Supporting information for

The influence of grain crushing and pore collapse on the formation of faults

N.A. Collins-Craft[1,†], I. Stefanou[2,⋆], J. Sulem[1], and I. Einav[3]

[1]Laboratoire Navier, Ecole Nationale des Ponts et Chaussées, Institut Polytechnique de Paris, Université Gustave Eiffel, CNRS, Champs-sur-Marne, France
[2]IMSIA (UMR 9219), CNRS, EDF, CEA, ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France
[3]Particles and Grains Laboratory, The School of Civil Engineering, The University of Sydney, Sydney, Australia
[†]Previously at Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000, Grenoble, France
[⋆]Previously at GeM (Institute de Recherche en Genie Civile et Mécanique), École Centrale de Nantes, Université de Nantes, CNRS, Nantes, France

1 Introduction

This supporting information contains the details of the linear stability analysis and the numerical solution of the material model used in §4.2 of the main article, as well as the details of the consistent elastoplastic tangent operator and the finite element implementation used in §4.3 of the main article.

2 Linear stability analysis

In order to study the localisation behaviour of the model, we conduct a series of linear stability analyses by assuming that there is a homogeneous solution obeying the governing equations of the system, and then performing a perturbation about that solution. The kinematic fields may be written as

$$u_i(x_j,t) = u_i^0(x_j,t) + \tilde{u}_i(x_j,t), \tag{S.1}$$

$$\omega_i^c(x_j,t) = {\omega^{c}}_i^0(x_j,t) + \tilde{\omega}_i^c(x_j,t), \tag{S.2}$$

where the 0 superscript indicates the homogeneous solution and tilde the perturbation of the fields. In order to linearise the system, we assume that the perturbations follow the incremental constitutive relationship in (70).

Recalling (25), the momentum balance equations of the Cosserat continuum are given by

$$\tau_{ij,j} - f_i - \rho\ddot{u}_i = 0, \tag{S.3}$$

$$\mu_{ij,j} - \epsilon_{ijk}\tau_{jk} - b_i - I\ddot{\omega}_i = 0, \tag{S.4}$$

where ρ is the mass density and I is the rotational moment of inertia of the RVE, which we have assumed is isotropic. As we assume the density of the grains remains constant, but the solid fraction of the RVE can change, the expression for the density is

$$\rho = \phi\rho_s, \tag{S.5}$$

where ρ_s is the mass density of the grains (Vardoulakis and Sulem, 1995). We apply this expression for the density of the RVE to the expression for I that was derived via an inertial upscaling in Collins-Craft et al. (2020), which we adopt (in the three dimensional case) here:

$$I = \frac{\pi}{60}\phi(1-\theta_I B)\rho_s {d_{\mathrm{Max}}}^5. \tag{S.6}$$

The governing equations (S.3) and (S.4) are satisfied by the full kinematic fields, as well as the homogeneous solution. Thus, by subtraction they must also be satisfied by the perturbation terms. We assume the body forces do not experience any perturbation and so do not appear in the perturbative part of the balance equations. Hence, we may insert the incremental constitutive relationships (70) and the definitions of the strain and curvature rates (10) and (11) into (S.3) and (S.4) to obtain

$$E^{ep}_{ijkl}\left(\tilde{u}_{k,lj}+\epsilon_{klo}\tilde{\omega}^{c}_{o,j}\right)+F^{ep}_{ijkl}\tilde{\omega}^{c}_{k,lj}-\rho\ddot{\tilde{u}}_{i}=0, \tag{S.7}$$

$$K^{ep}_{ijkl}\left(\tilde{u}_{k,lj}+\epsilon_{klo}\tilde{\omega}^{c}_{o,j}\right)+M^{ep}_{ijkl}\tilde{\omega}^{c}_{k,lj}-\epsilon_{ijk}\left(E^{ep}_{jklo}\left(\tilde{u}_{l,o}+\epsilon_{lor}\tilde{\omega}^{c}_{r}\right)+F^{ep}_{jklo}\tilde{\omega}^{c}_{l,o}\right)-I_{ij}\ddot{\tilde{\omega}}^{c}_{j}=0. \tag{S.8}$$

This results in a set of 6 equations with 6 unknowns, $\tilde{u}_j$ and $\tilde{\omega}^c_j$, that admit solutions of the form:

$$U^{\star}_{i}(x_j,t)=\bar{U}_{i}e^{st+\frac{2\pi\mathrm{i}}{\Lambda}x_j\tilde{n}_j}, \tag{S.9}$$

$$\Omega^{\star}_{i}(x_j,t)=\bar{\Omega}_{i}e^{st+\frac{2\pi\mathrm{i}}{\Lambda}x_j\tilde{n}_j}, \tag{S.10}$$

where $\bar{U}_i$ and $\bar{\Omega}_i$ are the components of the vectors of coefficients, s is the (potentially complex-valued) Lyapunov exponent, $\mathrm{i}^2=-1$ defines the imaginary unit, Λ is the perturbation wavelength, $\tilde{n}_j$ the components of the polarisation vector and $U^{\star}_i(x_j,t)$ and $\Omega^{\star}_i(x_j,t)$ are the components of the vectors that contain the unknown perturbation fields.

We may insert (S.9) and (S.10) into (S.7) and (S.8) to obtain two sets of equations that may be re-written in terms of a matrix acting on the coefficient vectors $\bar{U}_i$ and $\bar{\Omega}_i$. The existence of non-trivial solutions implies that the matrix is singular. We take the determinant of this matrix and write the resultant expression as a polynomial in s. For a given s that is a root of this polynomial, if $\Re(s)<0$, the homogeneous system is stable as the perturbation decreases with time. If $\Re(s)>0$, the perturbation grows in time and localisation takes place. Hence, a bifurcation from the unstable homogeneous system to a stable perturbation-dominated system occurs. If $\Re(s)>0$ and $\Im(s)\neq 0$, travelling instabilities may appear (see Stathas and Stefanou (2023)). If $\Re(s)=0$, we cannot draw any conclusions.

Given a particular set of state variables describing the system, we search over a range of wavelengths Λ and orientations $\tilde{n}_j$ to find the largest value of real positive s, corresponding to the fastest growing perturbation. The wavelength at which this occurs is denoted Λ_{Max}, which we interpret as twice the width of the shear band. As the polynomial in s results in terms only in powers divisible by two $(s^{12}, s^{10}, \dots)$, we simplify the problem slightly by substituting $\bar{s}=s^2$ into the polynomial. This has the benefit of substantially stabilising the numerical root-finding scheme that we use. As in this application we only consider horizontal bands, we can simplify the system further by setting $\{\tilde{n}_1,\tilde{n}_2,\tilde{n}_3\}=\{1,0,0\}$. Then, we have a polynomial in $\bar{s}$ we denote $\mathrm{P}(\bar{s},\Lambda)$ where Λ is a parameter. Denoting by $\mathrm{roots}(\mathrm{P}(\bar{s},\Lambda))$ the set of (possibly complex-valued) roots of the polynomial, we are able to write a formal optimisation problem:

$$\begin{aligned} \underset{\Lambda}{\text{maximise}} \quad & \sup(\Re(\mathrm{roots}(\mathrm{P}(\bar{s},\Lambda)))), \\ \text{subject to} \quad & \Lambda_{\min}\leq\Lambda\leq\Lambda_{\text{limit}}, \end{aligned} \tag{S.11}$$

where $\Lambda_{\min}$ is the smallest wavelength we allow to be searched, and Λ_{limit} is the largest wavelength we allow to be searched. We set these values respectively to $d_{\min}$ and 200.0 mm. Numerically, we perform the optimisation using the NLopt software package (Johnson, 2007) *via* its implementation in the Julia programming language (Bezanson et al., 2017) using the gradient-free Nelder–Mead algorithm (Nelder and Mead, 1965), where the initial guess of the algorithm is determined by the result of the previous time step. For the initial time step, we start with a guess of $\Lambda_{\text{limit}}/2$. We allow the algorithm to make 25 iterations and apply a tolerance in the optimisation variables of 1×10^{-5}. For a given value of Λ provided by the optimisation solver, the roots are determined by PolynomialRoots.jl which implements the root-finding algorithm developed by Skowron and Gould (2012). The largest real part of the set of roots returned by this package is returned to the optimiser as the value of the objective function to maximise.

Finally, we emphasise that this analysis is only strictly valid up to and including the moment of localisation, as we are assuming a bifurcation from a homogeneous state, and once the shear band appears this is no longer the case. However, the results can be taken as indicative of the expected future behaviour of the system, as the shear band system is still relatively close to the homogeneous system in the state-space.

A possible alternative method of determining localisation is to form the "acoustic tensor" and search for the orientation at which its determinant is zero, however this approach gives no information about the wavelength of the perturbation, and so we prefer using the linear stability analysis to determine whether (and how) a system has localised. However, using the orientation vector $\tilde{n}_j$ that we have imposed, we can still form the acoustic tensor by

$$\Gamma_{ik}=\tilde{n}_j E_{ijkl}\tilde{n}_l, \tag{S.12}$$

and then obtain the eigenvector g_i of this tensor that is associated with the (in principle) zero eigenvalue. However, given the extra information used in the linear stability analysis (momentum, Cosserat effects), the orientation we use is not necessarily exactly the same orientation we would have achieved by varying $\tilde{n}_j$ to obtain $|\Gamma_{ik}|=0$, and as such the possibility remains open that the acoustic tensor is not singular at the point of localisation. As such, we consider the eigenvector associated with the eigenvalue with the norm closest to zero. The inner product of this eigenvector g_i with $\tilde{n}_i$ characterises the nature of the predicted deformation band, with $\tilde{n}_i g_i = 1$ a pure compaction band, $\tilde{n}_i g_i = 0$ a pure shear band, $\tilde{n}_i g_i = -1$ a pure dilation band, $-1 < \tilde{n}_i g_i < 0$ a dilating shear band and $0 < \tilde{n}_i g_i < 1$ a compacting shear band.

3 Numerical method

While the plastic multiplier is in-principle calculable analytically (and indeed we ultimately do so to calculate the terms in (71), (72), (73), (74)), in practice it is quite onerous, and doing so renders implicit numerical methods unworkably slow. In the event that some entries are under stress control, the model must strictly be solved implicitly as explicit methods will fail to respect the stress rate conditions. We thus make use of a predictor-corrector method, where the value of λ is determined numerically to be that which best satisfies a root-finding problem.

We start by considering a time interval $t \in [t^0, t^{\mathrm{Max}}]$, which we discretise by the sequence $\{t^0, t^1, \dots, t^{\mathsf{k}}, \dots, t^{\mathsf{Max}}\}$, where the the terms are indexed by k and the difference between subsequent terms of the sequence is given by $(\Delta t)^{\mathsf{k}+1}$. Then, we can discretise the underlying translational and rotational velocities by $\dot{u}^{\mathsf{k}} \approx \dot{u}(t^{\mathsf{k}})$ and $\dot{\omega}^{c\mathsf{k}} \approx \dot{\omega}^c(t^{\mathsf{k}})$. The discretisation of all the other terms follows straightforwardly. We apply an implicit Euler time-stepping scheme, that is

$$\int_{t^{\mathsf{k}}}^{t^{\mathsf{k}+1}} x(t)\,\mathrm{d}t \approx (\Delta t)^{\mathsf{k}+1} x^{\mathsf{k}+1}, \tag{S.13}$$

for some arbitrary quantity x. We have chosen the implicit Euler scheme due to its numerical stability, allowing us to use rather large time steps to integrate the system. We consider the set of augmented state variables $\{{\gamma^e_{ij}}^{\mathsf{k}}, {\kappa^e_{ij}}^{\mathsf{k}}, \rho^{\mathsf{k}}, \phi^{\mathsf{k}}, B^{\mathsf{k}}, \lambda^{\mathsf{k}}\}$ (that is to say we add the plastic multiplier to the vector of the true state variables), with all other quantities evaluated at this time using these variables also denoted with a k superscript (*e.g.* ${\tau_{ij}}^{\mathsf{k}}, \chi^{\mathsf{k}}, y^{\mathsf{k}}, \dots$). We consider the discrete-in-time evolution of the state variables by dividing the problem at each time step into two stages, an elastic predictor and a plastic corrector. First, we treat the case where all the strain and curvature rates are known (and the corresponding stress and couple-stress rates are unknown), then consider the modifications for the case where some of the stress and couple-stress rates are known (and the corresponding strain and curvature rates are unknown). For convenience, we will label the two cases fully strain-driven and partially strain-driven respectively.

3.1 Fully strain-driven

In this case, all the elements of ${\dot{\gamma}_{ij}}^{\mathsf{k}+1}$ and ${\dot{\kappa}_{ij}}^{\mathsf{k}+1}$ are known, and correspondingly all the elements of ${\dot{\tau}_{ij}}^{\mathsf{k}+1}$ and ${\dot{\mu}_{ij}}^{\mathsf{k}+1}$ are unknown. We initially assume that the step is elastic (*i.e.* $\lambda^{\mathsf{trial}} = 0$ and $B^{\mathsf{trial}} = B^{\mathsf{k}}$) and the strain and curvature rates are accumulated to the elastic strains and curvatures to obtain the trial values ${\gamma^e_{ij}}^{\mathsf{trial}}$ and ${\kappa^e_{ij}}^{\mathsf{trial}}$.

Considering the density evolution equation given in (49), by our use of a fully implicit method we have a known and constant value of $\dot{\varepsilon}_v^{\mathsf{k}+1}$, making the equation a straightforward first order linear ordinary differential equation with constant coefficient over the time step, solved by an equation of the form $\rho(t) = C\exp(\dot{\varepsilon}_v^{\mathsf{k}+1} t)$. We have at the start of the time step at t^{k} that $\rho(t) = \rho^{\mathsf{k}}$ and hence $C = \rho^{\mathsf{k}}/\exp(\dot{\varepsilon}_v^{\mathsf{k}+1} t^{\mathsf{k}})$. Then, to get the trial density at the end of the time step we straightforwardly have

$$\rho^{\mathsf{trial}} = \frac{\rho^{\mathsf{k}} \exp(\dot{\varepsilon}_v^{\mathsf{k}+1} t^{\mathsf{k}+1})}{\exp(\dot{\varepsilon}_v^{\mathsf{k}+1} t^{\mathsf{k}})} = \rho^{\mathsf{k}} \exp(\Delta \varepsilon_v^{\mathsf{k}+1}). \tag{S.14}$$

It should be noted that in the fully strain-driven case, the trial density is identical to the final density, as the value is insensitive to how the total strain is split between elastic and plastic parts, nor does it depend on the breakage or the solid fraction.

Considering the elastic solid fraction evolution equation given in (62), expanding χ by its definition in (17), noting that as $B^{\mathsf{trial}} = B^{\mathsf{k}}$ the values of $\phi_{\min}$ and ϕ_{Max} for the trial state are taken at t^{k} and again taking the volumetric strain rate exclusively at $t^{\mathsf{k}+1}$ we have

$$\dot{\phi} = \phi\left(1 - \frac{\phi - \phi^{\mathsf{k}}_{\min}}{\phi^{\mathsf{k}}_{\mathrm{Max}} - \phi^{\mathsf{k}}_{\min}}\right)\dot{\varepsilon}_v^{\mathsf{k}+1} = -\frac{\dot{\varepsilon}_v^{\mathsf{k}+1}}{\phi^{\mathsf{k}}_{\mathrm{Max}} - \phi^{\mathsf{k}}_{\min}}\phi^2 + \left(1 + \frac{\phi^{\mathsf{k}}_{\min}}{\phi^{\mathsf{k}}_{\mathrm{Max}} - \phi^{\mathsf{k}}_{\min}}\right)\dot{\varepsilon}_v^{\mathsf{k}+1}\phi, \tag{S.15}$$

which is a Bernoulli differential equation with constant coefficients over the time step. We have two possible analytical solution cases which can be straightforwardly obtained by separation of variables. The first case is when the deformation is isochoric (*i.e.* $\dot{\varepsilon}_v^{\mathsf{k}+1} = 0$) in which case we trivially have $\phi^{\mathsf{trial}} = \phi^{\mathsf{k}}$. The second case is when the deformation is non-isochoric and requires some manipulation. Labelling the term that multiplies ϕ^2 in (S.15) as $\mathsf{A}\dot{\varepsilon}_v^{\mathsf{k}+1}$ and the term that multiplies ϕ as $\mathsf{B}\dot{\varepsilon}_v^{\mathsf{k+1}}$, we have

$$\mathsf{C} = \frac{\phi^{\mathsf{k}}}{\mathsf{A}\phi^{\mathsf{k}} + \mathsf{B}}, \tag{S.16}$$

$$\phi^{\mathsf{trial}} = \frac{\mathsf{CB}\exp(\mathsf{B}\Delta\varepsilon_v^{\mathsf{k+1}})}{1 - \mathsf{AC}\exp(\mathsf{B}\Delta\dot{\varepsilon}_v^{\mathsf{k+1}})}. \tag{S.17}$$

Then, y^{trial} is evaluated, and if it is less than zero, the trial values are accepted as the values at $t^{\mathsf{k+1}}$ and we set $\{\gamma_{ij}^{e\ \mathsf{k+1}}, \kappa_{ij}^{e\ \mathsf{k+1}}, \rho^{k+1}, \phi^{k+1}, B^{\mathsf{k+1}}, \lambda^{\mathsf{k+1}}\} = \{\gamma_{ij}^{e\ \mathsf{trial}}, \kappa_{ij}^{e\ \mathsf{trial}}, \rho^{\mathsf{trial}}, \phi^{\mathsf{trial}}, B^{\mathsf{k}}, 0\}$ and the system advances to the next time step.

In the event that $y^{\mathsf{trial}} \geq 0$, plastic correction is applied by means of a classical return-mapping algorithm where the total strain is held constant within the correction step. We denote the internal iterations of the correction step with the $(\mathsf{n}+1)$ superscript, and we set the initial internal iterate of the augmented state variables $\{\gamma_{ij}^{e\ (1)}, \kappa_{ij}^{e\ (1)}, \rho^{(1)}, \phi^{(1)}, B^{(1)}, \lambda^{(1)}\}$ to $\{\gamma_{ij}^{e\ \mathsf{trial}}, \kappa_{ij}^{e\ \mathsf{trial}}, \rho^{\mathsf{trial}}, \phi^{\mathsf{trial}}, B^{k}, 1\}$. The initial iterate value of $\lambda^{(1)} = 1$ is chosen to aid the convergence of the system. Then, the system of equations to be solved is

$$\begin{bmatrix} \gamma_{ij}^{e\ (\mathsf{n+1})} - \gamma_{ij}^{e\ \mathsf{trial}} + (\Delta t)^{\mathsf{k+1}}\lambda^{(\mathsf{n+1})}\overline{\gamma_{ij}^{p}}^{(\mathsf{n+1})} \\ \kappa_{ij}^{e\ (\mathsf{n+1})} - \kappa_{ij}^{e\ \mathsf{trial}} + (\Delta t)^{\mathsf{k+1}}\lambda^{(\mathsf{n+1})}\overline{\kappa_{ij}^{p}}^{(\mathsf{n+1})} \\ \rho^{(\mathsf{n+1})} - \rho^{\mathsf{trial}} \\ \phi^{(\mathsf{n+1})} - \phi^{\mathsf{trial}} - (\Delta t)^{\mathsf{k+1}}\lambda^{(\mathsf{n+1})}\chi^{(\mathsf{n+1})}\overline{\phi^{p}}^{(\mathsf{n+1})} \\ B^{(\mathsf{n+1})} - B^{\mathsf{trial}} - (\Delta t)^{\mathsf{k+1}}\lambda^{(\mathsf{n+1})}\overline{B}^{(\mathsf{n+1})} \\ y^{(\mathsf{n+1})} \end{bmatrix} = \begin{bmatrix} 0_{ij} \\ 0_{ij} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{S.18}$$

To solve this system we use a Newton–Raphson root finding method (we use the one implemented in the package NonlinearSolve.jl (Pal et al., 2024) with gradients calculated using forward-mode automatic differentiation through the package ForwardDiff.jl (Revels et al., 2016)). In the event that the solver indicates successful convergence, the state variables are updated to the values of the internal iterates that converged *i.e.* $\{\gamma_{ij}^{e\ \mathsf{k+1}}, \kappa_{ij}^{e\ \mathsf{k+1}}, \rho^{\mathsf{k+1}}, \phi^{\mathsf{k+1}}, B^{\mathsf{k+1}}, \lambda^{\mathsf{k+1}}\} = \{\gamma_{ij}^{e\ (\mathsf{n+1})}, \kappa_{ij}^{e\ (\mathsf{n+1})}, \rho^{(\mathsf{n+1})}, \phi^{(\mathsf{n+1})}, B^{(\mathsf{n+1})}, \lambda^{(\mathsf{n+1})}\}$.

The plastic corrector resembles the closest-point projection algorithm, however as Breakage Mechanics models — in common with Damage Mechanics models — do not generally possess the requisite convexity, we do not have guarantees of convergence to a unique solution. In practice the method is known to work for Damage Mechanics, albeit with a degradation in performance as damage (or breakage in our case) increases, which can be mitigated with better initial guesses (smaller time steps in our case), augmenting the Newton–Raphson method with line search or using a "projected stress" method (De Souza Neto et al., 1994, 2008). For our single element simulations in §4 of the main article we find that constant time steps are perfectly adequate, but for our finite element simulations in §4.3 of the main article we allow adaptive time-stepping where required to ensure convergence. Typically, the system requires small time steps during localisation as the solution is non-unique.

3.2 Partially strain-driven

In this case, some of the elements of $\dot{\gamma}_{ij}{}^{\mathsf{k+1}}$ and $\dot{\kappa}_{ij}{}^{\mathsf{k+1}}$ are known and some are unknown, and the corresponding elements of $\dot{\tau}_{ij}{}^{\mathsf{k+1}}$ and $\dot{\mu}_{ij}{}^{\mathsf{k+1}}$ are unknown or known, respectively. For simplicity, we will refer to the entries with known strain- or curvature-rates as strain-controlled, and those with known stress- or couple-stress-rates as stress-controlled. We once again make the assumption that the step is elastic ($\lambda^{\mathsf{trial}} = 0$, $B^{\mathsf{trial}} = B^{\mathsf{k}}$) and accumulate the strain and curvature rates to the elastic strains and curvatures. For the entries that are stress-controlled, and hence where the corresponding $\dot{\gamma}_{ij}{}^{\mathsf{k+1}}$ or $\dot{\kappa}_{ij}{}^{\mathsf{k+1}}$ entries are unknown, we use the values of the rate entries at t^{k} as an initial guess. Then, we initialise the guess for the density and the solid fraction by evaluating

$$\rho^{(1)} = \rho^{k}\exp\left(\varepsilon_v^{e\,(1)} - \varepsilon_v^{e\,\mathsf{k}}\right), \tag{S.19}$$

$$\phi^{(1)} = \frac{\mathsf{C}^{(1)}\mathsf{B}^{(1)}\exp(\mathsf{B}^{(1)}(\Delta t)^{\mathsf{k+1}})}{1 - \mathsf{A}^{(\mathsf{n+1})}\mathsf{C}^{(1)}\exp(\mathsf{B}^{(1)}(\Delta t)^{\mathsf{k+1}})}. \tag{S.20}$$

We form a residual by

$$\mathsf{Residual} = \begin{bmatrix} \mathsf{stress\text{-}controlled} \begin{cases} \tau_{ij}{}^{(1)} - \tau_{ij}{}^{\mathsf{k}} - (\Delta t)^{\mathsf{k+1}}\dot{\tau}_{ij}{}^{\mathsf{k+1}} \\ \mu_{ij}{}^{(1)} - \mu_{ij}{}^{\mathsf{k}} - (\Delta t)^{\mathsf{k+1}}\dot{\mu}_{ij}{}^{\mathsf{k+1}} \end{cases} \end{bmatrix}, \tag{S.21}$$

where the stress-controlled label indicates that the discrete stress rate balance is only evaluated for the stress-controlled entries, and then we take the L^2 norm of the residual. We vary only the unknown entries of the elastic strain and curvature, recalculating (S.19) and (S.20) at each iteration and subsequently calculating the norm of the residual until it is below the required tolerance. We once again perform the optimisation using the NLopt software package (Johnson, 2007) using the gradient-free Nelder–Mead algorithm (Nelder and Mead, 1965). If the solver successfully converges, the trial state variables are updated to the values of the internal iterates that converged *i.e.* $\{{\gamma^e_{ij}}^{\mathsf{trial}}, {\kappa^e_{ij}}^{\mathsf{trial}}, \rho^{\mathsf{trial}}, \phi^{\mathsf{trial}}, B^{\mathsf{trial}}, \lambda^{\mathsf{trial}}\} = \{{\gamma^e_{ij}}^{(\mathsf{n+1})}, {\kappa^e_{ij}}^{(\mathsf{n+1})}, \rho^{(\mathsf{n+1})}, \phi^{(\mathsf{n+1})}, B^{\mathsf{k}}, 0\}$. Then, y^{trial} is evaluated, and if it is less than zero, the trial values are accepted as the values at $t^{\mathsf{k+1}}$ and we set $\{{\gamma^e_{ij}}^{\mathsf{k+1}}, {\kappa^e_{ij}}^{\mathsf{k+1}}, \rho^{\mathsf{k+1}}, \phi^{\mathsf{k+1}}, B^{\mathsf{k+1}}, \lambda^{\mathsf{k+1}}\} = \{{\gamma^e_{ij}}^{\mathsf{trial}}, {\kappa^e_{ij}}^{\mathsf{trial}}, \rho^{\mathsf{trial}}, \phi^{\mathsf{trial}}, B^{\mathsf{k}}, 0\}$. These values are used to back-calculate the strain and curvature rates for the entries in which those quantities are unknown, and then the system advances to the next time step.

In the event that $y^{\mathsf{trial}} \geq 0$, plastic correction must once again be applied, this time using a modified return-mapping algorithm similar to that typically used in plane-stress plasticity. In this case, the total strain cannot be held constant within the correction step, as this quantity may need to be varied in order to allow the solver to respect the stress-controlled conditions. This results in the elastic part of the solid fraction rate needing to be taken into account during the plastic correction, producing a somewhat more complicated expression. We once again denote the internal iterations of the correction step with the $(\mathsf{n}+1)$ superscript, and we set the initial internal iterate of the augmented state variables $\{{\gamma^e_{ij}}^{(1)}, {\kappa^e_{ij}}^{(1)}, \rho^{(1)}, \phi^{(1)}, B^{(1)}, \lambda^{(1)}\}$ to $\{{\gamma^e_{ij}}^{\mathsf{trial}}, {\kappa^e_{ij}}^{\mathsf{trial}}, \rho^{\mathsf{trial}}, \phi^{\mathsf{trial}}, B^{\mathsf{trial}}, 1\}$. Then, the system of equations to be solved is

$$\begin{bmatrix} \textsf{strain-controlled}\begin{cases} {\gamma^e_{ij}}^{(\mathsf{n+1})} - {\gamma^e_{ij}}^{\mathsf{trial}} + (\Delta t)^{(\mathsf{k+1})}\lambda^{(\mathsf{n+1})}\overline{\gamma^p_{ij}}^{(\mathsf{n+1})} \\ {\kappa^e_{ij}}^{(\mathsf{n+1})} - {\kappa^e_{ij}}^{\mathsf{trial}} + (\Delta t)^{(\mathsf{k+1})}\lambda^{(\mathsf{n+1})}\overline{\kappa^p_{ij}}^{(\mathsf{n+1})} \end{cases} \\ \textsf{stress-controlled}\begin{cases} {\tau_{ij}}^{(\mathsf{n+1})} - {\tau_{ij}}^{\mathsf{k}} - (\Delta t)^{\mathsf{k+1}}{\dot{\tau}_{ij}}^{\mathsf{k+1}} \\ {\mu_{ij}}^{(\mathsf{n+1})} - {\mu_{ij}}^{\mathsf{k}} - (\Delta t)^{\mathsf{k+1}}{\dot{\mu}_{ij}}^{\mathsf{k+1}} \end{cases} \\ \rho^{(\mathsf{n+1})} - \rho^{\mathsf{k}}\exp\left({\varepsilon^e_v}^{(\mathsf{n+1})} - {\varepsilon^e_v}^{\mathsf{k}} + (\Delta t)^{(\mathsf{k+1})}\lambda^{(\mathsf{n+1})}\overline{\gamma^p_v}^{(\mathsf{n+1})}\right) \\ \left[1 - \left(1 - \chi^{(\mathsf{n+1})}\right)\left({\varepsilon^e_v}^{(\mathsf{n+1})} - {\varepsilon^e_v}^{\mathsf{k}}\right)\right]\phi^{(\mathsf{n+1})} - \phi^{\mathsf{k}} - (\Delta t)^{\mathsf{k+1}}\lambda^{(\mathsf{n+1})}\overline{\phi^p}^{(\mathsf{n+1})} \\ B^{(\mathsf{n+1})} - B^{\mathsf{trial}} - (\Delta t)^{(\mathsf{k+1})}\lambda^{(\mathsf{n+1})}\overline{B}^{(\mathsf{n+1})} \\ y^{(\mathsf{n+1})} \end{bmatrix} = \begin{bmatrix} \textsf{strain-controlled}\begin{cases} 0_{ij} \\ 0_{ij} \end{cases} \\ \textsf{stress-controlled}\begin{cases} 0_{ij} \\ 0_{ij} \end{cases} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \text{(S.22)}$$

where the strain-controlled label indicates that the constant total strain balance is only evaluated for the strain-controlled entries, the stress-controlled label indicates that the discrete stress rate balance is only evaluated for the stress-controlled entries, $\overline{\gamma^p_v}$ indicates the volumetric part of $\overline{\gamma^p_{ij}}$ and the system of equations in solved via the Newton–Raphson root finding method in NonlinearSolve.jl. In the event that the solver indicates successful convergence, the state variables are updated to the values of the internal iterates that converged *i.e.* $\{{\gamma^e_{ij}}^{\mathsf{k+1}}, {\kappa^e_{ij}}^{\mathsf{k+1}}, \rho^{\mathsf{k+1}}, \phi^{\mathsf{k+1}}, B^{\mathsf{k+1}}, \lambda^{\mathsf{k+1}}\} = \{{\gamma^e_{ij}}^{(\mathsf{n+1})}, {\kappa^e_{ij}}^{(\mathsf{n+1})}, \rho^{(\mathsf{n+1})}, \phi^{(\mathsf{n+1})}, B^{(\mathsf{n+1})}, \lambda^{(\mathsf{n+1})}\}$. These values are used to back-calculate the elastic strain and curvature rates for the stress-controlled entries, and the corresponding plastic rates can be obtained by evaluating the flow rules using the updated state variables. These two quantities can then be summed to obtain the total strain and curvature rates for the stress-controlled entries, which are used to improve the quality of the initial guess at the next time step.

We emphasise that in (S.22), the strains and curvatures in the stress-controlled entries still contribute to the value of $\overline{\gamma^p_{ij}}^{(\mathsf{n+1})}$ and $\overline{\kappa^p_{ij}}^{(\mathsf{n+1})}$, but do not have to fulfil the strain and curvature balances. Similarly, the strains and curvatures in the strain-controlled entries still contribute to the value of ${\tau_{ij}}^{(\mathsf{n+1})}$ and ${\mu_{ij}}^{(\mathsf{n+1})}$, but do not have to fulfil the stress and couple-stress balances. As an example, for a system where a constant confining stress is applied on the τ_{11} entry, but all other entries are strain-controlled, we must strictly have ${\tau_{11}}^{(\mathsf{n+1})} - {\tau_{11}}^{\mathsf{k}} - \Delta t^{\mathsf{k+1}}{\dot{\tau}_{11}}^{\mathsf{k+1}} = 0$ but ${\gamma^e_{11}}^{(\mathsf{n+1})} - {\gamma^e_{11}}^{\mathsf{trial}} + (\Delta t)^{(\mathsf{k+1})}\lambda^{(\mathsf{n+1})}\overline{\gamma^p_{11}}^{(\mathsf{n+1})} \neq 0$ (in general, it may incidentally be equal to zero). Once again, we find that constant time steps are perfectly adequate for our simulations in §4 of the main article.

3.3 Consistent elastoplastic tangent operator

For our finite element implementation, we require the consistent elastoplastic tangent operator that will be assembled to create the global stiffness matrix that is in turn used to satisfy global equilibrium of the system. As the (Lagrangian) finite element implementation is displacement-based, the solver varies the nodal displacements and as a consequence the strains at each Gauss point are imposed. Hence, the consistent elastoplastic operator is obtained from the fully strain-driven return-mapping scheme described in §3.1.

To obtain the consistent operator, we first define the incremental (generalised) stress update function using the compact notation $\bar{\sigma}^{\mathsf{k+1}}$ where σ includes both the true stresses $\tau^{\mathsf{k+1}}_{ij}$ and the couple stresses $\mu^{\mathsf{k+1}}_{ij}$, and consider the (generalised) strains $\varepsilon^{\mathsf{k+1}}$ which includes the strains $\gamma^{\mathsf{k+1}}_{ij}$ and the curvatures $\kappa^{\mathsf{k+1}}_{ij}$. Here, σ and ε are distinct from the Cauchy stresses and strains. Then, we can define the incremental stress update function as

$$\bar{\sigma}^{\mathsf{k}+1} = \bar{\sigma}(\varepsilon^{e\,\mathsf{trial}}, \rho^{\mathsf{trial}}, \phi^{\mathsf{k}}, B^{\mathsf{k}}), \tag{S.23}$$

where $\bar{\sigma}^{\mathsf{k}+1}$ is an implicit function that encapsulates the algorithmic process of applying the trial strains, checking the yield condition and either returning the generalised stresses directly in the case of elastic loading, or performing the return-mapping process (S.18) and returning the resulting generalised stresses in the case of plastic loading. Then, the consistent elastoplastic tangent operator is obtained by differentiation:

$$\mathsf{D}^{ep} = \frac{\partial \bar{\sigma}^{k+1}}{\partial \varepsilon^{e\,\mathsf{trial}}}. \tag{S.24}$$

In the case where loading is purely elastic, the operator is equivalent to the continuum elastic stiffness matrices. In the case of plastic loading, the consistent operator differs slightly from the continuum incremental elastoplastic tensors given in (70), (71), (72), (73) and (74), however as the step size $\Delta t \to 0$, the consistent operator converges to the continuum value. In combination with the Newton–Raphson method, the consistent tangent operator allows optimal convergence during the nonlinear solution of the global finite element balance equations (De Souza Neto et al., 2008).

In our implementation of the operator, we once again obtain our desired expression using automatic differentiation *via* ForwardDiff.jl (Revels et al., 2016). It should be noted that as the global finite element solver can pass increments of strain that are too large and cause the return-mapping process in (S.18) to fail, we implement a check within our implementation of $\bar{\sigma}^{\mathsf{k}+1}$ where if NonlinearSolve.jl returns an error code indicating it could not solve (S.18), $\bar{\sigma}^{\mathsf{k}+1}$ returns the generalised stresses associated with the purely elastic trial state. As this results in an out-of-equilibrium system, the solver tries again with a different increment of strain until convergence is obtained.

In general, we expect that the structural stiffness matrix will be singular as localisation is occurring. This can make it very difficult for the solver to converge, even though the consistent tangent operator has optimal convergence properties (in practice, at a given time step size the solver will cycle between several points until the iteration limit is achieved and the adaptive time-stepping is activated). In order to mitigate this problem, we allow the option to add part of the elastic stiffness matrix to the consistent tangent operator. Specifically, we add 10% of the elastic stiffness matrices to the operator. While this slightly compromises the convergence rate in elasticity and in homogeneous plasticity (or cases where the localisation is already well-established), it substantially aids convergence during localisation, with a more-or-less monotonic decrease in the error with internal iterations. This has the effect of speeding up the overall solution time.

4 Finite element implementation

In order to examine the system at more than one material point (*i.e.* when our model has a real spatial extent), we turn to the finite element method. As we do not wish to examine any transitory phenomena such as stress waves, and our problem is located at sufficient depth that the variation in gravitational forces with depth can be neglected, we consider the static momentum balance equations of the Cosserat continuum without body forces, namely:

$$\tau_{ij,j} = 0, \tag{S.25}$$

$$\mu_{ij,j} - \epsilon_{ijk}\tau_{jk} = 0, \tag{S.26}$$

Labelling the Dirichlet and Neumann boundary conditions as in §2.5, the weak form of the balance equations can be written as

$$-\int_{\mathcal{V}} \tau_{ij}\psi_{i,j}\,\mathrm{d}\mathcal{V} + \int_{\partial\mathcal{V}_N} \tau_{ij}\bar{n}_j\,\mathrm{d}\partial\mathcal{V} = 0, \tag{S.27}$$

$$-\int_{\mathcal{V}} \mu_{ij}\psi_{i,j}\,\mathrm{d}\mathcal{V} + \int_{\partial\mathcal{V}_N} \mu_{ij}\bar{n}_j\psi_i\,\mathrm{d}\partial\mathcal{V} - \int_{\mathcal{V}} \epsilon_{ijk}\tau_{jk}\psi_i\,\mathrm{d}\mathcal{V} = 0, \tag{S.28}$$

where ψ_i are test functions. We choose quadratic Lagrangian test functions for the displacement fields and linear Lagrangian test functions for the micro-rotation fields, as this has been shown to deliver performance that is superior to both doubly-linear and doubly-quadratic formulations in plane geometries (Providas and Kattis, 2002) and we are unconcerned with the out-of-plane bending behaviour for which a doubly-quadratic formulation has demonstrated superiority (Godio et al., 2015). We use full integration, which in the case of our one-dimensional elements amounts to two Gauss points per element. The system is integrated using the Numerical Geolab framework (Stefanou and Stathas, 2023), which provides an interface layer on top of the FEniCS finite element framework (Logg et al., 2012; Alnæs et al., 2015). Numerical Geolab allows us to control the global error tolerance for convergence, the adaptive time-stepping and the distribution of the materials. The results of the simulation are written to .h5 files indexed by .xdmf files on an element-by-element basis (that is the state variables written to file will be the average of the state variable values at each Gauss point in the element).

For the one dimensional simulations, we use a system with a height of 50 mm so that the initial localisation widths (on the order of at most 20 mm) avoid any influence from the boundary conditions. In order to ensure that the initial localisation occurs in the centre of the simulation, we induce an imperfection of 2 mm with the same material parameters, other than an E_c value set to 95% of the calibrated value. As the imperfection is much smaller than the predicted localisation size, it does not compromise the predicted width. We utilise 200 elements so that we have a finely resolved system. Within the material model (*i.e.* the return-mapping algorithm) we set a numerical tolerance of 1×10^{-8} on the norm of the residual and allow up to 100 iterations. For the global finite element system, we set a numerical tolerance for convergence of 1×10^{-2} in the L_2-norm of the product of the variational Jacobian and the residual. We allow up to 200 iterations before declaring that the time-step has failed to converge. Numerical Geolab uses adaptive time-stepping, with the time-step being divided by two if convergence fails. If the time step is not at the maximum size, and successfully converges five times in a row, the time step size will be doubled (up to the limit of the maximum time step size). If convergence failures continue and the time step size falls below the minimal value (1×10^{-8}), the simulation is terminated. All simulations are assumed to take place over an arbitrary time of 1, as this is a pseudo-time rather than a real time as the system is rate-independent and integrated quasi-statically. We set the maximum time step size to 0.005.

References

Alnæs, M.S. et al. (2015). “The FEniCS Project Version 1.5”. In: *Archive of Numerical Software* 3.100, pp. 9–23.

Bezanson, J. et al. (Jan. 2017). “Julia: A Fresh Approach to Numerical Computing”. In: *SIAM Review* 59.1, pp. 65–98. ISSN: 0036-1445. DOI: 10.1137/141000671. URL: http://www.siam.org/journals/sirev/59-1/100067.html.

Collins-Craft, N.A. et al. (Aug. 2020). “A Cosserat Breakage Mechanics Model for Brittle Granular Media”. In: *Journal of the Mechanics and Physics of Solids* 141, p. 103975. ISSN: 00225096. DOI: 10.1016/j.jmps.2020.103975.

De Souza Neto, E.A., D. Perić and D.R.J. Owen (Mar. 1994). “A Model for Elastoplastic Damage at Finite Strains: Algorithmic Issues and Applications”. In: *Engineering Computations* 11.3, pp. 257–281. ISSN: 0264-4401. DOI: 10.1108/02644409410799272.

— (Oct. 2008). *Computational Methods for Plasticity: Theory and Applications*. 1st ed. Wiley. ISBN: 978-0-470-69452-7 978-0-470-69462-6. DOI: 10.1002/9780470694626.

Godio, M. et al. (Mar. 2015). “Dynamic Finite Element Formulation for Cosserat Elastic Plates”. In: *International Journal for Numerical Methods in Engineering* 101.13, pp. 992–1018. ISSN: 00295981. DOI: 10.1002/nme.4833.

[SW] Johnson, S.G., *The NLopt nonlinear-optimization package* 2007. SWHID: ⟨swh:1:dir:0a9fa204d8ed9e2f88795013ddd59cfb1a2029e7;origin=https://github.com/stevengj/nlopt;visit=swh:1:snp:46dc48703f89a2764bb304190924820cdf6ced8e;anchor=swh:1:rev:7a7587e5ef1cb15f412515d852d1fc261c863e96⟩.

Logg, A., K.-A. Mardal and G. Wells, eds. (2012). *Automated Solution of Differential Equations by the Finite Element Method*. Vol. 84. 9. Berlin, Heidelberg: Springer Berlin Heidelberg. 1689-1699. ISBN: 978-3-642-23098-1. DOI: 10.1007/978-3-642-23099-8. URL: http://link.springer.com/10.1007/978-3-642-23099-8.

Nelder, J.A. and R. Mead (1st Jan. 1965). “A Simplex Method for Function Minimization”. In: *The Computer Journal* 7.4, pp. 308–313. ISSN: 0010-4620. DOI: 10.1093/comjnl/7.4.308. URL: https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/7.4.308.

Pal, A. et al. (2024). *NonlinearSolve.Jl: High-Performance and Robust Solvers for Systems of Nonlinear Equations in Julia*. DOI: 10.48550/arXiv.2403.16341.

Providas, E. and M.A. Kattis (Nov. 2002). “Finite Element Method in Plane Cosserat Elasticity”. In: *Computers & Structures* 80.27, pp. 2059–2069. ISSN: 0045-7949. DOI: 10.1016/S0045-7949(02)00262-6.

Revels, J., M. Lubin and T. Papamarkou (July 2016). *Forward-Mode Automatic Differentiation in Julia*. DOI: 10.48550/arXiv.1607.07892.

Skowron, J. and A. Gould (Mar. 2012). *General Complex Polynomial Root Solver and Its Further Optimization for Binary Microlenses*. DOI: 10.48550/arXiv.1203.1034.

Stathas, A. and I. Stefanou (June 2023). “Fault Friction under Thermal Pressurization during Large Seismic-Slip: Numerical Analyses and Extension of the Model of Frictional Slip”. In: *International Journal of Mechanical Sciences* 248, p. 108184. ISSN: 0020-7403. DOI: 10.1016/j.ijmecsci.2023.108184.

[SW Rel.] Stefanou, I. and A. Stathas, *Numerical Geolab* version 1.0.1, 12th Dec. 2023. IMSIA (UMR 9219), CNRS, EDF, CEA, ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France and Institut für Konstruktiver Ingenieurbau, Universität für Bodenkultur Wien, Vienna, Austria. LIC: GPL-3.0-only. VCS: https://github.com/AlexSTA1993/numerical_geolab, SWHID: ⟨swh:1:dir:0cdf8bb9499f9a10d4458df650f4c2f2bb3640be;origin=https://github.com/AlexSTA1993/numerical_geolab;visit=swh:1:snp:98f1a88c8a6128a66fd89cbb8b68ec02840e8dc1;anchor=swh:1:rev:ecad099c55f25d5e976c1be5d4a7331e61c859f0⟩.

Vardoulakis, I. and J. Sulem (1995). *Bifurcation Analysis in Geomechanics*. Glasgow: Blackie Academic & Professional. ISBN: 0-7514-0214-1. DOI: 10.1201/9781482269383.