AGU PUBLICATIONS

Geochemistry, Geophysics, Geosystems

TECHNICAL REPORTS: METHODS

10.1002/2016GC006727

Key Points:

- Efficient solvers for two-dimensional linear and power law viscous Stokes flow
- Codes are compared against analytical and benchmarked solutions
- We provide transparent MATLAB source codes for full reproducibility

Supporting Information:

- Software S1–S14

Correspondence to:
L. Räss,
ludovic.raess@unil.ch

Citation:
Räss, L., T. Duretz, Y. Y. Podladchikov, and S. M. Schmalholz (2017), M2Di: Concise and efficient MATLAB 2-D Stokes solvers using the Finite Difference Method, *Geochem. Geophys. Geosyst.*, *18*, 755–768, doi:10.1002/2016GC006727.

Received 8 NOV 2016
Accepted 13 JAN 2017
Accepted article online 20 JAN 2017
Published online 23 FEB 2017

M2Di: Concise and efficient MATLAB 2-D Stokes solvers using the Finite Difference Method

Ludovic Räss[1], Thibault Duretz[1], Yury Y. Podladchikov[1], and Stefan M. Schmalholz[1]

[1]Faculté des géosciences et de l'environnement, Institut des Sciences de la Terre, University of Lausanne, Lausanne, Switzerland

Abstract Recent development of many multiphysics modeling tools reflects the currently growing interest for studying coupled processes in Earth Sciences. The core of such tools should rely on fast and robust mechanical solvers. Here we provide M2Di, a set of routines for 2-D linear and power law incompressible viscous flow based on Finite Difference discretizations. The 2-D codes are written in a concise vectorized MATLAB fashion and can achieve a time to solution of 22 s for linear viscous flow on 1000^2 grid points using a standard personal computer. We provide application examples spanning from finely resolved crystal-melt dynamics, deformation of heterogeneous power law viscous fluids to instantaneous models of mantle flow in cylindrical coordinates. The routines are validated against analytical solution for linear viscous flow with highly variable viscosity and compared against analytical and numerical solutions of power law viscous folding and necking. In the power law case, both Picard and Newton iterations schemes are implemented. For linear Stokes flow and Picard linearization, the discretization results in symmetric positive-definite matrix operators on Cartesian grids with either regular or variable grid spacing allowing for an optimized solving procedure. For Newton linearization, the matrix operator is no longer symmetric and an adequate solving procedure is provided. The reported performance of linear and power law Stokes flow is finally analyzed in terms of wall time. All MATLAB codes are provided and can readily be used for educational as well as research purposes. The M2Di routines are available from Bitbucket and the University of Lausanne Scientific Computing Group website, and are also supplementary material to this article.

1. Introduction

Efficient and concise numerical algorithms are well-suited tools to study physical processes in Earth Sciences, enhancing clarity and portability without lacking in performance. Such routines can be used to independently explore specific processes, and turn out to be important building blocks that can be employed to further address coupled processes involving multiphysics. These coupled processes are mainly grouped into four major categories, which are referred as Thermo-Hydro-Chemico-Mechanical (THCM) processes [e.g., *Bea et al*., 2015; *Weatherley and Katz*, 2012; *Rutqvist*, 2011; *Rutqvist et al*., 2006]. While THC processes can be represented by scalar Poisson equations, mechanical processes involve vectorial and tensorial components, which are more challenging to take into account. We therefore investigate an efficient and accurate method using the Finite Difference approach to solve the mechanical problem (M) for linear and nonlinear rheologies in two-dimensions (2-D). The proposed framework will be extended in a future step to couple the mechanics to diffusion-like processes (THC) within powerful predictive tools.

With M2Di (Mechanics 2-D iterative), we provide here a simple but efficient framework to solve linear and power law viscous Stokes flow with Picard and Newton linearizations. While existing tools can efficiently model these processes in three-dimensions using high-performance computing [*May et al*., 2014; *Kaus et al*., 2016], our motivation is to deliver a set of concise and performant routines for research application that may also be used for educational purposes. The M2Di routines allow for simulating 2-D steady state flow at high-resolution within a minimal amount of time on modern personal computers. We focus on Finite Difference discretizations using a direct solver strategy on shared memory devices. In contrast to body-fitting Finite Element discretizations, Finite Difference discretizations have a limited geometrical flexibility [*Deubelbeiss and Kaus*, 2008; *Schmeling et al*., 2008]. It was however shown that Finite Difference discretizations are successful for modeling large deformations of heterogeneous materials [*Gerya and Yuen*, 2003, 2007; *Yamato et al.*, 2012a]. High numerical resolution is needed to overcome the discretization errors of the staggered grid, which is feasible with current hardware. Moreover, the simplicity of the Cartesian staggered grid discretization makes it extremely well suited for educational purposes.

Here, we describe MATLAB-based implementations of Stokes solvers on Cartesian grids (regular and variable grid spacing) and cylindrical staggered grids with emphasis on linear and power law viscous constitutive equations. The M2Di algorithms are optimized to take advantage of MATLAB-vectorized operations, Cholesky factorization for the direct solve [*Dabrowski et al.*, 2008], and Newton iterations for power law viscous flow. The M2Di routines were validated against the analytical solution of *Schmid and Podladchikov* [2003] for linear viscous flow with sharp viscosity contrast, and against analytic and numerical solutions computed with Folder [*Adamuszek et al.*, 2016]. Performance and calculation time are reported using the wall time metric. The hardware used within this study is a MacBook Pro hosting a dual-core Intel Core i7 @ 3.1 GHz processor with 16 GB memory (RAM) and running MATLAB R2016a. The MATLAB codes are available in the supporting information for full reproducibility. The M2Di routines are also available for download from Bitbucket at https://bitbucket.org/lraess/m2di and from the University of Lausanne Scientific Computing Group website www.unil.ch/geocomputing/tools/software/m2di/.

2. Methods

2.1. Mathematical Model

The incompressible Stokes equations in 2-D Cartesian coordinates are:

$$\begin{aligned}\frac{\partial\sigma_{xx}}{\partial x}+\frac{\partial\sigma_{xy}}{\partial y}&=0,\\ \frac{\partial\sigma_{yx}}{\partial x}+\frac{\partial\sigma_{yy}}{\partial y}&=-\rho g,\\ \frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}&=0,\end{aligned}\tag{1}$$

where x and y are the spatial coordinates, v_x and v_y correspond to the components of the velocity vector, p is the pressure, ρ is the density and g is the vertical gravity acceleration component. σ_{xx}, σ_{yy}, and $\sigma_{xy}=\sigma_{yx}$ are the components of the total stress tensor and are:

$$\begin{aligned}\sigma_{xx}&=-P+2\eta\frac{\partial v_x}{\partial x},\\ \sigma_{yy}&=-P+2\eta\frac{\partial v_y}{\partial y},\\ \sigma_{xy}&=\eta\left(\frac{\partial v_x}{\partial y}+\frac{\partial v_y}{\partial x}\right),\end{aligned}\tag{2}$$

where η stands for the dynamic fluid viscosity. A solution of the Stokes problem is achieved on the domain Γ for the given set of boundary conditions:

$$\begin{aligned}v_i&=v_i^{\mathrm{BC}}\text{ on }\Gamma_{\mathrm{Dirichlet}}\\ \sigma_{ij}n_j&=T_i^{\mathrm{BC}}\text{ on }\Gamma_{\mathrm{Neumann}},\end{aligned}\tag{3}$$

where n_j is the unit outward normal vector to the domain boundary; v_i^{BC} and T_i^{BC} stand for velocity and traction vectors boundary values, applied on corresponding non-overlapping parts of the domain boundary ($\Gamma_{\mathrm{Dirichlet}}$, $\Gamma_{\mathrm{Neumann}}$). If no pressure boundary condition is supplemented (e.g., free surface), the pressure null space causes the pressure field to be defined up to a constant value.

2.1.1. Linear and Power Law Viscous Stokes Flow

Newtonian, or linear viscous, shear rheology is used to describe systems for which the stresses linearly depend on the strain rates, the viscosity being the factor of proportionality. Even though the viscosity can be highly variable in space, its values do not depend on the strain rate.

In a power law viscous model, stresses exhibit a nonlinear dependence on the magnitude of strain rate. We quantify the strain rate ($\dot{\epsilon}_{II}$) with the square root of the second invariant of the deviatoric strain rate tensor:

$$\dot{\epsilon}_{II}=\sqrt{\frac{1}{2}\left(\dot{\epsilon}_{xx}{}^{2}+\dot{\epsilon}_{yy}{}^{2}\right)+\dot{\epsilon}_{xy}{}^{2}}, \tag{4}$$

where $\dot{\epsilon}_{xx}=\frac{\partial v_x}{\partial x}$, $\dot{\epsilon}_{yy}=\frac{\partial v_y}{\partial y}$ and $\dot{\epsilon}_{xy}=\frac{1}{2}\left(\frac{\partial v_x}{\partial y}+\frac{\partial v_y}{\partial x}\right)$ are, respectively, the x and y normal and the shear strain rates. The power law viscous rheological model is of particular interest for geodynamic modeling since it can be used to model numerous ductile creep mechanisms (e.g., diffusion, dislocation, and exponential creep) [see *Montési and Zuber*, 2002; *Schmalholz and Fletcher*, 2011]. The effective viscosity takes the form of:

$$\eta_{\mathrm{pwl}}=\eta_{\mathrm{ref}}\left(\frac{\dot{\epsilon}_{II}}{\dot{\epsilon}_{\mathrm{ref}}}\right)^{\frac{1}{n}-1}, \tag{5}$$

where n and η_{ref} corresponds to the stress exponent and the reference viscosity evaluated for a reference bulk strain rate $\dot{\epsilon}_{\mathrm{ref}}$, respectively. The power law model, however, can lead to unrealistically low or large viscosities in the limits of infinite and zero strain rates. It is hence convenient to employ constitutive models that allow for power law behavior over a limited range of strain rate magnitudes.

The Carreau fluid model [*Allen*, 1999] offers this type of regularization [e.g., *Adamuszek et al*., 2016] and provides a flow law of the following form:

$$\eta_{\mathrm{Carreau}}=\eta_{\infty}+(\eta_0-\eta_{\infty})\left[1+\left(\xi\frac{\dot{\epsilon}_{II}}{\dot{\epsilon}_{\mathrm{ref}}}\right)^{2}\right]^{\frac{1}{2}\left(\frac{1}{n}-1\right)}, \tag{6}$$

where η_{∞} and η_0 are the values of viscosities for infinite and zeros strain rates. The parameter ξ is:

$$\xi=\left(\frac{\eta_{\mathrm{ref}}}{\eta_0-\eta_{\infty}}\right)^{\frac{1}{\frac{1}{n}-1}}. \tag{7}$$

2.2. Numerical Implementation

The system of partial differential equations describing Stokes flow (1) and (2) can readily be discretized using the Finite Difference Method on a regular Cartesian staggered grid. The staggered grid relies on second-order conservative Finite Differences [*Patankar*, 1980] and is by essence devoid of oscillatory pressure modes [*Shin and Strikwerda*, 1997]. The pressure null space arising from the absence of boundary conditions on the pressure equation can be circumvented by subtracting the mean value of pressure within a segregated solve [*May and Moresi*, 2008].

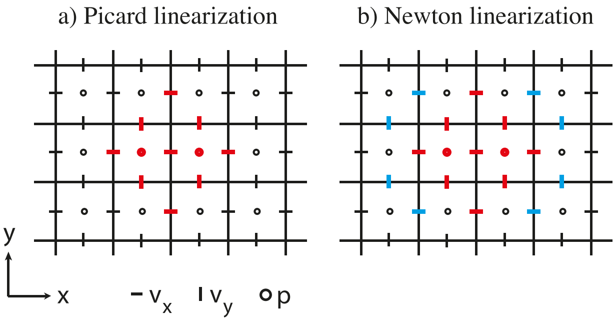

Figure 1. Examples of Finite Difference stencils used for discretizing the x-momentum equation. (a) the standard linear Stokes stencil (Picard linearization) and (b) the stencil arising from the Newton linearization of the Stokes equation with power law rheology. The red symbols outline the degrees of freedom involved in the Picard linearization and the blues symbols correspond to additional degrees of freedom required for the Newton linearization (i.e., stencil growth).

Pressure nodes are defined at cell centers while nodes corresponding to velocity components are defined at cells midfaces. Normal and shear components of the strain rate tensor are not collocated and are computed at cell centers and vertices, respectively. As a result, stress evaluations require viscosity to be defined at both cell centers and vertices [*Gerya and Yuen*, 2003]. In the linear viscous case, the stencil corresponding to the discrete x-momentum Stokes equations is the symmetric Picard linearized operator, composed of 11 entries (Figure 1a). In the power law viscous case, viscosity is a function of the magnitude of the strain rate. It is thus necessary to evaluate shear strain rate tensor components on cell centers and normal strain rate components on vertices. This can be achieved by either averaging neighboring strain rate components or averaging the second invariant contributions [*Kaus et al.*, 2016]. Both implementations were tested and are referred to as "inv 0" and "inv 1", respectively. The corresponding Finite Difference stencil thus requires information from additional neighboring nodes (Figure 1b). As a result, the discrete operator (i.e., stiffness matrix) representing the Newton linearization of Stokes equations (Jacobian) is denser (21 entries) than its linear viscous counterpart (or Picard linearized operator) and asymmetric.

2.2.1. Solving Procedure

The solution of Finite Difference discretization of the incompressible Stokes equations for velocity **u** and pressure **p** as primitive variables can be obtained by solving the following system expressed in defect correction form to conveniently handle linear and nonlinear problems [e.g., *Spiegelman et al.*, 2016]:

$$\underbrace{\begin{bmatrix} \mathbf{M} & \mathbf{G} \\ -\mathbf{G}^T & \mathbf{0} \end{bmatrix}}_{\mathbf{M}_{\mathrm{Stokes}}} \underbrace{\begin{bmatrix} \delta\mathbf{u} \\ \delta\mathbf{p} \end{bmatrix}}_{\delta\mathrm{x}} = -\underbrace{\begin{bmatrix} \mathbf{f}_{\mathrm{u}} \\ \mathbf{f}_{\mathrm{p}} \end{bmatrix}}_{\mathbf{f}}, \tag{8}$$

where **M** either stands for the symmetric positive-definite Picard linearized operator **K** or the asymmetric Jacobian matrix **J** (Newton linearization). $\delta\mathbf{u}$ and $\delta\mathbf{p}$ are the velocity and pressure iterative corrections. The vectors $\mathbf{f}_{\mathrm{u}}$ and $\mathbf{f}_{\mathrm{p}}$ represent the velocity and pressure residuals, which we seek to minimize. These residuals also include the velocity and pressure force vectors $\mathbf{b}_{\mathrm{u}}$ and $\mathbf{b}_{\mathrm{p}}$, that contain the contributions of both body forces and boundary conditions (BC). The coefficients of the **J** matrix were derived analytically (see appended Maple script JacobianStokes.mw). Note that the discrete divergence operator is equivalent to minus the transposed gradient operator $-\mathbf{G}^T$ after applying boundary conditions (see section 2.2.2). For the variable grid spacing case, matrix operators remain symmetric only if equations are scaled by their corresponding control volume (see section 3.3.2). For pressure-independent viscous flow laws, the $-\mathbf{G}^T$ and **G** blocks are not affected by the choice of linearization and symmetry is retained. The global correction vector $\delta\mathbf{x}$ can be obtained via a straightforward direct factorization of the $\mathbf{M}_{\mathrm{Stokes}}$ matrix. Nevertheless, such approach is likely to be not optimal as the $\mathbf{M}_{\mathrm{Stokes}}$ matrix has a missing diagonal for the pressure block. To overcome this, we use as preconditioner **P**, a weakly compressible Stokes formulation, thus adding a numerical bulk modulus γ to the pressure block diagonal resulting in:

$$\mathbf{P} = \begin{bmatrix} \mathbf{M} & \mathbf{G} \\ -\mathbf{G}^T & \gamma^{-1}\mathbf{I} \end{bmatrix}, \tag{9}$$

where the pressure correction $\delta\mathbf{p}$ can be substituted in the momentum equations resulting in the following velocity Schur complement $\tilde{\mathbf{M}} = \left[\mathbf{M} + \mathbf{G}\left(\gamma\mathbf{I}\,\mathbf{G}^T\right)\right]$, where $\gamma\mathbf{I} = \left(\gamma^{-1}\mathbf{I}\right)^{-1}$ is a trivial inverse. Note that the **P** matrix is not symmetric but positive definite; multiplying the second block row by -1 would make the matrix symmetric although indefinite. The velocity and pressure corrections can then be iteratively retrieved within Powell-Hestenes iterations [*Cuvelier et al.*, 1986] by applying the action of $\tilde{\mathbf{M}}^{-1}$, the inverse of the velocity Schur complement to $\tilde{\mathbf{f}}_{\mathrm{u}}$:

$$\delta\mathbf{u} = \tilde{\mathbf{M}}^{-1} \underbrace{\left[\mathbf{f}_{\mathrm{u}} - \mathbf{G}\left(\gamma\mathbf{I}\,\mathbf{f}_{\mathrm{p}} + \delta\mathbf{p}\right)\right]}_{\tilde{\mathbf{f}}_{\mathrm{u}}}, \tag{10}$$

and subsequently computing the pressure corrections:

$$\delta\mathbf{p} = \delta\mathbf{p} + \gamma\mathbf{I}\left(\mathbf{f}_{\mathrm{p}} + \mathbf{G}^T\delta\mathbf{u}\right). \tag{11}$$

The pressure null space is handled by ensuring that pressure corrections have a zero mean $\left(\int_{\Omega}\delta p\,\mathrm{d}V = 0\right)$. In order to ensure incompressibility, the Powell-Hestenes iterations are performed until the amount of flow divergence reaches the tolerance of the linear solver $\max||\mathbf{G}^T\,\delta\mathbf{u}||_{\mathrm{L2}} < \mathrm{tol}_{\mathrm{lin}}$. The solution is subsequently updated using the corresponding velocity and pressure corrections. A global overview of the linear solving strategy is detailed in Algorithm (1), applying to both Picard and Newton schemes.

Algorithm 1 Iterative Powell-Hestenes procedure

```
$\tilde{\mathbf{K}}=\mathbf{K}-\mathbf{G}(\gamma\mathbf{I}\,\mathbf{G}^T)$    ▷ Construct the Stokes Schur Complement
if Newton then
    $\tilde{\mathbf{J}}=\mathbf{J}-\mathbf{G}(\gamma\mathbf{I}\,\mathbf{G}^T)$    ▷ Construct the Jacobian Schur Complement
end if
$[\mathbf{Lc}, e, \mathbf{s}]=\mathrm{chol}(\tilde{\mathbf{K}},'\mathrm{lower}','\mathrm{vector}')$    ▷ Cholesky factorization with permutation vector
$\delta\mathbf{p}=0$
while $(\max||\mathbf{G}^T\,\delta\mathbf{u}||_{L2} < \mathrm{tol}_{\mathrm{lin}})$ do
    $\tilde{\mathbf{f}}_{\mathrm{u}}=\mathbf{f}_{\mathrm{u}}-\mathbf{G}(\gamma\mathbf{I}\,\mathbf{f}_{\mathrm{p}}+\delta\mathbf{p})$    ▷ Construct the velocity residual Schur Complement
    if Picard then
        $\delta\mathbf{u}(\mathbf{s})=\mathrm{cs\_ltsolve}\left(\mathbf{Lc}, \mathrm{cs\_lsolve}\left(\mathbf{Lc}, \tilde{\mathbf{f}}_{\mathrm{u}}(\mathbf{s})\right)\right)$    ▷ Compute velocity corrections
    else if Newton then
        $\delta\mathbf{u}(\mathbf{s})=\mathrm{kspgcr\_m}\left(\tilde{\mathbf{J}}, \tilde{\mathbf{f}}_{\mathrm{u}}, \delta\mathbf{u}, \mathbf{Lc}, \mathbf{s}, \ldots\right)$    ▷ Compute velocity corrections
    end if
    $\delta\mathbf{p}=\delta\mathbf{p}+\gamma\mathbf{I}(\mathbf{f}_{\mathrm{p}}-\mathbf{G}^T\,\delta\mathbf{u})$    ▷ Compute pressure corrections
end while
$[\alpha]=\mathrm{Linesearch}(\mathbf{u}, \delta\mathbf{u}, \ldots)$    ▷ Search for a globalization parameter
$\mathbf{u}=\mathbf{u}+\alpha\,\delta\mathbf{u}$    ▷ Update solutions
$\mathbf{p}=\mathbf{p}+\alpha\,\delta\mathbf{p}$
```

The iterative Powell-Hestenes workflow (Algorithm 1) avoids an expensive direct factorization of the $\mathbf{M}_{\mathrm{Stokes}}$ by applying the inverse of the Schur complements $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{J}}$. This leads to an efficient solve of the linear system of equations (8) in the incompressible limit. Since $\tilde{\mathbf{K}}$ is a symmetric positive-definite matrix, a Cholesky factorization can be achieved. Being mainly based on vector-vector type of operations, this factorization optimally exploits modern CPU capabilities. We employ the CHOLMOD routine, an efficient sparse supernodal Cholesky factorization package developed by T. Davis [*Chen et al.*, 2008] that relies on highly optimized libraries of BLAS2 operations (multiplication of dense matrices). Within MATLAB it is accessible via the `chol()` function.

Back-substitutions need to be performed iteratively in a different fashion depending if the Picard or the Newton scheme is used. For Picard, *Dabrowski et al.* [2008] proposed an efficient routine using the SuiteSparse `cs_ltsolve` and `cs_lsolve` functions as following `u(s) = cs_ltsolve(L,cs_lsolve(L,rhs(s)))`. A possible alternative without those functions would be to use the Backslash "\" command as following `u(s) = L' \ (L\rhs(s))`. The CHOLMOD related and optimized functions are available by installing the entire SuiteSparse package from the developers Web site (http://faculty.cse.tamu.edu/davis/suitesparse). For Newton iterations, $\tilde{\mathbf{J}}$ is not symmetric and is hence not a candidate for Cholesky factorization. We approximate the application of the inverse Jacobian velocity Schur complement via successive Krylov Subspace iterations (Generalized Conjugate Residuals) [*Eisenstat et al.*, 1983] and use the Cholesky factorization of $\tilde{\mathbf{K}}$ for preconditioning.

In the power law case, the linear solver (Algorithm 1) is embedded into a nonlinear iteration loop. In this case, $\mathbf{f}$ is the nonlinear residual that needs to be minimized (equation (8)) thus specifying the exit condition of the nonlinear loop according to $||\mathbf{f}_{\mathrm{u}}||_{L2} < ||\mathbf{f}_{\mathrm{u}}||_{L2}^{\mathrm{initial}}\mathrm{tol}_{\mathrm{nonlin}}$. A line-search algorithm determines the optimal globalization parameter via direct search after convergence of the linear solver. To do so, nonlinear residuals $(\mathbf{f}_{\mathrm{u}}, \mathbf{f}_{\mathrm{p}})$ are explicitly evaluated (matrix-free approach) for n values of $\alpha=]0\,;\alpha_{\mathrm{max}}]$. For Newton linearization, we set $\alpha_{\mathrm{max}}=1.0$ while for Picard linearization, we let $\alpha_{\mathrm{max}} > 1.0$. The value of α that yields to:

$$\min ||\mathbf{f}_{\mathrm{u}} + \alpha\, \delta\mathbf{x}||_{\mathrm{L2}} \tag{12}$$

is retained as the optimal globalization parameter. In the linear case, a global optimization parameter equal to 1 is applied and the problem converges in a single nonlinear iteration.

2.2.2. Optimized Assembly and Boundary Conditions

In the previous sections, optimized solving strategies were presented, assuming a global system of linear equations for the linear or nonlinear Stokes flow. Prior to solve, the subsystems matrices need to be assembled, and boundary conditions should be applied. Taking full advantage of MATLAB, the assembly of the submatrices should be performed in a vectorized fashion, instead of using loops. Transparency and optimal performance are the major outcomes of this choice. In a first step, the 2-D fields of the stiffness matrix coefficients are saved in submatrices taking into account the appropriate staggering. Equation numbering is also implemented following this strategy. In a second step, those fields are saved in a sparse triplet vector format using MATLAB's `sparse()` function or a more efficient `sparse2()` function if the SuiteSparse package is installed.

Within the solver, both Dirichlet and Neumann type BC are implemented for nodes that either conform or not the domain boundaries. For conforming Dirichlet BC (fixed normal velocity), scaled BC values are injected into the right hand side vector while corresponding lines of the stiffness matrix are canceled and diagonal values are set to the appropriate scaling factor. All related matrix coefficients connecting to these nodes are multiplied by corresponding Dirichlet BC values and are subtracted from the right hand side prior to cancellation. For nonconforming Dirichlets BC, a modified stencil accounting for fixed tangential velocity (ghost node approach) is assembled. These constraints induce modifications of both the diagonal of the stiffness matrix and the corresponding entry of the right hand side vector. In a similar manner, conforming Neumann type BC are constructed in order to impose fixed tangential stresses. Such assembly maintains a symmetric and positive-definite stiffness matrix, making it a good candidate for Cholesky factorization.

3. Results

3.1. Validation of the Linear Stokes Flow Solver

Validation of the linear Stokes flow solver can be achieved by performing a grid convergence analysis. To this end, we evaluate the magnitude of velocity and pressure truncation errors ($e_{\mathbf{u}}, e_{\mathbf{p}}$) for decreasing spatial discretization steps h. In practice, truncations error is calculated by subtracting numerically calculated solution fields from analytically calculated solutions ($\mathbf{u}_{\mathrm{analytic}}, \mathbf{p}_{\mathrm{analytic}}$). We characterize the magnitude of truncation errors by their L_1 norms:

$$\begin{aligned} e_{\mathbf{u}} &= ||\mathbf{u}_{\mathrm{analytic}} - \mathbf{u}||_1, \\ e_{\mathbf{p}} &= ||\mathbf{p}_{\mathrm{analytic}} - \mathbf{p}||_1. \end{aligned} \tag{13}$$

The analytical solution of *Schmid and Podladchikov* [2003] provides full 2-D flow and pressure fields for incompressible Stokes flow incorporating a viscous circular inclusion subjected to background shear. Confronting our numerical solution to this analytical solution is particularly challenging as Cartesian grids cannot conform to the circular geometry of inclusions. Nevertheless, numerical solutions for velocity and pressure exhibit first-order spatial accuracy. In other words, truncation error magnitudes decrease by a factor k as the grid spacing is reduced by the same factor (Figure 2). These results are in agreement with previous

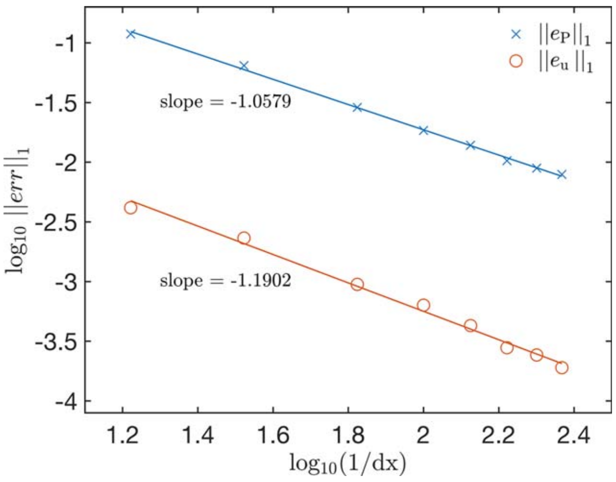

Figure 2. Evolution of velocity and pressure truncation errors (L1-norm) upon mesh refinement. The Finite Difference scheme is first-order accurate in space for both velocity and pressure. The symbols correspond to numerical runs and the lines corresponds to linear interpolants. We have set the domain size to $[-3, 3]\times[-3, 3]$, the inclusion radius to 1, the matrix viscosity to 1, and the inclusion viscosity to 10^4. Boundary conditions for the analytical solution were set to far-field pure shear. For the numerical solution, we use full Dirichlet boundary conditions where normal and tangential velocity components were analytically evaluated. These results can be reproduced using the appended script M2Di_Dani.m.

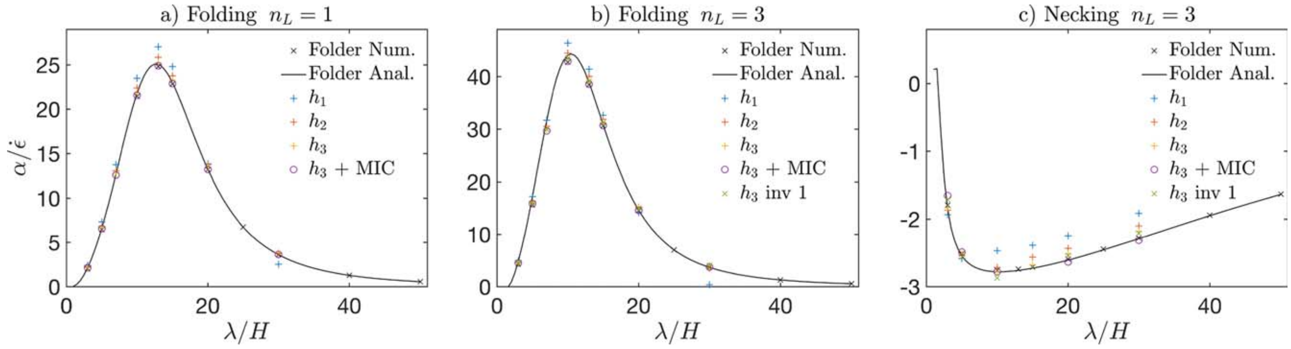

Figure 3. Numerically calculated growth rates $\alpha/\dot{\epsilon}$ for various types of mechanical instabilities: (a) linear viscous folding, (b) power law viscous folding, and (c) power law viscous necking. The results are compared to those obtained with the Folder toolbox [*Adamuszek et al*., 2016] (black dot and line). The domain dimensions were set to $[-\lambda/2, \lambda/2]\times[-3, 3]$, where λ corresponds to the wavelength of the sinusoidal perturbation. The layer thickness $H = 1$, the amplitude of the perturbation is set to 0.0500, and the reference viscosity contrast is 200. Growth rates were calculated using three different grid resolutions, vertical, and horizontal grid spacings were kept equal (h_1=0.0500, h_2=0.0250, h_3=0.0125). The highest resolution benchmark run, h_3, was also performed using a marker-in-cell (MIC) viscosity interpolation at the material interface and using the "inv 1" invariant formulation. The appended scripts can be used to reproduce these results (M2Di_Newton_Folding.m and M2Di_Newton_Necking.m).

studies [e.g., *Duretz et al*., 2011] that reported first-order spatial accuracy of the staggered grid for similar model configurations.

3.2. Validation of the Power Law Stokes Flow Solver

To validate the power law Stokes solver, we calculated numerical growth rates for folding and necking instabilities and compared them against reference solutions. We use the Folder software [*Adamuszek et al*., 2016] to compute both a reference numerical solution (body-fitted Finite Element Method) as well as analytical solutions (Thick Plate Finite Matrix Thickness). For both folding and necking, model configurations consist of a layer embedded in matrix subjected to either compression or stretching. To initiate mechanical instabilities, the position of layer/matrix interface requires a geometrical perturbation (sinusoidal). As in the previous section (3.1), this test is challenging when making use of regularly spaced Cartesian grids because the amplitude of the perturbation needs to be resolved over several nodal points to influence the flow field. Subsequently successful Finite Difference solutions require a large number of nodal points in both spatial dimensions (see Figure 3 caption). As an alternative, variable grid spacing or arbitrary mesh refinement [i.e., *Gerya et al*., 2013] could be used to achieve relatively finer resolution along material interfaces. For simplicity, we do not introduce this additional level of complexity at this point. We have carried out simulations for folding of a linear viscous layer (Figure 3a), power law viscous layer (Figure 3b), and necking of a power law viscous layer (Figure 3c) in a linear viscous matrix. Growth rates were measured for different wavelengths and using three different grid spacings ($h_1 > h_2 > h_3$) and "inv 0" invariant formulation. For all tests, overall grid convergence is observed as numerical solutions tend to benchmark solutions with decreasing grid spacing. Additionally, we have run simulations that accounts for marker-in-cell type viscosity evaluation (volume-average arithmetic interpolation) and the finest grid spacing h_3. Such an approach introduces sensitivity at the subgrid level and manages to better reproduce the benchmark growth rate solutions at lower resolutions. Growth rate calculation on the finest level (h_3) was also run with the "inv 1" second strain rate invariant formulation. The resulting growth rates show very little dependence on the type of invariant formulation. However the "inv 1" formulation requires less Newton iterations to converge.

3.3. Performance of the Algorithms

3.3.1. Overall Performance

Within this study, we propose an efficient approach using the efficient sparse Cholesky factorization to solve 2-D Stokes flow in the minimal amount of time on MATLAB, focusing on ease of usage rather than on ultimate performance. Therefore, to estimate the performance of our routines, we performed the related timings on a MacBook Pro laptop running OS X 10.11 equipped with 16 GB of memory (RAM) and a dual-core Intel Core i7 processor @ 3.1 GHz. MATLAB R2016a together with SuiteSparse v4.5.3 were installed. We use the wall time reported in Figure 4, or overall time to solution, as performance indicator. The reasons that motivate this choice are twofold. First, as MATLAB is an interpreted language, it is rather complicated to

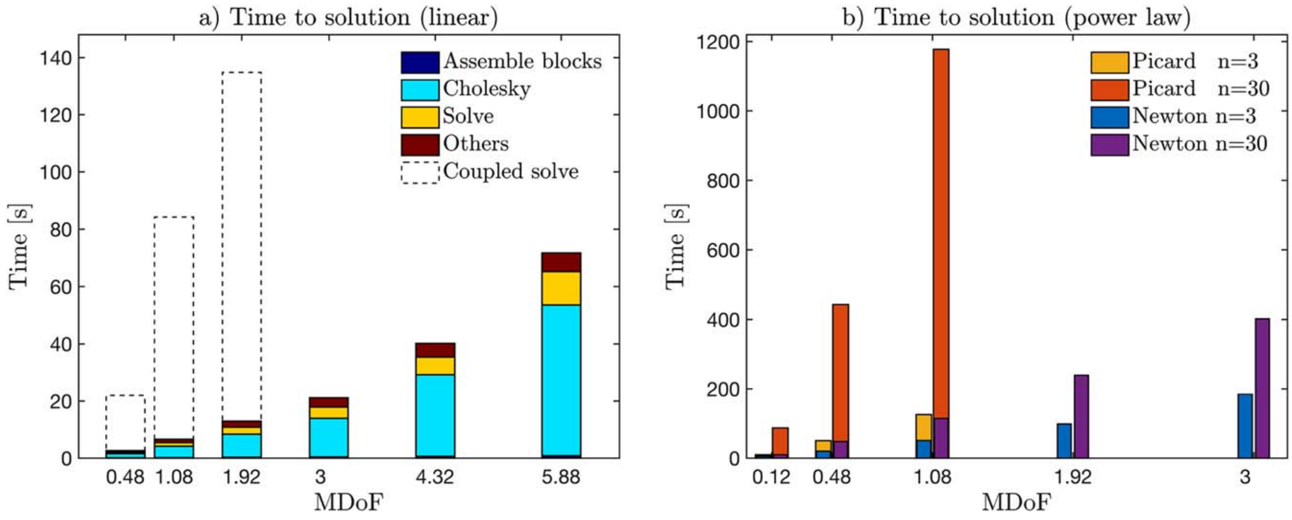

Figure 4. Time to solution in seconds for (a) the linear Stokes solver and (b) the power law Stokes solver for both Picard and Newton operators as function of Millions of Degrees of Freedom (MDoF—computed for the 2-D setup as $nx \times ny \times 3 \times 10^{-6}$). In Figure 4a, the wall time of the optimized solve (plain bars) is compared to the wall time of a coupled solve (dashed bars). The optimized solve is detailed in four parts and consists of a vectorized block assembly (Assemble blocks), a Cholesky factorization, iterative Powell-Hestenes back-substitutions using the SuiteSparse functions (Solve), and some other required operations. The postprocessing is excluded from the reported timings. The coupled solve is performed using MATLAB's backslash (calling UMFPACK) for the direct solve. In Figure 4b, the wall times for nonlinear power law exponents $n = 3$ and $n = 30$ are shown as function of MDoF. The setup used in both Figures 4a and 4b cases is a cylindrical inclusion of high viscosity located in the center of a square domain under simple shear with a background strain rate value set to 1. In Figure 4a, the viscosity of the matrix and of the inclusion is 1 and 10^4, respectively. In Figure 4b, the matrix has a reference viscosity of 1 and a power law exponent of n, the minimum and maximum viscosities were respectively set to 10^{-6} and 10^3. The viscosity of the inclusion was set to 10^3.

determine the exact memory usage, or the accurate number of flops involved for the entire workflow. Second, the end-user is mainly interested in the approximate time to solution, rather than in a complex metric that would depend on hardware technicalities, beyond the scope of this technical brief.

3.3.2. Linear Solver Performance

The performance of our linear Stokes flow solver is reported in Figure 4a. The model configuration is identical to the one used for the linear solver validation (section 3.1), featuring a circular inclusion located in the center of a square domain under pure shear with background strain rate value set to 1. Viscosity values for the inclusion and the matrix are, respectively, 10^4 and 1. The square domain has a size of 6×6 and the radius of the inclusion is 1. The straightforward solve, a direct factorization of the stiffness matrix (dashed bars) using the UMFPACK solver, could be sustained up to 1.92 MDoF, which represent a 2-D domain of 800^2 grid points. After that, the 16 GB of available RAM were saturated, while the time to solution seriously deviates from any linear trend. Moving toward an optimized solving strategy involving a sparse Cholesky factorization and Powell-Hestenes iterations (plain bars), both the time to solution and the maximal problem size could be greatly improved. A solution of 5.88 MDoF, representing a 2-D domain size of 1400^2 grid points could fit within the 16 GB of RAM, while the dependence of the solving time on the resolution is weakly nonlinear.

As the straightforward implementation gave unsatisfactory and rather limiting results, we will highlight some key parts of the successful optimizations. A closer look at the wall time reveals a linear scaling for the vectorized matrix assembly as well as back-substitutions, the scaling of the Cholesky factorization is close to linear. This type of behavior was also reported by *Dabrowski et al*. [2008]. We reach MATLAB limits in term of efficiency with a vectorized assembly and the usage of SuiteSparse optimized `sparse2()`, `cs_ltsolve()`, and `cs_lsolve` functions. Thus, the only computationally expensive part of the optimized solving procedure should reduce to the Cholesky factorization. In terms of memory requirement, the Cholesky approach performs rather well, as only the lower part of the stiffness matrix needs to be assembled, factorized, and stored, reducing the demand on the RAM by a factor close to two. Finally, the efforts of using a symmetric and positive definite stiffness matrix appropriate for a Cholesky type of solver with iterative back-substitutions resulting in faster solving times for a 1400^2 (72 s) versus a 600^2 (84 s) grid points domain for the straightforward approach.

As a general remark, the reported timing demonstrates that the iterative penalty (Powell-Hestenes) approach combined with the usage of Cholesky factorization (restricted to symmetric and positive definite matrices) is a powerful approach and further confirms the trend observed by *Dabrowski et al.* [2008].

3.3.3. Power Law Solver Performance

The performance of both the Picard and the Newton nonlinear solver is reported in Figure 4b for two different values of the power law exponent *n* (see section 2.1.1) per reported DoF. The model configuration is identical to the one used for the linear solver performance (section 3.3.2) evaluation. A background strain rate of 1 is applied and the reference matrix viscosity is set to 1. The minimum and maximum viscosity values for the Carreau model were set to $\eta_\infty = 10^{-6}\eta_{\mathrm{ref}}$ and $\eta_0 = 10^3\eta_{\mathrm{ref}}$, respectively (see section 2.1.1). Introducing a nonlinear rheology impacts the solving time, as now several linear solves have to be performed within the nonlinear iteration loop, until the nonlinear residual tolerance $\mathrm{tol}_{\mathrm{nonlin}}$ is reached. The wall time increases due to this requirements. The performance of the direct factorization approach, or coupled solve, is not reported for the nonlinear case, as it shows an order of magnitude lower performance compared to the optimized solving strategies, and only up to 400^2 grid points could be resolved with the available 16 GB of RAM.

For the Newton linearization, the stencil gets additional DoFs (Figure 1b), thus reducing the global domain size that can reside in memory. The Newton linearization is less sensitive to the increase in non-linearity. Moving from $n = 3$ to $n = 30$, the time to solution increased by an approximately constant factor 3 for the five tested grid resolutions (ranging from 0.12 to 3 MDoF). This is mainly due to the superlinear scaling of the Cholesky factorization together with a slight increase in number of Newton steps to converge the nonlinear residual.

The Picard linearization has a reduced stencil (Figure 1a) and uses therefore less memory for a similar domain size compared to the Newton linearization. Nevertheless, the main limitation of the linearized Picard nonlinear solver resides in its sensitivity to the nonlinearity of the problem. Even if every linear solve is performed faster than in the Newton case, their number tend to drastically increase with increase in the nonlinear exponent *n*. In highly nonlinear setups, the nonlinear iteration ratio between Picard and Newton type of solver reaches factors in the range of 200:10. While the Newton solver is a powerful tool, it is more sensible regarding the tuning parameters and may fail to converge in some cases while the Picard nonlinear solver, although requiring a growing number of iteration with increasing nonlinearity, will not. Note that in the provided Newton routines, the user has the possibility to perform a number of Picard steps before switching to Newton iterations.

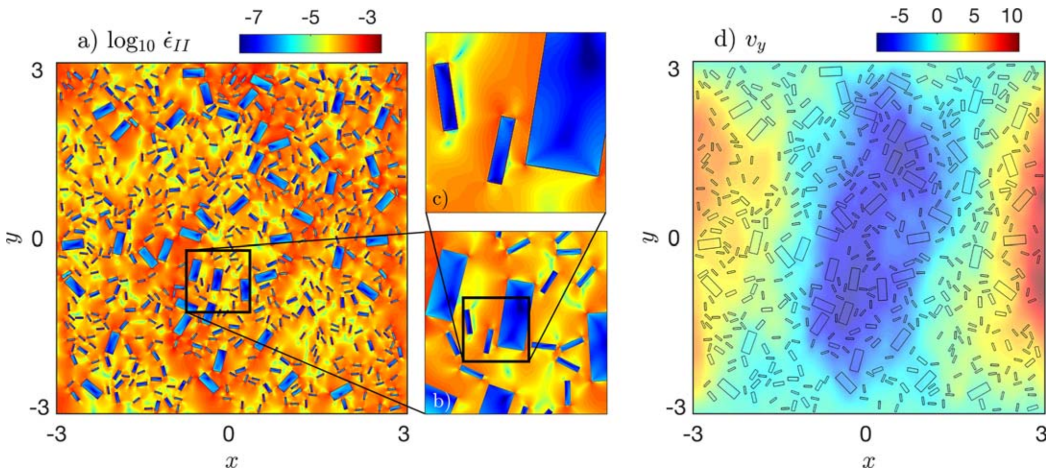

Figure 5. High-resolution (5000^2 cells) instantaneous model of crystal/melt settling using a random crystal distribution. (a) The strain rate over the entire model domain (dimensionless). (b) And (c) different figure enlargements and the black frames correspond to the respective areas. (d) The vertical velocity component (dimensionless), each individual crystal is outlined by a black contour line. The crystals have different aspect ratios (2.5:1, 3:1, 3:1) and densities (1.350, 1.3750, 1.400). The fluid has a density of 1.3 and a viscosity of 1. The viscosity ratio between the crystals and the fluid is set to 10^3 and the gravity is set to -1. The crystal fraction reaches 18%. A 500^2 resolution example is provided by the appended script M2Di_crystals.m.

4. Examples of Instantaneous Flow Field Calculations

In order to demonstrate the flexibility of the above-described Stokes solvers, we provide four examples of flow field calculations that are relevant for different types of geodynamics problems and that highlight several technical aspects described above.

4.1. Direct Numerical Simulations of Crystal-Melt Dynamics

The dynamics of systems consisting of multiple material phases of distinct rheologies and densities can be studied using direct numerical simulations. Magmatic suspensions and crystal mushes are often considered as laminar fluids [*Glazner*, 2014] and processes such as phase separation, mixing, and segregation can, to some extent, be investigated using Stokes flow models [*Suckale et al*., 2012; *Yamato et al*., 2012b,]. Numerical models, however, require high resolution in order to render the hydrodynamic interactions between crystals and the net effect of the crystalline load on the overall flow. We provide a high-resolution instantaneous flow model of crystal settling under gravity using the above-described Stokes flow solver (section 2.1.1). Three populations of prismatic crystals characterized by various dimensions, aspect ratio, and density were distributed quasi-randomly across the model domain. The crystals are considered as highly viscous inclusions and are hence characterized by regions of low strain rate (Figures 5a–5c). In contrast, the interstitial liquid accommodates the strain induced by the sinking of the denser crystals and hence exhibits large strain rate (Figures 5a–5c). The fraction of crystals is larger in the center of the domain than toward the laterals sides of the domain. The effective density is therefore greater in the center of the domain, which induces a convection flow pattern (Figures 5a and 5d). Here, we observe that crystals have a collective sinking behavior [e.g., *Hernandez-Ortiz et al*., 2005] in the center of the domain in response to hydrodynamic interactions (i.e., viscous drag).

4.2. Pressure Variations Around Viscous Inclusions

The prediction of pressure and flow field variations around viscous inclusions has numerous geological implications [*Schmid and Podladchikov*, 2003; *Moulas et al*., 2014]. For an isolated inclusion, the quality of the numerical solution can be improved by increasing the grid resolution close to the material boundary [e.g., *Gerya et al*., 2013]. Here we present numerical models of a single inclusion embedded in constant viscosity matrix subjected to far-field pure shear (see section 3.1) and show the benefits of using variable grid spacing. Pressure fields were computed using numerical resolutions of 500^2 cells using regular (Figure 6a) and variable grid spacings (Figure 6b). Variable grid spacing drastically improves the quality of the pressure field at the inclusion/matrix interface, which is known to be oscillatory for staggered grids [*Deubelbeiss et al*., 2011] in absence of a specific material jump treatment (e.g. ghost fluid method) [*Suckale et al*., 2010]. Moreover, we could show a better reduction of pressure and velocity truncation errors upon mesh refinement when using variable grid spacing (Figure 6c).

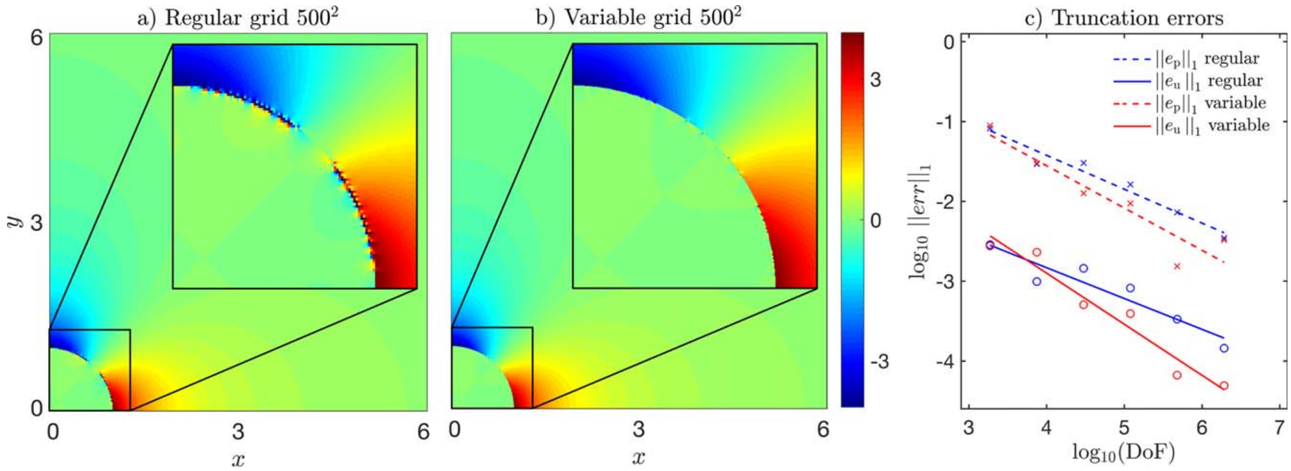

Figure 6. (a and b) Computed pressure fields for a single viscous inclusion embedded in constant viscosity matrix using a resolution of 500^2 nodes. The domain dimensions are set to $[0, 6]\times[0, 6]$, the inclusion radius to 1, the matrix viscosity to 1, and the inclusion viscosity to 10^4. (a) Solution obtained with regular grid spacing. (b) Solution computed with variable grid spacing in *x* and *y* dimensions and refined in the vicinity of the inclusion. (c) Evolution of pressure and velocity truncation errors (L1-norm) as function of the number of degrees of freedom (DoF). The results can be reproduced using the appended script M2Di_Linear_vardxy.m.

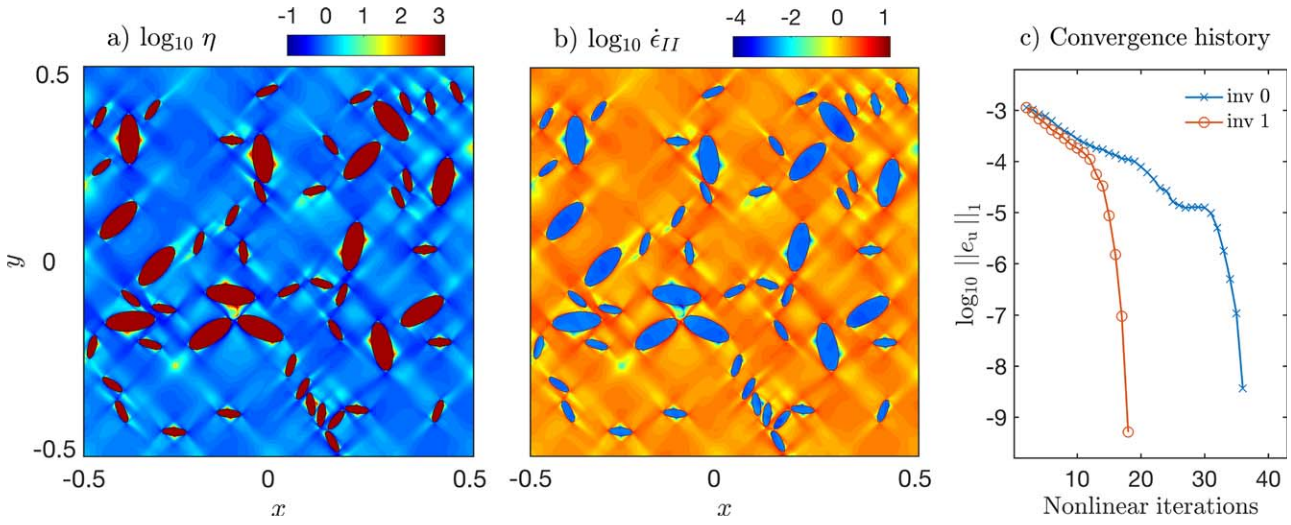

Figure 7. Horizontal compression of a material composed of randomly located highly viscous elliptical inclusions in a power law matrix (resolution 1000^2 cells). The nonlinear solver ran until reaching a relative residual of 10^{-8}. (a) The resulting effective viscosity (dimensionless) is depicted and (b) the corresponding strain rate (dimensionless) is depicted. The matrix has a reference viscosity of 1 and a power law exponent of 10, the minimum and maximum viscosities were, respectively, set to 10^{-3} and 10^{3}. The viscosity of the inclusions was set to 10^{3}. The background strain rate is set to 1. The nonlinear solver (using the Newton linearization) converged to relative $\mathrm{tol}_{\mathrm{nonlin}} < 10^{-8}$ in 18 iterations. (c) The $\log_{10}$ of the corresponding convergence history for the two different definitions of the second strain rate invariant $\dot{\epsilon}_{II}$ as function of nonlinear iteration count. inv 0 refers to a linear interpolation of the missing strain rate components while inv 1 refers to a linear interpolation of the second invariant contributions.

4.3. Deformation of Heterogenous Power Law Viscous Materials

Rocks often consist of different mineral phases and are therefore heterogeneous. If heterogeneities provide a strength contrast, they can act as stress concentrators. Experimentally derived flow laws characterizing the rheology of single mineral phases often exhibit a power law behavior. Modeling the deformation of heterogenous power law viscous materials allows to unravel the dynamics of shear localization and the formation of rock fabrics [*Mancktelow*, 2002; *Jessell et al.*, 2009; *Deubelbeiss et al.*, 2011; *Dabrowski et al.*, 2012; *Le Pourhiet et al.*, 2013]. We have set up a power law model consisting in randomly located highly viscous elliptical inclusions. The viscosity of the inclusions is constant and the matrix rheology follows a Carreau model (section 2.1.1). Horizontal pure-shear compression is applied at the boundaries of the domain and gravity is deactivated (see Figure 7 caption for model parameters). The model depicts the establishment of heterogeneous shear zone patterns in the vicinity of the inclusions. Areas of reduced viscosity (Figure 7a) develop in the regions of increased strain rate (Figure 7b). The shear zones develop with an angle of 45° to the compression direction and are locally intensified by the proximity and orientation of the inclusions. Both second strain rate invariant formulations ("inv 0" and "inv 1" — see section 2.2) were tested. While computed nonlinear viscosities and strain rate patterns do not significantly differ, the number of nonlinear iterations needed to converge is greatly reduced using "inv 1" formulation (18 for "inv 1," 36 for "inv 0," Figure 7c).

4.4. Mantle Flow and Dynamic Topography

Instantaneous flow models are a powerful tool for improving our knowledge of the viscosity and density structure of the Earth's mantle [e.g., *Billen et al*. 2003; *Gérault et al*., 2015]. At this scale, it is convenient to solve the flow problem in cylindrical coordinates. This framework allows to account for the nonnegligible role of Earth's curvature [*Jarvis*, 1993], for example in slab-mantle interaction [*Morra et al*., 2009]. The Stokes equations (equation (1)) can be formulated in cylindrical coordinates in a form suitable for staggered grid discretization (section Appendix A). The corresponding discrete operator conserves the properties of the Cartesian discretization (symmetric positive-definite), hence similar linear flow solvers can be used efficiently (see section 2.1.1). To demonstrate the flexibility of our approach, we have designed an instantaneous mantle flow model in cylindrical coordinates using linear (constant) viscosity (Figure 8). The model configuration incorporates a 120 km thick plate, a detached negatively buoyant slab, and a nonsmooth radial viscosity structure. Here viscous drag exerted by the sinking detached slab induces longitudinal plate motion ($v_\theta \approx 5$ cm/yr, Figure 8a) and positive dynamic topography ($h_d \approx 150$ m, Figure 8b). This model well illustrates the role of the slab suction force in inducing plate motions [*Conrad and Lithgow-Bertelloni*, 2002].

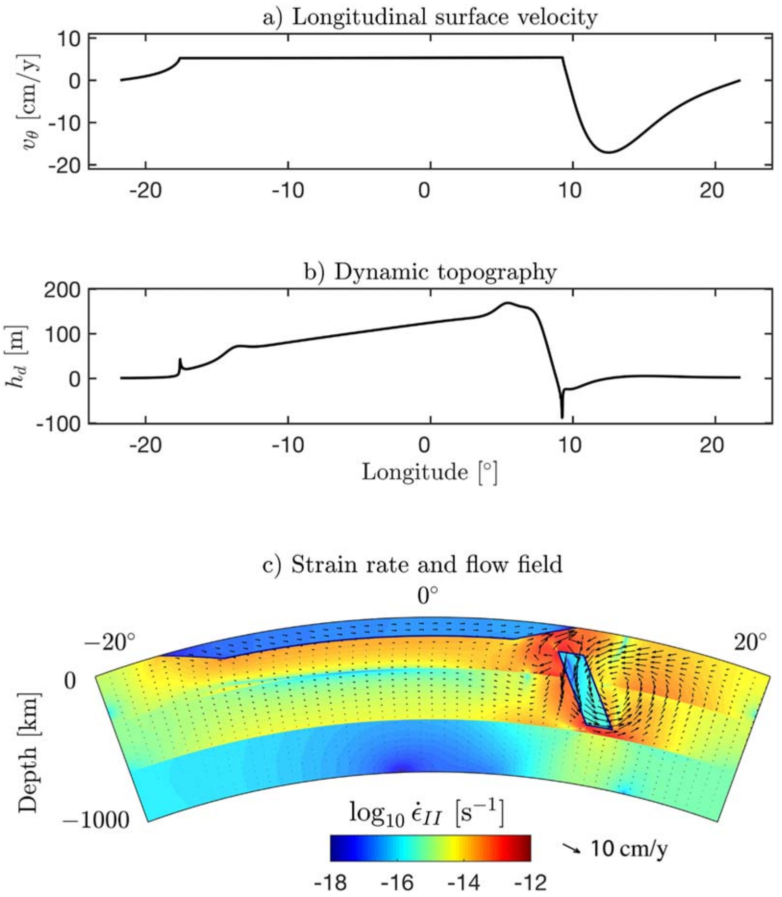

Figure 8. Instantaneous model of mantle flow induced by the sinking of a detached slab in cylindrical coordinates (Resolution 1000 × 2000 cells). (a) The longitudinal velocity along the surface. (b) The dynamic topography induced by the mantle flow. (c) A color plot of the effective strain rate, arrows correspond to velocity vectors and grey contours outline the location of a continental plate and the slab. The dynamic topography (h_d) was calculated according to the relationship $h_d = \frac{-\sigma_{rr}}{g_r \Delta\rho}$, where σ_{rr} is the total radial stress measured along the surface, g_r is the radial component of gravity, and $\Delta\rho$ is the density difference between the lithosphere and the air (assumed equal to 0 kg/m^3). The mantle density was set 3250 kg/m^3, its viscosity varies from 10^{19} Pa.s down to −330 km, 10^{20} Pa.s down to −660 km and 10^{22} Pa.s down to 1000 km. The slab and plate densities were, respectively, set to 3275 kg/m^3 and 3245 kg/m^3. Their viscosities were set to 10^{23} Pa.s. All boundary conditions are set to free slip. This simulation can be reproduced using the appended script M2Di_cyl_slab.m.

5. Conclusions

We developed and provide M2Di, an efficient set of routines that solve linear and nonlinear Stokes flow on a personal computer using MATLAB. We report a wall time of 22 s for the linear solver and both 180 and 400 s for the nonlinear Newton solver with power law exponent $n = 3$ and $n = 30$ (single inclusion setup), respectively, on a 1000^2 2-D domain (3 MDoF — Figure 4). These time to solution could be obtained using vectorized assembly of matrix operators and solvers relying on efficient sparse Cholesky factorization routines provided in T. Davis' SuiteSparse library (Accessible in native MATLAB directly via the `chol()` function). The routines were validated by comparing our numerical results to analytical and benchmark numerical solutions for both linear and power law viscous flow. Here we only considered Dirichlet and Neumann boundary conditions; however, the routines could also be extended to account for free surface flow [*Duretz et al.*, 2016].

Multiphysics applications necessitate coupling between a mechanical solver and diffusion-type solvers (fluid pressure, heat, compositional) as well as accurate resolution of nonlinearities. In this perspective, the development of 2-D THCM models will benefit from the presented mechanical solvers (M). The latter will be used as the main building block to further tackle studies of thermomechanical shear localization [*Duretz et al.*, 2014] (TM coupling), two-phase flow [*Yarushina and Podladchikov*, 2015] leading to fluid channeling instabilities [*Yarushina et al.*, 2015; *Räss et al.*, 2014] (HM coupling), reactive porosity waves [*Malvoisin et al.*, 2015] (HCM coupling), or THCM coupling [*Keller and Katz*, 2016; *Weatherley and Katz*, 2012]. Features arising from THCM coupling are often localized in space, close to material interfaces (e.g., shear zones, fluid channeling) and may depend on the local pressure field. Our results indicate that variable grid spacing (Cartesian grid) can greatly improve the solution of pressure fields close to material boundaries and may therefore be considered in multiphysics applications.

Appendix A: A Stokes Flow Equation in Cylindrical Coordinates

In order to perform the model presented in section 4.4, the momentum and continuity equations (equation (1)) were reformulated in cylindrical coordinates (r, θ):

$$\begin{gathered}\frac{1}{r}\frac{\partial\sigma_{\theta\theta}}{\partial\theta}+\frac{\partial\sigma_{r\theta}}{\partial r}+2\frac{\sigma_{r\theta}}{r}=0,\\ \frac{\partial\sigma_{rr}}{\partial r}+\frac{1}{r}\frac{\partial\sigma_{r\theta}}{\partial\theta}+\frac{\sigma_{rr}}{r}-\frac{\sigma_{\theta\theta}}{r}=-\rho g_r,\\ \frac{1}{r}\frac{\partial v_\theta}{\partial x}+\frac{1}{r}\frac{\partial(rv_r)}{\partial y}=0,\end{gathered}\tag{A1}$$

To allow discretization on a staggered grid, the above equations were rewritten using the product rule and multiplied by r:

$$\begin{gathered}\frac{\partial\sigma_{\theta\theta}}{\partial\theta}-r\frac{\partial\sigma_{r\theta}}{\partial r}+2\frac{\partial(r\sigma_{r\theta})}{\partial r}=0,\\ \frac{\partial(r\sigma_{rr})}{\partial r}+r\frac{\partial\sigma_{\theta\theta}}{\partial r}-\frac{\partial(r\sigma_{\theta\theta})}{\partial r}+\frac{\partial\sigma_{r\theta}}{\partial\theta}=-r\rho g_r,\\ \frac{\partial v_\theta}{\partial x}+\frac{\partial(rv_r)}{\partial y}=0,\end{gathered}\tag{A2}$$

In this form, the gradients of stress tensor components and the value of the components do not have to be spatially collocated. The Finite Difference discretization hence does not involve any stress interpolations and the stencil involves the same number of points (no stencil growth) as in the Cartesian formulation (see section 2.1.1). Similarly the stress tensor components were expressed as:

$$\begin{gathered}\sigma_{\theta\theta}=-P+2\eta\left(\frac{v_r}{r}+\frac{1}{r}\frac{\partial v_\theta}{\partial\theta}\right)=-P+2\eta\left(\frac{1}{r}\frac{\partial(rv_r)}{\partial r}-\frac{\partial v_r}{\partial r}+\frac{1}{r}\frac{\partial v_\theta}{\partial\theta}\right),\\ \sigma_{rr}=-P+2\eta\left(\frac{\partial v_r}{\partial r}\right),\\ \sigma_{r\theta}=\eta\left(\frac{\partial v_\theta}{\partial r}+\frac{1}{r}\frac{\partial v_r}{\partial\theta}-\frac{v_\theta}{r}\right)=\eta\left(2\frac{\partial v_\theta}{\partial r}-\frac{1}{r}\frac{\partial(rv_\theta)}{\partial r}\frac{1}{r}\frac{\partial v_r}{\partial\theta}\right),\end{gathered}\tag{A3}$$

where v_θ and v_r, respectively, correspond to the longitudinal and radial velocity vector components.

Acknowledgments
The authors are thankful to Marcin Dabrowski and Anton Popov for their constructive review which helped to significantly improve the quality and content of the proposed work. Dave A. May is thanked for its constructive comments on the earlier version of the draft that considerably improved the final manuscript. The authors acknowledge P. J. Yamato who provided the script used for generating the random particle distributions in Figures 5 and 7.

References

Adamuszek, M., M. Dabrowski, and D. W. Schmid (2016), Folder: A numerical tool to simulate the development of structures in layered media, *J. Struct. Geol.*, *84*, 85–101.

Allen, G. M. (1999), Rheology of polymeric systems, principles and applications, *AIChE J.*, *45*(8), 1836–1837.

Bea, S., U. Mayer, and K. MacQuarrie (2015), Reactive transport and thermo-hydro-mechanical coupling in deep sedimentary basins affected by glaciation cycles: Model development, verification, and illustrative example, *Geofluids*, *16*(2), 279–300.

Billen, M. I., M. Gurnis, and M. Simons (2003), Multiscale dynamics of the Tonga-Kermadec subduction zone, *Geophys. J. Int.*, *153*(2), 359–388.

Chen, Y., T. A. Davis, W. W. Hager, and S. Rajamanickam (2008), Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate, *ACM Trans. Math. Software*, *35*(3), 22.

Conrad, C. P., and C. Lithgow-Bertelloni (2002), How mantle slabs drive plate tectonics, *Science*, *298*(5591), 207–209.

Cuvelier, C., A. Segal, and A. A. Van Steenhoven (1986), *Finite Element Methods and Navier-Stokes Equations*, vol. 22, D. Reidel Publishing Company, 504 pp., Springer, Netherlands.

Dabrowski, M., M. Krotkiewski, and D. W. Schmid (2008), Milamin: Matlab-based finite element method solver for large problems, *Geochem. Geophys. Geosyst.*, *9*, Q04030, doi:10.1029/2007GC001719.

Dabrowski, M., D. W. Schmid, and Y. Y. Podladchikov (2012), A two-phase composite in simple shear: Effective mechanical anisotropy development and localization potential, *J. Geophys. Res.*, *117*, B08406, doi:10.1029/2012JB009183.

Deubelbeiss, Y., and B. Kaus (2008), Comparison of Eulerian and Lagrangian numerical techniques for the Stokes equations in the presence of strongly varying viscosity, *Phys. Earth Planet. Inter.*, *171*(1), 92–111.

Deubelbeiss, Y., B. J. P. Kaus, J. A. D. Connolly, and L. Caricchi (2011), Potential causes for the non-Newtonian rheology of crystal-bearing magmas, *Geochem. Geophys. Geosyst.*, *12*, Q05007, doi:10.1029/2010GC003485.

Duretz, T., D. A. May, T. V. Gerya, and P. J. Tackley (2011), Discretization errors and free surface stabilization in the finite difference and marker-in-cell method for applied geodynamics: A numerical study, *Geochem. Geophys. Geosyst.*, *12*, Q07004, doi:10.1029/2011GC003567.

Duretz, T., S. Schmalholz, Y. Podladchikov, and D. Yuen (2014), Physics-controlled thickness of shear zones caused by viscous heating: Implications for crustal shear localization, *Geophys. Res. Lett.*, *41*, 4904–4911, doi:10.1002/2014GL060438.

Duretz, T., D. May, and P. Yamato (2016), A free surface capturing discretization for the staggered grid finite difference scheme, *Geophys. J. Int.*, *204*(3), 1518–1530.

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

Gérault, M., L. Husson, M. S. Miller, and E. D. Humphreys (2015), Flat-slab subduction, topography, and mantle dynamics in southwestern Mexico, *Tectonics*, *34*, 1892–1909, doi:10.1002/2015TC003908.

Gerya, T. V., and D. A. Yuen (2003), Characteristics-based marker-in-cell method with conservative finite-differences schemes for modeling geological flows with strongly variable transport properties, *Phys. Earth Planet. Inter.*, *140*(4), 293–318.

Gerya, T. V., and D. A. Yuen (2007), Robust characteristics method for modelling multiphase visco-elasto-plastic thermo-mechanical problems, *Phys. Earth Planet. Inter.*, *163*(1), 83–105.

Gerya, T. V., D. A. May, and T. Duretz (2013), An adaptive staggered grid finite difference method for modeling geodynamic stokes flows with strongly variable viscosity, *Geochem. Geophys. Geosyst.*, *14*, 1200–1225, doi:10.1002/ggge.20078.

Glazner, A. F. (2014), Magmatic life at low Reynolds number, *Geology*, *42*(11), 935–938, doi:10.1130/G36078.1.

Hernandez-Ortiz, J. P., C. G. Stoltz, and M. D. Graham (2005), Transport and collective dynamics in suspensions of confined swimming particles, *Phys. Rev. Lett.*, *95*, 204501.

Jarvis, G. T. (1993), Effects of curvature on two-dimensional models of mantle convection: Cylindrical polar coordinates, *J. Geophys. Res.*, *98*(B3), 4477–4485.

Jessell, M. W., P. D. Bons, A. Griera, L. A. Evans, and C. J. Wilson (2009), A tale of two viscosities, *J. Struct. Geol.*, *31*(7), 719–736.

Kaus, B. J., A. A. Popov, T. Baumann, and A. E. Püsök (2016), Forward and inverse modelling of lithospheric deformation on geological timescales, in *Proceedings of NIC Symposium 2016, NIC Ser.*, vol. 48., edited by K. Binder et al., 299 pp., Forsch. J̈ulich GmbH, John von Neumann Inst. for Comput., Schr. des Forsch. Julich, Julich, Germany.

Keller, T., and R. F. Katz (2016), The role of volatiles in reactive melt transport in the asthenosphere, *J. Petrol*, *57*(6), 1073–1108.

Le Pourhiet, L., B. Huet, L. Labrousse, K. Yao, P. Agard, and L. Jolivet (2013), Strain localisation in mechanically layered rocks beneath detachment zones: Insights from numerical modelling, *Solid Earth*, *4*(1), 135–152.

Malvoisin, B., Y. Y. Podladchikov, and J. C. Vrijmoed (2015), Coupling changes in densities and porosity to fluid pressure variations in reactive porous fluid flow: Local thermodynamic equilibrium, *Geochem. Geophys. Geosyst.*, *16*, 4362–4387, doi:10.1002/2015GC006019.

Mancktelow, N. S. (2002), Finite-element modelling of shear zone development in viscoelastic materials and its implications for localisation of partial melting, *J. Struct. Geol.*, *24*(6–7), 1045–1053.

May, D. A., and L. Moresi (2008), Preconditioned iterative methods for stokes flow problems arising in computational geodynamics, *Phys. Earth Planet. Inter.*, *171*(1), 33–47.

May, D. A., J. Brown, and L. L. Pourhiet (2014), ptatin3d: High-performance methods for long-term lithospheric dynamics, in *SC '14 Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 274–284, IEEE Press, Piscataway, N. J.

Montési, L. G. J., and M. T. Zuber (2002), A unified description of localization for application to large-scale tectonics, *J. Geophys. Res.*, *107*(B3), 2045, doi:10.1029/2001JB000465.

Morra, G., P. Chatelain, P. Tackley, and P. Koumoutsakos (2009), Earth curvature effects on subduction morphology: Modeling subduction in a spherical setting, *Acta Geotech.*, *4*(2), 95–105.

Moulas, E., J.-P. Burg, and Y. Podladchikov (2014), Stress field associated with elliptical inclusions in a deforming matrix: Mathematical model and implications for tectonic overpressure in the lithosphere, *Tectonophysics*, *631*, 37–49.

Patankar, S. (1980), *Numerical Heat Transfer and Fluid Flow, Comput. Methods Mech. Thermal Sci. Ser.*, 214 pp., CRC Press, Boca Raton, Fla.

Räss, L., V. M. Yarushina, N. S. Simon, and Y. Y. Podladchikov (2014), Chimneys, channels, pathway flow or water conducting features-an explanation from numerical modelling and implications for co 2 storage, *Energy Procedia*, *63*, 3761–3774.

Rutqvist, J. (2011), Status of the tough-flac simulator and recent applications related to coupled fluid flow and crustal deformations, *Comput. Geosci.*, *37*(6), 739–750.

Rutqvist, J., et al. (2006), *Multiple-Code Benchmark Simulation Study of Coupled THMC Processes in the Excavation Disturbed Zone Associated With Geological Nuclear Waste Repositories*, Lawrence Berkeley Natl. Lab., Berkeley, Calif.

Schmalholz, S. M., and R. C. Fletcher (2011), The exponential flow law applied to necking and folding of a ductile layer, *Geophys. J. Int.*, *184*(1), 83–89.

Schmeling, H., et al. (2008), A benchmark comparison of spontaneous subduction models: Towards a free surface, *Phys. Earth Planet. Inter.*, *171*(1), 198–223.

Schmid, D. W., and Y. Y. Podladchikov (2003), Analytical solutions for deformable elliptical inclusions in general shear, *Geophys. J. Int.*, *155*(1), 269–288.

Shin, D., and J. C. Strikwerda (1997), Inf-sup conditions for finite-difference approximations of the Stokes equations, *J. Aust. Math. Soc. Ser. B*, *39*(01), 121–134.

Spiegelman, M., D. A. May, and C. R. Wilson (2016), On the solvability of incompressible stokes with viscoplastic rheologies in geodynamics, *Geochem. Geophys. Geosyst.*, *17*, 2213–2238, doi:10.1002/2015GC006228.

Suckale, J., J.-C. Nave, and B. H. Hager (2010), It takes three to tango: 1. simulating buoyancy-driven flow in the presence of large viscosity contrasts, *J. Geophys. Res.*, *115*, B07409, doi:10.1029/2009JB006916.

Suckale, J., J. A. Sethian, J.-D. Yu, and L. T. Elkins-Tanton (2012), Crystals stirred up: 1. direct numerical simulations of crystal settling in nondilute magmatic suspensions, *J. Geophys. Res.*, *117*, E08004, doi:10.1029/2012JE004066.

Weatherley, S., and R. Katz (2012), Melting and channelized magmatic flow in chemically heterogeneous, upwelling mantle, *Geochem. Geophys. Geosyst.*, *13*. [10.1029/2011GC003989]

Yamato, P., R. Tartèse, T. Duretz, and D. A. May (2012a), Numerical modelling of magma transport in dykes, *Tectonophysics*, *526*, 97–109.

Yamato, P., R. Tartèse, T. Duretz, and D. May (2012b), Numerical modelling of magma transport in dykes, *Tectonophysics*, *526*, 97–109.

Yamato, P., T. Duretz, D. May, and R. Tartèse (2015), Quantifying magma segregation in dykes, *Tectonophysics*, *660*, 132–147.

Yarushina, V. M., and Y. Y. Podladchikov (2015), (de) compaction of porous viscoelastoplastic media: Model formulation, *J. Geophys. Res. Solid Earth*, *120*, 4146–4170, doi:10.1002/2014JB011258.

Yarushina, V. M., Y. Y. Podladchikov, and J. A. Connolly (2015), (de) compaction of porous viscoelastoplastic media: Solitary porosity waves, *J. Geophys. Res. Solid Earth*, *120*, 4843–4862, doi:10.1002/2014JB011260.