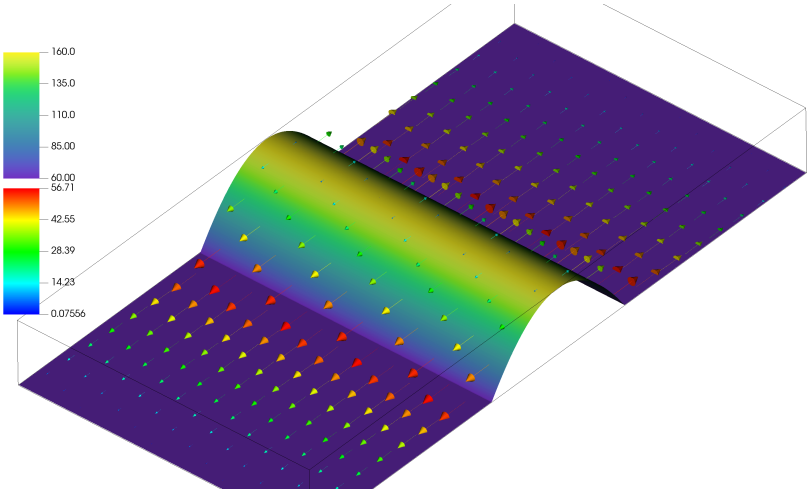


Figure 6: Domain and test problem for small volume fractions. The velocity field vectors are scaled by magnitude on the lower color bar, while the ice surface is shown with the top color bar.

4.2. Effect of small cells

We now conduct a simple experiment to demonstrate the indifference of the algorithm to small cut cell volumes, which are always present, especially with moving boundary problems. This is a known problem in the finite element context, where small and stretched elements can cause conditioning problems. Thus, it is important to establish that we can handle arbitrarily small cut cell volumes. As with the previous test,

κ	5e-1	5e-2	5e-3	5e-4	5e-5	5e-6	5e-7
P=2	1.40e3	1.62e3	1.64e3	1.52e3	1.92e3	1.99e3	2.00e3
P=4	1.13e3	1.12e3	1.28e3	1.39e3	1.41e3	1.41e3	1.41e3

Table 3: Condition number of operator with varying floating cell volume fraction κ for 1d test.

the domain ($[0, 130km]^2$) is open but with $B(x, y)$ a quadratic function in x, with roots at $32.5km + \epsilon$ and $97.5km - \epsilon$ (see Figure 6). If n is a power of 2, this means that the volume fraction κ of the floating part of the cells containing the grounding line will be $\kappa = \frac{n\epsilon}{130}$. We set $n = 32$ and vary κ, and measure the condition number of the operator. The condition number of the operator stays roughly constant even for very small volume fractions (see Table 3). This is the result of a flux-conservative discretization, where the small cell and its complement contribute to the interpolants in a way that respects the jump conditions. Although this is essentially a one-dimensional example, we see similar stability in our two-dimensional example, where volume fractions are not bounded below and can be as small as 10^{-7}.

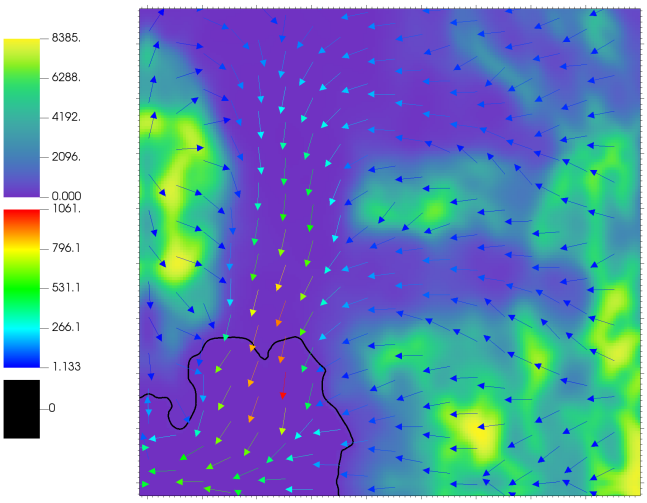
4.3. Pine Island Glacier

We next demonstrate our method on a more realistic problem. Pine Island glacier is an ice stream in West Antarctica which is closely studied and monitored by glaciologists because of its potential to contribute significantly to global sea level rise [22]. In this test we modify the Pine Island Glacier test in [19] to create a $[360\,\mathrm{km}]^2$ tile shown in Figure 7, replicated and flipped to create a 2×2 periodic domain. Basal topography and ice thickness data were obtained for 1 km grid spacing, and the basal friction coefficient was determined by solving an inverse problem [19]. We reconstruct the grounding line to partition the domain into grounded and floating regions, and linearize the equation around a constant viscocities $\mu_g = 1e7, \mu_f = 1e6$ in the respective parts of the domain.

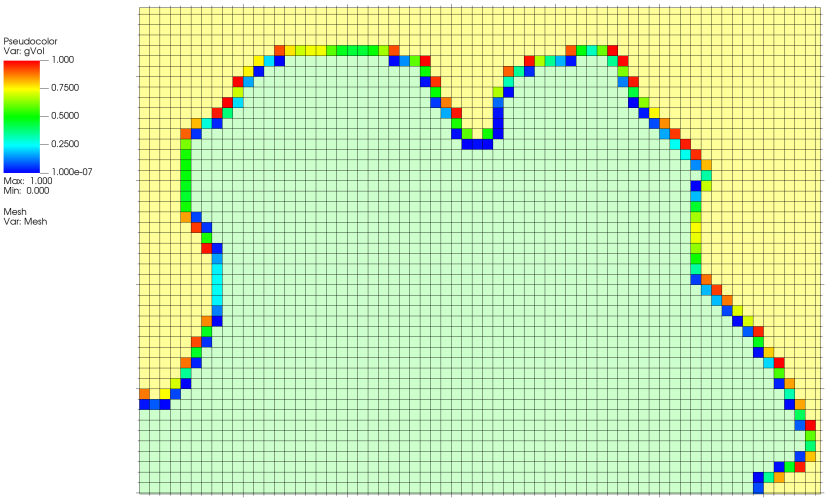
The resulting velocity field flows downhill from the grounded region to the floating area enclosed by the grounding line. In the grounded region, ice flow is fastest in the "channels" where the friction coefficient is lower. These features are typical in dynamic ice sheet calculations of the Pine Island glacier, and our algorithm handles the realistic flow and topography features without issue.

5. Conclusion

We have developed a higher-order cut-cell finite volume method for solving the shallow-shelf approximation to the stress balance equation in ice sheet dynamics, which is a two-dimensional system of nonlinear elliptic PDEs with variable coefficients that are discontinuous across the grounding line. Fourth order accuracy is achieved by extending the method developed in [9] for solving elliptic interface problems to the



(a)



(b)

Figure 7: (a) Upper right quadrant from the Pine Island glacier test. The background colormap is the basal friction coefficient field, with the velocity field vectors of unit length but colored by magnitude. (b) Reconstructed grounding line with colors indicating fraction of cell that is grounded. Note that the value varies dramatically along the interface between the grounded (light yellow) and floating (light green) ice.

shallow-shelf equations. To support the higher-order finite volume discretization, we generate a correspondingly higher-order representation of the grounding line from the thickness above flotation field.

Future research directions could include adding adaptive mesh refinement [20], which has been successful in resolving fine features critical for tracking the grounding line [19]. We would expect higher-order methods to do an even better job with additional refinement and provide lower errors at coarser resolutions. Higher-order time integrators and advection schemes could be paired with this algorithm, but careful analysis would be required for how the reconstruction and discretization would behave in a time-dependent system. In addition, computationally-expensive parts of the algorithm, such as the dense matrix pseudo-inverses and stencil calculations, might benefit from acceleration on GPUs.

References

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

[2] F. Pattyn, L. Perichon, G. Durand, L. Favier, O. Gagliardini, R. C. Hindmarsh, T. Zwinger, T. Albrecht, S. Cornford, D. Docquier, et al., Grounding-line migration in plan-view marine ice-sheet models: results of the ice2sea mismip3d intercomparison, Journal of Glaciology 59 (2013) 410–422. doi:10.3189/2013JoG12J129.

[3] S. L. Cornford, D. F. Martin, V. Lee, A. J. Payne, E. G. Ng, Adaptive mesh refinement versus subgrid friction interpolation in simulations of Antarctic ice dynamics, Annals of Glaciology 57 (2016) 1–9. doi:10.1017/aog.2016.13.

[4] K. M. Cuffey, W. S. B. Paterson, The physics of glaciers, 4 ed., Academic Press, San Diego, CA, 2010.

[5] F. Gibou, C. Min, R. Fedkiw, High resolution sharp computational methods for elliptic and parabolic problems in complex geometries, J. Sci. Comput. 54 (2013) 369–413.

[6] T.-P. Fries, T. Belytschko, The extended/generalized finite element method: An overview of the method and its applications, International Journal for Numerical Methods in Engineering 84 (2010) 253–304. doi:https://doi.org/10.1002/nme.2914.

[7] P. Hansbo, M. G. Larson, S. Zahedi, A cut finite element method for a stokes interface problem, Applied Numerical Mathematics 85 (2014) 90–114. doi:https://doi.org/10.1016/j.apnum.2014.06.009.

[8] R. Crockett, P. Colella, D. Graves, A Cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions, Journal of Computational Physics 230 (2011) 2451–2469. doi:10.1016/j.jcp.2010.12.017.

[9] W. Thacher, H. Johansen, D. Martin, A high order cartesian grid, finite volume method for elliptic interface problems, Journal of Computational Physics 491 (2023). doi:10.1016/j.jcp.2023.112351.

[10] L. W. Morland, Unconfined ice-shelf flow, in: C. J. Van der Veen, J. Oerlemans (Eds.), Dynamics of the West Antarctic Ice Sheet, Springer Netherlands, Dordrecht, 1987, pp. 99–116.

[11] J. K. Dukowicz, S. F. Price, W. H. Lipscomb, Consistent approximations and boundary conditions for ice-sheet dynamics from a principle of least action, Journal of Glaciology 56 (2010) 480–496. doi:10.3189/002214310792447851.

[12] D. R. MacAyeal, Large-scale ice flow over a viscous basal sediment: Theory and application to Ice Stream B, Antarctica, Journal of Geophysical Research: Solid Earth 94 (1989) 4071–4087. doi:10.1029/JB094iB04p04071.

[13] R. Greve, H. Blatter, Dynamics of Ice Sheets and Glaciers, 2009. doi:10.1007/978-3-642-03415-2.

[14] 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: Solid Earth 108 (2003). doi:10.1029/2002JB002329.

[15] C. Schoof, A variational approach to ice stream flow, Journal of Fluid Mechanics 556 (2006) 227–251. doi:10.1017/S0022112006009591.

[16] D. Devendran, D. Graves, H. Johansen, T. Ligocki, A fourth-order Cartesian grid embedded boundary method for Poisson's equation, Communications in Applied Mathematics and Computational Science 12 (2017) 51 – 79. doi:10.2140/camcos.2017.12.51.

[17] Y. Pan, P.-O. Persson, High-order accurate finite difference discretisations on fully unstructured dual quadrilateral meshes, Journal of Computational Physics 461 (2022) 111201. doi:/10.1016/j.jcp.2022.111201.

[18] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. C. McInnes, R. T. Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, J. Zhang, PETSc/TAO Users Manual, Technical Report ANL-21/39 - Revision 3.17, Argonne National Laboratory, 2022.

[19] S. L. Cornford, D. F. Martin, D. T. Graves, D. F. Ranken, A. M. Le Brocq, R. M. Gladstone, A. J. Payne, E. G. Ng, W. H. Lipscomb, Adaptive mesh, finite volume modeling of marine ice sheets, Journal of Computational Physics 232 (2013) 529–549. doi:10.1016/j.jcp.2012.08.037.

[20] M. Adams, P. Colella, D. Graves, J. Johnson, H. Johansen, N. Keen, T. Ligocki, D. Martin, P. McCorquodale, D. Modiano, P. Schwartz, T. Sternberg, B. V. Straalen, Chombo software package for AMR applications: design document, Technical Report, April 2021.

[21] V. A. Barker, L. S. Blackford, J. Dongarra, J. D. Croz, S. Hammarling, M. Marinova, J. Waśniewski, P. Yalamov, LAPACK95 Users' Guide, Society for Industrial and Applied Mathematics, 2001. doi:10.1137/1.9780898718201.

[22] A. Jenkins, P. Dutrieux, S. Jacobs, S. Mcphail, J. Perrett, A. Webb, D. White, Observations beneath Pine Island Glacier in West Antarctica and implications for its retreat, Nature Geoscience 3 (2010) 468–472. doi:10.1038/ngeo890.

Funding:

This work is supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, Base Math Program, under contract number DE-AC02-05CH11231.

6. Appendix

For clarity, we include the regular cell bilinear stencil for our operator for $P = 2$. Recall the stress tensor $\mathbf{F}$ has four components which are each functions of the two components of the velocity field. Due to the fact that $F_{xx}(u, v) = F_{yy}(v, u)$ and $F_{xy} = F_{yx}$, it is only necessary to include the bilinear stencils for $\nabla \cdot (\eta[F_{xx}, F_{yy}])$ and βu. From (30) we see that our bilinear stencil involves multiplying a matrix on the left by local point values of η and on the right by local cell averaged values of u, v. We label these matrices $\mathbf{S}_{u,\eta,x}, \mathbf{S}_{v,\eta,x}$ for the flux divergence term. Let the center cell of the stencil have index $(0, 0)$. The data

vectors and stencil matrices for the flux divergence term for $P = 2$ are:

$$\mathbf{d}_\eta^T = [\eta_{0,-1}\ \eta_{-1,0}\ \eta_{0,0}\ \eta_{1,0}\ \eta_{0,1}] \tag{50}$$

$$\mathbf{d}_u^T = \left[\langle u\rangle_{-1,-1}\ \langle u\rangle_{0,-1}\ \langle u\rangle_{1,-1}\ \langle u\rangle_{-1,0}\ \langle u\rangle_{0,0}\ \langle u\rangle_{1,0}\ \langle u\rangle_{-1,1}\ \langle u\rangle_{0,1}\ \langle u\rangle_{1,1}\right] \tag{51}$$

$$\mathbf{S}_{u,\eta,x} = \begin{bmatrix} 0 & \frac{\mathbf{1}}{\mathbf{2}} & 0 & 0 & -\frac{\mathbf{1}}{\mathbf{2}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{2} & \mathbf{-2} & 0 & 0 & 0 & 0 \\ 0 & \frac{\mathbf{1}}{\mathbf{2}} & 0 & \mathbf{2} & -\frac{\mathbf{1}}{\mathbf{2}} & \mathbf{2} & 0 & \frac{\mathbf{1}}{\mathbf{2}} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{-2} & \mathbf{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{\mathbf{1}}{\mathbf{2}} & 0 & 0 & \frac{\mathbf{1}}{\mathbf{2}} & 0 \end{bmatrix} \tag{52}$$

$$\mathbf{S}_{v,\eta,x} = \begin{bmatrix} \frac{\mathbf{1}}{\mathbf{8}} & 0 & -\frac{\mathbf{1}}{\mathbf{8}} & \frac{\mathbf{1}}{\mathbf{8}} & 0 & -\frac{\mathbf{1}}{\mathbf{8}} & 0 & 0 & 0 \\ \frac{\mathbf{1}}{\mathbf{4}} & \frac{\mathbf{1}}{\mathbf{4}} & 0 & 0 & 0 & 0 & -\frac{\mathbf{1}}{\mathbf{4}} & -\frac{\mathbf{1}}{\mathbf{4}} & 0 \\ \frac{\mathbf{3}}{\mathbf{8}} & 0 & -\frac{\mathbf{3}}{\mathbf{8}} & 0 & 0 & 0 & -\frac{\mathbf{3}}{\mathbf{8}} & 0 & \frac{\mathbf{3}}{\mathbf{8}} \\ 0 & -\frac{\mathbf{1}}{\mathbf{4}} & -\frac{\mathbf{1}}{\mathbf{4}} & 0 & 0 & 0 & 0 & \frac{\mathbf{1}}{\mathbf{4}} & \frac{\mathbf{1}}{\mathbf{4}} \\ 0 & 0 & 0 & -\frac{\mathbf{1}}{\mathbf{8}} & 0 & \frac{\mathbf{1}}{\mathbf{8}} & -\frac{\mathbf{1}}{\mathbf{8}} & 0 & \frac{\mathbf{1}}{\mathbf{8}} \end{bmatrix} \tag{53}$$

and $\mathbf{d}_v$ is defined analogously to $\mathbf{d}_u$. Then we finally have:

$$\langle \nabla \cdot (\eta[F_{xx}, F_{yy}])\rangle = \frac{1}{h^2}\mathbf{d}_\eta^T \left(\mathbf{S}_{u,\eta,x}\mathbf{d}_u + \mathbf{S}_{v,\eta,x}\mathbf{d}_v\right) + O(h^2) \tag{54}$$

For the friction term we simply have:

$$\langle \beta u\rangle = \beta_{0,0}\langle u\rangle_{0,0} + O(h^2) \tag{55}$$

We can similarly define stencils for $P = 4$ with a larger (21 point) footprint for the flux divergence term and a 5 point footprint for the friction term.