Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 290 (2015) 496–523

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

A scalable, matrix-free multigrid preconditioner for finite element discretizations of heterogeneous Stokes flow

D.A. May[a,*], J. Brown[b,c,d], L. Le Pourhiet[e,f]

[a] Department of Earth Sciences, ETH Zürich, Zürich, Switzerland
[b] Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie (VAW), ETH Zürich, Zürich, Switzerland
[c] Argonne National Laboratory, 9700 S Cass Ave., Lemont, IL, USA
[d] Department of Computer Science, University of Colorado Boulder, 430 UCB, Boulder, CO 80309, USA
[e] Sorbonne Universités, UPMC Univ Paris 06, UMR 7193, Institut des Sciences de la Terre Paris (iSTeP), F-75005 Paris, France
[f] Univ. Nice Sophia Antipolis, CNRS, IRD, Observatoire de la Côte d'Azur, Géoazur UMR 7329, 250 rue Albert Einstein, Sophia Antipolis 06560 Valbonne, France

Received 19 June 2014; received in revised form 3 March 2015; accepted 22 March 2015
Available online 27 March 2015

Abstract

In this paper we describe a computational methodology that is specifically designed for studying three-dimensional geodynamic processes governed by heterogeneous visco-plastic Stokes flow. The method employs a hybrid spatial discretization consisting of a Q_2-P_1^{disc} mixed finite element formulation for the Stokes problem, coupled to a material-point formulation which is used for representing material state and history-dependent variables. The applicability and practicality of this methodology is realized through the development of an efficient, scalable and robust variable viscosity Stokes preconditioner. In this work, these objectives are achieved through exploiting matrix-free operators and a geometric multigrid preconditioner employing hybrid coarse level operators, Chebyshev smoothers and hybrid Krylov coarse level solvers. The robustness and parallel efficiency of this strategy is demonstrated using an idealized geodynamic model. Lastly, we apply the new methodology to study geodynamic models of continental rifting and break-up in order to understand the diverse range of passive continental margins we observe on Earth today.

Keywords: Stokes flow; Preconditioners; Multigrid; Parallel scalability; Geodynamics

1. Introduction

1.1. Computational geodynamics

Using a wide range of naturally-available data and observations, geology, geophysics and geochemistry aim to develop an integrative understanding of dynamical processes associated with Earth's tectonic history. Understanding the Earth's evolution is essential to: (i) explain the formation of both small scale (e.g. fold and thrust belts in a single

[*] Corresponding author.
E-mail address: dave.may@erdw.ethz.ch (D.A. May).

http://dx.doi.org/10.1016/j.cma.2015.03.014

mountain range) and large scale structures (e.g. geometry of plate boundaries and morphology of subducting plates); (ii) be able to make reliable estimates of past vertical displacement and heat flow, thereby providing better assessment for mining and oil industry; (iii) be able to make reliable estimates of the present day state of stress thereby providing better risk assessment for geo-hazards (e.g. earthquakes, tsunami) and underground storage (e.g. CO_2, nuclear waste, clean drinking water).

The interpretation of observations obtained from rocks of diverse ages located at the surface of the present day Earth (geology, geochemistry), combined with even a detailed image of the present day structure provide insufficient data to fully constrain the spatio-retro-temporal evolution of the Earth. Unraveling the mechanisms leading to the formation of the structures we observe today requires a forward model which describes the evolution of tectonic processes.

Complementary to data-driven fields of Earth science is the field of geodynamics, in which physics-based (continuum) representations describing the temporal evolution of deforming rocks are employed. Such geodynamic forward models are then steered through a careful prescription of constitutive relationships, boundary conditions and initial conditions, each constrained by natural observations and laboratory experiments. In this framework, geodynamic model validation is obtained by comparing present day observations with the output of forward simulations, thereby facilitating a means to construct a thermo-mechanically consistent description of the Earth's tectonic history.

1.2. Geodynamic modeling challenges

The structures we observe in the present day Earth result from large-deformation processes which span million-year time scales. Whilst rocks behave elastically on short time scales (e.g. seconds to kyr), over periods of 10^5–10^9 years, the dominant mode of deformation exhibited by rocks is ductile. Consequently, the evolution of rocks in geodynamic contexts is frequently described by the equations of stationary Stokes flow, which model the dynamics of an incompressible, very viscous, creeping fluid.

From the standpoint of developing computational models of geological processes, a number of modeling requirements (MR) must be satisfied. Numerical methods must be able to represent and track compositionally and rheologically distinct materials which possess highly nonlinear, history-dependent constitutive behavior (MR1). The capacity to continue to track the deformation of materials post-failure is an essential requirement as most natural structures result from a multitude of deformation phases (MR2). The inherent complexity and three-dimensionality of most natural structures also mandates that the methodology employed must be extensible to three spatial dimensions (MR3). In addition, the inclusion of a deformable free-surface, or a dynamic surface governed by a landscape evolution model is also an essential physical component in both regional and global scale geodynamic models e.g. [1–5] (MR4). At all scales, the dynamics of the surface is strong influenced by the rheology of the lithosphere (40–200 km thick) and crustal layers (5–50 km thick). High spatial resolution (MR5) is necessary to numerically resolve these layers and the brittle shear zones that develop within them.

The inclusion of compositionally distinct materials, together with the nonlinearities (which manifest as an effective viscosity) result in a highly spatially heterogeneous coefficient within the elliptic operator associated with the Stokes equations. Within geodynamic applications, the variations of the effective viscosity may be discontinuous and exhibit viscosity jumps on the order of 10^9 Pa s. The solver for the discrete Stokes system must be extremely robust (MR6) to efficiently handle the heterogeneous viscosity structures. Since the CPU time associated with the solution of the nonlinear Stokes system represents a majority ($\sim$90%) of the overall runtime of computational geodynamics applications, the performance of the Stokes solver is the leading bottleneck for scientific throughput and furthering our understanding of geological processes.

1.3. Previous work

To facilitate large deformation modeling of distinct materials in three spatial dimensions (MR1, MR2, MR3), particle-mesh methods derived from the Marker-And-Cell (MAC) scheme of Harlow and Welch [6] and the Material Point Method (MPM) [7] have been exploited by the geodynamics community since the late 1980s, e.g. [8–23].

The geometric flexibility associated with finite element methods and the natural manner to impose stress-free boundary conditions make this spatial discretization suitable to address MR3 and MR4. Whilst internal structures within the Earth may be subject to large strains, topographic variations are typically <10 km. These small variations (compared to the length scale of the domain) enable boundary fitted, structured hexahedral meshes to be utilized for both regional and global scale (e.g. via a cubed sphere) models.

High spatial resolution (MR5) has motivated the use of low-order mixed finite elements, such as the inf–sup violating Q_1–P_0 element [11,15,19,22], or the stabilized Q_1–Q_1 element [24–26]. Both element types may exhibit pressure artifacts due the unstable "checker-board mode" (Q_1–P_0 [27]), or due to the artificial compressibility introduced by the polynomial projection stabilization. The inf–sup stable, Q_2–P_1^{disc} element has been applied to 3D models of crustal deformation and subduction [23,20]. The element-wise incompressibility provided by this mixed element rendered it applicable for flow problems driven by Dirichlet boundary conditions, or by purely buoyancy variations. A major downside of this mixed element in 3D is that the stencil of the Q_2 element has, on average, 3×64 non-zeros (with a maximum of 3×125), in contrast to the stencil of a Q_1 element which only has 3×27 non-zero entries. Consequently, compared to a given Q_1 element resolution, both the memory usage and solution time will significantly increase when Q_2 elements are used.

To address MR6, a large number of practitioners have simply advocated exploiting parallel (multi-threaded or distributed memory) sparse direct solvers [11,28,22,19]. Although the performance of direct solvers is stable when switching from Q_1 to Q_2 (because the vertex separators remain the same size for the same number of degrees of freedom, [29]), the lack of algorithmic scalability places strict limitations on the spatial resolution possible. To circumvent this limitation, algorithmically scalable approaches utilizing iterative Krylov solvers with multilevel preconditioners have been explored e.g. [30,31,26,32]. In [23], we found that a Galerkin geometric multigrid preconditioner with heavy smoothers (GMRES/ILU(0)) was extremely reliable and robust for a wide range of geodynamic applications with highly heterogeneous viscosity structures.

1.4. Present work

In this work, we describe a computational methodology which aims to simultaneously satisfy all of the geodynamic modeling requirements MR1–MR6. We have made a number of fundamental design choices which result in a fast, highly scalable and robust Q_2–P_1^{disc} finite element implementation that is suitable for solving a wide range of geodynamic applications. Specifically these include: (i) utilizing an inf–sup stable mixed finite element (with an "unmapped" pressure space) which provides a reliable velocity and pressure solution; (ii) making extensive use of matrix-free operators which drastically reduces the memory requirements, improves efficiency, and improves the parallel scalability of the sparse matrix–vector product; (iii) deferring a wide range of choices associated with the solver configuration to run-time.

1.5. Outline

The structure of this paper is as follows: Section 2 outlines the underlying description of the viscous flow problem characterizing long time-scale geological processes; Section 3 describes the hybrid spatial discretization used for studying large deformation creeping flow; Section 4 describes the formulation of the nonlinear solver used; Section 5 presents the preconditioning strategies utilized to solve the Stokes problem; Section 6 outlines the parallel strategy adopted; Section 7 describes how the finite element matrix-free operators are applied and demonstrates their parallel scalability; Section 8 discusses the robustness and parallel scalability of the proposed Stokes solver; and lastly, Section 9 demonstrates the overall methodology by examining a three-dimensional model of continental rifting and break-up.

2. Governing equations

The conservation of momentum for a creeping fluid with an isotropic viscosity in a domain Ω with boundary $\partial\Omega$ is given by:

$$\left[2\eta\dot{\varepsilon}_{ij}(\boldsymbol{u})\right]_{,j} - p_{,i} = f_i, \tag{1}$$

where $\boldsymbol{u}$, p are the fluid velocity and pressure, η is the (nonlinear) effective shear viscosity, f_i is the body force and the strain-rate operator $\dot{\varepsilon}_{ij}$ is given by

$$\dot{\varepsilon}_{ij}(\boldsymbol{u}) = \tfrac{1}{2}\left(u_{i,j} + u_{j,i}\right). \tag{2}$$

We assume incompressibility (simplified mass conservation):

$$-u_{k,k} = 0. \tag{3}$$

The system equations (1) and (3) are closed with the following boundary conditions

$$u_i = \bar{u}_i \quad \boldsymbol{x} \in \Gamma_D \tag{4}$$
$$\sigma_{ij} n_j = \bar{t}_i \quad \boldsymbol{x} \in \Gamma_N, \tag{5}$$

where n_j is the outward pointing normal to the boundary $\partial\Omega$, Γ_D, Γ_N denote the regions along the boundary $\partial\Omega$ where the Dirichlet and Neumann boundary conditions are applied respectively, subject to the restrictions that $\Gamma_D \cap \Gamma_N = \emptyset$ and $\Gamma_D \cup \Gamma_N = \partial\Omega$. For the isotropic media we consider, the total stress σ_{ij} is given by

$$\sigma_{ij} = \tau_{ij} - \delta_{ij} p, \quad \text{where } \tau_{ij} = 2\eta\dot{\varepsilon}_{ij}. \tag{6}$$

The rheology of rocks over long time scales is frequently debated and revised in Earth sciences. Undoubtedly, the choice depends on the length scale (e.g. regional versus global) and the time scale (e.g. seismic versus Myr) considered. Many constitutive models exist for the effective viscous strength (η) of rocks over long term tectonics time scales (Myr → Gyr). They all assume that the viscosity depends, to first order, on chemical composition, temperature, pressure, strain-rate and path dependence. Here, we have chosen to approximate the thermal activated creep mechanism defining the viscous strength of rock by the Frank–Kamenetskii flow rule

$$\eta' = \eta_0 \exp(-\theta T), \tag{7}$$

where the parameters η_0, θ account for the local chemical composition of the rock and T is the temperature. Pressure and strain-rate dependence are introduced by limiting the effective viscosity so that stresses in the model do not exceed the Drucker–Prager yield stress

$$\tau_y = \min\left[p \sin\phi(\varepsilon^p) + C_0 \cos\phi(\varepsilon^p), \tau_{\text{mms}}\right], \tag{8}$$

which is limited by a maximum Mises stress, τ_{mms}. The cohesion C_0 is regarded to be constant, however the internal friction angle ϕ is assumed to decrease as a function of the accumulated plastic strain ε^p according to

$$\phi(\varepsilon^p) = \max\left(\phi_\infty, \phi_0 - \frac{\varepsilon^p\,(\phi_0 - \phi_\infty)}{\varepsilon^p_\infty}\right). \tag{9}$$

Eq. (9) defines an empirical softening relation which reduces the friction angle linearly with accumulated plastic strain similar to that of Lavier et al. [33]. Here ϕ_0 defines the initial friction angle. ε^p_∞ represents the measure of plastic strain after which complete softening is achieved and internal friction angle reaches ϕ_∞. Plastic strain represents an integrated, tensorial invariant measure of the deformation which has occurred due to plastic yielding. Thus, the quantity ε^p can be regarded as a simplified measure of material damage. The evolution of plastic strain is given by;

$$\frac{D\varepsilon^p}{Dt} = \begin{cases} \dot{\varepsilon}_{II}, & \text{if } \eta' > \tau_y/2\dot{\varepsilon}_{II} \\ 0, & \text{otherwise,} \end{cases} \tag{10}$$

where $\dot{\varepsilon}_{II} = \left(\frac{1}{2}\dot{\varepsilon}_{ij}\dot{\varepsilon}_{ij}\right)^{1/2}$. Note that the evolution of Eq. (10) only occurs when the material undergoes yielding [34]. The effective viscosity η, which accounts for both the parameterized creeping of viscous rocks and brittle behavior, is given by

$$\eta = \min\left[\eta', \frac{\tau_y}{2\dot{\varepsilon}_{II}}\right]. \tag{11}$$

The forcing term in the momentum equation is typically associated with buoyancy variations due to infinitesimal changes in density associated with thermal contraction/expansion (Boussinesq approximation), or compositional variations between different rock types. Thus, in general, here we will consider a forcing term of the form

$$\boldsymbol{f} = \rho_0\big(1 - \alpha(T - T_0) + \beta(p - p_0)\big)\hat{\boldsymbol{g}}, \tag{12}$$

where ρ_0 is the density defined by the composition of the rock as measured at a reference temperature and pressure T_0, p_0. The gravity vector is denoted by $\hat{\boldsymbol{g}}$ and α, β are the coefficients of thermal expansivity and adiabatic compressibility respectively.

3. Spatial discretization

To facilitate large deformation simulations, we adopt a hybrid spatial discretization which couples mixed finite elements for the flow problem $(\boldsymbol{u}, p)$, with a Lagrangian marker-based discretization to track constitutive behavior and material properties (e.g. η, ρ). The usage of such coupled "mesh-marker" discretizations for modeling large deformation geodynamic processes has been exploited by the Earth science community since the late 1980s. The particular hybrid finite element, marker-and-cell discretization which we utilize is described in more detail below. Geodynamic applications frequently utilize a nonlinear rheology which renders the viscosity as a function of velocity $\boldsymbol{u}$, pressure p and temperature T. In this work, we will assume that the nonlinear coupling due to the temperature dependence is weak, and thus the explicit temperature dependence in the nonlinear equations will be removed.

Over the domain Ω, the weak form of the nonlinear Stokes problem (Eqs. (1), (3)) is stated as follows: Find $(\boldsymbol{u}, p) \in V \times Q$ such that

$$\mathcal{A}(\boldsymbol{u}, \boldsymbol{w}, \eta(\boldsymbol{u}, p)) + \mathcal{B}(\boldsymbol{u}, q) + \mathcal{B}(\boldsymbol{w}, p) = \mathcal{F}(\boldsymbol{w}, \boldsymbol{f}(\boldsymbol{u}, p)) \tag{13}$$

for all $(\boldsymbol{w}, q) \in V \times Q$, where the bilinear forms $\mathcal{A}(\cdot,\cdot,\cdot)$, $\mathcal{B}(\cdot,\cdot)$ and linear functional $\mathcal{F}(\cdot,\cdot)$ in Eq. (13) are defined as

$$\mathcal{A}(\boldsymbol{u}, \boldsymbol{w}, \eta(\boldsymbol{u}, p)) = \int_\Omega 2\eta(\boldsymbol{u}, p)\dot{\varepsilon}_{ij}(\boldsymbol{u})\dot{\varepsilon}_{ij}(\boldsymbol{w})\, dV, \tag{14}$$

$$\mathcal{B}(\boldsymbol{w}, q) = -\int_\Omega q \nabla \cdot \boldsymbol{w}\, dV, \tag{15}$$

$$\mathcal{F}(\boldsymbol{w}, \boldsymbol{f}(\boldsymbol{u}, p)) = -\int_\Omega \boldsymbol{w} \cdot \boldsymbol{f}(\boldsymbol{u}, p)\, dV + \int_{\Gamma_N} \boldsymbol{w} \cdot \bar{\boldsymbol{t}}\, dS. \tag{16}$$

We note that the solution for the Stokes problem when employing a nonlinear rheology is potentially not in $H^1(\Omega)$. Accordingly we define the function space for velocity as $V := \{\boldsymbol{u} : \Omega \to \mathbb{R}^d,\ \boldsymbol{u} = \bar{\boldsymbol{u}} \text{ on } \Gamma_D\}$, where d denotes the spatial dimension and the functions in V are assumed to possess sufficient regularity to ensure that the weak form is well behaved. The function space for pressure used is $Q = \{q \in L_2(\Omega) : \int_\Omega q\, dV = 0\}$ if $\Gamma_N = \emptyset$, otherwise $Q = L_2(\Omega)$.

In this work, we partition Ω using a structured mesh of hexahedral elements. Over each element, we employ the mixed Q_2–P_1^{disc} basis function for velocity and pressure [27]. We consider models in which the structured mesh may be deformed, thus to preserve the order of accuracy of the Q_2–P_1^{disc} discretization we define the pressure basis in the x, y, z coordinate system, as opposed to in the "mapped" coordinate system [35,36]. The locally conservative (element-wise) properties of this element are highly desirable as many of the processes we intend to model utilize a free surface boundary condition ($\sigma_{ij} n_j = 0$) and additionally, the dynamics are entirely driven by buoyancy variations. For this class of problems, elements which weakly enforce the incompressibility constraint, for example the globally conservative Q_2-Q_1, or the low order stabilized Q_1–Q_1 [24], may result in unphysical velocity and pressure fields unless a high mesh resolution is employed [24,37].

The material properties η, ρ are defined on a set of Lagrangian markers (or material points) with coordinates $\boldsymbol{x}_p$. The spatial discretization of an arbitrary material property φ is given by

$$\varphi(\boldsymbol{x}) \approx \delta(\boldsymbol{x} - \boldsymbol{x}_p)\varphi_p. \tag{17}$$

Evaluation of the finite element weak form is performed using Gauss–Legendre quadrature. At the time of this evaluation, we require the values of all material properties at each quadrature point. We utilize an approximate local L_2 projection to define all quadrature point values associated with coefficients in the Stokes problem which are represented by material points. The coefficient reconstruction employs a Q_1 space, with nodal interpolants denoted via $N_i(\boldsymbol{x})$. This Q_1 space overlaps the elements defining the Q_2 space used to approximate the velocity field. The local reconstruction for a node i associated with the Q_1 basis is defined via

$$\hat{\varphi}_i = \frac{\int_{\Omega_i} N_i(\boldsymbol{x})\varphi(\boldsymbol{x})}{\int_{\Omega_i} N_i(\boldsymbol{x})} \approx \frac{\sum\limits_p N_i(\boldsymbol{x}_p)\varphi_p}{\sum\limits_p N_i(\boldsymbol{x}_p)}, \tag{18}$$

where the summation over p includes all material points contained within the support Ω_i of the interpolant N_i. Within each element, the material property, denoted by $\varphi^h(\boldsymbol{x})$ is defined via interpolation

$$\varphi^h(\boldsymbol{x}) = \sum_i N_i(\boldsymbol{x})\hat{\varphi}_i. \tag{19}$$

The variational statement for the discrete nonlinear Stokes problem with interpolated material properties $\eta^h, \mathbf{f}^h$ can be stated as: Find $(\mathbf{u}^h, \mathbf{p}^h) \in V^h \times Q^h$ satisfying

$$\mathcal{A}^h(\mathbf{u}^h, \mathbf{w}^h, \eta^h) + \mathcal{B}(\mathbf{u}^h, \mathbf{q}^h) + \mathcal{B}(\mathbf{w}^h, \mathbf{p}^h) = \mathcal{F}^h(\mathbf{w}^h, \mathbf{f}^h) \tag{20}$$

for all $(\mathbf{w}^h, \mathbf{q}^h) \in V^h \times Q^h$ where $V^h \in V$, $Q^h \in Q$ are the discrete finite dimensional spaces and

$$\mathcal{A}^h(\mathbf{u}, \mathbf{w}, \eta^h) = \int_\Omega 2\eta^h \dot{\varepsilon}_{ij}(\mathbf{u})\dot{\varepsilon}_{ij}(\mathbf{w})\, dV, \tag{21}$$

$$\mathcal{F}^h(\mathbf{w}, \mathbf{f}^h) = -\int_\Omega \mathbf{w} \cdot \mathbf{f}^h\, dV + \int_{\Gamma_N} \mathbf{w} \cdot \bar{\mathbf{t}}\, dS. \tag{22}$$

4. Nonlinear solver

To permit the usage of Newton like methods, we express the discrete problem in Eq. (20) in defect correction form. Given an arbitrary state variable for the velocity and pressure solution $\mathbf{X}' = (\mathbf{u}', \mathbf{p}')$, the discrete nonlinear residuals for the momentum and continuity equations will be denoted as

$$\mathbf{F}_u(\mathbf{u}', \mathbf{p}') = \mathcal{A}^h(\mathbf{u}', \mathbf{w}^h, \eta^h(\mathbf{u}', \mathbf{p}')) + \mathcal{B}(\mathbf{w}^h, \mathbf{p}') - \mathcal{F}^h(\mathbf{w}^h, \mathbf{f}^h(\mathbf{u}', \mathbf{p}')) \tag{23}$$

and

$$\mathbf{F}_p(\mathbf{u}') = \mathcal{B}(\mathbf{u}', \mathbf{q}^h) \tag{24}$$

respectively. Newton's method $\mathbf{J}(\mathbf{X}')\delta\mathbf{X} = -\mathbf{F}(\mathbf{X}')$ can be expressed as

$$\mathbf{J}\delta\mathbf{X} = -\mathbf{F} \rightarrow \begin{bmatrix} \mathbf{J}_{uu} & \mathbf{J}_{up} \\ \mathbf{J}_{pu} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta\mathbf{u} \\ \delta\mathbf{p} \end{bmatrix} = -\begin{bmatrix} \mathbf{F}_u \\ \mathbf{F}_p \end{bmatrix}, \tag{25}$$

where the Jacobian $\mathbf{J}$ is given by

$$\mathbf{J}_{ij} = \frac{\partial \mathbf{F}_i}{\partial \mathbf{X}_j}.$$

For linear problems, the Jacobian simplifies to

$$\mathbf{J} = \mathbf{J}_p = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{bmatrix}, \tag{26}$$

where $\mathbf{A}$ is the discrete gradient of the deviatoric stress tensor (see Eq. (21) for bilinear form) and $\mathbf{B}, \mathbf{B}^T$ are the discrete gradient and divergence operators respectively (see Eq. (15) for bilinear form).

We solve Eq. (25) using the JFNK framework [38]. Newton iterations are guarded by a backtracking line search and tolerances for the linear solve are adaptively set using the Eisenstat–Walker method [39,40]. To evaluate the nonlinear residual defined in Eq. (25), the following procedure is adopted. The current state $\mathbf{u}', \mathbf{p}'$ is used to evaluate both the nonlinear rheology and forcing function (Eqs. (11) and (12) respectively) at each material point, defining an effective viscosity η_p and force $\mathbf{f}_p$. These material point values are then projected on the quadrature points within each element following the method described in Section 3 and defined by Eqs. (18), (19).

5. Preconditioning

5.1. Stokes operator, J

We utilize flexible, right-preconditioned Krylov methods (e.g. FGMRES [41]; GCR [42]) to solve Eq. (25). The Jacobian system is preconditioned using an upper block triangular preconditioner [43,44] inspired from the linear problem defined in Eq. (26). This particular preconditioner is given by

$$\mathbf{B}_U = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & -\mathbf{S} \end{bmatrix}, \tag{27}$$

where $\mathbf{S} = \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}$ is the Schur complement. The preconditioned operator $\mathbf{J}_p \mathbf{B}_U^{-1}$ has minimal polynomial $(\mathbf{J}_p \mathbf{B}_U^{-1} - 1)^2 = 0$, thus a suitable Krylov method will converge in at most 2 iterations [45]. Since $\mathbf{S}$ is impractical to assemble, and accurate solves involving $\mathbf{S}$ are prohibitively expensive, we replace the Schur complement with a spectrally equivalent operator $\mathbf{S}^*$ associated with the bilinear form

$$\mathcal{S}^{h,*}(p, q) = \int_\Omega \frac{1}{\bar{\eta}}\, pq \, dV, \tag{28}$$

where $\bar{\eta}$ is an element-wise average of the viscosity. $\mathbf{S}^*$ is block diagonal due to the discontinuous P_1^{disc} pressure space p, q, thus $(\mathbf{S}^*)^{-1}$ can be defined element-wise. Spectral equivalence between $\mathbf{S}$ and $\mathbf{S}^*$ for variable viscosity problems was proven (under certain regularity conditions upon η) in Grinevich and Olshanskii [32] and has been demonstrated to be a robust and practical choice for many geodynamic applications [31,26]. Here we further approximate $(\mathbf{S}^*)^{-1}$ via $(\text{diag}(\mathbf{S}^*))^{-1}$, which in practice proves to be as robust as the true inverse of $\mathbf{S}^*$ for many geodynamic applications employing linear rheologies e.g. [43,26]. Thus the approximate upper block triangular preconditioner is given by:

$$\mathbf{B}_U \approx \mathbf{B}_U^* = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & -\text{diag}(\mathbf{S}^*) \end{bmatrix}. \tag{29}$$

When considering nonlinear problems we adopt an upper block triangular system $\mathbf{J}_U^*$ defined from the Picard linearization of Eqs. (23), (24). The Picard linearization results in an approximate Jacobian which is identical to the linear Stokes problem (Eq. (26)) in which the effective viscosity is defined by evaluating Eq. (11) with the current state variable $(\mathbf{u}', \mathbf{p}')$. Consequently, our preconditioner for the Newton solve can be expressed as:

$$\mathbf{J}_U^*(\mathbf{u}', \mathbf{p}') = \begin{bmatrix} \mathbf{A}(\mathbf{u}', \mathbf{p}') & \mathbf{B} \\ \mathbf{0} & -\text{diag}(\mathbf{S}^*(\mathbf{u}', \mathbf{p}')) \end{bmatrix}. \tag{30}$$

Similarly to the nonlinear residual evaluation, the projected field viscosity field η^h is used to define the viscosity dependent operators appearing within Eq. (30). We note that when pressure-dependent rheologies are used (e.g. Eq. (8)), $\mathbf{J}_{up} \neq \mathbf{B}$ and thus $\mathbf{J}$ will no longer be symmetric.

5.2. Viscous block, A

Applying the action of the preconditioner $\mathbf{y} \leftarrow (\mathbf{J}_U^*)^{-1}\mathbf{x}$ within the inexact Newton framework requires that we solve the auxiliary problem, $\mathbf{A}\mathbf{y}_1 = \mathbf{x}_1$, for $\mathbf{y}_1$. In this work, we consider obtaining the solution vector $\mathbf{y}_1$ using Krylov methods preconditioned with multigrid. Multigrid preconditioning is an efficient technique for solving elliptic partial differential equations (PDEs) by eliminating errors across all scales using a hierarchy of coarse meshes, or coarse sub-spaces. Multigrid preconditioners are desirable for large scale, high resolution simulations as they can exhibit optimal $O(n)$ algorithmic complexity and storage.

A wide range of multigrid methods exist, ranging from stencil-based approaches on uniformly refined structured meshes to purely algebraic methods which require only the matrix entries and non-zero structure. We will refer to the construction of coarse level operators as being generated via: *geometric coarsening* if the construction process required information from an underlying mesh; or *algebraic coarsening* if the construction phase only required a matrix (non-zero entries and non-zero pattern). We summarize the spectrum of multigrid preconditioners in Fig. 1.

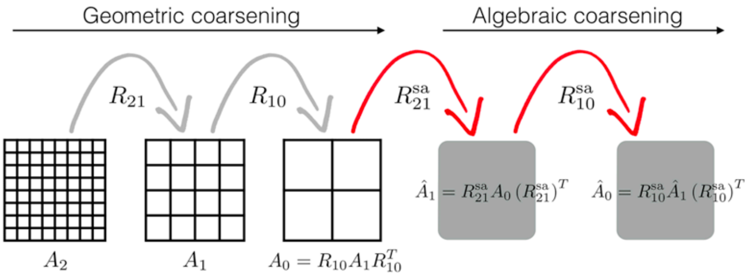
Geometric coarsening
Algebraic coarsening
R_{21}
R_{10}
R^{sa}_{21}
R^{sa}_{10}
$\hat{A}_1 = R^{\mathrm{sa}}_{21} A_0 \, (R^{\mathrm{sa}}_{21})^T$
$\hat{A}_0 = R^{\mathrm{sa}}_{10} \hat{A}_1 \, (R^{\mathrm{sa}}_{10})^T$
A_2
A_1
$A_0 = R_{10} A_1 R^T_{10}$

Fig. 1. Schematic of a hybrid, five level multigrid hierarchy composed from different coarse operator construction techniques. When considering problems with spatially variable coefficients, the robustness of the coarse operators will generally increase using the methods presented from left to right. Here R_{21}, R_{10} are geometric restriction operators between fine–coarse meshes, whilst R^{sa}_{21}, R^{sa}_{10} represent restriction operators defined via smoothed aggregation (SA) algebraic multigrid. The operators A_2, A_1 are constructed via re-discretizing (R) the PDE, the operator A_0 is constructed via Galerkin (G) projection using R_{10}. The operators $\hat{A}_1$, $\hat{A}_0$ are obtained from a Galerkin projection employing algebraically constructed restriction operators.

Purely geometric multigrid methods utilize a hierarchy of meshes, and define coarse level operators by re-discretizing the underlying PDEs on each level. The prolongation operator which maps field values from the coarse to fine level will be denoted via $\mathbf{P}$, whilst the restriction operator (mapping fields from fine to coarse levels) is denoted via $\mathbf{R}$. In methods employing geometric coarsening, $\mathbf{R}$ and $\mathbf{P}$ are defined by considering only the mesh geometry/topology. In the advent of an elliptic operator with a non-constant coefficient, employing re-discretized coarse level operators will require that a restriction operator (defined via geometric information) for the coefficient is provided. Galerkin multigrid approaches define coarse level operators $\mathbf{A}_c$ via Galerkin projection, i.e. $\mathbf{A}_c = \mathbf{RAP}$. Galerkin multigrid is regarded as using geometric coarsening as although the coarse operator construction is algebraic (i.e. employs matrix–matrix products), the underlying operator is defined using geometric information embedded within the $\mathbf{R}$ and $\mathbf{P}$ inter-grid transfer operators.

Elliptic operators possessing a non-constant coefficient which is highly spatially variable (smooth or rough) can adversely effect the convergence rate of multigrid preconditioners. In general, multigrid implementations employing re-discretized operators and interpolated coefficients will converge slower than an implementation using Galerkin coarse level operators. Likewise, for problems characterized by coefficients with highly localized small scale heterogeneities, purely algebraic methods are likely to yield the fastest convergence rate. There are two factors which deter us from simply relying on algebraic multigrid methods for all our geodynamic problems: (i) they require assembled matrices; (ii) the coarsening procedures generally possess non-scalable setup times; (iii) they have poor heuristics for discretizations which are higher order than Q_1; (iv) the cost of constructing $\mathbf{RAP}$ on all coarse levels is expensive (typically consuming approximately half of the setup time). Galerkin coarse level operators also suffer from points (i) and (iv), however the setup time required for symbolic and numerical evaluation of $\mathbf{RAP}$ are scalable. Assembled matrices are memory intensive, and have performance limited by memory bandwidth. In Section 7 we elaborate on this point further.

In this work, we advocate using a hybrid coarse operator construction which fuses re-discretized operators, Galerkin projection and purely algebraic approaches. This idea is illustrated in Fig. 1. *Through our hybrid coarse operator construction, we are optimistically seeking to exploit the most appropriate multilevel methods with the objective of minimizing memory and minimizing time to solution on the currently available massively parallel architectures.* The design of our geometric multigrid is flexible and permits a wide range of configurability in terms of how coarse level operators are defined, and in terms of the smoother that can be employed on each level. Such flexibility is desirable for the diverse range of geodynamic applications we wish to tackle—each of which possesses a very different viscosity structure. Furthermore, to exploit the known physical characteristics of the Stokes flow problem, choices related to the definition of the coarse level operators, smoothers and coarse level solver are deferred to run-time. We elaborate on the flexibility of the geometric multigrid preconditioner design below.

Our geometric multigrid hierarchy is composed from structured Q_2 finite element meshes. Levels within the hierarchy are numbered $0, 1, \ldots, N-1$, with the coarsest level denoted via the index 0. On each level k, the user can select to independently coarsen the number of nodes in the structured mesh in direction n_i, by an arbitrary factor (including 1) Δ_i, provided that $n_i - 1$ is exactly divisible by Δ_i. The prolongation $\mathbf{P}_k^{k+1}$ from level k to level $k+1$ are defined using trilinear interpolation (i.e., associated with an embedded Q_1 finite element space on the nodes of the Q_2

discretization). The restriction operator is defined as $\mathbf{R}_{k+1}^{k} = (\mathbf{P}_{k}^{k+1})^{T}$. Our finite element method employs Q_2 spaces, thus in the case of a constant coefficient, using a restriction and prolongation operator defined via an embedded Q_1 space will degrade the rate of the residual reduction of the multigrid preconditioner. However, we argue that when considering non-constant, non-smooth coefficients, the convergence rate will already be reduced, thus using lower order inter-grid transfer operators is unlikely to further adversely impact the multigrid convergence.

On the finest level $N-1$, we define a Q_1 representation of the material properties η^h using the projection described in Section 3. The fine level viscosity is interpolated to coarser levels using Q_1 interpolation. The interpolated viscosity is then evaluated at all quadrature points associated with the mesh at each level ($0 \leq k < N-1$). On each level, the coarse level operator $\mathbf{A}_k$ can be defined by re-discretizing the bilinear form for $\mathbf{A}$ using the interpolated viscosity field. Re-discretized operators may optionally be defined via assembled matrices or via a matrix-free representation (see Section 7 for more details). Additionally, if $\mathbf{A}_{k+1}$ is defined via an assembled matrix, we also allow users to optionally construct a Galerkin coarse level operator via $\mathbf{A}_k = \mathbf{R}_{k+1}^{k}\mathbf{A}^{k+1}\mathbf{P}_{k}^{k+1}$. This framework enables us to seamlessly blend combinations of re-discretized (assembled or matrix-free) operators and Galerkin coarse level operators within the same multigrid hierarchy.

We associate a preconditioned Krylov method with each level within the multigrid hierarchy. Depending on the level index within the hierarchy, the Krylov method is configured as a smoother, or as an inexact solver (e.g. for the coarse grid). In our current implementation, the only preconditioner available when using a matrix-free operator is Jacobi. Throughout the remainder of this manuscript, we will utilize smoothers which consist of a fixed number s of Chebyshev iterations (denoted Cheby(s)). Unless otherwise stated, the eigenvalue bounds used by Chebyshev are $[0.2\lambda_{\max}, 1.1\lambda_{\max}]$, where the estimate of the largest real eigenvalue ($\lambda_{\max}$) was determined using GMRES(5) preconditioned with Jacobi. The spectral estimate is only performed the first time the Chebyshev smoother is executed.

Chebyshev smoothers were investigated by Adams et al. [46] and have since been demonstrated to be effective parallel smoothers in a wide range of multigrid contexts [47–51]. In a massively parallel computing environment, the Chebyshev smoothers possess a number of desirable characteristics; (i) the application of each iteration does not require any global reductions to be performed; (ii) the smoothing characteristics are independent of the number of sub-domains used. In Section 8.2 we highlight the robustness of using Chebyshev smoothers for highly variable viscosity Stokes flow problems.

6. Parallelism

Support for all parallel linear algebra, in the form of matrices, vectors, preconditioners, Krylov methods and nonlinear solvers is provided by PETSc [52,53].

Parallelism is achieved by spatially decomposing the structured Q_2 finite element (FE) mesh containing $M \times N \times P$ elements, into structured sub-domains containing $m \times n \times p$ elements. One restriction of our implementation is that we require sub-domains to contain at least one Q_2 element. The underlying mesh data structure we use is defined by the DMDA object provided by PETSc.

Parallelism associated with material points is defined by the spatial decomposition of the FE mesh. Material points located within a particular mesh sub-domain associated with a processor k are managed by processor k. Following advection of all material points on a given sub-domain, a point location routine is applied. In the event that the point location routine determines that the material point is not located on the current sub-domain, the material point is inserted into a list $\mathcal{L}_s$. The material points in $\mathcal{L}_s$ are sent to all neighboring mesh sub-domains, and the point location is re-applied to the newly received material points $\mathcal{L}_r$. Material points in $\mathcal{L}_r$ which are not contained within the current mesh sub-domain are deleted. This simple strategy enables the communication of material points between processors and permits them to leave the domain if any outflow boundary conditions are prescribed.

7. Matrix-free operators

Sparse matrix–vector products (SpMV) are the most performance critical operation for Krylov methods and multigrid preconditioners with Chebyshev smoothers. Here we utilize matrix-free SpMVs for the operators $\mathbf{A}$, $\mathbf{B}$, $\mathbf{B}^T$, and $\mathbf{J}_p$ defined within the Stokes system and the upper block triangular preconditioner. Whilst mathematically equivalent, in comparison to assembled matrices, matrix-free operators have both reduced storage requirements and reduced memory bandwidth requirements, thereby leading to improved speed and scalability on modern multi-core hardware. These points are demonstrated in Section 7.1.

The general approach we adopt to efficiently evaluate the SpMV in a matrix-free manner is outlined in Algorithm 1, cf. [54]. Note that the procedure described (i) is relevant only for the $\mathbf{A}$ operator (discrete stress tensor gradient), (ii) does not include the procedure to impose Dirichlet boundary conditions and (iii) does not include function calls to facilitate parallel computations. Operations for points (ii)–(iii) can be placed around the procedure in Algorithm 1 and were thus omitted for clarity. The SpMV for the operators $\mathbf{B}$, $\mathbf{B}^T$, $\mathbf{J}_p$ can be defined in an analogous manner to that presented in Algorithm 1.

Algorithm 1 Matrix-free SpMV for $\mathbf{u} = \mathbf{A}\mathbf{v}$. Definitions: $\hat{\mathbf{J}}_q$ denotes the Jacobian of the coordinate transformation from physical space (x, y, z) to reference space (ξ, η, ζ), evaluated at the qth quadrature point located at ξ_q, with (reference-element) weight w_q. The evaluation of basis function derivatives at quadrature points $\nabla_\xi \mathbf{N}_q$ is precomputed on the reference element for each quadrature point q, and tabulated as a matrix of dimension $3|\mathcal{Q}| \times 27$ (there are 27 nodes in a Q_2 element and $\mathcal{Q}$ is the set of 27 points in a tensor-product Gauss rule, with 3 coordinate directions in the gradient). We note the number of flops (counting multiply-adds, FMAs) on the right.

```
procedure MatMult_MFKernalA($\mathcal{T}$, $\mathcal{Q}$, $\mathbf{v}$, $\mathbf{u}$)
    $\mathbf{u} = \mathbf{0}$
    for all $c \in \mathcal{T}$ do
        Get equation numbers $E_c$ for element $c$
        Get nodal coordinates $\mathbf{x}_c$ for element $c$
        Get nodal field $\mathbf{v}_c \leftarrow \mathbf{v}(E_c)$ for element $c$
        $\mathbf{u}_c \leftarrow \mathbf{0}$
        for all $q \in \mathcal{Q}$ do                                        ▷ 27 points/element
            Evaluate coordinate transformation: $\hat{\mathbf{J}}_q$                ▷ 243 FMA
            Invert transformation: $|\hat{\mathbf{J}}_q|, \hat{\mathbf{J}}_q^{-1}$                        ▷ 31 FMA
            Map gradient to physical: $\nabla_{\mathbf{x},q} = \hat{\mathbf{J}}_q^{-1}(\nabla_\xi \mathbf{N}_q) \otimes I_3$     ▷ 243 FMA
            Compute velocity gradient: $\nabla_{\mathbf{x},q}\mathbf{v}_c$                  ▷ 243 FMA
            Evaluate constitutive tensor: $\bar{\mathbf{D}}_q$
            $\dot{\varepsilon}_q = \frac{1}{2}\left[\nabla_{\mathbf{x},q}\mathbf{v}_c + (\nabla_{\mathbf{x},q}\mathbf{v}_c)^T\right]$          ▷ strain-rate tensor
            $\tau_q = \bar{\mathbf{D}}_q \, \dot{\varepsilon}_q$                                   ▷ stress tensor
            $\mathbf{u}_c \leftarrow \mathbf{u}_c + (\nabla_{\mathbf{x},q})^T w_q |\hat{\mathbf{J}}_q| \tau_q$                     ▷ 243 FMA
        end for
        $\mathbf{u}(E_c) \leftarrow \mathbf{u}(E_c) + \mathbf{u}_c$
    end for
end procedure
```

Performance model. To understand the performance of matrix-free operators versus assembled operators, we outline a performance model based on flops and memory bandwidth. In Algorithm 1, we have approximately 1000 FMAs per quadrature point. Whilst there are 27 quadrature points per element, in the asymptotic limit of increasing mesh sizes there are only 8 unique nodes per element within a Q_2 discretization, thus leading to approximately 3385 FMA per velocity node. With sufficient cache to achieve perfect reuse of the solution vector within a "wavefront", we need 3 `doubles` each for the velocity state and residual, 3 `doubles` for coordinates, and 27 `doubles` for the viscosity. This leads to an arithmetic intensity of 34 FMA/byte. Compared with a capability of about 4 FMA/byte for modern hardware (see Table 9 in Appendix A), the matrix-free operator is floating point-limited. Meanwhile, assembled matrices have 375 non-zeros per corner node, 225 non-zeros per edge node, 135 non-zeros per face node, and 81 non-zeros per interior node, leading to an average of 192 non-zeros per row, or 576 matrix entries per velocity node. With the AIJ/CSR sparse matrix format, each entry requires one `double` and one `int`, leading to an arithmetic intensity of 1/12 FMA/byte. This performance model indicates that the performance of assembled matrices is limited by memory bandwidth, and we indeed observe 70%–90% of STREAM Triad [55,56] on x86-64 architectures for operations such as SpMV and triangular solves, consistent with previous studies [57–59]. Moreover, assembled matrices have vastly higher bandwidth requirements than an equivalent matrix-free representation.

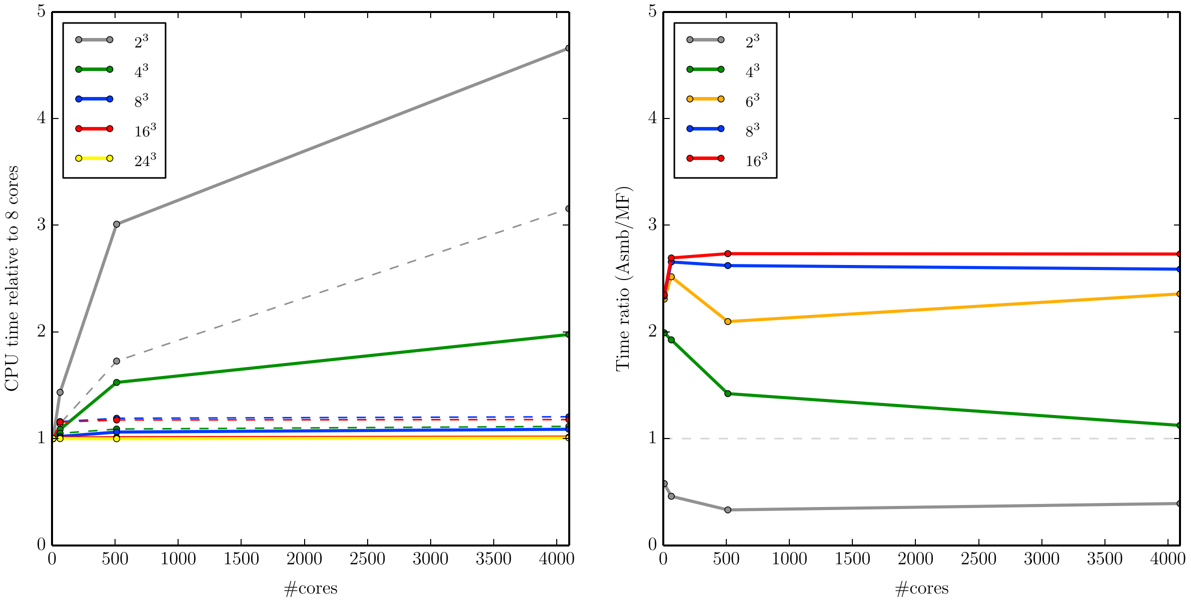

Fig. 2. Performance of SpMV for assembled (Asmb) and matrix-free (MF) operators. Left panel shows the weak scaling for Asmb (dashed lines) and MF operators (solid lines). The right panel shows the ratio of CPU time for Asmb/MF versus number of cores. The horizontal dashed line indicates the crossover point where the matrix-free implementation is faster.

7.1. Performance of SpMV for Q_2 operators

Here we examine the performance of our matrix-free (MF) SpMV implementation for **A** and compare it with an assembled matrix format (Asmb) provided by the MPIAIJ format from PETSc. All experiments were performed on “hexagon”, which is a Cray XE6 with 696 nodes, 2×16 AMD Interlagos chips (2.3 GHz) and 1 GB RAM/core. We note that at the time of writing, our MF SpMV kernel has been further optimized using AVX vector intrinsics [59].

In Fig. 2 we show the weak scaling obtained using sub-domains of differing sizes ranging from 2^3 Q_2 elements per core, to 24^3 Q_2 element per core. The case with 24^3 sub-domains could not be run with the assembled matrices as the application required more than 1 GB/core. To obtain representative timings in these experiments, we performed a fixed number of SpMV products such that the total execution time was approximately two minutes. The left panel shows the relative timing (with respect to 8 cores) for each of the sub-domains we considered. For ideal weak scaling, the relative timing should be equal to one across all core counts. The MF SpMV implementation is observed to scale poorly when sub-domains are $<8^3$ elements in comparison to the Asmb SpMV. The Asmb SpMV achieves a very flat relative time ($\approx$85%–90% efficiency) with sub-domains $\geq 4^3$. When sub-domains are $\geq 8^3$, the MF SpMV achieves nearly optimal weak scaling $\approx$99% efficiency. The efficiencies are reported in Table 1. The sub-optimal behavior observed for small sub-domains is attributed to latency in the network on the Cray XE6. When sub-domains are large and there is more work per core, the computation outweighs the cost associated with communication. The weak scaling of Asmb SpMV is limited by indirect memory access and the memory bandwidth. In contrast, the MF SpMV uses only local data and requires much less indirect memory access, thus resulting in better cache re-use.

In the right panel in Fig. 2, we report the CPU time ratio between the application time of Asmb SpMV and MF SpMV. It is apparent that the MF SpMV outperforms the assembled SpMV when sub-domains are $\geq 4^3$, with the speedup factor being approximately 1.2. A speedup of $\sim$2.7 is observed when sub-domains consist of $\geq 8^3 Q_2$ elements. The speedup factor is observed to increase as the sub-domain size increases. When the sub-domains are large, the speed of Asmb SpMV is entirely limited by memory bandwidth (attaining 70%–90% of STREAM Triad). Thus improved data locality and cache re-use in the MF SpMV yield faster matrix–vector products.

In Fig. 3 we report the speedup of the MF SpMV obtained using four different meshes with the total number of Q_2 elements given by 32^3, 64^3, 128^3, 384^3. For each of the four meshes, we increase the number of cores and report the speedup relative to the time obtained on 8, 64, 128, 512 cores respectively. In all experiments, we utilize all cores available on a node. The ideal speedup is denoted via the dashed lines for each test mesh. For each mesh, we scale the MF SpMV up to 4096 cores and obtain parallel efficiencies of 19%, 51%, 95% and 99% respectively.

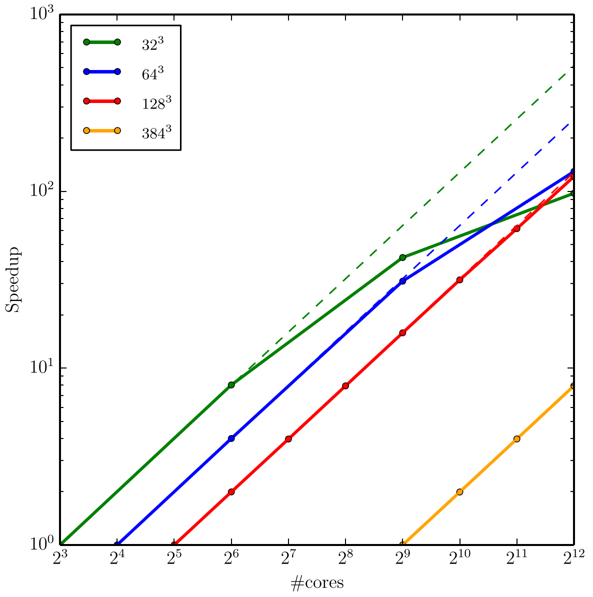

Fig. 3. Speedup (strong scaling) of the matrix-free SpMV on finite element meshes of different sizes; 32^3, 64^3, 128^3, 384^3. Solid lines indicate the actual speedup obtained; the ideal speedup is denoted via dashed lines.

8. Solver performance

8.1. Problem description

A simple model configuration which can be used to measure the robustness and scalability of the Stokes preconditioner for linear rheologies is one of particulate flow. Here, we consider a domain $\Omega \equiv [0, 1]^3$ filled with a fluid of density $\rho_b = 1$ and viscosity η_b. Embedded inside Ω are n_c spherical inclusions of radius r_c with density $\rho_i = 1.2$ and viscosity η_i. We will denote the viscosity ratio via $\Delta\eta = \max_\Omega[\eta(x)]/\min_\Omega[\eta(x)] = \eta_i/\eta_b$. On the upper boundary we impose $\sigma_{ij}n_j = 0$ (free surface) and on all other boundaries we impose $u_i n_i = 0$ and $\tau_{ij}t_j = 0$ (free slip), where t_j is a unit vector tangential to the domain boundary. Whilst simple in design, this experiment captures the necessary complexities required to study the dynamics of gas bubbles rising in a volcanic conduit [60], or the segregation and transport of crystals suspended in magma during their ascent within a dyke [61].

In the following sections, for any linear system denoted via $\mathbf{Lx} = \mathbf{b}$, we will terminate the iterative sequence at step k if $\|\mathbf{r}_k\|_2 < \delta^L_{rel}\|\mathbf{r}_0\|_2$, where $\mathbf{r}_k = \mathbf{Lx}_k - \mathbf{b}$. Unless otherwise stated, all iterative methods use a zero initial guess, thus $\mathbf{r}_0 = \mathbf{b}$.

8.2. Robustness

8.2.1. Single inclusion

First we consider the sedimentation of a single inclusion ($n_c = 1$) of radius $r_c = 0.25$, centered at (0.5, 0.5, 0.5). The viscosity ratio $\Delta\eta$ was varied to assess the robustness of the preconditioner. We explored viscosity ratios from $\Delta\eta = 1$ (iso-viscous) to $\Delta\eta = 10^8$. In each serial (single-core) experiment, we applied FGMRES to the Stokes system in Eq. (25) with $\delta^J_{rel} = 10^{-5}$. Within the block preconditioner, we used FGMRES on the $\mathbf{A}$ block together with $\delta^A_{rel} = 10^{-2}$. The definition of the coarse level operator used on each level was varied to gauge the performance gain arising from using a mixture of re-discretized and Galerkin operators. The smoother on each level was Cheby(4) preconditioned with Jacobi. The coarse level solver used the sparse LU factorization package UMFPACK [62]. The test results reported in Table 2 were performed on a single core using an Intel Xeon 2.67 GHz (Nehalem). In the column “Operator type” we identify the coarse level operators on each level in the multigrid hierarchy (ordered from coarse to fine) were defined. Coarse level operators defined by Galerkin projection are denoted via G, whereas levels employing re-discretized operators defined are denoted via R_a (assembled), or R (matrix-free).

Table 1
Weak scaling: Parallel efficiency of SpMV ($\mathbf{u} = \mathbf{A}\mathbf{v}$) from 8 → 4096 cores. Ideal weak scaling is represented by a parallel efficiency of 1.0.

Elements per core	Elements on 4096 cores	Efficiency	
		Asmb	MF
2^3	32^3	0.32	0.21
4^3	64^3	0.90	0.51
8^3	128^3	0.83	0.92
16^3	256^3	0.85	0.99
24^3	384^3	*	0.99

* Indicates the job required more than 1 GB RAM/core.

Table 2
Robustness of the Stokes and geometric multigrid preconditioner applied to the viscous inclusion example with $n_c = 1$ and 32^3 elements. A coarsening factor of two was employed between all mesh levels and in each i, j, k direction. Coarse level operators are identified as Galerkin (G), re-discretized using an assembled matrix (R_a), or re-discretized using a matrix-free representation (R).

Operator type	$\Delta\eta$	J its.	A its. min/max/total	Solve (s)	Mem. (MB)
R_a, R, R, R	1	4, 6, 6	1/2/28	2.1733e+02	979
R_a, R, R, R	10^2	17, 13, 15	2/8/132	9.3022e+02	1250
R_a, R, R, R	10^4	40, 25, 26	2/14/310	2.3489e+03	1364
R_a, R, R, R	10^6	99, 29, 37	2/45/994	6.8718e+03	1529
R_a, R, R, R	10^8	89, 14, 13	2/53/1105	7.8931e+03	1529
G, R_a, R, R	1	4, 6, 6	1/2/28	2.4150e+02	951
G, R_a, R, R	10^2	17, 13, 14	2/6/109	8.5308e+02	1209
G, R_a, R, R	10^4	27, 16, 17	2/10/146	1.1232e+03	1312
G, R_a, R, R	10^6	38, 15, 18	2/10/159	1.2444e+03	1312
G, R_a, R, R	10^8	40, 11, 14	2/10/143	1.1482e+03	1312
R_a, R, R	10^4	27, 16, 17	2/10/146	1.1765e+03	1621

From Table 2 the benefits are immediately apparent for even a mild viscosity ratio ($\Delta\eta = 10^4$), where using Galerkin projection on only the coarsest level yields an overall speed increase of 2.1× over three time steps. We also note that using a Galerkin operator on only the coarsest level does not significantly increase the memory requirements. The introduction of a single Galerkin coarse operator on the coarsest level allows us to solve problems with a viscosity ratio between 10^2–10^8 in almost constant time.

When using the multigrid preconditioner with four levels, the coarse mesh consisted of 4^3 Q_2 elements resulting in element edge lengths, $\Delta x = 0.25 = r_c$. The viscosity structure obtained from interpolating the fine grid viscosity will thus be poorly represented on the coarse mesh. From the number of iterations required to converge the viscous block (see first three rows Table 2), it is clear that re-discretized operators using this interpolated viscosity result in ineffective coarse grid correction. However, on the second coarsest level $\Delta x = 0.125 < r_c$, and thus a re-discretized operator should be effective. This hypothesis was verified by performing the same experiment using three multigrid levels (coarse mesh of 8^3) and re-discretized operators on each level (see final row in Table 2). With three levels, the re-discretized coarse level operator captures the viscosity structure adequately and the iteration counts on the Stokes system and viscous block sub-problems are significantly reduced in comparison with using four levels. From these experiments we conclude that Galerkin coarse level operators effectively capture the length scale of the viscous inclusion, even when these scales are smaller than the element size Δx. Thus, if mesh coarsening is used to define the multigrid hierarchy, whenever $\Delta x = 0.125 < r_c$, Galerkin coarse level operators should be utilized.

The residual history for the Stokes solve and viscous block solves are shown in Fig. 4 for the $\Delta\eta = 10^4$ case. We used a multigrid hierarchy of four levels and defined a Galerkin operator on the coarsest level. On the first time step, residuals in v, p dominate the solution, whilst the u, w residuals are small. Note that in our model configuration, gravity acts in the negative y-direction. Over the first 10 iterations, it is observed that the u, w, p residuals all increase to approximately the same order of magnitude of the largest residual (v). Once all the residuals are approximately the

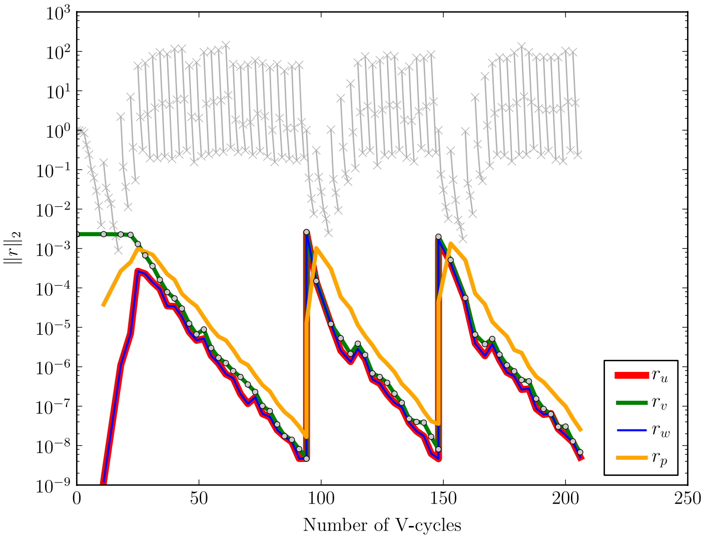

Fig. 4. Convergence history of u, v, w, p residuals (colored lines) during the linear Stokes solve with $\Delta\eta = 10^4$ over 3 time steps. The circles plotted on top of the v momentum denote the beginning of each Stokes iteration with respect to the number of V-cycles (x-axis). Gray lines indicate the convergence history associated with the **Au** = **v** solve performed during each application of the Stokes preconditioner, $(\mathbf{B}^*_U)^{-1}$. The pronounced jump in all residuals at 94 and 148 V-cycles corresponds to when the time step was taken. On subsequent time steps, the previous solution to the Stokes problem is used as an initial guess. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

same order of magnitude, convergence of each independent field occurs at approximately the same rate. We also note that the average number of iterations required to converge the viscous block is approximately constant within each Stokes solve and over the three time steps considered.

We also examined the convergence behavior of the **A** solve in isolation from the full Stokes solve. We conducted the same experiment with $\delta^A_{rel} = 10^{-8}$. For clarity, only the residual history is shown for the first application of the Stokes preconditioner $\mathbf{B}^*_U$. In practice, we find the first application of the Stokes preconditioner to yield the most challenging solve for **A**. The residual histories are shown in Fig. 5. Iteration counts are observed to depend on the viscosity ratio, with an initially large increase in iterations required when transitioning from iso-viscous (blue line) to a viscosity ratio of 10^2 (yellow line). Notably, we observe little difference in the iteration count when increasing the viscosity ratio from 10^4 to 10^8.

8.2.2. *Multiple inclusions*

A more challenging problem is the case when $\Delta\eta \gg 1$ and $n_c > 1$. The scenario of many sedimenting inclusions is more difficult to solve using multigrid due to the increased complexity of the eigenvectors associated with the low energy modes of the fine level operator **A**. To better illustrate this point, in Fig. 6 we compare the three eigenvectors associated with the three smallest eigenvalues for a 2D viscous inclusion problem using $n_c = 20$ with $\Delta\eta = 1$ (upper panel), and $\Delta\eta = 10^6$ (lower panel). The presence of the viscous inclusions is clearly evident in the variable viscosity example. For multigrid to be effective, the structure present within these low energy modes must be captured by the coarse level operator.

We performed several serial (single-core) experiments using 32^3 elements with $\Delta\eta = 10^4$, $r_c = 0.05$ and $n_c = 75$. The origins of the inclusions are chosen randomly, however we ensured that the inclusions do not overlap and that there is at least $2.1r_c$ separation between adjacent inclusions. The instantaneous flow field and the geometry of the inclusions is depicted in Fig. 7. FGMRES is applied to both the Stokes system and the inner viscous block, using the stopping conditions $\delta^J_{rel} = 10^{-5}$ and $\delta^A_{rel} = 10^{-2}$. For these serial experiments, the coarse level solver consisted of an LU factorization. In Table 3 we summarize the solver performance obtained using different coarse level operators and smoother combinations. In the column “Operator type” we identify how the coarse level operators within the multigrid hierarchy were defined. Coarse level operators defined by Galerkin projection are denoted via G, whereas levels employing re-discretized operators are denoted R_a (assembled), or R (matrix-free).

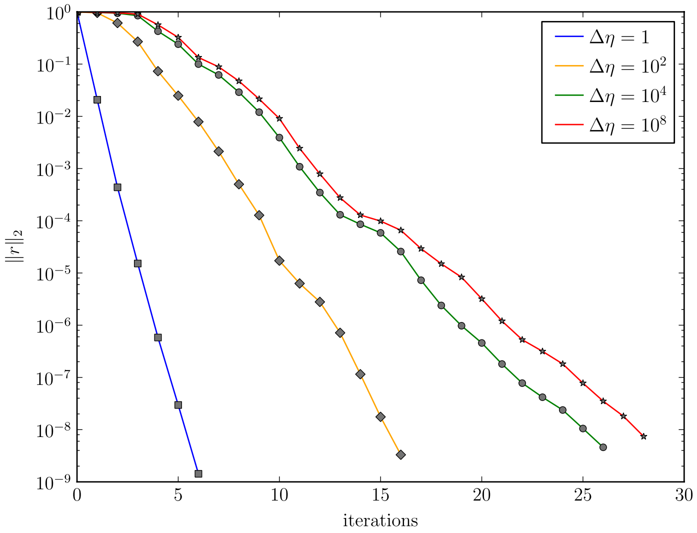

Fig. 5. Convergence history as a function of viscosity ratio $\Delta\eta$ for the $\mathbf{Au} = \mathbf{v}$ solve performed on the first iteration of the Stokes problem. Four multigrid levels were used with coarse level operators given by G, R_a, R, R. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

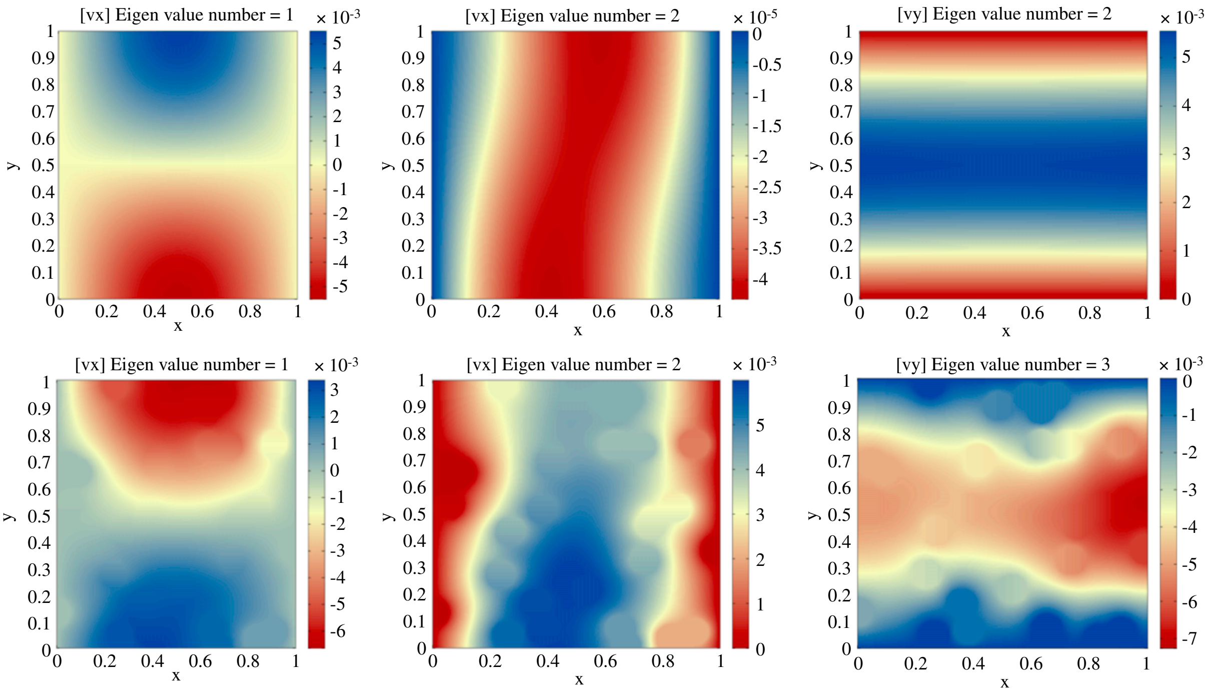

Fig. 6. Eigenfunctions of the viscous block $\mathbf{A}$ for a 2D viscous inclusion problem with $\Delta\eta = 1$ (upper panel) and $\Delta\eta = 10^4$ (lower panel).

Using re-discretized coarse level operators on all levels, with a smoother defined via Cheby(10) preconditioned with Jacobi (C10-J), utilizes the least amount of memory of all variants considered, however the solve time (Solve) is second to last. Introducing a Galerkin operator on only the coarsest level is not observed to reduce the overall run-time. However, using Galerkin operators on all levels proves effective in reducing the total run-time from 6453 s to 3425 s. The cost for this increased robustness is a 3.4× increase in memory. For all the C10-J experiments using Galerkin operators, the number of iterations required to converge the Stokes problem (J its.) remains approximately constant.

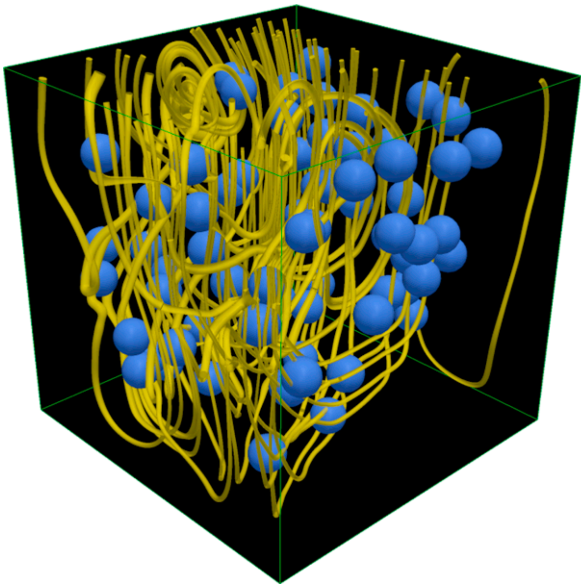

Fig. 7. Streamlines of the viscous inclusion example with $n_c = 75$ and $r_c = 0.05$. The streamline diameter is scaled according to the magnitude of the fluid velocity. The viscosity ratio between the inclusions (blue spheres) and the background material is $\Delta\eta = 10^4$.

Table 3
Performance of different smoother configurations for the sinker problem using 32^3 elements and with $\Delta\eta = 10^4$, $r_c = 0.05$ and $n_c = 75$. Column "Smoother G/R_a" indicates the smoother used on coarse levels which employed assembled matrices (Galerkin or re-discretized), whilst "Smoother R" indicates the smoother used on levels employing matrix-free operators. The smoother configurations are identified as follows: Cheby(10)-Jacobi → C10-J, FGMRES(10)-ILU(0) → F10-I.

Operator type	Smoother G/R_a	Smoother R	J its.	V-cycles (total)	Solve (s)	Mem. (GB)
R_a, R, R, R	–	C10-J	78	443	6.1901e+03	1.529
G, R_a, R, R	C10-J	C10-J	77	466	6.4532e+03	1.580
G, G, R_a, R	C10-J	C10-J	74	357	4.8965e+03	1.995
G, G, G, R_a	C10-J	–	81	287	3.4248e+03	5.354
G, G, R_a, R	F10-I	C10-J	66	292	4.4428e+03	2.480
G, G, G, R_a	F10-I	–	90	122	3.4204e+03	9.201

The total number of V-cycles (e.g. the total number of iterations performed on the viscous block sub-problem) is seen to decrease due to the increased robustness provided by the Galerkin coarse level operator.

To illustrate the reliability of the Chebyshev smoother, we compare the C10-J results with a more robust smoother configuration consisting of FMGRES(10) preconditioned with ILU(0) (F10-I). F10-I is applied on all levels with assembled operators, whilst C10-J is employed on any level using matrix-free operators. In comparison with using C10-J on both the assembled and matrix-free operators, the introduction of an FGMRES-ILU smoother does produce slight improvements in solve time, however at the expense of a significant memory increase.

8.3. Coarse level solver

Classic multigrid theory is based upon performing an exact solve on the coarsest level to obtain an accurate estimate of the error. In practice however, an exact coarse level solve is not required and inexact solves can be used without compromising the robustness, or convergence, of the multigrid preconditioner. The required accuracy of the inexact coarse level solve is determined by the convergence rate (residual reduction factor) of the multigrid preconditioner. For elliptic operators with highly spatially heterogeneous coefficients, the convergence rate is in general, not known a-priori.

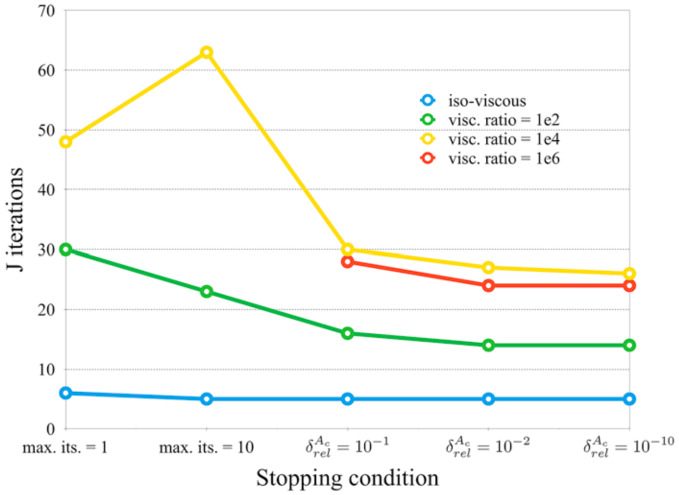

Fig. 8. Iterations required to solve the Stokes problem when using an inexact coarse level solver with different stopping conditions. Terminating the coarse level solve after a fixed number of iterations is denoted via “max. its.”, whilst terminating after a relative reduction of the residual is indicated via $\delta_{rel}^{A_c}$. When using a fixed number of iterations, the case $\Delta\eta = 10^6$ the Stokes problem failed to converge in 200 iterations.

Our motivation to use inexact coarse level solvers stems from a number of factors. Unlike elliptic operators with constant coefficients, the appropriate number of levels to be used within our geometric multigrid preconditioner is determined by the topology and length scales associated with the coefficient structure. Continued geometric coarsening (via re-discretized or Galerkin) of the variable viscosity viscous block operator without consideration of the viscosity structure results in an ineffective multigrid preconditioner (as discussed in Sections 8.2.1 and 8.2.2). This fact places an upper bound on the maximum number of levels within the geometric multigrid preconditioner and thus will often prevent us from coarsening to a mesh with few unknowns (e.g. meshes of 4^3 elements), where the problem can be efficiently solved exactly using sparse direct methods. We have experimented with using the parallel sparse direct solvers MUMPS [63] and SuperLU_Dist [64] as the coarse level solver for variable viscosity problems. We found that even with small sub-domain (e.g. $\leq 4^3 Q_2$ elements $\rightarrow <$ 3000 unknowns per core), these algorithms were not effective as a general purpose coarse level solver when using more than 512 cores as they either; (i) required too large a setup time, (ii) required more memory than was typically available on each core (1–2 GB), or (iii) were not stable over a wide enough range of mesh resolutions, parallel decompositions or model setups (e.g. produced segmentation errors). These problems have resulted in us pursuing inexact coarse level solvers.

To determine the necessary accuracy of the inexact coarse level solve, we consider the inclusion test case with $n_c = 1$ and report the total number of iterations required to solve the linear Stokes problem when the viscosity ratio $\Delta\eta$ is 1, 10^2, 10^4, 10^6. The Stokes preconditioner is defined as in Section 8.2.1, however the LU factorization on the coarse level is replaced with FMGRES preconditioned with Jacobi. As stopping conditions, we examine using either (i) a fixed number of iterations (max. its.), or (ii) a relative residual reduction of $\delta_{rel}^{A_c}$, where A_c denotes the operator on the coarsest level. In Fig. 8 we report the results of using these different stopping conditions. For the iso-viscous case, all stopping conditions result in convergence of the Stokes problem in at most, 6 iterations. However, for the variable viscosity cases, the number of Stokes iterations is strongly dependent on the choice of stopping condition. Overall, iteration counts are found to decrease as the accuracy of the coarse level solve is improved. For all cases, using $\delta_{rel}^{A_c} = 10^{-2}$ results in convergence of the Stokes problem in the same number of iterations as when an exact solve (stopping condition $\delta_{rel}^{A_c} = 10^{-10}$) on the coarse level was performed. We note that when using a fixed number of iterations (1 and 10), the Stokes problem failed to converge in 200 iteration for the case $\Delta\eta = 10^6$.

Given that problems involving highly spatially heterogeneous coefficients will potentially result in a geometric multigrid hierarchy containing a coarse mesh with many degrees of freedom, the choice of the inexact coarse level solver within the multigrid preconditioner is important for maintaining the robustness of the preconditioner, and to ensure good parallel scalability. One straightforward approach is simply to solve the coarse level problem using preconditioned Krylov methods. In the following we examine the feasibility of this naive approach. The test problem considered is identical to the single inclusion problem described in Section 8.1, with a viscosity ratio $\Delta\eta = 10^6$. We consider a one-level multigrid preconditioner using an element resolution of 16^3 and perform calculations from 8 to 4096 cores on “hexagon”. At the highest core count considered, each sub-domain contains only a single Q_2 element. We are primarily interested in the behavior of the coarse level solver at high core counts as this is when the convergence behavior is most likely to rapidly deteriorate and hinder scalability. Convergence of the coarse level solver is defined using $\delta_{rel}^{A_c} = 10^{-6}$. Iterations are terminated if convergence was not obtained in 800 iterations, or if the solve time exceeded 10 min. We note that depending on the actual convergence rate of the multigrid preconditioner used for the viscous block (which is problem dependent), such a high precision solve on the coarse level is not required to maintain the convergence of the multigrid preconditioner (see Fig. 8).

We explore two classes of coarse level solvers defined via (i) Krylov methods coupled with domain-decomposition style preconditioners and (ii) hierarchical Krylov methods [65]. The motivation behind using nested solvers stems from a desire to minimize the number of global reductions required to be performed during each Krylov iteration. To this end, in the following we considered using Krylov methods with an embedded, fixed number of Chebyshev iterations. In this way, we only incur the cost of the reductions associated with the Krylov method; we do not require any global reductions during the Chebyshev iterations. The results of our Krylov coarse level solver tests are presented in Table 10. The columns “Local solve”, “SpMV” and “Solve” indicate the CPU time required for: local solves performed by the domain-decomposition preconditioners; performing matrix–vector products; the entire solve. Below we summarize the main observations.

- FMGRES combined with an additive Schwarz preconditioner with overlap l (ASM(l)) and incomplete LU factorization with fill level k (ILU(k)) failed to converge within our specified wall time of 10 min for all overlap sizes considered ($l = 1, 2, 3$) and all fill factors considered ($k = 1, 3$). See solver configurations F1–F4 in Table 10.
- Chebyshev iterations preconditioned with ASM(l)/ILU(k) (denoted via C1, C2) resulted in convergence in all cases. Using $l = 2$ and $k = 3$, convergence could be achieved in 125 iterations and 357 iterations on 64 and 4096 cores respectively. The parallel efficiency from 64 to 4096 cores was approximately 24%.
- FMGRES with an embedded, fixed number of Chebyshev iterations are identified via configurations FNC1–FNC3 in Table 10. Chebyshev was preconditioned via block Jacobi (BJacobi) with ILU(k) as a sub-domain solver, or using ASM(l) with an incomplete Cholesky factorization with fill level k, ICC(k). All of the nested approaches tested were found to converge on 4096 cores in significantly less time (factors of $3\times - 20\times$) than when using Chebyshev alone as the iterative method. During our experiments, we observed that the results (iteration counts and thus converged residual values) were not reproducible when calculations were performed on the same number of cores, unless we used the slower, but more stable modified Gram–Schmidt orthogonalization (denoted by FNC1*–FNC3* in Table 10). With modified Gram–Schmidt orthogonalization, we observed a significant reduction in the iteration count when using ASM, and in one case a reduced CPU time.

  In the last four rows of Table 10, we consider the strong scaling efficiency of FNC2* (fastest configuration) from 8 to 4096 cores. Iteration counts are seen to grow from 19 to 42 and the total solve time achieves an overall parallel efficiency of 26% over this core count range. We also note that the total CPU time spent in performing the forward/backward solves (Local solve) and matrix–vector products (SpMV) also scale well.

8.4. Parallel scalability

We first consider the case of a single inclusion with $n_c = 1$ and $\Delta\eta = 1$ to assess the parallel scalability of the Stokes and geometric multigrid preconditioner. As in previous experiments, we used $\delta_{rel}^{J} = 10^{-5}$ and $\delta_{rel}^{A} = 10^{-2}$ as the stopping condition for the Stokes system and viscous block solve inside $\mathbf{B}_U^*$. Due to the simplicity of the viscosity structure, we use relatively weak smoothers and coarse level solver. On every level (except the coarsest) we define the smoother to be Cheby(10)/Jacobi. All levels utilize re-discretized (matrix-free) operators, except on the coarsest level where the operator is assembled. The coarse level solver consisted of FGMRES +[Cheby(30) +BJacobi/ILU(0)] (similar to FNC1* in Table 10). The stopping condition used for the coarse level solver was $\delta_{rel}^{A_c} = 10^{-2}$, with the maximum number of iterations set to 8. All calculations were performed on “hexagon”.

The weak and the strong scaling for the iso-viscous experiments are reported in Tables 4 and 5 respectively. The CPU times reported relate to one complete Stokes solve. Here “Coarse solve” refers to the total CPU time spent in the coarse level solver, whilst “Smoother” indicates the total CPU time spent in applying the Chebyshev iterations on the finest level. “Solve” indicates the total time required to obtain the solution to the Stokes problem.

Table 4
Weak scaling results for the iso-viscous sinker example, $\Delta\eta = 1$. The stopping conditions used were $\delta^J_{rel} = 10^{-5}$ and $\delta^A_{rel} = 10^{-2}$. The resolution of the coarse mesh in all experiments was 24^3.

Elements (fine)	MG levels	Cores	J its.	A its. min/max	Coarse solve (s)	Smoother (s)	Solve (s)
96^3	3	64	6	1/1	4.2450e+01	7.6928e+01	1.4905e+02
192^3	4	512	6	1/2	5.0023e+00	7.7124e+01	1.1335e+02
384^3	5	4096	7	1/2	1.1851e+00	8.8014e+01	1.2545e+02

Table 5
Strong scaling results for the iso-viscous sinker example, $\Delta\eta = 1$. The stopping conditions used were $\delta^J_{rel} = 10^{-5}$ and $\delta^A_{rel} = 10^{-2}$. The geometric multigrid preconditioner used three levels, the finest level contained 96^3 elements. Beside each CPU time reported, we indicate the parallel efficiency in brackets.

Cores	J its.	A its. min/max	Coarse solve (s)	Smoother (s)	Solve (s)
32	6	1/1	7.3177e+01 (-)	1.3241e+02 (-)	2.6029e+02 (-)
64	6	1/1	4.2450e+01 (86)	7.6928e+01 (86)	1.4905e+02 (87)
128	6	1/1	2.0297e+01 (90)	3.3216e+01 (100)	6.7289e+01 (97)
256	6	1/1	9.9570e+00 (92)	1.8972e+01 (87)	3.6436e+01 (89)
512	6	1/1	4.9781e+00 (92)	9.4518e+00 (88)	1.8371e+01(89)
1024	6	1/1	1.4890e+00 (154)	4.1281e+00 (100)	7.5922e+00 (107)
2048	6	1/1	8.7723e−01 (130)	2.4537e+00 (84)	4.5177e+00 (90)
4096	6	1/1	1.0344e+00 (55)	1.3337e+00 (78)	3.1571e+00 (64)

From Table 4, the weak scaling results show that the total CPU time is approximately constant from 64 to 4096 cores, as is the time spent in the smoother on the finest level. The time spent in the coarse level solver is seen to decrease with increasing core counts. We note that on 64 cores, all coarse level solves converged due to the relative tolerance of 10^{-2}, however on 512 cores, several of the coarse level solves terminated early as they exceeded the maximum number of iterations. On 4096 cores, every application of the coarse level solver failed to converge in 8 iterations. With increasing core counts, the sub-domain size decreases, thus the forward and backward solves associated with ILU(0) become cheaper and result in a decrease in the coarse level solve time. When using 64 cores, we observe that cost of the coarse level solve is out-weighed by the time spent applying the smoother on the finest level. This persists at higher core counts and assists in obtaining the good weak scaling observed. The iteration counts for the Stokes problem and the viscous block sub-problem are observed to be approximately constant in going from a 96^3 to 384^3 mesh. For this iso-viscous model, we observe excellent weak scaling efficiency when scaling from 32 to 4096 cores.

The strong scaling experiments summarized in Table 5 consider an iso-viscous model with a total of 96^3 Q_2 elements. A total of three geometric multigrid levels were used. The scaling was performed from 32 to 4096 cores. Iteration counts are observed to be constant, thereby establishing that the smoothing properties of the method adopted is independent of the number of cores. In regard to strong scaling, we observed an overall parallel efficiency of 64%.

We now consider a variable viscosity case with $\Delta\eta = 10^4$. Based on the results from Table 2, we configure the coarse level operators within the multigrid hierarchy in the following way: G, R_a, R. On every level (except the coarsest) we define the smoother to be Cheby(10)/Jacobi. The coarse level solver consisted of FGMRES +[Cheby(10) +ASM(1)/ICC(0)] (similar to FNC2* in Table 10). The stopping condition for the coarse level solver uses $\delta^{A_c}_{rel} = 10^{-2}$, or was terminated after 800 iterations. The strong scaling for the variable viscosity experiments are reported in Table 6.

Unlike the iso-viscous example, to obtain constant iteration counts for the Stokes and viscous block sub-problems as the number of cores increase, it is necessary to converge the coarse level problem to a given accuracy (see Fig. 8). For this reason, a more aggressive coarse level solver was used in these variable viscosity problems. From Table 6 we observe that the number of iterations used to converge the Stokes problem and the cumulative number of iterations used for all the **A** sub-problems is approximately constant from 64 to 4096 cores. Only the cumulative number of iterations used to solve the coarse level problems are seen to increase with increasing core counts. Thus, on 64 cores the coarse level solve time is 4.8× less than the time spent within the fine level smoother, however at 4096 cores, applying the

Table 6
Strong scaling results for the variable viscosity ($\Delta\eta = 10^4$) sinker example broken down into a number of different events within the preconditioner. The stopping conditions used were $\delta^J_{rel} = 10^{-5}$ and $\delta^A_{rel} = 10^{-2}$. The geometric multigrid preconditioner used three levels, the finest level contained 96^3 elements.

Cores	64	512	4096
Preconditioner setup (s)	2.6353e+00	5.6080e−01	2.3424e−01
Assemble $P^T AP$ (s)	1.1765e+01	1.4522e+00	2.3470e−01
SpMV (s)	8.7006e+02	9.7556e+01	1.4648e+01
Global reductions (s)	3.5105e+01	1.2900e+01	2.2512e+00
Local solve (s)	1.0196e+02	2.1817e+01	3.5088e+00
Coarse solve (s)	1.8872e+02	3.3849e+01	9.1787e+00
Smoother (s)	4.8848e+02	5.1566e+01	7.2146e+00
Solve (s)	9.9545e+02	1.1651e+02	1.9926e+01
J its.	24	24	23
A its.	100	101	98
A coarse its.	347	399	495

Table 7
Comparison of CPU times (s) for the variable viscosity ($\Delta\eta = 10^4$) sinker example using different coarse level solver strategies. The Stokes preconditioner used the following stopping conditions: $\delta^J_{rel} = 10^{-5}$, $\delta^A_{rel} = 10^{-2}$. The geometric multigrid preconditioner used three levels, the finest level contained 96^3 elements. The coarse level solver "GAMG" refers to the smoothed aggregation algebraic multigrid implementation within PETSc. "Re-partitioned" refers to a coarse level preconditioner which re-distributes the coarse grid onto processor sub-sets. See text for further details.

Cores		64	512	4096
GAMG	Coarse solve	3.3929e+01	4.8522e+00	3.6687e+00
	Smoother	3.4835e+02	4.3411e+01	6.9875e+00
	Solve	5.9950e+02	7.4663e+01	2.1039e+01
Re-partitioned	Coarse solve	1.4028e+02	1.7607e+01	2.9893e+00
	Nested coarse solve	1.5587e+01	1.9563e+00	3.3214e−01
	Smoother	3.0379e+02	3.8059e+01	5.6223e+00
	Solve	6.4287e+02	8.0635e+01	1.1826e+01

coarse level solver cost approximately the same as applying the fine level smoother. The Chebyshev smoother, together with the nested Krylov coarse level solver are effective at minimizing the number of global reductions, with only 11% of the total solve time at 4096 cores being spent in such operations. Despite the apparent non-scalable coarse level solver, when scaling from 64 to 4096 cores, the method achieves an overall strong scaling parallel efficiency of 78%.

We conclude our strong scaling study by comparing the naive Krylov coarse level solver (discussed above) with (i) a coarse level solver employing the smoothed aggregation algebraic multigrid preconditioner within PETSc (GAMG), and (ii) a coarse level solver employing a re-partitioned mesh. The coarse level solver within GAMG consisted of block Jacobi, with an exact LU factorization applied on each sub-domain. Furthermore, a thresholding value of 0.01 was used for constructing the aggregation graph. Assuming we have decomposed all levels within our geometric multigrid preconditioner over n_p processors, the re-partitioning approach used here re-distributes the coarsest mesh within the hierarchy onto n_p/c processors, where c defines the processor coarsening factor. The sub-set of processors used to define the re-partitioned mesh are associated with a unique MPI communicator, thus global reductions are only performed across n_p/c processors. The solver used on the re-partitioned mesh consisted of a two level geometric multigrid employing Galerkin coarse level operators. As a coarse level solver within the re-partitioned hierarchy, we used GAMG. The smoother used within both the geometric hierarchy (original spatial decomposition and the re-partitioned decomposition) and the algebraic multilevel hierarchy was given by Cheby(10)-Jacobi. In the experiments we conducted, $c = 16$ was used in all cases.

The comparison of the two alternative coarse level solvers is reported in Table 7. In comparison to the "Krylov only" coarse level solver (presented in Table 6), GAMG appears to be significantly faster on 64 and 512 cores, however it is only marginally faster on 4096 cores. GAMG exhibits an overall strong scaling efficiency of 44%. We note that our configuration of GAMG does not appear to scale well past 512 cores. The re-partitioned, nested geometric multigrid preconditioner appears to be significantly faster than the "Krylov only" coarse level solvers over all core counts. We find that while GAMG out-performs the re-partitioned approach on 64 and 512 cores, the re-partitioned approach is almost 1.8× faster then GAMG at 4096 cores. The overall strong scaling efficiency of the re-partitioned approach is 85%. The efficiency of the coarse solver, in isolation from the complete Stokes solve, is found to be 73%.

9. Geodynamic example: continental rifting and break-up

9.1. Background

Here we present results of 3D numerical models which simulate the propagation of continental break-up. The break-up of continents occurs after a phase of continental rifting, in which the crust thins either over wide regions, or narrow localized ones, thereby participating to the diversity of passive continental margins. The processes that lead to continental break up, as well as the rate of thinning and initial stages whereby the continental crust thins are important factors influencing the formation, maturation and the storage of hydrocarbons along passive margins, and hence are of prime economical interest.

The timing of continental break-up has been carefully documented over our planet, revealing that when two continents separate, there are periods of very fast propagation of break-up and periods of quiescence [66]. The period of quiescence is usually associated with large fracture zones and formation of oblique margins. Such oblique structures cannot be understood using 2D models. Whilst numerous 3D numerical simulations of continental break-up have been performed [67–72], due to the algorithmic challenges associated with high resolution 3D, time-dependent, nonlinear, large deformation modeling of the lithosphere and crust, few satisfy modeling requirement MR5 (see Section 1.2). Previous work consider model domains on the order of $100 \times 50 \times 100$ km^3, or use a low spatial resolution (∼50 km). Most other 3D simulations of rifting are limited to oblique continental extension [73–78], or consider models of oceanic spreading [79,80] which require fewer time steps and/or smaller model domains.

9.2. Model description

The models we present here are designed to determine how periods of quiescence might occur during the propagation of continental break-up and how such periods relate to the formation of oblique margins. In this demonstration we solve the Stokes problem (Eqs. (1), (3)) with the temperature, pressure and strain-rate dependent viscosity given by Eq. (11). This definition of the effective viscosity involves a stress limiter τ_y which parameterizes the brittle behavior of rocks near the surface. The rocks are assumed to have a buoyancy defined by Eq. (12) in which both α and β are non-zero and the reference pressure and temperature were taken as $p_0 = 0$ Pa and $T_0 = 0$ K.

The history-dependent plastic strain (damage) variable ε^p is defined on each material point. The evolution of Eq. (10) is updated using an explicit forward difference in time approximation. At each material point $\boldsymbol{x}_k$, if yielding occurs, the plastic strain is updated following the solution of the Stokes problem according to

$$\varepsilon^p(\boldsymbol{x}_k, t + \Delta t) = \varepsilon^p(\boldsymbol{x}_k, t) + \Delta t\, \dot{\varepsilon}_{II}(\boldsymbol{x}_k, t), \tag{31}$$

where Δt is the time step.

The model domain spans 1200 km × 600 km in the x–z direction and 200 km in the y direction. Initially, the model domain is divided into three regions which we refer to as "mantle", "weak crust" and "strong crust". Material points contained within the mesh are used to identify the different regions. Each region defines a unique set of material parameters. The layer geometry and material parameters are defined in Fig. 9 and Table 8.

In our models, the damage state variable ε^p is initially assigned a random value in the range [0, 0.03], except within a localized damaged zone where the amplitude of the noise is in the range [0, 0.3]. This damaged zone has been added near the front side ($z = 0$ km) of the model domain (see Fig. 9) to trigger rapid continental rifting and break-up, thereby allowing us to study how this localization propagates within the model domain.

It is known from 2D numerical simulations [81,3] that the rheological nature of the lower crust is of prime importance for the structuration (the fault spacing and sequence) of the thinned continental crust along passive margins. For this reason, we investigate how break-up propagates when using a weak lower crust (representing a mobile belt (MB)), or a strong lower crust (representing a continental shield (CS)). The upper and lower crust are both initially 20 km thick (see Fig. 9).

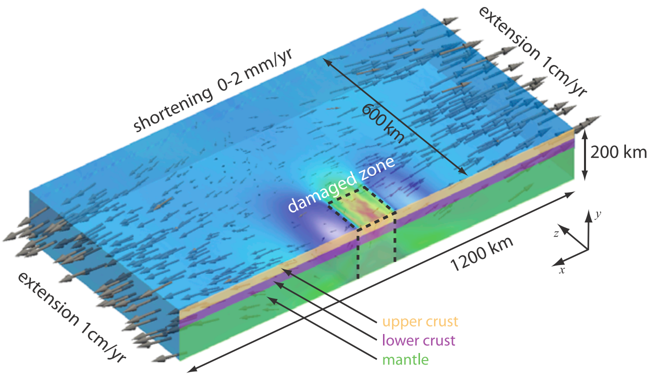

Fig. 9. The solution of the CS model with non cylindrical boundary conditions is used to illustrate the model set-up. The velocity field (gray arrows) illustrates the influence of the slight shortening component in the z-direction. The second invariant of strain-rate (shown as transparent) outlines the initial damaged zone.

Previous 3D instantaneous flow models have also shown that the direction of the flow in the mantle beneath the tip of the propagating rift might enhance, or annihilate the propagation [82]. Thus we additionally explore the effects of using two types of velocity boundary conditions. The first consists of purely cylindrical extension system of 2 cm/yr applied symmetrically in the x-direction. The second boundary condition (depicted in Fig. 9) consists of symmetric extension of 2 cm/yr applied in the x-direction, together with a slight component of shortening (2 mm/yr) in the z-direction, which is applied on the opposite side of the damaged zone.

The models employ an ALE formulation in which only the free surface is advected with the fluid velocity, whilst all other boundaries remain stationary through time. At the base of the model, we prescribe an inflow velocity v_{base} which is computed at each time step such that the average volume of the domain remains approximately constant.

The evolution of temperature T is given by the following energy conservation equation

$$\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \nabla T = \nabla \cdot (\kappa \nabla T) . \tag{32}$$

In our implementation, the temperature T in Eq. (32) is discretized using Q_1 finite elements which overlap the Q_2 velocity space. The advection–diffusion equation is stabilized via SUPG [83]. The Stokes problem and energy equation are solved in a de-coupled manner, thus the temperature dependence in the rheology (Eq. (7)) and forcing function (Eq. (12)) does not introduce an additional nonlinearity. Dirichlet boundary conditions are applied at top ($y = 200$ km) and bottom ($y = 0$ km) of the model domain, with respective values of $T_a = 0$°C and $T_b = 1400$°C. Neumann boundary conditions corresponding to zero normal heat flow are prescribed on all vertical boundaries. The initial conditions for temperature were set according to the analytical solution for 1D infinite half space cooling model [84], with a cooling time of 300 Myr and a thermal diffusivity $\kappa = 10^{-6}$ m^2 s^{-1}. In this analytic solution, the temperature at infinity and at the surface corresponds to the Dirichlet boundary conditions T_b and T_a respectively.

The inclusion of a stress limiter (τ_y) within the rheology and the choice of non-zero values for β in Eq. (12) introduce several nonlinearities into the Stokes problem. The solution of the resulting nonlinear system is described in Section 4.

The simulation results presented were performed using 512 cores on “Rostand”, an SGI ICE 8200 with compute nodes consisting of 2 × 6 Intel Xeon (Series 5600) cores running at 2.8 GHz. Each model utilized a mesh resolution of $256 \times 32 \times 128$ Q_2 elements. The multigrid preconditioner for $\mathbf{A}$ was configured with three levels, using the following element hierarchy (from the coarsest to the finest level); $32 \times 16 \times 16$, $128 \times 32 \times 64$, $256 \times 32 \times 128$. We used CG+ASM(4)/ILU(0) (see Table 10, configurations F1–F4) as the coarse level solver. The coarse level solver was terminated after 25 iterations, or if the initial residual was reduced by a factor of 10^4. Each model required approximately 1500 time steps, with the average CPU time per time step being ∼160–200 s. This CPU time reflects the total time taken for the solution of the nonlinear Stokes problem and all other components, including; nonlinear residual evaluations, interpolation between material points and the quadrature points, updating all material point properties (velocity, position, plastic strain), all mesh geometry updates required by the ALE formulation and any requested output.

Table 8
Physical quantities of the different material regions used in the rifting experiments. All parameters are defined in Section 2. The mobile belt (MB) lithosphere and continental shield (CS) lithosphere models only differ in the definition of the lower crust. In the CS model, the parameters listed under "strong crust" are applied in both the upper and the lower crust regions, whereas in the MB model they are only used in the upper crustal region.

Quantity	Unit	Strong crust	Weak crust	Mantle
η_0	Pa s	10^{27}	10^{26}	10^{30}
θ	$^\circ C^{-1}$	2×10^{-2}	2×10^{-2}	1.8×10^{-2}
ϕ_0	$^\circ$	30	30	30
C_0	Pa	2×10^7	2×10^7	2×10^7
τ_{mms}	Pa	2×10^8	2×10^8	3×10^8
ϕ_∞	$^\circ$	6	6	6
ε^p_∞	-	0.3	0.3	0.15171
ρ_0	kg m^{-3}	2.7×10^3	2.7×10^3	3.3×10^3
α	$^\circ C^{-1}$	2×10^{-5}	2×10^{-5}	2×10^{-5}
β	Pa^{-1}	0	0	3×10^{-12}
p_0	Pa	0	0	0
T_0	K	0	0	0
κ	m^2 s^{-1}	10^{-6}	10^{-6}	10^{-6}
Layer thickness [CS]	km	40	0	160
Layer thickness [MB]	km	20	20	160

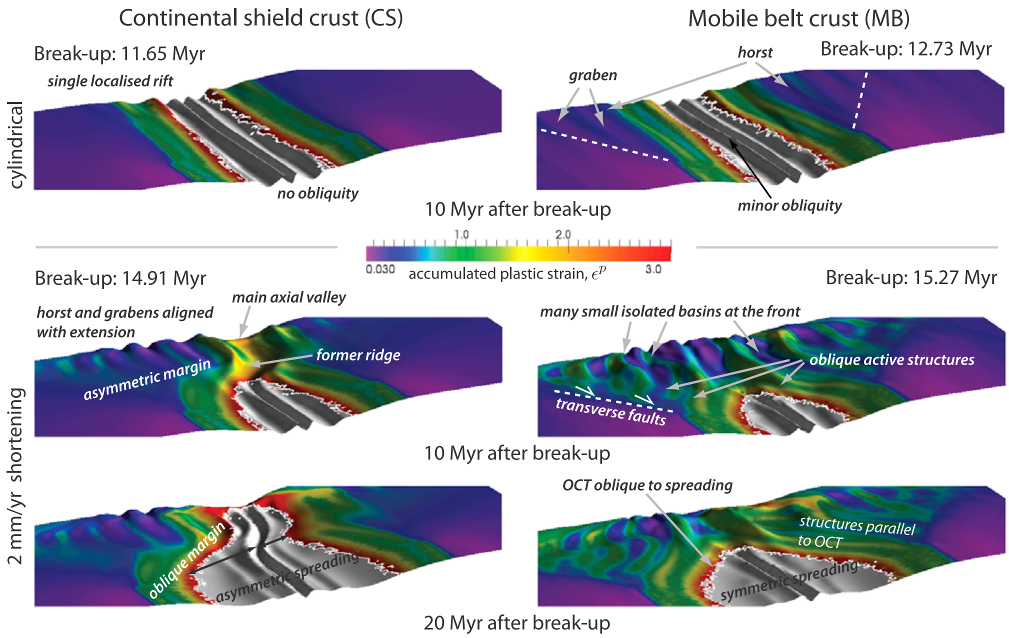

Fig. 10. Accumulated plastic strain (ε^p) outlining the passive margins structures which are represented on the deformed free surface of the models for the 3D rifting experiments. Gray area represents oceanic regions where the mantle has been exhumed to the surface. The white contour line denotes the Ocean–Continent Transition (OCT). For clarity, the topography shown has been exaggerated by a factor of five.

9.3. *Discussion*

In the following, we will regard the initial onset of continental break-up to have occurred at the moment the continental lithosphere is separated by "oceanic" spreading material at the surface at the level of the damaged zone (white contour lines in Fig. 10). Since we are interested in the timing of propagation, after this first event, all times reported are given with respect to the onset of initial continental break-up.

Comparing how continental break-up propagates through continental shield (CS) or mobile belt (MB) lithosphere using cylindrical boundary conditions, we observe no major differences in the rift geometry after break-up has occurred. In both cases, break-up propagates through the model domain in 5.5 Myr and 10 Myr after the initial break up (see upper two panels of Fig. 10). We find that the oceanic seafloor width ranges from 190 km at the front of the domain, to 100 km at the back ($z = 600$ km) of the model.

The major difference between the two models relates to the distribution of accumulated strain before break-up. In the MB case (Fig. 10, top right), a wide rift outlined by the concentration of plastic strain has developed at the front of the triggered continental rift, forming several horsts (topographic highs) and grabens (topographic lows). In the CS case (Fig. 10, top left), there is no evidence of horst and graben like topographic features, except for the main narrow rift. The observed differences in the mode of continental rifting in response to the rheology of the lower crust are consistent with previous 1D and 2D thermo-mechanical models [85,81].

In the MB case, the initial wide rift affecting the continental crust has led to a slight offset (in the x direction) between the triggered rift, and the continental grabens which form at the rear of the damaged zone before the initial continental break up. As a result, a slight obliquity in the spreading regime exists at ∼10 Myr after the initial break-up which is not observed in the CS case. Wide rift deformation seems therefore to favor slight obliquity in rifting and spreading when cylindrical boundary conditions are applied.

In the two models which have a slight component of normal shortening imposed normal to main extension direction (Fig. 10 lower four panels), the average rate of thinning of the crust is reduced by ∼20% and, as a result, the initial break-up is delayed by ∼3 Myr. This behavior was already noted by van Wijk and Blackman [82], however their models had a predefined 3D geometry and thus could not be used to investigate the changing geometry of the rift, or to induce obliquity.

Both models are seen to develop oblique structures, however the manner in which the obliquity is accommodated is distinctly different. In the model employing a continental shield lithosphere (strong lower crust), the obliquity is accommodated by the development of an offset mid-oceanic ridge, whereas in the mobile belt lithosphere model (weak lower crust) the obliquity it is accommodated by oblique normal faults and transverse relay faults on the passive margin.

In the CS model, spreading propagates rapidly during a first phase of 4 Myr after the initial break-up. From 4 to 11 Myr, the propagation is very slow and spreading occurs obliquely because it is attracted by the offset axial valley, which at this stage has not yet undergone break-up. At 11 Myr after the initial break-up, spreading starts in the offset axial valley and at 13 Myr, the two ridges merge together breaking the topographic ridge that previously separated them (Fig. 10 left central panel).

Examining the MB model, we observe that in places where offset ridges form, the fault structures on the passive margins are normal to the direction of extension and oblique to the Ocean Continent Transition (OCT). In this model, continental break-up stalls for more than 20 Myr after the initial break-up. As a result, the rift structures rotate, whilst remaining parallel to the OCT and spreading remains symmetrical. The rotation of the structure due to spreading largely reduces the extension rate at the front of the rift, giving rise to a strike slip regime outlined by the obliquity of the active structures (Fig. 10 right central panel).

Our models confirm that a weak lower crust (mobile belt) favor wider passive margins and that a quiescent period of propagation of continental rifting can be induced by shortening in the direction normal to the ridge. These models show, for the first time, that a very small amount of axial shortening also induces obliquity and that the accommodation of this obliquity by the mid-oceanic ridge, or by the continental structures is a strong function of the viscosity of the lower crust. A weak lower crust favors margins that are oblique to spreading, whilst a strong lower crust favor ridge jumps and transform margins. More work is needed to fully explore these regimes, and in particular, to test the influence of the mantle lithosphere buoyancy and rheology e.g. [86,87] on the dynamics and timing of propagation.

10. Summary

We have presented a computationally efficient strategy for solving highly variable viscosity Stokes problems on logically structured hexahedral meshes using a hybrid material-point, finite element spatial discretization. The method has been specifically developed to enable high resolution, three-dimensional simulations of processes related to the dynamics of Earth's lithosphere.

Table 9
Memory bandwidth (GB/s), peak performance (GF/s) and arithmetic intensity (F/B) for various processors, as measured using the STREAM Triad benchmark [55,56].

Processor	Bandwidth (GB/s)	Peak (GF/s)	Balance (F/B)
E5-2680 8-core	38	173	4.5
E5-2695v2 12-core	45	230	5.2
Magny Cours 16-core	49	281	5.7
Blue Gene/Q node	29.3	205	7
Tesla M2090	120	665	5.5
Kepler K20Xm	160	1310	8.2
Xeon Phi SE10P	161	1060	6.6
E5-2699v3 18-core	60	660	11

To obtain fast, scalable and robust solutions of the discrete Stokes problem, we have utilized efficient matrix-free operators and a flexible multigrid preconditioner. The underlying philosophy of the preconditioners design is to defer as many decisions related to the definition of smoother, coarse level solver, coarsening factor and type of coarse level operator to run-time. This degree of flexibility enables enormous “solver tuning” potential for each new geodynamic application.

The ability to define different coarse level operators on each level within the multigrid hierarchy was demonstrated to be particularly advantageous. We showed that using a Galerkin operator on only the coarsest level can provide significant acceleration of the multigrid preconditioner due to its ability to capture sub-element scale viscosity structures. Other factors we examined to improve the parallel scalability of the variable viscosity multigrid preconditioner included the use of Chebyshev-based smoothers, and nested Krylov coarse level solvers. On some architectures, direct application of such coarse level solvers is likely to represent a computational bottleneck when using $>10-30$k cores. To assist in maintaining the necessary robustness and scalability of the coarse level solver, we also explored using algebraic multigrid and a re-partitioned coarse mesh scheme. The re-partitioned approach was found to be the fastest alternative we examined, and provided the best strong scaling efficiency. We currently do not have a performance model to guide how frequently, or how aggressively processor coarsening should be performed within our re-partitioned approach. Developing automatic procedures to enable this is an on going effort.

Lastly, we demonstrated the potential of the hybrid discretization and Stokes preconditioner by examining a complex geodynamic problem associated with continental rifting and break-up. For the geodynamic application considered, the scientific throughput we obtained indicates that it is now possible to utilize Q_2–P_1^{disc} finite elements to solve high resolution 3D, nonlinear, time-dependent lithospheric dynamics models with modest computational resources.

Whilst we have utilized an example from regional scale lithospheric dynamics to demonstrate the new methodology, we emphasize that the approach described is completely applicable to global scale geodynamic processes such as the dynamics of subduction and mantle convection.

Acknowledgments

Ritske Huismans is thanked for CPU time on the Cray XE6 “hexagon” hosted by the Bergen Centre for Computational Science (BCCS), and supported by the Notur — the Norwegian Metacenter for Computational Science, which was used for the multi-core results presented in Sections 7 and 8. Philippe de Clarens at TOTAL is acknowledged for time on the SGI-ICE “Rostand” which was used for the simulations reported in Section 9. Caroline Baldassari (SGI) is thanked for helping install the code on “Rostand”. Paul J. Tackley and Taras V. Gerya are thanked for commenting on an early version of this manuscript. Constructive comments and criticism from two anonymous reviewers were appreciated and helped to improve the manuscript. JB was supported by Swiss National Science Foundation Grant 200021-113503/1 and the U.S. Department of Energy’s Office of Science under Contract DE-AC02-06CH11357.

Appendix A. Measured performance on modern hardware

See Table 9.

Table 10

Performance of different coarse level solver configurations for the sinker problem with $n_c = 1$, $\Delta\eta = 10^6$. Columns denoting iterations (A its.) which contain “F” indicate the solver failed to converge in 800 iterations, or if 10 min elapsed. See Section 8.3 for additional details.

Solver configuration	Solver ID	Cores	A its.	Local solve (s)	SpMV (s)	Solve (s)
FGMRES+ASM(1)/ILU(1)	F1	4096	F	–	–	–
FGMRES+ASM(1)/ILU(3)	F2	4096	F	–	–	–
FGMRES+ASM(2)/ILU(3)	F3	4096	F	–	–	–
FGMRES+ASM(3)/ILU(3)	F4	4096	F	–	–	–
Cheby+ASM(2)/ILU(3)	C1	64	125	2.7431e+01	9.9491e−01	2.1562e+02
Cheby+ASM(2)/ILU(3)	C2	4096	357	1.1863e+01	6.1683e−01	1.4036e+01
FGMRES+[Cheby(30)+BJacobi/ILU(3)]	FNC1	4096	298	1.1355e−01	3.6913e+00	3.9424e+00
FGMRES+[Cheby(10)+ASM(1)/ICC(3)]	FNC2	4096	110	9.2993e−01	9.7277e−01	1.7574e+00
FGMRES+[Cheby(10)+ASM(2)/ICC(3)]	FNC3	4096	106	2.2058e+01	1.6593e+01	2.4100e+01
FNC1 +modified Gram–Schmidt Orthog.	FNC1*	4096	300	1.1622e−01	3.6699e+00	4.2662e+00
FNC2 +modified Gram–Schmidt Orthog.	FNC2*	4096	42	3.5611e−01	3.7599e−01	7.3459e−01
FNC3 +modified Gram–Schmidt Orthog.	FNC3*	4096	22	4.5877e+00	3.4344e+00	5.9626e+00
	FNC2*	8	19	6.6884e+01	2.0700e+01	9.6883e+01
	FNC2*	64	19	1.1278e+01	6.4829e+00	1.5866e+01
	FNC2*	512	27	2.1224e+00	1.7332e+00	2.7305e+00
	FNC2*	4096	42	3.5611e−01	3.7599e−01	7.3459e−01

Appendix B. Evaluation of coarse level solvers using preconditioned Krylov methods

See Table 10.

References

[1] F.A. Dahlen, Critical taper model of fold-and-thrust belts and accretionary wedges, Annu. Rev. Earth Planet. Sci. 18 (1990) 55.

[2] F.O. Marques, P.R. Cobbold, Topography as a major factor in the development of arcuate thrust belts: insights from sandbox experiments, Tectonophysics 348 (4) (2002) 247–268.

[3] E. Burov, A. Poliakov, Erosion and rheology controls on synrift and postrift evolution: Verifying old and new ideas using a fully coupled numerical model, J. Geophys. Res. Solid Earth 106 (B8) (2001) 16461–16481.

[4] B.J.P. Kaus, H. Mühlhaus, D.A. May, A stabilization algorithm for geodynamic numerical simulations with a free surface, Phys. Earth Planet. Interiors 181 (1–2) (2010) 12–20.

[5] F. Crameri, P.J. Tackley, I. Meilick, t.V. Gerya, B.J.P. Kaus, A free plate surface and weak oceanic crust produce single-sided subduction on Earth, Geophys. Res. Lett. 39 (3) (2012).

[6] F.H. Harlow, E. Welch, Numerical calculation of time-dependent viscous flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.

[7] D. Sulsky, Z. Chen, H.L. Schreyer, A particle method for history-dependent materials, Numer. Meth. Appl. Mech. Engrg. 118 (1994) 179–196.

[8] R.F. Weinberg, H. Schmeling, Polydiapirs: Multiwave length gravity structures, J. Struct. Geol. 14 (1992) 425–436.

[9] S. Zaleski, P. Julien, Numerical simulation of Rayleigh-Taylor instability for single and multiple salt diapirs, Tectonophysics 206 (1–2) (1992) 55–69.

[10] A. Poliakov, Y. Podladchikov, Diapirism and topography, Geophys. J. Int. 109 (1992) 553–564.

[11] P. Fullsack, An arbitrary Lagrangian-Eulerian formulation for creeping flows and its application in tectonic models, Geophys. J. Int. 120 (1995) 1–23.

[12] P.E. van Keken, S.D. King, H. Schmeling, U.R. Christensen, D. Neumeister, M.-P. Doin, A comparison of methods for the modeling of thermochemical convection, J. Geophys. Res. 102 (B10) (1997) 22477–22496.

[13] A.Y. Babeyko, S.V. Sobolev, R.B. Trumbull, O. Oncken, L.L. Lavier, Numerical models of crustal-scale convection and partial melting beneath the Altiplano-Puna plateau, Earth Planet. Sci. Lett. 199 (2002) 373–388.

[14] T.V. Gerya, D.A. Yuen, Charaterictics-based marker method with conservative finite-difference schemes for modeling geological flows with strongly variable transport properties, Phys. Earth Planet. Interiors 140 (4) (2003) 293–318.

[15] L. Moresi, F. Dufour, H.-B. Mühlhaus, A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials, J. Comput. Phys. 184 (2003) 476–497.

[16] S.J.H. Buiter, A.Y. Babeyko, S. Ellis, T.V. Gerya, B.J.P. Kaus, A. Kellner, G. Schreurs, Y. Yamada, The numerical sandbox: Comparison of model results for a shortening and an extension experiment, in: Analogue and Numerical Modelling of Crustal-Scale Processes, in: Special Publication - Geological Society of London, vol. 253, Geological Society of London, London, 2006, pp. 29–64.

[17] T.V. Gerya, D.A. Yuen, Robust characteristics method for modelling multiphase visco-elasto-plastic thermo-mechanical problems, Phys. Earth Planet. Interiors 163 (2007) 83–105.

[18] L. Moresi, S. Quenette, V. Lemiale, C. Meriaux, B. Appelbe, H.B. Mühlhaus, Computational approaches to studying non-linear dynamics of the crust and mantle, Phys. Earth Planet. Interiors 163 (2007) 69–82.

[19] A.A. Popov, S.V. Sobolev, SLIM3D: A tool for three-dimensional thermomechanical modeling of lithospheric deformation with elasto-visco-plastic rheology, Phys. Earth Planet. Interiors 171 (1–4) (2008) 55–75. Recent Advances in Computational Geodynamics: Theory, Numerics and Applications.

[20] H. Schmeling, A.Y. Babeyko, A. Enns, C. Faccenna, F. Funiciello, T.V. Gerya, G.J. Golabek, S. Grigull, B.J.P. Kaus, G. Morra, S.M. Schmalholz, J. van Hunen, A benchmark comparison of spontaneous subduction models—Towards a free surface, Phys. Earth Planet. Interiors 171 (1–4) (2008) 198–223.

[21] M.E.T. Quinquis, S.J.H. Buiter, S. Ellis, The role of boundary conditions in numerical models of subduction zone dynamics, Tectonophysics 497 (1–4) (2011) 57–70.

[22] C. Thieulot, FANTOM: Two- and three-dimensional numerical modelling of creeping flows for the solution of geological problems, Phys. Earth Planet. Interiors 188 (1–2) (2011) 47–68.

[23] S.M. Lechmann, D.A. May, B.J.P. Kaus, S.M. Schmalholz, Comparing thin-sheet models with 3-D multilayer models for continental collision. Geophys, J. Int. 187 (1) (2011) 10–33.

[24] C.R. Dohrmann, P.B. Bochev, A stabilized finite element method for the Stokes problem based on polynomial pressure projections, Int. J. Numer. Meth. Engrg. 46 (2) (2004) 183–201.

[25] W. Landry, L. Hodkinson, S. Kientz, GALE User Manual, Tech. Rep. Version 2.0.1, Computational Infrastructure for Geodynamics, 2012. http://www.geodynamics.org/cig/software/gale/gale.pdf.

[26] C. Burstedde, O. Ghattas, G. Stadler, T. Tu, L.C. Wilcox, Parallel scalable adjoint-based adaptive solution of variable-viscosity Stokes flow problems, Comput. Methods Appl. Mech. Engrg. 198 (2009) 1691–1700.

[27] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer-Verlag, New York, Inc., New York, NY, USA, 1991.

[28] J. Braun, C. Thieulot, P. Fullsack, M. DeKool, C. Beaumont, R. Huismans, DOUAR: A new three-dimensional creeping flow numerical model for the solution of geological problems, Phys. Earth Planet. Interiors 171 (1) (2008) 76–91.

[29] A. George, J.W. Liu, Computer Solution of Large Sparse Positive Definite Systems, Prentice-Hall, 1981.

[30] D.A. May, L. Moresi, Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics, Phys. Earth Planet. Interiors 171 (1) (2008) 33–47.

[31] T. Geenen, M. ur Rehman, S.P. MacLachlan, G. Segal, C. Vuik, A.P. van den Berg, W. Spakman, Scalable robust solvers for unstructured FE geodynamic modeling applications: Solving the Stokes equation for models with large localized viscosity contrasts, Geochem. Geophys. Geosyst. 10 (9) (2009).

[32] P. Grinevich, M. Olshanskii, An iterative method for the Stokes-type problem with variable viscosity, SIAM J. Sci. Comput. 31 (5) (2009) 3959–3978.

[33] L.L. Lavier, R.W. Buck, A.N.B. Poliakov, Self-consistent rolling-hinge model for the evolution of large-offset low-angle normal faults, Geology 27 (12) (1999) 1127–1130.

[34] V. Lemiale, H.-B. Mühlhaus, L. Moresi, J. Stafford, Shear banding analysis of plastic models formulated for incompressible viscous flows, Phys. Earth Planet. Interiors 171 (1) (2008) 177–186.

[35] D. Boffi, L. Gastaldi, On the quadrilateral Q_2-P_1 element for the Stokes problem, Int. J. Numer. Meth. Fluids 39 (11) (2002) 1001–1011.

[36] G. Matthies, L. Tobiska, The Inf-Sup condition for the mapped Q_k-P_{k-1}^{disc} element in arbitrary space dimensions, Computing 69 (2) (2002) 119–139.

[37] P.B. Bochev, C.R. Dohrmann, M.D. Gunzburger, Stabilization of low-order mixed finite elements for the Stokes equations, SIAM J. Numer. Anal. 44 (1) (2006) 82–101.

[38] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2) (2004) 357–397.

[39] S.C. Eisenstat, H.F. Walker, Globally convergent inexact Newton methods, SIAM J. Optim. 4 (2) (1994) 393–422.

[40] S.C. Eisenstat, H.F. Walker, Choosing the forcing terms in an inexact Newton method, SIAM J. Sci. Comput. 17 (1) (1996) 16–32.

[41] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (2) (1993) 461–469.

[42] S.C. Eisenstat, H.C. Elman, M.H. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, SIAM J. Numer. Anal. 20 (2) (1983) 345–357.

[43] H.C. Elman, D.J. Silvester, A.J. Wathen, Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations, Numer. Math. 90 (4) (2002) 665–688.

[44] H. Elman, D. Silvester, A. Wathan, Finite Elements and Fast Iterative Solvers, Oxford Univ. Press, New York, 2005.

[45] M. Murphy, G. Golub, A. Wathen, A note on preconditioning for indefinite linear systems, SIAM J. Sci. Comput. 21 (6) (2000) 1969–1972.

[46] M.F. Adams, M. Brezina, J.J. Hu, R.S. Tuminaro, Parallel multigrid smoothing: polynomial versus Gauss-Seidel, J. Comput. Phys. 188 (2) (2003) 593–610.

[47] J.J. Hu, R.S. Tuminaro, P.B. Bochev, C.J. Garasi, A.C. Robinson, Toward an h-independent algebraic multigrid method for Maxwell's equations, SIAM J. Sci. Comput. 27 (5) (2005) 1669–1688.

[48] P. Arbenz, G.H. van Lenthe, U. Mennel, R. Müller, M. Sala, A scalable multi-level preconditioner for matrix-free μ-finite element analysis of human bone structures, Int. J. Numer. Meth. Engrg. 73 (7) (2008) 927–947.

[49] P. Bochev, K. Peterson, C. Siefert, Analysis and computation of compatible least-squares methods for div–curl equations, SIAM J. Numer. Anal. 49 (1) (2011) 159–181.

[50] A.H. Baker, R.D. Falgout, T. Gamblin, T.V. Kolev, M. Schulz, U.M. Yang, Scaling algebraic multigrid solvers: On the road to exascale, in: C. Bischof, H.-G. Hegering, W.E. Nagel, G. Wittum (Eds.), Competence in High Performance Computing 2010, Springer, Berlin, Heidelberg, 2012, pp. 215–226.

[51] T. Jönsthövel, M. Gijzen, S. MacLachlan, C. Vuik, A. Scarpas, Comparison of the deflated preconditioned conjugate gradient method and algebraic multigrid for composite materials, Comput. Mech. 50 (3) (2012) 321–333.

[52] S. Balay, J. Brown, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, 2013b PETSc Web page. http://www.mcs.anl.gov/petsc.

[53] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Users Manual. Tech. Rep. ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.

[54] J. Brown, Efficient nonlinear solvers for nodal high-order finite elements in 3D, J. Sci. Comput. 45 (2010) 48–63.

[55] J.D. McCalpin, Memory bandwidth and machine balance in current high performance computers. in: IEEE Computer Society Technical Committee on Computer Architecture, TCCA, Newsletter, 1995, pp. 19–25.

[56] J.D. McCalpin, STREAM: Sustainable Memory Bandwidth in High Performance Computers. Tech. Rep., University of Virginia, Charlottesville, Virginia, a continually updated technical report, 1991–2007. http://www.cs.virginia.edu/stream/.

[57] W.D. Gropp, D.K. Kaushik, D.E. Keyes, B.F. Smith, Performance modeling and tuning of an unstructured mesh CFD application, in: Proceedings of the SC2000, IEEE Computer Society, 2000, p. 34.

[58] B. Smith, H. Zhang, Sparse triangular solves for ILU revisited: Data layout crucial to better performance, Int. J. High Perform. Comput. Appl. 25 (4) (2011) 386–391.

[59] D.A. May, J. Brown, L. Le Pourhiet, pTatin3d: High-performance methods for long-term lithospheric dynamics, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC'14, IEEE Press, 2014, pp. 274–284.

[60] J. Suckale, B.H. Hager, L.T. Elkins-Tanton, J.-C. Nave, It takes three to tango: 2. Bubble dynamics in basaltic volcanoes and ramifications for modeling normal Strombolian activity, J. Geophys. Res. 115 (B7) (2010).

[61] P. Yamato, R. Tartese, T. Duretz, D.A. May, Numerical modelling of magma transport in dykes, Tectonophysics 526 (2012) 97–109.

[62] T.A. Davis, Algorithm 832: UMFPACK V4.3, An unsymmetric-pattern multifrontal method, ACM Trans. Math. Softw. 30 (2) (2004) 196–199.

[63] P.R. Amestoy, I.S. Duff, J. Koster, J.-Y. L'Excellent, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM J. Matrix Anal. Appl. 23 (1) (2001) 15–41.

[64] X.S. Li, An overview of SuperLU: Algorithms, implementation, and user interface, ACM Trans. Math. Softw. 31 (3) (2005) 302–325.

[65] L.C. McInnes, B. Smith, H. Zhang, R.T. Mills, Hierarchical Krylov and nested Krylov methods for extreme-scale computing, Parallel Comput. 40 (1) (2014) 17–31.

[66] V. Courtillot, Propagating rifts and continental breakup, Tectonics 1 (3) (1982) 239–250.

[67] S. Brune, J. Autin, The rift to break-up evolution of the Gulf of Aden: Insights from 3D numerical lithospheric-scale modelling, Tectonophysics 607 (2013) 65–79.

[68] T.V. Gerya, Initiation of transform faults at rifted continental margins: 3D petrological-thermomechanical modeling and comparison to the Woodlark Basin, Petrology 21 (6) (2013) 550–560.

[69] T.V. Gerya, Three-dimensional thermomechanical modeling of oceanic spreading initiation and evolution, Phys. Earth Planet. Interiors 214 (2013) 35–52.

[70] S. Brune, Evolution of stress and fault patterns in oblique rift systems: 3-D numerical lithospheric-scale experiments from rift to breakup, Geochem. Geophys. Geosyst. 15 (8) (2014) 3392–3415.

[71] C. Heine, S. Brune, Oblique rifting of the Equatorial Atlantic: Why there is no Saharan Atlantic Ocean, Geology 42 (3) (2014) 211–214.

[72] H. Koopmann, S. Brune, D. Franke, S. Breuer, Linking rift propagation barriers to excess magmatism at volcanic rifted margins, Geology (2014) G36085–1.

[73] S. Brune, A.A. Popov, S.V. Sobolev, Modeling suggests that oblique extension facilitates rifting and continental break-up, J. Geophys. Res. Solid Earth 117 (B8) (2012).

[74] L. Le Pourhiet, B. Huet, D.A. May, L. Labrousse, L. Jolivet, Kinematic interpretation of the 3D shapes of metamorphic core complexes, Geochem. Geophys. Geosyst. 13 (9) (2012) Q09002.

[75] V. Allken, R.S. Huismans, C. Thieulot, Factors controlling the mode of rift interaction in brittle-ductile coupled systems: A 3D numerical study, Geochem. Geophys. Geosyst. 13 (5) (2012).

[76] V. Allken, R.S. Huismans, H. Fossen, C. Thieulot, 3D numerical modelling of graben interaction and linkage: a case study of the Canyonlands grabens, Utah. Basin Res. 25 (4) (2013) 436–449.

[77] L. Le Pourhiet, B. Huet, N. Traoré, Links between long-term and short-term rheology of the lithosphere: Insights from strike-slip fault modelling, Tectonophysics 631 (2014) 146–159.

[78] E. Burov, T. Gerya, Asymmetric three-dimensional topography over mantle plumes, Nature 513 (7516) (2014) 85–89.

[79] E. Choi, L. Lavier, M. Gurnis, Thermomechanics of mid-ocean ridge segmentation, Phys. Earth Planet. Interiors 171 (1) (2008) 374–386.

[80] T. Gerya, Dynamical instability produces transform faults at mid-ocean ridges, Science 329 (5995) (2010) 1047–1050.

[81] J. Braun, C. Beaumont, Dynamical models of the role of crustal shear zones in asymmetric continental extension, Earth Planet. Sci. Lett. 93 (3) (1989) 405–423.

[82] J.W. Van Wijk, D.K. Blackman, Dynamics of continental rift propagation: the end-member modes, Earth Planet. Sci. Lett. 229 (3) (2005) 247–258.

[83] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1) (1982) 199–259.

[84] B. Parsons, J.G. Sclater, An analysis of the variation of ocean floor bathymetry and heat flow with age, J. Geophys. Res. 82 (5) (1977) 803–827.

[85] W.R. Buck, Modes of continental lithospheric extension, J. Geophys. Res. Solid Earth 96 (B12) (1991) 20161–20178.

[86] R.S. Huismans, Y.Y. Podladchikov, S. Cloetingh, Transition from passive to active rifting: Relative importance of asthenospheric doming and passive extension of the lithosphere, J. Geophys. Res. 106 (B6) (2001) 11271–11291.

[87] L. Watremez, E. Burov, E. d'Acremont, S. Leroy, B. Huet, L. Le Pourhiet, N. Bellahsen, Buoyancy and localizing properties of continental mantle lithosphere: Insights from thermomechanical models of the eastern Gulf of Aden, Geochem. Geophys. Geosyst. (2013).