SAND2021-0728R

FROSCH PRECONDITIONERS FOR LAND ICE SIMULATIONS OF GREENLAND AND ANTARCTICA[*]

ALEXANDER HEINLEIN[†], MAURO PEREGO[‡], AND SIVASANKARAN RAJAMANICKAM[‡]

Abstract. Numerical simulations of Greenland and Antarctic ice sheets involve the solution of large-scale highly nonlinear systems of equations on complex shallow geometries. This work is concerned with the construction of Schwarz preconditioners for the solution of the associated tangent problems, which are challenging for solvers mainly because of the strong anisotropy of the meshes and wildly changing boundary conditions that can lead to poorly constrained problems on large portions of the domain. Here, two-level GDSW (Generalized Dryja–Smith–Widlund) type Schwarz preconditioners are applied to different land ice problems, i.e., a velocity problem, a temperature problem, as well as the coupling of the former two problems. We employ the MPI-parallel implementation of multi-level Schwarz preconditioners provided by the package FROSch (Fast and Robust Schwarz) from the Trilinos library. The strength of the proposed preconditioner is that it yields out-of-the-box scalable and robust preconditioners for the single physics problems.

To our knowledge, this is the first time two-level Schwarz preconditioners are applied to the ice sheet problem and a scalable preconditioner has been used for the coupled problem. The preconditioner for the coupled problem differs from previous monolithic GDSW preconditioners in the sense that decoupled extension operators are used to compute the values in the interior of the subdomains. Several approaches for improving the performance, such as reuse strategies and shared memory OpenMP parallelization, are explored as well.

In our numerical study we target both uniform meshes of varying resolution for the Antarctic ice sheet as well as non uniform meshes for the Greenland ice sheet are considered. We present several weak and strong scaling studies confirming the robustness of the approach and the parallel scalability of the FROSch implementation. Among the highlights of the numerical results are a weak scaling study for up to 32 K processor cores (8 K MPI-ranks and 4 OpenMP threads) and 566 M degrees of freedom for the velocity problem as well as a strong scaling study for up to 4 K processor cores (and MPI-ranks) and 68 M degrees of freedom for the coupled problem.

Key words. domain decomposition methods, monolithic Schwarz preconditioners, GDSW coarse spaces, multiphysics simulations, parallel computing

AMS subject classifications. 65F08, 65Y05, 65M55, 65N55

1. Introduction.

Greenland and Antarctic ice sheets store most of the fresh water on earth and mass loss from these ice sheets significantly contributes to sea-level rise (see, e.g. [11]). In this work, we propose overlapping Schwarz domain decomposition preconditioners for efficiently solving the linear systems arising in the context of ice sheet modeling.

We first consider the solution of the ice sheet momentum equations for computing the ice velocity. This problem is at the core of ice sheet modeling and has been largely addressed in literature and several solvers have been considered [40, 6, 18, 35, 50, 19, 10, 9]. Most solvers from the literature rely on Newton-Krylov methods, using, e.g., the conjugate gradient (CG) [31] or the generalized minimal residual (GMRES) [44] method as the linear solver, and either one-level

[*] Submitted to the editors DATE.

[†] Institute for Applied Analysis and Numerical Simulation, University of Stuttgart, Germany. Department of Mathematics and Computer Science, University of Cologne (alexander.heinlein@uni-koeln.de). Center for Data and Simulation Science, University of Cologne (http://www.cds.uni-koeln.de).

[‡] Center for Computing Research, Scalable Algorithms Department, Sandia National Laboratories (mperego@sandia.gov, srajama@sandia.gov). Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Schwarz preconditioners, hierarchical low-rank methods, or multigrid preconditioners to accelerate the convergence. In particular, the ones that have been demonstrated on problems with hundreds of millions of unknowns [6, 35, 50, 19, 10] use tailored multigrid preconditioners or hierarchical low-rank methods. Multigrid preconditioners [6, 35, 50, 19] require careful design of the grid transfer operators for properly handling the anisotropy of the mesh and the basal boundary conditions that range from no-slip to free-slip. Hierarchical low-rank approaches have also been used for the velocity problem [10, 9]. Chen et al. [10] developed a parallel hiearchical low-rank preconditioner that is aysmptotically scalable, but it has a large constant overhead and the trade-off between memory usage and solver convergence does not make it an ideal choice for the large problems considered here. The hierarchical low-rank approach that showed the most promise in terms of solver scalability is a sequential implementation limiting its usage to small problems [9].

In addition to the velocity problem, we also consider the problem of finding the temperature of an ice sheet using an enthalpy formulation ([1, 46, 32]) and the steady-state thermo-mechanical problem coupling the velocity and the temperature problems. The robust solution of this coupled problem is crucial for finding the initial thermo-mechanical state of the ice sheet under the assumption that the problem is almost at thermodynamic equilibrium. In fact, the initial state is estimated solving a PDE-constrained optimization problem where the loss function is the mismatch with observations and the constraint is the coupled velocity-temperature problem considered here. To our knowledge, while there are works in the literature targeting the solution of unsteady versions of the coupled problem ([5, 39, 43]), none of them targets the steady thermo-mechanical problem at the ice sheet scale.

Both the velocity problem and the coupled velocity-temperature problem are characterized by strong nonlinearities and anisotropic meshes (due to the shallow nature of ice sheets). The coupled problem presents additional complexities due to the different nature of the velocity and temperature equations, the former being a purely diffusive elliptic problem, whereas the second is an advection dominated problem. In our experience, the naive use of multigrid methods leads to convergence failure for the coupled problem.

Our approach is to employ a preconditioning framework based on two-level Schwarz methods with GDSW (Generalized Dryja–Smith–Wildund) [12, 13, 22, 23] type coarse spaces. To our knowledge, scalable domain decomposition methods such as the GDSW preconditioner used in this work have not been shown to work on the ice sheet problems. The main contributions of this work are:

- We demonstrate that two-level Schwarz preconditioners such as GDSW (Generalized Dryja–Smith–Widlund) type preconditioners work out-of-the-box to solve two single physics problems (the velocity problem and the temperature problem) on land ice simulations.
- We introduce a scalable two-level preconditioner for the coupled problem that is tailored for the coupled problem by decoupling the extension operators to compute the values in the interior of the subdomains.
- We present results using an MPI-parallel implementation of multi-level Schwarz preconditioners provided by the package FROSch (Fast and Robust Schwarz) from the Trilinos software framework.
- Finally, we demonstrate the scalability of the approach with several weak and strong scaling studies confirming the robustness of the approach and the parallel scalability of the FROSch implementation. We conduct a weak scaling study for up to 32 K processor cores and 566 M degrees of freedom for

the velocity problem as well as a strong scaling study for up to 4 K processor cores and 68 M degrees of freedom for the coupled problem. We compare against the multigrid method in [48, 50] for the velocity problem.

The remainder of the paper is organized as follows. Sections 2 and 3 introduces the ice sheet problems and the finite element discretization used in this study. We describe the Schwarz precondtioners, the reuse strategies for better performance and the way we tailor the preconditioner for the coupled problem in Section 4. Our software framework, which is based on Albany and FROSch, is briefly described in Section 5. Finally, the scalability and the performance of the proposed preconditioners are shown in Section 6.

2. Mathematical model.

At the scale of glaciers and ice sheets, ice can be modeled as a very viscous shear-thinning fluid with a rheology that depends on the ice temperature. Complex phenomena like the formation of crevasses and ice calving would require more complex damage mechanics models, however the fluid description accounts for most of the large scale dynamics of ice sheets and it is adopted by all ice sheet computational models. The ice temperature depends on ice flow (velocity/deformation). Given the large characteristic time scale of the temperature evolution, it is reasonable to assume the temperature to be relatively constant over a few decades and solve the flow problem uncoupled from the temperature problem. However, when finding the initial state of an ice sheet (by solving an inverse problem) it is important to consider the coupled flow/temperature model, to find a self consistent initial thermo-mechanical state. In this case, we assume the ice temperature to be almost in steady-state. Therefore, in this paper, we consider a steady-state temperature solver. In this section, we first introduce separately the flow model and the temperature model and then the coupled model.

2.1. Flow model.

We model the ice as a very viscous shear-thinning fluid with velocity $\boldsymbol{u}$ and pressure p satisfying the Stokes equations:

$$\begin{cases} -\nabla\cdot\sigma(\boldsymbol{u},p) & = \rho_i\,\boldsymbol{g}, \\ \nabla\cdot\boldsymbol{u} & = 0, \end{cases}$$

where $\boldsymbol{g}$ is the gravity acceleration, ρ_i the ice density and σ the stress tensor. In what follows, we use the so called first-order (FO) or Blatter-Pattyn approximation of the Stokes equations derived using scaling arguments based on the fact that ice sheets are shallow. Following [42] and [47], we have

$$\begin{cases} -\nabla\cdot(2\mu\,\dot{\boldsymbol{\epsilon}}_1) & = & -\rho_i\,g\,\partial_x s, \\ -\nabla\cdot(2\mu\,\dot{\boldsymbol{\epsilon}}_2) & = & -\rho_i\,g\,\partial_y s, \end{cases} \tag{2.1}$$

where x and y are the horizontal coordinate vectors in a Cartesian reference frame, $s(x,y)$ is the ice surface elevation, $g=|\boldsymbol{g}|$, and $\dot{\boldsymbol{\epsilon}}_1$ and $\dot{\boldsymbol{\epsilon}}_2$ are given by

$$\dot{\boldsymbol{\epsilon}}_1 = \begin{pmatrix} 2\dot\epsilon_{xx}+\dot\epsilon_{yy}, & \dot\epsilon_{xy}, & \dot\epsilon_{xz} \end{pmatrix}^T \quad\text{and}\quad \dot{\boldsymbol{\epsilon}}_2 = \begin{pmatrix} \dot\epsilon_{xy}, & \dot\epsilon_{xx}+2\dot\epsilon_{yy}, & \dot\epsilon_{yz}. \end{pmatrix}^T. \tag{2.2}$$

Denoting with u and v the horizontal components of the velocity $\boldsymbol{u}$, the stress components are defined as $\epsilon_{xx}=\partial_x u$, $\epsilon_{xy}=\frac{1}{2}(\partial_y u+\partial_x v)$, $\epsilon_{yy}=\partial_y v$, $\epsilon_{xz}=\frac{1}{2}\partial_z u$ and $\epsilon_{yz}=\frac{1}{2}\partial_z v$. The ice viscosity μ in Eq. (2.1) is given by

$$\mu \;=\; \frac{1}{2}A(T)^{-\frac{1}{n}}\,\dot\epsilon_e^{\frac{1-n}{n}}, \tag{2.3}$$

where $A(T) = \alpha_1 e^{\alpha_2 T}$ is a temperature-dependent rate factor (see [47] for the definition of coefficients α_1 and α_2), $n = 3$ is the power-law exponent, and the effective strain rate, $\dot{\epsilon}$, is defined as

$$\dot{\epsilon}_e \equiv \left(\dot{\epsilon}_{xx}^2 + \dot{\epsilon}_{yy}^2 + \dot{\epsilon}_{xx}\dot{\epsilon}_{yy} + \dot{\epsilon}_{xy}^2 + \dot{\epsilon}_{xz}^2 + \dot{\epsilon}_{yz}^2\right)^{\frac{1}{2}}, \tag{2.4}$$

where $\dot{\epsilon}_{ij}$ are the corresponding strain-rate components. Given that the atmospheric pressure is negligible compared to the pressure in the ice, we prescribe stress-free conditions at the the upper surface:

$$\dot{\boldsymbol{\epsilon}}_1 \cdot \boldsymbol{n} = \dot{\boldsymbol{\epsilon}}_2 \cdot \boldsymbol{n} = 0, \tag{2.5}$$

where $\mathbf{n}$ is the outward pointing normal vector at the ice sheet upper surface, $z = s(x, y)$. The lower surface can slide according to the following Robin-type boundary condition

$$2\mu_e \dot{\boldsymbol{\epsilon}}_1 \cdot \boldsymbol{n} + \beta u = 0, \quad 2\mu \dot{\boldsymbol{\epsilon}}_2 \cdot \boldsymbol{n} + \beta v = 0,$$

where β is a spatially variable friction coefficient and u and v are the horizontal components of the velocity $\boldsymbol{u}$. The field β is set to zero beneath floating ice. On lateral boundaries we prescribe the conditions

$$2\mu \dot{\boldsymbol{\epsilon}}_1 \cdot \boldsymbol{n} = \frac{1}{2} g H \left(\rho_i - \rho_w r^2\right) n_1 \quad \text{and} \quad 2\mu \dot{\boldsymbol{\epsilon}}_2 \cdot \boldsymbol{n} = \frac{1}{2} g H \left(\rho_i - \rho_w r^2\right) n_2, \tag{2.6}$$

where $\mathbf{n}$ is the outward pointing normal vector to the lateral (i.e., parallel to the (x, y) plane), ρ_w is the density of ocean water, n_1 and n_2 are the x and y component of $\mathbf{n}$, and r is the ratio of ice thickness that is submerged. On terrestrial ice margins $r = 0$, whereas on floating ice $r = \frac{\rho_i}{\rho_w}$. Additional details on the momentum balance solver can be found in [47].

2.2. Temperature model. As apparent from (2.3), the ice rheology depends on the ice temperature T. In order to model the ice sheet thermal state, we consider an enthalpy formulation similar to the one proposed by Aschwanded et al. in [1]. We assume that, for cold ice, the enthalpy h depends linearly on the temperature, whereas for temperate ice, the enthalpy grows linearly with the water content ϕ

$$h = \begin{cases} \rho_i c\,(T - T_0), & \text{for cold ice } (h \leq h_m), \\ h_m + \rho_w L\,\phi, & \text{for temperate ice.} \end{cases}$$

Here, the melting enthalpy h_m is defined as $h_m := \rho_w c(T_m - T_0)$ and T_0 is a uniform reference temperature.

The steady state enthalpy equation reads

$$\nabla \cdot \boldsymbol{q}(h) + \boldsymbol{u} \cdot \nabla h = 4\mu\, \epsilon_e^2. \tag{2.7}$$

Here, $\boldsymbol{q}(h)$ is the enthalpy flux, given by

$$\boldsymbol{q}(h) = \begin{cases} \frac{k}{\rho_i c_i} \nabla h, & \text{for cold ice } (h \leq h_m), \\ \frac{k}{\rho_i c_i} \nabla h_m + \rho_w L \boldsymbol{j}(h), & \text{for temperate ice,} \end{cases}$$

$\boldsymbol{u} \cdot \nabla h$ is the drift term, and $4\mu\, \epsilon_e^2$ is the heat associated to ice deformation. The water flux term

$$\boldsymbol{j}(h) := \frac{1}{\eta_w} (\rho_w - \rho_i) k_0 \phi^\gamma \boldsymbol{g}$$

has been introduced by Schoof and Hewitt ([46, 32]), and it describes the percolation of water driven by gravity. The parameter c_i is the heat capacity of ice, k its thermal conductivity, and L is the latent heat of fusion. At the upper surface, the enthalpy is set to $h = \rho_i c(T_s - T_0)$, where T_s is the temperature of the air at the ice upper surface. At the bed, the ice is either in contact with a dry bed or with a film of water at the melting point temperature and, in first approximation, satisfies the Stefan condition:

$$m = G + \beta\sqrt{u^2+v^2} - k\nabla T\cdot\boldsymbol{n} \quad \text{and} \quad m\,(T - T_m) = 0 \quad \text{and} \quad T_m \leq 0.$$

Here, m is the melting rate. Ice at the bed is melting when $m > 0$ and refreezing when $m < 0$. Moreover, G is the geothermal heat flux (positive if entering the ice domain), $\beta\sqrt{u^2+v^2}$ is the frictional heat, and $-k\nabla T\cdot\boldsymbol{n}$ is the temperature heat flux exiting the domain as $\boldsymbol{n}$ is the outer normal to the ice domain. Depending on whether the ice is cold at the bed, melting or refreezing, the Stefan condition translates into natural or essential boundary conditions for the enthalpy equation. Further details on the enthalpy formulation and its discretization are provided in [41].

2.3. Coupled model.

The ice velocity depends on the temperature through (2.4), and the enthalpy depends on the velocity field through the drift term $\boldsymbol{u}\cdot\nabla h$ and the fractional heat term at the ice sheet lower surface. The first order problem (2.1) only provides the horizontal velocities u and v, but we also need the vertical velocity w to solve the enthalpy equations. The vertical velocity w is computed using the incompressibility condition

$$\partial_x u + \partial_y v + \partial_z w = 0, \tag{2.8}$$

with the Dirichlet boundary condition at the ice lower surface

$$\boldsymbol{u}\cdot\boldsymbol{n} = \frac{m}{L\,(\rho_i - \rho_w\phi)}.$$

The coupled problem is formed by problems (2.1), (2.8) and (2.7) and their respective boundary conditions. For further details, see [41]. Figure 1 shows the ice velocity and temperature computed solving the coupled thermo-mechanical model. For details about the problem setting and the Greenland data set, see [41].

3. Finite element discretization.

The ice sheet mesh is generated by extruding in the vertical direction a two dimensional unstructured mesh of the ice sheet horizontal extension ([47]) and it is constituted of layers of prisms. The problems described in section 2 are discretized with continuous piece-wise bi-linear (for triangular prisms) or tri-linear (for hexahedra) finite elements using a standard Galerkin formulation, for each component of the velocity and for the enthalpy. We use upwind stabilization for the enthalpy equation. The nonlinear discrete problems can be written in the residual form

$$F(x) = 0, \tag{3.1}$$

where x is the problem unknown (velocity, enthalpy, or both, depending on the problem). The nonlinear problems are then solved using a Newton-Krylov approach. More precisely, we linearize the problem using Newton's method, and solve the resulting linear tangent problems

$$DF(x^{(k)})\Delta\, x^{(k)} = -F(x^{(k)}) \tag{3.2}$$

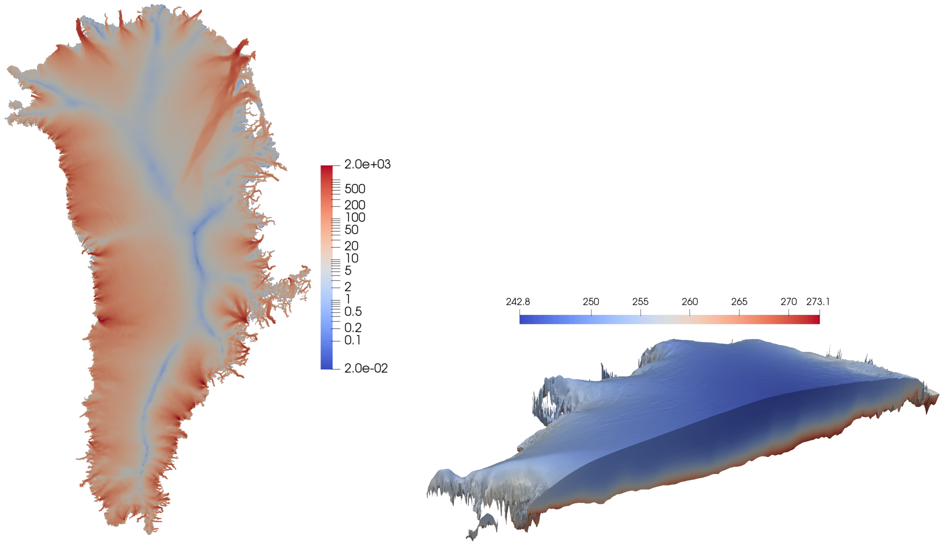

Fig. 1. *Solution of a Greenland ice sheet simulation. Left: ice surface speed in [m/yr], Right: ice temperature in [K] over a vertical section of the ice sheet.*

using a Krylov subspace method. The Jacobian DF is computed through automatic differentiation. At each nonlinear iteration, we solve a problem of the form

$$Ax = r, \tag{3.3}$$

where A is the tangent matrix $DF(x^{(k)})$, and r is the residual vector $-F(x^{(k)})$. Using a block matrix notation, the tangent problem of the velocity problem can be written as

$$\begin{bmatrix} A_{uu} & A_{uv} \\ A_{vu} & A_{vv} \end{bmatrix} \begin{bmatrix} x_u \\ x_v \end{bmatrix} = \begin{bmatrix} r_u \\ r_v \end{bmatrix} \tag{3.4}$$

where the tangent matrix is symmetric positive definite. When considering also the vertical velocity w, the tangent problem becomes

$$\underbrace{\begin{bmatrix} A_{uu} & A_{uv} & \\ A_{vu} & A_{vv} & \\ A_{wu} & A_{wu} & A_{ww} \end{bmatrix}}_{=:A_u} \underbrace{\begin{bmatrix} x_u \\ x_v \\ x_w \end{bmatrix}}_{=:x_u} = \underbrace{\begin{bmatrix} r_u \\ r_v \\ r_w \end{bmatrix}}_{=:r_u} \tag{3.5}$$

Note that the matrix is lower block-triangular because in the FO approximation, the horizontal velocities are independent of the vertical velocity. Similarly, the temperature equation reads

$$A_T\, x_T = r_T. \tag{3.6}$$

The coupled problem is a multiphysics problem coupling the velocity and the temperature problem. Hence, the tangent system can be written as

$$\begin{bmatrix} A_u & C_{uT} \\ C_{Tu} & A_T \end{bmatrix} \begin{bmatrix} x_u \\ x_T \end{bmatrix} = \begin{bmatrix} \tilde{r}_u \\ \tilde{r}_T \end{bmatrix}, \tag{3.7}$$

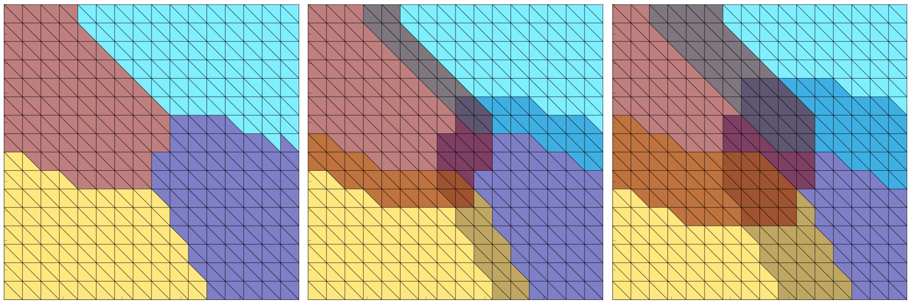

FIG. 2. *Extending two-dimensional nonoverlapping subdomains (left) by layers of elements to obtain overlapping domain decompositions with an overlap of $\delta = 1h$ (middle) and $\delta = 2h$ (right).*

where the blocks A_u and A_T and solution vectors x_u x_T are the same as in the single physics problems; cf. (3.5) and (3.6). The residual vectors $\tilde{r}_u$ and $\tilde{r}_T$ differ from the single physics residuals r_u and r_T due to the coupling of velocity and temperature, which also results in the nonzero coupling blocks coupling blocks C_{uT} and C_{Tu} in the tangent matrix.

4. Preconditioners.

In order to solve the tangent problems (3.2) in our Newton iteration, we apply the generalized minimal residual (GMRES) method [44] and speed up the convergence using generalized Dryja–Smith–Widlund (GDSW) type domain decomposition preconditioners. In particular, we will use classical GDSW and reduced dimension GDSW (RGDSW) preconditioners, as described in subsection 4.1, as well as corresponding monolithic preconditioners, as introduced in subsection 4.3. In order to improve the performance of the first level of the Schwarz preconditioners, we will always apply scaled prolongation operators; cf. subsection 4.2. As we will describe in subsection 4.4, domain decomposition preconditioners and, in particular, GDSW type preconditioners are well-suited for the solution of land ice problems because of the specific structure of the meshes. In order to improve the efficiency of the preconditioners in our Newton-Krylov algorithm, we will also apply strategies to reuse, in later Newton iterations, certain components of the preconditioners set up in the first Newton iteration; see subsection 4.5.

For the sake of clarity, we will restrict ourselves to the case of uniform meshes with characteristic element size h for the description of the preconditioners. However, the methods can also be applied to non-uniform meshes as the ones for Greenland; see Figure 4.

4.1. GDSW type preconditioners.

Let us consider the general linear system

$$Ax = b \tag{4.1}$$

arising from a finite element discretization of an elliptic boundary value problem on Ω. Our aim is then to apply the preconditioners to the tangent problems (3.3) of the model problems described in section 2.

The GDSW preconditioner was originally introduced by Dohrmann, Klawonn, and Widlund in [13, 12] for elliptic problems. It is a two-level Schwarz preconditioner with energy minimizing coarse space and exact solvers. To describe the construction of the GDSW preconditioner, let Ω be partitioned into N nonoverlapping subdomains $\Omega_1, ..., \Omega_N$ with characteristic size H. We extend these subdomains by adding k layers of finite elements resulting in overlapping subdomains $\Omega_1', ..., \Omega_N'$ with an overlap

$\delta = kh$; cf. Figure 2 for a two-dimensional example. In general, two-level Schwarz preconditioners for (4.1) with exact solvers are of the form

$$M_{OS-2} = \underbrace{\Phi A_0^{-1} \Phi^T}_{\text{coarse level}} + \underbrace{\sum_{i=1}^{N} R_i^T A_i^{-1} R_i}_{\text{first level}} . \tag{4.2}$$

Here, $A_0 = \Phi^T A \Phi$ is the coarse matrix corresponding to a Galerkin projection onto the coarse space, which is spanned by the columns of matrix Φ. The local matrices A_i are submatrices of A corresponding to the overlapping subdomains $\Omega_1', ..., \Omega_N'$. They can be written as $A_i = R_i A R_i^T$, where $R_i : V^h \rightarrow V_i^h$ is the restriction operator from the global finite element space V^h to the local finite element space V_i^h on Ω_i'; the R_i^T is the corresponding prolongation.

We first present the framework enabling the construction of energy-minimizing coarse spaces for elliptic problems based on a partition of unity on the interface

$$\Gamma = \left\{ x \in (\overline{\Omega}_i \cap \overline{\Omega}_j) \setminus \partial\Omega_D | i \neq j, 1 \leq i, j \leq N \right\} \tag{4.3}$$

of the nonoverlapping domain decomposition, where $\partial\Omega_D$ is the Dirichlet boundary. This will allow us to construct classical GDSW coarse spaces [13, 12] and reduced dimension GDSW coarse spaces [16] as used in our simulations. Note that other types of coarse spaces can be constructed using this framework as well, e.g., coarse spaces based on the MsFEM (Multiscale Finite Element Method) [34]; see also [7]. However, in our experiments, we restrict ourselves to the use of GDSW type coarse spaces.

Let us first decompose Γ into connected components $\Gamma_1, ..., \Gamma_M$. This decomposition of Γ may be overlapping or nonoverlapping. Furthermore, let R_{Γ_i} be the restriction from all interface degrees of freedom to the degrees of freedom of the interface component Γ_i. In order to account for overlapping decompositions of the interface, we introduce diagonal scaling matrices D_{Γ_i}, such that

$$\sum_{i=1}^{M} R_{\Gamma_i}^T D_{\Gamma_i} R_{\Gamma_i} = I_\Gamma, \tag{4.4}$$

where I_Γ is the identity matrix on Γ. This means that the scaling matrices correspond to a partition of unity on the interface Γ.

Using the scaling matrices D_{Γ_i}, we can now build a space which can represent the restriction of the null space of our problem to the interface. Therefore, let the columns of the matrix Z form a basis of the null space of the operator $\hat{A}$, which is the global matrix corresponding to A but with homogeneous Neumann boundary conditions on the full boundary, and let the Z_Γ be restriction of Z to the interface Γ. Because of (4.4), we have

$$\sum_{i=1}^{M} R_{\Gamma_i}^T D_{\Gamma_i} R_{\Gamma_i} Z_\Gamma = Z_\Gamma.$$

Now, for each Γ_i, we construct a matrix Φ_{Γ_i} such that its columns are a basis of the space spanned by the columns of $D_{\Gamma_i} R_{\Gamma_i} Z_\Gamma$. Then, the interface values of our coarse space are given by the matrix

$$\Phi_\Gamma = \left[\begin{array}{ccc} R_{\Gamma_1}^T \Phi_{\Gamma_1} & \dots & R_{\Gamma_M}^T \Phi_{\Gamma_M} \end{array} \right] . \tag{4.5}$$

Based on these interface values, the coarse basis functions are finally computed as energy-minimizing extensions to the interior of the nonoverlapping subdomains. Therefore, we partition all degrees of freedom into interface (Γ) and interior (I) degrees of freedom. Then, the system matrix can written as

$$A = \begin{bmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{bmatrix}$$

and the energy-minimizing extensions are computed as $\Phi_I = -A_{II}^{-1} A_{I\Gamma} \Phi_\Gamma$, resulting in the coarse basis

$$\Phi = \left[\begin{array}{c} \Phi_I \\ \Phi_\Gamma \end{array} \right] = \left[\begin{array}{c} -A_{II}^{-1} A_{I\Gamma} \Phi_\Gamma \\ \Phi_\Gamma \end{array} \right]. \tag{4.6}$$

As mentioned earlier, this construction allows for a whole family of coarse spaces, depending on decomposition of the interface into components Γ_i and the choice of scaling matrices D_{Γ_i}.

GDSW coarse spaces. We obtain the interface components of the GDSW coarse space $\Gamma_i^{(\mathrm{GDSW})}$ by decomposing the interface Γ into the largest connected components γ belonging to the same sets of subdomains $\mathcal{N}_\gamma$, i.e., into vertices, edges, and faces; cf., e.g., [38]. More precisely,

$$\mathcal{N}_\gamma := \left\{ i : x \in \overline{\Omega_i} \ \forall x \in \gamma \right\}.$$

Because these components are disjoint by construction, the scaling matrices $D_{\Gamma_i^{(\mathrm{GDSW})}}$ have to be chosen as identity matrices $I_{\Gamma_i^{(\mathrm{GDSW})}}$ in order to satisfy (4.4). Using this choice, we obtain the classical GDSW coarse space as introduced by Dohrmann, Klawonn, and Widlund in [13, 12]. If the boundaries of the subdomains are uniformly Lipschitz, the condition number estimate for the resulting two-level GDSW preconditioner,

$$\kappa \left(M_{\mathrm{GDSW}}^{-1} A \right) \leq C \left(1 + \frac{H}{\delta} \right) \left(1 + \log \left(\frac{H}{h} \right) \right), \tag{4.7}$$

holds for scalar elliptic and compressible linear elasticity model problems; the constant C is then independent of the geometrical parameters H, h, and δ. For the general case of $\Omega \subset \mathbb{R}^2$ being decomposed into John domains, we can obtain a condition number estimate with a second power logarithmic term, i.e., with $\left(1 + \log\left(\frac{H}{h}\right)\right)^2$ instead of $\left(1 + \log\left(\frac{H}{h}\right)\right)$; cf. [12, 13]. Please also refer to [14, 15] for other variants with linear logarithmic term.

RGDSW coarse spaces. Another choice of the Γ_i leads to reduced dimension GDSW (RGDSW) coarse spaces; cf. [16]. In order to construct the interface components $\Gamma_i^{(\mathrm{RGDSW})}$, we first define a hierarchy of the previously defined $\Gamma_i^{(\mathrm{GDSW})}$. In particular, we call an interface component γ *ancestor* of another interface component γ' if $\mathcal{N}_{\gamma'} \subset \mathcal{N}_\gamma$; conversely, we call γ *offspring* of γ' if $\mathcal{N}_{\gamma'} \supset \mathcal{N}_\gamma$. Now, let $\left\{ \hat{\Gamma}_i^{(\mathrm{GDSW})} \right\}_{i=1,\ldots,M^{(\mathrm{RGDSW})}}$ be the set of all GDSW interface components which have no ancestors; we call these *coarse components*. Now, we define the RGDSW interface components as

$$\Gamma_i^{(\mathrm{RGDSW})} := \bigcup_{\mathcal{N}_\gamma \subset \mathcal{N}_{\hat{\Gamma}_i^{(\mathrm{GDSW})}}} \gamma, \quad \forall i = 1, \ldots, M^{(\mathrm{RGDSW})}. \tag{4.8}$$

The $\Gamma_i^{(\mathrm{RGDSW})}$ may overlap in nodes which do not belong to the coarse components. Hence, we have to introduce scaling operators $D_{\Gamma_i^{(\mathrm{RGDSW})}} \neq I_{\Gamma_i^{(\mathrm{RGDSW})}}$ to obtain a partition of unity on the interface; cf. (4.4). Different scaling operators D_{Γ_i} lead to different variants of RGDSW coarse spaces, e.g., Options 1, 2.1, and 2.2, introduced in [16] and another variant introduced in [25]. Here, we will only consider the algebraic variant, Option 1, where an inverse multiplicity scaling

$$D_{\Gamma_i^{(\mathrm{RGDSW})}} = R_{\Gamma_i^{(\mathrm{RGDSW})}} \left(\sum_{j=1}^{M^{(\mathrm{RGDSW})}} R^T_{\Gamma_j^{(\mathrm{RGDSW})}} R_{\Gamma_j^{(\mathrm{RGDSW})}} \right)^{-1} R^T_{\Gamma_i^{(\mathrm{RGDSW})}}.$$

is employed. Under the condition that all subdomains are Lipschitz domains, we then obtain the same condition number estimate as previously for GDSW coarse spaces

$$\kappa(M^{-1}_{\mathrm{RGDSW}} A) \leq C \left(1 + \frac{H}{\delta}\right) \left(1 + \log\left(\frac{H}{h}\right)\right); \tag{4.9}$$

for scalar elliptic and compressible linear elasticity model problems; cf. [16].

The only missing ingredient to construct the GDSW and RGDSW coarse spaces is the respective the null space Z of the global Neumann matrix corresponding to A. For the velocity and the temperature problem, the preconditioners can be directly constructed and applied using the corresponding null spaces spanned by

$$r_{u,1} := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, r_{u,2} := \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ and } r_{u,3} := \begin{bmatrix} y \\ -x \end{bmatrix}$$

or

$$r_T := \begin{bmatrix} 1 \end{bmatrix},$$

respectively, on each finite element node. Here, $r_{u,1}$ and $r_{u,2}$ correspond to the translations and $r_{u,3}$ to the linearized rotation building the null space of the velocity problem. The r_T is the constant null space element of the temperature problem.

Remark 4.1. Sometimes it may be beneficial to only consider a subspace $\hat{Z}$ of the full space Z. This results in a smaller coarse space, at the cost of slower convergence of the linear solver. In particular, in theory, numerical scalability is not provided in this case. However, since the coarse solve is typically a parallel scaling bottleneck, it may still be faster to neglect a part of the coarse space for a large number of subdomains. In our numerical results, we will actually observe that neglecting rotational rigid body modes improves the parallel performance of our solver; see also [28, 24] for similar experiments for elasticity problems.

Note that, if rotations are neglected, the GDSW and RGDSW coarse spaces are actually constructed in an algebraic way because the translational coarse basis functions can be computed without geometric information; see also the discussion in [24].

For the coupled problem described in subsection 2.3, we will describe an monolithic preconditioner in subsection 4.3, where we use the same construction but with decoupled extensions operators. Before that, however, we will describe the scaled prolongation operators used in the first level in our numerical experiments.

4.2. Scaled prolongation operators. As first shown in [8], the convergence of additive Schwarz preconditioners can often be improved using restricted or scaled variants of the prolongation operators R_i^T in (4.2); see also [17, 23]. For the sake of brevity, we will not compare the performance of the standard, the restricted, and the scaled variants for the different model problems considered in this paper. We only show results using the scaled variant because it performed best in preliminary tests.

We construct the scaled prolongation operator $\tilde{R}_i^T$ such that $\sum_{i=1}^N \tilde{R}_i^T R_i = I$:

$$\tilde{R}_i^T := \left(\sum_{j=1}^N R_j^T R_j \right)^{-1} R_i^T.$$

Note that the matrix $\sum_{i=1}^N R_i^T R_i$ is just a diagonal scaling matrix, and its inverse can therefore be specified directly. The two-level Schwarz preconditioner with scaled prolongations then reads

$$M_{OS-2} = \Phi A_0^{-1} \Phi^T + \sum_{i=1}^N \tilde{R}_i^T A_i^{-1} R_i.$$

4.3. Monolithic preconditioning the coupled problem. For the coupled problem, A is structured as follows

$$A = \begin{bmatrix} A_{uu} & A_{uT} \\ A_{Tu} & A_{TT} \end{bmatrix}, \tag{4.10}$$

where the off-diagonal blocks formally account for the coupling of the different variables; cf. section 3. We will construct monolithic two-level Schwarz preconditioners as introduced in [36, 37] and extended to monothic GDSW preconditioners in [22, 23]. Formally, the monolithic preconditioners for the coupled problem can again be written as

$$M_{OS-2} = \Phi A_0^{-1} \Phi^T + \sum_{i=1}^N \tilde{R}_i^T A_i^{-1} R_i. \tag{4.11}$$

However, all matrices are now 2×2 block-matrices. In particular, the monolithic restriction and prolongation matrices are of the form

$$R_i = \begin{bmatrix} R_{i,u} & 0 \\ 0 & R_{i,T} \end{bmatrix} \quad \text{and} \quad \tilde{R}_i = \begin{bmatrix} \tilde{R}_{i,u} & 0 \\ 0 & \tilde{R}_{i,T} \end{bmatrix},$$

where $R_{i,u}$ and $R_{i,T}$ are the restriction operators to the overlapping subdomain Ω_i' on the velocity and temperature degrees of freedom, and $\tilde{R}_{i,u}$ and $\tilde{R}_{i,T}$ are the respective prolongations operators.

The coarse space can be constructed in a similar way as in the single physics case. In particular, the interface components Γ_i and the scaling matrices D_{Γ_i} are constructed in the same way, and the null space Z of the multi physics problem is composed of the null spaces of the individual single physics problems. However, as we will observe in the numerical results, it is necessary to remove the coupling blocks between the velocity and the temperature problem before computing the extensions (4.6). Hence, instead of A, the matrix

$$\tilde{A} = \begin{bmatrix} A_{uu} & 0 \\ 0 & A_{TT} \end{bmatrix} \tag{4.12}$$

is used in the computation of the harmonic extensions, i.e., $\Phi_I = -\tilde{A}_{II}^{-1}\tilde{A}_{I\Gamma}\Phi_\Gamma$. This can be viewed as applying a block Jacobi preconditioner with two blocks corresponding to the single physics problems instead of solving the systems corresponding to A_{II}^{-1} monolithically. Consequently, the coarse basis functions corresponding to the velocity and the temperature problem can be computed independently. Then, the matrix Φ is of the form

$$\Phi = \begin{bmatrix} \Phi_{u,u_0} & 0 \\ 0 & \Phi_{T,T_0} \end{bmatrix}, \tag{4.13}$$

where the row indices u and T indicate the finite element functions of the original problem, and column indices u_0 and T_0 correspond to the basis functions of the coarse space. A similar decoupling approach for the coarse basis functions was performed in [22, 23] for a monolithic preconditioner for fluid problems. However, it was necessary to first compute the fully coupled extensions (4.6) and to drop the off diagonal blocks in the matrix Φ afterwards. This was due to the fact that the system matrix was of the form $\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}$, such that the decoupled matrix would become singular. Here, the decoupled matrix (4.12) remains invertible since the individual blocks correspond to the single physics velocity and temperature problems. Therefore, our coarse basis matrix is also of the same structure for Lagrangian coarse spaces in [36, 37].

It is important to note that, even though the coarse basis functions do not contain any coupling blocks, the coarse problem is still a coupled problem with a coarse matrix of the form

$$\begin{aligned} A_0 &= \begin{bmatrix} \Phi_{u,u_0} & 0 \\ 0 & \Phi_{T,T_0} \end{bmatrix}^T \begin{bmatrix} A_{uu} & A_{uT} \\ A_{Tu} & A_{TT} \end{bmatrix} \begin{bmatrix} \Phi_{u,u_0} & 0 \\ 0 & \Phi_{T,T_0} \end{bmatrix} \\ &= \begin{bmatrix} \Phi_{u,u_0}^T A_{uu}\Phi_{u,u_0} & \Phi_{u,u_0}^T A_{uT}\Phi_{T,T_0} \\ \Phi_{T,T_0}^T A_{Tu}\Phi_{u,u_0} & \Phi_{T,T_0}^T A_{TT}\Phi_{T,T_0} \end{bmatrix}. \end{aligned}$$

Because we use equal order discretizations for the velocity and temperature variables in the coupled problem, we can formally apply a node wise ordering to our degrees of freedom. Then, the monolithic preconditioner can be constructed exactly as in the elliptic case (see section 4), however, using the previously described decoupled matrix (4.12) to compute the extension.

We then obtain all three velocity degrees of freedom and one temperature degree of freedom for each finite element node. Therefore, the full null space is spanned by the null space corresponding to the three velocity degrees of freedom

$$r_{u,1} := \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; r_{u,2} := \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \; r_{u,3} := \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \text{ and } r_{u,4} := \begin{bmatrix} y \\ -x \\ 0 \\ 0 \end{bmatrix}$$

as well as the null space on the temperature degree of freedom

$$r_T := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Here, $r_{u,4}$ corresponds to a linearized rotation, which will be neglected in some of our numerical experiments to reduce the computing time on the coarse level.

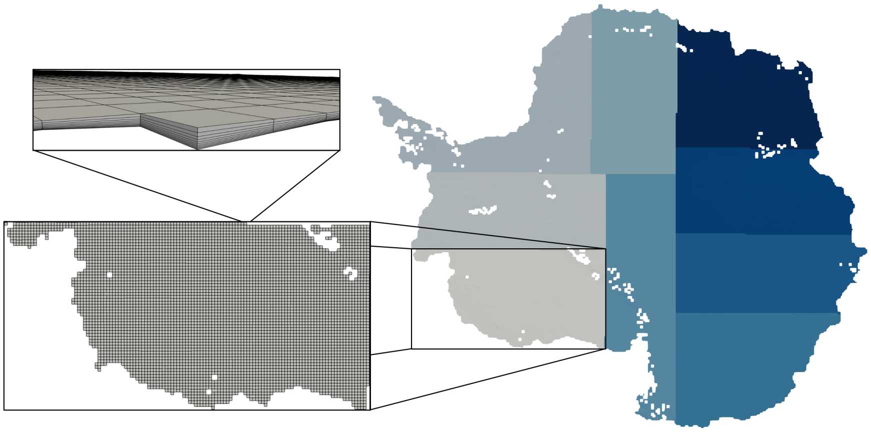

Fig. 3. *Uniform hexahedral mesh for the Antarctica ice sheet with a horizontal resolution of* 16 km *decomposed into nine subdomains. The domain decomposition is performed on the two-dimensional top surface mesh, and the subdomains are extruded in vertical direction to obtain three-dimensional subdomains with 10 layers height.*

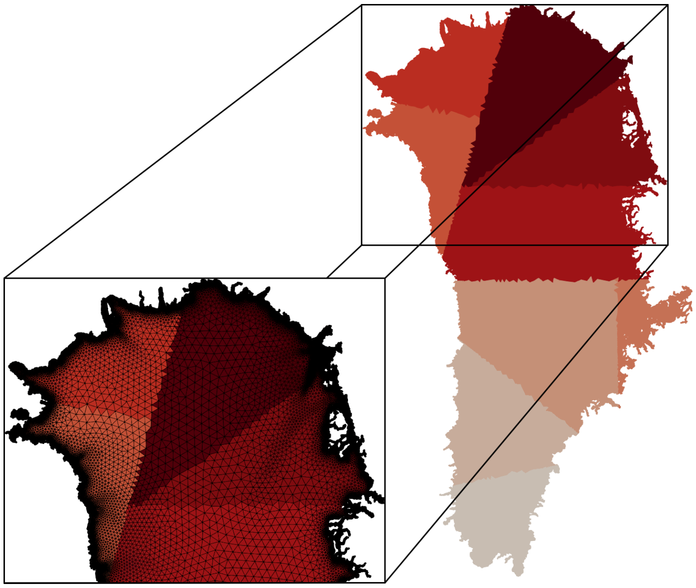

Fig. 4. *Non-uniform triangulation of the top surface mesh for the Greenland ice sheet with a horizontal resolution of* 3 km *to* 30 km *decomposed into nine subdomains. The three-dimensional mesh is then obtained by extrusion in vertical direction.*

4.4. Remarks on domain decomposition methods for land ice problems. The geometries for the ice sheets in Antarctica and Greenland are visualized in Figures 3 and 4. Generally, the horizontal extensions of the ice sheets are in the order of hundreds or thousands of kilometers, whereas their thickness is at maximum only a few kilometers. Therefore, the geometries and the corresponding meshes used in our simulations are clearly anisotropic; cf. section 3 for a description of the mesh generation procedure and Figure 3 for a visualization of a exemplary mesh of Antarctica

reuse strategy	short description
NR *(no reuse)*	Set up the preconditioner from scratch in each nonlinear iteration.
IS *(index sets)*	Reuse the index sets for the overlapping subdomains and the interface components
SF1 *(symb. fact. lvl 1)*	Reuse the symbolic factorization of A_i.
SF2 *(symb. fact. lvl 2)*	Reuse the symbolic factorization of A_0.
CB *(coarse basis)*	Reuse the coarse basis Φ.
CM *(coarse matrix)*	Reuse the coarse matrix A_0.

Table 1

Reuse strategies for monolithic GDSW preconditioners (4.2) *for nonlinear model problems.*

with a horizontal mesh resolution of 16 km and 10 layers of elements in z direction.

Due to this specific structure of the meshes, we perform the domain decomposition into nonoverlapping subdomains as follows: First, we decompose the two-dimensional mesh of the top surface. We extrude the two-dimensional subdomains in z direction next resulting in a domain decomposition of the whole three-dimensional domain. Hence, the domain decomposition is essentially a two-dimensional domain decomposition, and the partition of the domain decomposition interface Γ into the components $\Gamma_i^{(\mathrm{GDSW})}$ only yields edges and faces but no vertices. However, as can be seen in Figures 3 and 4, the subdomain geometries can be very irregular due to the complex shape of the boundary of the ice sheets. Hence, the domain decomposition is not well suited for the use of classical Lagrangian coarse spaces, which would require the construction of a coarse triangulation of the geometry. However, this is not required for GDSW type coarse spaces which can be constructed without an additional coarse triangulation. Hence they can easily be constructed for the considered land ice problems.

4.5. Reuse strategies for nonlinear problems. The model problems in section 2 are highly nonlinear; as can be seen in section 6, the coupled problem requires a particularly high number of nonlinear iterations. Therefore, we will investigate several strategies to reuse information from the first iteration in later Newton iterations, such that the total time to solution can be improved. Note that other approaches where the information is updated in certain multiple Newton iterations, e.g. in every nth iteration, are also possible but out of the scope of this paper.

The different reuse strategies, which are listed in Table 1, are used in different numerical experiments in section 6. Since neither the topology nor the domain decomposition of our problem changes during the nonlinear iteration, it is a safe assumption that the index sets of the overlapping subdomains and the interface components stay the same. This saves mostly communication, which dominates the time for identifying the index sets; see section 5. If the sparsity pattern of the system matrix is also constant during the nonlinear iteration, the symbolic factorizations for A_i and A_0 can be easily reused as well.

In GDSW type preconditioners, the coarse basis functions Φ change with the tangent matrix, which is used to compute the extensions (4.6) in each nonlinear iteration. However, in practice, the coarse basis computed with the tangent matrix in the first Newton iteration can also be used in later iterations. In some cases, the complete coarse matrix A_0 and its factorization can even be reused.

5. Software framework. The land ice problems are implemented in Albany Land Ice (formerly referred to as Albany FELIX) [47, 45], a C++ finite element library that relies on the Trilinos packages [49] for MPI+X parallelism (Tpetra, Kokkos), linear (Belos/AztecOO) and nonlinear (NOX) solvers, preconditioners (Ifpack2, Muelu, ShyLU, FROSch [20, 28, 27, 26]), discretization tools (STK, Intrepid2, Phalanx) and automatic differentiation (Sacado). Albany Land Ice is part of the land ice code MALI [33].

The GDSW type preconditioners described in section 4 are implemented in the FROSch framework [27, 26], which is part of Trilinos [49]. FROSch can use both distributed-memory parallelism using the Tpetra package of Trilinos and shared-memory parallelism while using the direct solvers interfaced through Amesos2 package of Trilinos [2]. With respect to shared-memory parallelism, in this paper, we restrict ourselves to using CPU threads. Specifically, we use the Pardiso solver provided with the Intel MKL software, which can also make use of shared-memory parallelism using OpenMP threads. FROSch is called from Albany Land Ice using the unified Trilinos solver interface Stratimikos and directly uses the Tpetra matrices and vectors which have been assembled in. Moreover, FROSch makes use of the index set of the nonoverlapping domain decomposition and the null space basis provided by Albany Land Ice in form of Tpetra map and multivector objects; cf. the discussion in [21].

6. Numerical results. In this section, we will present numerical results for the flow (subsection 2.1), temperature (subsection 2.2), and coupled (subsection 2.3) problems. For the flow problem, we will use the uniform meshes for Antarctica, whereas we will use the non-uniform Greenland meshes for the two other model problems; cf. Figures 3 and 4. The experiments were performed using the Haswell compute nodes (2 sockets with a 16-core Intel Xeon Processor E5-2698 v3 with 2.3 GHz each) of the Cori supercomputer at NERSC (National Energy Research Scientific Computing Center); we always employed one processor core per thread. The code was compiled using Intel 19.0.3.199 compilers and Intel MKL. The subdomain problems and the coarse problem are solved on one MPI rank using used Pardiso from the Intel MKL with OpenMP parallelization if more than one OpenMP thread is used.

The nonlinear problems are solved using the inexact Newton method with backtracking implemented in the Trilinos package NOX up to a relative reduction of the residual of 10^{-5}. As the linear solver we employ the GMRES (generalized minimal residual) method [44] from Trilinos AztecOO preconditioned by two-level overlapping Schwarz domain decomposition preconditioners from Trilinos FROSch (part of the package ShyLU) as described in section 4; cf. [28, 27, 22, 23, 26]. We iterate the GMRES method up to a relative reduction of the residual of 10^{-7} for the flow and temperature problems or 10^{-9} for the coupled problem. Since the number of nonlinear iterations is not influenced by our preconditioners, we always report the number of linear iterations averaged over the number of Newton iterations.

With respect to the Schwarz preconditioners, if not stated otherwise, we will always use one layer of overlap as determined from the sparsity pattern of the matrix. On the first level, we apply scaled prolongation operators; cf. subsection 4.2. As already discussed in [28], we will use two communication steps in order to transfer information from the first to the second level (scatter and gather); only during the discussion in subsection 6.1.3, we will also present results using only one or three communication steps.

6.1. Flow problem for Antarctica. In this section, we will present an extensive numerical study of GDSW type preconditioners for the land ice flow problem

Without rotational coarse basis functions (2 rigid body modes)								
	GDSW (IS & SF1 & SF2 & CB)				**RGDSW** (IS & SF1 & SF2 & CB)			
MPI ranks	$\dim V_0$	avg. its (nl its)	avg. setup	avg. solve	$\dim V_0$	avg. its (nl its)	avg. setup	avg. solve
512	4 598	**40.8** (11)	15.36 s	**12.38 s**	1 834	42.6 (11)	**14.99 s**	12.50 s
1 024	9 306	**43.3** (11)	5.80 s	6.27 s	3 740	44.5 (11)	**5.65 s**	**6.08 s**
2 048	18 634	**41.7** (11)	3.27 s	2.91 s	7 586	42.7 (11)	**3.11 s**	**2.79 s**
4 096	37 184	**41.4** (11)	2.59 s	2.07 s	15 324	42.5 (11)	**1.07 s**	**1.54 s**
8 192	72 964	**39.5** (11)	1.51 s	1.84 s	30 620	42.0 (11)	**1.20 s**	**1.16 s**
With rotational coarse basis functions (3 rigid body modes)								
	GDSW (IS & SF1 & SF2 & CB)				**RGDSW** (IS & SF1 & SF2 & CB)			
MPI ranks	$\dim V_0$	avg. its (nl its)	avg. setup	avg. solve	$\dim V_0$	avg. its (nl its)	avg. setup	avg. solve
512	6 897	**35.5** (11)	15.77 s	**11.21 s**	2 751	40.7 (11)	**15.23 s**	12.22 s
1 024	13 959	**35.6** (11)	6.16 s	**5.78 s**	5 610	42.9 (11)	**5.65 s**	6.04 s
2 048	27 951	**33.5** (11)	3.78 s	3.45 s	11 379	42.2 (11)	**3.17 s**	**2.81 s**
4 096	55 776	**31.8** (11)	2.21 s	3.80 s	22 986	44.3 (11)	**1.95 s**	**2.70 s**
8 192	109 446	**29.3** (11)	2.49 s	5.33 s	45 930	40.8 (11)	**1.19 s**	**3.13 s**

TABLE 2

Comparison of different coarse spaces for the flow problem on the Antarctica mesh with 4 km horizontal resolution and 20 layers of elements in vertical direction and a total of 35.3 m degrees of freedom. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its). Lowest average iterations counts, setup times, and solve times in each row are marked in **bold**.

# subdomains	512	1 024	2 048	4 096	8 192
GDSW	2 299	4 653	9 317	18 592	36 482
RGDSW	917	1 870	3 793	7 662	15 310

TABLE 3

Number of coarse components Γ_i for the Antarctica mesh with 4km horizontal resolution. The dimension of the coarse space is the number of coarse components multiplied by the dimension of the null space.

for Antarctica. Most of the simulations are performed on a medium size mesh with 4 km horizontal resolution and 20 layers of elements in vertical direction. We compare one level Schwarz methods and different GDSW type coarse spaces (subsection 6.1.1) and investigate several reuse strategies (subsection 6.1.2) as well as certain parallelization aspects (subsection 6.1.3). Moreover, we investigate the robustness with respect to an increasing number of mesh layers of elements in vertical direction (subsection 6.1.4), and compare our results using FROSch against the algebraic multigrid package MueLu [4, 3] (subsection 6.1.6).

Finally, we provide weak scaling results ranging from the coarsest mesh with 16 km horizontal resolution to the finest mesh with 1 km horizontal resolution. The largest computation in this weak scaling study was performed on 32 768 processor cores using 8192 MPI ranks and 4 OpenMP threads per MPI rank solving a problem with more than 566 m degrees of freedom.

6.1.1. Comparison of different Schwarz preconditioners. First, we compare the classical GDSW and the reduced dimension GDSW (RGDSW) coarse spaces in a strong scaling study using both the full three-dimensional null space and a two-dimensional null space where the rotation has been omitted; cf. the discussion in subsection 4.1. In this study, we reuse the index sets (IS), the symbolic factorizations (SF1 & SF2), and the coarse basis (CB) from the first nonlinear iteration. As can be seen in Table 2, all preconditioners scale numerically, but the iteration counts are

One-level Schwarz						
	one layer of algebraic overlap			two layers of algebraic overlap		
MPI ranks	avg. its (nl its)	avg. setup	avg. solve	avg. its (nl its)	avg. setup	avg. solve
512	67.7 (11)	**13.80 s**	19.55 s	**56.2** (11)	17.95 s	**18.40 s**
1 024	79.1 (11)	**5.00 s**	10.60 s	**66.5** (11)	6.74 s	**10.56 s**
2 048	96.1 (11)	**1.74 s**	**6.09 s**	**80.8** (11)	2.58 s	6.31 s
4 096	113.3 (11)	**0.81 s**	**3.59 s**	**94.8** (11)	1.21 s	3.99 s
8 192	132.0 (11)	**0.47 s**	**2.15 s**	**109.5** (11)	0.65 s	2.35 s
RGDSW (IS & SF1 & SF2 & CB & CM)						
	one layer of algebraic overlap			two layers of algebraic overlap		
MPI ranks	avg. its (nl its)	avg. setup	avg. solve	avg. its (nl its)	avg. setup	avg. solve
512	46.7 (11)	**14.94 s**	**13.81 s**	**42.1** (11)	18.89 s	14.13 s
1 024	49.2 (11)	**5.75 s**	**6.78 s**	**44.3** (11)	6.95 s	7.21 s
2 048	47.7 (11)	2.92 s	**3.10 s**	**44.3** (11)	**2.66 s**	3.56 s
4 096	48.9 (11)	**0.95 s**	**1.75 s**	**45.5** (11)	1.28 s	2.15 s
8 192	50.1 (11)	**0.63 s**	**1.35 s**	**46.0** (11)	0.76 s	1.66 s

TABLE 4

Comparison of one-level and RGDSW Schwarz preconditioners for the flow problem on the Antarctica mesh with 4 km horizontal resolution and 20 layers of elements in vertical direction and a total of 35.3 m degrees of freedom. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its). Lowest average iterations counts, setup times, and solve times in each row are marked in **bold**.

better for the classical GDSW coarse spaces compared to the respective RGDSW coarse spaces. In particular, the best iteration counts are obtained using the classical GDSW coarse space with the full null space. However, the parallel performance is clearly better when reducing the dimension of the coarse space by either omitting the rotational rigid body mode or by using the RGDSW coarse space; see also Table 3 for the number coarse components used in the GDSW and the RGDSW coarse spaces, which, together with the dimension of the employed subspace of the null space, determines the size of the coarse space. In total, the variant with the smallest coarse space, i.e., RGDSW without rotation, yields both the highest iteration counts but the best parallel performance. Hence, we will concentrate on this coarse space in the following experiments.

Moreover, we compare one-level and two-level Schwarz methods in Table 4. We observe that the one-level methods do not scale numerically. However, due to the geometry of the ice sheet, the increase in the iteration count of the one-level preconditioners is lower compared to usual fully three-dimensional domain decompositions. Due to the reuse strategies for the two-level methods used in this comparison, the setup cost for the one-level preconditioners is only slightly lower; even the coarse matrix is reused. However, due to numerical scalability, the two level methods perform clearly better in the solve phase.

6.1.2. Reuse strategies. In Table 5, we investigate the performance improvements due to the use of reuse strategies on the coarse level. As the baseline, we consider reusing the index sets (IS) and the symbolic factorization for the first level (SF1). We then consider reusing only the symbolic factorization of the coarse matrix (SF2) and coarse basis functions (CB) as well as also reusing the coarse matrix itself (CM). As can be observed, the iteration counts increase and, at the same time, the setup cost reduces if parts of the second level are reused. In particular, for lower numbers of MPI ranks and large subdomain problems, the setup cost is significantly

	IS & SF1			IS & SF1 & SF2 & CB			IS & SF1 & SF2 & CB & CM		
MPI	avg. its	avg.	avg.	avg. its	avg.	avg. its	avg. its	avg.	avg.
ranks	(nl its)	setup	solve	(nl its)	setup	solve	(nl its)	setup	solve
512	**41.9** (11)	25.10 s	**12.29 s**	42.6 (11)	14.99 s	12.50 s	46.7 (11)	**14.94 s**	13.81 s
1 024	**43.3** (11)	9.18 s	**5.85 s**	44.5 (11)	**5.65 s**	6.08 s	49.2 (11)	5.75 s	6.78 s
2 048	**41.4** (11)	4.15 s	**2.63 s**	42.7 (11)	3.11 s	2.79 s	47.7 (11)	**2.92 s**	3.10 s
4 096	**41.2** (11)	1.66 s	**1.49 s**	42.5 (11)	1.07 s	1.54 s	48.9 (11)	**0.95 s**	1.75 s
8 192	**40.2** (11)	1.26 s	**1.06 s**	42.0 (11)	1.20 s	1.16 s	50.1 (11)	**0.63 s**	1.35 s

Table 5

Comparison of different reuse strategies for the two-level RGDSW Schwarz preconditioner for the flow problem on the Antarctica mesh with 4 km horizontal resolution and 20 layers of elements in vertical direction and a total of 35.3 m degrees of freedom. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its). Lowest average iterations counts, setup times, and solve times in each row are marked in **bold**.

	1 comm. step		2 comm. step		3 comm. step	
MPI	avg.	avg.	avg.	avg.	avg.	avg.
ranks	setup	solve	setup	solve	setup	solve
512	15.38 s	13.8 s	**14.99 s**	**12.50 s**	15.75 s	13.85 s
1 024	5.68 s	6.25 s	5.65 s	**6.08 s**	**5.63 s**	6.10 s
2 048	**2.91 s**	3.27 s	2.94 s	**2.78 s**	3.40 s	2.75 s
4 096	1.35 s	3.77 s	**1.07 s**	**1.54 s**	1.15 s	1.56 s
8 192	2.5 s	12.22 s	**1.29 s**	**1.13 s**	**1.29 s**	1.17 s

Table 6

Variation of the number of communication steps for the scatter and gather operations on the coarse level for the RGDSW Schwarz preconditioner for the flow problem on the Antarctica mesh with 4 km horizontal resolution and 20 layers of elements in vertical direction and a total of 35.3 m degrees of freedom. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its). Lowest average iterations counts, setup times, and solve times in each row are marked in **bold**.

reduced. Due to the better overall performance, we will only consider results using IS & SF1 & SF2 & CB or IS & SF1 & SF2 & CB & CM for the following results using two-level preconditioners for the flow problem.

6.1.3. Parallelization aspects.

Here, we discuss two parallelization aspects.

First, we discuss the communication between all MPI ranks and the single MPI rank which computes the coarse problem, the *coarse rank*. In particular, both all-to-one and one-to-all communication patterns are necessary in our implementation: In the setup phase, the coarse matrix, which is computed by an RAP product on all MPI ranks, has to be communicated to the coarse rank. Then, in each linear iteration of the solve phase, the right hand side of the coarse problem has to be communicated from all ranks to the coarse rank and the corresponding solution has to be communicated back. As already discussed in [28, section 4.7], this type of communication does not perform well for large numbers of MPI ranks using the Trilinos import and export objects. In [28, section 4.7] Epetra import and export objects were employed, whereas their Tpetra counterparts are considered here. Therefore, we introduce nested sets of MPI ranks, beginning with all MPI ranks and ending with the single coarse rank, and perform the all-to-one and one-to-all communication using multiple steps; cf. [28, section 4.7] for a more detailed discussion.

In Table 6, we present corresponding results, varying the number of communication steps between one to three. As can be observed, using two or three communication steps, we obtain good the parallel scalability. However, if only a singe import/export

	OpenMP parallelization (512 MPI ranks)				MPI parallelization			
cores	OpenMP threads	avg. its (nl its)	avg. setup	avg. solve	MPI ranks	avg. its (nl its)	avg. setup	avg. its solve
512	1	**42.6** (11)	**14.99 s**	**12.50 s**	512	**42.6** (11)	**14.99 s**	**12.50 s**
1 024	2	**42.6** (11)	9.43 s	6.80 s	1 024	44.5 (11)	**5.65 s**	**6.08 s**
2 048	4	**42.6** (11)	5.50 s	4.02 s	2 048	42.7 (11)	**3.11 s**	**2.79 s**
4 096	8	42.6 (11)	3.65 s	2.71 s	4 096	**42.5** (11)	**1.07 s**	**1.54 s**
8 192	16	42.6 (11)	2.56 s	2.32 s	8 192	**42.0** (11)	**1.20 s**	**1.16 s**

Table 7

Comparison of increasing the numbers of OpenMP threads or MPI ranks for the RGDSW Schwarz preconditioner for the flow problem on the Antarctica mesh with 4 km horizontal resolution and 20 layers of elements in vertical direction and a total of 35.3 m degrees of freedom. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its). Lowest average iterations counts, setup times, and solve times in each row are marked in **bold**.

		Constant number of MPI ranks				128 MPI ranks per 5 layers			
# layers	# dofs	MPI ranks	avg. its (nl its)	avg. setup	avg. solve	MPI ranks	avg. its (nl its)	avg. setup	avg. solve
5	10.1 m	2 048	39.2 (11)	0.42 s	0.58 s	128	38.8 (12)	5.47 s	7.79 s
10	18.5 m		41.0 (11)	0.79 s	1.15 s	256	37.8 (11)	8.46 s	8.57 s
20	35.3 m		42.7 (11)	2.94 s	2.78 s	512	42.6 (11)	14.99 s	12.50 s
40	69.0 m		45.6 (12)	5.77 s	6.67 s	1 024	47.8 (12)	19.00 s	15.72 s
80	136.3 m		45.3 (15)	14.41 s	14.53 s	2 048	45.3 (15)	14.41 s	14.53 s

Table 8

Performance of the RGDSW Schwarz preconditioner for an increasing number of layers for the flow problem on the Antarctica mesh with 4 km horizontal resolution and 20 layers of elements in vertical direction. Left: constant number of MPI ranks and subdomains. Right: increasing the number of MPI ranks and subdomains proportial to the number of layers. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its).

call from Tpetra is performed in each scatter/gather operation, the parallel scalability deteriorates due to a significant communication overhead. In particular, the solve time, where one scatter and one gather operation is performed in each linear iteration, is increased significantly. Hence, in all other experiments, we use two communication steps.

In Table 7, we compare OpenMP parallelization and MPI parallelization. Starting with 512 MPI ranks, we increase the number of processor cores up to 8 192 using either OpenMP threads or a higher number of MPI ranks. As can be observed, MPI parallelization is clearly superior in this comparison even though the size of the coarse problem increases with an increasing number of MPI ranks and subdomains, whereas it stays constant for OpenMP parallelization. Only for large numbers of MPI ranks and subdomains, it may be reasonable to additionally use OpenMP parallelization since it does not further increase the coarse problem size. Alternatively, more levels could be added to the the GDSW type preconditioners; cf. [29, 30]. Hence, we will restrict ourselves to using MPI parallelization; only in the largest weak scalability study in subsection 6.1.5, we also show results using OpenMP parallelization in addition to MPI parallelization.

6.1.4. Increasing the number of layers of elements in vertical direction. In most of our numerical simulations, we use 20 layers of elements in vertical direction; this corresponds to a rather fine resolution in vertical direction, which would also be used in production runs of the land ice simulations. However, we are also interested in

1 OpenMP thread								
			IS & SF1 & SF2 & CB			IS & SF1 & SF2 & CB & CM		
MPI ranks	mesh	# dofs	avg. its (nl its)	avg. setup	avg. solve	avg. its (nl its)	avg. setup	avg. solve
32	16 km	2.2 m	24.1 (11)	11.97 s	9.47 s	**24.0** (11)	**11.18 s**	**9.45 s**
128	8 km	8.8 m	**32.0** (10)	14.08 s	**8.71 s**	32.6 (10)	**14.06 s**	8.93 s
512	4 km	35.3 m	**42.6** (11)	**14.99 s**	**12.50 s**	**42.6** (11)	16.14 s	14.19 s
2 048	2 km	141.5 m	**61.0** (11)	22.83 s	**19.76 s**	67.1 (11)	**22.65 s**	21.69 s
8 192	1 km	566.1 m	**67.1** (14)	17.36 s	**22.91 s**	73.0 (14)	**16.80 s**	28.48 s
4 OpenMP threads								
			IS & SF1 & SF2 & CB			IS & SF1 & SF2 & CB & CM		
MPI ranks	mesh	# dofs	avg. its (nl its)	avg. setup	avg. solve	avg. its (nl its)	avg. setup	avg. solve
32	16 km	2.2 m	**23.5** (11)	4.15 s	**3.25 s**	23.8 (11)	**3.93 s**	3.28 s
128	8 km	8.8 m	**32.0** (10)	4.97 s	2.85 s	32.6 (10)	**4.62 s**	**2.82 s**
512	4 km	35.3 m	**42.6** (11)	5.50 s	**4.02 s**	46.7 (11)	**5.27 s**	4.45 s
2 048	2 km	141.5 m	**61.0** (11)	7.36 s	**6.55 s**	67.1 (11)	**7.15 s**	7.34 s
8 192	1 km	566.1 m	**67.1** (14)	6.20 s	**7.39 s**	73.0 (14)	**5.75 s**	7.92 s

TABLE 9

Weak scalability studies for the RGDSW Schwarz preconditioner for the flow problem on the Antarctica mesh with 4 km horizontal resolution and 20 layers of elements in vertical direction. We consider the cases of 1 OpenMP thread (top) and 4 OpenMP threads (bottom) per MPI rank as well as IS & SF1 & SF2 & CB (left) and IS & SF1 & SF2 & CB & CM (right) reuse strategies. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its). Lowest average iterations counts, setup times, and solve times in each row are marked in **bold**.

investigating the influence of an increasing number of layers on the performance of our preconditioners. In Table 8, we employ the RGDSW preconditioner and fix the top surface mesh while increasing the number of vertical layers of elements from 5 up to 80. For both cases, keeping the number of MPI ranks fixed and increasing it proportional to the number of layers, the iterations counts are very robust. However, the number of nonlinear iterations increases slightly from 11 to 15. Note that we use 2048 MPI ranks for all problems in this experiment when we keep constant number of MPI ranks. This also allows comparing scalability of the solver for different problems to 2048 ranks. For example, even the 5 layer problem achieves 13.4x speedup in average solve going from 128 MPI ranks to 2048 MPI ranks demonstrating good parallel scalability.

6.1.5. Weak scaling. In Table 9, we provide four weak scalability studies, where we increase the number of MPI ranks proportional to the resolution of the top surface mesh; the number of vertical layers is again fixed to 20. In particular, we consider 1 or 4 OpenMP threads per MPI rank combined with the IS & SF1 & SF2 & CB and IS & SF1 & SF2 & CB & CM reuse strategies; cf. subsections 4.5 and 6.1.2.

We observe good weak scalability from 32 to 8 192 (1 OpenMP thread per MPI rank) and from 128 to 32 768 (4 OpenMP threads per MPI rank) processor cores. However, there is a moderate increase in the number of iterations, which is most likely caused by the unstructured domain decomposition, where subdomains with irregular shape and bad aspect ratio may occur in certain cases, in particular, at the boundary of the top surface mesh; cf. Figure 3. For all configurations, the setup time scales very well, whereas the increase in the solve time is more pronouced; however, except for the case of 1 OpenMP rank and IS & SF1 & SF2 & CB & CM reuse, the solve times does increase clearly less than the number of iterations.

Generally, we observe a speedup by a factor of approximately 3 when using 4 threads instead of 1 OpenMP thread. However, the former uses 4 times the number

	FROSch						MueLu		
	IS & SF1			IS & SF1 & SF2 & CB & CM			Vertical Semi-Coarsening		
MPI ranks	avg. its (nl its)	avg. setup	avg. solve	avg. its (nl its)	avg. setup	avg. solve	avg. its (nl its)	avg. setup	avg. solve
512	41.9 (11)	25.10 s	12.29 s	46.7 (11)	14.94 s	13.81 s	**31.0** (11)	**0.35 s**	**3.00 s**
1024	43.3 (11)	9.18 s	5.85 s	49.2 (11)	5.75 s	6.78 s.	**30.7** (11)	**0.32 s**	**1.66 s**
2 048	41.4 (11)	4.15 s	2.63 s	47.7 (11)	2.92 s	3.10 s	**31.0** (11)	**0.36 s**	**1.02 s**
4 096	41.2 (11)	1.66 s	**1.49 s**	48.9 (11)	0.95 s	1.75 s	**30.9** (11)	**0.80 s**	1.69 s
8 192	**40.2** (11)	1.26 s	**1.06 s**	50.1 (11)	**0.63 s**	1.35 s	48.5 (11)	1.05 s	2.55 s

TABLE 10

Comparison of the RGDSW Schwarz preconditioner with two different reuse strategies against MueLu algebraic multigrid for the flow problem on the Antarctica mesh with 4 km horizontal resolution and 20 layers of elements in vertical direction and a total of 35.3 m degrees of freedom. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its). Lowest average iterations counts, setup times, and solve times in each row are marked in **bold**.

of cores compared to the latter. Hence, OpenMP parallelization has to be carefully considered with respect to the size of the problems and the available parallelism.

6.1.6. Comparison against multigrid. As a final result for the velocity problem for Antarctica, we compare the strong scalability for the RGDSW preconditioner in the FROSch package to an algebraic multigrid preconditioner described in [50] and using MueLu. The method uses a vertical semi-coarsening approach designed for the ice sheet problems. As can be observed in Table 10, for small numbers of MPI ranks and subdomains, the total time is clearly higher for FROSch compared to MueLu. This is caused by the superlinear complexity of the direct solvers which are used to solve the problems on the overlapping subdomains. However, when increasing the number of subdomains and therefore reducing the size of the overlapping subdomains, we observe a better speedup compared to MueLu. We note that MueLu settings were not fine-tuned for this particular problem. However, it is fair to say that FROSch is competitive for large number of sub-domains especially considering the fact that FROSch is used almost as a black box.

6.2. Temperature problem for Greenland. As a second problem for land ice simulations, we consider the temperature problem described in subsection 2.2 for Greenland; see also Figure 4. In Table 11, we compare one-level Schwarz preconditioners and RGDSW preconditioner using one and two layers of algebraic overlap. As can be observed, already the one-level methods scale well since all subdomains are adjacent to the Dirichlet boundary, which is the whole upper surface; cf. subsection 2.2. Due to the lower setup and application cost of the one-level method, both the setup and the solve times are also lower. Therefore, one-level Schwarz methods are very well suited for solving the temperature problem, and hence, it is not necessary to add a second level. Note that the standalone steady-state temperature problem is not physically meaningful because the temperature equilibration is on time scales that are much larger than the velocity ones. For this reason, we focus our attention on the coupled problem.

6.3. Coupled problem for Greenland. Finally, we consider the coupled problem for the non-uniform Greenland meshes and present, for the first time, results for scalable monolithic two-level preconditioners for this problem. Note that the nonlinear iteration is very sensitive for the coupled problem. In particular, even though a very strict stopping tolerance of 10^{-9} is used for the GMRES iteration, changing the

One-level Schwarz						
	one layer of algebraic overlap			two layers of algebraic overlap		
MPI ranks	avg. its	avg. setup	avg. solve	avg. its	avg. setup	avg. solve
512	18.1 (11)	**0.42 s**	**0.35 s**	**17.1** (11)	0.51 s	0.40 s
1 024	23.7 (11)	**0.25 s**	**0.25 s**	**22.1** (11)	0.27 s	0.27 s
2 048	29.6 (11)	**0.16 s**	**0.17 s**	**27.6** (11)	0.23 s	0.20 s
4 096	39.8 (11)	**0.15 s**	**0.15 s**	**35.6** (11)	0.17 s	0.17 s
RGDSW (IS & SF1 & SF2 & CB)						
	one layer of algebraic overlap			two layers of algebraic overlap		
MPI ranks	avg. avg. its	avg. setup	avg. solve	avg. avg. its	avg. setup	avg. solve
512	19.5 (11)	**0.44 s**	**0.41 s**	**18.7** (11)	0.55 s	0.46 s
1 024	25.2 (11)	**0.28 s**	**0.29 s**	**23.9** (11)	0.35 s	0.33 s
2 048	31.5 (11)	0.26 s	**0.24 s**	**29.5** (11)	**0.25 s**	0.27 s
4 096	42.2 (11)	**0.25 s**	**0.27 s**	**38.2** (11)	**0.25 s**	0.29 s

TABLE 11

Comparison of one-level and RGDSW Schwarz preconditioners for the temperature problem on the Greenland mesh with 1-10 km horizontal resolution (fine mesh) and 20 layers of elements in vertical direction and a total of 1.9 m degrees of freedom. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its). Lowest average iterations counts, setup times, and solve times in each row are marked in **bold**.

fully coupled extensions							
		NR			IS & CB		
MPI ranks	dim V_0	avg. its (nl its)	avg. setup	avg. solve	avg. its (nl its)	avg. setup	avg. solve
256	1 400	100.1 (27)	4.10 s	6.40 s	**18.5** (70)	**2.28 s**	**1.07 s**
512	2 852	129.1 (28)	1.88 s	4.20 s	**24.6** (38)	**1.04 s**	**0.70 s**
1 024	6 036	191.2 (65)	1.21 s	4.76 s	**34.2** (32)	**0.66 s**	**0.70 s**
2 048	12 368	237.4 (30)	0.96 s	4.06 s	**37.3** (30)	**0.60 s**	**0.58 s**
decoupled extensions							
		NR			IS & CB		
MPI ranks	dim V_0	avg. its (nl its)	avg. setup	avg. solve	avg. its (nl its)	avg. setup	avg. solve
256	1 400	23.6 (29)	3.90 s	1.32 s	**21.5** (34)	**2.23 s**	**1.18 s**
512	2 852	27.5 (30)	1.83 s	0.78 s	**26.4** (33)	**1.13 s**	**0.78 s**
1 024	6 036	30.1 (29)	1.19 s	**0.60 s**	**28.6** (43)	**0.66 s**	0.61 s
2 048	12 368	36.4 (30)	0.69 s	0.56 s	**31.2** (50)	**0.57 s**	**0.55 s**

TABLE 12

Comparison of monolithic RGDSW Schwarz preconditioners with different coarse spaces **neglecting rotational coarse basis functions for the velocity degrees of freedom** *for the coupled problem on the Greenland mesh with 3-30 km horizontal resolution (coarse mesh) and 20 layers of elements in vertical direction and a total of 7.5 m degrees of freedom. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its). Lowest average iterations counts, setup times, and solve times in each row are marked in* **bold**.

preconditioner may result in significant variations in the number of nonlinear iterations; cf. Tables 12, 13, 15, and 16. Note again that, in this work, we report linear iteration counts averaged over the total number of Newton iterations, so that our results are not influenced much by the sensitivity of the nonlinear solver.

First, we compare different monolithic coarse spaces for a coarse Greenland mesh with 3-30 km horizontal resolution, 20 layers of elements in vertical direction, and a total of more than 7.5 m degrees of freedom. In order to focus only on the coarse basis, we only consider two following reuse strategies. On the one hand, we do not reuse any

fully coupled extensions							
		NR			IS & CB		
MPI ranks	dim V_0	avg. its (nl its)	avg. setup	avg. solve	avg. its (nl its)	avg. setup	avg. solve
256	1 750	99.3 (27)	4.20 s	6.35 s	**21.9** (30)	**2.35 s**	**1.22 s**
512	3 565	131.4 (28)	1.95 s	4.40 s	**22.8** (50)	**1.09 s**	**0.66 s**
1 024	7 545	261.7 (31)	1.22 s	5.47 s	**31.3** (29)	**0.73 s**	**0.61 s**
2 048	15 460	325.7 (27)	1.08 s	8.53 s	**41.7** (25)	**0.74 s**	**1.16 s**
decoupled extensions							
		NR			IS & CB		
MPI ranks	dim V_0	avg. its (nl its)	avg. setup	avg. solve	avg. its (nl its)	avg. setup	avg. solve
256	1 750	**22.0** (28)	3.98 s	**1.23 s**	22.8 (27)	**2.23 s**	1.28 s
512	3 565	**24.7** (32)	1.92 s	0.72 s	23.8 (39)	**1.11 s**	**0.69 s**
1 024	7 545	**31.9** (27)	1.23 s	**0.62 s**	33.1 (27)	**0.74 s**	0.76 s
2 048	15 460	**31.2** (38)	0.99 s	**0.77 s**	34.7 (34)	**0.69 s**	1.05 s

TABLE 13

Comparison of monolithic RGDSW Schwarz preconditioners with different coarse spaces **including rotational coarse basis functions for the velocity degrees of freedom** *for the coupled problem on the Greenland mesh with 3-30 km horizontal resolution (coarse mesh) and 20 layers of elements in vertical direction and a total of 7.5 m degrees of freedom. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its). Lowest average iterations counts, setup times, and solve times in each row are marked in* **bold**.

# subdomains		256	512	1 024	2 048	4 096
RGDSW	3-30 km	350	713	1 509	3 092	6 245
	1-10 km	-	721	1 536	3 230	6 615

TABLE 14

Number of coarse components Γ_i for the two non-uniform Greenland meshes with 3-30 km and 1-10 km horizontal resolution. The dimension of the coarse space is the number of coarse components multiplied by the dimension of the null space.

information from the first Newton iteration (NR), on the other hand, we only reuse index sets and the coarse basis (IS & CB); in both cases, we do not reuse symbolic factorizations because of variations in the sparsity pattern of the system matrix. In combination with these two reuse strategies, we consider monolithic RGDSW preconditioners (see subsection 4.3) with fully coupled extensions using (4.10) and decoupled extensions using (4.12), respectively. As in subsection 6.1.1, we consider neglecting the rotational coarse basis functions and including the rotational coarse basis functions for the velocity part in Table 12 and Table 13, respectively. We clearly observe that using the standard monolithic coarse space (without reuse of the coarse basis functions) does not yield a scalable two-level method. Adding the rotational coarse basis function even yields higher iterations counts compared to neglecting rotational coarse basis functions. However, using the decoupled extensions described in subsection 4.3 instead, we obtain a scalable monolithic RGDSW preconditioner. Moreover, it seems that the coupling terms in the first Newton iteration do not deteriorate the scalability. Hence, reusing the coarse basis from the first Newton iteration even yields a scalable preconditioner for both cases, the fully coupled and the decoupled extensions.

Moreover, as for the velocity problem (see subsection 6.1.1), the time to solution is lower when neglecting the rotational coarse basis functions due to the lower coarse space dimension; see also Table 14 for the numbers of interface components. Consequently, we will only consider the case of neglecting rotational coarse basis functions for the monolithic RGDSW coarse spaces in the following experiments.

	decoupled (NR)			fully coupled (IS & CB)			decoupled (IS & SF1 & CB)		
MPI	avg.	avg.	avg.	avg.	avg.	avg.	avg.	avg.	avg.
ranks	(nl its)	setup	solve	(nl its)	setup	solve	(nl its)	setup	solve
512	**41.3** (36)	18.78 s	**4.99** s	45.3 (32)	11.84 s	5.35 s	45.0 (35)	**10.53** s	5.36 s
1 024	53.0 (29)	8.68 s	4.22 s	**47.8** (37)	5.36 s	**3.82 s**	54.3 (32)	**4.59 s**	4.31 s
2 048	62.2 (86)	4.47 s	4.23 s	66.7 (38)	2.81 s	4.53 s	**59.1** (38)	**2.32 s**	**3.99 s**
4 096	**68.9** (40)	2.52 s	**2.86 s**	79.1 (36)	1.61 s	3.30 s	78.7 (38)	**1.37 s**	3.30 s

Table 15

Comparison of monolithic RGDSW Schwarz preconditioners with different reuse strategies for the coupled problem on the Greenland mesh with 1-10 km horizontal resolution (fine mesh) and 20 layers of elements in vertical direction and a total of 68.6 m degrees of freedom. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its). Lowest average iterations counts, setup times, and solve times in each row are marked in **bold**.

One-level Schwarz (NR)						
	$\delta = 1h$			$\delta = 2h$		
MPI	avg. its	avg.	avg.	avg. its	avg.	avg.
ranks	(nl its)	setup	solve	(nl its)	setup	solve
512	48.7 (35)	**11.3 s**	**5.41** s	**42.6** (33)	15.2 s	5.80 s
1 024	61.9 (40)	**5.29 s**	**4.75** s	**58.8** (30)	6.92 s	5.48 s
2 048	89.9 (30)	**2.52 s**	**5.70** s	**73.5** (34)	3.83 s	6.24 s
4 096	116.1 (31)	**1.17 s**	**3.68** s	**103.1** (33)	1.86 s	4.87 s
One-level Schwarz (NR & SF1)						
	$\delta = 1h$			$\delta = 2h$		
MPI	avg. its	avg.	avg.	avg. its	avg.	avg.
ranks	(nl its)	setup	solve	(nl its)	setup	solve
512	52.2 (32)	**10.16 s**	**5.88 s**	**42.6** (39)	13.80 s	5.77 s
1 024	66.2 (35)	**4.32 s**	**4.91 s**	**35.7** (72)	5.98 s	3.19 s
2 048	82.0 (37)	**2.07 s**	**5.27 s**	**68.5** (39)	3.20 s	5.81 s
4 096	120.39 (31)	**0.92 s**	**3.83 s**	**95.5** (32)	1.48 s	4.53 s

Table 16

Strong scaling study for monolithic one-level Schwarz preconditioners with one or two layers of algebraic overlap for the coupled problem on the Greenland mesh with 1-10 km horizontal resolution (fine mesh) and 20 layers of elements in vertical direction and a total of 68.6 m degrees of freedom. The linear iteration counts (avg. its), setup times (avg. setup), and solve times (avg. solve) are averaged over the number of Newton iterations (nl its). Lowest average iterations counts, setup times, and solve times in each row are marked in **bold**.

Next, we investigate different reuse strategies in Table 15 for a fine Greenland mesh with 1-10 km horizontal resolution, 20 layers of elements in vertical direction, and a total of more than 68 m degrees of freedom. As can be observed, the best parallel performance can be obtained when reusing the index sets (IS) as well as the symbolic factorization on the first level (SF1) and the coarse basis (CB) from the first Newton iteration. Note that reusing the symbolic factorization on the second level, the iteration counts always deteriorated in our experiments.

Finally, we also provide results for monolithic one-level Schwarz preconditioners in comparison to the two-level monolithic RGDSW preconditioner. As can be observed in Table 16, the iteration counts for the one-level preconditioners with one level of overlap are clearly higher compared to the RGDSW preconditioner with one layer of overlap in Table 15. Therefore, the solve time is reduced by adding an appropriate second level. On the other hand, the setup cost for the two-level methods is again higher; in particular, the additional coarse problem is also a fully coupled multi-physics problem in this case. The computing time for an overlap of two layers was higher for both the one-level and the two-level method.

Note that we observed that the matrix structure of the coupled problem is not well-suited for OpenMP parallelization of the node-level solver Pardiso. In particular, the speedup was always lower than a factor of 2 when using 4 OpenMP threads and one processor core per OpenMP thread. For the case of 4 096 MPI ranks, the speedup was even reduced to a factor of less than 1.2.

7. Conclusions.

We have presented a flexible preconditioning framework based on the GDSW method, which yields scalable and robust preconditioners for all considered land ice problems. In particular, the implementation of this framework in FROSch can be applied out-of-the-box; between the different problems, only minor changes of the input parameters are necessary. Moreover, to the best of our knowledge, we have presented the first scalable two-level method for the coupled problem for land ice simulations. Compared to the single physics problems, the extension operators have to be decoupled, which can easily be done be done by changing one parameter in FROSch. Otherwise, the coarse basis from the first Newton iteration also resulted in a scalable method.

The parallel results of several strong and weak scaling studies, involving different coarse space variants and reuse strategies as well as OpenMP parallelization and MPI communication aspects, prove both the robustness and numerical scalability of the methods as well as the parallel scalability of the implementation in FROSch.

Furthermore, we have observed that the direct solvers in our two-level method are the main bottleneck. On one hand, the direct solvers on the first level determine the computing time for a small number of MPI ranks and large subdomain problems. On the other hand, the direct solver on the coarse level may become the scaling bottleneck for very large numbers of MPI ranks and subdomains. The improvement of the subdomain and coarse solvers for these complex problems will be subject of future research.

Acknowledgments. Support for this work was provided through the SciDAC projects FASTMath and ProSPect, funded by the U.S. Department of Energy (DOE) Office of Science, Advanced Scientific Computing Research and Biological and Environmental Research programs. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231.

Disclaimer. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

REFERENCES

[1] A. Aschwanden, E. Bueler, C. Khroulev, and H. Blatter, *An enthalpy formulation for glaciers and ice sheets*, Journal of Glaciology, 58 (2012), pp. 441–457, https://doi.org/doi:10.3189/2012JoG11J088.

[2] E. Bavier, M. Hoemmen, S. Rajamanickam, and H. Thornquist, *Amesos2 and belos: Direct and iterative solvers for large sparse linear systems*, Scientific Programming, 20 (2012), pp. 241–255.

[3] L. Berger-Vergiat, C. A. Glusa, J. J. Hu, M. Mayr, A. Prokopenko, C. M. Siefert, R. S. Tuminaro, and T. A. Wiesner, *MueLu multigrid framework*. http://trilinos.org/packages/muelu, 2019.

[4] L. Berger-Vergiat, C. A. Glusa, J. J. Hu, M. Mayr, A. Prokopenko, C. M. Siefert, R. S. Tuminaro, and T. A. Wiesner, *MueLu user's guide*, Tech. Report SAND2019-0537, Sandia National Laboratories, 2019.

[5] D. J. Brinkerhoff and J. V. Johnson, *Data assimilation and prognostic whole ice sheet modelling with the variationally derived, higher order, open source, and fully parallel ice sheet model varglas*, The Cryosphere, 7 (2013), pp. 1161–1184, https://doi.org/10.5194/tc-7-1161-2013, https://tc.copernicus.org/articles/7/1161/2013/.

[6] J. Brown, B. Smith, and A. Ahmadia, *Achieving textbook multigrid efficiency for hydrostatic ice sheet flow*, SIAM Journal on Scientific Computing, 35 (2013), pp. B359–B375, https://doi.org/10.1137/110834512, https://doi.org/10.1137/110834512, https://arxiv.org/abs/https://doi.org/10.1137/110834512.

[7] M. Buck, O. Iliev, and H. Andrä, *Multiscale finite elements for linear elasticity: Oscillatory boundary conditions*, in Domain Decomposition Methods in Science and Engineering XXI, vol. 98 of LNCSE, Springer, 2014, pp. 237–245.

[8] X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 792–797.

[9] L. Cambier, C. Chen, E. G. Boman, S. Rajamanickam, R. S. Tuminaro, and E. Darve, *An algebraic sparsified nested dissection algorithm using low-rank approximations*, SIAM Journal on Matrix Analysis and Applications, 41 (2020), pp. 715–746.

[10] C. Chen, L. Cambier, E. G. Boman, S. Rajamanickam, R. S. Tuminaro, and E. Darve, *A robust hierarchical solver for ill-conditioned systems with applications to ice sheet modeling*, Journal of Computational Physics, 396 (2019), pp. 819–836.

[11] J. Church, P. Clark, A. Cazenave, J. Gregory, S. Jevrejeva, A. Levermann, M. Merrifield, G. Milne, R. Nerem, P. Nunn, A. Payne, W. Pfeffer, D. Stammer, and A. Unnikrishnan, *Sea Level Change*, Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2013, book section 13, pp. 1137–1216, https://doi.org/10.1017/CBO9781107415324.026, www.climatechange2013.org.

[12] C. R. Dohrmann, A. Klawonn, and O. B. Widlund, *Domain decomposition for less regular subdomains: overlapping Schwarz in two dimensions*, SIAM J. Numer. Anal., 46 (2008), pp. 2153–2168.

[13] C. R. Dohrmann, A. Klawonn, and O. B. Widlund, *A family of energy minimizing coarse spaces for overlapping Schwarz preconditioners*, in Domain decomposition methods in science and engineering XVII, vol. 60 of Lect. Notes Comput. Sci. Eng., Springer, Berlin, 2008, pp. 247–254.

[14] C. R. Dohrmann and O. B. Widlund, *An overlapping schwarz algorithm for almost incompressible elasticity*, SIAM J. Numer. Anal., 47 (2009), pp. 2897–2923.

[15] C. R. Dohrmann and O. B. Widlund, *An alternative coarse space for irregular subdomains and an overlapping Schwarz algorithm for scalar elliptic problems in the plane*, SIAM J. Numer. Anal., 50 (2012), pp. 2522–2537, https://doi.org/10.1137/110853959, https://doi.org/10.1137/110853959.

[16] C. R. Dohrmann and O. B. Widlund, *On the design of small coarse spaces for domain decomposition algorithms*, SIAM J. Sci. Comput., 39 (2017), pp. A1466–A1488.

[17] E. Efstathiou and M. J. Gander, *Why restricted additive Schwarz converges faster than additive Schwarz*, BIT, 43 (2003), pp. 945–959.

[18] O. Gagliardini, T. Zwinger, F. Gillet-Chaulet, G. Durand, L. Favier, B. de Fleurian, R. Greve, M. Malinen, C. Martín, P. Råback, J. Ruokolainen, M. Sacchettini, M. Schäfer, H. Seddik, and J. Thies, *Capabilities and performance of Elmer/Ice, a new-generation ice sheet model*, Geoscientific Model Development, 6 (2013), pp. 1299–1318, https://doi.org/10.5194/gmd-6-1299-2013, http://www.geosci-model-dev.net/6/1299/2013/.

[19] F. Habbal, E. Larour, M. Morlighem, H. Seroussi, C. P. Borstad, and E. Rignot, *Optimal numerical solvers for transient simulations of ice flow using the ice sheet system model (issm versions 4.2.5 and 4.11)*, Geoscientific Model Development, 10 (2017), pp. 155–168, https://doi.org/10.5194/gmd-10-155-2017, https://gmd.copernicus.org/articles/10/155/2017/.

[20] A. Heinlein, *Parallel Overlapping Schwarz Preconditioners and Multiscale Discretizations with Applications to Fluid-Structure Interaction and Highly Heterogeneous Problems*, PhD thesis, Universität zu Köln, Germany, 2016.

[21] A. Heinlein, C. Hochmuth, and A. Klawonn, *Fully algebraic two-level overlapping schwarz preconditioners for elasticity problems*, technical report, Universität zu Köln, December 2019, https://kups.ub.uni-koeln.de/10441/.

[22] A. Heinlein, C. Hochmuth, and A. Klawonn, *Monolithic overlapping Schwarz domain decomposition methods with GDSW coarse spaces for incompressible fluid flow problems*, SIAM J. Sci. Comput., 41 (2019), pp. C291–C316, https://doi.org/10.1137/18M1184047. Preprint https://kups.ub.uni-koeln.de/8355/.

[23] A. Heinlein, C. Hochmuth, and A. Klawonn, *Reduced dimension GDSW coarse spaces for monolithic Schwarz domain decomposition methods for incompressible fluid flow problems*, International Journal for Numerical Methods in Engineering, n/a (2019), https://doi.org/10.1002/nme.6258, https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6258. Preprint https://kups.ub.uni-koeln.de/8645/.

[24] A. Heinlein, C. Hochmuth, and A. Klawonn, *Fully algebraic two-level overlapping Schwarz preconditioners for elasticity problems*, tech. report, April 2020. Accepted for publication in the proceedings of the ENUMATH 2019 conference. April 2020. Preprint https://kups.ub.uni-koeln.de/id/eprint/10441.

[25] A. Heinlein, A. Klawonn, J. Knepper, O. Rheinbach, and O. B. Widlund, *Adaptive gdsw coarse spaces of reduced dimension for overlapping schwarz methods*, technical report, Universität zu Köln, September 2020, https://kups.ub.uni-koeln.de/12113/.

[26] A. Heinlein, A. Klawonn, S. Rajamanickam, and O. Rheinbach, *FROSch – A framework for parallel Schwarz preconditioners in Trilinos*. In preparation.

[27] A. Heinlein, A. Klawonn, S. Rajamanickam, and O. Rheinbach, *FROSch: A fast and robust overlapping Schwarz domain decomposition preconditioner based on Xpetra in Trilinos*, tech. report, April 2019. Accepted for publication in the proceedings of the International Conference on Domain Decomposition Methods 25. April 2019. Preprint https://kups.ub.uni-koeln.de/9018/.

[28] A. Heinlein, A. Klawonn, and O. Rheinbach, *A parallel implementation of a two-level overlapping Schwarz method with energy-minimizing coarse space based on Trilinos*, SIAM J. Sci. Comput., 38 (2016), pp. C713–C747, https://doi.org/10.1137/16M1062843. Preprint http://tu-freiberg.de/sites/default/files/media/fakultaet-fuer-mathematik-und-informatik-fakultaet-1-9277/prep/2016-04_fertig.pdf.

[29] A. Heinlein, A. Klawonn, O. Rheinbach, and F. Röver, *A Three-Level Extension of the GDSW Overlapping Schwarz Preconditioner in Two Dimensions*, Springer International Publishing, Cham, 2019, pp. 187–204, https://doi.org/10.1007/978-3-030-14244-5_10, https://doi.org/10.1007/978-3-030-14244-5_10. Preprint https://tu-freiberg.de/sites/default/files/media/fakultaet-fuer-mathematik-und-informatik-fakultaet-1-9277/prep/preprint_2018_04_heinlein_klawonn_rheinbach_roever.pdf.

[30] A. Heinlein, A. Klawonn, O. Rheinbach, and F. Röver, *A three-level extension of the gdsw overlapping schwarz preconditioner in three dimensions*, in Domain Decomposition Methods in Science and Engineering XXV, R. Haynes, S. MacLachlan, X.-C. Cai, L. Halpern, H. H. Kim, A. Klawonn, and O. Widlund, eds., Cham, 2020, Springer International Publishing, pp. 185–192. Preprint https://kups.ub.uni-koeln.de/9017/.

[31] M. R. Hestenes, E. Stiefel, et al., *Methods of conjugate gradients for solving linear systems*, Journal of research of the National Bureau of Standards, 49 (1952), pp. 409–436.

[32] I. J. Hewitt and C. Schoof, *Models for polythermal ice sheets and glaciers*, The Cryosphere, 11 (2017), pp. 541–551, https://doi.org/10.5194/tc-11-541-2017, https://tc.copernicus.org/articles/11/541/2017/.

[33] M. J. Hoffman, M. Perego, S. F. Price, W. H. Lipscomb, T. Zhang, D. Jacobsen, I. Tezaur, A. G. Salinger, R. Tuminaro, and L. Bertagna, *Mpas-albany land ice (mali): a variable-resolution ice sheet model for earth system modeling using voronoi grids*, Geoscientific Model Development, 11 (2018), pp. 3747–3780, https://doi.org/10.5194/gmd-11-3747-2018, https://www.geosci-model-dev.net/11/3747/2018/.

[34] T. Y. Hou and X.-H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media*, Journal of Computational Physics, 134 (1997), pp. 169 – 189, https://doi.org/http://dx.doi.org/10.1006/jcph.1997.5682, http://www.sciencedirect.com/science/article/pii/S0021999197956825.

[35] T. Isaac, G. Stadler, and O. Ghattas, *Solution of nonlinear stokes equations discretized by high-order finite elements on nonconforming and anisotropic meshes, with application to ice sheet dynamics*, SIAM Journal on Scientific Computing, 37 (2015), pp. B804–B833, https://doi.org/10.1137/140974407.

[36] A. Klawonn and L. Pavarino, *Overlapping Schwarz methods for mixed linear elasticity and Stokes problems*, Comput. Methods Appl. Mech. Engrg., 165 (1998), pp. 233–245.

[37] A. Klawonn and L. Pavarino, *A comparison of overlapping Schwarz methods and block preconditioners for saddle point problems*, Numer. Linear Algebra Appl., 7 (2000), pp. 1–25.

[38] A. Klawonn and O. Rheinbach, *A parallel implementation of dual-primal FETI methods for three-dimensional linear elasticity using a transformation of basis*, SIAM J. Sci. Comput., 28 (2006), pp. 1886–1906.

[39] W. Leng, L. Ju, M. Gunzburger, and S. Price, *A parallel computational model for three-*

dimensional, thermo-mechanical stokes flow simulations of glaciers and ice sheets, Communications in Computational Physics, 16 (2014), pp. 1056–1080, https://doi.org/10.4208/cicp.310813.010414a.

[40] W. Leng, L. Ju, M. Gunzburger, S. Price, and T. Ringler, *A parallel high-order accurate finite element nonlinear stokes ice sheet model and benchmark experiments*, Journal of Geophysical Research: Earth Surface, 117 (2012), https://doi.org/10.1029/2011JF001962, http://dx.doi.org/10.1029/2011JF001962.

[41] M. Perego, A. Barone, L. Bertagna, T. Hilbrandt, M. Hoffman, S. Price, and G. Stadler, *A steady-state thermo-mechanical solver for ice-sheet modeling*, The Cryosphere, ("in preparation").

[42] M. Perego, M. Gunzburger, and J. Burkardt, *Parallel finite-element implementation for higher-order ice-sheet models*, Journal of Glaciology, 58 (2012), pp. 76–88, https://doi.org/10.3189/2012JoG11J063.

[43] M. Rückamp, A. Humbert, T. Kleiner, M. Morlighem, and H. Seroussi, *Extended enthalpy formulations in the ice flow model issm version 4.17: discontinuous conductivity and anisotropic supg*, Geoscientific Model Development Discussions, 2020 (2020), pp. 1–18, https://doi.org/10.5194/gmd-2020-78, https://gmd.copernicus.org/preprints/gmd-2020-78/.

[44] Y. Saad and M. H. Schultz, *Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869, https://doi.org/10.1137/0907058, https://doi.org/10.1137/0907058, https://arxiv.org/abs/https://doi.org/10.1137/0907058.

[45] A. G. Salinger, R. A. Bartlett, A. M. Bradley, Q. Chen, I. P. Demeshko, X. Gao, G. A. Hansen, A. Mota, R. P. Muller, E. Nielsen, J. T. Ostien, R. P. Pawlowski, M. Perego, E. T. Phipps, W. Sun, and I. K. Tezaur, *Albany: Using component-based design to develop a flexible, generic multiphysics analysis code*, International Journal for Multiscale Computational Engineering, 14 (2016), pp. 415–438.

[46] C. Schoof and I. J. Hewitt, *A model for polythermal ice incorporating gravity-driven moisture transport*, Journal of Fluid Mechanics, 797 (2016), pp. 504–535, https://doi.org/10.1017/jfm.2016.251, https://www.cambridge.org/core/article/div-class-title-a-model-for-polythermal-ice-incorporating-gravity-driven-moisture-transport-div/F2EB0C5650440FFAF7899D1A38C9D424.

[47] I. K. Tezaur, M. Perego, A. G. Salinger, R. S. Tuminaro, and S. Price, *Albany/FELIX: a parallel, scalable and robust, finite element, first-order Stokes approximation ice sheet solver built for advanced analysis*, Geoscientific Model Development, 8 (2015), pp. 1–24, https://doi.org/10.5194/gmd-8-1-2015.

[48] I. K. Tezaur, R. S. Tuminaro, M. Perego, A. G. Salinger, and S. F. Price, *On the scalability of the albany/felix first-order stokes approximation ice sheet solver for large-scale simulations of the greenland and antarctic ice sheets*, Procedia Computer Science, 51 (2015), pp. 2026–2035.

[49] T. Trilinos Project Team, *The Trilinos Project Website*.

[50] R. Tuminaro, M. perego, I. Tezaur, A. Salinger, and S. F. Price, *A matrix dependent/algebraic multigrid approach for extruded meshes with applications to ice sheet modeling*, SIAM Journal on Scientific Computing, 38 (2016), pp. c504–c532.