SOLUTION OF NONLINEAR STOKES EQUATIONS
DISCRETIZED BY HIGH-ORDER FINITE ELEMENTS
ON NONCONFORMING AND ANISOTROPIC MESHES,
WITH APPLICATION TO ICE SHEET DYNAMICS∗

TOBIN ISAAC†, GEORG STADLER†, OMAR GHATTAS†§

Abstract. Motivated by the need for efficient and accurate simulation of the dynamics of the polar ice sheets, we design high-order finite element discretizations and scalable solvers for the solution of nonlinear incompressible Stokes equations. In particular, we focus on power-law, shear thinning rheologies commonly used in modelling ice dynamics and other geophysical flows. We use nonconforming hexahedral finite element meshes and the inf-sup stable finite element velocity-pressure pairings \( \mathbb{Q}_k \times \mathbb{Q}_{disc}^{k-2} \) or \( \mathbb{Q}_k \times \mathbb{P}_{disc}^{k-1} \), where \( k \geq 2 \) is the polynomial order of the velocity space. To solve the nonlinear equations, we propose a Newton-Krylov method with a block upper triangular preconditioner for the linearized Stokes systems. The diagonal blocks of this preconditioner are sparse approximations of the (1,1)-block and of its Schur complement. The (1,1)-block is approximated using linear finite elements based on the nodes of the high-order discretization, and is inverted using algebraic multigrid. This preconditioner is designed to be efficient on anisotropic meshes, which are necessary to match the high aspect ratio domains typical for ice sheets. Two different Schur complement approximations are compared, namely an inverse viscosity-weighted mass matrix and a least squares commutator approximation. In a comprehensive numerical study we find that our solver yields fast convergence that is independent of the mesh refinement, the occurrence of nonconforming element faces and the element aspect ratio, and depends only weakly on the polynomial finite element order. We simulate the ice flow in a realistic description of the Antarctic ice sheet derived from field data, and study the parallel scalability of our solver for problems with up to 615 million unknowns.

Key words. Viscous incompressible flow, nonlinear Stokes equations, shear-thinning, high-order finite elements, preconditioning, multigrid, Newton-Krylov method, ice sheet modeling, Antarctic ice sheet.

1. Introduction. We design high-order finite element discretizations and scalable solvers for incompressible nonlinear Stokes equations describing creeping flows of power-law rheology fluids. Applications include ice sheet dynamics [31], mantle convection [52], magma dynamics [43] and other problems involving non-Newtonian fluids [26]. Among the main challenges for the solution of these problems are the presence of local features that emerge from the nonlinear constitutive relation, the strongly varying and anisotropic coefficients arising upon linearization, the incompressibility condition leading to indefinite matrix problems, complex geometry and boundary conditions, a wide range of length scales that may require highly-adapted meshes with high aspect ratios, and large problem sizes that necessitate parallel solution on large supercomputers. Our approach to cope with these challenges uses adaptively refined nonconforming meshes, high-order inf-sup stable finite elements, and iterative Newton-Krylov solvers combined with multilevel preconditioning techniques. We focus in particular on the construction of efficient solvers and preconditioners for discrete

---

*Support for this work was provided by the U.S. Department of Energy Office of Science (DOE-SC), Advanced Scientific Computing Research (ASCR), Scientific Discovery through Advanced Computing (SciDAC) program, under award numbers DE-FG02-09ER25914 and DE-FC02-13ER2612, and the U.S. National Science Foundation (NSF) Cyber-enabled Discovery and Innovation (CDI) program under awards CMS-1028889 and OPP-0941678. Allocations of computing time on TACC’s Stampede under XSEDE, TG-DPP130002, and on ORNL’s Titan, which is supported by the Office of Science of the DOE under DE-AC05-00OR22725 are gratefully acknowledged.

†Institute for Computational Engineering & Sciences, The University of Texas at Austin, Austin, TX, USA
§Department of Geological Sciences and Department of Mechanical Engineering, The University of Texas at Austin, Austin, TX, USA

arXiv:1406.6573v1 [cs.NA] 25 Jun 2014
systems resulting from high-order discretizations.

High-order finite element methods for partial differential equations (PDEs) are attractive because, in many situations, the discrete solution converges rapidly to the true solution as the approximation order \( k \) is increased or the characteristic mesh size \( h \) is decreased. However, this increased accuracy per degree of freedom compared to low-order methods does not automatically translate into increased accuracy per unit of computational work. This is due to the fact that matrices arising from high-order discretizations are denser and, thus, more expensive to apply and invert. The cost of applying a matrix arising from a high-order discretization can be reduced drastically if the work is shifted from memory operations to floating point operations. This can be achieved using, and matrix-free implementations and tensor-product approximation spaces and element operations on hexahedral finite element meshes. To precondition matrices arising from high-order discretizations, low-order preconditioners based on the nodes of the high-order discretization have proven efficient \([5, 10, 14, 29]\). These preconditioners allow fast construction and the use of methods established for low-order discretizations.

Our approach to solving the nonlinear Stokes equations is an inexact Newton-Krylov method, with a block preconditioning strategy for the linearized equations, built from preconditioners for the \((1,1)\)-block and for its Schur complement. We consider a power-law rheology that involves the second invariant of the strain rate tensor, for which the Newton linearization results in a fourth-order anisotropic tensor viscosity. We pay particular attention to the interplay between discretization and solver: seemingly minor differences in either the discretization or the low-order preconditioner can have a large impact on the performance of conventional solution methods for both diagonal blocks of the preconditioner.

Our driving application is the simulation of the dynamics of continental-scale ice flows, which is a critical component of coupled climate modeling. Predicting the contribution of ice sheets to sea-level rise is difficult because of the complexity of accurately modeling ice sheet dynamics for the full polar ice sheets and the large uncertainties in unobservable parameters governing these dynamics \([44, \text{Chapter 10, Appendix 6}]\). To address these uncertainties, significant effort has been focused on the development of inverse methods to infer ice sheet model parameters from observations \([46, 49]\). These inverse methods require the repeated solution of ice flow equations for numerous parameter fields that may vary over wide ranges, and many also require the repeated solution of related adjoint ice flow equations. Hence, inverse methods particularly stress the efficiency and robustness of solvers for nonlinear Stokes equations.

A particular difficulty in ice sheet simulations is the high aspect ratio of the computational domains, which is inherited by the discretization, leading to anisotropic meshes. Discretizations with high-aspect ratio elements (and problems with highly anisotropic material properties, which have many of the same properties) are known to be challenging for implicit solvers and preconditioners. The development of robust solvers for high aspect ratio domains is also important in other earth science and climate modeling problems. In ocean flow models, for instance, three-dimensional implicit PDE models are now being used \([36]\), whereas in the past they were often replaced by asymptotic two-dimensional approximations.

Related work. Several recent articles develop scalable solvers for Stokes problems with varying viscosity \([7, 9, 19, 21, 25, 27, 42]\). These contributions use low-order stable or stabilized finite elements for the discretization of the Stokes equations and address
nonlinearity mostly via a Picard fixed point approach. Scalable solvers for high-order
discretizations of nonlinear scalar problems and extensions to linear incompressible
flow problems are studied in [5]. Various asymptotics-based approximations of the
Stokes equations are used for ice sheet and glacier modeling, which reduce the in-
definite Stokes equations to positive definite elliptic systems. These simplifications
are justified by the large differences between horizontal and vertical components of
the velocity; we refer to [30] for a comparison and discussion of the validity of these
different models. Ice sheet simulations using the full nonlinear Stokes equations can
be found, e.g., in [22,41,48].

**Contributions.** One of the main contributions of this paper is the design of dis-
cretizations, solvers and preconditioners that allow the fast and scalable iterative
solution of nonlinear Stokes problems. In particular, we obtain convergence that is
independent of the mesh size, the presence of nonconforming elements in the mesh,
and the element aspect ratio, and depends only weakly on the polynomial order.
Another contribution is the extension of low-order preconditioners for high-order dis-
cretized problems to meshes with nonconforming and high aspect ratio elements. In
addition to analyzing our solver techniques on workstation-sized model problems, we
also demonstrate their performance and scalability on a series of larger problems re-
quiring a distributed memory parallel implementation, including a simulation of the
dynamics of the Antarctic ice sheet. The simulation uses a geometry and temper-
ature field derived from field data and constitutes what we believe to be the first
highly resolved nonlinear Stokes-based continental scale simulations of the Antarctic
ice sheet.

**Limitations.** We employ our own code framework for all computations that re-
quire information about the discretization (matrix-free nonlinear and linear residual
calculations, and matrix construction for preconditioning), but we use PETSc [3] for
generic numerical linear algebra, e.g., Krylov solvers. This framework allows for flex-
ibility in choosing between well-tested, high-quality implementations of established
numerical methods, such as matrix factorization and algebraic multigrid, but this
flexibility comes at the cost of fine-grained control over the behavior of these methods,
e.g., over the component operations of algebraic multigrid, as discussed in section 5.1.

A specific combination of factors can degrade the performance of our solver,
namely high aspect ratio elements combined with weakly constraining basal boundary
conditions and significant variations in the effective viscosity and basal topogra-
phy. This is caused by an unstable incomplete matrix factorization, when used as a
smoother in the preconditioning of the (1,1)-block of the linearized Stokes operator.
This issue can be avoided by using a smoother that converges more robustly but is
slower than the incomplete factorization smoother in other problems.

Our simulations of ice sheet dynamics use a fixed temperature field and geome-
try. Simulations of evolving ice sheets would require coupling of the nonlinear Stokes
equations with a time-dependent advection-diffusion equation for the evolving tem-
perature, and with a kinematic equation for the evolution of the ice sheet surface.
However, the solvers presented in this paper carry over as important components in a
time-stepping procedure for the simulation of time-evolving nonlinear viscous flows.

**Overview.** This paper is organized as follows. In section 2, we discuss the form
of the nonlinear Stokes equations and boundary conditions that are the focus of this
work, their variational formulation, and their linearization. In section 3, we present
stable mixed-space finite element discretizations and a discussion of adaptive mesh
refinement. In section 4, we give an overview of our approach to solving the resulting
discrete system of nonlinear equations and the linearized counterparts. The precon-
ditioner for the (1,1)-block of the linearized Stokes systems that arise at each Newton
iteration, which is a critical component of the linear solver, is presented in detail in
section 5, followed by a discussion on the preconditioner for the Schur complement of
the (1,1)-block in section 6. We test our solver on a model geometry in section 7, and
then on a discretization of the Antarctic ice sheet in section 8, where we also study
the scalability of our method. We conclude with a discussion in section 9.

2. Nonlinear incompressible Stokes equations. After specifying the Stokes
equations with strain rate thinning power-law rheology in section 2.1, we present the
corresponding variational form and argue existence of a unique solution in section 2.2.
The linearization of the nonlinear equations is presented in section 2.3.

2.1. Problem statement. On an open, bounded domain $\Omega \subset \mathbb{R}^3$ we consider
the incompressible Stokes equations

\begin{align}
- \nabla \cdot \sigma &= f, \quad x \in \Omega, \\
\nabla \cdot u &= 0, \quad x \in \Omega,
\end{align}

(2.1a)

where $u$ is the flow velocity and $f$ is a body force. The Cauchy stress tensor $\sigma$ depends
on the strain rate tensor $D(u) = \frac{1}{2}(\nabla u + \nabla u^T)$, its second invariant $D_\mathrm{II}(u) := \frac{1}{2}D(u) : D(u)$ and, possibly, other physical quantities such as a temperature field. Here, “$:$”
denotes the Frobenius product between tensors $A = (A_{ij})$ and $B = (B_{ij})$ defined by
$A : B = \sum_{i,j} A_{ij} B_{ij}$. In the ice sheet problem, which is our driving application, the
stress tensor is given by Glen’s flow law

\[ \sigma = pI + B(T)(D_{\Omega}(u) + \varepsilon \frac{1}{1-n} D u), \]

(2.2)

where $p$ is the pressure, $B(T)$ a positive-valued function of temperature $T$, $n \geq 1$ is
the strain rate exponent, and $\varepsilon > 0$ a small regularization parameter that prevents
infinite effective viscosity for $n > 1$. For $n = 1$, eq. (2.2) reduces to a linear rheology,
and it describes a strain-rate weakening non-Newtonian fluid for $n > 1$. A common
value used for modeling the flow of glacial ice is $n = 3$. To complete the definition of
the boundary value problem eq. (2.1), it remains to specify the boundary conditions.

In ice sheet simulations, different parts of the boundary require different combi-
nations of Dirichlet, Neumann, and Robin-type boundary conditions. This makes ice
flow a good problem for developing methods for other creeping flow problems with
complicated boundary conditions. At the ice-air interface, the homogeneous Neu-
mann condition $\sigma n = 0$ holds. At the ice-water interface, the normal stress matches
the hydrostatic water pressure, i.e., $\sigma n \cdot n = -p_w$ and the tangential components
of the boundary traction vanish. At the base of the ice sheet, complex interactions
occur between ice, water, rock, and till. In cold regions, the ice sticks to the bedrock,
while in temperate regions, water accumulates at the base and the ice can slide but is
subject to some amount of friction. A general way to describe these phenomena is to
use a Dirichlet condition in normal direction to describe melting and freezing at the
base of the ice sheet, combined with a Robin-type sliding law relating the tangential
component of velocity $T_\parallel u = (I - n \otimes n) u$ to the tangential component of the stress
through a function $\beta(\cdot, \cdot, \ldots)$, i.e.,

\[ T_\parallel \sigma n = -\beta(|T_\parallel u|, T, \ldots). \]

(2.3)

Physically realistic descriptions of sliding must include the dependence of $\beta$ on $|T_\parallel u|
[59], which makes eq. (2.3) a nonlinear boundary condition for the flow; in this work,
However, we consider the linear case $\beta'(T, u) = \beta(x)T_n$, and no basal freezing or melting. To summarize, we use the following boundary conditions for the base $\Gamma_R$:

\[
\begin{align*}
T_n \sigma n + \beta(x)T_n u &= 0 \quad & x \in \Gamma_R, \\
\mathbf{u} \cdot n &= 0 \quad & x \in \Gamma_R.
\end{align*}
\]

Polar ice sheets have a characteristic depth of less than 5 kilometers, while they extend horizontally for thousands of kilometers. Because of this difference between length scales, modelers often simplify eq. (2.1) using models based on asymptotic expansions that require assumptions about the magnitude of the velocities and stresses in the ice sheet, for instance, the shallow ice approximation [31] and the hydrostatic approximation [4]. The assumptions justifying these simplifications do not hold for the entire ice sheet, which has led to approaches that combine simplified models in the interior with Stokes equations at outlet glaciers [53]. To avoid these complications, we do not use simplified models and focus on the efficient solution of the Stokes equations eq. (2.1) instead.

2.2. Variational formulation. Here, we define a variational form of eqs. (2.1) and (2.4) that defines the fields $(\mathbf{u}, p)$ as the unique solution in a to-be-specified vector space $\mathcal{V} \times \mathcal{M}$ to

\[
\int_\Omega \mu(u)D(v):D(u) = p\nabla \cdot v - q\nabla \cdot u \, dx + \int_{\Gamma_R} \beta T_n v \cdot T_n u \, ds = f(v),
\]

for all $(v, q) \in \mathcal{V} \times \mathcal{M}$, where $\mu(u) = \mu(u, T) = B(T)(D_0(u) + \varepsilon)^\frac{1-n}{2n}$, $f$ is the sum of body and boundary forces. We assume that $B(T) \in \mathcal{L}^\infty(\Omega)$ is uniformly bounded from below, that $\beta \in \mathcal{L}^\infty(\Gamma_R)$ is nonnegative and that $\partial \Omega$ is Lipschitz. For simplicity, we assume that the Dirichlet boundary conditions are homogeneous and are incorporated in $\mathcal{V}$.

For a similar problem, Jouvet and Rappaz [35] show that a unique solution $(\mathbf{u}, p)$ exists in the Dirichlet-conforming subspace of $[W^{1,r}(\Omega)]^3 \times \mathcal{L}^r(\Omega)$, where $r = 1 + 1/n$ and $r' = 1 + n$. In appendix A we show that eq. (2.5) is well-posed in a slightly modified pair of spaces.

2.3. Newton linearization. The Newton linearization of eq. (2.5) about a velocity-pressure pair $(\mathbf{u}, p)$ are equations whose solution $(\tilde{\mathbf{u}}, \tilde{p}) \in \mathcal{V} \times \mathcal{M}$ satisfies

\[
\int_\Omega D(v):(\mu'(u)D(\tilde{u})) - \frac{\partial \mu(u)}{\partial u} D(u) \otimes D(u) = \mu(u) \left( I - \frac{n-1}{2n} D(u) \otimes D(u) \right) + \varepsilon
\]

and $r(\cdot)$ is the residual of eq. (2.5). Here, “$\otimes$” denotes the outer product between two second-order tensors. Compared to the Newton linearization eq. (2.6), the Picard linearization of eq. (2.5) neglects the anisotropic part of the fourth-order tensor $\mu'(u)$. Using a finite element discretization of eqs. (2.5) and (2.7) is only marginally more complex than the corresponding Picard linearization, as the action of $\mu'(u)$ on $D(\tilde{u})$, which is all that is required in a finite element method can be computed using Frobenius products with trial and test functions. The operator $\mu'(u)$ is also found in the adjoint equations corresponding to eq. (2.5), which are used in inverse methods to infer uncertain parameters from observations [46, 49].

5
3. Discretization. Our goal for discretizing eqs. (2.5) and (2.6) is to obtain discrete solutions that converge to the continuous solutions rapidly in terms of the number of unknowns and time-to-solution. Our approach combines locally refined meshes with high-order finite element approximation spaces. The adaptive meshes we use are discussed in section 3.1, the finite element approximation spaces for the Stokes equations are described in section 3.2, and computational aspects of the discretization are addressed in section 3.3.

3.1. Meshing. We use a hierarchical approach to mesh refinement, which starts with a coarse mesh of conforming hexahedra. This coarse mesh is expected to roughly describe the geometry of the domain \( \Omega \), and its elements should be well-shaped. The fine mesh used for the finite element discretization is obtained by isotropic hierarchical refinement of this coarse mesh. Refinement can be used to improve the resolution of the geometry of \( \Omega \) or the resulting small hexahedra can simply inherit the geometry, which ensures that the newly created hexahedra are as well-shaped as the original hexahedron. For practical as well as numerical reasons, we require our refined meshes to obey a 2:1 condition, where neighboring hexahedra can differ by only one level of refinement, as illustrated in fig. 3.1. We use the p4est library to manage refinement, to enforce this 2:1 condition and to partition the mesh between processes in parallel computations [8,32].

Meshes for the simulation of ice flow must address the different length scales inherent in the problem. To accurately capture the vertical variations of the state variables, a minimum vertical mesh resolution is necessary. Most ice sheet models use \( \sim 10 \) nodes in each vertical column and have a horizontal resolution of 5 km. Since the average thickness of the polar ice sheets is \( \sim 2 \) km, the width-to-height aspect ratio \( \phi \) of these discretizations is \( \sim 25 \).\(^1\) The Antarctic ice shelves, moreover, are typically \( \sim 500 \) m thick, so the same horizontal resolution results in \( \phi \sim 100 \). We seek discretizations and methods for the Stokes equations that support these aspect ratios.

3.2. Finite element discretization. In this section, we describe the finite element spaces used to discretize the velocity and pressure spaces \( V \) and \( M \).

3.2.1. Discrete velocity spaces. Given a mesh \( \mathcal{T} \) of possibly nonconforming hexahedra \( \{K_i\} \), we define a finite-dimensional subspace of \( C^0(\Omega) \) using isoparametric Lagrange finite elements. The \( \mathcal{N}_k := (k+1)^3 \) nodes \( \Xi_k = \{ \xi_{rs} \} \) that define our Lagrange finite elements are the tensor-product Gauss-Lobatto nodes of polynomial degree \( k \) on the reference domain \( \hat{K} = [-1,1]^3 \). These basis functions span \( Q_k(\hat{K}) \), the space of functions on \( \hat{K} \) that are the univariate polynomials of degree at most \( k \) in each of the coordinate directions. We map \( \hat{K} \) to an element \( K_i \) by \( \varphi_i \in Q_k(\hat{K}) \) and use the tensor-product Gauss nodes of order \( k \) for numerical quadrature. We define the finite-dimensional space

\[ V_{\mathcal{T},k} = \{ v \in C^0(\Omega) : \forall K_i \in \mathcal{T}, v \circ \varphi_i \in Q_k(\hat{K}) \}, \]

and the velocity space \( W_{\mathcal{T},k} = [V_{\mathcal{T},k}]^3 \cap V \). For a conforming mesh, the set of element nodes \( \Xi_k \) naturally defines a set of global nodes \( X_{\mathcal{T},k} \) for \( V_{\mathcal{T},k} \) by the images of element nodes. For a nonconforming mesh \( \mathcal{T} \), however, not all element nodes correspond to

\(^1\)In glaciology, one often uses the thickness-to-width aspect ratio \( \epsilon = \phi^{-1} \) as the relevant limit in asymptotic expansions is \( \epsilon \to 0 \). In this work, however, we prefer using \( \phi \) because we consider the thickness of an ice sheet to be its characteristic length, in which we also measure its horizontal extent.
The nodal values along a nonconforming interface. Shown in green are the Gauss-Lobatto nodes of the smaller element, which do not align with those of the larger element: function values at these nodes depend on the values at the nodes of the larger element, so they are not included in the global nodal basis. The matrix \( R_i \) that interpolates a function to the nodes of the smaller element must interpolate cubic polynomials to the hanging nodes. This polynomial interpolation is dense: the value at each of the hanging nodes is dependent on all of the independent nodes. For two-dimensional nonconforming interfaces, \( R_i \) is defined by tensor-product polynomial interpolation.

In general, function values at the nodes of element \( K_i \) must be interpolated from the global vector of nodal values by a restriction matrix \( R_i \). If \( K_i \) has no hanging nodes, then \( R_i \) is simply a one-to-one association of \( \Xi_k \) to a subset of \( \mathcal{X}_{\mathcal{T},k} \); if \( K_i \) has hanging nodes, then \( R_i \) interpolates values as described in fig. 3.1. We use identical trial and test spaces, so a global nodal matrix \( A \) is assembled from element nodal matrices \( \{ A_i \} \) by

\[
A = \sum_i R_i^T A_i R_i.
\]

3.2.2. Discrete pressure spaces. We use inf-sup stable mixed finite element spaces to avoid the artificial compressibility that can be introduced by stabilized discretizations of incompressible flow. Additionally, to satisfy element-wise incompressibility, we favor piecewise discontinuous pressure spaces \( M_{\mathcal{T},k} \). This mass conservation is particularly important for ice sheet simulations, where the change of the mass of the ice sheet is an important quantity of interest in climate projections.

The two most common choices for approximation on the reference cube are \( P_{k-1}(\hat{K}) \), which is the space of polynomials on \( \hat{K} \) of degree at most \((k - 1)\), and \( Q_{k-2}(\hat{K}) \). We study two possibilities for inf-sup stable velocity-pressure finite element spaces. The pairing \( Q_k(\hat{K}) \times P_{k-1}(\hat{K}) \) has an optimal order of convergence, and has an inf-sup constant that is independent of \( k \) and of the type of hierarchical local mesh refinement we use [28]. Its inf-sup stability, however, degrades with increasing \( \phi \) [2]. We find (see section 6) that this degradation can be significant for the element aspect ratios in our meshes. An alternative pairing is \( Q_k(\hat{K}) \times Q_{k-2}(\hat{K}) \), which has a suboptimal order of convergence, but its inf-sup stability is uniform with respect to boundary layer refinement making it appropriate for large values of \( \phi \) [55]. For this pairing, the inf-sup constant decreases as \( O(k^{-1}) \); however, for the moderate values of \( k \) used in this work, this dependence is not problematic.

3.3. Computational aspects. For a mesh with \( N_{el} \) elements, the number of degrees of freedom \( N_{dof} \) in a \( k \)-order finite element discretization is \( \mathcal{O}(N_{el} k^3) \), and the
number of nonzero entries in the matrix for a system of equations defined on that space is \( O(N_{el}k^6) = O(N_{dof}k^3) \). This means that sparse matrix-vector products (matvecs) are not efficient for large values of \( k \) in terms of memory operations. We therefore compute nonlinear residuals and apply linear operators using a matrix-free approach to finite elements, where only the coefficients and fields that define an operation are stored in memory and the operation’s application to a specific vector is assembled from all element contributions at the time of application. This approach requires \( O(N_{dof}) \) memory operations per matvec or residual. For high-order elements, it is thus better suited to modern computer architectures, where the bandwidth for memory operations is much narrower than the bandwidth for floating-point operations. This reduction in memory operations comes at the expense of more floating point operations, but the tensor structure of \( Q_k(\hat{K}) \) allows for all such element computations to be reduced to repeated applications of one-dimensional compute kernels, which can be highly optimized \([5,15]\). The same one-dimensional kernels are used to apply the restriction operators \( R_i \) for hanging nodes.

4. **Newton-Krylov method for nonlinear Stokes equations.** Our goal is to design a robust and scalable solver for the nonlinear Stokes equations eqs. (2.1) and (2.2) with boundary conditions that include eq. (2.4). Ideally, the convergence should be stable with respect to: (1) the element size and the mesh refinement pattern, (2) parameters in the rheology \( \mu(u) \), (3) the Robin coefficient field \( \beta \), (4) the polynomial order \( k \), and (5) the element aspect ratio \( \phi \). Here, we propose an inexact Newton-Krylov solver, outlined in sections 4.1 and 4.2 below. In section 4.3, we present a test problem that allows us to study the effects of the factors listed above. Variations of this problem will be used in sections 5 and 6 to analyze and optimize the convergence of our linear solver, and in section 7 for a nonlinear Stokes problem.

4.1. **Newton’s method for nonlinear Stokes equations.** Given a velocity and pressure pair \((u,p)\), we (approximately) solve eq. (2.6) for a search direction \((\tilde{u}, \tilde{p})\). We then conduct a line search in the direction \((\tilde{u}, \tilde{p})\) using the weak Wolf conditions \([47]\) to guarantee that the nonlinear residual decreases. Each Newton update is computed inexactly via a Krylov-space iterative method, but with a tolerance that decreases in subsequent steps so as to guarantee quadratic convergence of the Newton iterations \([17]\) close to the solution. We ensure that discretization and differentiation commute, so that the Jacobian obtained from discretizing eq. (2.6) is equivalent to differentiating the discretization of eq. (2.5). Discretization with one of the stable finite element pairs discussed in section 3.2 results in a linear system with the symmetric saddle-point system matrix

\[
A(u) = \begin{pmatrix}
F(u) & B^T \\
B & 0
\end{pmatrix},
\]

where the (1,1)-block \( F(u) \) is the discretization of the sum of the \( \beta \) and \( \mu'(u) \) terms in eqs. (2.6) and (2.7), and \( B \) is the discretized divergence operator.

4.2. **Preconditioned Krylov method for linearized Stokes equations.** We solve systems involving \( A \) using preconditioned Krylov space methods, typically restarted GMRES, or, if the preconditioner is not stationary, its flexible variant FGMRES \([50]\). As is well known, the performance of Krylov methods critically depends on the availability of an efficient preconditioner \( \tilde{A} \) for \( A(u) \). In the following, we use the notation \( \tilde{A} = A(u) \) and \( F = F(u) \), i.e., in our notation we neglect the dependence of \( F \) and \( A \) on \( u \). Due to the elliptic nature of \( F \), a purely local preconditioner
for $A$ cannot provide $h$-independent convergence and a multilevel preconditioner is required. There are two main approaches for multilevel preconditioners for incompressible flow problems, namely monolithic and block preconditioning approaches. The former approximates the saddle point system on multiple levels and uses approximate local solution saddle point solutions [13,34]. This approach typically requires a geometric mesh hierarchy or involves stabilized discretizations. In contrast, block preconditioners are built from preconditioners for $F$ and for the Schur complement with respect to the $(1,1)$-block, $S := -BF^{-1}B^T$. They allow to build on existing solvers for elliptic systems and do not impose restrictions on the discretization underlying $A$. Due to this flexibility, we follow this latter approach and use an upper-triangular block preconditioner $\tilde{A}$, such that the preconditioned system becomes

$$AA^{-1} = \begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \tilde{F} & B^T \\ 0 & \tilde{S} \end{pmatrix}^{-1} = \begin{pmatrix} I + \varepsilon_F & \varepsilon_F B \tilde{S}^{-1} \\ BF^{-1} & I + \varepsilon_S \end{pmatrix},$$

where $\varepsilon_F = I - F\tilde{F}^{-1}$ and $\varepsilon_S = I - (-B\tilde{F}^{-1}B^T)\tilde{S}^{-1}$. Here, the matrix $\tilde{F}$ is an approximation of the $(1,1)$-block $F$, and $\tilde{S}$ is an approximation to the Schur complement $S$ with respect to the $(1,1)$-block. If $\varepsilon_F$ and $\varepsilon_S$ vanished, $AA^{-1}$ would be a lower triangular matrix with all eigenvalues clustered at 1, and a preconditioned Krylov method would converge in two iterations. Thus, our target is to devise $F^{-1}$ and $\tilde{S}^{-1}$ such that $\varepsilon_F$ and $\varepsilon_S$ are small and $A$-preconditioned Krylov methods converge quickly. Additionally, it is important that the setup time for the preconditioner $\tilde{A}$ is small, because $\tilde{A}$ has to be recomputed as the Jacobian $F = F(u)$ changes. In sections 5 and 6, we develop block preconditioners $\tilde{F}$ and $\tilde{S}$ and study their properties.

4.3. Test problem setup. Our test problems are based on problem C from the Ice Sheet Model Intercomparison Project [48]: $\Omega$ is a cutout of an “infinite slab”, i.e., a sheet that is periodic in $x$- and $y$-directions. We use homogeneous Neumann boundary conditions on the top surface of $\Omega$; on its base, we employ homogeneous Dirichlet conditions in the normal direction and Robin-type conditions in the tangential directions, as in eq. (2.4). We use the same boundary conditions when testing systems in just the $(1,1)$-block $F$. In linear test problems, we use $\mu = 1$ and $\beta = 1$ for the material and the sliding coefficients, unless specified otherwise.

To study the behavior of our solvers in the presence of nonconforming faces, we use meshes for $\Omega$ that have nonconforming faces throughout, as illustrated in fig. 4.1. We use two base meshes: $\mathcal{T}_{xy}$, which has nonconforming interfaces in only the $x$ and $y$-directions, and $\mathcal{T}_{xyz}$, which has nonconforming interfaces in all directions. In all test problems, $\Omega$ has unit thickness, but we vary the length and width of the domain. The finite elements stretch with the domain, so that the number and pattern of elements is the same, but the element aspect ratio $\phi$ varies between 1 and 100; see fig. 4.1b.

For linear test problems, we test the effectiveness of our methods when the residual contains multiple length scales. To achieve this, we compute right hand sides from manufactured solutions, i.e., $b = A(u^*)(u^*, p^*)^T$ when testing the linear Stokes solver and $b = F(u^*)u^*$ when testing the $(1,1)$-block solver. For that purpose, we create scalar-valued, spatially variable fields $s$ as the sum of a Fourier series with random coefficients and pointwise random component:

$$s(x, y, z) = \sum_{j, k=0}^{N} \sum_{(j, k) \neq (0,0)} (a^{j, k}, b^{j, k}, c^{j, k}, d^{j, k})^T \begin{pmatrix} \cos \omega_j x \cos \omega_k y \\ \cos \omega_j x \sin \omega_k y \\ \sin \omega_j x \cos \omega_k y \\ \sin \omega_j x \sin \omega_k y \end{pmatrix} |(j, k)|^{-\gamma} + e(x, y, z),$$
Fig. 4.1: Meshes and manufactured solution used for the test problems. Elements are colored according to their level of refinement. (a) Mesh $T_{xy}$, which contains nonconforming faces in $x$ and $y$ directions, for $\phi = 1$. (b) Mesh $T_{xy}$ for $\phi = 10$. (c) Mesh $T_{xyz}$, which contains nonconforming faces in all directions. (d) A manufactured solution $u^*$, constructed to contain variations with several length scales.

where $N = 10$, $\omega = 2\pi/L$, $\gamma = 3/2$, and $i = 1, 2, 3$. The coefficients $a_{j,k}^i$, $b_{j,k}^i$, $c_{j,k}^i$ and $d_{j,k}^i$ are randomly chosen from $[-1, 1]$, but the Fourier coefficients decay because of the $|\gamma| = (j^2 + k^2)^{-\gamma/2}$ term, making $\gamma$ a control of the smoothness of $s$. The extra term $e$ is a random value from $[-1/4, 1/4]$, added at each node of the discrete vector. To generate the vector field $u^*$ that is the manufactured solution of our test problems for the (1,1)-block solver, we generate a field $s$ for each component of $u^*$. The magnitude of such a velocity field is shown in fig. 4.1d. In all our tests, the norm we use to report the convergence of iterative solvers is the $\ell_2$-norm of the residual.

5. Preconditioning the (1,1)-block. The (1,1)-block $F$ occurring in the Stokes system is similar to the operator arising in linear elasticity. If we neglect boundary conditions, its nullspace is given by the rigid-body modes. To approximately invert $F$, we use algebraic multigrid (AMG) and, in particular, we use the smoothed aggregation multigrid (SA), which has theoretically proven convergence bounds [57]. SA multigrid uses Galerkin projections to create coarse approximation spaces, where the coarse space is embedded in the fine space by a prolongation matrix $P$. $P$ is constructed by first creating a projector $\hat{P}$ that projects coarse “aggregate” nodes onto disjoint sets of fine nodes, followed by creating $P$ from $\hat{P}$ by applying a local smoothing operation based on $F$, while ensuring that the near nullspace of $F$ (the nullspace in the absence of boundary conditions) is well-approximated in the coarse space [58]. To construct the prolongation and coarse matrices of the hierarchy, we use SA as implemented either by ML [24] or PETSc’s PCGAMG preconditioner. To build smoothers for each level, we use PETSc’s KSP and PC libraries. In this framework, we have three main design parameters, namely

1. the matrix $\hat{F}$ used to construct the multigrid hierarchy. As multigrid is only used as preconditioner and not as solver, $\hat{F}$ can be based on a lower-order
(a) Preconditioner for $F$, using $\tilde{F}$ constructed with $Q_1$ elements

| AMG threshold $\theta$ | 0 |
|------------------------|---|
| AMG smoother           | IC(0) |
| $\tilde{F}$ quadrature | Gauss-Lobatto |
| mesh                   | $T_{xy}$ (fig. 4.1a) |
| res. in smoother       | $F$ |

Fig. 4.2: (a) Values for key solver parameters, which we found to result in the best convergence behavior in our tests. The AMG threshold $\theta$ influences the coarse grid aggregation; together with the type of smoother, it is discussed in section 5.1; the quadrature for the low-order preconditioner, the influence of nonconforming element faces and choices for the residual computation for the smoother on the finest level are discussed in section 5.2. (b) Relative residual $\|r_j\|/\|r_0\|$ versus number of Krylov iterations for parameters in fig. 4.2a, for different element aspect ratios $\phi$.

(b) $\|r_j\|/\|r_0\|$: $k = 3$

$\phi = 1$
$\phi = 10$
$\phi = 100$

Krylov iteration $j$

Fig. 4.2: (a) Values for key solver parameters, which we found to result in the best convergence behavior in our tests. The AMG threshold $\theta$ influences the coarse grid aggregation; together with the type of smoother, it is discussed in section 5.1; the quadrature for the low-order preconditioner, the influence of nonconforming element faces and choices for the residual computation for the smoother on the finest level are discussed in section 5.2. (b) Relative residual $\|r_j\|/\|r_0\|$ versus number of Krylov iterations for parameters in fig. 4.2a, for different element aspect ratios $\phi$.

5.1. AMG connection thresholding and smoothers. The convergence of multigrid using pointwise smoothers such as Jacobi and Gauss-Seidel is known to degrade for anisotropic problems when isotropic coarsening is used [56, Chapter 4]. This slowdown can also be seen in fig. 5.1, where, for different element aspect ratios $\phi$, we show the convergence based on a symmetric Gauss-Seidel smoother for hierarchies created with aggregation threshold $\theta = 0$. This behavior occurs as for large $\phi$, vertical variations in the error are more decreased than horizontal variations.

In theory, a hierarchy can be constructed such that these undamped error components are well represented on the coarser meshes and can be corrected there. This technique of combining pointwise smoothing with selective hierarchy construction is known as semicoarsening. Although semicoarsening should be applicable for our problems, we have not had success with it. In SA, semicoarsening is accomplished by only aggregating strongly connected nodes, for which the corresponding matrix entries

\[ \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \]

\[ \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \]
Fig. 5.1: Relative residual $\|r_j\|/\|r_0\|$ versus number of Krylov iterations for the test problems described in section 4.3, with $k = 3$. Compared to the description in fig. 4.2a, the AMG preconditioner uses symmetric Gauss-Seidel as smoother instead of IC(0).

Fig. 5.1: Relative residual $\|r_j\|/\|r_0\|$ versus number of Krylov iterations for the test problems described in section 4.3, with $k = 3$. Compared to the description in fig. 4.2a, the AMG preconditioner uses symmetric Gauss-Seidel as smoother instead of IC(0).

\[ |F_{ij}|^2 \geq \theta |F_{ii}| |F_{jj}|. \]  
(5.1)

While this heuristic is generally good for scalar elliptic problems, for the vector system $F$ it is problematic. It is known that for fixed $\theta > 0$, sufficiently large $\phi$ will cause the standard SA algorithm to break down for operators whose nullspace contains the rigid-body modes [23, Section 5.3]. Even before this breakdown, we observed poor convergence rates with $\theta > 0$ and pointwise smoothers. A better heuristic for anisotropic problems than eq. (5.1), which is reported to result in good convergence for anisotropic elasticity problems, can be found in [37, Section 4]. However, this heuristic is not implemented in the SA libraries we have used. We therefore choose $\theta = 0$, which results in aggressively coarsened hierarchies, and design a smoother that is compatible with this choice. The choice $\theta = 0$ has the additional advantage that the sparsity structure of every matrix in the AMG hierarchy is independent of the values of $\mu'$ and $\beta$. This allows us to reuse the projection matrices for multiple Newton steps, reducing the cost of the preconditioner setup.

As for strongly anisotropic elements a pointwise smoother combined with aggressive coarsening is inefficient, we use an incomplete factorization smoother for these problems. Using local Fourier analysis, one can show that (geometric) multigrid smoothed by incomplete factorization can give $\phi$-independent convergence for scalar elliptic problems [56, Chapter 7]. Although we are using AMG on a nonuniform mesh for a non-scalar problem, we observe $\phi$-independent convergence for our test problems when using an incomplete Cholesky factorization with zero fill-in (IC(0)). For our anisotropic problems, the fast rate of convergence using IC(0) more than compensates for the additional setup time of computing the incomplete factorization, even when only a few Krylov iterations are required. In a parallel framework, we only compute the incomplete factorization for each process’s diagonal block of the distributed $\tilde{F}$ matrix, which amounts to a block-Jacobi smoother. This does not affect the rate of convergence if the elements of the mesh are distributed such that vertical columns of elements are not split between processes. If a column is divided between processes, the convergence can be worse as nodes in the same column are strongly coupled. To remedy this, one can either avoid partitioning across columns, or use IC(0) within
an overlapping additive Schwarz method ASM($j$), where $j$ is the degree of overlap between processes. The resulting smoothers can be categorized as ASM($j)$/IC($k$) smoothers, and we use $j = k = 0$ as default.

To build a stationary smoother from ASM($j$)/IC($k$), an estimate of the largest eigenvalue of the smoothed operator is required, either to calculate a single damping parameter or the coefficients of a Chebyshev polynomial. We numerically estimate the largest eigenvalue using an iterative method, which adds to the setup cost of the multigrid preconditioner. To avoid this setup cost, one can alternatively use a non-stationary smoother that does not require damping, such as a few iterations of a Krylov method. This is a good choice when only a small reduction in the residual is required, as for instance in the early iterations of an inexact Newton-Krylov method.

5.2. Construction of low-order approximation $\tilde{F}$. AMG requires assembled matrices for the construction of the coarse grid hierarchy. For matrices arising from high-order discretizations, this assembly requires significant memory and computation compared to low-order discretizations [5,29]. In three dimensions, the cost to construct an element matrix $F_i$ is $O(k^7)$ per element, or in terms of the number of nodes per element $O(n_k^{7/3})$. Instead of the true element matrix $F_i$ for element $K_i$, we construct an approximation $\tilde{F}_i$ that treats the variables associated with nodes of the $Q_k(\hat{K})$ finite element as variables for a $k \times k \times k$ grid of $Q_1(\hat{K})$ finite elements with corners at the high-order node locations. This matrix $\tilde{F}_i$ can be constructed in $O(n_k)$ steps.

Aspects of this lower-order preconditioning technique have been studied for simple problems: spectral equivalence between high-order and lower-order discretizations of the Laplacian has been proven in two dimensions [39] and is demonstrated numerically in three dimensions. The $(1,1)$-block $F$ differs from the operators used in previous studies in that it involves variable coefficients, high-aspect ratio elements, and non-conforming interfaces between elements. As we will show below, the interaction of these factors affects the stability and effectiveness of low-order preconditioning.

5.2.1. Influence of quadrature on low-order preconditioning. The effectiveness of the AMG preconditioner for $F$ also depends on the choice of the quadrature used to construct $\tilde{F}$. Let us denote by $\tilde{F}_G$ and $\tilde{F}_{GL}$ the low-order matrices computed with Gaussian and Gauss-Lobatto quadrature, respectively. We find that $\tilde{F}_{GL}$ leads to a better and more robust preconditioner, because the performance of the AMG preconditioner computed from $\tilde{F}_G$ is sensitive to the other AMG design parameters: fig. 5.2 shows that the convergence depends on both the ordering of the nodes when computing the incomplete factorization and on the aggregation strategy when creating the AMG hierarchy. The greater stability of Gauss-Lobatto quadrature comes from its diagonal-lumping behavior [16]. The factors of an incomplete factorization can become ill-conditioned when the original matrix is far from being diagonally dominant, particularly if there is no pivoting performed during the factorization [51, Chapter 10.5]. The diagonal-lumping inherent in Gauss-Lobatto quadrature increases the magnitude of diagonal entries relative to off-diagonal entries. An additional advantage of Gauss-Lobatto quadrature is that $\tilde{F}_{GL}$ is sparser than $\tilde{F}_G$, containing $\sim30\%$ fewer nonzero entries. This reduces the cost of the incomplete factorization and the hierarchy construction.

5.2.2. Influence of nonconforming meshes on low-order preconditioning. As discussed in section 3.2, nonconforming meshes require element restriction matrices $R_i$ that contain dense blocks for nonconforming interfaces. For an element with hanging nodes, the product $R_i^T \tilde{F}_i R_i$ can at best be computed in $O(k^3)$ time, or
in terms of $n_k$, $O(n_k^{5/3})$. One possibility for recovering $O(n_k)$ element assembly is to replace $R_i$ with a matrix $\tilde{R}_i$, in which the values of hanging nodes only depend on the nearest independent nodes, as illustrated in fig. 5.3. For the mesh $\mathcal{T}_{xy}$ (fig. 4.1a), which is the default mesh for our test problems, using these sparse-approximation restriction matrices $\{\tilde{R}_i\}$ does not affect the convergence. If we use the mesh $\mathcal{T}_{xyz}$ (fig. 4.1c), the errors incurred by these sparse restriction operators increase as $\phi$ increases. Using the true $\{R_i\}$ for $\mathcal{T}_{xyz}$ improves the convergence, but we still observe slight $\phi$-dependence (see fig. 5.4).
\[ \| r_j \| / \| r_0 \|: k = 3, \mathcal{T}_{xyz} \]

![Graph showing relative residual versus number of Krylov iterations](image)

**Fig. 5.4:** Relative residual \( \| r_j \| / \| r_0 \| \) versus number of Krylov iterations for the test problems described in section 4.3 for the mesh \( \mathcal{T}_{xyz} \) (fig. 4.1c), which has nonconforming faces in all directions. We compare two possibilities of handling nonconforming interfaces in the construction of the low-order preconditioner \( \tilde{F} \) at different element aspect ratios. The first possibility is to use \( \{ R_i \} \), the same hanging node restriction matrices as in the high-order discretization. An alternative is to use \( \{ \tilde{R}_i \} \), a piecewise linear approximation of \( \{ R_i \} \).

\[ (a) \; F u_j = \lambda_j \tilde{F} u_j, \mathcal{T}_{xy} \]

\[ (b) \; F u_j = \lambda_j \tilde{F} u_j, \mathcal{T}_{xyz} \]

![Graph comparing generalized eigenvalues](image)

**Fig. 5.5:** Comparison of the generalized eigenvalues \( \lambda \) that satisfy \( F u = \lambda \tilde{F} u \) for two meshes, \( \mathcal{T}_{xy} \) and \( \mathcal{T}_{xyz} \), and for \( \tilde{F} \) constructed with either the true restriction operator \( R_i \) or its sparse approximation \( \tilde{R}_i \). (a) The generalized eigenvalues for the mesh \( \mathcal{T}_{xy} \) are almost the same whether \( R_i \) or \( \tilde{R}_i \) is used. (b) The generalized eigenvalues for the mesh \( \mathcal{T}_{xyz} \) are sensitive to the use of \( \tilde{R}_i \).

To understand this behavior, note that in essence we are comparing two approximations to the high-order element matrix \( R_i^T F_i R_i \): one, \( R_i^T \tilde{F}_i R_i \), where the element matrix is replaced by a low-order approximation, and another, \( \tilde{R}_i^T \tilde{F}_i \tilde{R}_i \), where additionally the high-order interpolation of the nodal values at nonconforming faces is replaced by a low-order interpolation. The latter approximation only affects nonconforming faces; the fact that nonconforming interfaces are much larger in \( \mathcal{T}_{xyz} \) than in \( \mathcal{T}_{xy} \) for large \( \phi \) seems to explain the different convergence behavior for problems discretized on the two meshes.

To investigate the influence of nonconforming faces on the low-order preconditioner numerically, we consider the generalized eigenvalue equation \( F u = \lambda \tilde{F} u \) for \( \phi = 100 \) on the meshes \( \mathcal{T}_{xy} \) and \( \mathcal{T}_{xyz} \); the results are shown in fig. 5.5. For eigenvec-
Krylov iteration
\[ j \]
\[ \| r_j \| / \| r_0 \| : k = 3, \]
\[ \tilde{F} \text{-based AMG residual} \]

\[ 10^1 \]
\[ 10^{-7} \]
\[ 10^{-15} \]

Krylov iteration \( j \)

Fig. 5.6: Relative residual \( \| r_j \| / \| r_0 \| \) versus number of Krylov iterations for the test problems described in section 4.3, when the residual in the smoother on the finest level of the AMG hierarchy is computed with \( \tilde{F} \) instead of \( F \).

(a) \( \| r_j \| / \| r_0 \| : \phi = 100 \)

(b) \( \| r_j \| / \| r_0 \| : k = 3, \beta \equiv 10^{-10} \)

Fig. 5.7: Relative residual \( \| r_j \| / \| r_0 \| \) versus number of Krylov iterations for the test problems described in section 4.3, for variations on the parameters given in fig. 4.2a. (a) Varying orders of approximation are compared on mesh \( T_{xy} \) for \( \phi = 100 \). (section 5.3). (b) The parameters in fig. 4.2a are tested on problems with small Robin coefficient \( \beta \) (section 5.4).

tors \( u \) with eigenvalues far from 1, \( \tilde{F} \) is a poor approximation of \( F \). For the mesh \( T_{xy} \), the errors incurred by the sparse restriction matrices \( \{ \tilde{R}_i \} \) due to the nonconforming interfaces are small. For \( T_{xyz} \), errors incurred by \( \{ \tilde{R}_i \} \) are more significant, and inspection of the extremal eigenvalues shows that they are associated with vectors \( u \) for which \( F u \) is large only on nonconforming interfaces that are normal to the z-axis.

5.2.3. Residual computation for smoother on the finest mesh. When the matrix \( \tilde{F} \) used to generate a multigrid hierarchy differs from the true matrix \( F \), one has two possibilities for the residual computation in the smoothing step on the finest mesh. Namely, one can use the high-order discretized operator, i.e., \( r = b - F x \), or its low-order approximation, i.e., \( r = b - \tilde{F} x \). As shown in fig. 5.6, we observe faster convergence using \( F \) rather than \( \tilde{F} \). Thus, in the following, we use the high-order operator for the residual computation in the smoother on the finest mesh.

5.3. Convergence for different orders \( k \). In fig. 5.7a we show the convergence of our solver for polynomial orders \( k=3, 6, \) and 9, where we use the parameters from
As can be seen, the iteration number is independent of the polynomial order. To illustrate that the preconditioner is also $k$-independent with respect to computational work, note that the operator complexities of the AMG hierarchies with respect to $\tilde{F}$ (i.e., the sum of nonzero matrix entries in all operators in the AMG hierarchy divided by the number of nonzero entries in $\tilde{F}$) reported by PETSc are 1.22, 1.26, and 1.27 for $k = 3, 6$ and 9, respectively, and that the average numbers of nonzeros per row in $\tilde{F}$ are 86.83, 82.33, and 82.36. This demonstrates that the cost to construct $\tilde{F}$ and the coarse hierarchy are proportional to the problem size, but independent of $k$. The same can be said of the cost of computing the incomplete factorization smoother on each level as we do not allow fill-in. Thus, the overall computational complexity for preconditioning $F$ is independent of $k$.

5.4. Solver behavior as $\beta \to 0$. The behavior of the solver for $\beta = 0$ (or very small $\beta$) is important for our target application, because this condition is seen under ice streams. For $\beta = 0$, the basal Robin boundary condition degenerates into a homogeneous Neumann condition and the matrix $F$ is singular as any constant vector field $\mathbf{u}$ tangential to the basal boundary is in the null space of $F$. For varying values of $\phi$, we have tested our solver for $\mu = 1$ and $\beta \to 0$. We find that for very small $\beta$, the convergence of our solver is $\phi$-dependent. It appears, however, that for a fixed $\phi$ the convergence rate is bounded away from zero as $\beta \to 0$, so that the same convergence rate is seen for all $\beta$ below a certain threshold. In fig. 5.7b we show the convergence rates for $\beta = 10^{-10}$, but the convergence behavior is identical for $\beta = 10^{-8}$ and $\beta = 10^{-9}$. Our attempts to eliminate the $\phi$-dependence through modifications to our smoothing operation, such as reordering the nodes to make the incomplete factorization closer to a true plane smoother, were unsuccessful.

6. Preconditioning the Schur complement of the (1,1)-block. In this section, we analyze how different approximations of the Schur complement $S$ of the (1,1)-block and different choices of the basis for the pressure space $M$ affect the convergence rate for Stokes linear problems preconditioned as in eq. (4.2). For all our tests, the preconditioner for the (1,1)-block $F$ is a single multigrid V-cycle with the parameter choices described above. We test the effectiveness of our preconditioner on Stokes problems whose setup is discussed in section 4.3. As discussed in section 3.2, the discrete inf-sup constant for the $Q_k \times P_{k-1}$ mixed element is $\phi$-dependent. In fig. 6.1a, we demonstrate this $\phi$-dependence numerically. Because of this instability we prefer the $Q_k \times Q_k - 2$ mixed element for anisotropic problems.

For constant viscosity $\mu > 0$, $S$ is known to be spectrally equivalent to the scaled pressure mass matrix $-\mu^{-1}M$. Because of this equivalence, a common choice is to approximate $S$ by $-\tilde{M}(\mu^{-1})$, which is a diagonally-lumped approximation to the $\mu^{-1}$-weighted mass matrix. This Schur complement approximation does not take into account the anisotropic part of the 4th-order tensor $\mu'(\mathbf{u})$ defined in (2.7). As an alternative to the weighted mass matrix, the least-squares commutator, also known as the BFBt preconditioner, has proven to be a good Schur complement approximation, in particular in the presence of strongly varying coefficients [18, 42].

We have implemented both of these Schur complement preconditioners, and have found $-\tilde{M}(|\mu|^{-1})$ to be the most efficient for the problems targeted in this paper. One reason for this good performance is that we choose a basis for the discontinuous finite element pressure space that nearly diagonalizes the mass matrix $M$ for the pressure

\[2\text{For realistic ice sheet geometries, this is not the case because of the varying basal topography: a nonzero constant flow field } \mathbf{u} \text{ cannot be tangential to the boundary everywhere.}\]
space. As a consequence, the effect of replacing $-M(|\mu|^{-1})$ with its mass-lumped counterpart $-\tilde{M}(|\mu|^{-1})$ is minimal. Such a basis for $Q_{k-2}(\tilde{K})$ is given by a Lagrange basis for the tensor-product Gauss nodes. For a mapped element $K_i$, the mass matrix remains nearly diagonal provided the mapping is moderately nonlinear. To illustrate the effect of the choice of the basis, in fig. 6.1b we compare the convergence when using this Lagrange basis for the Gauss nodes to the convergence when using the Lagrange basis for Gauss-Lobatto nodes, which is more commonly used as a basis for tensor-polynomial finite elements.

In fig. 6.2a, we observe that the preconditioner for the Stokes system $A$ results in convergence independent of the aspect ratio $\phi$; however, as shown in fig. 6.2b, we find a dependence of the convergence on $k$. Note that this differs from our findings obtained for the (1,1)-block shown in fig. 5.7a, which shows independence of $k$ for solving systems with $F$. The $O(k^{-1})$ decay of the lower bound for the discrete inf-sup constant for $Q_k \times Q_{k-2}$ suggests that this $k$-dependence cannot be avoided.

7. Nonlinear ice stream problems with smooth and rough beds. To test the nonlinear solver for eq. (2.5), we adapt a model problem from [12]. As in section 4.3, the domain is a cutout of an infinite slab that is periodic in the horizontal dimensions, but the pitch of the domain relative to the direction of gravity is $0.5^\circ$, so that a flow is induced. The Robin coefficient field $\beta$ is shown in fig. 7.1a. Although $\beta$ varies smoothly, the nonlinearity of the rheology causes the velocity $u$ to develop a narrow region of fast flow similar to an ice stream, as shown in fig. 7.1b. The constants in the constitutive relationship eq. (2.2) are $n = 3$ and $B(T) \equiv 2.15 \times 10^5 \text{ Pa a}^{1/3}$, which equals $A^{-1/3}$, where $A = 10^{-16} \text{ Pa}^{-3} \text{ a}^{-1}$, which is taken from [48]. We use $\varepsilon = 1 \times 10^{-6} \text{ a}^{-2}$, which has a negligible effect for stresses of $10^5 \text{ Pa}$ or greater, following the recommendation in [31, Chapter 2]. The periodic domain is 400 km $\times$ 400 km $\times$ 1 km, and we again use $\mathcal{T}_{xy}$ as our mesh so that the elements are stretched to $\phi = 100$.\[\]
In fig. 7.2, we show the convergence behavior of the inexact Newton method for $k = 3, 4,$ and $5,$ and compare with the convergence of an inexact Picard method for $k = 3$. As can be seen, the inexact Newton method converges faster than Picard’s method, both in terms of the number of nonlinear iterations and in terms of the total number of Krylov iterations. As each nonlinear iteration requires a preconditioner setup (or update), the superiority of Newton’s method compared to the Picard method is even more pronounced if we consider time-to-solution rather than the number of Krylov iterations. Although not included in the convergence plot, we did test Picard’s method with tighter tolerances on the linear solves, but found that this did not improve the convergence rate in terms of total Krylov iterations.

We next test our method on the same problem, but with the Robin coefficient field reduced to 1% of the previous field, and with rough bed geometries instead of the flat slab used in the previous test. We generate topographies using random coefficients in a truncated Fourier series, as in eq. (4.3). By changing the exponent $\gamma$, which controls the decay of the Fourier coefficients, we are able to control how rough the generated topography is. In fig. 7.3, we show three topographies, generated with ten Fourier modes and $\gamma = 1.5, 1.1,$ and $1.0$. In fig. 7.4 we show the convergence behavior of our
method on domains with these bed topographies.

We observe that the solver is not robust for very rough topographies: in later Newton iterations, the linear solver may either converge slowly or fail to converge. We attribute this behavior to the incomplete factorization used as a smoother for the (1,1)-block preconditioner. If we substitute a Gauss-Seidel smoother, all problems converge robustly, albeit with a poor convergence rate due to the high aspect ratio of the elements. The instability of the incomplete factorization smoother seems related to the variability of the effective viscosity $\mu'$ in the Newton linearization. When applied to linear PDEs with constant, isotropic viscosity, the incomplete factorization smoother converges rapidly. Similar observations are made in [6], where the authors apply similar numerical methods to the hydrostatic approximation of the Stokes equations.

8. Antarctic ice sheet problem. We now demonstrate the performance of our inexact Newton-Krylov solver for the simulation of the dynamics of the Antarctic ice sheet. We only consider the grounded portion of the Antarctic ice sheet, i.e., we neglect ice shelves, the extension of the sheet onto the surface of the ocean. Below, we detail how real data is used to define the computational domain and describe the mesh generation. In section 8.2, we study the performance of the solver based on incomplete factorization preconditioning on a regularized Antarctica geometry, which avoids extreme thickness variations. Finally, in section 8.3, we solve the Antarctic
Fig. 7.3: Bed topographies with increasing roughness for the ice stream model problem. The vertical scale is exaggerated by a factor of 100. The maximum variation in the bed topography in (7.3c) is about half of the ice thickness.

\[ \frac{\|r_j\|}{\|r_0\|} : \phi = 100, k = 3, \text{ISMIP-C, stream variant, CG(2)/ASM(0)/IC(0), three bed topographies} \]

Fig. 7.4: Convergence of the inexact Newton solver for the ice stream model problem as described in fig. 7.1, but with the rough bed topographies shown in fig. 7.3.

ice problem on the real geometry; here, we limit the amount of element anisotropy by using proper mesh refinement, and can thus use Gauss-Seidel smoothing in the preconditioner.

8.1. Problem Description. For the following simulations, we define the ice density to be \( \rho = 917 \text{ kg/m}^3 \), we define the pre-exponential in Glen’s power law ((2.2)) to be \( B(T) = 4.1 \times 10^5 \text{ Pa a}^{1/3} \), and the regularizing constant that prevents infinite effective viscosity to be \( \epsilon = 9.95 \times 10^{-6} \text{ a}^{-2} \). The assumed sliding coefficient field \( \beta \) can be seen in fig. 8.1b.

The geometric description of the ice sheet is constructed from the ALBMAP dataset [40]. From the ice thickness data, given on a latitude-longitude grid, we obtain a polygon describing the lateral boundaries of the ice sheet. We create a mesh of triangles from this polygon using the Delaunay-based Triangle code [54]. To keep the aspect ratios of our three-dimensional elements reasonable, we restrict the maximum size of triangles. Each triangle is then split into three quadrilaterals, and each quadrilateral is extruded into a hexahedron. Elevation values in the ALBMAP dataset are given relative to the EIGEN-GL04C geoid [20]: we convert these values to
Fig. 8.1: Antarctic ice sheet problem: (a) A detail of the fine mesh, showing mesh refinement at the boundaries of the mesh, i.e., the grounding line. (b) The coefficient field used for the Robin-type boundary condition at the base of the ice sheet.

elevations relative to the WGS84 ellipsoid [1] using the software library GeographicLib [38], and then map the resulting (latitude, longitude, elevation) geodetic coordinates into Cartesian coordinates. After mesh smoothing to improve the element quality, this results in the coarse hexahedral mesh containing ~100,000 elements, from which we construct finer meshes using hierarchical refinement. We require that the elements of our mesh have a footprint smaller than $(2.5 \text{ km})^2$ at this grounding line. If this is not already the case in the coarse mesh, we use refinement to enforce this resolution, as shown in fig. 8.1a.

8.2. Performance of IC(0)-based smoothing on regularized geometry.
We first demonstrate our solver on the Antarctic geometry using the (1,1)-block preconditioner based on an incomplete factorization smoother (see section 5). Because of the instabilities encountered in section 7 for rough beds, we perform this test on a regularized geometry, in which we artificially set the minimum thickness of the ice sheet to 400 m. This artificial thickness is enforced by modifying the basal topography wherever necessary, while leaving the surface topography unchanged.

We first test our solver for different polynomial orders on a mesh consisting of a fixed number of elements. Using the mesh described above, we discretize eq. (2.5) using $Q_3 \times Q_{k-2}$ elements for $k = 3, 4, 5$, resulting in problems with 22M, 52M, and 101M degrees of freedom. The experiments were conducted on TACC’s Stampede supercomputer, with each MPI process assigned to one Sandy Bridge Xeon core. Each discretization is distributed across 512 MPI processes. The parallel mesh partitioning used for these tests does not guarantee that each column of elements remains on the same processor. As discussed in section 5.1, the performance of ASM(0)/IC(0) can degrade if nodes along columns, which are strongly coupled, are separated on different processors. As a remedy, we use ASM(1)/IC(0), combined with two iterations of CG as a smoother.

Nonlinear convergence for different orders $k$. The convergence of our inexact Newton-Krylov solver is shown in fig. 8.2. Compared to the model problem in section 7, the inexact Newton method requires more iterations before it reaches the asymptotic convergence regime. This is likely due to a combination of a greater degree of nonlinearity in the problem and the fact that the overlap in ASM(1) does not perfectly compensate for the splitting of columns of elements across processors.
\[ \| r_j \| / \| r_0 \| : k = 3, \text{ Antarctica, CG}(2)/\text{ASM}(1)/\text{IC}(0) \]

Fig. 8.2: Antarctic ice sheet problem: convergence of the incomplete Newton-Krylov solver for the regularized geometry. The mesh is distributed over 512 processes. The bracketed numbers are the total number of nonlinear iterations.

Table 8.1: Antarctic ice sheet problem: strong scaling of the nonlinear solver for \( k = 3 \) for the regularized geometry. For different numbers of processes \( P \), we report the total time (in seconds) to solve the nonlinear problem to a relative tolerance of \( 10^{-12} \) in the \( \ell^2 \)-norm from a zero initial guess, the parallel efficiency (eff.) as well as the number of Newton iterations (#N) and overall Krylov iterations (#K) performed during the solution. We also report the average time and efficiency for a single matrix-vector product (matvec), a preconditioner application (apply), and a preconditioner setup (setup).

| \( P \) | solve | eff. | #N | #K | matvec | eff. | apply | eff. | setup | eff. |
|-------|-------|------|----|----|--------|------|-------|------|-------|------|
| 128   | 156.0 | 1.00 | 21 | 114 | 0.0499 | 1.00 | 0.549 | 1.00 | 3.51  | 1.00 |
| 256   | 112.0 | 0.69 | 18 | 113 | 0.0255 | 0.98 | 0.290 | 0.95 | 2.56  | 0.69 |
| 512   | 79.7  | 0.49 | 22 | 124 | 0.0131 | 0.95 | 0.157 | 0.87 | 1.76  | 0.50 |
| 1024  | 38.7  | 0.35 | 21 | 127 | 0.00679| 0.92 | 0.0925| 0.74 | 0.72  | 0.61 |
| 2048  | 27.5  | 0.35 | 20 | 157 | 0.00342| 0.91 | 0.0563| 0.61 | 0.55  | 0.40 |

Strong parallel scalability. We next test the strong parallel scalability of our solver on the \( Q_3 \times Q_1 \) discretization above and report the results in table 8.1. The table also includes timings for each of the main components of the nonlinear solver: the matrix-vector product, the preconditioner application, and the preconditioner reconstruction. The scalability of the nonlinear solver in terms of outer Newton iterations is good, but the convergence degrades when more than 1024 processes are used. This is due to the splitting of strongly coupled ice columns between processes, which happens more frequently as the number of processes increases and thus reduces the effectiveness of the smoother.

The time to apply the preconditioner is almost entirely spent in applying the AMG V-cycle to the (1,1)-block of the Stokes operator. This step requires more communication than the matrix-vector product, both to project and restrict vectors in the hierarchy and to apply the smoothers. The CG iterations require global synchronization to perform the reductions involved, which is not true of a stationary smoother.

23
Table 8.2: Antarctic ice sheet problem: weak scaling of the linear solver for \( k = 3 \) for the regularized geometry. The smoother for the \((1,1)\)-block AMG V-cycle uses ASM(0): all other solver parameters are the same as those used in table 8.1. We report the convergence of the linear Stokes problem with constant \( \mu \), which we solve to a relative tolerance of \( 10^{-15} \) in the \( \ell^2 \) norm.

| \( P \) | \( N_{\text{dof}} \) | solve eff. | apply eff. | setup eff. | solve | apply |
|-------|-----------------|------------|------------|------------|-------|-------|
| 128   | 11.8M           | 14.6       | 0.246      | 0.0223     | 1.00  | 3.78  |
| 1024  | 82.6M           | 15.9       | 0.245      | 0.0217     | 1.03  | 4.32  |
| 8192  | 615M            | 28.7       | 0.258      | 0.0234     | 0.95  | 11.7  |

Because of the ASM(1) preconditioning of the CG iterations, the work performed in each smoother application does not scale as \( 1/P \): the overlap that augments each process’s local block of \( \tilde{F} \) is larger relative to the size of that diagonal block as more processes are used.

It is well known that the setup phase in parallel implementations of algebraic multigrid requires significant communication to properly aggregate degrees of freedom across processor boundaries. As a consequence, the setup is often the least scalable component of the solver [7, 11], as can also be seen in table 8.1. Here, the first preconditioner construction, which includes the construction of the AMG projection operators, is averaged in with the time to reconstruct the preconditioner for subsequent iterations, which reuses these projection operators to recompute the Galerkin coarse grid matrices. In all cases, the contribution of the initial AMG setup to the average is relatively small, less than 10% of all preconditioner construction time.

Note that the number of inexact Newton iterations in table 8.1 remains roughly constant at about 20. Many of these iterations occur before the region of asymptotic convergence is reached. While these initial Newton iterations require few Krylov iterations, the cost of updating the preconditioner between Newton steps contributes to the overall solution time. Recent work [6] on solving the hydrostatic approximation to the Stokes equations for ice sheet dynamics has demonstrated the effectiveness of grid continuation in obtaining an initial guess that is near the region of asymptotic convergence of Newton’s method. The hierarchical mesh refinement we use lends itself naturally to grid continuation, so this approach could improve the efficiency of our nonlinear solver.

Weak parallel scalability. We next study the weak scalability of our linear solver on \( Q_3 \times Q_1 \) discretizations of the Antarctic ice sheet problem. We start with the same mesh of Antarctica used above, which has \( \sim 781 \) elements per process when partitioned between 128 processes, and isotropically refine all elements to construct the next finer mesh, larger by a factor of 8, and then solve using 8 times as many processes. Because of the uniform refinement in these meshes, we are able to avoid splitting columns of ice between processes, and thus use an ASM(0) smoother within the AMG V-cycle for the \((1,1)\)-block. We study only the weak scalability of the linear solver, because despite the regularized geometry, we encountered similar problems with the incomplete factorization as for the nonlinear stream problem in section 7. While the convergence of the linear Krylov solver did not stall as in section 7, the variability of the effective viscosity and of the Robin coefficient combined with the changes in the bedrock topography resulted in suboptimal weak scalability of the full nonlinear problem.

In table 8.2, we report the weak scalability of the system arising in the first Newton
linearization. As we use a zero initial guess, the effective viscosity for this Newton step is constant. The table shows that on the Antarctic mesh our linear solver is capable of almost $h$-independent convergence. Because we study a single linear solve, the cost of the AMG hierarchy construction is not amortized over multiple linear solves, so the preconditioner setup is the bottleneck in parallel scalability. PETSc’s PCGAMG implementation of SA-AMG, which is used for the (1,1)-block uses sparse matrix-matrix multiplications—to compute the square of the node-adjacency graph, and to compute the coarse matrix through Galerkin projection—at each level of the hierarchy construction. This likely causes most of the loss of parallel efficiency for the $P = 8192$ case.

8.3. Performance of SSOR-based smoothing on the true Antarctica geometry. In [33], we have used the nonlinear and linear solver framework developed in this paper to infer the Robin coefficient field $\beta$ in the Stokes boundary value problem such that the velocity fields of the solution closely matches satellite observations of the Antarctic ice sheet’s surface (see fig. 8.3). This inference uses methods of adjoint-based PDE-constrained optimization to find an optimal $\beta$, which requires the solution of the Stokes problem described in this work, as well as linearized adjoint problems, whose operators are similar to the linearized Stokes operator in (2.6).

Within the optimization procedure, it is necessary that the convergence of the nonlinear and linear solvers is particularly robust. To avoid instabilities with the incomplete factorization-based smoother due to the strongly varying thickness of the Antarctic ice sheet, we thus replace the incomplete factorization smoother in the (1,1)-block V-cycle with a symmetric Gauss-Seidel (SSOR) smoother. Although SSOR converges slower for anisotropic elements, it is more robust to geometric variability, and do not have to regularize the topography as described above (the extent of the ice sheet that we include in the model domain has a minimum thickness of 50 m). To avoid splitting columns of elements, as frequently happens with octree-based mesh management, we used a hybrid AMR scheme that uses quadtrees to distribute columns of hexahedra, and then refines the columns independently in the vertical direction. This allows us to use ASM(0)/SSOR instead of ASM(1)/SSOR, and thus to reduce the amount of work required in the preconditioner setup and application.

In table 8.3, we reproduce some of the performance scalings from [33], which were conducted on Oak Ridge National Laboratory’s Titan for a similar Robin coefficient field as shown in fig. 8.1b. The total runtime is separated in setup (matrix construction and preconditioner setup) and solve time (FGMRES orthogonalization, matrix-vector products, and preconditioner applications). For these two problem sizes, we see good strong scalability in the solve time and near $h$-independence in the number of Newton iterations. Because of the limitation of the SSOR smoother, however, we see $h$-dependence in the convergence rate of the Krylov iterations.

9. Conclusions. Several issues related to high-order, adaptive mesh discretizations and solvers for the simulation of nonlinear Stokes flow in three-dimensional anisotropic domains are addressed in this work. Our main target problem has been the nonlinear Stokes boundary value problem arising in ice sheet dynamics. We presented an efficient solver for the linearized Stokes equations, with particular emphasis on the design of algebraic multigrid solvers for high-order discretizations, anisotropic domains, and hanging nodes. Our solver handles these complicating factors well, but some issues remain. For meshes with large element aspect ratios, the combination of highly variable geometry and variable effective viscosity can cause the convergence to stall. For moderately rough bed topographies, using incomplete factorization-based
Fig. 8.3: Antarctic ice sheet problem: magnitude of the surface velocity field optimized to match satellite observations. The inversion procedure to infer the basal coefficient is described in [33].

Table 8.3: Antarctic ice sheet problem: strong scaling results for two different problem sizes on Antarctic ice sheet on ORNL’s Titan. Shown are the number of overall degrees of freedom $N_{dof}$, the number of processes $P$, the number of Newton and overall Krylov iterations $\#N$ and $\#K$, respectively, and the overall time (in seconds) and parallel efficiency for the solve and the preconditioner setup. This data is reproduced from [33].

| $N_{dof}$ | $P$ | $\#N$ | $\#K$ | solve eff. | setup eff. |
|-----------|-----|-------|-------|------------|------------|
| 38M       | 128 | 8     | 149   | 504.8      | 1.00       |
|           | 256 | 8     | 153   | 259.6      | 0.97       |
|           | 512 | 8     | 157   | 134.3      | 0.94       |
|           | 1024| 8     | 147   | 70.1       | 0.90       |
| 268M      | 1024| 9     | 240   | 796.6      | 1.00       |
|           | 2048| 9     | 245   | 414.3      | 0.96       |
|           | 8192| 9     | 243   | 130.7      | 0.76       |

smoothing for the (1,1)-block yields efficient and fast convergence. For very rough bed topographies, the incomplete factorization-based additive Schwarz smoothers can become inefficient; in this case we use moderately anisotropic meshes, for which Gauss-Seidel smoothing can be used. These smoothers can be applied in parallel using a block Jacobi scheme. This scheme becomes less effective when columns of elements are split between processes, due to the strong coupling of elements in vertical direction. In this case, additive Schwarz overlap helps to maintain fast convergence.

Appendix A. Well-posedness off eq. (2.5).

The main difference between eq. (2.5) and the variational form in [35] is the boundary integral $\int_{\Gamma_R} \beta T_\nu \nabla u \cdot \nabla u \, ds$: for $u \in W^{1,r}(\Omega)$, this form is not meaningful because the trace of $u$ is not necessarily in $L^2(\Gamma_R)$. We compensate for this by bringing the boundary integral into the definition of $V$. Let $[C^\infty(\Omega)]^3_{0}$ be the space of smooth vector-valued functions in $\Omega$ that satisfy the homogeneous Dirichlet part of the boundary conditions and let $r = 1 + \frac{1}{n}$. Under the above assumptions, the functional

$$
\mathcal{J}(u) = \left\{ \int_\Omega |\nabla u|^r \, dx \right\}^{1/r} + \left\{ \int_{\Gamma_R} \beta |T_\nu u|^2 \, ds \right\}^{1/2}
$$

defines a seminorm on $[C^\infty(\Omega)]^3_{0}$. We assume a problem setup in which $\mathcal{J}(u)$ also
defines a norm, which amounts to requiring that if \( u \in [C^\infty(\Omega)]^3 \) is a rigid-body motion that satisfies the Dirichlet conditions, then \( \int_{\Gamma_N} \beta |T_n u|^2 \, ds > 0 \). We define \( \mathcal{V} \) to be the closure of \([C^\infty(\Omega)]^3\) in this norm.

Following the same steps as in [35], one can show that the minimization problem

\[
\min_u \frac{1}{2} \int_{\Omega} \mu(u) \nabla u : \nabla u \, dx + \frac{1}{2} \int_{\Gamma_N} T_n u \cdot T_n u \, ds - f(u) \quad (A.2)
\]

is well-posed in \( \mathcal{V}_{\text{div}} \), the subspace of \( \mathcal{V} \) containing only the divergence-free functions. To prove that eq. (2.5) is well-posed, we additionally need to choose a space \( \mathcal{M} \) for which we can prove the inf-sup condition

\[
\inf_{q \in \mathcal{M}} \sup_{u \in \mathcal{V}} \frac{\int_{\Omega} q \nabla \cdot u \, dx}{\|q\|_\mathcal{M} \|u\|_{\mathcal{V}}} \geq \gamma > 0. \quad (A.3)
\]

This inequality still holds for \( \mathcal{M} = \mathcal{L}^{s'}(\Omega) \). To see this, consider the subspace \( \hat{\mathcal{V}} = \{ u \in \mathcal{V} : u|_{\Gamma_N} = 0 \} \); for \( u \in \hat{\mathcal{V}} \), \( \|u\| = \|u\|_{\mathcal{V}}^{\mathcal{W}^{s',r}(\Omega)} \), and so

\[
\inf_{q \in \mathcal{L}^{s'}(\Omega)} \sup_{u \in \hat{\mathcal{V}}} \frac{\int_{\Omega} q \nabla \cdot u \, dx}{\|q\|_{\mathcal{L}^{s'}(\Omega)} \|u\|_{\mathcal{V}}^{\mathcal{W}^{s',r}(\Omega)}}. \quad (A.4)
\]

As long as \( \Gamma_N \neq \emptyset \), the term on the right is bounded from below as a particular case of the inf-sup condition in [35]. Thus, (2.5) is well posed and has a unique solution.

REFERENCES

[1] Department of Defense World Geodetic System 1984: Its definition and relationships with local geodetic systems, Tech. Rep. NGA TR8350.2, U.S. National Geospatial-Intelligence Agency, 1984. Revised in 2004.

[2] M. Ainsworth and P. Coggins, The stability of mixed hp-finite element methods for Stokes flow on high aspect ratio elements, SIAM Journal on Numerical Analysis, 38 (2000), pp. 1721–1761.

[3] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11 - Revision 3.3, Argonne National Laboratory, 2012.

[4] H. Blatter, Velocity and stress fields in grounded glaciers: a simple algorithm for including deviatoric stress gradients, Journal of Glaciology, 41 (1995), pp. 333–344.

[5] J. Brown, Efficient nonlinear solvers for nodal high-order finite elements in 3D, Journal of Scientific Computing, 45 (2010), pp. 48–63.

[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.

[7] C. Burstedde, O. Ghattas, G. Stadler, T. Tu, and L. C. Wilcox, Parallel scalable adjoint-based adaptive solution for variable-viscosity Stokes flows, Computer Methods in Applied Mechanics and Engineering, 198 (2009), pp. 1691–1700.

[8] C. Burstedde, L. C. Wilcox, and O. Ghattas, p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees, SIAM Journal on Scientific Computing, 33 (2011), pp. 1103–1133.

[9] M. Cai, A. Nonaka, J. Bell, B. Griffith, and A. Donev, Efficient variable-coefficient finite-volume Stokes solvers, arXiv preprint arXiv:1308.4605, (2013).

[10] C. Canuto, P. Gervasio, and A. Quarteroni, Finite-element preconditioning of G-N1 spectral methods, SIAM Journal on Scientific Computing, 31 (2010), pp. 4422–4451.

[11] E. Chow, R. D. Falgout, J. J. Hu, R. S. Tuminaro, and U. M. Yang, A survey of parallelization techniques for multigrid solvers, in Parallel Processing for Scientific Computing, M. A. Heroux, P. Raghavan, and H. D. Simon, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2006, pp. 179–201.

[12] S. L. Cornford, D. F. Martin, D. T. Graves, D. F. Ranken, A. M. Le Brocq, R. M. Gladstone, A. J. Payne, E. G. Ng, and W. H. Lipscomb, Adaptive mesh, finite volume modeling of marine ice sheets, Journal of Computational Physics, 232 (2013), pp. 529–549.
[13] H. Damanik, J. Hron, A. Ouazzi, and S. Turek, A monolithic fem-multigrid solver for non-isothermal incompressible flow on general meshes, Journal of Computational Physics, 228 (2009), pp. 3869–3881.

[14] M. Deville and E. Mund, Finite-element preconditioning for pseudospectral solutions of elliptic problems, SIAM Journal on Scientific and Statistical Computing, 11 (1990), pp. 311–342.

[15] M. O. Deville, P. F. Fischer, and E. H. Mund, High-Order Methods for Incompressible Fluid Flow, vol. 9 of Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2002.

[16] M. Duruflé, P. Ghob, and P. Joly, Influence of Gauss and Gauss-Lobatto quadrature rules on the accuracy of a quadrilateral finite element method in the time domain, Numerical methods for partial differential equations, 25 (2009), pp. 526–551.

[17] S. C. Eisenstat and H. F. Walker, Choosing the forcing terms in an inexact Newton method, SIAM Journal on Scientific Computing, 17 (1996), pp. 16–32.

[18] H. Elman, V. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro, Block preconditioners based on approximate commutators, SIAM Journal on Scientific Computing, 27 (2006), pp. 1651–1668.

[19] H. C. Elman, D. J. Silvester, and A. J. Wathen, Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics, Oxford University Press, Oxford, 2005.

[20] C. Fö rste, R. Schmidt, R. Stubenvoll, F. Flechtner, U. Meyer, R. König, H. Neumayer, R. Biancale, J.-M. Lemoine, S. Bruinsma, et al., The Geoforschungszentrum Potsdam/groupe de Recherche de Geodesie Spatiale satellite-only and combined gravity field models: EIGEN-GL04S1 and EIGEN-GL04C, Journal of Geodesy, 82 (2008), pp. 331–346.

[21] M. Furuichi, D. A. May, and P. J. Tackley, Development of a Stokes flow solver robust to large viscosity jumps using a Schur complement approach with mixed precision arithmetic, Journal of Computational Physics, 230 (2011), pp. 8835–8851.

[22] O. Gagliardini and T. Zwinger, The ISMIP-HOM benchmark experiments performed using the finite-element code Elmer, The Cryosphere, 2 (2008), pp. 67–76.

[23] M. W. Gee, J. J. Hu, and R. S. Tuminaro, A new smoothed aggregation multigrid method for anisotropic problems, Numer. Linear Algebra Appl., 16 (2009), pp. 19–37.

[24] M. W. Gee, C. M. Siebert, J. J. Hu, R. S. Tuminaro, and M. G. Sala, ML 5.0 smoothed aggregation user's guide, Tech. Rep. SAND2006-2649, Sandia National Laboratories, 2006.

[25] T. Geelen, M. U. Rehman, S. P. MacLachlan, G. Segal, C. Vuik, A. P. Van den Berg, and W. Spakman, Scalable robust solvers for unstructured FE geodynamic modeling applications: Solving the Stokes equation for models with large localized viscosity contrasts, Geochimica Geophysica Geosystems, 10 (2009), p. Q09002.

[26] R. Glowinski and J. Xu, Numerical Methods for Non-Newtonian Fluids: Special Volume, vol. 16 of Handbook of numerical analysis, North-Holland, 2011.

[27] P. P. Grinevich and M. A. Olshanskii, An iterative method for the Stokes-type problem with variable viscosity, SIAM Journal on Scientific Computing, 31 (2009), pp. 3959–3978.

[28] V. Heuveline and F. Schieweck, On the inf-sup condition for higher order mixed FEM on meshes with hanging nodes, ESAIM: Mathematical Modelling and Numerical Analysis, 41 (2007), pp. 1–20.

[29] J. Heys, T. Mantuffel, S. McCormick, and L. Olson, Algebraic multigrid for higher-order finite elements, Journal of Computational Physics, 204 (2005), pp. 520–532.

[30] R. C. A. Hindmarsh, A numerical comparison of approximations to the Stokes equations used in ice sheet and glacier modeling, Journal of Geophysical Research, 109 (2004), p. F01012.

[31] K. Hutter, Theoretical Glaciology, Mathematical Approaches to Geophysics, D. Reidel Publishing Company, 1983.

[32] T. Isaac, C. Burstedde, and O. Ghittas, Low-cost parallel algorithms for 2:1 octree balance, in Proceedings of the 26th IEEE International Parallel & Distributed Processing Symposium, IEEE, 2012.

[33] T. Isaac, N. Petra, G. Stadler, and O. Ghittas, From data to prediction with quantified uncertainties: Scalable parallel algorithms and applications to the dynamics of the Antarctic ice sheet, preprint, 2014.

[34] A. Janka, Smoothed aggregation multigrid for a Stokes problem, Computing and Visualization in Science, 11 (2008), pp. 169–180.

[35] G. Jouvet and J. Rappaz, Analysis and finite element approximation of a nonlinear stationary Stokes problem arising in glaciology, Advances in Numerical Analysis, 2011 (2012).

[36] Y. Kanarska, A. Shchepetkin, and J. McWilliams, Algorithm for non-hydrostatic dynamics in the regional oceanic modeling system, Ocean Modelling, 18 (2007), pp. 143–174.
[37] E. Karer and J. Kraus, Algebraic multigrid for finite element elasticity equations: Determination of nodal dependence via edge-matrices and two-level convergence, International Journal for Numerical Methods in Engineering, 83 (2010), pp. 642–670.

[38] C. Karney, Algorithms for geodesics, Journal of Geodesy, 87 (2013), pp. 43–55.

[39] S. Kim, Piecewise bilinear preconditioning of high-order finite element methods, Electronic Transactions on Numerical Analysis, 20 (2007), pp. 228–242.

[40] A. M. Le Brocq, A. J. Payne, and A. Vieli, An improved Antarctic dataset for high resolution numerical ice sheet models (ALBMAP v1), Earth System Science Data, 2 (2010), pp. 247–260.

[41] 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, 117 (2012).

[42] D. A. May and L. Moresi, Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics, Physics of the Earth and Planetary Interiors, 171 (2008), pp. 33–47.

[43] D. McKenzie, The generation and compaction of partially molten rock, Journal of Petrology, 25 (1984), pp. 713–765.

[44] G. A. Meehl, T. F. Stocker, W. D. Collins, A. T. Friedlingstein, A. T. Gaye, J. M. Gregory, A. Kitoh, R. Knutti, J. M. Murphy, A. Noda, S. C. B. Raper, I. G. Watson, A. J. Weaver, and Z.-C. Zhao, Global climate projections, in Climate Change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change, Cambridge University Press, 2007, pp. 747–845.

[45] G. A. Meurant, Computer Solution of Large Linear Systems, vol. 28 of Studies in Mathematics and Its Applications, North-Holland, Amsterdam, 1999.

[46] M. Morlighem, H. Seroussi, E. Larour, and E. Rignot, Inversion of basal friction in Antarctica using exact and incomplete adjoints of a higher-order model, Journal of Geophysical Research: Earth Surface, 118 (2013), pp. 95–108.

[47] N. Petra, H. Zhu, G. Stadler, T. J. R. Hughes, and O. Ghattas, An inexact Gauss-Newton method for inversion of basal sliding and rheology parameters in a nonlinear Stokes ice sheet model, Journal of Glaciology, 58 (2012), pp. 889–903.

[48] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM Journal on Scientific Computing, 14 (1993), pp. 461–469.

[49] Y. Saad, Iterative methods for sparse linear systems, Society for Industrial and Applied Mathematics, Philadelphia, PA, second ed., 2003.

[50] G. Schubert, D. Turcotte, and P. Olsen, Mantle Convection in the Earth and Planets, Cambridge University Press, 2001.

[51] H. Seroussi, H. Ben Dhia, M. Morlighem, E. Larour, E. Rignot, and D. Aubry, Coupling ice flow models of varying orders of complexity with the Tiling method, Journal of Glaciology, 58 (2012).

[52] A. Toselli and C. Schwab, Mixed hp-finite element approximations on geometric edge and boundary layer meshes in three dimensions, Numerische Mathematik, 94 (2003), pp. 771–801.

[53] U. Trottenberg, C. Oosterlee, and A. Schüller, Multigrid, Academic Press, London, 2001.

[54] P. Vaněk, M. Brezina, J. Mandel, et al., Convergence of algebraic multigrid based on smoothed aggregation, Numerische Mathematik, 88 (2001), pp. 559–579.

[55] P. Vaněk, J. Mandel, and M. Brezina, Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, Computing, 56 (1996), pp. 179–196.

[56] A. Zmitrowicz, Glaciers and laws of friction and sliding, Acta mechanica, 166 (2003), pp. 185–206.