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 29210, 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

Abstract

As a part of the abrupt climate change model (IMPACTS), we proposed a fully three-dimensional coupled ice-sheet model to explore how ice-sheet changes under realistic conditions. The governing equations for the thermo-mechanical ice-sheet modeling were formulated as a three-dimensional nonlinear full-Stokes system for momentum, an advective-diffusive energy equation for temperature evolution, and a mass-conservation equation for ice thickness change. Finite element approximations and parallel algorithms based on domain decomposition were employed to solve these respective sub-models. The solvers were integrated by a well-designed coupler for exchange of parametric data in these sub-models, such as ice velocity, temperature and ice thickness. By taking advantages of the up-to-date high-performance computing power, the elaborately designed parallel solvers for the coupled system was able to handle the large-scale realistic data from land and space-based observatories and laboratory experiments. In particular, our implementation utilized the high-quality variable-resolution centroidal Voronoi Delaunay triangulation (CVDT) meshing technology and existing parallel linear system solvers. Efficiency and scalability of our parallel solvers were demonstrated by extensive numerical tests.

1 Introduction

Over the past decades, glacier and ice-sheet simulation has been a subject of growing interest because of the influential role they play in global sea-level and climate change studies [7–9, 16, 47]. Among the numerous different types of approaches employed, the full three-dimensional Stokes ice-sheet approximation appears to be a widely adopted method owing to its capability for dealing with realistic parameters and boundary conditions [19, 26, 34, 35]. A complete solver for the 3D full-Stokes ice-sheet modeling requires the integration of effective numerical approximation schemes and elaborately designed couplers for data exchange between different sub-models. Finite difference, finite volume, and finite element methods have been proven successful for high-resolution simulations based on the 3D full-stress equations with free surface evolution [20, 32, 35]. Although many numerical methods have been utilized for ice-sheet modeling, the question arises as to whether they are also applicable to large scale, high-resolution simulations of realistic glaciers. Given the realistic

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

[†]Corresponding author (ju@math.sc.edu).

data from land and space-based observatories [6, 28–30], another key component is a high-quality, adaptive, variable resolution meshing scheme that can often significantly reduce the computational costs 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 [14, 24], the Navier-Stokes equations [25], and the shallow water equations [36]; CVT-based mesh generation offers significant advantages, compared other meshing algorithms, for improving discretization/solution accuracy and controlling local mesh sizes.

Current work focuses on the development of an efficient, parallel finite element solver package for 3D full-Stokes ice-sheet modeling. The flow of the ice-sheet is greatly influenced by temporal and spatial temperature patterns and the ice-sheet topography evolves along with the velocity and velocity gradient. The tightly coupled thermo-mechanical processes are numerically approximated by a scalable system using Message Passing Interface (MPI) that is computing intensive and is capable of performing large amount of data transfer during the simulation processes. In this paper, we first present the governing equations and their numerical approximations. We then validate our numerical schemes and corresponding parallel implementations by performing benchmark tests put forward by the ice-sheet simulation community. The feasibility of our parallel solvers is then demonstrated by applying our package to realistic simulations of the Greenland ice sheet. We then apply our parallel solvers to investigate the influence of, e.g., ambient temperature changes, on flow and thickness changes of that ice sheet.

We consider the evolution of momentum, temperature, and thickness of an ice sheet; Figure 1 provides a sketch of what an ice sheet looks like. However, we consider a simpler situation for which we only have ice-atmosphere and ice-bedrock boundaries. 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 temperature in the ice sheet and a conservation equation for the advance or retreat of the ice-atmosphere boundary;
- implement high-quality, variable-resolution prismatic grids based on centroidal Voronoi Delaunay triangulation surface grids;
- develop and test efficient parallel solvers for the discretized system;
- demonstrate through the above ingredients that the combined package of finite element discretizations, high-quality variable-resolution grids, and efficient parallel solvers results in powerful tool for ice-sheet modeling.

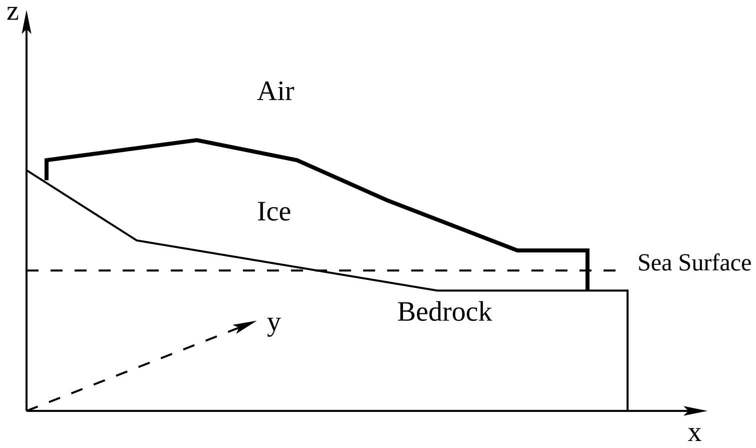

Figure 1: Sketch of the cross section of an ice-sheet.

2 The coupled 3D full-Stokes ice sheet model

In this section, we present the set of governing equations along with 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 the ice sheet is modeled 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. The system is considered incompressible so that the dynamic momentum equation and the kinematic continuity equation are given by

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

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

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 $\mathbf{g} = (0, 0, -|\mathbf{g}|)$ the gravitational acceleration. The stress tensor $\boldsymbol{\sigma}$ can be decomposed into the viscous 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 and $\mathbf{I}$ the identity tensor. The convective contribution $\mathbf{u} \cdot \nabla \mathbf{u}$ has been neglected from the acceleration term due to the very slow motion of the ice flow. Because the time scale of variations of the velocity and pressure fields is large compared to that for the temperature field, we also assume that the remaining acceleration term $\rho \partial \mathbf{u}/\partial t$ is negligible, i.e., we assume that the dynamical behavior of flow in the ice sheet is quasi-static with respect to the evolution of the temperature. 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 $\boldsymbol{\varepsilon}_{\mathbf{u}}$ is defined in terms of the velocity $\mathbf{u}$ by

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

The constitutive law for ice connects the viscous stress tensor $\boldsymbol{\tau}$ to the strain rate tensor $\boldsymbol{\varepsilon}_{\mathbf{u}}$ by the generalized Glen's flow law [31, 33]

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

with

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

where $\eta_{\mathbf{u},T}$ 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 ε_e the equivalent strain rate defined as

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

For the ice evolution, A depends on the temperature through an Arrhenius relation defined by

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

where T denotes the absolute temperature measured in Kelvin, a an empirical constant often used as a model tuning parameter, Q the activation energy, and R the universal gas constant. 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., $\Gamma_t = \Gamma_{ib;t} \cup \Gamma_{ia}$. 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; instead the initial velocity field is determined from the initial temperature field and initial ice thickness distribution by solving the system (2.2), (2.4), (2.10), and (2.11).

2.2 Evolution of temperature

The second prognostic equation describes the evolution of temperature and is given by

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

where $T(\mathbf{x},t)$ denotes the temperature, c the constant heat capacity, κ the constant thermal conductivity, $q(\mathbf{x},t)$ a volumetric heat source, and ρ the constant density of ice. Note that for the two tensors $\boldsymbol{\varepsilon}$ and $\boldsymbol{\tau}$, $\boldsymbol{\varepsilon} : \boldsymbol{\tau}$ denotes the sum of the element-wise products, i.e., $\boldsymbol{\varepsilon} : \boldsymbol{\tau} = \sum_{i,j}^{3} \varepsilon_{ij}\tau_{ij}$. 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,[1] 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 boundary takes the form

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

where T_{ia} denotes a given temperatire on $\Gamma_{ia;t}$, and

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

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

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

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

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

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

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

and the obvious constraint equation

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

[1] 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 determined through an 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 term, i.e., an accumulation-ablation function and/or the known annual average precipitation, that, in general, embodies non-zero contributions to the ice thickness at the ice-atmosphere boundary.

Note that there is two-way coupling between each of 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 [12,13]. 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. The 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 [22, 24] for details about CVDT construction algorithms and related properties. The point density function could be determined, as demonstrated in [36], 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 an ice thickness-related density to generate a two-dimensional CVDT horizontal grid for Greenland. The point density function can also be used to refine 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. We note some smoothing process can be applied to improve the quality of the stretched two-dimensional grid. 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 Figure 2 for a sample three-dimensional prismatic mesh generated for Greenland based on realistic topography data with 5 km resolution, where the local mesh size of the layered CVDT grid is inversely proportional to the ice thickness. In the sequel, 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 $\mathcal{P}_{h;t}$.

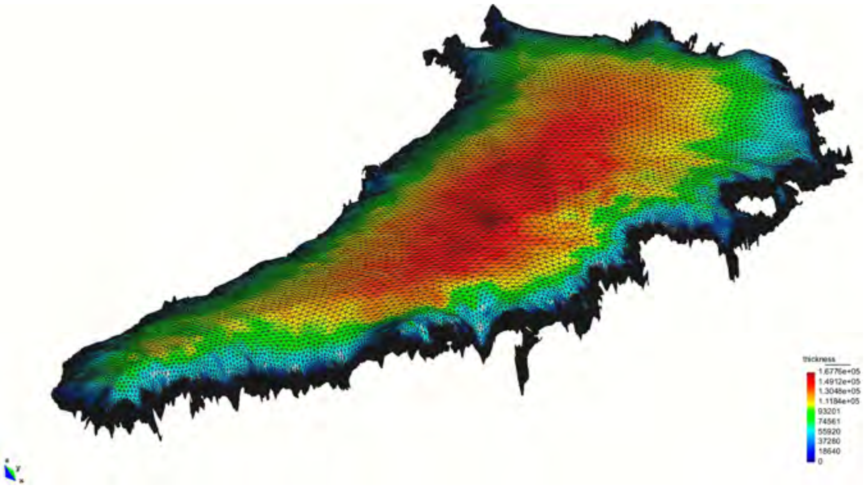

Figure 2: A three-dimensional prismatic mesh of the Greenland ice-sheet with 26 layers and 33,343 nodes per layer; the colors display the local ice thickness and, to aid visualization, the z-direction of the ice sheet is enlarged by a factor of 15.

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

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

where $(\cdot,\cdot)_{\mathcal{D}}$ denotes the $L^2(\mathcal{D})$ inner product. For example, for pairs of scalars, vectors, and tensors, we have $(f, g)_{\Omega_t} = \int_{\Omega_t} fg\, d\mathbf{x}$, $(\mathbf{u}, \mathbf{v})_{\Omega_t} = \int_{\Omega_t} \mathbf{u} \cdot \mathbf{v}\, d\mathbf{x}$, and $(\boldsymbol{\varepsilon}, \boldsymbol{\tau})_{\Omega_t} = \int_{\Omega_t} \boldsymbol{\varepsilon} : \boldsymbol{\tau}\, d\mathbf{x}$, respectively. At this point, the only requirements of the test functions $\mathbf{v}$ and r are that the integrals we encounter below 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

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

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

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

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:

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

In addition to (4.5), $\mathbf{u}$ is required to satisfy (2.10). However, $\mathbf{u}$ 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 as that led from the Stokes system to (4.5), we can recast (2.12)–(2.15) into the variational formulation

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

At this point, the only requirement on the test function S is that the integrals encountered are well defined and that $S = 0$ on Γ_{ia}. 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 (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, the (4.7) simply defines a least-squares projection onto that space.

4.2 Discretized system

At any time t, let $\mathcal{P}_{h;t}$ 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 $a_0 + a_1x + a_2y + a_3z + a_5xz + a_6yz$ for constants a_i, $i = 1, \ldots, 6$. 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 $\mathcal{P}_{h;t}$. We also define the constrained space $P^0_{h;t}$ to consist of functions in $P_{h;t}$ that vanish on Γ_{ib}. 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 $\mathcal{P}_{h;t}$ change in time, the corresponding finite element space $P_{h;t}$ also changes in time.

For the approximation of the height 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 $\mathcal{T}_h$ 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 that the corresponding finite element space $\mathcal{V}_h$ is also fixed in time.

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 $k = 0, \ldots, K$, 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. [2]

[2] Because any update of the domain Ω_t requires recomputing the grid, in practice, one does not update the domain

We need to define an approximation $T_{h,0}(\mathbf{x}) \in P_{h;0}$ of the initial data $T_0(\mathbf{x})$ and, for each $k = 1, \ldots, K$; 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 $k = 1, \ldots, K$, 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 $\mathcal{T}_h$.

The choice we made for the finite element spaces for the velocity and pressure does not satisfy the LBB stability condition [17] 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 $k = 0, 1, \ldots, K$, 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 $p_{h,k} \in P_{h;t_k}$ such that

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

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

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

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 $\mathbf{u}_{h,k} = \mathbf{u}^{(j)}_{h,k}$ and $p_{h,k} = p^{(j)}_{h,k}$. 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 $\mathbf{u}^{(0)}_{h,k} = \mathbf{u}_{h,k-1}$. We expect that this nonlinear iteration is linearly convergent with a contraction constant $\frac{n-1}{n}$, 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 $k = 0, 1, \ldots, K$, 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

every time step; instead, one may update the domain every M time steps, where $M > 0$ is a specified integer. Correspondingly, we only have to update the prismatic grid $\mathcal{P}_{h;t}$ and the finite element space $P_{h;t}$ every M time steps. Thus, in the sequel, we have, for $k = 0, 1, \ldots, K$,

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

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

$T_{h,k+1}(\mathbf{x}) = T_{ia;h,k+1}(\mathbf{x})$ on $\Gamma_{ia;t}$ and

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

For the discretization of the free-surface equation, we use the explicit forward Euler method so that for $k=0,1,\ldots,K$, given $\mathbf{u}_{h,k}$, we solve the linear system

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

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 $\mathbf{U}_{h,k}(x,y)\in(\mathcal{V}^h)^3$ from the solution $\mathbf{u}_{h,k}(x,y,z)$ of (4.8) by evaluating, at each node of the triangulation $\mathcal{T}^h$, the integral of the latter with respect to the vertical coordinate z which 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 $H_{h,k+1}$, we can define the new domain $\Omega_{t_{k+1}}$.

5 Parallel implementation

Parallel computations often apply a *divide and conquer* strategy to solve large-scale problems. In our parallel solvers, we adopted the *domain decomposition method* (DDM) [11,49], 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 meshes are first partitioned into a number of sub-meshes whose number is consistent with the number of processors to be used in the parallel calculation. 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 overlap or could be non-overlapping depending on the specific scheme employed. A variety of preconditioners [2] can used to tackle the inner interfaces [10, 39, 46, 50]. An variety of methods, including Lagrange multiplier-based substructuring methods [43], preconditioned Krylov subspace iterative methods [4, 5, 21, 45], parallel direct solvers [1], and multi-grid [11, 37, 42] and multi-level [5] 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 also 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 then the global ordering of the variables also must be updated systematically. In our implementation, the mesh partition is fulfilled by the multi k-way partitioning algorithm (the package "METIS") in order to obtain good load balance [38]. The partitioned sub-domains have one layer of elements overlapping. We have developed a parallel subsystem to directly generate unstructured sparse graph [44] 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 Figure 3. The inner-boundaries are needed in our implementation for data communication during the solution processes.

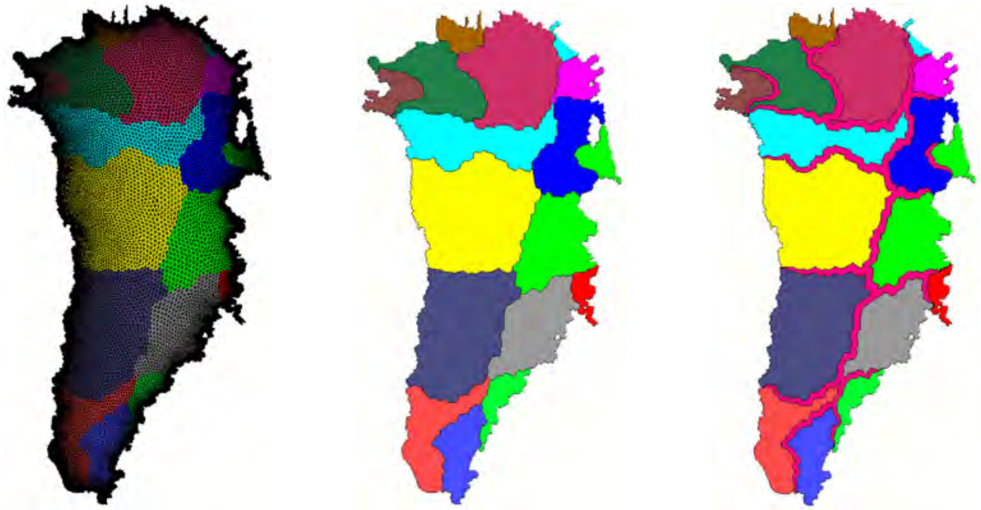

Figure 3: Partition of the Greenland CVDT mesh into 16 sub-meshes and inner boundary formation. Left: the two-dimensional CVDT grid of Greenland. Middle: the mesh is transformed into an unstructured graph which is used as input into the METIS package for partitioning. Right: all sub-domains are expanded by one layer to form the inner interfaces between sub-domains.

The large, sparse linear systems arising in many geoscience problems and models often have millions of unknowns to obtain high resolution in the numerical simulations. Solving such large-scale systems is a challenging task due to the high demand on computing power and memory. Iterative solution techniques for massive linear systems based on Krylov subspace methods as well as preconditioning techniques (such as multi-grid and incomplete LU factorization preconditioner) [40]) 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 public domain. These include MUltifrontal Massively Parallel sparse direct Solver (MUMPS) [3], SuperLU [40], AZTEC [48], and parallel Algebraic Recursive Multilevel Solvers (pARMS) [41]. Basically, there are two types of parallel iterative linear system solvers: the GMRES type for nonsymmetric systems and conjugate gradient type (such as CG stabilized or BiCG stabilized) for symmetric systems [27]. Message Passing Interface (MPI) is used as the parallel environment and we also chose to use the parallel ILU(ε) preconditioned GMRES solver for solution of the linear systems resulting from our numerical discretizationsl; in particular, the popular 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 test cases considered in [34]. We use the parameters given in that paper; see Table 1.

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

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

The first test (Experiment A in [34]) considers ice flow over a bumpy bed in and idealized geometry, i.e., we have a parallel-sided slab of ice having a mean thickness of $1000\,m$ lying on a sloping bed with a mean slope $\alpha = 0.5°$. The basal topography of the ice-bedrock boundary is then defined as a series of sinusoidal oscillations with an amplitude of $500m$:

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

where the ice-atmosphere boundary is given by

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

Here, $(x,y) \in [0,L] \times [0,L]$ and we consider the cases $L = 40$, 80, and $160\,km$. Note that the basal bumps have a frequency of $\omega = 2\pi/L$. The no-slip boundary condition $\mathbf{u} = \mathbf{0}$ is imposed at the ice-bedrock boundary. At the lateral boundaries, 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 [34]) 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 difficult 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 $\mathbf{u} = 0$; for display purposes, we then focus attention on a center area of size $L \times L$.

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, we found the relative error reduced by roughly a factor of 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 $L = 160km$ are given in Table 2.

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

	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 Figures 4 and 5. 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 [32, 34] for the higher-order and full-Stokes ice sheet models (ISMIP-HOM). This is illustrated in Figure 6 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 [34] are the background of Figure 6 and our results are given by the red curve. In that figure, PFEM denotes our parallel finite element, three-dimenaional, full-Stokes solver and FS and NFS denotes the full-Stokes and higher-orders models from [34].

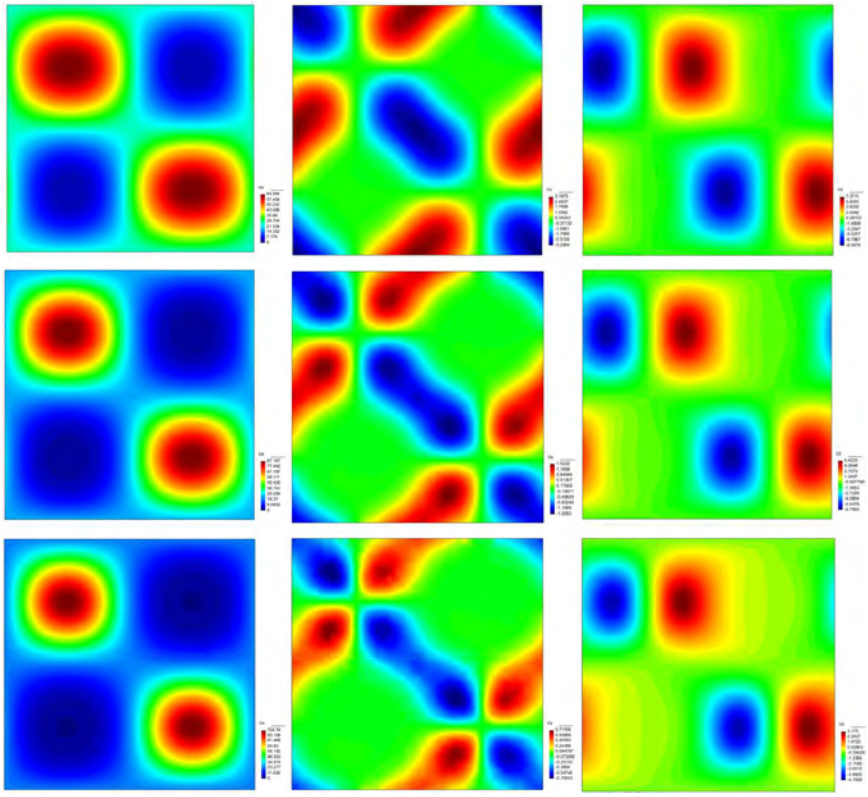

Figure 4: Numerical simulation results for Experiments. From left to right: the components u_1, u_2, and u_3 of the top-surface velocity; from top to bottom: $L = 40$, 80, $160\,km$.

7 Preliminary results of Greenland ice-sheet simulations

In this section, we present the results of some preliminary Greenland ice-sheet simulations based on the full-Stokes model, using our parallel, finite element, three-dimensional solvers for the velocity, pressure, temperature, and ice-sheet thickness.

7.1 Input data and parameters

Observational data with $5\,km$-resolution of current bedrock topography and ice thickness of the Greenland ice-sheet are presented in Figure 7. We use the observational data to define the horizontal extent Ω_H of the ice sheet. We also use the observational bedrock topography data to define the ice-bedrock boundary $z_{ib}(x, y)$, then add the observational thickness data to define the initial ice-atmosphere boundary $z_{ia}(x, y, 0)$, and, of course, use the thickness data to define the initial ice-sheet thickness $H_0(x, y)$. The maximum and minimum values of the ice thickness are $3340.553\,m$ and $5\,m$, respectively. The peak values of bedrock topography and ice thickness do not occur at the same location; the largest value of the ice elevation, i.e., of the ice-atmosphere boundary, is $3536.0m$. The observational data of the average ice surface temperature is also given in Figure 7. We note that the temperature field is a NPEC re-analysis product. As shown in the right of Figure 7, it is also clear that the current temperature in some boundary areas of the Greenland ice-sheet is over 2.0^oC, already in the melting state.

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

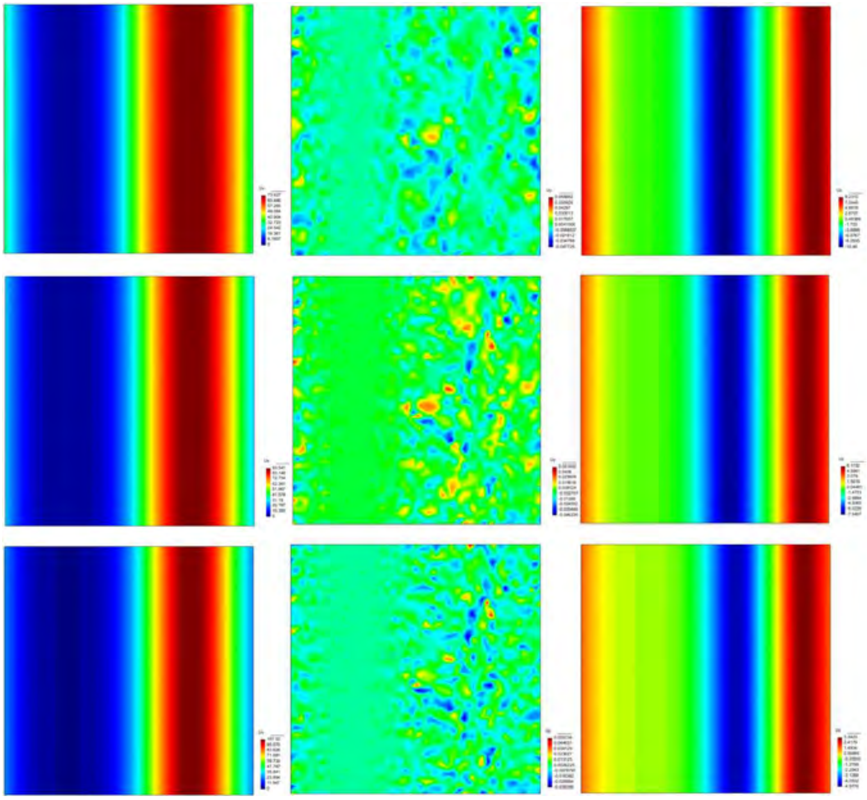

Figure 5: Numerical simulation results of Experiments B. From left to right: the components u_1, u_2, and u_3 of the top-surface velocity; from top to bottom: $L = 40$, 80, $160\,km$.

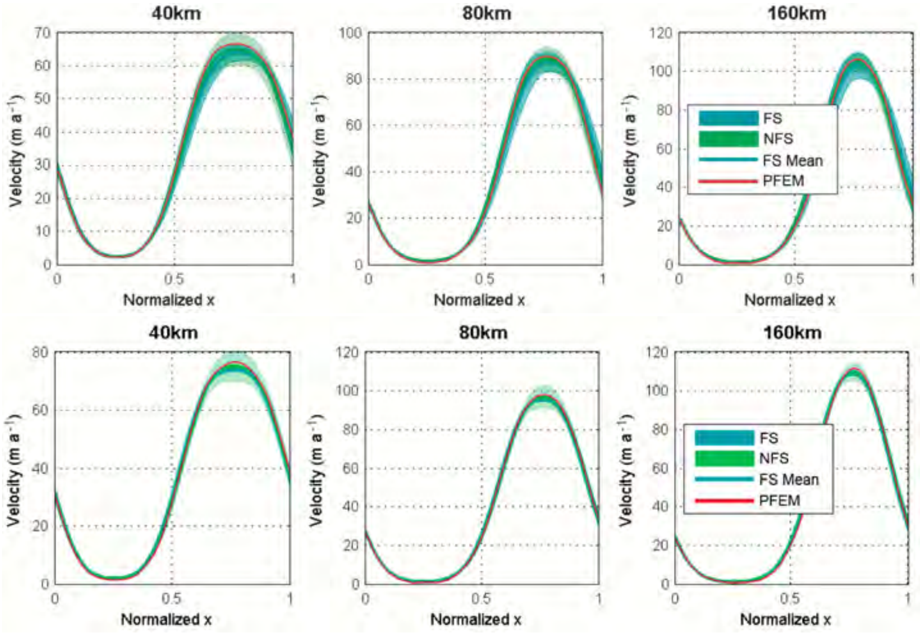

Figure 6: The horizontal speed of the ice flow at the top surface vs. x at $y = L/4$ for Experiments A (top row) and B (bottom row) and for $L = 40$, 80, $160\,km$ (from left to right).

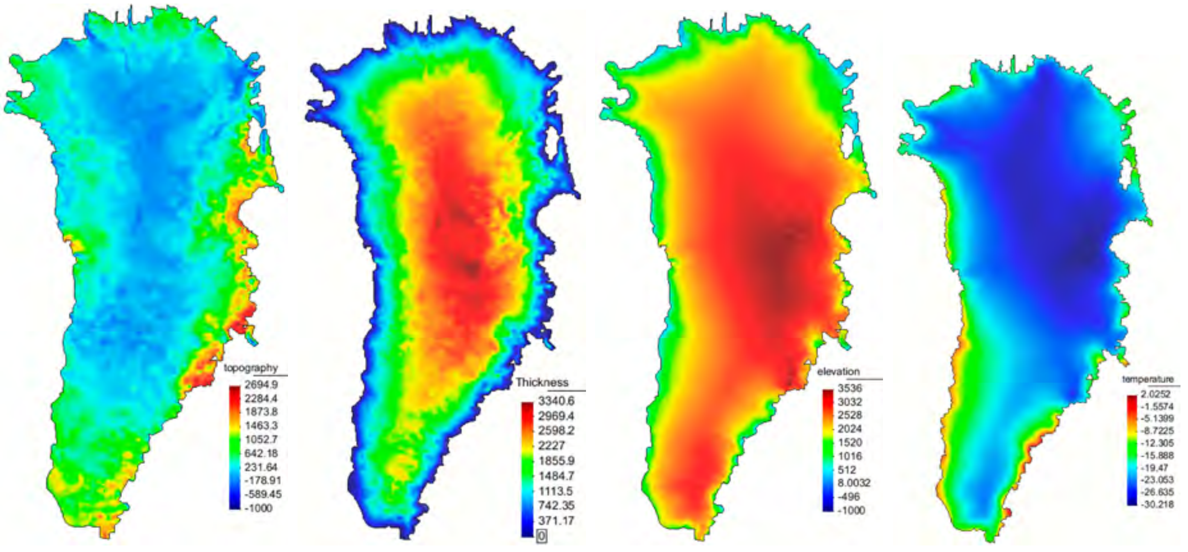

Figure 7: Observational data with 5 km-resolution of the Greenland ice sheet. From left to right: topography of the bedrock (the color indicates the elevation), ice thickness, elevation of the ice surface, and the ice surface temperature.

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 along the z-direction, resulting in a total total 866,918 nodes and 1,603,125 prismatic elements; see Figure 2 for an illustration. Note that 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 thickness change from one horizontal grid point to another. In Table 3, we list all the parameters and constants used for our Greenland ice-sheet simulations.

7.2 Simulations for 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) Glen's flow law rate $A = 10^{-16}$ and the power law exponent $n = 3$. This case also provides a baseline for comparing, in Section 7.3, with results for a temperature-dependent effective viscosity.

The no-slip boundary condition at the bedrock boundary and the zero Dirichlet condition on the lateral boundaries are imposed. We note that the zero Dirichlet boundary condition for the Greenland ice-sheet simulation was first used in [35] as a tentative choice, but it is not realistic. Thus the resulting approximate velocity is not accurate for regions very close to the Greenland ice-sheet 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 Figure 8; 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.

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

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	^oC
activation energy $(T < 263K)$	Q	60×10^3	$Jmol^{-1}$
activation energy $(T > 263K)$	Q	139×10^3	$Jmol^{-1}$
flow law constant $(T < 263K)$	a	1.3×10^{-5}	$Pa^{-3}a^{-1}$
flow law constant $(T > 263K)$	a	6.26×10^{10}	$Pa^{-3}a^{-1}$
second per year	spy	31556926	s/a
ice density	ρ	918.0	kgm^{-3}
gravitational constant	g	9.81	ms^{-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	$Jmol^{-1}K^{-1}$
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	Wm^{-3}

Note that in (7.1), $\sqrt{u_1^2 + u_2^2}$ is the surface speed. Comparing v_{ia} with the bed-rock topography and ice thickness data shown in Figure 7 shows that, to a large extent, it is closely related to the elevation of the ice sheet and its thickness distribution, as expected. This has already been observed by the ice-sheet modeling community [32, 34–36].

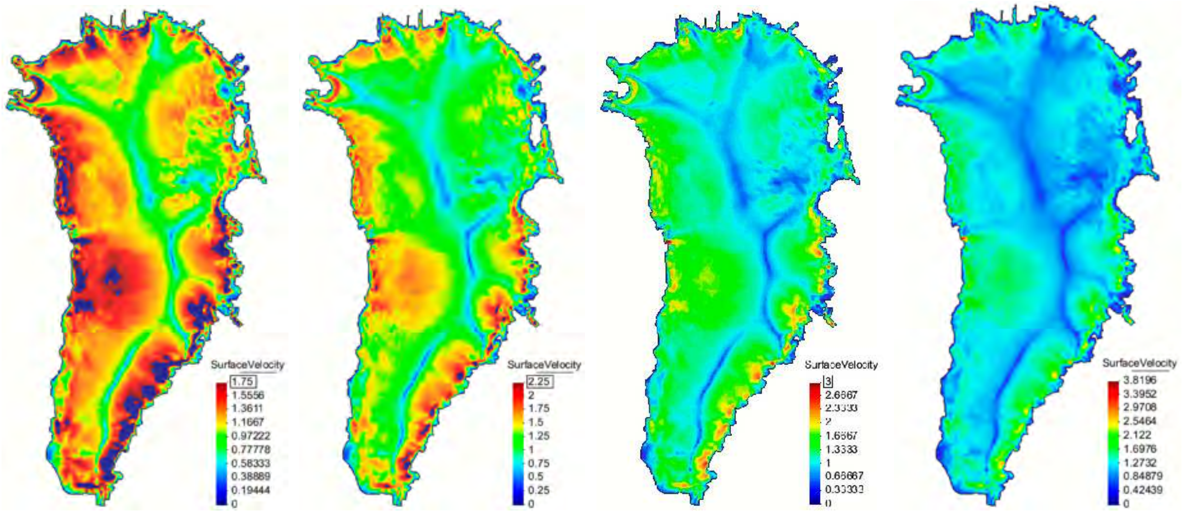

Figure 8: Distribution of v_{ia} for the Greenland ice-sheet by using the full-Stokes model with a constant Glen's flow law rate $A = 10^{-16}$ and the power law exponent $n = 3$. For better visualization, we set the upper limit of the values displayed to each of the values 1.75, 2.25, 3, and $\max v_{ia}$ which correspond to surface speeds of 55.23, 176.83, 999, and $6,600\,ma^{-1}$, respectively.

7.3 Simulations for a temperature-dependent effective viscosity

To start, we need a good approximation of the current temperature field throughout the Greenland ice sheet. As mentioned above, we have the observational and re-analysis data for the ice surface temperature. The bottom bedrock temperature is set to be −36°C everywhere. Thus, we can obtain an approximate temperature field of the Greenland ice sheet by using linear interpolation on the prismatic mesh based on these data. However, this approximation is prone to roughness and is not physically realistic. In order to obtain a better approximation, the interpolated temperature field is adopted as an initial conditions for (4.10) and we also set $\mathbf{u} = 0$ and assume that there $q = 0$ and $q_{ib} = 0$. We then solve (4.10) which is now a pure diffusion problem with the fixed Dirichlet boundary condition determined from the observational data and use a time step $\Delta t = 0.5$ year until a steady state is obtained. We found that this takes about 10 years in our simulations [18]. This results in the smoother temperature field shown in the left of Figure 11 which we use as the initial condition for the temperature for our subsequent temperature-dependent evolution simulations.

As a preliminary experiment, we ran a one-year evolution simulation for the Greenland ice sheet. The simulation procedure is described as follows.

- We use the current Greenland ice-sheet thickness to define the initial computational domain and the smoothed initial interpolated temperature field described above as inputs into the discretized full-Stokes system (4.8). 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 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 be 12. This step results in new approximate velocity and pressure fields.

- Next, We use the approximate velocity field determined in the previous step as input into the discrete energy equation (4.10). The heat source term q and the strain heating term are set to zero. We take a time step of two months and run a one-year simulation. The values of the parameters used for (4.10) are $c = 2093$ J(kg°C)$^{-1}$ and $\kappa = 2.2$ W(m°C)$^{-1}$. The geothermal heat flux from the bedrock is given by $\alpha = 0.055 W/m^2$. We used 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 sub-domain is 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.

- Then the new height of the ice sheet is updated from (4.11), again using the approximate velocity field determined in the first step as input.

We are specially interested in the current rate of change of the Greenland ice-sheet movement, temperature, and thickness.

7.3.1 Surface velocity

We want to examine how the temperature affects the surface speed by comparing to the results of Section 7.2. The power-law exponent n is still set to 3, but now the Glen's flow law coefficient A is temperature-dependent as in (2.9). The rheology coefficient a and ice activation energy Q are also fully temperature dependent as shown in Table 3. When the temperature is below $263\,K$, a and Q are 60×10^3 and 1.3×10^{-5}, respectively. However, when the temperature is greater than $263\,K$, they are 139×10^3 and 6.26×10^{10}, respectively. The simulation results for v_{ia} in such a setting are displayed in Figure 9. As was the case for Figure 8, we again use different ranges for display purposes.

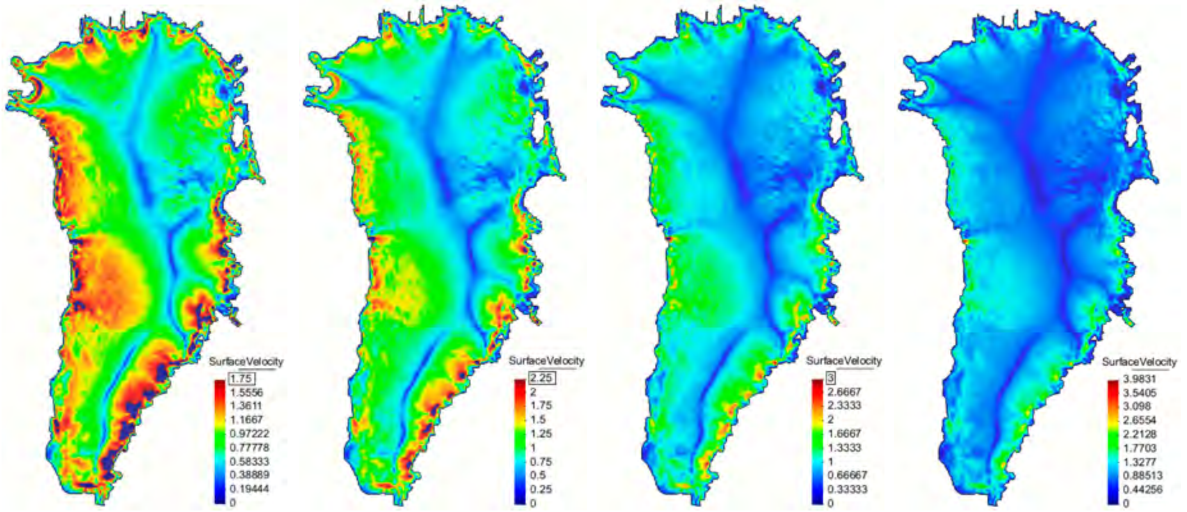

Figure 9: Distribution of v_{ia} for the Greenland ice-sheet using the full-Stokes model with a temperature-dependent Glen's flow law rate A with power law exponent $n = 3$. For better visualization, we set the upper limit of the values displayed to each of the values 1.75, 2.25, 3, and $\max v_{ia}$ which correspond to surface speeds of 55.23, 176.83, 999, and $9,617ma^{-1}$, respectively.

For the most part, the values of the surface speed obtained using the temperature-dependent Glen's flow rate model are even far below $55.23ma^{-1}$ (shown on the left of Figure 9); for the temperature independent rate, the surface velocity values within this range are only located at the center region of the Greenland ice-sheet (shown in the left of Figure 8). At locations of the Greenland ice sheet where the bedrock topography has large gradients and high surface temperatures, the ice velocity is relatively large. We found the largest surface speeds are $v_{ia} = 3.98$ and $v_{ia} = 3.82$ for the full-Stokes model with the temperature-dependent and constant flow rates, respectively. These values appear at some tiny spots very close to the Greenland boundary as shown by Figure 9. We also note that these values very possibly overestimate the real ice-sheet velocity at these spots partially caused by the zero Dirichlet condition on the lateral boundary and some inaccuracies of the observation data we used due to low resolution.

To more clearly clarify their difference, we calculated the surface speed differences between the two cases

$$v_{diff} = \log_{10}\left(1.0 + \sqrt{\triangle u_1^2 + \triangle u_2^2}\right), \tag{7.2}$$

where $\triangle u_1$ and $\triangle u_2$ are the difference in the velocity components in x and y-directions, respectively, between the two models. The values of v_{diff} are visualized in Figure 10.

It seems that the temperature distribution does have strong effects on the surface velocity. A small surface velocity change ($v_{diff} = [0 \sim 5ma^{-1}]$) occurs in the low-temperature region in the interior of the Greenland ice-sheet where the gradient of the elevation is relatively small and at the marginal rim where the ice thickness is relatively small (shown in the left of Figure 10). When the gradient of the ice-sheet elevation is large and the temperature in these areas is relatively high, the surface velocity change becomes large as well. As shown in the middle-left of Figure 10, the surface velocity difference is approximately $v_{diff} = [5.0 \sim 10.0ma^{-1}]$. If the ice sheet is thick and the elevation gradient is small and if the ice temperature is high, the difference in surface speed will

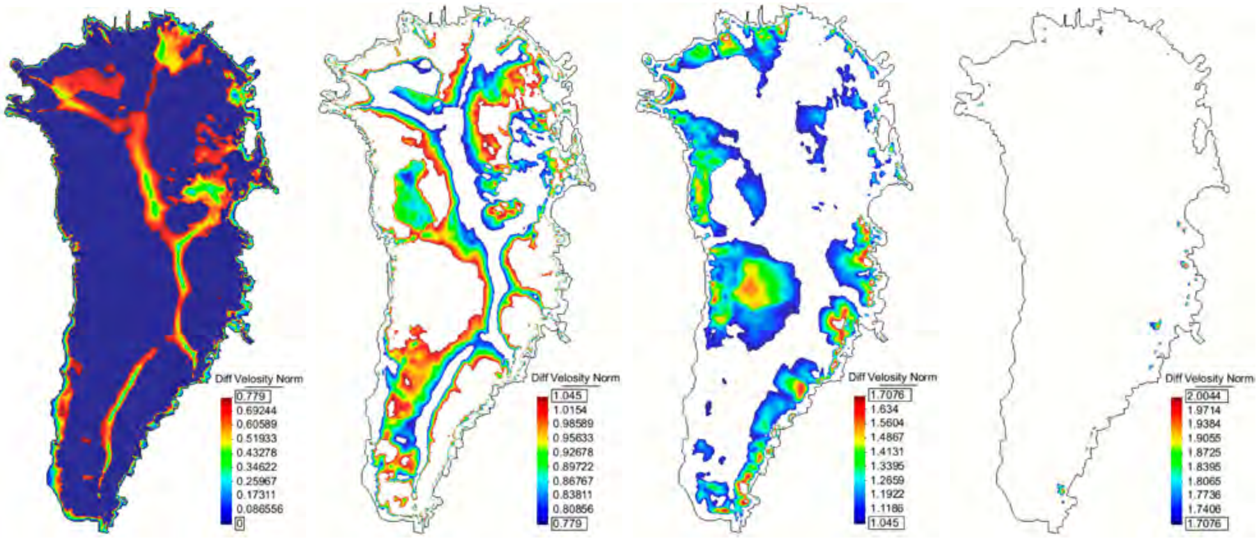

Figure 10: Distribution of v_{diff} between the surface velocities computed using the full-Stokes models with the temperature-dependent and constant Glen's flow law rate. For better visualization, we set the upper limit of the values displayed to each of the values 0.779, 1.045, 1.7076, and 2.0044 which correspond to surface speed differences of 5, 10, 50, and $1,000\,ma^{-1}$, respectively.

be dramatically large. This is typically displayed by the middle-right plot of Figure 10; the surface speed difference can reach up to $v_{diff} = [10.0 \sim 100.0 ma^{-1}]$. However, the effect of the temperate on the surface speed seldom exceeds $100.0ma^{-1}$ as shown in the right plot of Figure 10. For more details on the role of the temperature, see the discussions in Section 7.4.

7.3.2 Temperature evolution

We conducted several numerical experiments to test our parallel solver for the discretized energy equation (4.6). In [33], 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 stablized temperature field is adopted as the initial condition for (4.6) and the discretized convection velocity field is obtained from (4.8). We also assume there are no internal heat sources due strain heating and focus on examining the effects that convection and the basal heat flux have on the evolution of the temperature field. Adiabatic conditions are imposed on the top surface.

We first tested our parallel solver assuming there is no bedrock heat flux. The corresponding simulated temperature field after one year is shown in the middle-left of Figure 11; its difference from the initial temperature is shown in the middle-right of Figure 11. We see significant changes in the temperature profile only at small spots where the peak values of the surface speed are located. For the second case, we also compared the evolution of temperature field guided by (4.6) without and with a basal heat flux term but with the convection term activated. In this experiment, 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 found that the maximum value of the temperature difference of these two cases is only about $0.0492K$ and the average temperature difference is $5.7087 \times 10^{-5}K$. The detailed differences are displayed in the right plot of Figure 11. Thus, it seems that the influence of basal heat flux may actually be very small, relative to convection effects, and could be neglected over short time periods.

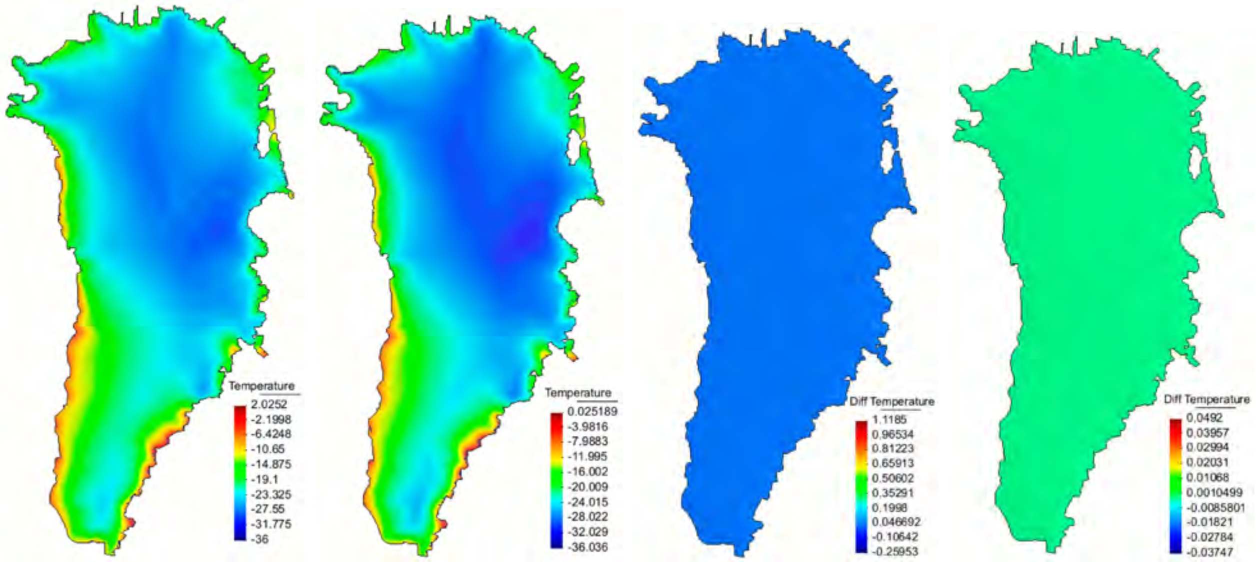

Figure 11: Simulation results for the temperature evolution of the Greenland ice-sheet after one year. Left: the initial temperature field after stabilization by a pure difussion process; middle-left: the simulated evolved temperature field after one year with convection only; middle-right: the variation between the inital temperature field and the simulated evolved temperature field after one year with convection only; right: the differences between the simulated evolved temperature fields after one year with and without the bedrock heat flux.

7.3.3 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 $\delta t = 1$ year. We find that, for the most part, the ice-sheet thickness change is negative. To clearly display the change of the free ice-sheet surface, we only display the decreasing part of the ice-sheet change in Figure 12. Is is easy to see that the average ice-sheet thickness change per year is mostly within the range of $[0.0 \sim 1.5]ma^{-1}$ (the first two figures in Figure 12). There are lots of small regions close to the Greenland boundary for which the ice-sheet thickness change is much larger than $1.5ma^{-1}$.

As the viscosity, pressure, and velocity fields converge over the course of the iterative process, we can update the domain geometry at one-year time steps. We note that this frequency is adequate to model the evolution of the ice-sheet surface topography. Small interpolation errors occur during the re-gridding of the existing finite element mesh but 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 precess of the ice-sheet evolution simulation becomes self-consistent.

7.4 Sensitivity of the temperature

To investigate how temperature affect the ice-sheet surface speed and thickness evolution, we carried out several further numerical experiments. We consider uniform decreases of $1.0°C$, $1.5°C$, and $2.0°C$ from the current observational temperature field of the Greenland ice-sheet. We repeated the simulations of the previous section with the new modified temperature fields; the computational

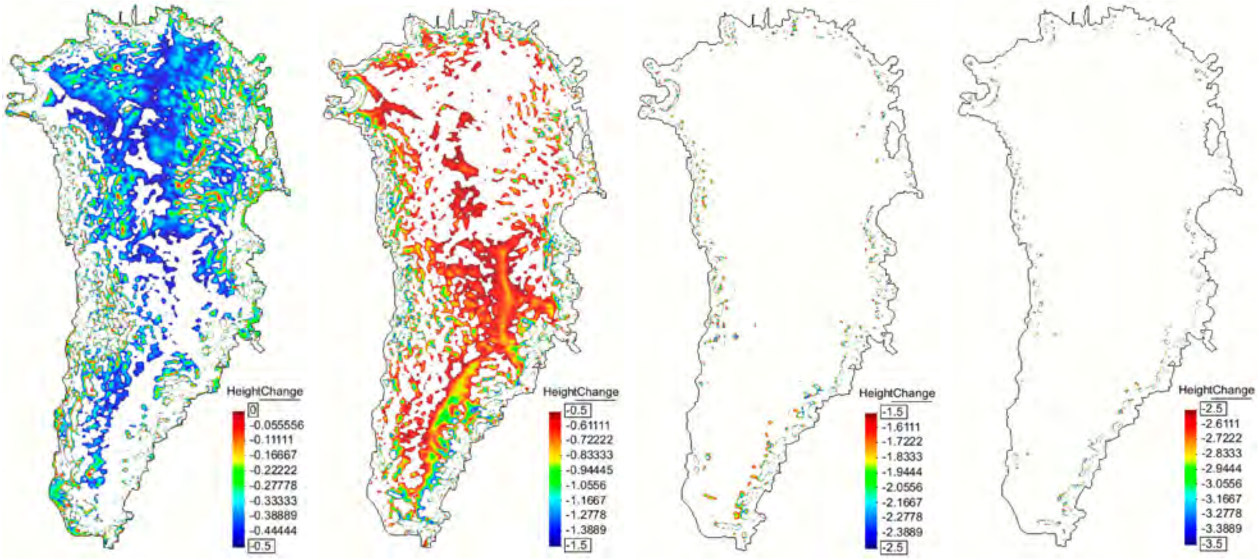

Figure 12: Simulated ice-thickness change of the Greenland ice-sheet after one year; the values plotted are restricted to different ranges to aid in the visualization and interpretation.

results are displayed in Figures 13 (which should be compared to Figure 10) and 14 (which should be compared to Figure 12). Note that we did not test cases with an increased temperature field because we found the nonlinear coefficient of $\eta_{\mathbf{u}}$ to be extremely temperature sensitive. The current temperature in some areas of the Greenland ice-sheet is over $2.0^o C$, already in the melting state shown in the right Figure 7. Even a $1.0°C$ increase in the temperature field results in the divergence of our parallel, three-dimensional full-Stokes solver. Note that in these calculations, the geometry and thermal properties of the ice sheet are held fixed over time.

From Figure 13, we can see that the speed increases with increasing temperature. Although the overall configurations of the surface velocity field for each temperature field are almost the same, as the temperature increases becomes higher, this tendency will be further intensified. With the same limits on the values plotted, the surface speed difference v_{diff} within the range of $[0.0 \sim 2.5] ma^{-1}$ is clearly shown in the first column of Figure 14.

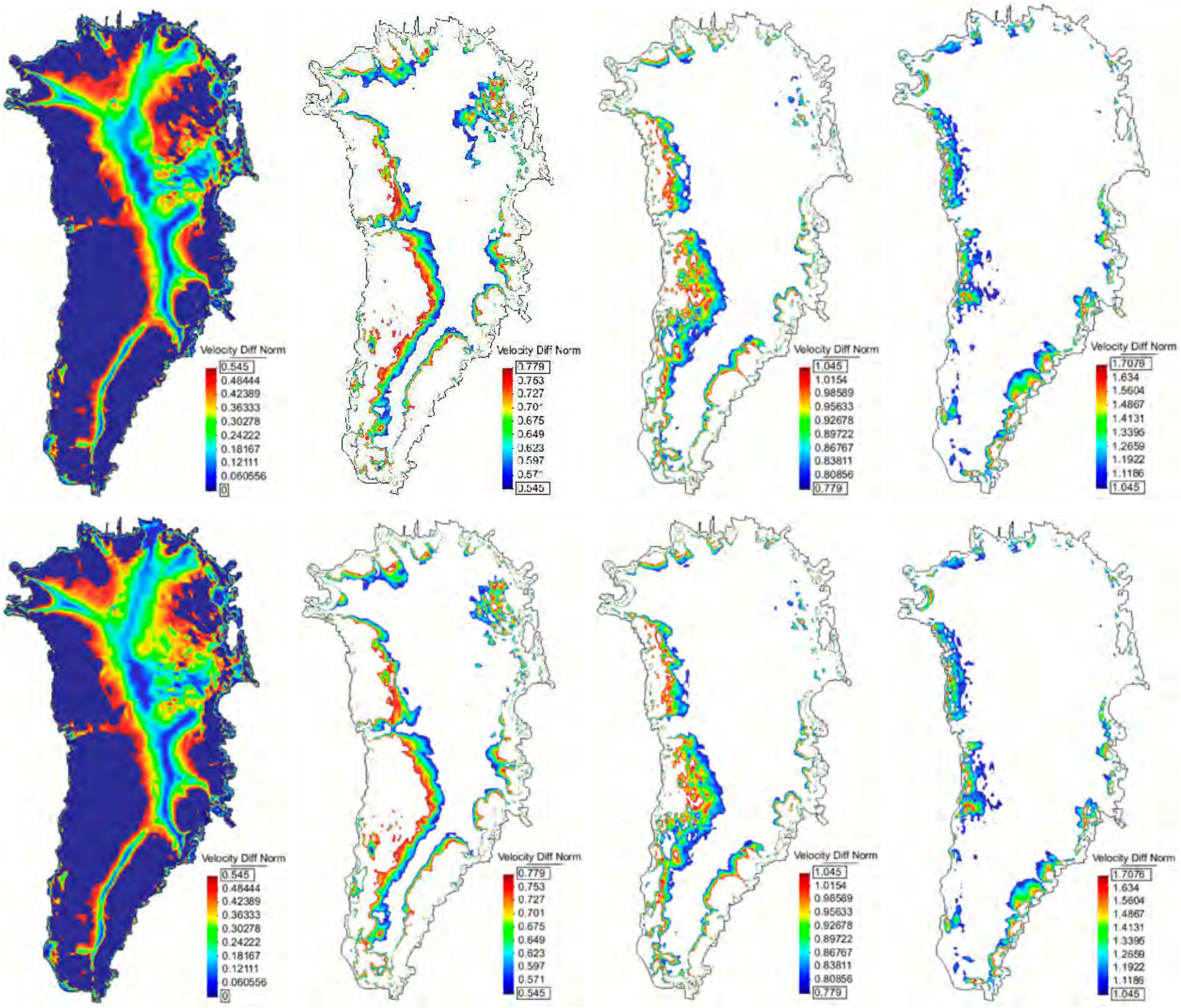

Figure 13: Distribution of v_{diff} with a $-2.0°C$ (top row) and $-1.0°C$ (bottom row) drop in the current observational temperature data. In the plots, the limits on the values plotted are 0.545, 0.779, 1.045 and 1.7076 which represent surface speed differences of 2.5, 5.0, 10.0, and $50.0ma^{-1}$, respectively.

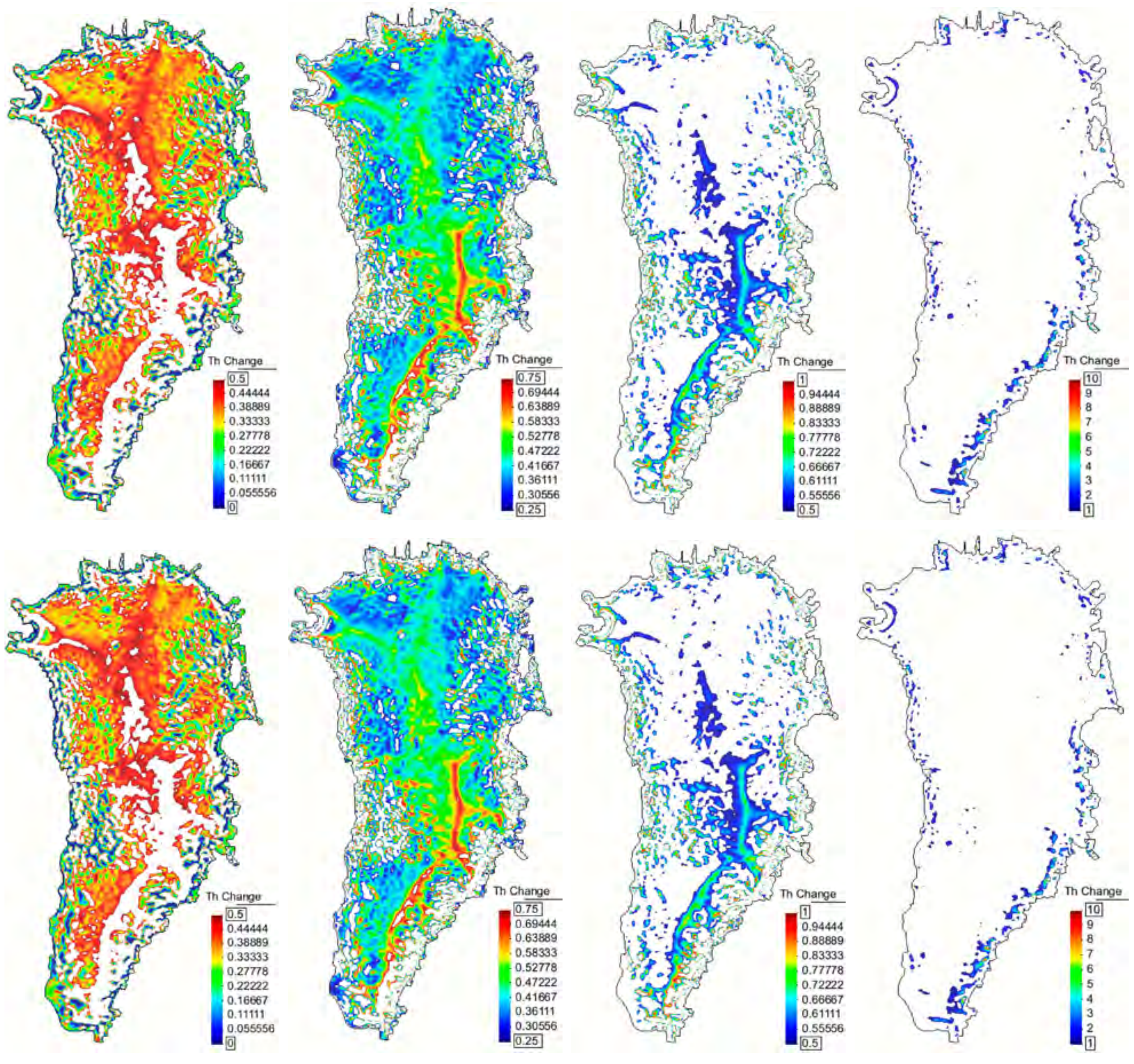

Figure 14: Simulated ice thickness change of the Greenland ice sheet after one year with $-2.0°C$ (**top row**) and $-1.0°C$ (bottom row) drop in the current observational temperature data.

8 Concluding remarks

We put forward and analyzed the parallel finite element schemes for the discretization of governing partial differential equations on ice-sheet evolution model. We introduced and discussed finite element approximations of the boundary conditions of the momentum, energy and free surface evolution equations. The full 3D high-quality nonuniform CVDT meshes were utilized to construct the realistic model of the Greenland ice-sheet. We conducted computational experiments to illustrate the accuracy and good performance of our parallel solver developed in conjunction with the finite element methods and their flexible numerical schemes. For complicated boundary conditions and geometries, the model compares well with full-stress solutions obtained by previous authors. Our numerical experiments showed that our parallel finite element simulations agreed well with the benchmark results of the current ice simulation community.

An unique aspect of our parallel finite element solvers of the ice-sheet evolution modeling is its feasibility of using the realistic data from observations, which simplifies the implementation of complex boundary conditions of different kinds of partial differential equations, leads to straightforward numerical investigations, and finally results in intuitive visualization of the scientific calculation data. This can enable us to perform the realistic case study of the ice-sheet evolution on ice-sheet or glaciers, such as Greenland and etc.

References

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

[2] Ainsworth, M. (1996). A preconditioner based on domain decomposition for h-p finite-element approximation on quasi-uniform meshes, *SIAM Journal on Numerical Analysis*, **33**(4), pp. 1358-1376.

[3] Amestoy, P.R., Duff, I.S., L'Excellent, J.Y., and Koster, J. (2001). A fully asynchronous multifrontal solver using distributed dynamic scheduling, *SIAM Journal on Matrix Analysis and Applications*, **23**(1), pp. 15-41.

[4] Axelsson, O., Bai, Z.Z., and Qiu, S.X. (2004). A class of nested iteration schemes for linear systems with a coefficient matrix with a dominant positive definite symmetric part, *Numerical Algorithms*, **35**(2-4), pp. 351-372.

[5] Axelsson, O. and Larin, M. (1998). An algebraic multilevel iteration method for finite element matrices, *Journal of Computational and Applied Mathematics*, **89**(1), pp. 135-153.

[6] Bamber, J., Ekholm, S., and Krabill, W. (2001). A new, high resolution digital elevation model of Greenland fully validated with airborne laser altimeter data, *J. Geophys. Res.*, **106** (B4), pp. 6733-6745.

[7] Bengtsson, L. (1994). Climate of the 21st-Century, *Agricultural and Forest Meteorology*, **72**(1-2), pp. 3-29.

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

[9] Bromwich, D.H., Bai, L.H., and Bjarnason, G.G. (2005). High-resolution regional climate simulations over Iceland using Polar MM5, *Monthly Weather Review*, **133**(12), pp. 3527-3547.

[10] Cao, J.W. (2005). An efficient and effective nonlinear solver in a parallel software for large scale petroleum reservoir simulation, *International Journal of Numerical Analysis and Modeling*, **2**, pp. 15-27.

[11] Douglas, C.C. (1996). Multigrid methods in science and engineering, *IEEE Computational Science & Engineering*, **3**(4), pp. 55-68.

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

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

[14] Du, Q. and Ju, L. (2005). Finite volume methods on spheres and spherical centroidal Voronoi meshes, *SIAM Journal on Numerical Analysis*, **43**, pp. 1673-1692.

[15] Gerya, T.V., and Yuen, D.A. (2003) Rayleigh-Taylor instabilities from hydration and melting propel "cold plumes" at subduction zones. Earth and Planet Sci. Lett., 212, 47-62.

[16] Gregory, J.M. and Huybrechts, P. (2006). Ice-sheet contributions to future sea-level change, *Philosophical Transactions of the Royal Society-Mathematical, Physical and Engineering Sciences*, **364**, pp. 1709-1731.

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

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

[19] Holland, D.M., Ingram, R.G., Mysak, L.A., and Oberhuber, J.M. (1995). A numerical simulation of the sea-ice cover in the Northern Greenland sea, *Journal of Geophysical Research-Oceans*, **100**(C3), pp. 4751-4760.

[20] Holland, D.M., Mysak, L.A., Manak, D.K., and Oberhuber, J.M. (1993). Sensitivity study of a dynamic thermodynamic sea ice model, *Journal of Geophysical Research-Oceans*, **98**(C2), pp. 2561-2586.

[21] Ito, K. and Toivanen, J. (2006). Preconditioned iterative methods on sparse subspaces, *Applied Mathematics Letters*, **19**(11), pp. 1191-1197.

[22] Ju, L. (2007). Conforming centroidal Voronoi Delaunay triangulation for quality mesh generation, *International Journal of Numerical Analysis and Modeling*, **4**, pp. 531-547.

[23] Ju, L. and Du, Q. (2009). A finite volume method on general surfaces and its error estimates, *Journal of Mathematical Analysis and Applications*, **352**(2), pp. 645-668.

[24] Ju, L., Gunzburger, M., and Zhao, W.-D. (2006). Adaptive finite element methods for elliptic PDEs based on conforming centroidal Voronoi Delaunay triangulations, *SIAM Journal on Scientific Computing*, **28**, pp. 2023-2053.

[25] Ju, L., Lee, H.-C., and Tian, L. (2008). Numerical simulations of the steady Navier-Stokes equations using adaptive meshing schemes, *International Journal for Numerical Methods in Fluids*, **56**(6), pp. 703-721.

[26] Le Meur, E., Gagliardini, O., Zwinger, T., and Ruokolainen, J. (2004). Glacier flow modelling: A comparison of the shallow ice approximation and the full-Stokes solution. *Comptes Rendus Physique*, **5**(7), pp. 709-722.

[27] May, D.A. and Moresi L. (2008). Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics, *Phys. Earth Planet. Interiors*, **171** (1-4), pp. 33-47, DOI:10.1016/j.pepi.2008.07.036.

[28] Mernild, S.H., Liston, G.E., Hiemstra, C.A., and Steffen, K. (2008). Surface melt area and water balance modeling on the Greenland ice sheet 1995-2005, *Journal of Hydrometeorology*, **9**(6), pp. 1191-1211.

[29] Mernild, S.H., Liston, G.G., Kane, D.L., Knudsen, N.F., and Hasholt, B. (2008). Snow, runoff, and mass balance modeling for the entire Mittivakkat Glacier (1998-2006), Ammassalik Island, SE Greenland, *Geografisk Tidsskrift-Danish Journal of Geography*, **108**(1), pp. 121-136.

[30] Nick, F.M., Vieli, A., Howat, I.M., and Joughin, I. (2009). Large-scale changes in Greenland outlet glacier dynamics triggered at the terminus, *Nature Geoscience*, **2**(2), pp. 110-114.

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

[32] Pattyn, F. (2003). A new 3D higher-order thermomechanical ice-sheet model: Basic sensitivity, ice-stream development and ice flow across subglacial lakes, *J. Geophys. Res.*, **108**(B8), 2382, doi:10.1029/2002JB002329, 2003.

[33] Paterson, W. (1994). *The Physics of Glaciers*, Elsevier Science, Oxford, U. K.

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

[35] Price, S.F., Waddington, E.D., and Conway, H. (2007). A full-stress, thermomechanical flow band model using the finite volume method, *Journal of Geophysical Research-Earth Surface*, **112**, F03020, doi:10.1029/2006JF000724.

[36] Ringler, T., Ju, L., and Gunzburger, M. (2008). A multiresolution method for climate system modeling: Application of spherical centroidal Voronoi tessellations, *Ocean Dynamics*, **58**(5-6), pp. 475-498.

[37] Jung, M. (1997). On the parallelization of multi-grid methods using a non-overlapping domain decomposition data structure, *Applied Numerical Mathematics*, **23**(1), pp. 119-137.

[38] Karypis, G. (2003). Multi-constraint mesh partitioning for contact/impact computations, *Supercomputing'03*, Phoenix, Arizona, USA.

[39] Korneev, V.G. and Jensen, S. (1999). Domain decomposition preconditioning in the hierarchical p-version of the finite element method, *Applied Numerical Mathematics*, **29**(4), pp. 479-518.

[40] Li, X.Y. and Demmel, J.W. (2003). SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems, *ACM Transactions on Mathematical Software*, **29**(2), pp. 110-140.

[41] Li, Z.-Z., Saad, Y., and Sosonkina, M. (2003). pARMS: A parallel version of the algebraic recursive multilevel solver, *Numerical Linear Algebra with Applications*, **10**(5-6), pp. 485-509.

[42] Lv, X., Zhao, Y., Huang, X.-Y., Xia, G.-H., and Wang, Z.-J. (2006). An efficient parallel/unstructured-multigrid preconditioned implicit method for simulating 3D unsteady compressible flows with moving objects, *Journal of Computational Physics*, **215**(2), pp. 661-690.

[43] Mandel, J. and Tezaur, R. (1996). Convergence of a substructuring method with Lagrange multipliers, *Numerische Mathematik*, **73**(4), pp. 473-487.

[44] Michel, J., Pellegrini, F., and Roman, J. (1997). Unstructured graph partitioning for sparse linear system solving, *Solving Irregularly Structured Problems In Parallel*, Lecture Notes In Computer Science, pp. 273-286.

[45] Prudencio, E.E., Byrd, R., and Cai, X.C. (2006). Parallel full space SQP Lagrange-Newton-Krylov-Schwarz algorithms for PDE-constrained optimization problems, *SIAM Journal on Scientific Computing*, **27**(4), pp. 1305-1328.

[46] Rakowsky, N. (1999). The Schur complement method as a fast parallel solver for elliptic partial differential equations in oceanography, *Numerical Linear Algebra with Applications*, **6**(6), pp. 497-510.

[47] Saito, F. (2002). *Development of a Three Dimensional Ice Sheet Model for Numerical Studies of Antarctic and Greenland Ice Sheets*, Ph.D. thesis, Univ. of Tokyo, Tokyo, Japan.

[48] Shadid, J.N. and Tuminaro, R.S. (1992). Sparse iterative algorithm software for large-scale MIMD machines - An initial discussion and implementation. *Concurrency and Computation: Practice and Experience*, **4**(6), pp. 481-497.

[49] Valli, A.Q. (1999). *Domain Decomposition Methods for Partial Differential Equations.* Oxford Science Publications.

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