Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2011.m1031

Vol. 4, No. 3, pp. 396-418
August 2011

Coupled Models and Parallel Simulations for Three-Dimensional Full-Stokes Ice Sheet Modeling

Huai Zhang[1,4], Lili Ju[1,*], Max Gunzburger[2], Todd Ringler[3] and Stephen Price[3]

[1] *Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA.*

[2] *Department of Scientific Computing, Florida State University, Tallahassee, FL 32306, USA.*

[3] *Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA.*

[4] *Laboratory of Computational Geodynamics, Graduate University of Chinese Academy of Sciences, Beijing, 100049, China.*

Received 25 September 2010; Accepted (in revised version) 22 December 2010

Available online 20 June 2011

Abstract. A three-dimensional full-Stokes computational model is considered for determining the dynamics, temperature, and thickness of ice sheets. The governing thermomechanical equations consist of the three-dimensional full-Stokes system with nonlinear rheology for the momentum, an advective-diffusion energy equation for temperature evolution, and a mass conservation equation for ice-thickness changes. Here, we discuss the variable resolution meshes, the finite element discretizations, and the parallel algorithms employed by the model components. The solvers are integrated through a well-designed coupler for the exchange of parametric data between components. The discretization utilizes high-quality, variable-resolution centroidal Voronoi Delaunay triangulation meshing and existing parallel solvers. We demonstrate the gridding technology, discretization schemes, and the efficiency and scalability of the parallel solvers through computational experiments using both simplified geometries arising from benchmark test problems and a realistic Greenland ice sheet geometry.

AMS subject classifications: 65M60, 65M55, 65Y05, 65Z05, 68U20, 68W10

Key words: Ice sheet modeling, nonlinear Stokes equation, finite element method, parallel implementation, centroial Voronoi Delaunay meshes.

1. Introduction

The computational modeling of glaciers and ice sheets has been a subject of growing interest because of the influential role they play in global sea level and climate change studies

[*]Corresponding author. *Email addresses:* ju@math.sc.edu (L. Ju), mgunzburger@fsu.edu (M. Gunzburger), hzhang@gucas.ac.cn (H. Zhang), ringler@lanl.gov (T. Ringler), sprice@lanl.gov (S. Price)

[6–8, 13]. Among the different types of approaches employed, the full three-dimensional Stokes ice sheet model is generally accepted to truly model ice sheet flows [18, 28, 41, 42]. A three-dimensional full-Stokes ice sheet computational model requires the integration of effective gridding strategies, discretization schemes, couplers for data exchange between model components, and efficient, scalable solvers.

Finite difference, finite volume, and finite element methods have proven successful for high-resolution computational simulations based on the full-Stokes model with free surface evolution [19, 40, 42]. However, the applicability of the methods for large-scale, high-resolution simulations of realistic glaciers and ice sheets remains an open question. Our approach uses finite element discretizations of the mechanical and thermal components. Based on observations from field and satellite-based studies [5, 34, 35, 37], another key component is a high-quality, adaptive, variable resolution meshing scheme that can often significantly reduce the computational cost while maintaining comparable solution accuracy relative to quasi-uniform grids. Recently, centroidal Voronoi tessellation (CVT) based mesh generation techniques have been widely incorporated into finite volume and finite element approximation schemes for convection-diffusion equations [12, 22], the Navier-Stokes equations [23], and the shallow water equations [43]; CVT-based mesh generation offers significant advantages, compared to other meshing algorithms, for improving discretization/solution accuracy and controlling local mesh sizes [11, 17, 21, 22, 24, 43].

This paper reports on progress made towards the development of an efficient, parallel, finite element solver for three-dimensional, full-Stokes ice sheet modeling. The coupled thermo-mechanical processes are numerically approximated by a scalable system using Message Passing Interface (MPI) that is computing intensive and capable of performing large amounts of data transfer during the simulation processes. We first present the governing equations, the methods we use for their discretization, and the parallel solvers we use. We then validate our numerical schemes and parallel implementations by applying them to benchmark tests having simple geometries put forward by the ice sheet modeling community. The feasibility of our parallel solvers is then demonstrated on realistic geometries by applying our package to idealized simulations of the Greenland ice sheet.

We consider the evolution of momentum, temperature, and thickness of an ice sheet having ice-atmosphere and ice-bedrock boundaries only. Moreover, we consider a simplified set of boundary conditions along those boundaries. These simplifications are in concert with the main goals of this paper, which are as follows:

- develop and test a finite element discretization of the full, three-dimensional Stokes model for ice sheet dynamics, coupled to an energy equation for the evolution of the temperature in the ice sheet and a conservation equation for the evolution of the ice sheet thickness;
- implement high-quality, variable-resolution prismatic grids based on centroidal Voronoi Delaunay triangulations;
- develop and test efficient parallel solvers for the discretized coupled system;
- demonstrate that the combined components result in a potentially powerful tool for ice sheet modeling.

2. The coupled 3D full-Stokes ice sheet model

In this section, we present the set of governing equations and associated initial and boundary conditions that define the thermo-mechanical ice sheet mathematical model we consider.

2.1. The full-Stokes equations for ice sheet dynamics

The dynamical behavior of an ice sheet is governed by the full-Stokes equations for a viscous fluid in a low Reynolds-number flow over the time interval $(0, t_{max}]$ and in the three-dimensional spatial domain Ω_t occupied by the ice sheet. Note that the ice sheet domain changes in time due to the variation of the ice thickness which can affect both the vertical and lateral extent of the ice sheet. The system is considered incompressible so that the dynamic momentum equation and the kinematic continuity equation are given by

$$\rho\frac{\partial \mathbf{u}}{\partial t} = \nabla \cdot \sigma + \rho \mathbf{g}, \quad \text{in } \Omega_t \times (0, t_{max}] \tag{2.1}$$

$$\nabla \cdot \mathbf{u} = 0, \quad \text{in } \Omega_t \times (0, t_{max}], \tag{2.2}$$

respectively, where $\mathbf{u}(\mathbf{x}, t) = (u_1, u_2, u_3)^T$ denotes the velocity, $\boldsymbol{\sigma}$ the full stress tensor, ρ the constant density of ice, and $\mathbf{g} = \left(0, 0, -|\mathbf{g}|\right)$ the gravitational acceleration. The stress tensor $\boldsymbol{\sigma}$ can be decomposed into the viscous (or deviatoric) stress $\boldsymbol{\tau}$ and the static pressure p as

$$\boldsymbol{\sigma} = \boldsymbol{\tau} - p\mathbf{I} \qquad \text{or} \qquad \sigma_{ij} = \tau_{ij} - p\delta_{ij}, \tag{2.3}$$

where δ_{ij} denotes the Krönecker tensor equal to the identity tensor $\mathbf{I}$. The convective contribution $\mathbf{u} \cdot \nabla \mathbf{u}$ has been neglected from the acceleration term due to the slow, creeping motion of the ice flow. Similarly, scaling arguments can be used to obtain that the remaining acceleration term $\rho \partial \mathbf{u}/\partial t$ is negligible, i.e., we assume that the ice flow is quasi-static. Thus, we obtain from (2.1) the *instantaneous* momentum balance equation

$$-\nabla \cdot \boldsymbol{\tau} + \nabla p = \rho \mathbf{g}, \qquad \text{in } \Omega_t \times (0, t_{max}]. \tag{2.4}$$

The strain rate tensor $\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}}$ is defined in terms of the velocity $\mathbf{u}$ by

$$\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}} = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) \qquad \text{or} \qquad \dot{\varepsilon}_{ij}(\mathbf{u}) = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{2.5}$$

The constitutive law for ice connects the viscous stress tensor $\boldsymbol{\tau}$ to the strain rate tensor $\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}}$ by the generalized form of Glen's flow law [38, 39]

$$\boldsymbol{\tau} = 2\eta_{\mathbf{u},T}\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}} \qquad \text{or} \qquad \tau_{ij} = 2\eta_{\mathbf{u},T}\dot{\varepsilon}_{ij}, \tag{2.6}$$

where

$$\eta_{\mathbf{u},T} = \frac{1}{2}A^{-1/n}\dot{\varepsilon}_e^{(1-n)/n}, \tag{2.7}$$

and $\eta_{\mathbf{u},T}$ denotes the temperature and strain rate-dependent rheology coefficient (referred to as the effective viscosity), n the power-law exponent, A the inverse deformation rate factor, and $\dot{\varepsilon}_e$ the equivalent strain rate defined as

$$\begin{aligned}\dot{\varepsilon}_e =& \sqrt{\frac{1}{2}\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}} : \dot{\boldsymbol{\varepsilon}}_{\mathbf{u}}} = \sqrt{\frac{1}{2}\sum_{i,j}^{3}\dot{\varepsilon}_{ij}^2} \\ =& \sqrt{\frac{1}{2}\left(\dot{\varepsilon}_{11}^2+\dot{\varepsilon}_{22}^2+\dot{\varepsilon}_{33}^2+2\dot{\varepsilon}_{12}^2+2\dot{\varepsilon}_{23}^2+2\dot{\varepsilon}_{31}^2\right)}.\end{aligned} \tag{2.8}$$

The parameter A depends on the temperature through an Arrhenius relation defined by

$$A = A(T) = A_0 \exp(-Q/RT), \tag{2.9}$$

where T denotes the absolute temperature measured in Kelvin, A_0 is an empirical flow law constant, Q the activation energy, and R the universal gas constant; see, e.g., [38, 42] for more details. Obviously, the constitutive coefficient (2.7) is a function of both the strain rate and the thermodynamic state.

Let Γ_t denote the outer boundary of the ice which consists of two parts: the ice-bedrock boundary Γ_{ib} which we assume to be fixed in time and the ice-atmosphere boundary $\Gamma_{ia;t}$, i.e., $\Gamma_t = \Gamma_{ib} \cup \Gamma_{ia;t}$. At the ice-bedrock boundary Γ_{ib}, we impose the no-slip boundary condition

$$\mathbf{u} = \mathbf{0}. \tag{2.10}$$

At the ice-atmosphere boundary $\Gamma_{ia;t}$, we have

$$\boldsymbol{\sigma}\cdot\mathbf{n} = -p\mathbf{n} + \boldsymbol{\tau}\cdot\mathbf{n} = -p_{atm}\mathbf{n}, \tag{2.11}$$

where p_{atm} denotes the standard atmospheric pressure. Note that, due to the quasi-static assumption about the dynamics of the ice sheet, no initial condition is required for the velocity field; the initial velocity field is fully determined from the boundary conditions, the initial temperature field, and the initial ice sheet geometry by solving the system (2.2), (2.4), (2.10), and (2.11).

2.2. Evolution of temperature

The prognostic equation describing the evolution of temperature is given by

$$\rho c\frac{\partial T}{\partial t} + \rho c\mathbf{u}\cdot\nabla T = \nabla\cdot(\kappa\nabla T) + 2\eta_{\mathbf{u},T} + \dot{\boldsymbol{\varepsilon}}_{\mathbf{u}} : \dot{\boldsymbol{\varepsilon}}_{\mathbf{u}} + q, \qquad \text{in } \Omega_t \times (0, t_{max}], \tag{2.12}$$

where $T(\mathbf{x}, t)$ denotes the temperature, c the constant heat capacity, κ the constant thermal conductivity, $q(\mathbf{x}, t)$ a volumetric internal heat source, and ρ the constant density of ice. Note that for the two tensors $\dot{\boldsymbol{\varepsilon}}$ and $\boldsymbol{\tau}$, $\dot{\boldsymbol{\varepsilon}} : \boldsymbol{\tau}$ denotes the sum of the element-wise products, i.e., $\dot{\boldsymbol{\varepsilon}} : \boldsymbol{\tau} = \sum_{i,j}^{3} \dot{\varepsilon}_{ij}\tau_{ij}$. Strain heating is included through the second to last term in (2.12).

We assume that the temperature is known at the initial time $t = 0$ at every point in the ice sheet,[†] i.e., we have, for a given function $T_0(\mathbf{x})$, that

$$T|_{t=0} = T_0(\mathbf{x}), \qquad \text{in } \Omega_0. \tag{2.13}$$

The boundary condition at the ice-sheet-atmosphere boundary is given by

$$T = T_{ia}(\mathbf{x}, t), \qquad \text{on } \Gamma_{ia;t}, \tag{2.14}$$

where T_{ia} denotes a given temperature on $\Gamma_{ia;t}$ that is generally chosen as the mean annual surface temperature, i.e., a "skin" temperature. The boundary condition at the ice-sheet-bedrock boundary is given by

$$\kappa \frac{\partial T}{\partial \mathbf{n}} = \frac{1}{\rho c} q_{ib}(\mathbf{x}, t), \qquad \text{on } \Gamma_{ib}, \tag{2.15}$$

where q_{ib} denotes a given geothermal flux on Γ_{ib}.

2.3. Evolution of ice thickness

The coupled ice sheet system also includes a prognostic equation describing the evolution of the top free surface of the ice sheet. The ice sheet domain Ω_t at a time t can be defined as

$$\Omega_t = \Big\{(x, y, z) \ \mid \ S_{ib}(x, y) \le z \le S_{ia}(x, y, t), \ \text{for } (x, y) \in \Omega_H, \, t \in [0, t_{max}]\Big\}, \tag{2.16}$$

where Ω_H denotes the horizontal extent of the ice sheet and where $S_{ia}(x, y, t)$ and $S_{ib}(x, y)$ denote the elevation of the top and fixed bottom surfaces of the ice sheet, respectively. Thus, the ice thickness is given by

$$H(x, y, t) = S_{ia}(x, y, t) - S_{ib}(x, y), \qquad \text{for } (x, y) \in \Omega_H \ \text{ and } \ t \in [0, t_{max}].$$

The prognostic equation for the evolution of the ice thickness is given by the hyperbolic equation

$$\frac{\partial H}{\partial t} + U_1 \frac{\partial H}{\partial x} + U_2 \frac{\partial H}{\partial y} - U_3 = H_{flux}, \qquad \text{for } (x, y) \in \Omega_H \ \text{ and } \ t \in (0, t_{max}], \tag{2.17}$$

along with the initial condition

$$H\Big|_{t=0} = H_0(x, y) = S_{ia}(x, y, 0) - S_{ib}(x, y), \qquad \text{for } (x, y) \in \Omega_H \tag{2.18}$$

and the obvious constraint equation

$$H(x, y, t) \ge 0, \qquad \text{for } (x, y) \in \Omega_H \ \text{ and } \ t \in [0, t_{max}], \tag{2.19}$$

[†] Assuming that the temperature at the top and bottom surfaces and the thickness distribution of the ice sheet are known at $t = 0$, the initial temperature at every location within the ice sheet is estimated through a smoothed interpolation process; see Section 7.3.

where $S_{ia}(x,y,0)$ is given, $\mathbf{U}=(U_1,U_2,U_3)^T$ denotes the vertical average of the ice sheet velocity $\mathbf{u}$ (that is determined from the full-Stokes system (2.2), (2.4), (2.10), and (2.11)), and H_{flux} is a given source/sink term, i.e., an accumulation-ablation function representing the known annual average addition or subtraction of ice at the ice-atmosphere boundary.

Note that there is A two-way coupling between the ice-thickness evolution system (2.17)–(2.19), the full-Stokes system (2.2), (2.4), (2.10), and (2.11), and the temperature evolution system (2.12)–(2.15).

3. CVDT-based prismatic meshes

Constructing a high-quality variable-resolution mesh for a three-dimensional ice sheet with realistic geometry is a critical and challenging task in computational ice sheet modeling. Our approach is to first to generate a variable-resolution triangular grid for the two-dimensional domain Ω_H that defines the horizontal extent of the ice sheet. For this purpose, we use *centroidal Voronoi tessellation* (CVT)-based meshing techniques [10, 11, 21]. CVTs are special Voronoi tessellations having the property that the *generators* of the Voronoi tessellations are also the *centroids*, with respect to a given density function, of the corresponding Voronoi cells. The dual mesh of a CVT is referred to as a *centroidal Voronoi Delaunay triangulation* (CVDT) which can then be used for finite element approximations. CVT-based meshing methods have proven to be very useful in producing high-quality grids; the local mesh size can be precisely controlled through the use of a point density function; see [21, 22] for details about CVDT construction algorithms and related properties. The point density function could be determined, as demonstrated in [43], from the ice topography and/or velocity data according to practical needs for reducing the computational costs. For example, in our Greenland ice sheet simulations, we use the inverse of the ice thickness to define a point density function that is used to generate a two-dimensional CVDT horizontal grid for Greenland. As one moves from the ice sheet interior towards the margins, the ice gets thinner but the complexity and speed of the flow also increase. Thus, increasing the mesh density as the ice gets thinner is a good zeroth-order guess for the point density function that does not bias the calculated flow field by putting too much resolution in any one place as might happen if we instead use, e.g., a map of observed velocities. The point density function can also be used to effect refinement so that boundaries are adequately resolved. In the sequel, we denote the two-dimensional CVDT of Ω_H as $\mathcal{T}_h$.

The two-dimensional CVDT grid is then transformed to a three-dimensional surface triangular grid of the top surface of the ice sheet by adding the z-coordinate obtained from the topography of the bedrock plus the ice thickness. Finally, at any time t, we produce a full three-dimensional prismatic mesh of the ice sheet domain Ω_t by mapping the surface meshes linearly along the z-direction down to the bedrock; see Fig. 1 for a sample three-dimensional prismatic mesh generated for Greenland based on realistic topography data with 5 km resolution [5], where the local mesh size of the layered CVDT grid is inversely proportional to the ice thickness. Throughout, we denote the three-dimensional prismatic grid of Ω_t as $\mathcal{P}_{h;t}$; note that because Ω_t changes in time, so does $\mathcal{P}_{h;t}$.

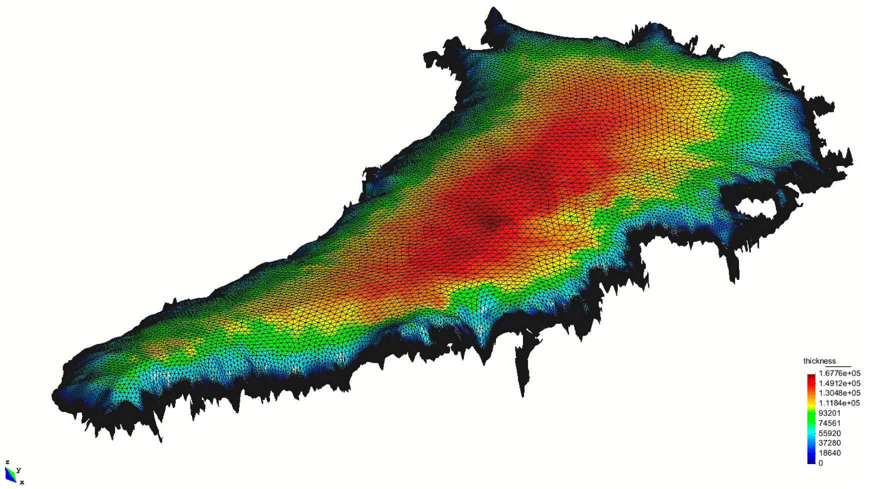

Figure 1: A three-dimensional prismatic mesh of the Greenland ice sheet with 26 layers and 33,343 nodes per layer; the colors display the local ice thickness and, to aid visualization, the z-direction of the ice sheet is enlarged by a factor of 15.

4. Variational formulations of the governing equations and their discretizations

4.1. Variational formulations

Taking the dot product of (2.4) with a vector function $\mathbf{v}$, the product of (2.2) with a scalar function r and integrating both over Ω_t results in

$$\begin{cases} -(\nabla\cdot\boldsymbol{\tau},\mathbf{v})_{\Omega_t}+(\nabla p,\mathbf{v})_{\Omega_t}=(\rho\mathbf{g},\mathbf{v})_{\Omega_t}, \\ (\nabla\cdot\mathbf{u},r)_{\Omega_t}=0, \end{cases} \tag{4.1}$$

where $(\cdot,\cdot)_{\mathscr{D}}$ denotes the $L^2(\mathscr{D})$ inner product and $\mathscr{D}$ could denote Ω_t or Γ_t or some other domain. For pairs of scalars, vectors, and tensors, we have $(f,g)_{\Omega_t}=\int_{\Omega_t} f\,g\;d\mathbf{x}$, $(\mathbf{u},\mathbf{v})_{\Omega_t}=\int_{\Omega_t}\mathbf{u}\cdot\mathbf{v}\;d\mathbf{x}$, and $(\dot{\boldsymbol{\varepsilon}},\boldsymbol{\tau})_{\Omega_t}=\int_{\Omega_t}\dot{\boldsymbol{\varepsilon}}:\boldsymbol{\tau}\;d\mathbf{x}$, respectively. At this point, the only requirements of the test functions $\mathbf{v}$ and r are that the integrals we encounter are well defined and that $\mathbf{v}$ satisfies the boundary condition (2.10), i.e., $\mathbf{v}=\mathbf{0}$ on Γ_{ib}.

Integration by parts on the first two terms of the first equation in (4.1) results in

$$\begin{cases} (\boldsymbol{\tau},\nabla\mathbf{v})_{\Omega_t}-(p,\nabla\cdot\mathbf{v})_{\Omega_t}-(\boldsymbol{\tau}\cdot\mathbf{n},\mathbf{v})_{\Gamma_t}+(p\mathbf{n},\mathbf{v})_{\Gamma_t}=(\rho\mathbf{g},\mathbf{v})_{\Omega_t}, \\ (\nabla\cdot\mathbf{u},r)_{\Omega_t}=0. \end{cases} \tag{4.2}$$

Using the symmetry of $\boldsymbol{\tau}$ and (2.6), we have

$$(\boldsymbol{\tau},\nabla\mathbf{v})_{\Omega_t}=\Big(\boldsymbol{\tau},\frac{1}{2}(\nabla\mathbf{v}+(\nabla\mathbf{v})^T)\Big)_{\Omega_t}=\big(\boldsymbol{\tau},\dot{\boldsymbol{\varepsilon}}_{\mathbf{v}}\big)_{\Omega_t}=\big(2\eta_{\mathbf{u},T}\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}},\dot{\boldsymbol{\varepsilon}}_{\mathbf{v}}\big)_{\Omega_t}. \tag{4.3}$$

Using (2.11) and the fact that $\mathbf{v}=\mathbf{0}$ on Γ_{ib}, we also have

$$(-p\mathbf{n}+\boldsymbol{\tau}\cdot\mathbf{n},\mathbf{v})_{\Gamma_t}=(\boldsymbol{\sigma}\cdot\mathbf{n},\mathbf{v})_{\Gamma_t}=(\boldsymbol{\sigma}\cdot\mathbf{n},\mathbf{v})_{\Gamma_{ia;t}}+(\boldsymbol{\sigma}\cdot\mathbf{n},\mathbf{v})_{\Gamma_{ib}}=-(p_{atm},\mathbf{v}\cdot\mathbf{n})_{\Gamma_{ia;t}}. \tag{4.4}$$

Substituting (4.3) and (4.4) into (4.2) results in the variational problem we discretize:

$$\begin{cases} (2\eta_{\mathbf{u},T}\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}},\dot{\boldsymbol{\varepsilon}}_{\mathbf{v}})_{\Omega_t} - (p,\nabla\cdot\mathbf{v})_{\Omega_t} = (\rho\mathbf{g},\mathbf{v})_{\Omega_t} - (p_{atm},\mathbf{v}\cdot\mathbf{n})_{\Gamma_{ia;t}}, \\ -(\nabla\cdot\mathbf{u},r)_{\Omega_t} = 0. \end{cases} \tag{4.5}$$

In addition to (4.5), **u** is required to satisfy (2.10). However, **u** and p are not required to explicitly satisfy the boundary condition (2.11); that boundary condition is natural to the variational formulation and thus is automatically satisfied.

Through a similar process we can recast (2.12)–(2.15) into the variational formulation

$$\begin{aligned} &\Big(\rho c\frac{\partial T}{\partial t},S\Big)_{\Omega_t} + (k\nabla T,\nabla S)_{\Omega_t} + (\rho c\mathbf{u}\cdot\nabla T,S)_{\Omega_t} \\ =&(q,S)_{\Omega_t} + (2\eta_{\mathbf{u},T}\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}}:\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}},S)_{\Omega_t} + \frac{1}{\rho c}(q_{ib},S)_{\Gamma_{ib}}. \end{aligned} \tag{4.6}$$

At this point, the only requirement on the test function S is that the integrals encountered are well defined and that $S=0$ on $\Gamma_{ia;t}$. In addition to (4.6), T is required to satisfy the initial condition (2.13) and the boundary condition (2.14) but is not required to explicitly satisfy the natural boundary condition (2.15); that condition is automatically satisfied by solutions of the variational formulation through the inclusion of the last term on the right-hand side of (4.6).

We also define a variational formulation of the ice-thickness equation (2.17):

$$\Big(\frac{\partial H}{\partial t},V\Big)_{\Omega_H} = \Big(H_{flux} + U_3 - U_1\frac{\partial H}{\partial x} - U_2\frac{\partial H}{\partial y},V\Big)_{\Omega_H}. \tag{4.7}$$

The only requirement on the test function V is that the integrals encountered are well defined. In addition to (4.7), H is required to satisfy the constraint (2.19) and the initial condition (2.18). If H and V are chosen to belong to the same function space, then (4.7) simply defines a least-squares projection onto that space.

We also would like note that in our numerical experiments, due to the geometrical settings of target ice sheets, the ice-atmosphere boundary $\Gamma_{ia;t}$ is decomposed into two parts – lateral boundary and top surface boundary, and special conditions such as periodic or zeros boundary conditions are often imposed on the lateral parts which will cause slight differences in the above variational weak forms.

4.2. Discretized system

At any time t, let $\mathscr{P}_{h;t}$ denote the prismatic subdivision of the ice domain Ω_t into finite elements described in Section 3. Here, h is a measure of the spatial grid size, e.g., the maximum diameter of any of the prismatic finite elements. The finite element space $P_{h;t}$ used for approximating the components of the velocity, the pressure, and the temperature consists of functions that, within each prismatic element, are the product of a two-dimensional linear polynomial in the two horizontal directions and a linear polynomial in the vertical direction, e.g., in x,y,z space, they are functions of the form $a_0+a_1x+a_2y+a_3z+a_4xz+$

a_5yz for constants a_i, $i = 0, \cdots, 5$. Such functions are uniquely determined by their values at the six vertices of a prismatic element. In addition, the functions in $P_{h;t}$ are required to be continuous across element faces. This continuity requirement is automatically met by specifying the value of the functions in $P_{h;t}$ at the nodes of the prismatic subdivision $\mathscr{P}_{h;t}$. We also define the constrained space $P^0_{h;t}$ to consist of functions in $P_{h;t}$ that vanish on Γ_{ib} so that boundary condition (2.10) is satisfied. Such functions are characterized by setting their values at the nodes located on Γ_{ib} to zero. Note that because Ω_t and therefore the prismatic grid $\mathscr{P}_{h;t}$ change in time, the corresponding finite element space $P_{h;t}$ also changes in time.

For the approximation of the thickness field $H(x,y,t)$, we use the finite element space V_h consisting of continuous piecewise linear polynomials defined with respect to the two-dimensional CVDT triangulation $\mathscr{T}_h$ of the horizontal extent of the ice sheet, i.e., of Ω_H, into finite elements described in Section 3. Note that, because we are assuming that Ω_H is fixed, the corresponding finite element space V_h is also fixed in time. Recall that this a simplification introduced here and that in subsequent work, Ω_H will be free to evolve in time as well.

We also divide the time interval $[0, t_{max}]$ into K subintervals $\{[t_{k-1}, t_k]\}_{k=1}^K$, where $t_k = k\Delta t$ and $\Delta t = t_{max}/K$. Then, for each $k = 0, \cdots, K$, let $\mathbf{u}_{h,k}(\mathbf{x})$, $p_{h,k}(\mathbf{x})$, $T_{h,k}(\mathbf{x})$, and $H_{h,k}(x,y)$ denote the approximations to $\mathbf{u}(\mathbf{x}, t_k)$, $p(\mathbf{x}, t_k)$, $T(\mathbf{x}, t_k)$, and $H(x, y, t_k)$, respectively.[‡]

We need to define an approximation $T_{h,0}(\mathbf{x}) \in P_{h;0}$ of the initial data $T_0(\mathbf{x})$; this issue is discussed in Section 7.3. We also need to define an approximation $T_{ia;h,k}(\mathbf{x})$ of the boundary data $T_{ia}(\mathbf{x}, t_k)$. We do so by, for each $k = 1, \cdots, K$, setting each nodal value of the discrete temperature $T_{ia;h,k}(\mathbf{x})$ along the boundary $\Gamma_{ia;t_k}$ to be the value of $T_{ia}(\mathbf{x}, t_k)$ at that node. In same way, we define an approximation $H_{h,0}(x,y) \in V^h$ of the initial data for the height field $H_0(x,y)$ by setting the nodal values of the former equal to those of the latter over the nodes of the CVDT triangulation $\mathscr{T}_h$.

The choice we made for the finite element spaces for the velocity and pressure does not satisfy the LBB (or inf-sup) stability condition [14] required for mixed finite element methods. Thus, to circumvent the LBB condition, we discretize the Stokes equation using a penalty method, i.e., we add a penalty term to the variational formulation (4.5). Thus, for each $k = 0, 1, \cdots, K$, given Ω_{t_k} and $T_{h,k}(\mathbf{x})$, we seek functions $\mathbf{u}_{h,k} \in (P^0_{h;t_k})^3$ and $p_{h,k} \in P_{h;t_k}$

[‡] Any update of the domain Ω_t requires recomputing the grid. To lessen the number of times such updates are invoked, in practice, we do not update the domain every time step; instead we update the domain every M time steps, where $M > 0$ is a specified integer whose value is determined through computational experimentation. Correspondingly, we only have to update the prismatic grid $\mathscr{P}_{h;t}$ and the finite element space $P_{h;t}$ every M time steps. Thus, in the sequel, we have, for $k = 0, 1, \cdots, K$,

$$\Omega_{t_k} = \Omega_{t_{k'}}, \qquad \mathscr{P}_{h;t_k} = \mathscr{P}_{h;t_{k'}}, \qquad \text{and} \qquad P_{h;t_k} = P_{h;t_{k'}},$$

where $k' = M\left[\frac{k}{M}\right]$ with $\left[\frac{k}{M}\right]$ = greatest integer less than or equal to $\frac{k}{M}$.

such that

$$\begin{cases} \big(2\eta_{\mathbf{u}_{h,k},T_{h,k}}\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}_{h,k}},\dot{\boldsymbol{\varepsilon}}_{\mathbf{v}_h}\big)_{\Omega_{t_k}} - (p_{h,k},\nabla\cdot\mathbf{v}_h)_{\Omega_{t_k}} & \\ \qquad = (\rho\mathbf{g},\mathbf{v}_h)_{\Omega_{t_k}} - (p_{atm},\mathbf{v}_h\cdot\mathbf{n})_{\Gamma_{ia;t_k}}, & \text{for all } \mathbf{v}_h\in(P^0_{h;t_k})^3, \\ -(\nabla\cdot\mathbf{u}_{h,k},r_h)_{\Omega_{t_k}} + \epsilon(\nabla p_{h,k},\nabla r_h)_{\Omega_{t_k}} = 0, & \text{for all } r_h\in P_{h;t_k}, \end{cases} \tag{4.8}$$

where $\epsilon > 0$ denotes a penalty parameter that is usually chosen so that $\epsilon = \mathscr{O}(\Delta x^2 + \Delta y^2 + \Delta z^2)$. We use a direct Picard iteration algorithm to solve the nonlinear discrete system (4.8). Specifically, at each time step $k = 0,1,\cdots,K$ and for $j = 1,2,\cdots$, we solve the linear system

$$\begin{cases} \big(2\eta_{\mathbf{u}^{(j-1)}_{h,k},T_{h,k}}\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}^{(j)}_{h,k}},\dot{\boldsymbol{\varepsilon}}_{\mathbf{v}_h}\big)_{\Omega_{t_k}} - (p^{(j)}_{h,k},\nabla\cdot\mathbf{v}_h)_{\Omega_{t_k}} & \\ \qquad = (\rho\mathbf{g},\mathbf{v}_h)_{\Omega_{t_k}} - (p_{atm},\mathbf{v}_h\cdot\mathbf{n})_{\Gamma_{ia;t_k}}, & \text{for all } \mathbf{v}_h\in(P^0_{h;t_k})^3, \\ -(\nabla\cdot\mathbf{u}^{(j)}_{h,k},r_h)_{\Omega_{t_k}} + \epsilon(\nabla p^{(j)}_{h,k},\nabla r_h)_{\Omega_{t_k}} = 0, & \text{for all } r_h\in P_{h;t_k}, \end{cases} \tag{4.9}$$

for $\mathbf{u}^{(j)}_{h,k}$ and $p^{(j)}_{h,k}$. The iteration over the index j is terminated when $L^2(\Omega)$ norm of the difference between two successive iterates $\mathbf{u}^{(j-1)}_{h,k}$ and $\mathbf{u}^{(j)}_{h,k}$ is smaller than some prescribed tolerance or if a prescribed maximum number of iterations is reached. Upon terminating the iteration, we set

$$\mathbf{u}_{h,k} = \mathbf{u}^{(j)}_{h,k} \quad \text{and} \quad p_{h,k} = p^{(j)}_{h,k}.$$

For $k = 0$, the iteration over the index j is initialized by setting $\mathbf{u}^{(0)}_{h,0} = \mathbf{0}$ almost everywhere and, for $k > 0$, by setting $\mathbf{u}^{(0)}_{h,k} = \mathbf{u}_{h,k-1}$. Based on standard analyses of Picard-type iterations, we expect that this nonlinear iteration is linearly convergent with a contraction constant $(n-1)/n$, where n denotes the exponent in the Glen's flow law.

We use the implicit backward-Euler method to effect the temporal discretization of the energy equation; however, the velocity field appearing in this equation is treated explicitly. Thus, for $k = 0,1,\cdots,K$, given Ω_{t_k}, $T_{h,k}(\mathbf{x})$[§], and $\mathbf{u}_{h,k}$, we seek a function $T_{h,k+1}(\mathbf{x}) \in P_{h;t_k}$ such that $T_{h,k+1}(\mathbf{x}) = T_{ia;h,k+1}(\mathbf{x})$ on $\Gamma_{ia;t}$ and

$$\begin{aligned} &\Big(\rho c\frac{T_{h,k+1}-T_{h,k}}{\Delta t},S_h\Big)_{\Omega_{t_k}} + (\kappa\nabla T_{h,k+1},\nabla S_h)_{\Omega_{t_k}} + (\rho c\mathbf{u}_{h,k}\cdot\nabla T_{h,k+1},S_h)_{\Omega_{t_k}} \\ =&(q,S_h)_{\Omega_{t_k}} + (2\eta_{\mathbf{u}_{h,k}}\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}_{h,k}}:\dot{\boldsymbol{\varepsilon}}_{\mathbf{u}_{h,k}},S_h)_{\Omega_{t_k}} + \frac{1}{\rho c}(q_{ib},S_h)_{\Gamma_{ib}}, \qquad \text{for all } S_h\in P_{h;t_k}. \end{aligned} \tag{4.10}$$

For the discretization of the thickness evolution equation, we use the explicit forward Euler method (for efficiency issue) so that for $k = 0,1,\cdots,K$, given $\mathbf{u}_{h,k}$, we solve the

[§] If a update of the domain Ω_t and its mesh happens at the time t_k, we must project the temperature $T_{h,k}$ obtained from the previous mesh $\mathscr{P}_{h;t_{k-1}}$ onto the updated mesh $\mathscr{P}_{h;t_k}$, for example, linear interpolation or L^2 projection can be employed here without much loss of accuracy since low-order finite element spaces are used in the discretizations.

linear system

$$
\begin{aligned}
&\big(H_{h,k+1}, V_h\big)_{\Omega_H} = \big(H_{h,k}, V_h\big)_{\Omega_H} \\
&\quad + \Delta t \Big(H_{flux;k} + U_{3;h,k} - U_{1;h,k}\frac{\partial H_{h,k}}{\partial x} - U_{2;h,k}\frac{\partial H_{h,k}}{\partial y}, V_h\Big)_{\Omega_H}, \qquad \text{for all } V_h \in V^h, \quad (4.11)
\end{aligned}
$$

for $H_{h,k+1}(x,y) \in V^h$, where $(\cdot,\cdot)_{\Omega_H}$ denotes the $L^2(\Omega_H)$ inner product. To enforce the $H \geq 0$ constraint, we simply post process the result of (4.11) by setting any negative nodal values of $H_{h,k+1}$ to almost zero. In (4.11), we determine the vertical average velocity approximation $\mathbf{U}_{h,k}(x,y) \in (\mathscr{V}^h)^3$ from the solution $\mathbf{u}_{h,k}(x,y,z)$ of (4.8) by evaluating, at each node of the triangulation $\mathscr{T}^h$, the integral of the latter with respect to the vertical coordinate z. This is easy to do because, for fixed values of x and y, $\mathbf{u}_{h,k}(x,y,z)$ is a piecewise linear function of z. Once $H_{h,k+1}$ has been determined, we define the new domain $\Omega_{t_{k+1}}$. Although the CFL condition needs to be satisfied for the time step Δt, it usually can be quite large for the ice-sheet evolution [28].

5. Parallel implementation

Parallel computations apply a *divide and conquer* strategy to solve large-scale problems. In our parallel solvers, we adopted the *domain decomposition method* (DDM) [9,45], i.e., we divide the modeling domain into sub-domains and solve for the unknowns in the sub-domains simultaneously on distributed computer processors. The finite element mesh is first partitioned into a number of sub-meshes whose number is consistent with the number of available processors. This results in dividing the computing domain Ω_{t_k} into interconnected sub-domains. By doing so, the original large-scale computing problem is decomposed into a group of relatively simpler and smaller problems that can be simultaneously solved on different processors. The sub-domains could be overlapping or non-overlapping, depending on the specific scheme employed. A variety of preconditioners [2] can used to tackle inner interfaces, i.e., the degrees of freedom corresponding to the grid points common to more than one subdomain [27,46]. A variety of methods, including Lagrange multiplier-based substructuring methods [32], preconditioned Krylov subspace iterative methods [4,20], parallel direct solvers [1], and multi-grid [9,25] and multi-level [4] solvers, can be used to solve the linear systems introduced in Section 4.2.

To decompose the model domain for parallel computing, either edge- or vertex-based partition methods can be used. In addition, we need to define variables for both local and global interfaces, i.e., as the computing domain is partitioned, local variables have to be re-ordered and the global ordering of variables must be updated systematically. In our implementation, the mesh partition is handled by the multi k-way partitioning algorithm (the package "METIS") in order to obtain good load balance [26]. The partitioned sub-domains have one layer of overlapping elements, that are needed for data communication, i.e., interface updates, during the solution process. We have developed a parallel subsystem to directly generate unstructured sparse graphs [36] from hybrid finite element meshes that can be separated from the parallel package as a preprocessor. A partitioning of the Greenland CVDT mesh into 16 sub-meshes is shown in Fig. 2.

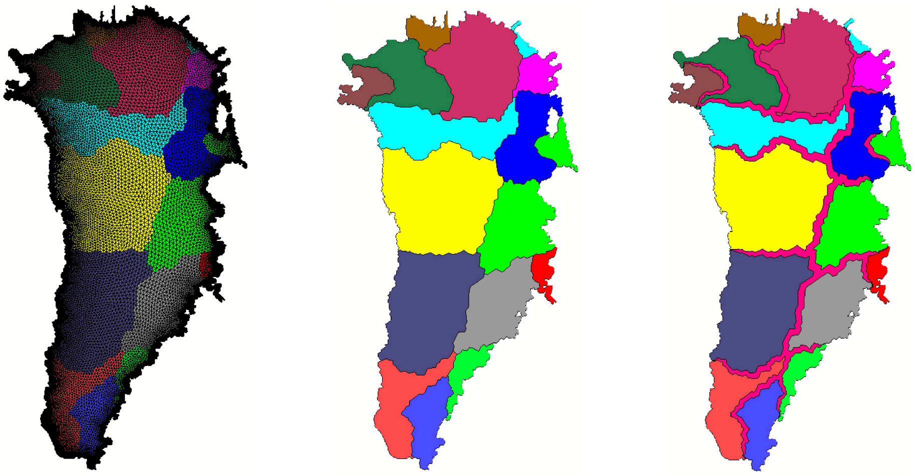

Figure 2: Partition of the Greenland CVDT mesh into 16 sub-meshes and inner boundary formation. Left: the two-dimensional CVDT grid of Greenland. Middle: the mesh is transformed into an unstructured graph which is used as input into the METIS package for partitioning. Right: all sub-domains are expanded by one layer to form the inner interfaces for communication between sub-domains.

The large, sparse linear systems arising in high-resolution geoscience problems and models often involve millions of unknowns. Solving such large-scale systems is a challenging task due to the high demand on computing power and memory. Iterative solution techniques based on Krylov subspace methods (as well as preconditioning techniques, such as multi-grid and incomplete LU factorization preconditioner [29]) are commonly used because the Krylov subspace iteration methods require only matrix-vector products. In recent years, many parallel solvers for large-scale linear systems have become available in the public domain. These include Multifrontal Massively Parallel sparse direct Solver (MUMPS) [3], SuperLU [29], AZTEC [44], and parallel Algebraic Recursive Multilevel Solvers (pARMS) [30]. Basically, there are two types of parallel iterative linear system solvers based on Krylov methods: the GMRES and BiCG stabilized type for nonsymmetric systems and the conjugate gradient type for symmetric systems [33]. Message Passing Interface (MPI) is used as the parallel environment and we also chose to use the parallel ILU($\dot{\varepsilon}$) preconditioned GMRES solver for solution of the linear systems resulting from our numerical discretizationsl; in particular, the package AZTEC2.0 is employed in our implementation due to its reliability and robustness.

6. Verification of the 3D full-Stokes solver

We verify our parallel, three-dimensional full-Stokes solver using two of the idealized test cases considered in [41]. We use the parameters given in that paper; see Table 1. Note that in these experiments, the rate factor A is chosen independent of the temperature.

The first test (Experiment A in [41]) considers ice flow within a parallel-sided slab of mean thickness 1000 m, on a bed having periodic topography of amplitude 500 m and a

Table 1: Parameters and constants for verification experiments for the 3D full-Stokes solver.

Parameter name	Symbol	Value	Units
rate factor	A	10^{-16}	$Pa^{-n}a^{-1}$
power law exponent	n	3	–
ice density	ρ	910	$kg\,m^{-3}$

mean slope $\alpha = 0.5°$. The basal topography is defined as:

$$z_{ib}(x,y) = z_{ia}(x,y) - 1000 + 500\sin(\omega x)\sin(\omega y).$$

The ice-atmosphere boundary is given by

$$z_{ia}(x,y) = -x\tan(\alpha).$$

Here, $(x,y) \in [0,L] \times [0,L]$ and although we have run tests with other values of L with similar results, we only report on the cases $L = 40$, 80, and 160 km. Note that the basal bumps have a frequency of $\omega = 2\pi/L$. A no-slip boundary condition $\mathbf{u} = \mathbf{0}$ is imposed at the ice-bedrock boundary. At the lateral boundaries, doubly periodic boundary conditions are imposed on the velocity $\mathbf{u}$ and a free-surface condition is imposed at the top boundary, i.e., we set $p_{atm} = 0$ in (2.11). The set-up for the second test (Experiment B in [41]) is the same except that the basal topography does not vary in y, i.e., we have

$$z_{ib}(x,y) = z_{ia}(x,y) - 1000 + 500\sin(\omega x).$$

For both Experiments A and B, we use a prismatic mesh having 41 layers; the horizontal CVDT mesh is uniform and consists of 15,739 nodes and 30,972 triangles. Thus, the three-dimensional prismatic grid has 645,299 nodes and 1,238,880 prismatic elements. The penalty parameter is set to $\epsilon = 5 \times 10^{-5}$. Because it is cumbersome to apply periodic boundary conditions in a finite element approximation on unstructured meshes, we use a larger domain of size $3L \times 3L$ and set the lateral boundary conditions to $\mathbf{u} = 0$. We then focus attention on the center region of area $L \times L$.

For the linear system solver, we use a preconditioned GMRES method with an ILU(0)) preconditioner. We set the maximum number of iterations to 2,500 and the tolerance for the residual is 10^{-10}. For the nonlinear solver, the relative error was reduced by a factor of approximately 2/3 at each step of the Picard iteration as was expected for the value of $n = 3$ of the Glen's flow rate exponent. We stopped the iterative nonlinear solver whenever the relative error reached 10^{-4}. The CPU running times of our parallel, three-dimensional full-Stokes solver for the case $L = 160$ km are given in Table 2.

Table 2: Running times of the parallel 3D full-Stokes solver for Experiments A and B with $L = 160$ km.

Parallel performance	Experiment A			Experiment B		
Number of processors	20	40	80	20	40	80
CPU time in seconds	4796	2304	1024	4773	2315	1019

Numerical simulation results for the ice velocity $\mathbf{u} = (u_1, u_2, u_3)^T$ at the top surface are presented in Figs. 3 and 4. With the same geometric settings and parameters, the numerical results produced by our parallel, three-dimensional, full-Stokes solver show excellent agreement with the benchmark results given in [40, 41] for the higher-order and full-Stokes ice sheet models (ISMIP-HOM). This is illustrated in Fig. 5 where the horizontal surface speed

$$u_s = \sqrt{u_1^2 + u_2^2}$$

is plotted vs. x for $y = L/4$. The results from [41] are the background of Fig. 5 and our results are given by the red curve.[¶] In that figure, PFEM denotes our parallel finite element, three-dimensional, full-Stokes solver and FS and NFS denote the full-Stokes and higher-orders models from [41], respectively.

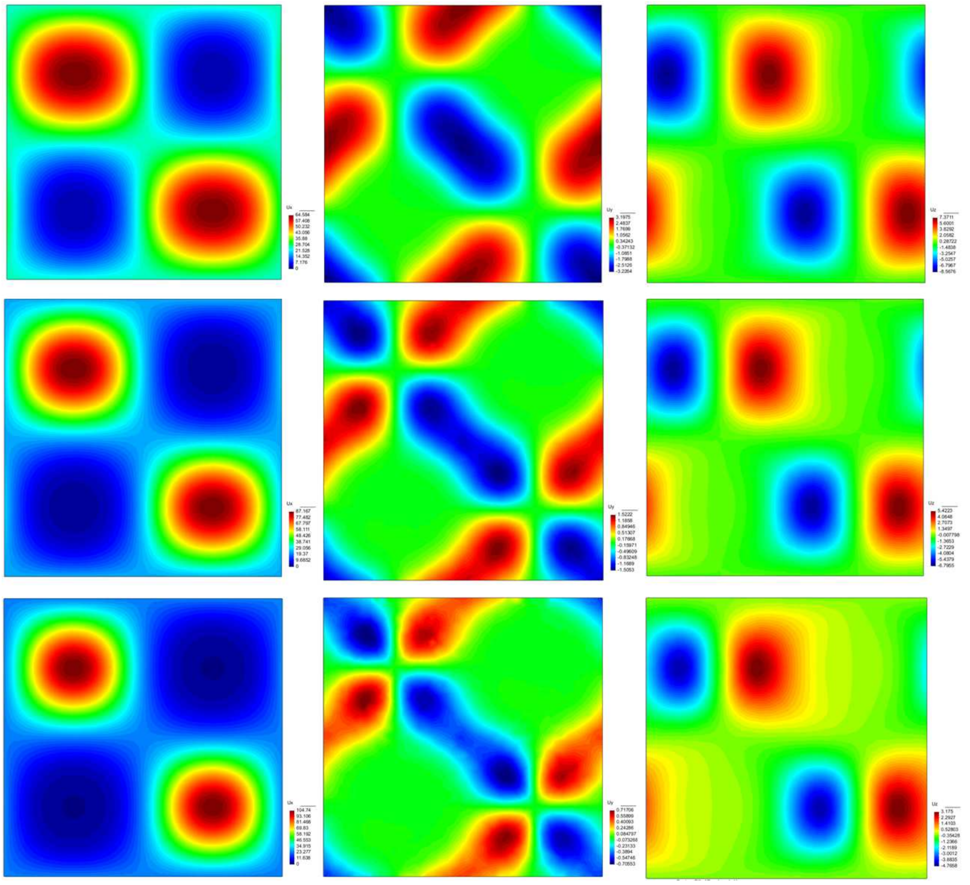

Figure 3: Numerical simulation results of Experiment A. From left to right: the components u_1, u_2, and u_3 of the top-surface velocity; from top to bottom: $L = 40$, 80, 160 km.

[¶] We have taken the background of Fig. 5 directly from the paper [41].

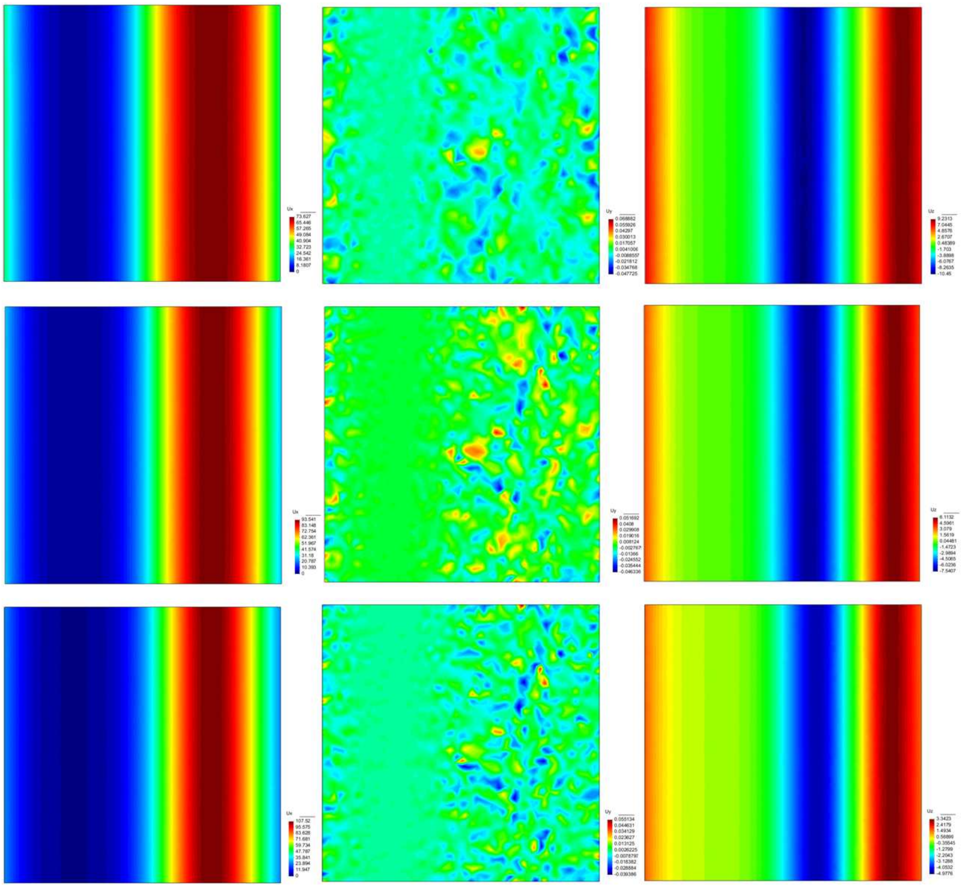

Figure 4: Numerical simulation results of Experiment B. From left to right: the components u_1, u_2, and u_3 of the top-surface velocity; from top to bottom: $L = 40$, 80, 160 km.

7. Preliminary results of Greenland ice sheet simulations

In this section, we present the results from some preliminary simulations of the Greenland ice sheet based on the model discussed above. We emphasize that our goal here is not to mimic observed velocities, temperatures, or thickness changes in Greenland, but rather to illustrate the capabilities of the full-Stokes computational model as well as the energy and mass conservation solvers on problems with realistic geometries and grid resolutions.

7.1. Input data and parameters

Observational data of current bedrock topography and ice thickness for Greenland at 5 km-resolution are presented in Fig. 6. We use the thickness data to define the horizontal extent Ω_H of the ice sheet and the initial ice sheet thickness $H_0(x,y)$. We use the bedrock topography data to define the ice-bedrock boundary $z_{ib}(x,y)$ and, by adding to it the observed thickness data, to define the initial ice-atmosphere boundary $z_{ia}(x,y,0)$. The maximum and minimum values of the ice thickness are 3340 m and 5 m, respectively. Observational data of the annual average ice surface temperature [15] is also given in Fig. 6. As shown in the right of Fig. 6, the temperature in some marginal areas of the Greenland ice sheet is already above the melting point.

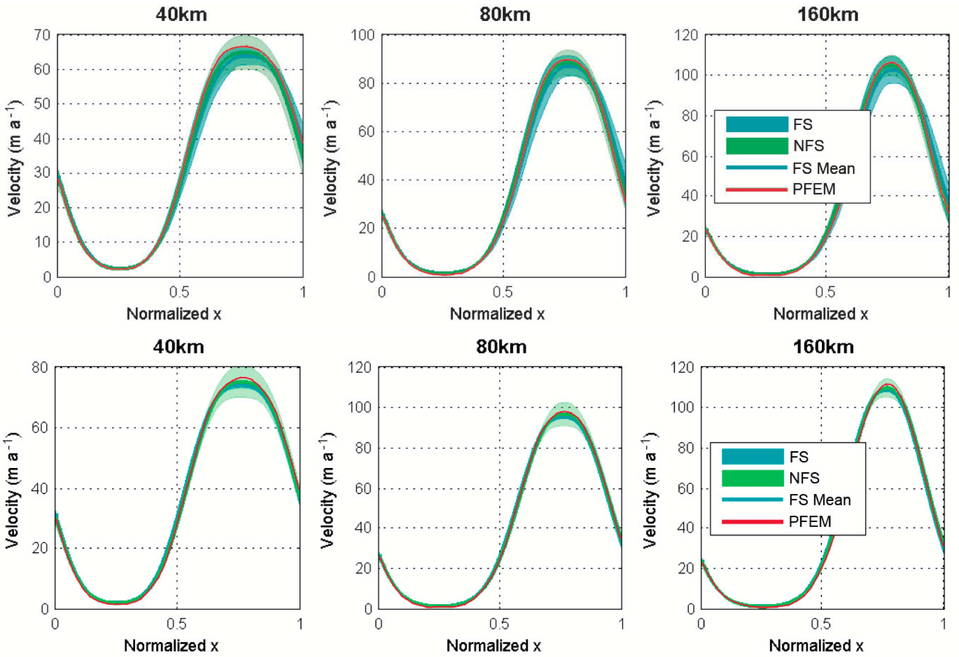

Figure 5: The horizontal speed of the ice flow at the top surface vs. x at $y = L/4$ for Experiments A (top row) and B (bottom row) and for $L = 40$, 80, 160 km (from left to right).

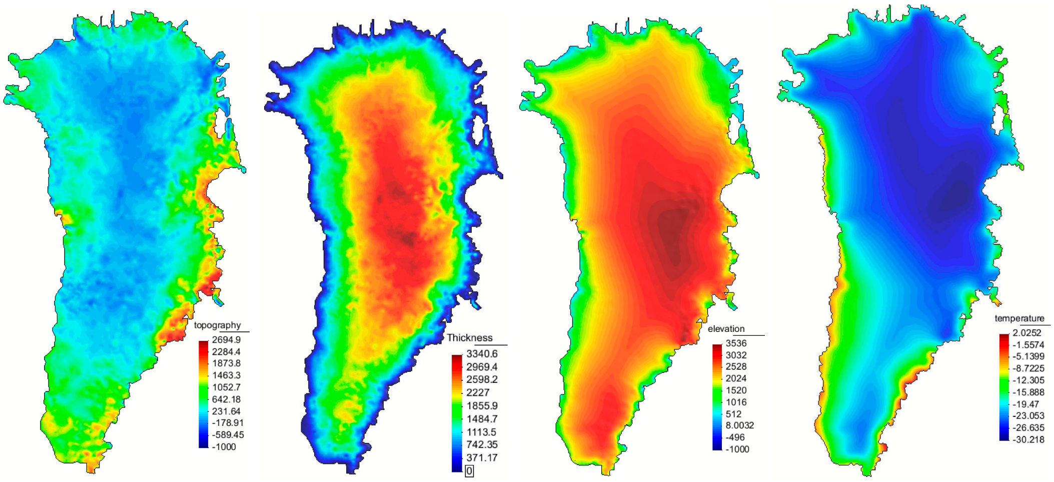

Figure 6: Observational data with 5 km-resolution of the Greenland ice sheet. From left to right: topography of the bedrock (the color indicates the elevation), ice thickness, elevation of the ice surface, and the ice surface temperature.

We first generated a two-dimensional CVDT grid of the Greenland ice sheet horizontal domain Ω_H as determined from the observational data. The CVT point density function is chosen such that the local mesh size is inversely proportional to the ice thickness. The resulting two-dimensional grid has 33,343 nodes and 64,125 high-quality triangles. Then, a three-dimensional prismatic finite element mesh was produced by dividing the ice sheet into 26 layers in the vertical direction, resulting in a total 866,918 nodes and 1,603,125 prismatic elements; see Fig. 1 for an illustration. The layers are defined by dividing the ice thickness at each horizontal grid point into equal subintervals. As a result, at each grid point of the horizontal CVDT mesh, the layers have equal thickness but that, because the thickness of the ice sheet is not constant, the layer thicknesses change from one horizontal grid point to another. In Table 3, we list all of the relevant parameters and constants used for our Greenland ice sheet simulations.

Table 3: Parameters and constants for the Greenland ice sheet simulations.

Parameter name	Symbol	Value	Units
power law exponent	n	3	–
penalty parameter	ϵ	$1 \times 10^{-4} \sim 5 \times 10^{-5}$	–
absolute zero temperature	T_0	-273.15	^{o} C
activation energy ($T < 263K$)	Q	60×10^3	J mol^{-1}
activation energy ($T > 263K$)	Q	139×10^3	J mol^{-1}
flow law constant ($T < 263K$)	a	1.3×10^{-5}	Pa^{-3} a^{-1}
flow law constant ($T > 263K$)	a	6.26×10^{10}	Pa4^{-3} a^{-1}
second per year	spy	31556926	s/a
ice density	ρ	918.0	kg m $^{-3}$
gravitational constant	g	9.81	m s $^{-2}$
initial equivalent strain rate	$\dot{\varepsilon}_e^0$	1×10^{-4}	–
minimum equivalent strain rate	$\dot{\varepsilon}_{min}$	1×10^{-30}	–
geothermal heat flux from seabed	β	0.055	W/m^2
universal gas constant	R	8.314	J mol^{-1} K^{-1}
thermal conductivity of ice	κ	2.2	W(m°C)$^{-1}$
heat capacity of ice	c	2093.0	J(kg°C)$^{-1}$
heat source term of ice	q	0.0	W m^{-3}

7.2. Simulations with a temperature-independent effective viscosity

We first test our three-dimensional full-Stokes solver by simulating the velocity field in the Greenland ice sheet for a constant (temperature-independent) flow law rate factor $A = 10^{-16}$. In this case, we can focus on the Stokes solver itself without encumbrances resulting from coupling to temperature and thickness evolution.

The no-slip boundary condition at the ice-bedrock boundary and a zero velocity/flux condition are imposed on the lateral boundaries. We set the convergence criteria of the ILU(0)-preconditioned GMRES parallel linear solver to 1.0×10^{-8}. We found that 10 Picard iterations are enough for the solutions to converge well.

Numerical results for the surface speed are presented in Fig. 7; specifically, we display the quantity

$$v_{ia} = \log_{10}\left(1.0 + \sqrt{u_1^2 + u_2^2}\right), \tag{7.1}$$

where the horizontal velocity components u_1 and u_2 are evaluated at the ice-atmosphere boundary and where (7.1), $\sqrt{u_1^2+u_2^2}$ is the surface speed. Comparing v_{ia} with the bedrock topography and ice thickness data shown in Fig. 6 shows that, to a large extent, speed is closely related to the elevation of the ice sheet and its thickness distribution, as expected [40–43].

7.3. Simulations with the full-Stokes model coupled to temperature and thickness evolution

We now turn to simulation for which the full-Stokes model for ice sheet dynamics is coupled to the ice temperature and thickness evolution. Our goal here is to test the solvers for temperature and thickness evolution and the coupling of these solvers to the full-Stokes dynamical solver. Although these simulation lack realism from a geophysical point of view, they include sufficient features to make the computational tests relevant.

The coupled simulation using the Greenland ice sheet geometry follows the following recursive process.

1. We prescribe the temperature and thickness distributions as well as a three-dimenaional prismatic grid as inputs into the discretized full-Stokes system (4.8). For the first time step, these are prescribed as initial conditions; for subsequent time steps, these are determined as outputs of Steps 2 and 3 at the previous time steps. We apply the no-slip boundary condition at the bedrock and zero Dirichlet condition on the lateral boundaries. The parallel ILU(0)-preconditioned GMRES method is used to solve the linear system. The convergence criteria is 1.0×10^{-8}, the maximum number of linear solver iterations is set to 2,500, and the maximum number of nonlinear Picard iterations is set to 12. This step results in new approximate velocity and pressure fields.

2. Next, we use the approximate velocity field determined in Step 1 as input into the discrete energy equation (4.10). The strain heating term is set to zero which is unrealistic in true ice sheet simulations, but is adequate for testing couplings to other system components. We take a time step of two months. The values of the parameters used for (4.10) are c = 2093 J(kg°C)$^{-1}$ and κ = 2.2 W(m°C)$^{-1}$. The geothermal heat flux from the bedrock is given by $\alpha = 0.055W/m^2$. We use the parallel biconjugate gradient stabilized (BiCG_stab) method as the solver for the resulting linear system. Ten iterations of a Gauss-Seidel preconditioner for each subdomain are used and the convergence criteria for solving the linear system is set to 1.0×10^{-9}. This step results in a new approximate temperature field. The results of this step can be used to repeat the process for the next time step, starting with Step 1, in case the ice sheet thickness is not updated.

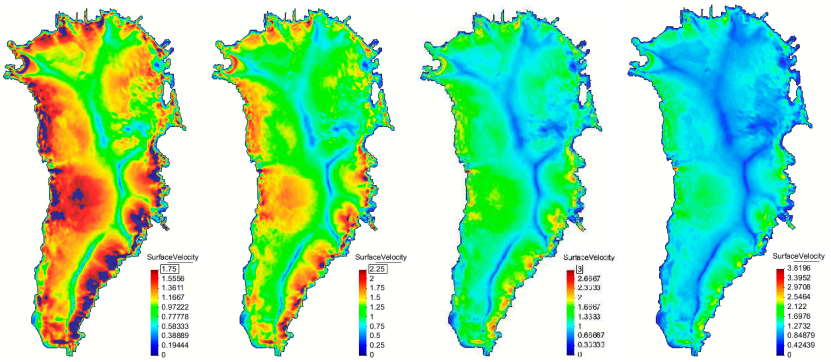

Figure 7: Distribution of v_{ia} for the Greenland ice sheet by using the full-Stokes model with a constant Glen's flow law rate factor $A = 10^{-16}$ and the power law exponent $n = 3$. For better visualization, we set the upper limit of the values displayed to each of the values 1.75, 2.25, 3, and $\max v_{ia}$ which correspond to surface speeds of 55.23, 176.83, 999, and $6,600\,ma^{-1}$, respectively.

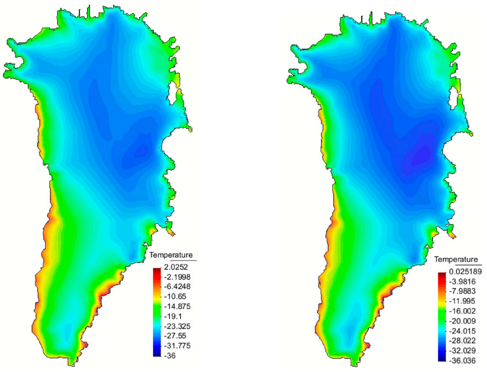

Figure 8: Simulation results for the temperature evolution of the Greenland ice sheet after one year. Left: the initial surface temperature field after stabilization by a pure diffusion process; right: the simulated evolved surface temperature field after one year.

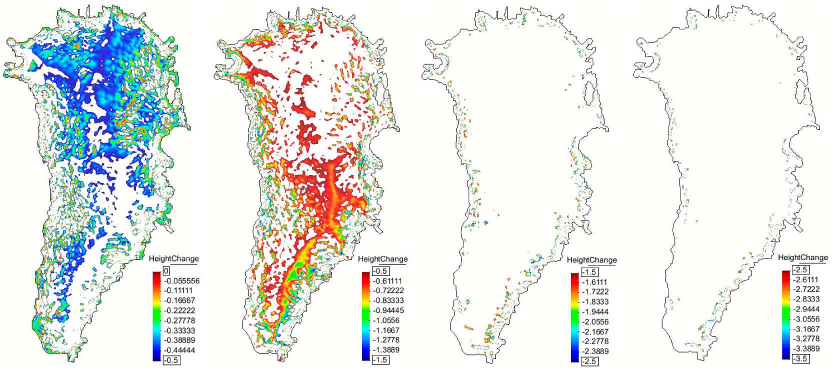

Figure 9: Simulated ice-thickness change of the Greenland ice sheet after one year; the values plotted are restricted to different ranges to aid in the visualization and interpretation.

3. Then, every $M = 6$ time steps of the temperature evolution, the ice sheet thickness is updated from (4.11), again using the approximate velocity field determined in the Step 1 as input. This step results in a new approximate thickness field. The new thickness field is used to define a new computational domain and a new prismatic grid as explained before. The results of Steps 2 and 3 can used to repeat the process, starting with Step 1, for the next time step.

To start the process, we need an approximation of the present-day temperature field throughout the Greenland ice sheet. As mentioned above, we have observational and re-analysis data for the ice surface temperature. For our testing purposes, we set the bottom bedrock temperature to $-36°$C everywhere. We then obtain an approximate temperature field in the interior of the Greenland ice sheet by using linear interpolation on the prismatic mesh based on these data. However, this approximation is prone to roughness so that we use the interpolated temperature field as an initial condition for a pure diffusion problem on the prismatic grid which we march, with a time step of 0.5 years, to steady state. We found that this takes about 10 years and results in the smoother temperature field shown in the left of Fig. 8 and which we use as the initial condition for the first time we visit Step 1 of the above procedure.

7.3.1. Temperature evolution

We conducted several numerical experiments to test our parallel solver for the discretized energy equation (4.6). In [39], it was suggested that the temperature dependence of both c and k is small over the range of temperature expected in polar ice sheets. We thus assume that both these parameters are constant, as listed in Table 3. As mentioned above, the computed smoothed temperature field is adopted as the initial condition for (4.6) and the discretized convection velocity field is obtained from (4.8). Adiabatic conditions are imposed on the top surface. A specified geothermal flux $\kappa\partial T/\partial\mathbf{n}$ is imposed at the bedrock boundary. It is set to be an uniform, upward (into the ice) flux with a value of 0.055 W/m^2. We assume there are no internal heat sources due to strain heating. Fig. 8 shows the initial surface temperature field after stabilization and the evolved surface temperature field after one year.

7.3.2. Free surface evolution

We next computed the free surface evolution of the Greenland ice sheet using (4.11) with the discretized output velocity obtained from (4.8). The time step for updating the ice thickness is one year so that, because the time step used for the temperature evolution is two months, we have that $M = 6$ in footnote 2. To clearly display the change of the ice sheet free surface, we only display the decreasing part of the ice sheet change in Fig. 9.

We note that small interpolation errors occur whenever re-gridding of the existing finite element mesh is needed due to ice-thickness change; however, the topology of the mesh remains unchanged. Finally, some elements of the ice sheet will vanish because of the successive loss of the ice mass. Then, the total number of finite element cells is reduced and the numerical calculation process is started over from the point at which the mesh and data partition procedures are invoked. As a result, the whole process of the ice sheet evolution simulation becomes self-consistent.

8. Concluding remarks

A finite element discretization of a three-dimensional, full-Stokes computational model for ice sheet dynamics and temperature and height evolution has been developed. The model features high-quality, variable resolution grids, finite element discretizations, and scalable parallel solvers. It has been tested on benchmark problems and applied to high-resolution modeling of the Greenland ice sheet, albeit with simplified boundary conditions. The efficacy of the computational model has been demonstrated through the numerical experiments so that its further development is warranted. In this regard, future work includes the implementation of higher-order accurate finite element and time-stepping discretization methods, implementing realistic basal sliding conditions along the ice-bedrock boundary, accounting for ice-ocean interaction at the lateral boundaries, testing parallel scalability on large numbers of processors, applying the model to realistic ice sheet settings such as Greenland and Antarctica, and incorporating the dynamical core of the model into next-generation ice sheet models [31] such as the Community Ice Sheet Model.

Acknowledgments This work was partially supported by the US DOE Office of Science's Climate Change Prediction Program through DE-FG02-07ER64431, DE-FG02-07ER64432 and DOE 07SCPF152, and by the US National Science Foundation under grant number DMS-0913491.

References

[1] E. Agullo, A. Guermouche, and J. L'Excellent, A preliminary out-of-core extension of a parallel multifrontal solver, *Euro-Par 2006 Parallel Processing*, pp. 1053-1063.

[2] M. Ainsworth, A preconditioner based on domain decomposition for h-p finite-element approximation on quasi-uniform meshes, *SIAM J. Numer. Anal.* **33** 1996, pp. 1358-1376.

[3] P. Amestoy, I. Duff, J. L'Excellent, and J. Koster, A fully asynchronous multifrontal solver using distributed dynamic scheduling, *SIAM J. Matrix Anal. Appl.* **23** 2001, pp. 15-41.

[4] O. Axelsson and M. Larin, An algebraic multilevel iteration method for finite element matrices, *J. Comput. Appl. Math.* **89** 1998, pp. 135-153.

[5] J. Bamber, S. Ekholm, and W. Krabill, A new, high resolution digital elevation model of Greenland fully validated with airborne laser altimeter data, *J. Geophys. Res.* **106** 2001, pp. 6733-6745.

[6] L. Bengtsson, Climate of the 21st-Century, *Agric. Forest Meteor.* **72** 1994, pp. 3-29.

[7] H. Blatter, Velocity and stress fields in grounded glaciers: a simple algorithm for including deviatoric stress gradients, *J. Glaciol.* **41** 1995, pp. 33-44.

[8] D. Bromwich, L. Bai, and G. Bjarnason, High-resolution regional climate simulations over Iceland using Polar MM5, *Month. Weath. Rev.* **133** 2005, pp. 3527-3547.

[9] C. Douglas, Multigrid methods in science and engineering, *IEEE Comput. Sci. Engrg.* **3** 1996, pp. 55-68.

[10] Q. Du, V. Faber, and M. Gunzburger, Centroidal Voronoi tessellations: Applications and algorithms, *SIAM Review* **41** 1999, pp. 637-676.

[11] Q. Du, M. Gunzburger, and L. Ju, Advances in studies and applications of centroidal Voronoi tessellations, *Numer. Math. Theor. Meth. Appl.* **3** 2010, pp. 119-142.

[12] Q. Du and L. Ju, Finite volume methods on spheres and spherical centroidal Voronoi meshes, *SIAM J. Numer. Anal.* **43** 2005, pp. 1673-1692.

[13] J. Gregory and P. Huybrechts, Ice-sheet contributions to future sea-level change, *Philo. Trans. Royal Soc. - Math., Phys. Engrg. Sci.* 2006 **364**, pp. 1709-1731.

[14] M. Gunzburger, *Finite Element Methods for the Navier-Stokes Equations*, Academic, Boston, 1989.

[15] E. Hanna, P. Huybrechts, I. Janssens, J. Cappelen, K. Steffen, and A. Stephens, Runoff and mass balance of the Greenland ice sheet: 1958-2003, *J. Geophys. Res.* **110** 2005, D13108, doi:10.1029/2004JD005641.

[16] R. Hindmarsh and A. Payne, Time-step limits for stable solutions of the ice-sheet equation, *Ann. Glaciol.* **23** 1996, pp. 74-85.

[17] H. Nguyen, J. Burkardt, M. Gunzburger, L. Ju, and Y. Saka, Constrained CVT meshes and a comparison of triangular mesh generators, *Comp. Geom. Theo. Appl.* **42** 2009, pp. 1-19.

[18] D. Holland, R. Ingram, L. Mysak, and J. Oberhuber, A numerical simulation of the sea-ice cover in the Northern Greenland sea, *J. Geophys. Res. - Oceans* **100** 1995, pp. 4751-4760.

[19] D. Holland, L. Mysak, D. Manak, and J. Oberhuber, Sensitivity study of a dynamic thermodynamic sea ice model, *J. Geophys. Res. - Oceans* **98** 1993, pp. 2561-2586.

[20] K. Ito and J. Toivanen, Preconditioned iterative methods on sparse subspaces, *Appl. Math. Lett.* **19** 2006, pp. 1191-1197.

[21] L. Ju, Conforming centroidal Voronoi Delaunay triangulation for quality mesh generation, *Inter. J. Numer. Anal. Model.* **4** 2007, pp. 531-547.

[22] L. Ju, M. Gunzburger, and W.-D. Zhao, Adaptive finite element methods for elliptic PDEs based on conforming centroidal Voronoi Delaunay triangulations, *SIAM J. Sci. Comput.* **28** 2006, pp. 2023-2053.

[23] L. Ju, H.-C. Lee, and L. Tian, Numerical simulations of the steady Navier-Stokes equations using adaptive meshing schemes, *Inter. J. Numer. Meth. Fluids* **56** 2008, pp. 703-721.

[24] L. Ju, T. Ringler, and M. Gunzburger, Voronoi tessellations and their application to climate and global modeling, *Numerical Techniques for Global Atmospheric Models,* Springer, 2011, pp. 313-342.

[25] M. Jung, On the parallelization of multi-grid methods using a non-overlapping domain decomposition data structure, *Appl. Numer. Math.* **23** 1997, pp. 119-137.

[26] G. Karypis, Multi-constraint mesh partitioning for contact/impact computations, *Supercomputing'03*, Phoenix, Arizona, USA.

[27] V. Korneev and S. Jensen, Domain decomposition preconditioning in the hierarchical p-version of the finite element method, *Appl. Numer. Math.* **29** 1999, pp. 479-518.

[28] E. Le Meur, O. Gagliardini, T. Zwinger, and J. Ruokolainen, Glacier flow modelling: A comparison of the shallow ice approximation and the full-Stokes solution, *Compt. Rendus Phys.* **5** 2004, pp. 709-722.

[29] X. Li and J. Demmel, SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems, *ACM Trans. Math. Soft.* **29** 2003, pp. 110-140.

[30] Z.-Z. Li, Y. Saad, and M. Sosonkina, pARMS: A parallel version of the algebraic recursive multilevel solver, *Numer. Lin. Alg. Appl.* **10** 2003, pp. 485-509.

[31] W. Lipscomb, R. Bindschadler, E. Bueler, D. Holland, J. Johnson, and S. Price, A community ice sheet model for sea level prediction, *Eos* **90** 2009, pp. 23.

[32] J. Mandel and R. Tezaur, Convergence of a substructuring method with Lagrange multipliers, *Numer. Math.* **73** 1996, pp. 473-487.

[33] D. May and L. Moresi, Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics, *Phys. Earth Planet. Inter.* **171** 2008, pp. 33-47, doi:10.1016/j.pepi.2008.07.036.

[34] S. Mernild, G. Liston, C. Hiemstra, and K. Steffen, Surface melt area and water balance modeling on the Greenland ice sheet 1995-2005, *J. Hydrometeor.* **9** 2008, pp. 1191-1211.

[35] S. Mernild, G. Liston, D. Kane, N. Knudsen, and B. Hasholt, Snow, runoff, and mass balance modeling for the entire Mittivakkat Glacier (1998-2006), Ammassalik Island, SE Greenland, *Geografisk Tidsskrift-Danish J. Geogra.* **108** 2008, pp. 121-136.

[36] J. Michel, F. Pellegrini, and J. Roman, Unstructured graph partitioning for sparse linear system solving, *Solving Irregularly Structured Problems in Parallel*, Springer, Berlin, pp. 273-286.

[37] F. Nick, A. Vieli, I. Howat, and I. Joughin, Large-scale changes in Greenland outlet glacier dynamics triggered at the terminus, *Nat. Geosci.* **2** 2009, pp. 110-114.

[38] J. Nye, The distribution of stress and velocity in glaciers and ice sheets, *Proc. Roy. Soc. London - Ser. A* **239** 1957, pp. 113-133.

[39] W. Paterson, *The Physics of Glaciers*, Elsevier Science, Oxford, 1994.

[40] F. Pattyn, A new 3D higher-order thermomechanical ice-sheet model: Basic sensitivity, ice-stream development and ice flow across subglacial lakes, *J. Geophys. Res.* **108** 2003, 2382, doi:10.1029/2002JB002329.

[41] F. Pattyn, L. Perichon, A. Aschwanden, B. Breuer, B. de Smedt, O. Gagliardini, G. Gudmundsson, R. Hindmarsh, A. Hubbard, J. Johnson, T. Kleiner, Y. Konovalov, C. Martin, A. Payne, D. Pollard, S. Price, M. Ruckamp, F. Saito, S. Sugiyama, and T. Zwinger, Benchmark experiments for higher-order and full-Stokes ice sheet models (ISMIP-HOM), *The Cryosphere* **2** 2008, pp. 95-108.

[42] S. Price, E. Waddington, and H. Conway, A full-stress, thermomechanical flow band model using the finite volume method, *J. Geophy. Res. - Earth Surface* **112** 2007, F03020, doi:10.1029/2006JF000724.

[43] T. Ringler, L. Ju, and M. Gunzburger, A multiresolution method for climate system modeling: Application of spherical centroidal Voronoi tessellations, *Ocean Dyn.* **58** 2008, pp. 475-498.

[44] J. Shadid and R. Tuminaro, Sparse iterative algorithm software for large-scale MIMD machines - An initial discussion and implementation. *Concur. Comput.: Pract. Exper.* **4** 1992, pp. 481-497.

[45] A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, Oxford, 1999.

[46] G. Zumbusch, *Schur Complement Domain Decomposition Methods in Diffpack*, Technical report, Sintef Applied Mathematics, Oslo, Norway, 1996.