Geophysical Journal International

Geophys. J. Int. (2019) **218,** 873–894
Advance Access publication 2019 April 20
GJI Geodynamics and Tectonics

doi: 10.1093/gji/ggz183

Efficient and practical Newton solvers for non-linear Stokes systems in geodynamic problems

M.R.T. Fraters [1], W. Bangerth [2], C. Thieulot [1], A.C. Glerum [1,3] and W. Spakman [1,4]

[1] *Department of Earth Sciences, Utrecht University, Vening Meineszgebouw A, Princetonlaan 8a, NL-*3584 *CB UtrechtNetherlands. E-mail: menno.fraters@outlook.com*
[2] *Weber 214, Department of Mathematics, Colorado State University,* 1874 *Campus Delivery, Fort Collins, CO* 80523-1874 *CO, USA*
[3] *GFZ German Research Center for Geosciences, Albert-Einstein-Strasse* 42-46, *Building A* 46, *Room* 211, *D-*14473 *Potsdam, Germany*
[4] *Centre of Earth Evolution and Dynamics (CEED), University of Oslo, P.O. Box* 1028 *Blindern, NO-*0315 *Oslo, Norway*

Accepted 2019 April 19. Received 2019 January 21; in original form 2018 August 10

SUMMARY

Many problems in geodynamic modelling result in a non-linear Stokes problem in which the viscosity depends on the strain rate and pressure (in addition to other variables). After discretization, the resulting non-linear system is most commonly solved using a Picard fixed-point iteration. However, it is well understood that Newton's method – when augmented by globalization strategies to ensure convergence even from points far from the solution – can be substantially more efficient and accurate than a Picard solver. In this contribution, we evaluate how a straightforward Newton method must be modified to allow for the kinds of rheologies common in geodynamics. Specifically, we show that the Newton step is not actually well posed for strain rate-weakening models without modifications to the Newton matrix. We derive modifications that guarantee well posedness and that also allow for efficient solution strategies by ensuring that the top left block of the Newton matrix is symmetric and positive definite. We demonstrate the applicability and relevance of these modifications with a sequence of benchmarks and a test case of realistic complexity.

Key words: Non-linear differential equations; Numerical modelling; Dynamics and mechanics of faulting; Dynamics of lithosphere and mantle; Subduction zone processes.

1 INTRODUCTION

Geodynamics aims to understand the dynamics of processes in and on the Earth on a wide range of spatial and temporal scales, typically by connecting physical processes to geological observations through either analogue or numerical modelling. The physical basis of most numerical modelling codes in the geodynamics community are continuum mechanics conservation laws for momentum, mass and energy. A fundamental assumption that underlies most models is that we can average over the small scales at which natural materials exhibit heterogeneity, and that we can approximate the *macroscopic* material properties to obtain equations that are well understood.

When considering long enough timescales, the dynamics of the mantle – and, to some degree, the crust – can then be described as a slow-moving fluid that is governed by the Stokes equations, together with advection-diffusion equations for the temperature, chemical compositions and possibly other quantities. In the case of the Stokes equations, the fluid's effective viscosity will then depend on the material's temperature, pressure, composition and possibly other factors such as mechanical stress. Rheology – the science of determining how a material flows – is therefore of key importance to this approach. Unfortunately, the rheology of Earth materials over geological timescales is also one of the least constrained ingredients in modelling the physical processes of the solid Earth. For both philosophical and computational reasons, many studies use linear rheologies (e.g. Baumann *et al.* 2014; Pusok & Kaus 2015; Fritzell *et al.* 2016), that is, a viscosity that may depend on the external temperature and chemical composition, but not on the fluid variables velocity (or its derivatives, e.g. the strain rate) and pressure. However, experiments have shown that the rheology of Earth materials can behave in a very non-linear way (Karato & Wu 1993). Specifically, in deformation regimes, the mechanical stress leads to material weakening with increasing strain rate and consequently an effective viscosity that is a decreasing function of the strain rate. Furthermore, many widely used rheological models – in particular if they try to incorporate plastic effects – include a pressure dependence of the viscosity. This non-linearity of the rheology results in models best described by a non-linear variation of the Stokes equations. An additional source of non-linearities arises from the fact that Earth materials are compressible, that is, that their density depends on the pressure. Because there is no convenient way of solving this kind of non-linear partial differential equation exactly, it is important to develop numerical methods that can discretize and iteratively resolve the non-linearity in the equations.

A simple and frequently used way to solve such non-linear problems is to use Picard iterations, a particular form of fixed-point iterations (Kelly 1995). In it, one computes the viscosity and density as a function of the previous iteration's strain rate and pressure, solves for a new velocity and pressure field, and then repeats the process. The Picard iteration owes its popularity to the fact that it is relatively easy to implement in codes that only support linear rheologies because it only requires the repeated solution of linear problems. It is also often globally convergent, that is, with sufficiently many iterations it is possible to approximate the solution of the non-linear problem regardless of the choice of initial guess. Consequently, it is the method that is likely used in the majority of mantle convection papers that actually iteratively resolve the non-linearity in each time step; most papers do not explicitly state so, but van Keken *et al.* (2008), Tosi *et al.* (2015), Buiter *et al.* (2016) and Glerum *et al.* (2018) are some examples.

On the other hand, Picard iterations are often slow to converge, requiring dozens or hundreds of iterations for strongly non-linear problems – something we also observe in our results in Section 3. This slow convergence may make the solution of non-linear problems to high accuracy prohibitively expensive. Consequently, commonly used approaches to cope with the high computational cost are, for example, limiting the allowed number of Picard iterations per time step (e.g. Lemiale *et al.* 2008), combining Picard iterations with small time steps to ensure good starting guesses (e.g. Ruh *et al.* 2013), or other mostly *ad hoc* approaches. In practice, however, many studies do not adequately document the exact algorithm used and how this affects the accuracy of the solutions of the equations considered.

Here, we address the slow convergence of non-linear solvers by replacing the Picard solver by a Newton solver (Kelly 1995). Previous applications of Newton's method to geodynamics problems can be found in Kaus *et al.* (2015), May *et al.* (2015), Rudi *et al.* (2015) and Spiegelman *et al.* (2016). Newton's method promises quadratic convergence towards the solution, compared to the linear convergence of the Picard iteration, when the initial guess is close enough to the solution of the non-linear problem and therefore offers the prospect of vastly faster solution procedures. On the other hand, the implementation of Newton's method is substantially more involved than a Picard iteration. Furthermore, requiring an initial guess that is close enough to the exact solution is often impractical, and may require running a number of initial Picard iterations before starting the Newton iteration.

In this paper, we present the details of an improvement on the Newton method for the non-linear Stokes problem, and discuss an implementation of this improved Newton solver along with recommendations on how to use it. Specifically, and going beyond what is available in the literature, we will show that a naive application of Newton's method may break both the symmetry and the positive definiteness of the elliptic part of the (linearized) Jacobian of the Stokes operator. While the lack of symmetry is annoying from a practical perspective because it makes the solution of the linear system associated with each Newton step more complicated, a lack of positive definiteness implies that the Newton step is ill-posed and may not have a solution. We will analyse both of these issues in detail and propose modifications to the Newton equations that retain the symmetry and restore the positive definiteness. We will also consider whether there are special classes of material models where these modifications are not necessary. Unfortunately, as we will show, many rheologies that have been used extensively in the literature do not fall into these classes; our methods are therefore strict improvements over the current state of the art and will allow solving problems that were not previously solvable with an unmodified Newton method.

While there are previous reports on using a Newton method for Stokes problems in geodynamics applications (see e.g. May *et al.* 2015; Rudi *et al.* 2015; Kaus *et al.* 2015; Spiegelman *et al.* 2016), we will provide a more in-depth discussion of the mathematical properties of the operators and linear systems associated with each Newton step. We will underpin our claims with numerical experiments and demonstrate that the approach advocated herein is, indeed, more efficient and robust than previous approaches. In particular, we will show that our implementation of the Newton solver significantly decreases computational time for realistic problems, with greatly improved accuracy. Our implementation is available as open source as part of the ASPECT code (Kronbichler *et al.* 2012; Heister *et al.* 2017), an open source geodynamics community code.

The layout of the remainder of this paper is as follows: we will first describe the mathematical formulation of the non-linear Stokes problem we consider here, its discretization, and linearization in Section 2. This section also contains our main results on how the Newton method has to be modified ('stabilized') in order to make it well posed, as well as a discussion of practical aspects of how this method can be embedded in efficient non-linear and linear solvers. We then show how the above works in practice in Section 3, first using three artificial test cases and then using a realistic application of modelling subduction. We conclude in Section 4.

2 PROBLEM STATEMENT AND NUMERICAL METHODS

2.1 The model

Let us begin by concisely stating the equations we want to solve herein. We are concerned with modelling convection in the Earth mantle, a process that is typically described by a coupled system of differential equations. Under commonly used assumptions – see for example Schubert *et al.* (2001) – typical models include a Stokes-like, compressible fluid flow system for the velocity $\mathbf{u}$ and pressure p defined in the volume $\Omega \subset \mathbb{R}^d$ (where the space dimension $d = 2$ or 3) under consideration,

$$-\nabla \cdot \left[2\eta\left(\epsilon(\boldsymbol{u}) - \frac{1}{3}(\nabla \cdot \boldsymbol{u})\boldsymbol{I}\right)\right] + \nabla p = \rho \boldsymbol{g} \qquad \text{in } \Omega, \tag{1}$$

$$-\nabla \cdot (\rho \boldsymbol{u}) = 0 \qquad \text{in } \Omega, \tag{2}$$

where η is the viscosity, ρ the density, $\mathbf{g}$ the gravity vector, $\epsilon(\,\cdot\,)$ denotes the symmetric gradient operator defined by $\varepsilon(\mathbf{v}) = \frac{1}{2}(\nabla\mathbf{v} + \nabla\mathbf{v}^{\mathsf{T}})$ and I is the $d \times d$ identity matrix. (The sign in eq. (2) is chosen in this way because $-\nabla\cdot$ is the adjoint operator to the gradient in the first equation, leading to a symmetric system if the density is constant, as shown below.)

While these equations describe a compressible model, we will assume for the purposes of this paper that the fluid is in fact incompressible, that is, that $\nabla\cdot(\rho\boldsymbol{u}) = \rho\nabla\cdot\boldsymbol{u} = 0$. We do so because we can illustrate all difficulties associated with the Newton method using this simplification already, and because many of the approximations used in geodynamics (e.g. the Boussinesq approximation) also assume incompressibility. In addition to this simplification, we have to scale the equations to ensure that we can numerically compare the residuals of the two equations and consequently have a basis for numerically stable algorithms. Consequently, we multiply the second equation by a constant $s_p = \frac{\eta_0}{L}$ where η_0 is a 'reference viscosity' and L a length scale of the domain we are solving the equations in. (See Kronbichler *et al.* 2012 for a more detailed discussion.) In order to retain the symmetry between the divergence in the second equation and the gradient in the first, we also replace the pressure by a scaled version, $\tilde{p} = \frac{1}{s_p}p$. The properly scaled, incompressible equations then read as follows:

$$-\nabla\cdot[2\eta\epsilon(\boldsymbol{u})] + s_p\nabla\tilde{p} = \rho\boldsymbol{g}, \tag{3}$$

$$-s_p\nabla\cdot\boldsymbol{u} = 0. \tag{4}$$

It is this form of the equations we will attempt to solve, using $\boldsymbol{u}$, $\tilde{p}$ as the primary variables. Of course, the physical pressure can be recovered as $p = s_p\tilde{p}$ after the system has been solved.

In geodynamic models, the fluid flow model is coupled to an equation for the temperature T,

$$\begin{aligned}\rho C_p\left(\frac{\partial T}{\partial t} + \boldsymbol{u}\cdot\nabla T\right) - \nabla\cdot k\nabla T = \rho H\\ +2\eta\left(\epsilon(\boldsymbol{u}) - \frac{1}{3}(\nabla\cdot\boldsymbol{u})\boldsymbol{I}\right) : \left(\epsilon(\boldsymbol{u}) - \frac{1}{3}(\nabla\cdot\boldsymbol{u})\boldsymbol{I}\right)\\ +\alpha_T T\,(\boldsymbol{u}\cdot\nabla p)\\ +\rho T\Delta S\left(\frac{\partial X}{\partial t} + \boldsymbol{u}\cdot\nabla X\right) \qquad \text{in } \Omega,\end{aligned} \tag{5}$$

and possibly other equations that describe the transport of chemical compositions. Here, C_p is the specific heat, α_T is the thermal expansion coefficient, k the thermal conductivity, H is the internal heat production and ΔS and X are related to the entropic effects of phase changes. All coefficients that appear in these equations typically depend on the pressure, temperature, chemical composition and – in the case of the viscosity – the strain rate $\varepsilon(\boldsymbol{u})$.

Even though the *entire* system is coupled in non-linear ways, in this paper, we will only concern ourselves with the first set of these equations, (3) and (4), and how they can efficiently be solved through a Newton scheme. In principle, one may want to solve the entire system with a Newton scheme, given that the velocity appears in eq. (5), the temperature in eqs (3) and (4) via the temperature dependence of the viscosity and density, and more generally all coefficients may depend on pressure and temperature. While this is beyond the scope of the current paper, being able to apply a Newton method to the Stokes subsystem is clearly a necessary ingredient to the larger goal. Consequently, the efficient solution of non-linear Stokes problems is of interest in itself. As we will show, this alone is not trivial, and will therefore serve as a worthwhile target for the investigations in this paper. In fact, the incompressible formulation already poses all of the mathematical difficulties we will encounter in deriving well-posed Newton schemes. In other words, it serves as a good model problem to illustrate and understand both difficulties and solutions related to the linearization. The incorporation of compressible terms (i.e. solving eqs 1 and 2) would then only complicate the exposition of our methods. At the same time, we point out that our methods immediately carry over to compressible models – albeit with significantly more cumbersome formulae; we will investigate this generalization in future work.

2.2 Discretization

We convert eqs (3) and (4) above into a finite-dimensional system by utilizing the finite-element method for discretization. To this end, we seek approximations

$$\boldsymbol{u}_h(\boldsymbol{x}) = \sum_j U_j\boldsymbol{\varphi}^{\boldsymbol{u}}_j(\boldsymbol{x}) \tag{6}$$

$$\tilde{p}_h(\boldsymbol{x}) = \sum_j \tilde{P}_j\varphi^p_j(\boldsymbol{x}) \tag{7}$$

where $\boldsymbol{\varphi}^{\boldsymbol{u}}_j$ and φ^p_j are the finite-element basis functions for the velocity and pressure, respectively.

The expansion coefficients U_j, $\tilde{P}_j$ are found by solving the discrete weak form of the equations. Discretization of the incompressible system then leads to a non-linear system in $\boldsymbol{X} = (\boldsymbol{U}, \tilde{\boldsymbol{P}})$,

$$\boldsymbol{Q}(\boldsymbol{X})\,\boldsymbol{X} = \boldsymbol{b}(\boldsymbol{X}), \tag{8}$$

where the matrix $\boldsymbol{Q}$ and right-hand side $\boldsymbol{b}$ have an internal substructure. For our incompressible formulation, this substructure has the form

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{B}^{\mathsf{T}} \\ \boldsymbol{B} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{U} \\ \tilde{\boldsymbol{P}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{h} \end{pmatrix}. \tag{9}$$

Here, the matrix and right-hand side blocks are defined as

$$A_{ij} = (\varepsilon(\boldsymbol{\varphi}_i^{\boldsymbol{u}}), 2\eta\, \varepsilon(\boldsymbol{\varphi}_j^{\boldsymbol{u}})), \qquad B_{ij} = -s_p(\varphi_i^q, \nabla \cdot \boldsymbol{\varphi}_j^{\boldsymbol{u}}), \tag{10}$$

$$f_i = (\boldsymbol{\varphi}_i^{\boldsymbol{u}}, \rho \boldsymbol{g}), \qquad h_i = 0, \tag{11}$$

where as usual we denote $(\alpha, \beta) = \int_\Omega \alpha(\boldsymbol{x})\, \beta(\boldsymbol{x})\, \mathrm{d}x$. Because the viscosity η may depend on the pressure and strain rate, and the density ρ on the pressure, the system is in general non-linear in the coefficients U_j, $\tilde{P}_j$ as both $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{X})$ and $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{X})$. (The coefficients η, ρ may of course also depend on the temperature or other factors, but we consider these fixed for the purposes of the current paper.)

Much of the content of this paper is concerned with the question of how to solve the non-linear system (8) *in practice*, that is, how a naive application of the standard Newton iteration solver needs to be adapted to make it practical and efficient.

2.3 Newton linearization

In order to resolve the non-linearity in eq. (8), let us introduce the residual $\boldsymbol{r}(\boldsymbol{X}) = \boldsymbol{Q}(\boldsymbol{X})\boldsymbol{X} - \boldsymbol{b}(\boldsymbol{X})$. In Newton iteration $k+1$, starting with the previous guess $\boldsymbol{X}_k$, we then need to solve

$$\boldsymbol{J}_k\, \delta \boldsymbol{X}_k = -\boldsymbol{r}_k \tag{12}$$

where $\boldsymbol{r}_k = \boldsymbol{r}(\boldsymbol{X}_k)$ and $\boldsymbol{J}_k = \nabla_{\boldsymbol{X}} \boldsymbol{r}(\boldsymbol{X}_k)$. This system has the internal substructure

$$\begin{pmatrix} \boldsymbol{J}_k^{uu} & \boldsymbol{J}_k^{up} \\ \boldsymbol{J}_k^{pu} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \delta \boldsymbol{U}_k \\ \delta \tilde{\boldsymbol{P}}_k \end{pmatrix} = - \begin{pmatrix} \boldsymbol{r}_k^{\boldsymbol{u}} \\ \boldsymbol{r}_k^{p} \end{pmatrix}. \tag{13}$$

After solving for $\delta \boldsymbol{X}_k$, we can compute $\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + \alpha_k\, \delta \boldsymbol{X}_k$ where α_k is a step length parameter that can be determined, for example, using a line search (Kelly 1995; Nocedal & Wright 1999).

There are a number of approaches to determining the entries of the matrix $\boldsymbol{J}_k$ and to solving the resulting linear system. For example, in the geodynamics community alone, May *et al.* (2015) and Kaus *et al.* (2015) make use of a Jacobian-free Newton–Krylov framework (see Knoll & Keyes 2004), which essentially computes a finite-difference approximation of $\boldsymbol{J}$ by evaluating $\boldsymbol{r}$ at different values of its argument, and integrates this directly into the solver so that the full Jacobian matrix is never built. On the other hand, Rudi *et al.* (2015) and Spiegelman *et al.* (2016) use the same approach as we will take here and compute derivatives analytically or semi-analytically, except that Rudi *et al.* (2015) implemented this in a Jacobian-free manner.

Regardless of how exactly these derivatives are computed, the blocks of the linear system for the Newton updates will have to have the following form (again omitting dependencies on quantities we consider frozen, such as the temperature):

$$\begin{aligned} \left(\boldsymbol{J}_k^{uu}\right)_{ij} &= \frac{\partial}{\partial U_j} \left(\boldsymbol{A}_k \boldsymbol{U}_k + \boldsymbol{B}^{\mathsf{T}} \tilde{\boldsymbol{P}}_k - \boldsymbol{f}_k\right)_i \\ &= (\boldsymbol{A}_k)_{ij} + \left(\boldsymbol{\varepsilon}(\boldsymbol{\varphi}_i^{\boldsymbol{u}}), 2\left(\frac{\partial \eta(\boldsymbol{\varepsilon}(\boldsymbol{u}_k), p_k)}{\partial \varepsilon} : \boldsymbol{\varepsilon}(\boldsymbol{\varphi}_j^{\boldsymbol{u}})\right)\boldsymbol{\varepsilon}(\boldsymbol{u}_k)\right), \end{aligned} \tag{14}$$

$$\begin{aligned} \left(\boldsymbol{J}_k^{up}\right)_{ij} &= \frac{\partial}{\partial \tilde{P}_j} \left(\boldsymbol{A}_k \boldsymbol{U}_k + \boldsymbol{B}^{\mathsf{T}} \tilde{\boldsymbol{P}}_k - \boldsymbol{f}_k\right)_i \\ &= B_{ij}^{\mathsf{T}} + \left(\boldsymbol{\varepsilon}(\boldsymbol{\varphi}_i^{\boldsymbol{u}}), 2\left(\frac{\partial \eta(\boldsymbol{\varepsilon}(\boldsymbol{u}_k), p_k)}{\partial \tilde{p}} \varphi_j^p\right)\varepsilon(\boldsymbol{u}_k)\right), \\ &= B_{ij}^{\mathsf{T}} + \left(\varepsilon(\boldsymbol{\varphi}_i^{\boldsymbol{u}}), 2\left(\frac{\partial \eta(\varepsilon(\boldsymbol{u}_k), p_k)}{\partial p} \frac{\partial p}{\partial \tilde{p}} \varphi_j^p\right)\varepsilon(\boldsymbol{u}_k)\right), \\ &= B_{ij}^{\mathsf{T}} + s_p \left(\varepsilon(\boldsymbol{\varphi}_i^{\boldsymbol{u}}), 2\left(\frac{\partial \eta(\varepsilon(\boldsymbol{u}_k), p_k)}{\partial p} \varphi_j^p\right)\varepsilon(\boldsymbol{u}_k)\right), \end{aligned} \tag{15}$$

$$\begin{aligned} \left(\boldsymbol{J}_k^{pu}\right)_{ij} &= \frac{\partial}{\partial U_j} \left(\boldsymbol{B} \boldsymbol{U}_k - \boldsymbol{h}_k\right)_i \\ &= B_{ij}. \end{aligned} \tag{16}$$

It is easy to see that – as expected – the Newton system (13) reverts to the simple Stokes problem if the viscosity does not depend on strain rate or pressure, that is, if the system is linear.

As we will show below, while eq. (12) (and its block structure 13) is the correct linearization of the (discretized) original, non-linear system (3) and (4), it turns out that this does not necessarily lead to a well-posed problem. This is not uncommon in optimization problems where a function $f(\boldsymbol{x})$ may have a well-defined minimizer, but the Hessian matrix $\boldsymbol{H}_k = \nabla^2 f(\boldsymbol{x}_k)$ at early iterates may be singular or have negative eigenvalues; consequently, the solution of the linear system $\boldsymbol{H}_k\, \delta\boldsymbol{x}_k = -\nabla f(\boldsymbol{x}_k)$ may not have a solution $\delta\boldsymbol{x}_k$ or the solution may not be a direction of descent. There are standard techniques described in the optimization literature for these cases (see e.g. the section on 'Hessian modification' methods in Nocedal & Wright 1999) that we will adapt in the following sections, though we will work at the level of the partial differential equations that give rise to the Newton matrix, rather than at the algebraic level of the matrix we wish to modify. Furthermore, the linear system we obtain in each Newton step may be difficult to solve for practical reasons if it is not symmetric.

We will therefore discuss the practical implications of Newton linearization in Sections 2.4 and 2.5 below, along with remedies to the problems we identify. It is important to stress that the modifications we propose only change the matrix $\boldsymbol{J}_k$ in eq. (12) but not the right-hand side. As a consequence, we can hope that the iterates $\boldsymbol{X}_k$ still converge to the correct solution $\boldsymbol{X}$ of eq. (8), and this is indeed the case in our numerical experiments as we observe that $\|\boldsymbol{r}_k\| \to 0$ as the iterations proceed. In other words, we replace an exact (though potentially ill-posed) Newton iteration by an approximate (and well-posed) Newton iteration, but we continue to solve the original physical problem.

2.4 Restoring symmetry of $\boldsymbol{J}^{\boldsymbol{uu}}$

Even for incompressible models, given the form of the individual blocks in eqs (14)–(16), the Newton system (13) is in general not symmetric. This is despite the fact that the matrix $\boldsymbol{Q}$ in the non-linear model (8) and in particular $\boldsymbol{A}$ in eq. (9) are of course symmetric, as shown in eq. (10).

On the other hand, symmetry of matrices is an important property from a practical perspective because it allows for the construction of efficient solvers and pre-conditioners. As a consequence, we advocate replacing eq. (12) by an approximation. This of course yields a different Newton update $\delta\boldsymbol{x}_k$ and may destroy the quadratic convergence order of the Newton method. On the other hand, we retain our ability to construct efficient solvers and pre-conditioners; in practice, one does not often run a large number of Newton iterations in each time step, and consequently a reduction from quadratic to possibly only superlinear convergence order may be acceptable. As pointed out above, we do not modify the right-hand side of the Newton update equation and consequently converge to the solution of the original non-linear problem.

Specifically, then, we advocate for the following approximation of eq. (14):

$$\left(\boldsymbol{J}_k^{uu}\right)_{ij} \approx (\boldsymbol{A}_k)_{ij} + \left(\varepsilon(\boldsymbol{\varphi}_i^{\boldsymbol{u}}), \left(\frac{\partial \eta(\varepsilon(\boldsymbol{u}_k),\, p_k)}{\partial \varepsilon} : \varepsilon(\boldsymbol{\varphi}_j^{\boldsymbol{u}})\right)\varepsilon(\boldsymbol{u}_k)\right) + \left(\varepsilon(\boldsymbol{\varphi}_j^{\boldsymbol{u}}), \left(\frac{\partial \eta(\varepsilon(\boldsymbol{u}_k),\, p_k)}{\partial \varepsilon} : \varepsilon(\boldsymbol{\varphi}_i^{\boldsymbol{u}})\right)\varepsilon(\boldsymbol{u}_k)\right).$$

This approximation ensures that the top left block in eq. (13) is indeed symmetric, and as we will see below, this and the modification discussed in the next section will then allow for the construction of efficient, multigrid-based pre-conditioners and the use of the Conjugate Gradient method. Indeed, the modification simply symmetrizes the second term in eq. (14). In order to analyse the effect of the underlying approximation, it is useful to rewrite the original term in eq. (14) in sum notation:

$$\begin{aligned}
&\left(\varepsilon(\boldsymbol{\varphi}_i^{\boldsymbol{u}}), 2\left(\frac{\partial \eta(\varepsilon(\boldsymbol{u}_k),\, p_k)}{\partial \varepsilon} : \varepsilon(\boldsymbol{\varphi}_j^{\boldsymbol{u}})\right)\, \varepsilon(\boldsymbol{u}_k)\right) \\
&\qquad = \int_\Omega \sum_{mn} \varepsilon(\boldsymbol{\varphi}_i^{\boldsymbol{u}})_{mn} \left[\sum_{pq} 2\frac{\partial \eta(\varepsilon(\boldsymbol{u}_k),\, p_k)}{\partial \varepsilon_{pq}} \varepsilon(\boldsymbol{\varphi}_j^{\boldsymbol{u}})_{pq}\right] \varepsilon(\boldsymbol{u}_k)_{mn} \\
&\qquad = \int_\Omega \sum_{mn,pq} \varepsilon(\boldsymbol{\varphi}_i^{\boldsymbol{u}})_{mn}\, E(\varepsilon(\boldsymbol{u}_k))_{mnpq}\, \varepsilon(\boldsymbol{\varphi}_j^{\boldsymbol{u}})_{pq}
\end{aligned}$$

where the rank-4 tensor E is defined as $E(\varepsilon(\boldsymbol{u}))_{mnpq} = \left[2\varepsilon(\boldsymbol{u})_{mn} \frac{\partial \eta(\varepsilon(\boldsymbol{u}),\, p)}{\partial \varepsilon_{pq}}\right]$. Clearly, the matrix $\boldsymbol{J}_k^{uu}$ is symmetric if the tensor E is symmetric, that is, $E_{mnpq} = E_{pqmn}$, but this is not always the case. (By its definition, we already have $E_{mnpq} = E_{nmpq} = E_{mnqp}$.) The modification we propose is equivalent to explicitly symmetrizing this tensor, that is, replacing E_{mnpq} by $E^{\text{sym}}_{mnpq} = \frac{1}{2}\left(E_{mnpq} + E_{pqmn}\right)$ and replacing the matrix in eq. (14) by

$$\left(\boldsymbol{J}_k^{uu}\right)_{ij} = (\boldsymbol{A}_k)_{ij} + \left(\varepsilon(\boldsymbol{\varphi}_i^{\boldsymbol{u}}),\, E^{\text{sym}}(\varepsilon(\boldsymbol{u}_k))\, \varepsilon(\boldsymbol{\varphi}_j^{\boldsymbol{u}})\right). \tag{17}$$

It is instructive to consider whether there are cases in which the tensor E is *already* symmetric, and replacing it by its symmetrized version consequently does not change anything. Specifically, this is the case if the viscosity $\eta(\varepsilon(\boldsymbol{u}))$ can be written as a scalar function of the square of the strain rate, that is, $\eta(\varepsilon(\boldsymbol{u})) = f(\|\varepsilon(\boldsymbol{u})\|^2)$ where $\|\varepsilon\|^2 = \sum_{ij} \varepsilon_{ij}^2$. In this case, the chain rule implies that

$$\frac{\partial \eta(\varepsilon(\boldsymbol{u}))}{\partial \varepsilon_{pq}} = f'(\|\varepsilon(\boldsymbol{u})\|^2) \frac{\partial \|\varepsilon(\boldsymbol{u})\|^2}{\partial \varepsilon_{pq}} = 2 f'(\|\varepsilon(\boldsymbol{u})\|^2)\, \varepsilon(\boldsymbol{u})_{pq}.$$

We then have that $E(\varepsilon(\boldsymbol{u}))_{mnpq} = 4 f'(\|\varepsilon(\boldsymbol{u})\|^2) \varepsilon(\boldsymbol{u}_k)_{mn} \varepsilon(\boldsymbol{u})_{pq}$, which satisfies the desired symmetry condition.

Furthermore, for incompressible materials, we have that trace $\varepsilon(\boldsymbol{u}) = \operatorname{div} \boldsymbol{u} = 0$, and in that case, the second invariant of the strain rate can be simplified to $\mathbb{I}_2(\varepsilon(\boldsymbol{u})) = \frac{1}{2}\left[(\text{trace}\ \varepsilon(\boldsymbol{u}))^2 - \text{trace}\ (\varepsilon(\boldsymbol{u})^2)\right] = -\frac{1}{2}\text{trace}\ (\varepsilon(\boldsymbol{u})^2) = -\frac{1}{2}\|\varepsilon(\boldsymbol{u})\|^2$. In other words, for incompressible materials, the second invariant is a function of the square of the norm of the strain rate, and consequently any material model that only depends

on the second invariant then also satisfies the criteria for cases where the explicit symmetrization does not actually change anything. Indeed, many incompressible material models define the viscosity only in terms of the second invariant of the strain rate, see, for example, Schellart & Moresi (2013). [We note that the geodynamics literature uses varying definitions for the second invariant. In contrast to the one used above, some papers use the definition $\mathbb{I}_2(\varepsilon(\boldsymbol{u})) = \left(\frac{1}{2}\varepsilon(\boldsymbol{u}) : \varepsilon(\boldsymbol{u})\right)^{1/2} = \left(\frac{1}{2}\|\varepsilon(\boldsymbol{u})\|^2\right)^{1/2}$ – see, for example, Gerya (2010, p. 56) or May *et al.* (2015). However, even with this convention the second invariant is a function of the square of the norm of the strain rate, and the conclusion above about material models that are functions of only the second invariant of the strain rate remains valid.]

We end by pointing out that the entire Jacobian remains non-symmetric since, in general, $\boldsymbol{J}^{up} \neq (\boldsymbol{J}^{pu})^{\mathsf{T}}$ because of the added term due to the derivative of the viscosity with regard to the pressure (see eqs 15 and 16). We will come back to this in Section 2.6.2.

2.5 Restoring well posedness of the Newton step

The Stokes-like system (13) that arises from Newton linearization can only be well posed if the top left block is invertible. However, it turns out that this is not always the case, as we will see shortly. It is important to realize, however, that a lack of well posedness of the Newton step is not equivalent to a lack of well posedness of the original, non-linear problem from which it arises. Indeed, it is easy to conceive of situations where a Newton method applied to finding solutions of 1-D equations $f(x) = 0$ fails because one of the intermediate iterates x_k happens to land at a location where $f'(x_k) = 0$ and the next iteration fails because there is no δx_k so that $f'(x_k)\delta x_k = -f(x_k)$. In multiple dimensions, and in particular in the case of the infinite-dimensional operator from which the top left matrix block $\boldsymbol{J}^{uu}$ is derived, the situation is clearly more complex, but not much more complicated to understand.

To this end, recall that after the symmetrization discussed in the previous section, the matrix $\boldsymbol{J}^{uu}$ has entries

$$\begin{aligned}(\boldsymbol{J}^{uu})_{ij} &= \left(\varepsilon(\boldsymbol{\varphi}_i^{\boldsymbol{u}}), 2\eta(\varepsilon(\mathbf{u}))\varepsilon(\boldsymbol{\varphi}_j^{\boldsymbol{u}})\right) + \left(\varepsilon(\boldsymbol{\varphi}_i^{\boldsymbol{u}}), E^{\text{sym}}(\varepsilon(\mathbf{u}))\varepsilon(\boldsymbol{\varphi}_j^{\boldsymbol{u}})\right)\\ &= \left(\varepsilon(\boldsymbol{\varphi}_i^{\boldsymbol{u}}), \underbrace{\left[2\eta(\varepsilon(\mathbf{u}))\boldsymbol{I}\otimes\boldsymbol{I} + E^{\text{sym}}(\varepsilon(\mathbf{u}))\right]}_{=:H}\varepsilon(\boldsymbol{\varphi}_j^{\boldsymbol{u}})\right),\end{aligned}$$

where the rank-4 tensor $(\boldsymbol{I}\otimes\boldsymbol{I})_{ijkl} = \delta_{ik}\delta_{jl}$ maps a symmetric rank-2 tensor onto itself. A sufficient (though not necessary) condition for the matrix $\boldsymbol{J}^{uu}$ to be invertible (i.e. to have no zero eigenvalues) is if the corresponding differential operator, $-\nabla\cdot[H\varepsilon(\bullet)]$ is elliptic. This is the case if and only if the tensor H (as a map from rank-2 symmetric tensors to rank-2 symmetric tensors) has only positive eigenvalues, that is, if $\varepsilon : (H\varepsilon) > 0$ for all symmetric, non-zero rank-2 tensors ε. (We provide a bit more mathematical background for this connection between the coefficient H and the ellipticity of the corresponding differential equation in **Appendix A.)

Informally, for a strain hardening material model, $\frac{\partial\eta(\varepsilon(\boldsymbol{u}))}{\partial\varepsilon}$ is positive, and then so is $2\eta(\varepsilon(u))\boldsymbol{I}\otimes\boldsymbol{I} + E^{\text{sym}}(\varepsilon(\boldsymbol{u}))$ because E^{sym} is a *positive* correction to the already positive-definite tensor $2\eta\boldsymbol{I}\otimes\boldsymbol{I}$. In other words, H would then be positive as would the differential operator, and $\boldsymbol{J}^{uu}$ would be an invertible matrix. The same would be true if the material model is strain weakening and if the amount of weakening is 'small enough' because then the 'small correction' E^{sym} does not offset the positive definiteness of $2\eta\boldsymbol{I}\otimes\boldsymbol{I}$. That said, we will need to be more formal with arguments as we are dealing with tensors instead of scalars; the remainder of the section is therefore devoted to formalizing these arguments and providing a solution to the problem.

Specifically, given the definitions above, the tensor H can be written as

$$\begin{aligned}H &= 2\eta(\varepsilon(\mathbf{u}))\boldsymbol{I}\otimes\boldsymbol{I} + E^{\text{sym}}(\varepsilon(\mathbf{u}))\\ &= 2\eta(\varepsilon(\mathbf{u}))\boldsymbol{I}\otimes\boldsymbol{I} + \varepsilon(\boldsymbol{u})\otimes\frac{\partial\eta(\varepsilon(\boldsymbol{u}),p)}{\partial\varepsilon} + \frac{\partial\eta(\varepsilon(\boldsymbol{u}),p)}{\partial\varepsilon}\otimes\varepsilon(\boldsymbol{u}),\end{aligned}$$

that is, H is a rank-2 update of a multiple of the identity operator. The first of these three terms has all eigenvalues equal to 2η, and the other two terms then lead to a perturbation of two of these eigenvalues corresponding to eigendirections that are spanned by $\varepsilon(\boldsymbol{u})$ and $\frac{\partial\eta(\varepsilon(\boldsymbol{u}),p)}{\partial\varepsilon}$. As mentioned above, unless a material's strain weakening rate is sufficiently small, these perturbations may be strong enough to make one or both of the perturbed eigenvalues negative, and in this case the Newton step fails to be well posed.

To avoid this, we introduce a tensor

$$\begin{aligned}H^{\text{spd}} &= 2\eta(\varepsilon(\mathbf{u}))\boldsymbol{I}\otimes\boldsymbol{I} + \alpha E^{\text{sym}}(\varepsilon(\mathbf{u}))\\ &= 2\eta(\varepsilon(\boldsymbol{u}))\boldsymbol{I}\otimes\boldsymbol{I} + \alpha\left[\varepsilon(\boldsymbol{u})\otimes\frac{\partial\eta(\varepsilon(\boldsymbol{u}),p)}{\partial\varepsilon} + \frac{\partial\eta(\varepsilon(\boldsymbol{u}),p)}{\partial\varepsilon}\otimes\varepsilon(\boldsymbol{u})\right],\end{aligned}$$

where $0 < \alpha \leq 1$ is chosen in such a way that H^{spd} is positive definite. Using this modified form of H^{spd} at every quadrature point at which we perform the integration of the bilinear form for the Newton matrix, we then build the matrix $\boldsymbol{J}^{uu}$ used in the iteration. As before, since we do not change the right-hand side of the Newton update equation, we converge to the solution of the original non-linear problem.

Clearly, if $\alpha = 0$, then H^{sym} is the identity operator times 2η and has positive eigenvalues. Because the eigenvalues depend continuously on α, there must be an $\alpha > 0$ so that H^{spd} is indeed positive definite. Ideally, to retain the convergence rate of Newton's method, we would like to choose $\alpha = 1$. We therefore propose the following choice: we want to choose α so that (i) we have $\alpha = 1$ if H is already positive definite, and (ii) α is as large as possible so that H^{spd} is positive definite. In practice, however, we will also choose α small enough to avoid the case where one of the eigenvalues of H^{spd} is positive but very small compared to 2η, to avoid the numerical difficulties resulting from trying to solve a linear problem with a poorly conditioned matrix $\boldsymbol{J}^{uu}$.

It turns out that we can use the rank-2 update form of H and H^{spd} to explicitly compute the value of α. Let us abbreviate $E^{\text{sym}} = a\otimes b + b\otimes a$ where $a = \varepsilon(\boldsymbol{u})$ and $b = \frac{\partial \eta(\varepsilon(\boldsymbol{u}),p)}{\partial \varepsilon}$. Then, it is clear that the (non-trivial) eigenvectors of E^{sym} must lie in the plane spanned by a, b, that is, have the form $v = \cos(\theta)\frac{a}{\|a\|} + \sin(\theta)\frac{b}{\|b\|}$. The two non-trivial eigenvalues of E^{sym} are then the extremal values of the Rayleigh quotient

$$
\begin{aligned}
R(\alpha) &= v : (E^{\text{sym}} : v) \\
&= \left[\cos(\theta)\frac{a}{\|a\|} + \sin(\theta)\frac{b}{\|b\|}\right] : \left([a\otimes b + b\otimes a] : \left[\cos(\theta)\frac{a}{\|a\|} + \sin(\theta)\frac{b}{\|b\|}\right]\right) \\
&= 2\left[\cos(\theta)\|a\| + \sin(\theta)\frac{b:a}{\|b\|}\right]\left[\cos(\theta)\frac{b:a}{\|a\|} + \sin(\theta)\|b\|\right] \\
&= 2\left[(b:a)\cos(\theta)^2 + \left(\frac{(b:a)^2}{\|b\|\|a\|} + \|a\|\|b\|\right)\sin(\theta)\cos(\theta) + (b:a)\sin(\theta)^2\right] \\
&= 2\left[(b:a) + \left(\frac{(b:a)^2}{\|b\|\|a\|} + \|a\|\|b\|\right)\sin(\theta)\cos(\theta)\right] \\
&= 2\left[(b:a) + \frac{1}{2}\left(\frac{(b:a)^2}{\|b\|\|a\|} + \|a\|\|b\|\right)\sin(2\theta)\right] \\
&= 2\left[\frac{b:a}{\|a\|\|b\|} + \frac{1}{2}\left(\frac{(b:a)^2}{\|b\|^2\|a\|^2} + 1\right)\sin(2\theta)\right]\|a\|\|b\|.
\end{aligned}
$$

Thus, the eigenvalues of E^{sym} are given by $\left[2\frac{b:a}{\|a\|\|b\|} \pm \left(\frac{(b:a)^2}{\|b\|^2\|a\|^2} + 1\right)\right]\|a\|\|b\|$. In other words, there is one positive eigenvalue $\lambda_{\max}(E^{\text{sym}}) = \left[1 + \frac{b:a}{\|a\|\|b\|}\right]^2\|a\|\|b\|$ and one negative or zero eigenvalue $\lambda_{\min}(E^{\text{sym}}) = -\left[1 - \frac{b:a}{\|a\|\|b\|}\right]^2\|a\|\|b\|$.

The only eigenvalue of H^{spd} we have to worry about becoming negative is therefore the one associated with the (possibly) negative eigenvalue of E^{sym}, that is, $2\eta(\epsilon(\boldsymbol{u})) - \alpha\left[1 - \frac{b:a}{\|a\|\|b\|}\right]^2\|a\|\|b\|$. This implies that we can choose the damping factor α as follows to ensure positive semidefiniteness:

$$
\alpha = \begin{cases} 1 & \text{if } \left[1 - \dfrac{b:a}{\|a\|\|b\|}\right]^2\|a\|\|b\| < 2\eta(\varepsilon(\boldsymbol{u})) \\ \dfrac{2\eta(\varepsilon(\boldsymbol{u}))}{\left[1 - \dfrac{b:a}{\|a\|\|b\|}\right]^2\|a\|\|b\|} & \text{otherwise.} \end{cases}
$$

In practice, we would like to stay well away from a zero eigenvalue and instead choose α as follows:

$$
\alpha = \begin{cases} 1 & \text{if } \left[1 - \dfrac{b:a}{\|a\|\|b\|}\right]^2\|a\|\|b\| < c_{\text{safety}}\, 2\eta(\varepsilon(\boldsymbol{u})) \\ c_{\text{safety}}\dfrac{2\eta(\varepsilon(\boldsymbol{u}))}{\left[1 - \dfrac{b:a}{\|a\|\|b\|}\right]^2\|a\|\|b\|} & \text{otherwise,} \end{cases} \tag{18}
$$

where $0 \le c_{\text{safety}} < 1$ is a safety factor that ensures that the smaller eigenvalue of H^{spd} is at least $(1 - c_{\text{safety}})2\eta$ and thus bounded away from zero. This computation is easily performed at every quadrature point during the assembly of $\boldsymbol{J}^{uu}$. This procedure then guarantees that the resulting matrix is symmetric and positive definite, implying that the Newton direction is well defined.

It is again instructive to consider whether there are cases where we can always choose $\alpha = 1$, that is, use the unmodified Newton step (possibly up to the symmetrization discussed in the previous section). The simplest case is if a: $b = \|a\|\|b\|$ because in that case the definition of α in eq. (18) always ends up in the first branch, regardless of the size of $\eta(\varepsilon(\boldsymbol{u}))$. Given the definition of a, b, this is specifically the case if $\frac{\partial \eta(\varepsilon(\boldsymbol{u}),p)}{\partial \varepsilon}$ is a *positive* multiple of $\varepsilon(\boldsymbol{u})$. Similarly to the discussion in the previous section, this is the case if $\eta(\varepsilon(\boldsymbol{u})) = f(\|\varepsilon(\boldsymbol{u})\|^2)$ and if $f' \ge 0$, that is, for a strain-hardening material. It is not difficult to show that this extends to the case where the viscosity is given by a non-decreasing function $\eta(\varepsilon(\boldsymbol{u})) = f(\|P\varepsilon(\boldsymbol{u})\|^2)$ where P is an orthogonal projection applied to the strain rate; an example is the operator that extracts the deviatoric component of the strain rate.

A more interesting case is where the material exhibits strain weakening. In that case, intuitively the conditions in eq. (18) imply that we can only choose $\alpha = 1$ if the material 'weakens slowly enough'. Let us, for example, consider the class of materials for which $\eta(\varepsilon(\boldsymbol{u})) = \eta_0\,[\mathbb{I}_2(\varepsilon(\boldsymbol{u}))]^{\frac{1}{n}-1}$. Such laws are typically used to describe either diffusion ($n = 1$) or dislocation creep ($n > 1$), see Karato (2012). Indeed, we show in Appendix B that in these cases one has to *always* choose $\alpha < 1$ if n exceeds a certain threshold.

2.6 Algorithms for the solution of the non-linear problem

The discussions of the previous sections show that a naive application of Newton's method may lead to matrices that are neither symmetric nor positive definite. Indeed, in some cases the equations for the Newton update may not be well posed at all (see e.g. the discussion in Appendix B), even if the original, non-linear model has all of these properties.

The remedies outlined above restore symmetry, positive definiteness and well posedness, and consequently lend themselves for a practical implementation. On the other hand, the resulting equations for the update are different from the ones obtained by linearizing the residual, and consequently we may not be able to expect quadratic convergence of the resulting non-linear iteration. Indeed, this is what we will observe in the experiments we show in Section 3. Regardless, the modifications have to be incorporated into an actual algorithm to solve the non-linear problem. The algorithm we propose for this – which is also the one implemented in the ASPECT code (Kronbichler *et al.* 2012; Heister *et al.* 2017) – is therefore outlined below. As for many other non-linear problems, it is not easy to universally achieve convergence, and the resulting algorithm is therefore complicated.

2.6.1 Non-linear iteration

As with many other non-linear problems, it is not generally possible to solve the non-linear Stokes equation we consider here using only a Newton iteration. Rather, we use a strategy where we use the following sequence to solve the non-linear Stokes problem in each time step:

(i) We always use one initial Picard step. That is, we solve the original Stokes equations in which we 'freeze' all coefficient using values for the strain rate and pressure extrapolated from previous time steps; this corresponds to solving $\boldsymbol{Q}(\tilde{\boldsymbol{X}})\boldsymbol{X}_1 = \boldsymbol{b}(\tilde{\boldsymbol{X}})$ (in analogy to eq. 8) where $\tilde{\boldsymbol{X}}$ is the extrapoled solution. This allows us, in particular, to enforce the correct boundary conditions on all boundaries where the velocity is prescribed.

(ii) We then solve $N_{\mathrm{DC}} \geq 0$ steps using the Picard method written in defect correction (DC) form. This corresponds to eq. (12) if one were to omit all terms that contain derivatives of η in the definition of the blocks in eqs (14)–(16). Equivalently, this corresponds to solving an update form of eq. (8), namely $\boldsymbol{Q}(\boldsymbol{X}_k)\,\delta\boldsymbol{X}_k = \boldsymbol{b}(\boldsymbol{X}) - \boldsymbol{Q}(\boldsymbol{X}_k)\,\boldsymbol{X}_k = -\boldsymbol{r}_k$ followed by computing $\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + \delta\boldsymbol{X}_k$. It is well known that the Picard iteration is more stable than a pure Newton method and often converges even in cases where Newton's method does not. It therefore allows us to compute an iterate close enough to the exact solution from which we can then successfully start the Newton iteration. (For this second set of iterations, we use the DC form because the updates $\delta\boldsymbol{X}_k$ then have a zero velocity on all boundaries where the velocity is prescribed.)

(iii) We continue with full Newton steps, that is, we attempt to solve the unmodified Newton equations stated in eq. (12) with blocks defined as in eqs (14)–(16). We know that these equations will eventually lead to quadratic convergence, but they may not be symmetric, positive definite, or even solvable. Consequently, the linear solvers we will discuss in the next subsection may fail to converge.

(iv) If the linear solver failed in one of the previous, unmodified Newton steps, we continue with Newton-like steps that modify the matrix blocks as shown in eqs (14)–(16) by the methods of Sections 2.4 and 2.5. By construction, the resulting linear system is then guaranteed to be invertible, and indeed our linear solvers always succeed in our experiments.

These iterations are terminated once the non-linear residual $\|\boldsymbol{r}_k\|$ has been reduced by a user-defined factor compared to the starting non-linear residual at the beginning of each time step. We use a line search (see Kelly 1995) to determine an acceptable step length for all Newton-type steps to further globalize convergence.

In addition to the outline above, we have tried a method suggested to us by Riad Hassani (private communication, 2017) in which the switchover between Picard DC iteration as defined above in (ii) (corresponding to using a Newton matrix in which we have dropped all terms involving derivatives of the coefficients) and Newton iterations (i.e. the same blocks but *including* the derivative terms) is done gradually by scaling the derivatives in overall iteration k by a factor c_k between zero and one. We will in the rest of this paper refer to this as the residual scaling method (RSM). The initial N_{DC} iterations can then be interpreted as using $c_k = 0$, after which we choose

$$c_k = \max\left(0.0,\, 1 - \frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{r}_{N_{\mathrm{DC}}}\|}\right)$$

where $\boldsymbol{r}_k$ is the current non-linear residual and $\boldsymbol{r}_{N_{\mathrm{DC}}}$ the residual in the first iteration after switching to the Newton or Newton-like method. This choice guarantees that $c_k \approx 1$ once Newton's method has reduced the residual significantly, that is, once we are close to the solution.

This variation often allows us to choose N_{DC} smaller, that is, to try a method with a faster convergence rate earlier in the process. On the other hand, it sometimes requires more Newton-type iterations. Using this variation leads to somewhat mixed improvements over the strategy outlined above, as will be shown in our numerical results below.

2.6.2 Linear solvers

Regardless of whether we solve the Picard or any of the Newton-type problems above, we always end up with having to solve a linear system with the same block structure as eq. (13) in each non-linear step. This problem may or may not be symmetric, and the top left block $\boldsymbol{J}^{uu}$ may or may not be positive definite. However, regardless of the these details, we use variations of the solvers discussed in Kronbichler *et al.* (2012) and Heister *et al.* (2017) to solve the linear problem.

More specifically, we use F-GMRES as the outer solver, with the following matrix as a pre-conditioner:

$$\boldsymbol{P}^{-1} = \begin{pmatrix} \widetilde{(\boldsymbol{J}^{uu})^{-1}} & \widetilde{(\boldsymbol{J}^{uu})^{-1}}\,\boldsymbol{J}^{up}\,\widetilde{\boldsymbol{S}^{-1}} \\ 0 & -\widetilde{\boldsymbol{S}^{-1}} \end{pmatrix}, \tag{19}$$

where a tilde indicates an approximation of the matrix under the tilde, and $\boldsymbol{S} = \boldsymbol{J}^{pu}(\boldsymbol{J}^{uu})^{-1}\boldsymbol{J}^{up}$ is the Schur complement of the system. Specifically, motivated by the discussions in Kronbichler *et al.* (2012) and Heister *et al.* (2017), we use the following approximations for each of these blocks:

(i) $\widetilde{(\boldsymbol{J}^{uu})^{-1}}$: we approximate this matrix using either one multigrid cycle or a full solve with an approximation $\widetilde{\boldsymbol{J}^{uu}}$ of $\boldsymbol{J}^{uu}$ that is constructed in a similar way as discussed in Kronbichler *et al.* (2012). In addition, because both multigrid and the Conjugate Gradient method used here require $\widetilde{\boldsymbol{J}^{uu}}$ to be symmetric and positive definite, we always apply the modifications of Sections 2.4 and 2.5, even if they are not applied to $\boldsymbol{J}^{uu}$ itself.

(ii) $\widetilde{\boldsymbol{S}^{-1}}$: this block is an approximation to the inverse of the Schur complement $\boldsymbol{S} = \boldsymbol{J}^{pu}(\boldsymbol{J}^{uu})^{-1}\boldsymbol{J}^{up}$. Like for the original Stokes problem, the appropriate approximation is to use $\widetilde{\boldsymbol{S}^{-1}} = \boldsymbol{M}_p^{-1}$ where $(\boldsymbol{M}_p)_{ij} = \left(\varphi_i^p, \frac{1}{\eta(\varepsilon(\boldsymbol{u}))}\varphi_j^p\right)$ is the mass matrix on the pressure space scaled by the inverse of the viscosity; the inversion of $\boldsymbol{M}_p$ is facilitated by a Conjugate Gradient solve.

The approximation $\widetilde{\boldsymbol{S}^{-1}} = \boldsymbol{M}_p^{-1}$ is known to be good if $\boldsymbol{J}^{pu} = (\boldsymbol{J}^{up})^{\mathsf{T}}$, see Silvester & Wathen (1994). On the other hand, this is not the case if the viscosity depends on the pressure, given the additional term in eq. (15). However, the difference between the two matrices is small if the viscosity does not strongly depend on the pressure. This is, in fact, a commonly made assumption, at least for deep Earth mantle models, though it may not be valid for crustal models that employ pressure-dependent plasticity models.

It is conceivable that one can construct a better approximation for – leading to fewer outer F-GMRES iterations – by also incorporating the viscosity derivative terms somehow, but we did not pursue this direction as it is tangential to the purpose of this paper.

It is, in general, not necessary to solve the linear systems in the first few non-linear iterations with high accuracy. Rather, without significant loss of non-linear solver performance, one can solve with a loose tolerance and terminate F-GMRES substantially earlier. Consequently, we have implemented both choices 1 and 2 of Eisenstat & Walker (1996) for stopping criteria for the linear solver, where for choice 2 we followed Kelly (1995) in using $\gamma = 0.9$ and $\alpha = 2$. For the definition of these symbols see the original paper. We noticed for some of the problems that the difference between these two approaches where significant, where the first choice allowed for a much loser tolerance. Eisenstat & Walker (1996) stated that choice one represents a direct relation between the Newton right-hand side F and its local linear model at the previous non-linear iteration, while choice two is only an approximation of this. Therefore, we have chosen the first of these approaches for this paper.

2.6.3 Computation of derivatives

Implementations of Newton solvers require concrete implementations of the formulae for the derivatives $\frac{\partial\eta(\varepsilon(\mathbf{u}))}{\partial\varepsilon}$ and $\frac{\partial\eta(\varepsilon(\mathbf{u}))}{\partial p}$. These can be computed either using simple finite-differencing approaches or analytically. Fortunately, even for relatively complicated material models, exact formulae for these derivatives can be derived with modest effort. Examples for the material models we consider in our numerical results below are provided in Appendix B.

3 NUMERICAL EXPERIMENTS USING COMMON BENCHMARKS

In this section, let us illustrate the performance of the methods layed out above, using several benchmarks that vary both in which specific elements of the solver they test as well as in the difficulty they present to solvers. In particular, we will assess whether and how fast different variations of our algorithms converge. This includes ensuring that the non-linear residual can be reduced to any small value desired. Furthermore, we will investigate optimal values and relative trade-offs for a variety of parameters that affect the non-linear solver scheme, as discussed in Section 2.6.

The benchmarks we describe here have all been used for similar purposes in the literature. Details of all of our experiments are, sometimes in a simplified form, also part of the ASPECT test suite. All codes necessary to run these experiments are available among the benchmarks included in ASPECT releases starting from version 2.1. The ASPECT repository can be found at https://github.com/geodynamics/aspect.

3.1 Non-linear channel flow

The simplest non-linear Stokes flow one can think of is probably a generalization of incompressible Poiseuille flow to include a strain-rate-dependent viscosity. In it, one forces a fluid through a pipe or channel where the velocity is zero at the pipe sides and in- and outflow velocities are prescribed in such a way that the result is a flow field parallel to the pipe axis and constant in the along-pipe direction. The across-pipe variation of the velocity field can then be computed easily once a rheology law is chosen, leading to an analytically known flow field from which the in- and outflow boundary conditions can also be drawn either via prescribed velocities or prescribed tractions.

A visualization of the solution can be found, for example, in Turcotte & Schubert (2002) and Gerya (2010).

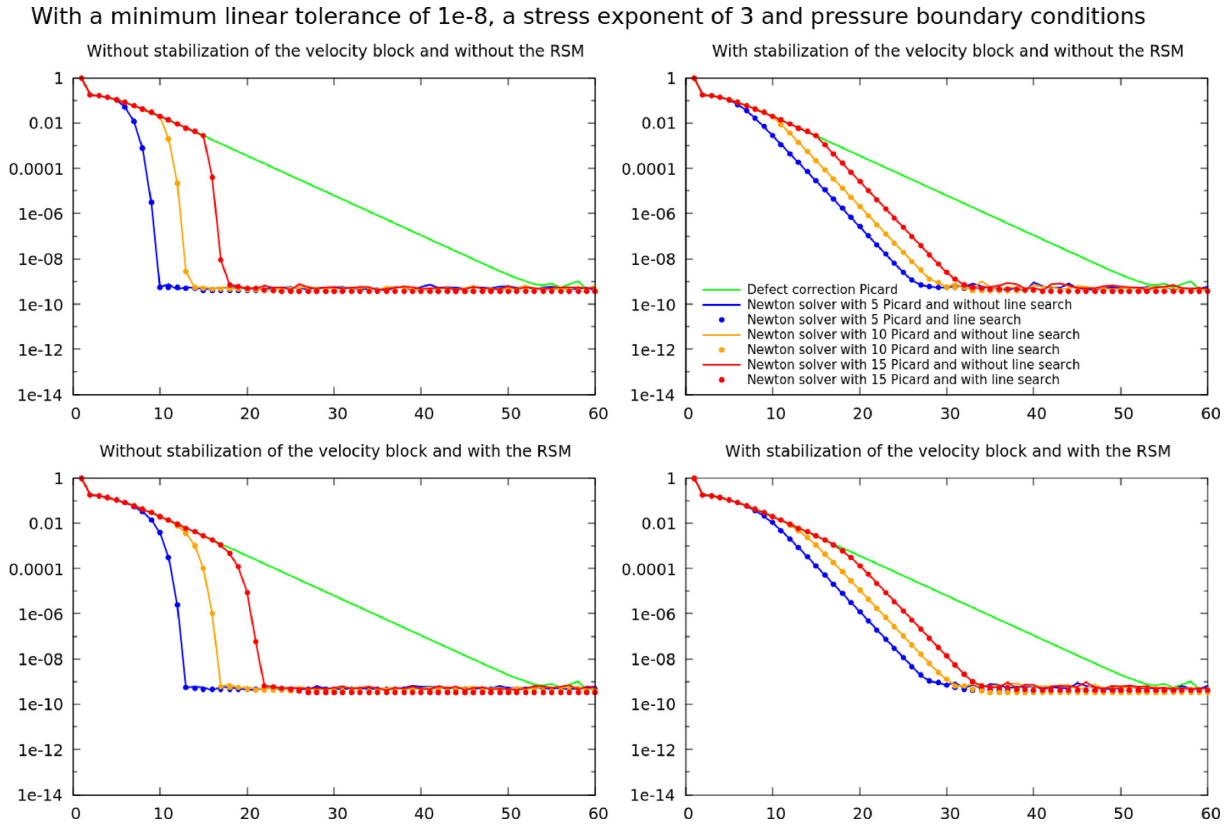

Figure 1. Non-linear channel flow benchmark: convergence history for several methods for a rheology with $n = 3$ where in- and outflow are described by prescribing the traction. Top row: computations in which we switch abruptly from Picard iterations to Newton iterations. Bottom row: with the RSM in which we switch continuously between the Picard iteration and the Newton method. Left-hand column: unmodified Newton iterations. Right-hand column: results where we applied the modifications of Sections 2.4 and 2.5 to the Newton matrix. Horizontal axes: number of the non-linear (outer) iteration. Vertical axes: non-linear residual.

3.1.1 Setup

We use the 2-D benchmark setup of Gerya (2010, section 16.4). In it, the viscosity is chosen in accordance with a power-law approach as

$$\begin{aligned}\eta(\varepsilon(\mathbf{u})) &= C^{-\frac{1}{n}}\,[2\,\mathbb{I}_2(\varepsilon(\mathbf{u}))]^{\frac{1}{n}-1}\\ &= C^{-\frac{1}{n}}\left[2\sqrt{\frac{1}{2}\varepsilon(\mathbf{u}):\varepsilon(\mathbf{u})}\right]^{\frac{1}{n}-1}\\ &= \left[\sqrt{2}\right]^{\frac{1}{n}-1} C^{-\frac{1}{n}}\,\|\varepsilon(\mathbf{u})\|^{\frac{1}{n}-1},\end{aligned} \tag{20}$$

using the definition of the second invariant found in Gerya (2010, p. 59, equation 4.14). Here, C is a prefactor, and n is a stress exponent that allows for easy tuning of the non-linearity of the problem. The model geometry we use here is a box of 10 000 m $\times$ 8000 m, subdivided into 16 $\times$ 16 cells; we use quadratic finite elements for the velocity.

3.1.2 Results

Figs 1 and 2 show results for a number of methods and settings when the in- and outflow boundary conditions are either prescribed through tractions or velocity values. The latter turns out to generally be a more difficult problem to solve, but all methods eventually converge to a residual whose size is related to the tolerance with which we solve the linear systems.

Fig. 1 shows that for this problem, when boundary values are given as tractions, line search is neither necessary nor useful, and similarly it is not necessary to run many initial Picard iterations to get close enough to the solution for the Newton method to start working. In addition, the Newton matrix modifications of Section 2.5 (right two panels of Fig. 1) actually destroy the quadratic convergence rate of Newton's method and result in only linear convergence as speculated at the beginning of Section 2.6 – though with a substantially better linear rate than Picard iterations.

On the other hand, Fig. 2 shows that for the more complicated problem when the flow is driven by prescribed velocity boundary conditions, either a line search method or sufficiently many initial Picard iterations are necessary to achieve convergence. Alternatively, the

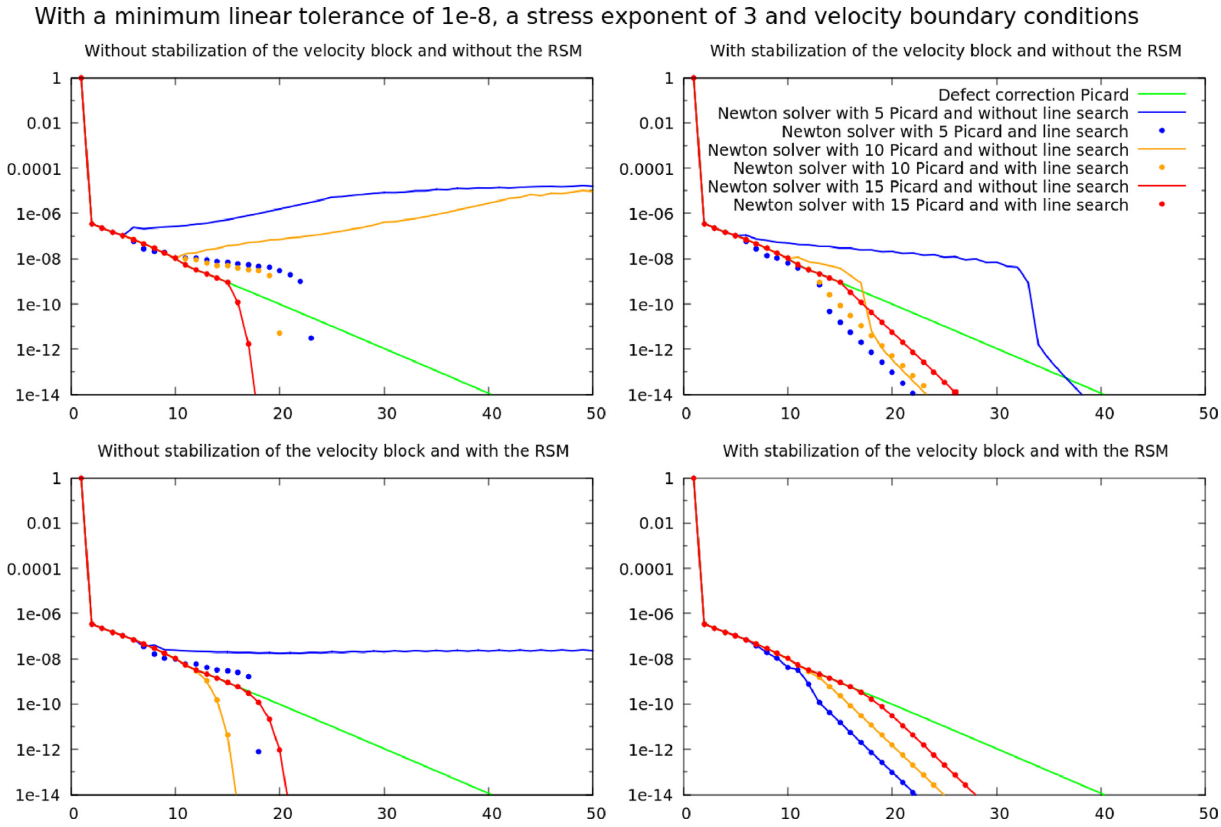

Figure 2. Non-linear channel flow benchmark: convergence history for several methods for a rheology with $n = 3$ where in- and outflow are described by prescribing the velocity. Panels as in Fig. 1.

matrix modifications also yield a convergent scheme. The RSM in conjunction with the matrix modifications appears the most robust method, though not always the fastest. Indeed, as explained in Appendix B1, a stress exponent of $n = 3$ causes the matrix modifications to *always* scale down the derivative terms in the Newton matrix, resulting in a similar effect as the RSM.

In all cases, a pure Picard iteration always converges linearly, though at a rate that is not competitive with well-designed Newton iterations.

3.2 Spiegelman et al. benchmark

The Spiegelman et al. benchmark (see Spiegelman *et al.* 2016) is an extended form of the brick benchmark of Lemiale *et al.* (2008) and focuses on solving for the behaviour of a material with plastic rheology under compression.

3.2.1 Setup

The benchmark specifies two layers, see Fig. 3. The lower layer, which includes a regularized weak seed, has a constant viscosity. The upper layer has a viscosity given by the harmonic mean $\eta_{\text{eff}} = \frac{\eta_1 \eta_p}{\eta_1 + \eta_p}$. Here, η_1 is the background viscosity of the upper layer, and

$$\eta_p = \frac{A + B(p_{\text{lith}} + \alpha p')}{2\,\mathbb{I}_2(\varepsilon(\mathbf{u}))},$$

where p_{lith} is the depth dependent lithostatic pressure and $p' = p - p_{\text{lith}}$ is the dynamic component of the total pressure. η_1 can have three different values: $1e23$, $1e24$ and $5e24$ Pa s. For von Mises plasticity, one would choose $A = C$, $B = 0$ where C is the cohesion of the material; in this case, the viscosity is strain rate but not pressure dependent. For a depth-dependent von Mises-type model, one would choose $A = C\cos(\phi)$, $B = \sin(\phi)$ and $\alpha = 0$ where ϕ is the friction angle; in this case, the viscosity depends on the static, lithospheric pressure but not the dynamic pressure component. Finally, Drucker–Prager plasticity fits this formula with $A = C\cos(\phi)$, $B = \sin(\phi)$ and $\alpha = 1$ where the viscosity now depends on both the strain rate and the (total) pressure $p = p_{\text{lith}} + p'$. We will only consider the von Mises and Drucker–Prager cases of the Spiegelman benchmark because these are the most interesting ones.

The benchmark is completed by prescribing an inbound velocity (2.5, 5 and 12.5 mm yr^{-1}) on the two sides of the geometry, requiring tangential flow at the bottom, and no stress at the top, allowing material to leave the domain. The three options for the inbound velocity and the three options for the reference background viscosity together form a set of nine test cases whose difficulty increases with velocity and

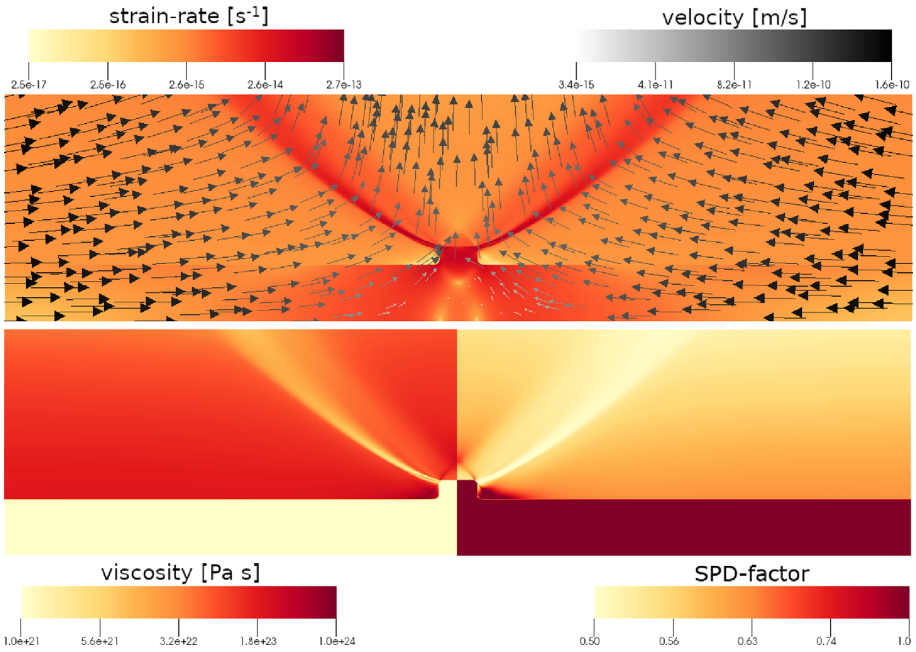

Figure 3. Spiegelman et al. benchmark: depiction of the strain rate and velocity field (top), viscosity (bottom left) and the SPD factor α (bottom right) as a result of the deformation induced by the prescribed horizontal velocity on the sides of the domain. The results shown here are for the case where the velocities are 5 mmyr^{-1} (the colour bar shows it in m s^{-1}, $\eta_{\text{ref}} = 10^{24}$ Pa s, the angle of internal friction is 30° and the mesh consists of 1024 × 256 cells. This is one of the more difficult cases (Drucker–Prager) of the benchmark, and the data shown here is the result of simulations that are only converged to a relative non-linear residual of about 10^{-6}.

reference background viscosity. In the original paper, an unstructured grid of stable Taylor–Hood elements was used. Based on their fig. 1, the results shown there should, based on the length of the edges, correspond to a uniform ASPECT mesh that has been refined globally approximately 7 or 8 times (i.e. 512 × 128 or 1024 × 256 cells). This does not, however, account for details of the unstructured mesh used in Spiegelman *et al.* (2016).

3.2.2 Results

We compare the results of our Newton implementation to the results of Spiegelman *et al.* (2016) and the pre-existing Picard solver in ASPECT. The von Mises case results are quite similar to the results from Spiegelman *et al.* (2016); consequently, we will focus on the more difficult Drucker–Prager case. We have exhaustively explored the space of parameters affecting the non-linear solver (see Section 2.6) at different mesh resolutions. In general, we found that higher mesh resolution (more refinement steps) made the problem more difficult to solve. As expected, the benchmarks also become more difficult as the prescribed velocity u_0 at the boundary is increased, leading to a larger strain rate and more pronounced non-linearity. This is visible in Fig. 3 where we also show the viscosity and the α factor necessary to keep the matrix positive definite. This factor drops to approximately one-half in the vicinity of the shear band – a result consistent with the theoretical considerations discussed in Appendix B3.

Figs 4 and 5 show results obtained for four and eight global mesh refinement steps, corresponding to meshes with 64 × 16 and 1024 × 256 cells, respectively. A comparison shows that the problem is indeed more difficult to solve on the finer meshes. Without enforcing the symmetry and positive definiteness of the top left block of the Jacobian, the linear solver converges quickly, but also crashes easily in a number of configurations because the matrix lacks the necessary structural properties; this is particularly the case for $u_0 = 12.5$ mm yr^{-1}. On the other hand, enforcing these properties on the matrix leads to some loss in speed of convergence of the non-linear iterations (because the computed search direction is no longer second order accurate), though we then also no longer encountered any linear solver failures.

We noticed that the runs without the matrix modifications are very sensitive to changes in many of the solver parameters. In the following, let us discuss a number of settings that we have found useful when working with the unmodified matrix, though for many cases the linear solver still fails with these settings. In particular, using a few line search iterations may help reduce the amount of iterations; however, allowing the step length parameter to decrease too much generally leads to too small steps and slow overall convergence. A good default value for the maximum number of line search step length reductions appears to be 5 or 10. We also found that performing 5–10 Picard iterations before switching to Newton iterations is a good number. Setting the relative linear tolerance (LT) of the GMRES solver rather strict typically results in fewer outer iterations, but at the price of more inner iterations that then increase the run time per outer iteration. A good compromise for this parameter is to require that the linear residual is reduced to a factor of 0.1 or 0.01 of its initial value. Using the RSM, that is, determining the linear solver tolerance automatically, usually requires one or two more outer iterations, but greatly decreases the chance that the linear solver will fail.

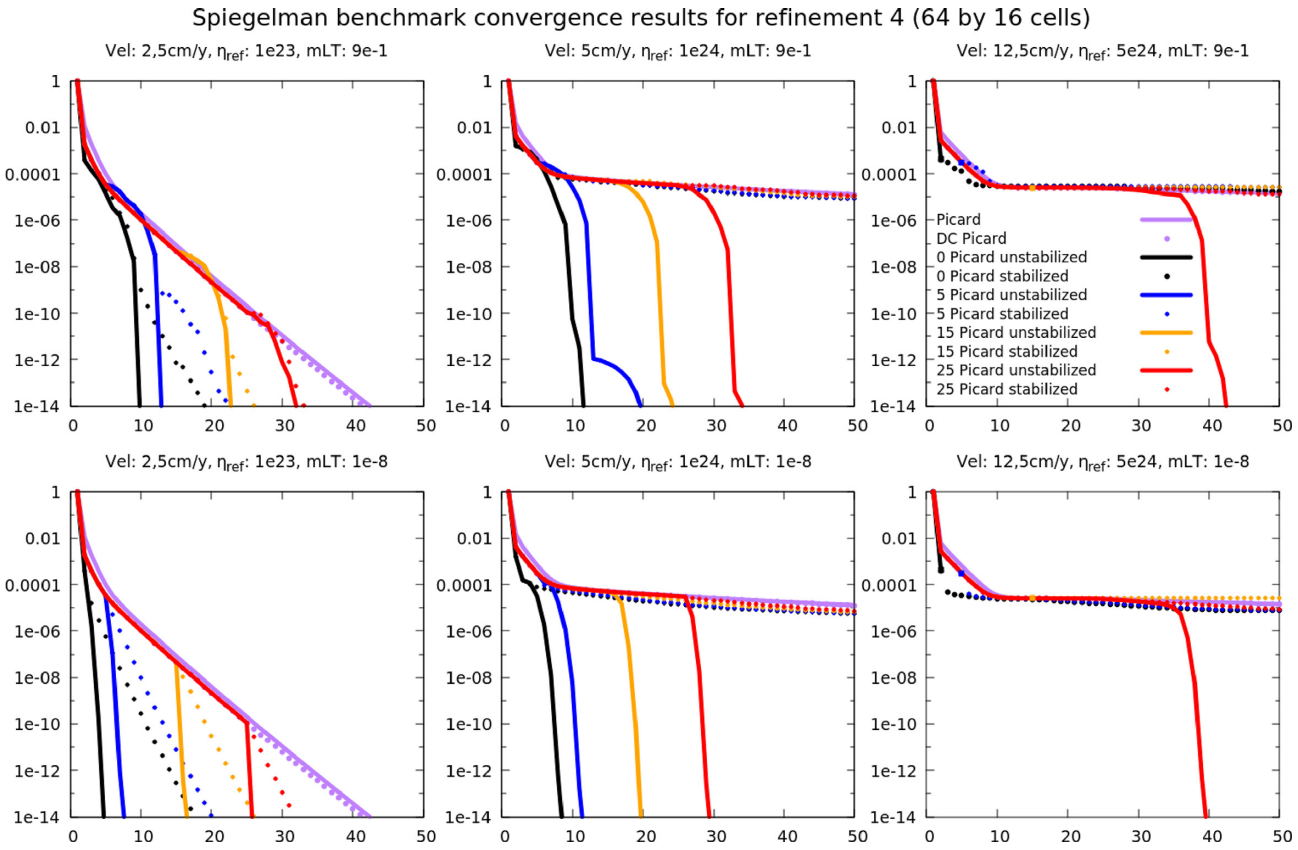

Figure 4. Spiegelman et al. benchmark: a reproduction of three of the nine pressure dependent Drucker–Prager cases with a resolution of 64 × 16 cells (substantially coarser than the resolution in the Spiegelman *et al.* 2016). Top: results for computations where linear systems are solved with a relative tolerance of 0.9. Bottom: with a tolerance of 10^{-8}. The initial Picard iteration is always solved to a linear tolerance of 10^{-16}. Left to right: different prescribed velocities of $u_0 = 2.5$, 5 and 12.5 mm yr^{-1} and different reference viscsocities of respectively $\eta_{\text{ref}} = 10^{23}$, 10^{24} and 5×10^{24} Pa s . Horizontal axis: number of the non-linear (outer) iteration; and vertical axis: non-linear residual. DC Picard refers to a Defect Correction Picard iteration, see Section 2.6.1.

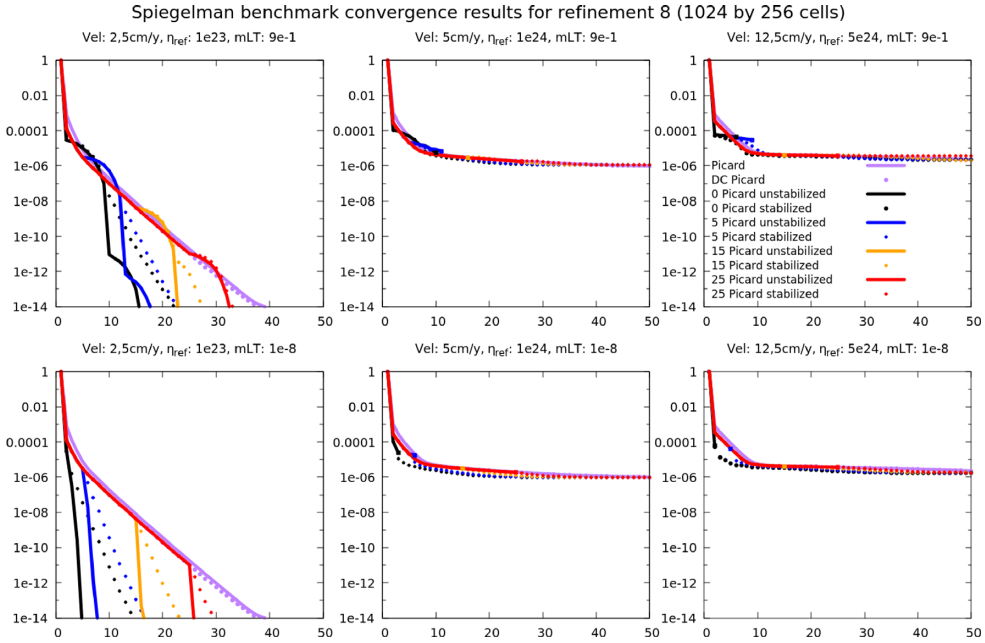

Figure 5. Spiegelman et al. benchmark: a reproduction of three of the nine pressure dependent Drucker–Prager cases with a resolution of 1024 × 256 cells (similar to the resolution in the Spiegelman *et al.* 2016). Panels as in Fig. 4.

On the contrary, enforcing the $\boldsymbol{J}^{uu}$ to be symmetric and positive definite will yield a very different behaviour. As stated above, the linear solver then *always* converges. Furthermore, the sensitivity of the convergence history to all the parameters above is greatly reduced. This has the advantage that a very loose LT can be used without a significant penalty in the number of iterations; that said, and as is apparent from the figures, modifying the matrix may require tens of non-linear iterations more to converge.

We have obtained the best performance by combining the two methods: we do not enforce the symmetry and positive definiteness of the matrix in the first non-linear iterations until (if) the linear solver fails; after this, we continue with the matrix modifications. This strategy combines the fast convergence of the unmodified method with the stability and robustness of the modified one.

3.3 Tosi et al. benchmark

The Tosi benchmark of Tosi *et al.* (2015) is designed as a community benchmark for mantle flow based on non-linear rheologies featuring a temperature, pressure and strain rate-dependent viscosity. Here, we specifically consider case 4 from Tosi *et al.* (2015), which seeks the steady state (at a large, unspecified end time) of a time-dependent problem. Unlike the two previous benchmarks, this benchmark has a temperature field that is coupled to the viscosity and therefore evolves over time.

3.3.1 Setup

The benchmark is posed in a 2-D square unit box with all free slip boundaries and an initial temperature given by $T(x, z) = (1 - z) + A\cos(\pi x)\sin(\pi z)$. The viscosity is chosen as the harmonic mean

$$\eta(T, z, \varepsilon(\mathbf{u})) = 2\left(\frac{1}{\eta_{\text{lin}}(T, z)} + \frac{1}{\eta_{\text{plast}}(\varepsilon(\mathbf{u}))}\right)^{-1}, \tag{21}$$

where the two components of the viscosity are defined as a linear but depth- and temperature-dependent viscosity as well as a plastic yield criterion respectively:

$$\eta_{\text{lin}}(T, z) = e^{-\gamma_T T + \gamma_z z}, \qquad \eta_{\text{plast}}(\varepsilon(\mathbf{u})) = \eta^* + \frac{\sigma_Y}{\|\varepsilon(\mathbf{u})\|}.$$

Here, η^* is the constant effective viscosity at high strain rate, and σ_Y the yield stress. Numeric values for all of these constants can be found in the original paper.

3.3.2 Results

The original paper does not contain convergence plots. Consequently, we can only compare between the methods available in our reference implementation. Specifically, these are: (i) a method whereby we solve the advection equation, then the Stokes equation with frozen coefficients, and then iterate these two steps out until we have reached convergence for the current time step; we will refer to this scheme as 'Picard' even though this stretches the term (as, strictly speaking, a 'Picard' iteration for the coupled system would solve both the Stokes and advection problem linearized around the previous solution; in our implementation, the Stokes system is linearized around the already computed advection solution). In ASPECT, this scheme is called 'iterated advection and Stokes'. (ii) A method where we first solve the advection equation and then do one Newton step on the non-linear Stokes system; again, these two parts are iterated out in each time step. We will refer to this method as 'Newton'; in ASPECT, it is called 'iterated advection and Newton Stokes'.

Fig. 6 shows results for this benchmark, where the horizontal axis indicates the number of the non-linear iteration performed. Each spike corresponds to a time step starting at a large residual that is gradually decreased. A method that converges quickly shows a steeper decrease, requires fewer non-linear iterations, and can consequently fit more time steps (spikes) into the same number of non-linear iterations. Because the computational effort is largely confined to building and solving the linear systems, the horizontal axis also corresponds closely to the elapsed wall time.

The different panels of the figure can be summarized as follows: (i) the Newton method converges much faster than Picard iterations. For example, after the initial few time steps, the Newton method (with and without matrix stabilization) only requires two non-linear iterations per time step, whereas the Picard iteration requires six. This also translates to a speed up in wall time of around the same factor. (ii) Matrix stabilization is not necessary for this benchmark and in fact leads to a slight but not substantial degradation of performance. (iii) Convergence behaviour can differ substantially between timesteps (both for Picard and Newton). (iv) For this experiment, the DC form of the Picard iteration is slightly, but not substantially faster than the original Picard iteration.

3.4 A 3-D subduction test case

In order to verify that our implementation is not only applicable to academic benchmarks, but also to settings that occur in geodynamic modelling, let us consider a 3-D simulation of oceanic plate subduction. The model is inspired by the geodynamic setting of the Caribbean region, and simulates a slab that subducts while the subducting plate has a motion oblique to the trench which causes the slab to be dragged laterally through the mantle, a motion called slab dragging (Spakman *et al.* 2018).

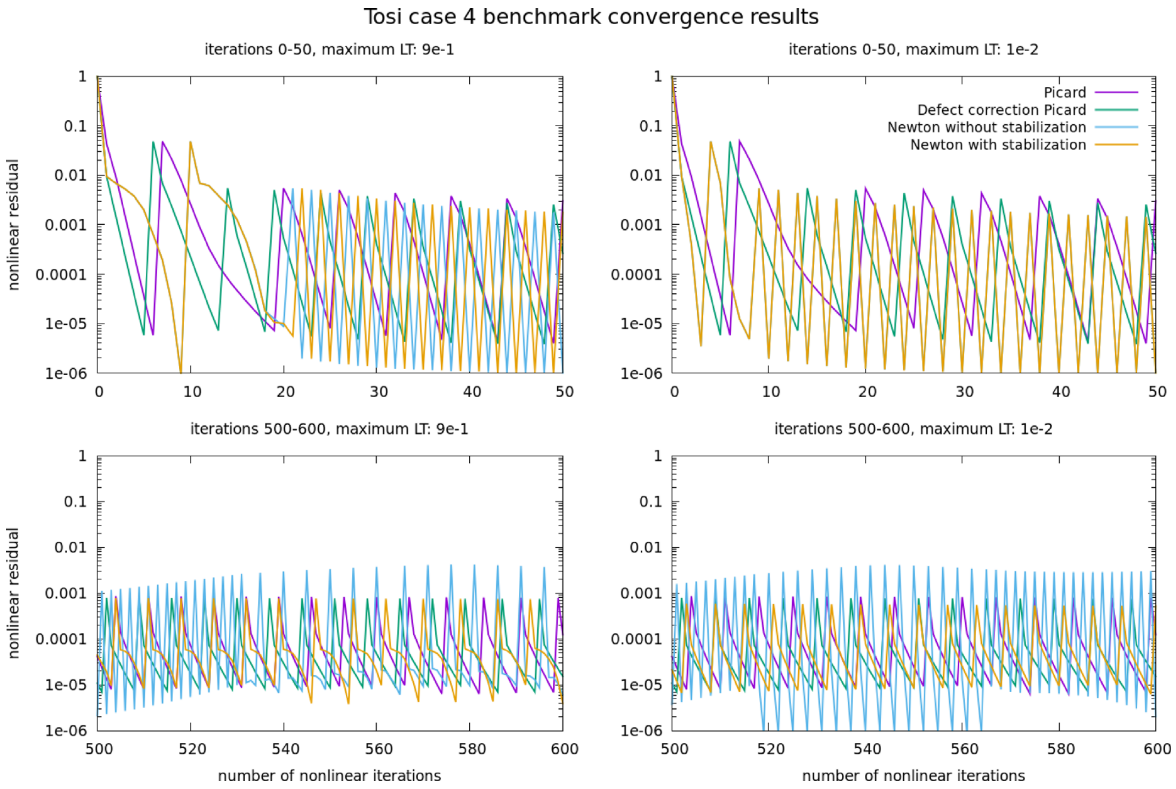

Figure 6. Tosi et al. benchmark: non-linear Stokes residual (vertical axis) as a function of the number of the non-linear solves. Each spike corresponds to one time step during which iterations start with a large residual that is gradually decreased until it reaches the desired non-linear tolerance of 10^{-5}. No line search is used here. Top: the first 50 non-linear iterations. Bottom: non-linear iteration 500–600. Left: linear systems are solved to a relative linear tolerance (LT) of 0.9. Right: with a linear tolerance of 0.01. In the top right panel, the results for the two Newton variations coincide.

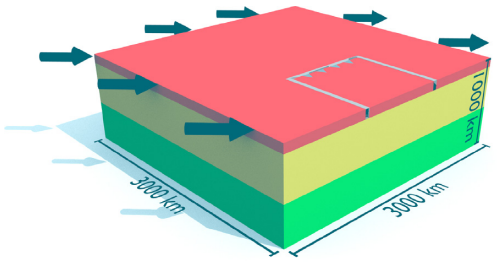

Figure 7. A conceptual visualization of the 3-D subduction test case setup. TThe red layer represents the lithosphere, the yellow layer the upper mantle and the green layer the lower mantle. The arrows indicate the lithospheric boundary velocity direction.

3.4.1 Setup

We situate our test case in a Cartesian box with dimensions of 3000 km in width and length and 1000 km in depth (see Fig. 7). The viscoplastic rheology includes dislocation creep, diffusion creep and plasticity. The viscosity is then given by

$$\eta_{\text{eff}} = \max\left(\min\left\{\left(\frac{1}{\eta_{\text{diff}}} + \frac{1}{\eta_{\text{disl}}}\right)^{-1},\ \eta_{\text{plastic}},\ \eta_{\text{max}}\right\},\ \eta_{\text{min}}\right) \tag{22}$$

where

$$\eta_{\text{x}} = \frac{1}{2}\nu_{\text{x}} A_{\text{x}}^{-\frac{1}{n_{\text{x}}}} \left[\frac{1}{\sqrt{2}}\|\varepsilon(\mathbf{u})\|\right]^{\frac{1}{n_{\text{x}}}-1} \exp\left(\frac{E_{\text{x}} + PV_{\text{x}}}{n_{\text{x}} RT}\right),$$

$$\eta_{\text{plastic}} = \frac{6\,(C\cos(\phi) + P\sin(\phi))}{\sqrt{3}\,(3-\sin(\phi))} \frac{1}{\sqrt{2}\,\|\varepsilon(\mathbf{u})\|}.$$

Here, a symbol of the form $\square_{\text{x}}$ stands for the corresponding property of either diffusion (if x = diff) or dislocation creep (if x = disl), ν is a constant factor which can be used to scale the rheology, R is the gas constant, T is temperature and P is pressure. A_{x} are pre-factors, E_{x} are the activation energies and V_{x} are the activation volumes. The values of all of these parameters used for the simulations are listed in Appendix C.

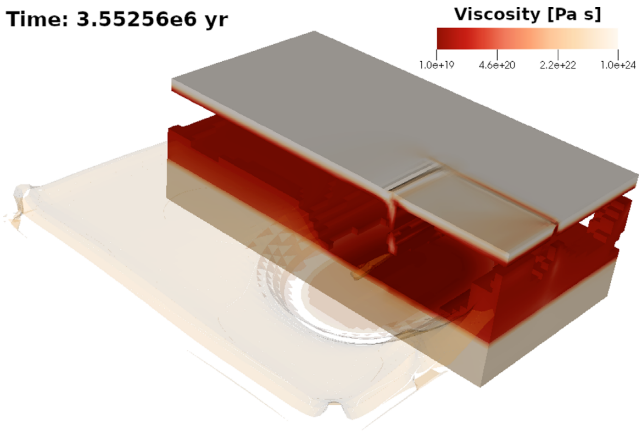

Figure 8. Right: the viscosity field of the 3-D subduction test case after 3.55×10^6 yr, just as the tip of the subducting slab starts to detach from the rest of the slab. In the left half, two isosurfaces show where in the bottom half of the model the viscosity equals 3.25×10^{19} and 5.5×10^{19} Pa s. In the right half, colours indicate the viscosity when restricted to areas where the viscosity is greater than 5×10^{19} Pa s, i.e., to cold areas of the upper mantle, as well as the lower mantle.

The model consists of two layers, the upper and lower mantle (above and below 660 km, see Fig. 7), that differ only in strength through a 100-fold increase in η_{diff} and η_{disl} by choosing ν_{diff} and ν_{disl} larger by a factor of 100 in the lower mantle. This means that the lithosphere is fully defined by temperature. This thermal lithosphere is divided into two regions: a U-shaped region representing an oceanic plate that surrounds a region representing a Large Igneous Province (LIP), both modelled by a plate (Fowler 2005) of thickness 95 km and the ridge far outside the domain. The slab is also 95 km thick and is divided into three segments in which an analytic temperature field is prescribed following McKenzie (1970). The first segment is 200 km long with a dip angle relative to the surface starting with 20° that smoothly steepens to a dip angle of 30°. The second segment is 150 km long and the dip angle has at the end of the segment smoothly increased to 70°. The third, straight segment is 50 km long and has a constant dip angle of 70°. We describe the fault zones between the oceanic plate and the LIP as thin, vertical regions with an elevated initial temperature. The U-shaped lithosphere has a prescribed boundary velocity of 1 cm yr^{-1} in each component of the horizontal directions (for direction, see the arrows on Fig. 7), and zero velocity in the vertical direction. The LIP has a prescribed boundary velocity of zero in all directions. Below the lithosphere we use open boundary conditions. The top is a free surface (Rose *et al.* 2017) and the bottom has a zero velocity boundary condition.

This model is discretized on a mesh that has a total of 153 046 cells, resulting in 39 38 115 velocity and 172 097 pressure unknowns (in addition to another 13 12 705 unknowns each for the temperature and a compositional field). All results shown below were obtained on the Dutch national cluster Cartesius. Each node is equipped with Intel Xeon E5-2690 v3 (‘Haswell’) processors and has 24 cores; we used 10 nodes and 20 MPI processes per node. The model is run with a Courant–Friedrichs–Lewy number of 0.1 and the time step size is limited to grow by a maximum of 25 per cent from one time step to the next. The time step sizes computed by all non-linear solver methods used below are essentially identical.

The experiments in previous sections show that it is in general not necessary to solve the linear systems in DC schemes (i.e. the DC version of Picard iterations, as well as Newton iterations) particularly accurately. As a consequence, we only use a linear solver tolerance for these methods that requires a reduction of the linear residual in the F-GMRES solve by a factor of 0.1. On the other hand, the initial Picard iteration solves for the solution, not an update, and consequently requires a substantially larger reduction of the linear residual; we use a factor of 10^{-6} (the default value of ASPECT).

3.4.2 Results

We have performed this experiment using our implementation of the Newton method, as well as the Picard iteration and its DC variation. In the following, let us provide two perspectives on this comparison.

First, Fig. 9 shows how all three methods reduce the non-linear residual in each time step, for two selected periods of the simulation (from time steps 75 to 100, and 275 to 300). In the early phases of the simulation, all methods quickly converge the non-linear residual to the desired tolerance of 10^{-6}, though even here, the Newton method requires fewer iterations. Interestingly, starting around time step 275 – corresponding to about 3.92 Myr of model time – the problem appears to become substantially more difficult to solve. This time corresponds to the break-off of the deeper part of the slab (the necking in Fig. 8), which then rapidly sinks, and the increased velocity implies a larger strain rate and consequently stronger non-linearity. Indeed, as Fig. 9 shows, both variations of the Picard iteration are then no longer able to converge the residual to the desired tolerance, even though we allow up to 250 iterations per time step. Any results of these simulations must then necessarily be suspicious. On the other hand, the Newton iteration continues to rapidly converge. These data therefore underline the stability and robustness of the Newton method, and illustrate that it can solve problems that are otherwise not solvable with reasonable effort.

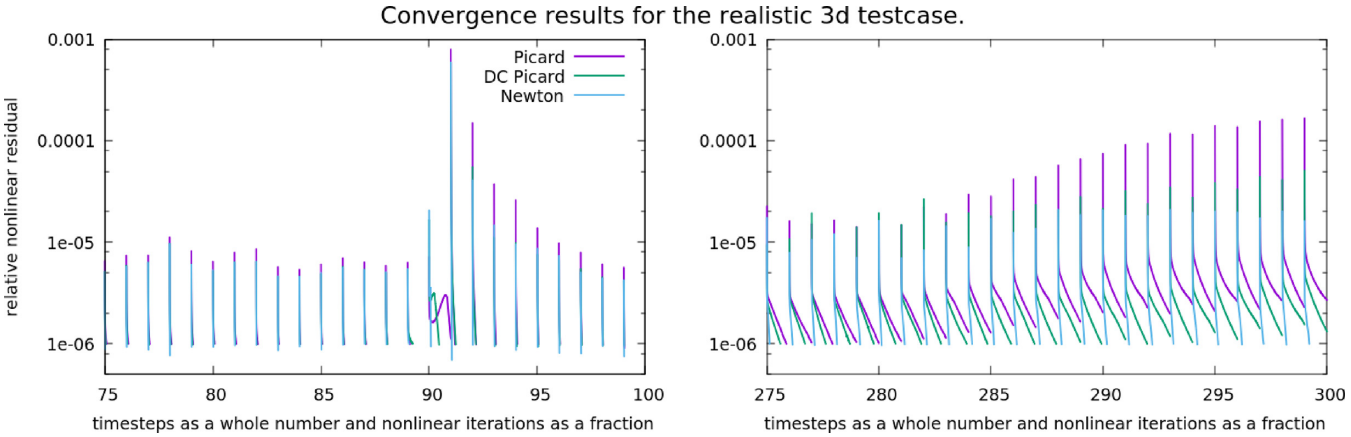

Figure 9. The 3-D subduction test case: non-linear convergence results for the Picard, defect correction (DC) Picard, and Newton iterations. The horizontal axis shows time steps, with non-linear iterations depicted at $\frac{1}{250} = 0.004$ increments given that we allow at most 250 non-linear iterations per time step. The vertical axis represents the non-linear relative residual.

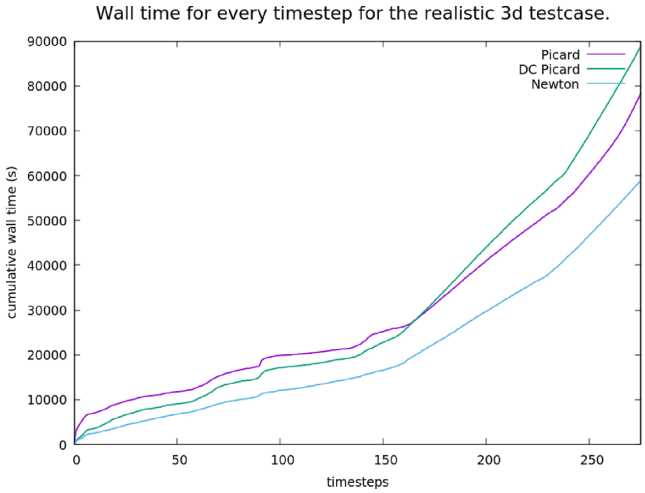

Figure 10. The 3-D subduction test case: wall clock time necessary to compute up to a certain time step for the Picard, defect correction (DC) Picard and Newton iterations.

A second perspective is shown in Fig. 10, where we plot the wall time necessary to solve the problem up to a given time step. The figure demonstrates that our implementation of Newton's method is approximately one-third faster than the other two methods for the time steps shown. The difference becomes particular notable after time step 170, where the subducting slab starts to thin under its own gravity. We did not include data beyond time step 275 because, as discussed above, the Picard variants do not converge any more after this point. Consequently, even though the curves would substantially diverge after this time step due to the far larger number of non-linear iterations taken by the Picard variants, the solutions would no longer be comparable.

These models mostly use ASPECT's default parameter values. It is possible that we could further optimize any of the methods shown by changing these values. Furthermore, we have also so far not put much effort into optimizing the efficiency of the Newton solver code itself. That said, both of these issues are outside of the scope of the paper.

4 CONCLUSIONS

Newton's method is generally considered the best method to solve non-linear systems, but it is also known to be difficult to make work in practice. In this contribution, we have demonstrated that a naive application of Newton linearization may lead to a system for the updates $\delta \boldsymbol{X}_k$ that may be ill posed even if the original non-linear problem is well posed. We have shown how one can modify the Newton system to guarantee a stable solution of the update equations. Specifically, we have modified the partial differential operators that give rise to the Newton matrix so that the matrix is symmetric and positive definite. Importantly, however, we do not modify the right-hand side of the Newton update equation, and consequently the iterates of the modified Newton method still converge to the solution of the original non-linear problem. We have also described the globalization strategies necessary to guarantee that the Newton method actually converges in nearly all cases, and demonstrated our methods using standard geodynamic benchmarks and a real application.

The results we have shown demonstrate that with this combination of methods, Newton's method (with globalization approaches and stabilization of the matrix) really is the better choice: it converges more rapidly, in fewer iterations, is robust, and takes less computational time. Furthermore, it can be applied to problems that are known to be very non-linear and difficult to solve, such as the Spiegelman *et al.* (2016) benchmark. Finally, we have applied this method to a rheologically and geodynamically complex, 3-D case of oceanic subduction that is sufficiently non-linear that the typical Picard iteration no longer converges in a reasonable number of steps. The Newton method we have discussed here not only converges, but does so in relatively few steps and with substantial savings in wall clock time.

There are, of course, cases where a simple Picard solver would have been sufficient. In those cases, however, it is worth pointing out that using a Newton method is not substantially more expensive than a Picard method: the assembly of the Newton matrix is marginally more complicated because of the terms involving the derivatives of the viscosity, but other than that the solution procedure is the same between the two methods because of the structurally very similar matrices involved. Consequently, it is reasonable to advocate for *always* using a Newton method instead of Picard iterations.

ACKNOWLEDGEMENTS

The authors greatly appreciate the support by all developers of the ASPECT code, and in particular by Juliane Dannberg, Rene Gassmöller, and Timo Heister. WB would like to thank Denis Davydov, Andrew McBride and Jean-Paul Pelteret for helpful discussions about issues such as those discussed in Sections 2.4 and 2.5 in the context of strain-weakening solid mechanics.

This work is funded by the Netherlands Organization for Scientific Research (NWO), as part of the Caribbean Research program, grant number 858.14.070, as well as by the NWO project 'Large scale finite element models of the Caribbean region: Newton versus Picard non-linear iterations' with project number 15820.

We acknowledge computational support by theNetherlands Research Centre for Integrated Solid Earth Science (ISES).

WB's work was partially supported by the Computational Infrastructure in Geodynamics initiative (CIG), through the National Science Foundation under award number EAR-0949446 and The University of California – Davis; and by the National Science Foundation under awards OCI-1148116 and OAC-1835673 as part of the Software Infrastructure for Sustained Innovation (SI2) program (now the Cyberinfrastructure for Sustained Scientific Innovation, CSSI).

WS acknowledges support from the Research Council of Norway through its Centres of Excellence funding scheme, project number 223272.

REFERENCES

Baumann, T.S., Kaus, B.J. & Popov, A.A., 2014. Constraining effective rheology through parallel joint geodynamic inversion, *Tectonophysics,* **631,** 197–211.

Brenner, S.C. & Scott, R.L., 2002. *The Mathematical Theory of Finite Elements,* Springer, 2nd edn.

Buiter, S. *et al.*, 2016. Benchmarking numerical models of brittle thrust wedges, *J. Struct. Geol.,* **92,** 140–177.

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

Fowler, C., 2005. *The Solid Earth: An Introduction to Global Geophysics,* Cambridge University Press.

Fritzell, E., Bull, A. & Shephard, G., 2016. Closure of the Mongol-Okhotsk Ocean: insights from seismic tomography and numerical modelling, *Earth planet. Sci. Lett*., **445,** 1–12.

Gerya, T., 2010. *Introduction to Numerical Geodynamic Modelling,* Cambridge University Press.

Glerum, A., Thieulot, C., Fraters, M., Blom, C. & Spakman, W., 2018. Nonlinear viscoplasticity in ASPECT: benchmarking and applications to subduction, *Solid Earth,* **9**(2), 267–294.

Heister, T., Dannberg, J., Gassmöller, R. & Bangerth, W., 2017. High accuracy mantle convection simulation through modern numerical methods. II: realistic models and problems, *Geophys. J. Int*., **210**(2), 833–851.

Karato, S., 2012. *Deformation of Earth Materials: An Introduction to the Rheology of Solid Earth,* Cambridge University Press.

Karato, S. & Wu, P., 1993. Rheology of the upper mantle: a synthesis, *Science,* **260,** 771–778.

Kaus, B.J.P, Popov, A.A., Baumann, T.S., Püsök, A.E., Bauville, A., Fernandez, N. & Collignon, M., 2015. Forward and inverse modelling of lithospheric deformation on geological timescales, in *NIC Symposium 2016, SC '15,* pp. 5:1–5:12, ACM.

Kelly, C.T., 1995. *Iterative Methods for Linear and Nonlinear Equations,* SIAM.

Knoll, D.A. & Keyes, D.E., 2004. Jacobian-free Newton-Krylov methods: a survey of approaches and applications, *J. Comput. Phys.,* **193,** 357–397.

Kronbichler, M., Heister, T. & Bangerth, W., 2012. High accuracy mantle convection simulation through modern numerical methods, *Geophys. J. Int*., **191,** 12–29.

Lemiale, V., Mhlhaus, H.-B., Moresi, L. & Stafford, J., 2008. Shear banding analysis of plastic models formulated for incompressible viscous flows, *Phys. Earth planet. Inter*., **171**(1), 177–186.

May, D.A., Brown, J. & Le Pourhiet, L., 2015. A scalable, matrix-free multigrid preconditioner for finite element discretizations of heterogeneous Stokes flow, *Comput. Methods Appl. Mech. Eng*., **290,** 496–523.

McKenzie, D., 1970. Temperature and potential temperature beneath island arcs, *Tectonophysics,* **10**(1), 357–366.

Nocedal, J. & Wright, S.J., 1999. *Numerical Optimization*, *Springer Series in Operations Research,* Springer.

Pusok, A.E. & Kaus, B.J.P., 2015. Development of topography in 3-d continental-collision models, *Geochem. Geophys. Geosyst*., **16**(5), 1378–1400.

Rose, I., Buffett, B. & Heister, T., 2017. Stability and accuracy of free surface time integration in viscous flows, *Phys. Earth planet. Inter*., **262,** 90–100.

Rudi, J. *et al.*, 2015. An extreme-scale implicit solver for complex PDEs: highly hetrogeneous flow in Earth's mantle, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '15,* pp. 5:1–5:12, ACM.

Ruh, J.B., Gerya, T. & Burg, J.-P., 2013. High-resolution 3D numerical modeling of thrust wedges: influence of dcollement strength on transfer zones, *Geochem. Geophys. Geosyst*., **14**(4), 1131–1155.

Schellart, W. & Moresi, L., 2013. A new driving mechanism for backarc extension and backarc shortening through slab sinking induced toroidal and poloidal mantle flow: results from dynamic subduction models with an overriding plate, *J. geophys. Res.,* **118**(6), 3221–3248.

Schubert, G., Turcotte, D.L. & Olson, P., 2001. *Mantle Convection in the Earth and Planets,* Cambridge University Press.

Silvester, D. & Wathen, A., 1994. Fast iterative solution of stabilised Stokes systems. Part II: using general block preconditioners, *SIAM J. Numer. Anal.*, **31,** 1352–1367.

Spakman, W., Chertova, M.V., van den Berg, A. & van Hinsbergen, D.J.J., 2018. Puzzling features of western Mediterranean tectonics explained by slab dragging, *Nat. Geosci.,* **11**(3), 211–216.

Spiegelman, M., May, D.A. & Wilson, C.R., 2016. On the solvability of incompressible Stokes with viscoplastic rheologies in geodynamics, *Geochem. Geophys. Geosyst.*, **17**(6), 2213–2238.

Tosi, N. *et al.*, 2015. A community benchmark for viscoplastic thermal convection in a 2-d square box, *Geochem. Geophys. Geosyst.*, **16**(7), 2175–2196.

Turcotte, D.L. & Schubert, G., 2002. *Geodynamics,* Cambridge University Press.

van Keken, P. *et al.*, 2008. A community benchmark for subduction zone modelling, *Phys. Earth and Planet. Interiors,* **171,** (1-4), 187–197.

APPENDIX A: THE CONNECTION BETWEEN ELLIPTIC OPERATORS, WELL POSEDNESS OF THE NEWTON UPDATE EQUATION AND EIGENVALUES OF COEFFICIENTS

The discussion in Section 2.5 made use of the fact that a positive definite coefficient H^{spd} in the definition of the $\boldsymbol{J}^{uu}$ block of the matrix implies that the underlying differential operator is elliptic, and that consequently the $\boldsymbol{J}^{uu}$ block is invertible. Since this connection is not obvious unless one works daily with partial differential equations, let us discuss this step in slightly more detail in this appendix.

To this end, let us first note that we call an operator $\mathcal{A}$ acting on functions $u, v \in V$ (where V is a function space) bounded and coercive if there are constants $c > 0$, $C < \infty$ so that the following two conditions are satisfied:

$$\begin{aligned} &\text{boundedness :} && \langle u, \mathcal{A}v\rangle \le C\ ||u||_V\ ||v||_V && \forall u, v \in V,\\ &\text{coercivity :} && \langle u, \mathcal{A}u\rangle \ge c\ ||u||_V^2 && \forall u \in V. \end{aligned}$$

Here, $\langle\cdot,\cdot\rangle$ is the inner (or duality) product in V. It is well known from the theory of partial differential equations that such operators lead to unique solutions of the equation $\mathcal{A}u = f$, see Brenner & Scott (2002).

In the context of second-order partial differential operator – such as the one that governs the top left block of the Stokes problem –, an operator of the form $\mathcal{A}\mathbf{u} = -\text{div}\,[H\varepsilon(\mathbf{u})]$ acting on a velocity field $\mathbf{u} \in V = H_0^1(\Omega)^d$ is said to be elliptic if there exists $c_1 > 0$ so that

$$\tau : (H\tau) \ge c_1 \|\tau\|^2 \tag{A1}$$

for all symmetric tensors τ. Here, H is the rank-4 tensor that maps strain rates to stresses. For such operators, we have by integration by parts that $\langle \mathbf{u}, \mathcal{A}\mathbf{v}\rangle = (\varepsilon(\mathbf{u}), H\varepsilon(\mathbf{v})) = \int_\Omega \varepsilon(\mathbf{u}) : [H\varepsilon(\mathbf{v})]$. It is then easy to see that ellipticity implies coercivity by recognizing that $\|\mathbf{v}\|_V = \|\mathbf{v}\|_{H_0^1(\Omega)^d} = \left(\int_\Omega |\mathbf{v}|^2 + \int_\Omega |\nabla\mathbf{v}|^2\right)^{1/2}$ and knowing that $c_2\|\mathbf{v}\|_V \le \|\varepsilon(\mathbf{v})\| \le c_3\|\mathbf{v}\|_V$ for some constants $c_2 > 0$, $C_3 < \infty$.

In other words, if the coefficient H inside the differential operator satisfies condition (A1), then the associated operator is well posed and invertible. This condition carries over to the case where we let V be a finite-dimensional subspace of $H_0^1(\Omega)^d$ – for example, the set of all finite-element functions defined on a mesh.

The important realization is now that the constant c in eq. (A1) equals the smallest eigenvalue of H where we consider H as an operator that maps a symmetric tensor to a symmetric tensor. To see this, assume that H had a negative or zero eigenvalue λ. Then, we can choose σ as the corresponding eigenvector and obtain that $\sigma : (H\sigma) = \sigma : (\lambda\sigma) = \lambda\|\sigma\|^2 \le 0$, in violation of eq. (A1). Consequently, eq. (A1) can only be satisfied if all eigenvalues of H are positive.

This argument proves that if all eigenvalues of H are positive, then $\mathcal{A}\mathbf{u} = -\text{div}\,[H\varepsilon(\mathbf{u})]$ is elliptic and consequently coercive, and as a result the top left block $\boldsymbol{J}^{uu}$ of the matrix is positive definite and invertible.

It may be of interest to note that the operator $\mathcal{A}$ may be invertible even if it is not elliptic, that is, it is only *sufficient* but not *necessary* that H only have positive eigenvalues. However, it is much more difficult to specify the exact conditions that have to hold for H to ensure that $\mathcal{A}$ is invertible, and we will not attempt to do so here. This is particularly true if $H = H(\mathbf{x})$ is spatially variable with eigenvalues that may also be different from one location to another.

APPENDIX B: A LOOK AT SOME COMMON RHEOLOGIES

The results of Sections 2.4 and 2.5 were obtained for general rheologies in which the viscosity is a function of the strain rate $\varepsilon(\mathbf{u})$ and possibly the pressure p. However, if we know the specific form of this dependence, we can say more about whether or not it is necessary to symmetrize the matrix, and/or whether it is necessary to use a scaling factor α that is less than one. In the following, we will consider some common rheologies from this perspective.

B1 Power-law rheology

Some of the simplest rheology are of power-law type where the viscosity is defined as

$$\eta(\varepsilon(\mathbf{u})) = \eta_0^{-\frac{1}{n}} \left(\sqrt{\frac{1}{2}\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{u})}\right)^{\frac{1}{n}-1} = \eta_0^{-\frac{1}{n}} 2^{\frac{1}{2}-\frac{1}{2n}} \left(\|\varepsilon(\mathbf{u})\|^2\right)^{\frac{1}{2n}-\frac{1}{2}},$$

with $n \geq 1$. The form on the right shows that the viscosity is a function of the square of the norm of the strain rate. This implies that the matrix is automatically symmetric and does not have to be explicitly symmetrized with the procedure of Section 2.4.

Furthermore, we can compute the derivative of the viscosity with respect to the strain rate and obtain

$$\frac{\partial \eta(\varepsilon(\mathbf{u}))}{\partial \varepsilon} = \eta_0^{-\frac{1}{n}} 2^{\frac{1}{2}-\frac{1}{2n}} \left(\frac{1}{2n} - \frac{1}{2}\right) \left(\|\varepsilon(\mathbf{u})\|^2\right)^{\frac{1}{2n}-\frac{1}{2}-1} 2\varepsilon(\mathbf{u}) = \eta(\varepsilon(\mathbf{u})) \left(\frac{1}{n} - 1\right) \frac{\varepsilon(\mathbf{u})}{\|\varepsilon(\mathbf{u})\|^2}.$$

Identifying $a = \varepsilon(\mathbf{u})$ and $b = \frac{\partial \eta(\varepsilon(\mathbf{u}))}{\partial \varepsilon}$ as in Section 2.5, we see that the important term in the scaling factor definition (18) is

$$\left[1 - \frac{b : a}{\|a\|\|b\|}\right]^2 \|a\|\|b\| = [1 - (-1)]^2 \left|\frac{1}{n} - 1\right| \eta = 4\left(1 - \frac{1}{n}\right)\eta.$$

It is clear that this term grows with n toward a value of 4η and will eventually exceed its limit $c_{\text{safety}} 2\eta$ as defined in eq. (18). In particular, even if we choose $c_{\text{safety}} = 1$ (i.e. allow the smaller eigenvalue of H^{spd} to be equal to zero, then the condition will only be satisfied for $n \leq 2$, i.e. if the strain weakening is not too pronounced. For smaller values of c_{safety}, we can only choose $\alpha = 1$ if n is even less than that – for example, with $c_{\text{safety}} = 0.9$, the condition is only satisfied only if $n \lesssim 1.82$.

It is interesting to note that the condition is independent of the the flow field – what value α one has to choose is entirely decided by n and c_{safety}, and α will be the same at every quadrature point at which we integrate the bilinear form. Indeed, for this class of material model, we need to choose $\alpha = \min\left\{\frac{1}{2} c_{\text{safety}} \frac{n}{n-1}, 1\right\}$.

B2 Drucker–Prager rheology

For Drucker–Prager rheologies, the viscosity in 2-D is typically given by

$$\eta(\varepsilon(\mathbf{u})) = \frac{C\cos\phi + p\sin\phi}{2\sqrt{\frac{1}{2}\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{u})}} = \frac{C\cos\phi + p\sin\phi}{2\sqrt{\frac{1}{2}}} \left(\|\varepsilon(\mathbf{u})\|^2\right)^{-1/2}.$$

Here, C is the cohesion and ϕ the friction angle. As for the power-law case, the viscosity only depends on $\|\varepsilon(\mathbf{u})\|^2$ and we know that the resulting matrix will always be symmetric.

By comparing the formula for η with the one from the power-law rheology above, we see that up to a different (and possibly pressure dependent) pre-factor, the Prager–Drucker law corresponds to a power law with $n = \infty$. Thus, we expect that we will have to choose $\alpha < 1$ in eq. (18). Indeed, we can compute that

$$\frac{\partial \eta(\varepsilon(\mathbf{u}))}{\partial \varepsilon} = -\eta(\varepsilon(\mathbf{u})) \frac{\varepsilon(\mathbf{u})}{\|\varepsilon(\mathbf{u})\|^2},$$

which matches the corresponding formula for the power law with $n = \infty$. Thus,

$$\left[1 - \frac{b : a}{\|a\|\|b\|}\right]^2 \|a\|\|b\| = [1 - (-1)]^2 \ |-1| \ \eta = 4\eta,$$

which is of course *never* less than 2η and consequently *always* violates the necessary condition in eq. (18) to choose $\alpha = 1$. In other words, we can expect that the original Newton method will always lead to an ill-posed equation for the Newton update. On the other hand, eq. (18) tells us that the choice

$$\alpha = \frac{1}{2} c_{\text{safety}}$$

will always lead to a well-posed equation with $c_{\text{safety}} < 1$.

It is interesting to note that for both the power law rheology with large n and the Prager–Drucker rheology, one *always* needs to choose $\alpha < 1$. This implies that the equations that define our stabilized Newton update are *never* the derivative of the residual, and we can consequently not expect quadratic convergence. As the calculations above show, this has, in fact, nothing to do with the concrete test case or setup: the choice of α does not depend on the value of the strain rate or other solution variables at a given point, but is the same throughout the entire domain.

B3 The rheology of the Spiegelman *et al.* benchmark

To demonstrate that α does not need to be constant throughout the domain and may, in fact, depend on the flow field, we need to consider a rheology in which $\frac{\partial \eta}{\partial \varepsilon} : \varepsilon$ is not a fixed multiple of the viscosity as in the last two cases. This is indeed the case for the rheology of the benchmark by Spiegelman *et al.* discussed in Section 3.2. There, the viscosity is – up to a factor of 2 – given by the harmonic average of a linear rheology and the Drucker–Prager model considered above:

$$\eta(\varepsilon(\mathbf{u})) = \frac{1}{\frac{1}{\eta_{\text{ref}}} + \frac{1}{\eta_{\text{DP}}(\varepsilon(\mathbf{u}))}} = \frac{\eta_{\text{ref}}\ \eta_{\text{DP}}(\varepsilon(\mathbf{u}))}{\eta_{\text{ref}} + \eta_{\text{DP}}(\varepsilon(\mathbf{u}))}$$

where η_{ref} is a constant reference viscosity and $\eta_{\text{DP}}(\varepsilon(\mathbf{u}))$ is the viscosity computed with the Drucker–Prager rheology as shown above. The derivative of this equation is easily computed using the formulae from the previous section:

$$\begin{aligned}\frac{\partial\eta(\varepsilon(\mathbf{u}))}{\partial\varepsilon} &= \frac{\eta_{\text{ref}}^2}{(\eta_{\text{ref}}+\eta_{\text{DP}}(\varepsilon(\mathbf{u})))^2}\frac{\partial\eta_{\text{DP}}(\varepsilon(\mathbf{u}))}{\partial\varepsilon} = -\frac{\eta_{\text{ref}}^2\,\eta_{\text{DP}}(\varepsilon(\mathbf{u}))}{(\eta_{\text{ref}}+\eta_{\text{DP}}(\varepsilon(\mathbf{u})))^2}\frac{\varepsilon(\mathbf{u})}{\|\varepsilon(\mathbf{u})\|^2}\\ &= -\frac{\eta(\varepsilon(\mathbf{u}))^2}{\eta_{\text{DP}}(\varepsilon(\mathbf{u}))}\frac{\varepsilon(\mathbf{u})}{\|\varepsilon(\mathbf{u})\|^2}.\end{aligned}$$

Since this expression is again proportional to $\varepsilon(\mathbf{u})$, the matrix $\boldsymbol{J}^{uu}$ is again symmetric by construction, but not necessarily positive definite unless the expression

$$\left[1-\frac{b:a}{\|a\|\|b\|}\right]^2\|a\|\|b\| = [1-(-1)]^2\,\frac{\eta(\varepsilon(\mathbf{u}))^2}{\eta_{\text{DP}}(\varepsilon(\mathbf{u}))} = 4\frac{\eta(\varepsilon(\mathbf{u}))}{\eta_{\text{DP}}(\varepsilon(\mathbf{u}))}\eta(\varepsilon(\mathbf{u}))$$

is less than $c_{\text{safety}}2\eta(\varepsilon(\mathbf{u}))$. It is obvious that this is the case exactly if

$$\frac{\eta(\varepsilon(\mathbf{u}))}{\eta_{\text{DP}}(\varepsilon(\mathbf{u}))} \le \frac{1}{2}c_{\text{safety}},$$

which is equivalent to the condition $\eta_{\text{DP}}(\varepsilon(\mathbf{u})) \ge \left(1+\frac{2}{c_{\text{safety}}}\right)\eta_{\text{ref}}$. This condition makes sense since large values of η_{DP} (compared to η_{ref}) yield a roughly constant viscosity $\eta(\varepsilon(\mathbf{u})) \approx \eta_{\text{ref}}$ for which the negative contributions to $\boldsymbol{J}^{uu}$ are small and we can consequently choose $\alpha = 1$.

On the other hand, if the condition above is not satisfied, then eq. (18) tells us that we need to choose $\alpha = \frac{1}{2}c_{\text{safety}}\frac{\eta_{\text{DP}}(\varepsilon(\mathbf{u}))}{\eta(\varepsilon(\mathbf{u}))} = \frac{1}{2}c_{\text{safety}}\frac{\eta_{\text{ref}}+\eta_{\text{DP}}(\varepsilon(\mathbf{u}))}{\eta_{\text{ref}}}$. Because this comparison does not simplify to anything that is independent of $\varepsilon(\mathbf{u})$, the factor α that ensures positive definiteness will in general be spatially variable.

It is worth noting that the choice for α above is consistent with the results of the previous section. Namely, if one chooses $\eta_{\text{ref}} = \infty$, then the viscosity of this section equals the Prager–Drucker rheology. In that case, first, the condition $\eta_{\text{DP}} \ge \left(1+\frac{2}{c_{\text{safety}}}\right)\eta_{\text{ref}}$ can never be satisfied; and second, the choice for α just derived simplifies to $\alpha = \frac{1}{2}c_{\text{safety}}$, which is exactly the value we have obtained for the Prager–Drucker rheology before. Fig. 3 also nicely shows that this is exactly the behaviour we get in the benchmark: Where the strain rate is large, α drops to one-half, whereas it is close to or exactly one in areas where the viscosity is dominated by the background viscosity.

B4 The rheology of the Tosi *et al.* benchmark

Similar considerations also hold for the rheology used by the benchmark by Tosi *et al.* considered in Section 3.3. Indeed, based on the definition in eq. (21) and repeated application of the chain rule, we have that

$$\begin{aligned}\frac{\partial\eta(\varepsilon(\mathbf{u}))}{\partial\varepsilon} &= 2\left(\frac{1}{\eta_{\text{lin}}}+\frac{1}{\eta_{\text{plast}}(\varepsilon(\mathbf{u}))}\right)^{-2}\left(\eta_{\text{plast}}(\varepsilon(\mathbf{u}))\right)^{-2}\frac{\partial\eta_{\text{plast}}(\varepsilon(\mathbf{u}))}{\partial\varepsilon}\\ &= -2\left(\frac{1}{\eta_{\text{lin}}}+\frac{1}{\eta_{\text{plast}}(\varepsilon(\mathbf{u}))}\right)^{-2}\left(\eta_{\text{plast}}(\varepsilon(\mathbf{u}))\right)^{-2}\frac{\sigma_Y}{\|\varepsilon(\mathbf{u})\|}\frac{\varepsilon(\mathbf{u})}{\|\varepsilon(\mathbf{u})\|^2}\\ &= -\frac{1}{2}\left(\frac{\eta(\varepsilon(\mathbf{u}))}{\eta_{\text{plast}}(\varepsilon(\mathbf{u}))}\right)^2\left(\eta_{\text{plast}}(\varepsilon(\mathbf{u}))-\eta^*\right)\frac{\varepsilon(\mathbf{u})}{\|\varepsilon(\mathbf{u})\|^2},\end{aligned}$$

where we have omitted the dependence of η on T, z for the moment because it is not relevant to our considerations. Since this expression is again proportional to $\varepsilon(\mathbf{u})$, the matrix $\boldsymbol{J}^{uu}$ is again symmetric by construction, but not necessarily positive definite unless the expression

$$\left[1-\frac{b:a}{\|a\|\|b\|}\right]^2\|a\|\|b\| = 2\left(\frac{\eta(\varepsilon(\mathbf{u}))}{\eta_{\text{plast}}(\varepsilon(\mathbf{u}))}\right)^2\left(\eta_{\text{plast}}(\varepsilon(\mathbf{u}))-\eta^*\right)$$

is less than $c_{\text{safety}}2\eta$. As in the previous section, the factor α that ensures positive definiteness will in general again be spatially variable.

APPENDIX C: PARAMETERS FOR THE 3-D SUBDUCTION TEST CASE

The following table provides the numeric values of all material parameters used in the subduction example shown in Section 3.4:

In addition, we choose dimensionless scaling factors $\nu_{\text{disl}} = \nu_{\text{diff}} = 1$ in the upper mantle, and $\nu_{\text{disl}} = \nu_{\text{diff}} = 100$ in the lower mantle.

Thermal conductivity (W m^{-1} K^{-1})	4
Specific heat capacity (J kg^{-1} K^{-1})	1250
Reference temperature (K)	273.0
Reference densities (kg)	3300
Initial viscosity (Pa s)	10^{20}
Cohesion C (Pa)	20×10^{6}
Angle of internal friction ϕ (°)	30
Dislocation stress exponent n_{disl}	3
Dislocation pre-factor A_{disl} (Pa^{-n} s^{-1})	3.12504×10^{-14}
Dislocation activation energy E_{disl} (J mol^{-1})	4.3×10^{5}
Dislocation activation volume V_{disl} (m^{-3} mol^{-1})	25×10^{-6}
Diffusion stress exponent n_{diff}	1
Diffusion pre-factor A_{diff} (Pa^{-n} s^{-1})	1.92×10^{-11}
Diffusion activation energy E_{diff} (J mol^{-1})	335×10^{3}
Diffusion activation volume V_{diff} (m^{-3} mol^{-1})	4×10^{-6}
Minimum viscosity η_{min} (Pa s)	10^{19}
Maximum viscosity η_{max} (Pa s)	10^{24}