Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 414 (2023) 116101
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Multifield finite strain plasticity: Theory and numerics

Karol Lewandowski[a], Daniele Barbera[a], Paul Blackwell[c,d], Amir H. Roohi[c], Ignatios Athanasiadis[a], Andrew McBride[a,*], Paul Steinmann[a,b], Chris Pearce[a], Łukasz Kaczmarczyk[a]

[a] *Glasgow Computational Engineering Centre, James Watt School of Engineering, University of Glasgow, Glasgow, G12 8QQ, United Kingdom*
[b] *Institute of Applied Mechanics, Universität Erlangen–Nürnberg, Erlangen, D-91052, Germany*
[c] *Department of Design, Manufacturing & Engineering Management, University of Strathclyde, Glasgow, G12 1XJ, United Kingdom*
[d] *Advanced Forming Research Centre, Inchinnan, Glasgow, PA4 9LJ, United Kingdom*

Received 19 January 2023; received in revised form 1 April 2023; accepted 28 April 2023
Available online xxxx

Abstract

Motivated by the inability of classical computational plasticity to fully exploit modern scientific computing, a multifield formulation for finite strain plasticity is presented. This avoids a local integration of the elastoplastic model. In the multifield approach, the balance of linear momentum, the flow relation and the Karush–Kuhn–Tucker constraints are collectively cast in a variational format. In addition to the deformation, both the plastic strain and the consistency parameter are global degrees of freedom in the resulting spatially discrete problem. The ensuing proliferation of global degrees of freedom in the multifield approach is addressed by exploiting the block sparse structure of the algebraic system together with a tailored block matrix solver which can utilise emerging hardware architectures. A series of numerical problems demonstrate the validity, capability and efficiency of the proposed approach.

Keywords: Plasticity; Finite element method (FEM); Finite deformation; Solid mechanics

1. Introduction

Models and algorithms for elastoplasticity at finite deformations are well established [see e.g., 1,2, and the references therein]. They typically entail a *local* integration of the elastoplastic model within a strain-driven format wherein the deformation is treated as a field variable at the global level. The integration process is referred to as a *return mapping algorithm* and the local points at which it is performed are typically the quadrature points of a finite element. The local plastic state is then characterised by a set of internal variables associated with each quadrature point. These features characterise classical computational plasticity.

The classical approach to computational plasticity was devised to be efficient on central processing unit (CPU) based computing architectures where floating-point operations were the primary bottleneck. By contrast, modern computing architectures, based on, e.g., graphical processing units (GPUs), can only be effectively exploited

* Corresponding author.
E-mail address: Andrew.McBride@glasgow.ac.uk (A. McBride).

https://doi.org/10.1016/j.cma.2023.116101

by algorithms that minimise memory access often at the expense of floating-point operations. This requires a fundamental change in algorithm design philosophy and provides the primary motivation for matrix-free solvers where the storage of large global sparse matrices is circumvented and the underlying weak forms computed in place [see e.g., 3–5]. Matrix-free methods are not widely adopted in solid mechanics; a recent exception is a work by Davydov et al. [6] on nonlinear elasticity. This is due to the complex nonlinearities (material and geometric) and the widespread use of low-order finite element approximations, especially in commercial codes. Furthermore, the evolution of quantities characterising inelastic processes – including classical plasticity – is nearly always approximated at the local level of the quadrature point thereby inhibiting the use of matrix-free approaches. The conventional framework for plasticity needs to be reconsidered for large-scale computations on modern, energy-efficient architectures.

A key contribution of this work is to introduce a novel multifield formulation for finite strain plasticity well suited for matrix-free approaches. To this end, the balance of linear momentum, the flow relation, and the Karush–Kuhn–Tucker constraints are collectively cast in a variational format. Thus, in addition to the deformation, the plastic strain tensor and the consistency parameter are global degrees of freedom in the resulting spatially discrete problem that follows from a finite element discretisation. We adopt a finite strain plasticity formulation set in logarithmic strain space that mimics the structure of the small strain setting [7–12]. The Karush–Kuhn–Tucker inequality constraints on the evolution of plastic flow are recast as a variant of the nonlinear Fischer–Burmeister complementarity function; for applications of this approach to various formulations of plasticity, see [13–15]. This circumvents the need for an active set search at the global level [see e.g., 16,17, in the context of gradient crystal plasticity]. Attention is restricted to implicit time integration schemes. The multifield approach results in a proliferation of global degrees of freedom. This is addressed by exploiting the block sparse structure of the algebraic system together with a tailored block matrix solver.

Multifield algorithms for classical local plasticity are relatively few. A three-field variational formulation, which expressed the momentum balance equation, the plastic consistency condition, and the dilatational constitutive equation in weak form was proposed by Pinsky [18]. Mixed formulations of plasticity – wherein the stress is treated as an independent field variable – can also give rise to algorithmic structures that do not enforce the flow rule at the quadrature point level [see e.g., 19]. A multifield formulation of small strain plasticity was developed by Schröder and Kuhl [20,21] following a variational approach wherein the primary fields were the deformation, plastic strain, stress and the consistency parameter. The deformation was treated as a continuous field while the inelastic fields were continuous only at the element level. The ensuing system of coupled nonlinear equations was solved using the finite element method combined with an active set strategy which required the local assessment of the consistency condition. Static condensation of the discontinuous inelastic fields was used to reduce the overall size of the linear system [21].

Multifield-type algorithms are often adopted for the solution of extended plasticity formulations used to describe the size-dependent mechanical response of crystalline materials at the micro-scale [see e.g., 22–24, among others] via the inclusion of a plastic strain gradient measure and a length scale [25–36]. The inclusion of the gradient of a plastic strain measure dictates that the resulting approximation be continuous and hence a global approximation thereof is convenient [see e.g., 16,17,37–47]. However, the continuity requirement is not applicable to multifield formulations of classical plasticity as these necessitate only a dependence on a measure of the plastic strain and not the gradient thereof.

A key contribution of the work presented here is to develop a robust and efficient computational scheme for multifield plasticity at finite strains. The proposed scheme does not require a local return mapping algorithm thereby reducing memory storage requirements. It is therefore appropriate for matrix-free approaches on modern computing architectures generally limited by memory bandwidth. Furthermore, by avoiding internal variables and treating all unknowns as field variables, the scheme is well suited for the development and assessment of *a posteriori* error indicators to guide mesh adaptivity. In addition, the rates of plastic quantities are explicitly calculated, allowing for the straightforward calculation of material time derivatives and Arbitrary Lagrangian–Eulerian formulations.

The structure of the presentation is as follows. Standard notation is briefly summarised. Thereafter we recap the key kinematic relations for an elastoplastic body subject to finite deformations in Section 2. The various relevant stress measures, constitutive and flow relations are presented in Section 3 together with details of the variant of the Fischer—Burmeister complementarity function. The governing equations are then stated in strong and variational form in Section 4. The discretisation of the variational form of the governing equations in both time and space, together with details of the Newton–Raphson solution scheme are provided in Section 5. Section 6 provides a series

of numerical examples that serve to validate the formulation and demonstrate scalability. The presentation concludes with a summary and discussion.

Nomenclature

Direct notation is adopted throughout. Occasional use is made of index notation, the summation convention for repeated indices being implied.

The scalar products of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, and two second-order tensors $\boldsymbol{A}$ and $\boldsymbol{B}$ are respectively denoted by

$$\boldsymbol{a}\cdot\boldsymbol{b} = a_i b_i \qquad \text{and} \qquad \boldsymbol{A}:\boldsymbol{B} = A_{ij}B_{ij}.$$

The conventional dyadic product of two vectors, and of two second-order tensors, are respectively given by

$$\boldsymbol{a}\otimes\boldsymbol{b} = a_i b_j \boldsymbol{e}_i\otimes\boldsymbol{e}_j \qquad \text{and} \qquad \boldsymbol{A}\otimes\boldsymbol{B} = A_{ij}B_{kl}\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l,$$

where $\boldsymbol{e}_i$ are the basis vectors of the Cartesian coordinate frame. The upper and lower dyadic products of pairs of second-order tensors are respectively given by

$$\boldsymbol{A}\overline{\otimes}\boldsymbol{B} = A_{ik}B_{jl}\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l \qquad \text{and} \qquad \boldsymbol{A}\underline{\otimes}\boldsymbol{B} = A_{il}B_{jk}\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l.$$

The second-order identity tensor is defined by

$$\boldsymbol{I} = \delta_{ij}\boldsymbol{e}_i\otimes\boldsymbol{e}_j.$$

The various fourth-order identity tensors required are defined by

$$\begin{aligned}
\mathbb{I} &:= \boldsymbol{I}\overline{\otimes}\boldsymbol{I} = \delta_{ik}\delta_{jl}\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l,\\
\bar{\mathbb{I}} &:= \boldsymbol{I}\underline{\otimes}\boldsymbol{I} = \delta_{il}\delta_{jk}\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l,\\
\mathbb{I}^{\mathrm{vol}} &:= \frac{1}{3}\delta_{ij}\delta_{kl}\boldsymbol{e}_i\otimes\boldsymbol{e}_j\otimes\boldsymbol{e}_k\otimes\boldsymbol{e}_l,\\
\mathbb{I}^{\mathrm{sym}} &:= \frac{1}{2}\left[\mathbb{I}+\bar{\mathbb{I}}\right],\\
\mathbb{I}^{\mathrm{dev}} &:= \mathbb{I}^{\mathrm{sym}} - \mathbb{I}^{\mathrm{vol}}.
\end{aligned}$$

2. Finite strain kinematics

The kinematics of an elastoplastic body undergoing finite deformations are briefly summarised. The presentation of the additive plasticity model in logarithmic strain space follows that of Miehe et al. [7]. For further information on nonlinear continuum mechanics, refer to [1], among others.

Consider a continuum body composed of matter. The deformation of the body from its reference configuration $\Omega\subset\mathbb{R}^3$ to its current configuration Ω_t at time t is defined via the deformation map $\boldsymbol{\varphi}:\Omega\to\mathbb{R}^3$ as $\boldsymbol{x}=\boldsymbol{\varphi}(\boldsymbol{X},t)$, where $\boldsymbol{x}\in\Omega_t$ and $\boldsymbol{X}\in\Omega$ are physical points in the current and reference configurations, respectively. The boundary of the reference configuration is denoted by Γ with outward unit normal $\boldsymbol{N}$.

The invertible linear tangent map $\boldsymbol{F}:\Omega\to\mathbb{GL}_+(3)$[1] (i.e., the deformation gradient) maps a line element $\mathsf{d}\boldsymbol{X}$ in the reference configuration to a line element $\mathsf{d}\boldsymbol{x}$ in the current configuration and is defined by the derivative of the deformation with respect to the material placement; that is,

$$\boldsymbol{F} := \mathrm{Grad}\boldsymbol{\varphi} = \boldsymbol{R}\cdot\boldsymbol{U} \qquad \text{with} \qquad \mathsf{d}\boldsymbol{x} = \boldsymbol{F}\cdot\mathsf{d}\boldsymbol{X},$$

[1] $\mathbb{GL}_+(3)$ is the set of positive definite 2nd order tensors in $\mathbb{R}^3$.

where $\boldsymbol{R} \in SO(3)$[2] is the rotation tensor and $\boldsymbol{U} \in \mathbb{S}_+$ is the right stretch tensor. The determinant of $\boldsymbol{F}$ is defined by $J \coloneqq \det \boldsymbol{F} > 0$. The symmetric right Cauchy–Green tensor $\boldsymbol{C} : \Omega \to \mathbb{S}_+$[3] is defined by

$$\boldsymbol{C} \coloneqq \boldsymbol{F}^{\mathsf{T}} \cdot \boldsymbol{F} = \boldsymbol{U} \cdot \boldsymbol{U} = \sum_{A=1}^{3} \lambda_A^2 \boldsymbol{N}^A \otimes \boldsymbol{N}^A,$$

where λ_A^2 and $\boldsymbol{N}^A$ are the eigenvalues, i.e., the squared principal stretches, and eigenvectors of $\boldsymbol{U}$, respectively, following from the spectral decomposition. The Hencky or logarithmic strain is defined by

$$\boldsymbol{E} \coloneqq \frac{1}{2} \ln\big[\boldsymbol{C}\big] = \sum_{A=1}^{3} \ln[\lambda_A] \boldsymbol{N}^A \otimes \boldsymbol{N}^A = \ln\big[\boldsymbol{U}\big].$$

The plastic logarithmic strain is denoted by $\boldsymbol{E}^{\mathrm{p}}$ with the elastic logarithmic strain then defined in an additive format that mimics the infinitesimal deformation setting by

$$\boldsymbol{E}^{\mathrm{e}} = \boldsymbol{E} - \boldsymbol{E}^{\mathrm{p}},$$

where the plastic incompressibility constraint follows as

$$\mathrm{tr}\big[\boldsymbol{E}^{\mathrm{p}}\big] = 0. \tag{1}$$

3. Stress measures and constitutive relations

The relevant stress measures are recalled, and the stress power relation is then used to define the Hencky stress. The constitutive relations for the kinetic measures follow from a Coleman–Noll procedure. The plastic flow relations are then stated and the Karush–Kuhn–Tucker constraints recast in the form of a nonlinear relation which is central to the multifield plasticity formulation.

3.1. Stress measures

The stress power relation identifies the stress quantity conjugate to the rate of change (i.e., the material time derivative) of the total logarithmic strain $\dot{\boldsymbol{E}}$, denoted by $\boldsymbol{T}$ and termed the Hencky stress, as follows

$$\boldsymbol{P} : \dot{\boldsymbol{F}} = \frac{1}{2} \boldsymbol{S} : \dot{\boldsymbol{C}} = \boldsymbol{T} : \dot{\boldsymbol{E}},$$

where $\boldsymbol{P}$ and $\boldsymbol{S}$ are the familiar Piola and Piola–Kirchhoff stress tensors, respectively. The key relations between the various stress and strain measures are given by

$$\begin{aligned} \boldsymbol{P} &= \boldsymbol{T} : \mathbb{P} \qquad \text{with} \qquad \mathbb{P} = \partial_{\boldsymbol{F}} \boldsymbol{E}, \\ \boldsymbol{S} &= \boldsymbol{T} : \mathbb{P}_{\mathrm{L}} \qquad \text{with} \qquad \mathbb{P}_{\mathrm{L}} = 2 \partial_{\boldsymbol{C}} \boldsymbol{E}. \end{aligned}$$

3.2. Constitutive and flow relations

Consider a stored energy function ψ of the form

$$\psi = \psi(\boldsymbol{E} - \boldsymbol{E}^{\mathrm{p}}, \boldsymbol{A}, \alpha)$$

where $\boldsymbol{A} \in \mathbb{S}$ is a symmetric second-order tensor that models kinematic hardening and α a scalar variable describing isotropic hardening. Following a now standard Coleman–Noll procedure [48], the conjugate kinetic measures are obtained as

$$\boldsymbol{T} = \partial_{\boldsymbol{E}^{\mathrm{e}}} \psi = -\partial_{\boldsymbol{E}^{\mathrm{p}}} \psi, \qquad \boldsymbol{B} = -\partial_{\boldsymbol{A}} \psi, \qquad \beta = -\partial_{\alpha} \psi. \tag{2}$$

[2] $SO(3)$ denotes the special orthogonal group of rotations in $\mathbb{R}^3$.
[3] $\mathbb{S}_+$ denotes the group of symmetric positive definite tensors.

Consider now an additive decomposition of the stored energy function ψ into elastic and plastic parts, given by

$$\psi = \psi^{\mathrm{e}}(\boldsymbol{E} - \boldsymbol{E}^{\mathrm{p}}) + \underbrace{\psi^{\mathrm{k}}(\boldsymbol{A}) + \psi^{\mathrm{i}}(\alpha)}_{\psi^{\mathrm{p}}}.$$

The plastic part ψ^{p} is composed of contributions from kinematic and isotropic hardening, denoted by ψ^{k} and ψ^{i}, respectively. More concretely, a quadratic form of the stored energy function is chosen, given by

$$\psi = \underbrace{\frac{1}{2}\|\boldsymbol{E} - \boldsymbol{E}^{\mathrm{p}}\|^2_{\mathbb{D}}}_{\psi^{\mathrm{e}}} + \underbrace{\frac{K}{2}\|\boldsymbol{A}\|^2}_{\psi^{\mathrm{k}}} + \underbrace{\frac{H}{2}\alpha^2}_{\psi^{\mathrm{i}}}$$

where

$$\|\boldsymbol{E}^{\mathrm{e}}\|_{\mathbb{D}} := \sqrt{\boldsymbol{E}^{\mathrm{e}} : \mathbb{D} : \boldsymbol{E}^{\mathrm{e}}}.$$

$\mathbb{D}$ is the fourth-order tensor of elastic moduli,

$$\mathbb{D} = 2\mu\mathbb{I}^{\mathrm{sym}} + 3\lambda\mathbb{I}^{\mathrm{vol}} = \underbrace{2\mu\mathbb{I}^{\mathrm{dev}}}_{\mathbb{D}^{\mu}} + 3\kappa\mathbb{I}^{\mathrm{vol}},$$

with λ and μ the Lamé parameters related to the Young's modulus E, Poisson's ratio ν and bulk modulus κ as follows

$$\lambda = \frac{\nu E}{[1+\nu][1-2\nu]}, \qquad \mu = \frac{E}{2[1+\nu]}, \qquad \kappa = \lambda + \frac{2\mu}{3}.$$

The fourth-order symmetric, deviatoric and volumetric identity tensors, $\mathbb{I}^{\mathrm{sym}}$, $\mathbb{I}^{\mathrm{dev}}$ and $\mathbb{I}^{\mathrm{vol}}$, respectively, were defined in the Nomenclature. The isotropic and kinematic hardening moduli are denoted by H and K, respectively. Hence, from Eq. (2) one obtains the relations for the various kinetic measures as

$$\boldsymbol{T} = \mathbb{D} : \left[\boldsymbol{E} - \boldsymbol{E}^{\mathrm{p}}\right], \qquad \boldsymbol{B} = -K\boldsymbol{A}, \qquad \beta = -H\alpha. \tag{3}$$

The convex yield function f defines the elastic domain $\mathcal{E}$ and the yield surface and thereby restricts the admissible stress and kinetic fields as follows

$$f(\boldsymbol{T}, \boldsymbol{B}, \beta) \leq 0$$

where

$$\begin{cases} f < 0, & (\boldsymbol{T}, \boldsymbol{B}, \beta) \text{ in the elastic domain (set)} \\ f = 0, & (\boldsymbol{T}, \boldsymbol{B}, \beta) \text{ on the yield surface (plastic set)} \end{cases}.$$

The plastic flow relations follow from the postulate of maximum plastic dissipation as

$$\dot{\boldsymbol{E}}^{\mathrm{p}} = \dot{\lambda}\partial_{\boldsymbol{T}} f = \dot{\lambda}\boldsymbol{N}^{\mathrm{p}}, \tag{4}$$

$$\dot{\boldsymbol{A}} = \dot{\lambda}\partial_{\boldsymbol{B}} f, \tag{5}$$

$$\dot{\alpha} = \dot{\lambda}\partial_{\beta} f, \tag{6}$$

where $\dot{\lambda}$ is the consistency parameter and the second-order tensor $\boldsymbol{N}^{\mathrm{p}}$ is the flow direction (i.e., the normal to the yield function). For the rate-independent case considered here, the plastic flow relations are subject to the Karush–Kuhn–Tucker constraints given by

$$\dot{\lambda} \geq 0, \qquad f \leq 0, \qquad \dot{\lambda} f = 0. \tag{7}$$

The inequality constraints (7) can be expressed in the form of a variant of the Fischer—Burmeister complementarity function $c(\dot{\lambda}, f)$ [see e.g., 14], where

$$c(\dot{\lambda}, f) = c_{\lambda}\dot{\lambda} - f - \left|c_{\lambda}\dot{\lambda} + f\right|. \tag{8}$$

Note, the parameter $c_{\lambda} > 0$ has units of stress times time. The Fischer–Burmeister complementarity function is piecewise smooth but non-differentiable at the line $\mathcal{Z} = \{(\dot{\lambda}, f) : c_{\lambda}\dot{\lambda} + f = 0\}$, as indicated in Fig. 1. The point

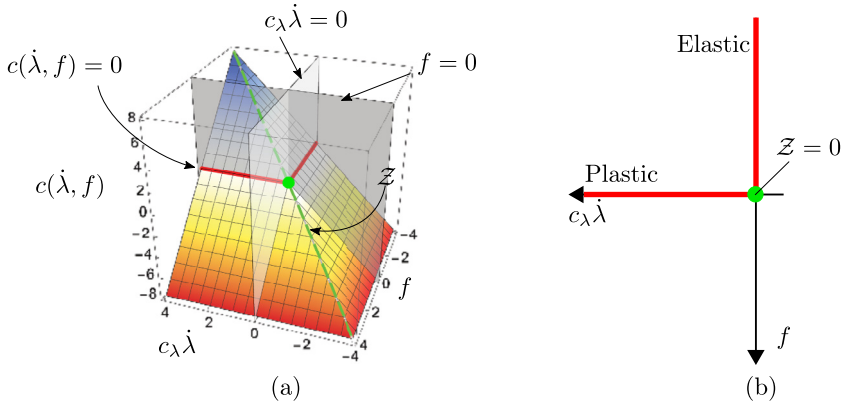

Fig. 1. In (a), a schematic of the Fischer–Burmeister complementarity function $c(\dot\lambda, f)$ indicating the elastic set ($c_\lambda\dot\lambda = 0$) and the plastic set ($f = 0$), together with the line $\mathcal{Z}$ dividing them. The point $\mathcal{Z} = 0$ corresponds to the particular case of neutral loading. The view in (b) is the plane at $c(\dot\lambda, f) = 0$.

$\mathcal{Z} = 0$ divides the space spanned by the independent variables of c into two sets: a plastic (active) set where $f = 0$ and an elastic (inactive) set where $\dot\lambda = 0$.

Remark 1. The non-differentiability of the Fischer–Burmeister complementarity function can be addressed via regularisation of the function in the vicinity of $\mathcal{Z}$ [14]. The alternative strategy adopted here is to enhance the Newton–Raphson solver with a line-search method in order to attain quasi-quadratic convergence rates as discussed in Section 5.3 and demonstrated in Section 6.2.

The role of the stabilisation parameter c_λ is to improve the numerical robustness. Its inclusion does not affect the rate of convergence or the converged solution. In general, c_λ should be chosen to avoid large truncation and other floating-point precision errors. Alternatively, as discussed in Section 5.5, a rescaling of the governing equations allows c_λ to be chosen as unity. □

Remark 2. We consider here an extension of the isotropic Mises–Huber yield condition to accommodate Prager-type kinematic hardening where the yield function is given by

$$f(\boldsymbol{T}, \boldsymbol{B}, \beta) = \|\boldsymbol{T} + \boldsymbol{B}\|_{\mathbb{I}^{\mathrm{dev}}} - \sqrt{\frac{2}{3}}\big[\sigma_y + \beta\big], \tag{9}$$

where $\boldsymbol{B}$ is the back stress, and $\|\bullet\|_{\mathbb{I}^{\mathrm{dev}}} := \sqrt{\bullet : \mathbb{I}^{\mathrm{dev}} : \bullet}$. The plastic flow relations (4)–(6) thus take the form

$$\dot{\boldsymbol{E}}^{\mathrm{p}} = \dot\lambda \frac{\mathbb{I}^{\mathrm{dev}} : [\boldsymbol{T} + \boldsymbol{B}]}{\|\boldsymbol{T} + \boldsymbol{B}\|_{\mathbb{I}^{\mathrm{dev}}}}, \tag{10}$$

$$\dot{\boldsymbol{A}} = \dot{\boldsymbol{E}}^{\mathrm{p}}, \tag{11}$$

$$\dot\alpha = \dot\lambda\sqrt{\frac{2}{3}} = \sqrt{\frac{2}{3}}\|\dot{\boldsymbol{E}}^{\mathrm{p}}\|. \quad \square \tag{12}$$

Remark 3. From the properties of the projection operator, the deviatoric projection of the Hencky stress term in the plastic flow relation (10) can be expressed as

$$\mathbb{I}^{\mathrm{dev}} : \underbrace{\mathbb{D} : \boldsymbol{E}^{\mathrm{e}}}_{\boldsymbol{T}} = \mathbb{I}^{\mathrm{dev}} : \underbrace{\mathbb{D}^{\mu} : \boldsymbol{E}^{\mathrm{e}}}_{\boldsymbol{T}^{\mu}}. \quad \square \tag{13}$$

4. The governing equations

The set of governing equations – the equilibrium equation, flow relation (4) and complementarity condition (8) – are now stated in strong form. The point of departure for the novel multifield formulation proposed here is then to

recast these equations into a variational format following a Galerkin approach. Recall that the classical approach to computational plasticity typically entails a local integration of the elastoplastic model within a strain-driven format wherein only the deformation is treated as a field variable at the global level. In the multifield approach, in addition to the deformation, the plastic strain tensor, and the consistency parameter are global fields whose solution satisfies the corresponding variational formulation.

4.1. Strong form

The strong form of the equilibrium problem is given by the (quasi-static) balance of linear momentum in the absence of inertial forces and the set of Dirichlet and Neumann boundary conditions as

$$\begin{aligned} \mathrm{Div}\boldsymbol{P} + \boldsymbol{b} &= \boldsymbol{0} \quad \text{in } \Omega, \\ \boldsymbol{u} &= \overline{\boldsymbol{u}} \quad \text{on } \Gamma_{\boldsymbol{u}}, \\ \boldsymbol{P}\cdot\boldsymbol{N} &= \overline{\boldsymbol{t}} \quad \text{on } \Gamma_{\boldsymbol{t}}. \end{aligned} \tag{14}$$

Here $\boldsymbol{b}$ is the body force per unit of reference volume and the boundary of the domain is decomposed into non-overlapping portions $\Gamma = \Gamma_{\boldsymbol{u}} \cup \Gamma_{\boldsymbol{t}}$. These are augmented by the flow relation (4) and the complementarity condition (8) as summarised in Box 4.1.1.

Box. 4.1.1: Governing equations

$$\mathrm{Div}\boldsymbol{P} + \boldsymbol{b} = \boldsymbol{0}\,, \tag{15a}$$

$$\dot{\boldsymbol{E}}^{\mathrm{p}} - \dot{\lambda}\boldsymbol{N}^{\mathrm{p}} = \boldsymbol{0}\,, \tag{15b}$$

$$c(\dot{\lambda}, f) = 0\,. \tag{15c}$$

Remark 4. For the case of Prager-type kinematic hardening as introduced in Eq. (9), no additional governing equation is required for the evolution of the back stress $\boldsymbol{B}$. More complex kinematic hardening models involving, for example, two or more back stresses [49], would require additional governing equations. □

Remark 5. The elastoplasticity problem cannot be solved using the balance of linear momentum (15a) and either the flow relation (15b) or the complementarity function (15c). All three relations are required. The complementarity function (15c) is a scalar equation and hence knowledge of $\|\dot{\boldsymbol{E}}^{\mathrm{p}}\|$ is not sufficient to uniquely characterise $\dot{\boldsymbol{E}}^{\mathrm{p}}$. A tensorial relationship is required and is provided by the flow relation (15b). Similarly, the flow relation requires knowledge of the plastic state, which is provided by the complementarity function (15c). Hence, both the plastic multiplier field and the logarithmic plastic strain field are approximated and discretised, as presented in Sections 5 and 4.2. □

4.2. Variational formulation

The variational form of the governing equations presented in Box 4.1.1 follows from a Bubnov–Galerkin procedure and results in three coupled nonlinear equations as stated in Box 4.2.1. Here $\delta\boldsymbol{u}$, $\delta\boldsymbol{E}^{\mathrm{p}}$ and $\delta\lambda$ are admissible variations of the deformation, plastic strain and the consistency parameter, respectively. $\mathbb{D}$ was introduced in (17) for dimensional scaling. Note that one only requires the components of the variations in the plastic strain and the consistency parameter to be square integrable.

Box. 4.2.1: Variational formulation

Find $\boldsymbol{u} \in [H_0^1(\Omega)]^3$, $\boldsymbol{E}^{\mathrm{p}} \in [L^2(\Omega)]^{3\times 3}$, $\lambda \in L^2(\Omega)$ such that:

$$\int_\Omega \delta\boldsymbol{F} : \boldsymbol{P}\,\mathrm{d}V - \int_\Omega \delta\boldsymbol{u}\cdot\boldsymbol{b}\,\mathrm{d}V - \int_{\Gamma_t} \delta\boldsymbol{u}\cdot\bar{\boldsymbol{t}}\,\mathrm{d}A = 0 \qquad \forall \delta\boldsymbol{u} \in [H_0^1(\Omega)]^3\,, \tag{16}$$

$$\int_\Omega \delta\boldsymbol{E}^{\mathrm{p}} : \mathbb{D} : \left[\dot{\boldsymbol{E}}^{\mathrm{p}} - \dot{\lambda}\boldsymbol{N}^{\mathrm{p}}\right] = 0 \qquad \forall \delta\boldsymbol{E}^{\mathrm{p}} \in [L^2(\Omega)]^{3\times 3}\,, \tag{17}$$

$$\int_\Omega \delta\lambda\left[c_\lambda\dot{\lambda} - f - \left|c_\lambda\dot{\lambda} + f\right|\right] = 0 \qquad \forall \delta\lambda \in L^2(\Omega)\,. \tag{18}$$

5. Discrete problem

The weak form of the problem is now discretised in both time and space. A general implicit time integration procedure is employed first to obtain a semi-discrete system of equations. The finite element method is then used to discretise the problem in space. Details of the Newton–Raphson strategy employed to approximately solve the fully discrete problem are given.

5.1. Time discrete problem

Let $[0, T] \subset \mathcal{R}_+$ be the time interval of interest. The time interval is partitioned into steps such that

$$[0, T] = \bigcup_{n\in\mathcal{I}} [t_n, t_{n+1}] \qquad \text{where} \qquad t_{n+1} = t_n + \Delta t\,.$$

We consider here implicit time-integration schemes. The approximation of the time derivative of a quantity $(\dot{\bullet})$ is denoted by $(\mathring{\bullet})$. The time-discrete form of the governing Eqs. (16)–(18) at t_{n+1} is thus given by

$$R_{\boldsymbol{u}} := \int_\Omega \delta\boldsymbol{F} : \boldsymbol{P}\,\mathrm{d}V - \int_\Omega \delta\boldsymbol{u}\cdot\boldsymbol{b}\,\mathrm{d}V - \int_{\Gamma_t} \delta\boldsymbol{u}\cdot\bar{\boldsymbol{t}}\,\mathrm{d}A = 0, \tag{19}$$

$$R_{\boldsymbol{E}^{\mathrm{p}}} := \int_\Omega \delta\boldsymbol{E}^{\mathrm{p}} : \mathbb{D} : \left[\mathring{\boldsymbol{E}}^{\mathrm{p}} - \mathring{\lambda}\boldsymbol{N}^{\mathrm{p}}\right] = 0, \tag{20}$$

$$R_\lambda := \int_\Omega \delta\lambda\left[c_\lambda\mathring{\lambda} - f - \left|c_\lambda\mathring{\lambda} + f\right|\right] = 0. \tag{21}$$

The various quantities are evaluated at t_{n+1} although this is not indicated here and henceforth for the sake of brevity. Note that $\boldsymbol{P} = \boldsymbol{P}(\boldsymbol{E}, \boldsymbol{E}^{\mathrm{p}})$, $\boldsymbol{N}^{\mathrm{p}} = \boldsymbol{N}^{\mathrm{p}}(\boldsymbol{E}, \boldsymbol{E}^{\mathrm{p}})$ and $f = f(\boldsymbol{E}, \boldsymbol{E}^{\mathrm{p}}, \lambda)$ and $\boldsymbol{E}^{\mathrm{p}}$ is related to λ and $\boldsymbol{N}^{\mathrm{p}}$ in a weak sense.

5.2. Fully discrete problem

The triangulation of the reference configuration Ω into non-overlapping tetrahedral elements is denoted by $\mathcal{T}_\Omega^h$. The displacement field $\boldsymbol{u}$ is approximated using a finite element space of globally continuous polynomials of degree p. The fields for the plastic strain and the consistency parameter are both approximated using a finite element space of element-wise continuous polynomials of degree $p-1$.

The macroscopic motion $\boldsymbol{u} \in [H^1(\Omega)]^3$, the plastic strain $\boldsymbol{E}^{\mathrm{p}} \in \{E_{ij}^{\mathrm{p}} \in L^2(\Omega) : E_{ij}^{\mathrm{p}} = E_{ji}^{\mathrm{p}}\}$, and the consistency parameter $\lambda \in L^2(\Omega)$ are respectively given in a vector space spanned by vector-, tensor-, and scalar-valued finite element basis functions (polynomials with local support), respectively denoted by $\boldsymbol{N}_{\boldsymbol{u}}$, $\boldsymbol{N}_{\boldsymbol{E}^{\mathrm{p}}}$ and N_λ. That is, the primary fields and their associated rates and variations are approximated by

$$\boldsymbol{u}^h =: \sum_{I\in\mathcal{I}_{\boldsymbol{u}}} \mathsf{u}_I \boldsymbol{N}_{\boldsymbol{u}}^I(\boldsymbol{X}), \qquad \delta\boldsymbol{u}^h := \sum_{I\in\mathcal{I}_{\boldsymbol{u}}} \delta\mathsf{u}_I \boldsymbol{N}_{\boldsymbol{u}}^I(\boldsymbol{X}), \tag{22}$$

$$\boldsymbol{E}^{\mathrm{p},h} := \sum_{I\in\mathcal{I}_{\boldsymbol{E}^{\mathrm{p}}}} \mathsf{E}^{\mathrm{p}}{}_I \boldsymbol{N}_{\boldsymbol{E}^{\mathrm{p}}}^I(\boldsymbol{X}), \qquad \mathring{\boldsymbol{E}}^{\mathrm{p},h} := \sum_{I\in\mathcal{I}_{\boldsymbol{E}^{\mathrm{p}}}} \mathring{\mathsf{E}}^{\mathrm{p}}_I \boldsymbol{N}_{\boldsymbol{E}^{\mathrm{p}}}^I(\boldsymbol{X}), \qquad \delta\boldsymbol{E}^{\mathrm{p},h} := \sum_{I\in\mathcal{I}_{\boldsymbol{E}^{\mathrm{p}}}} \delta\mathsf{E}^{\mathrm{p}}{}_I \boldsymbol{N}_{\boldsymbol{E}^{\mathrm{p}}}^I(\boldsymbol{X}), \tag{23}$$

$$\lambda^h := \sum_{I\in\mathcal{I}_\lambda} \lambda_I N^I_\lambda(\boldsymbol{X}), \qquad \mathring{\lambda}^h := \sum_{I\in\mathcal{I}_\lambda} \mathring{\lambda}_I N^I_\lambda(\boldsymbol{X}), \qquad \delta\lambda^h := \sum_{I\in\mathcal{I}_\lambda} \delta\lambda_I N^I_\lambda(\boldsymbol{X}), \tag{24}$$

where the superscript h indicates that the representation is related to the finite element mesh with size function $h(\boldsymbol{X})$. Upright Greek and Roman letters are used to denote a global vector containing the degrees of freedom associated with one of the three primary fields. The sets $\mathcal{I}_{\boldsymbol{u}}$ and $\mathcal{I}_{\boldsymbol{E}^{\mathrm{p}}}$ and $\mathcal{I}_\lambda$ contain the degrees of freedom for the displacement, plastic strain and consistency parameter fields, respectively. The discrete representation of the gradient of the displacement field and its variation follows directly as

$$\boldsymbol{F}^h = \sum_{I\in\mathcal{I}_{\boldsymbol{u}}} \mathsf{u}_I \mathrm{Grad}\boldsymbol{N}^I_{\boldsymbol{u}}(\boldsymbol{X}) \qquad \text{and} \qquad \delta\boldsymbol{F}^h = \sum_{I\in\mathcal{I}_{\boldsymbol{u}}} \delta\mathsf{u}_I \mathrm{Grad}\boldsymbol{N}^I_{\boldsymbol{u}}(\boldsymbol{X}).$$

5.3. Newton–Raphson strategy

Substituting the discrete representations (22)–(24) into the time-discrete form of the governing relations (19)–(21), and omitting the size function h for brevity, yields the following three sets of coupled non-linear residual equations:

$$R^I_{\boldsymbol{u}} = \int_\Omega \Big[\boldsymbol{P} : \mathrm{Grad}\boldsymbol{N}^I_{\boldsymbol{u}} - \boldsymbol{b}\cdot\boldsymbol{N}^I_{\boldsymbol{u}}\Big]\,\mathrm{d}V - \int_{\Gamma_t} \bar{\boldsymbol{t}}\cdot\boldsymbol{N}^I_{\boldsymbol{u}}\,\mathrm{d}A \qquad \forall I\in\mathcal{I}_{\boldsymbol{u}}, \tag{25}$$

$$R^I_{\boldsymbol{E}^{\mathrm{p}}} = \int_\Omega \big[\mathring{\boldsymbol{E}}^{\mathrm{p}} - \mathring{\lambda}\boldsymbol{N}^{\mathrm{p}}\big] : \mathbb{D} : \boldsymbol{N}^I_{\boldsymbol{E}^{\mathrm{p}}}\,\mathrm{d}V \qquad \forall I\in\mathcal{I}_{\boldsymbol{E}^{\mathrm{p}}}, \tag{26}$$

$$R^I_\lambda = \int_\Omega \big[c_\lambda\mathring{\lambda} - f - \big|c_\lambda\mathring{\lambda} + f\big|\big]N^I_\lambda\,\mathrm{d}V \qquad \forall I\in\mathcal{I}_\lambda. \tag{27}$$

The three global residual vectors, obtained by assembling the individual contributions from the residual expressions associated with the respective degrees of freedom, are denoted by

$$\mathbf{R} := [\mathbf{R}^{\boldsymbol{u}}\ \mathbf{R}^{\boldsymbol{E}^{\mathrm{p}}}\ \mathbf{R}^{\lambda}]^{\mathsf{T}},$$

and the global vector of degrees of freedom by

$$\mathbf{d} := [\mathbf{u}\ \mathbf{E}^{\mathrm{p}}\ \boldsymbol{\lambda}]^{\mathsf{T}}.$$

Note that $\dim\mathbf{R}_{\boldsymbol{u}} = \dim\mathbf{u} = |\mathcal{I}_{\boldsymbol{u}}|$, $\dim\mathbf{R}_{\boldsymbol{E}^{\mathrm{p}}} = \dim\mathbf{E}^{\mathrm{p}} = |\mathcal{I}_{\boldsymbol{E}^{\mathrm{p}}}|$, and $\dim\mathbf{R}_\lambda = \dim\boldsymbol{\lambda} = |\mathcal{I}_\lambda|$.

The coupled nonlinear residual Eqs. (25)–(27) are solved approximately using a Newton–Raphson strategy whereby within each iteration (i) of the current load (time) step the linearised problem is given by

$$\begin{aligned}\mathbf{0} \doteq \mathbf{R}^{(i+1)} &\approx \mathbf{R}^{(i)} + \left[\frac{\partial\mathbf{R}^{(i)}}{\partial\mathring{\mathbf{d}}}\frac{\partial\mathring{\mathbf{d}}}{\partial\mathbf{d}} + \frac{\partial\mathbf{R}^{(i)}}{\partial\mathbf{d}}\right]\Delta\mathbf{d}\\ &= \mathbf{R}^{(i)} + \underbrace{\left[\sigma\frac{\partial\mathbf{R}^{(i)}}{\partial\mathring{\mathbf{d}}} + \frac{\partial\mathbf{R}^{(i)}}{\partial\mathbf{d}}\right]}_{\mathbf{K}^{(i)}}\Delta\mathbf{d}\\ \implies \mathbf{K}^{(i)}\Delta\mathbf{d} &= -\mathbf{R}^{(i)},\end{aligned} \tag{28}$$

where the shift $\sigma := \partial\mathring{\bullet}/\partial\bullet$ depends on the ordinary differential equation integrator and time step duration Δt but not on the function being integrated [50]. This allows the time integration scheme to be chosen subsequent to and independent of the numerical implementation of the finite element problem. The tangent matrix is denoted by $\mathbf{K}$. The Newton–Raphson algorithm is enhanced using a line-search algorithm [50]. The decomposition of the

tangent matrix into time-dependent and time-independent parts allows for the straightforward application of a range of implicit time integration schemes. Thus,

$$\underbrace{\left[\sigma\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \partial\mathbf{R}_{\boldsymbol{E}^{\mathrm{p}}}/\partial\mathring{\mathbf{E}}^{\mathrm{p}} & \partial\mathbf{R}_{\boldsymbol{E}^{\mathrm{p}}}/\partial\mathring{\boldsymbol{\lambda}}\\ \mathbf{0} & \partial\mathbf{R}_{\lambda}/\partial\mathring{\mathbf{E}}^{\mathrm{p}} & \partial\mathbf{R}_{\lambda}/\partial\mathring{\boldsymbol{\lambda}}\end{bmatrix}^{(i)} + \begin{bmatrix}\partial\mathbf{R}_{\boldsymbol{u}}/\partial\mathbf{u} & \partial\mathbf{R}_{\boldsymbol{u}}/\partial\mathbf{E}^{\mathrm{p}} & \mathbf{0}\\ \partial\mathbf{R}_{\boldsymbol{E}^{\mathrm{p}}}/\partial\mathbf{u} & \mathbf{0} & \mathbf{0}\\ \partial\mathbf{R}_{\lambda}/\partial\mathbf{u} & \mathbf{0} & \mathbf{0}\end{bmatrix}^{(i)}\right]}_{\begin{bmatrix}\mathbf{K}_{\boldsymbol{uu}} & \mathbf{K}_{\boldsymbol{u}\boldsymbol{E}^{\mathrm{p}}} & \mathbf{0}\\ \mathbf{K}_{\boldsymbol{E}^{\mathrm{p}}\boldsymbol{u}} & \mathbf{K}_{\boldsymbol{E}^{\mathrm{p}}\boldsymbol{E}^{\mathrm{p}}} & \mathbf{K}_{\boldsymbol{E}^{\mathrm{p}}\lambda}\\ \mathbf{K}_{\lambda\boldsymbol{u}} & \mathbf{K}_{\lambda\boldsymbol{E}^{\mathrm{p}}} & \mathbf{K}_{\lambda\lambda}\end{bmatrix}^{(i)}}\begin{bmatrix}\Delta\mathbf{u}\\ \Delta\mathbf{E}^{\mathrm{p}}\\ \Delta\boldsymbol{\lambda}\end{bmatrix}^{(i)} = -\begin{bmatrix}\mathbf{R}_{\boldsymbol{u}}\\ \mathbf{R}_{\boldsymbol{E}^{\mathrm{p}}}\\ \mathbf{R}_{\lambda}\end{bmatrix}^{(i)}. \tag{29}$$

The various contributions to the tangent matrix $\mathbf{K}$ are given by

$$\big[\mathsf{K}_{\boldsymbol{uu}}\big]_{IJ} = \partial_{\mathsf{u}^J}\mathsf{R}^I_{\boldsymbol{u}} = \int_{\Omega}\partial_{\mathsf{u}^J}\big[\boldsymbol{P} : \mathrm{Grad}\boldsymbol{N}^I_{\boldsymbol{u}}\big]\,\mathrm{d}V = \int_{\Omega}\big[\partial_{\boldsymbol{F}}\boldsymbol{P} : \mathrm{Grad}\boldsymbol{N}^J_{\boldsymbol{u}}\big] : \mathrm{Grad}\boldsymbol{N}^I_{\boldsymbol{u}}\,\mathrm{d}V\,,$$

$$\big[\mathsf{K}_{\boldsymbol{u}\boldsymbol{E}^{\mathrm{p}}}\big]_{IJ} = \partial_{\mathsf{E}^{\mathrm{p}J}}\mathsf{R}^I_{\boldsymbol{u}} = \int_{\Omega}\partial_{\mathsf{E}^{\mathrm{p}J}}\big[\boldsymbol{P} : \mathrm{Grad}\boldsymbol{N}^I_{\boldsymbol{u}}\big]\,\mathrm{d}V = \int_{\Omega}\big[\partial_{\boldsymbol{E}^{\mathrm{p}}}\boldsymbol{P} : \mathrm{Grad}\boldsymbol{N}^J_{\boldsymbol{E}^{\mathrm{p}}}\big] : \mathrm{Grad}\boldsymbol{N}^I_{\boldsymbol{u}}\,\mathrm{d}V\,,$$

$$\big[\mathsf{K}_{\boldsymbol{E}^{\mathrm{p}}\boldsymbol{u}}\big]_{IJ} = \partial_{\mathsf{u}^J}\mathsf{R}^I_{\boldsymbol{E}^{\mathrm{p}}} = \int_{\Omega}\partial_{\mathsf{u}^J}\big[\mathbb{D} : \big[\mathring{\boldsymbol{E}}^{\mathrm{p}} - \mathring{\lambda}\boldsymbol{N}^{\mathrm{p}}\big] : \boldsymbol{N}^I_{\boldsymbol{E}^{\mathrm{p}}}\big]\,\mathrm{d}V = -\int_{\Omega}\big[\mathbb{D} : \mathring{\lambda}\partial_{\boldsymbol{F}}\boldsymbol{N}^{\mathrm{p}} : \mathrm{Grad}\boldsymbol{N}^J_{\boldsymbol{u}}\big] : \boldsymbol{N}^I_{\boldsymbol{E}^{\mathrm{p}}}\,\mathrm{d}V\,,$$

$$\begin{aligned}\big[\mathsf{K}_{\boldsymbol{E}^{\mathrm{p}}\boldsymbol{E}^{\mathrm{p}}}\big]_{IJ} &= \sigma\,\partial_{\hat{\mathsf{E}}^{\mathrm{p}J}}\mathsf{R}^I_{\boldsymbol{E}^{\mathrm{p}}} + \partial_{\mathsf{E}^{\mathrm{p}J}}\mathsf{R}^I_{\boldsymbol{E}^{\mathrm{p}}}\\ &= \sigma\int_{\Omega}\partial_{\hat{\mathsf{E}}^{\mathrm{p}J}}\big[\mathbb{D} : \big[\mathring{\boldsymbol{E}}^{\mathrm{p}} - \mathring{\lambda}\boldsymbol{N}^{\mathrm{p}}\big] : \boldsymbol{N}^I_{\boldsymbol{E}^{\mathrm{p}}}\big]\,\mathrm{d}V + \int_{\Omega}\partial^J_{\boldsymbol{E}^{\mathrm{p}}}\big[\mathbb{D} : \big[\mathring{\boldsymbol{E}}^{\mathrm{p}} - \mathring{\lambda}\boldsymbol{N}^{\mathrm{p}}\big] : \boldsymbol{N}^I_{\boldsymbol{E}^{\mathrm{p}}}\big]\,\mathrm{d}V\\ &= \sigma\int_{\Omega}\mathbb{D} : \big[\boldsymbol{N}^J_{\boldsymbol{E}^{\mathrm{p}}}\big] : \boldsymbol{N}^I_{\boldsymbol{E}^{\mathrm{p}}}\,\mathrm{d}V - \int_{\Omega}\big[\mathbb{D} : \mathring{\lambda}\partial_{\boldsymbol{E}^{\mathrm{p}}}\boldsymbol{N}^{\mathrm{p}} : \boldsymbol{N}^J_{\boldsymbol{E}^{\mathrm{p}}}\big] : \boldsymbol{N}^I_{\boldsymbol{E}^{\mathrm{p}}}\,\mathrm{d}V\\ &= \int_{\Omega}\Big[\mathbb{D} : \big[\sigma\mathbb{I} - \mathring{\lambda}\partial_{\boldsymbol{E}^{\mathrm{p}}}\boldsymbol{N}^{\mathrm{p}}\big] : \boldsymbol{N}^J_{\boldsymbol{E}^{\mathrm{p}}}\Big] : \boldsymbol{N}^I_{\boldsymbol{E}^{\mathrm{p}}}\,\mathrm{d}V\,,\end{aligned}$$

$$\big[\mathsf{K}_{\boldsymbol{E}^{\mathrm{p}}\lambda}\big]_{IJ} = \sigma\,\partial_{\mathring{\lambda}^J}\mathsf{R}^I_{\boldsymbol{E}^{\mathrm{p}}} = \sigma\int_{\Omega}\partial_{\mathring{\lambda}^J}\big[\mathbb{D} : \big[\mathring{\boldsymbol{E}}^{\mathrm{p}} - \mathring{\lambda}\boldsymbol{N}^{\mathrm{p}}\big]\,\mathrm{d}V = -\sigma\int_{\Omega}\big[\mathbb{D} : \boldsymbol{N}^{\mathrm{p}}N^J_{\lambda}\big] : \boldsymbol{N}^I_{\boldsymbol{E}^{\mathrm{p}}}\,\mathrm{d}V\,,$$

$$\big[\mathsf{K}_{\lambda\boldsymbol{u}}\big]_{IJ} = \partial_{\mathsf{u}^J}\mathsf{R}^I_{\lambda} = \int_{\Omega}\partial_{\boldsymbol{u}^J}\big[c_{\lambda}\mathring{\lambda} - f - \big|c_{\lambda}\mathring{\lambda} + f\big|\big]\,\mathrm{d}V = \int_{\Omega}\big[\partial_{\boldsymbol{F}}c : \mathrm{Grad}\boldsymbol{N}^J_{\boldsymbol{u}}\big]N^I_{\lambda}\,\mathrm{d}V\,,$$

$$\big[\mathsf{K}_{\lambda\boldsymbol{E}^{\mathrm{p}}}\big]_{IJ} = \partial_{\mathsf{E}^{\mathrm{p}J}}\mathsf{R}^I_{\lambda} = \int_{\Omega}\partial_{\boldsymbol{E}^{\mathrm{p}J}}\big[c_{\lambda}\mathring{\lambda} - f - \big|c_{\lambda}\mathring{\lambda} + f\big|\big]\,\mathrm{d}V = \int_{\Omega}\big[\partial_{\boldsymbol{E}^{\mathrm{p}}}c : \boldsymbol{N}^J_{\boldsymbol{E}^{\mathrm{p}}}\big]N^I_{\lambda}\,\mathrm{d}V\,,$$

$$\big[\mathsf{K}_{\lambda\lambda}\big]_{IJ} = \sigma\,\partial_{\mathring{\lambda}^J}\mathsf{R}^I_{\lambda} = \sigma\int_{\Omega}\partial_{\mathring{\lambda}^J}\big[c_{\lambda}\mathring{\lambda} - f - \big|c_{\lambda}\mathring{\lambda} + f\big|\big]\,\mathrm{d}V = \sigma\int_{\Omega}\big[\partial_{\mathring{\lambda}}c : N^J_{\lambda}\big]N^I_{\lambda}\,\mathrm{d}V\,.$$

The partial derivatives of the various quantities in the components of the tangent matrix can be obtained with aid of the following relations:

$$\mathbb{C}^{\mathrm{ep}}_{\mathrm{N}} := \partial_{\boldsymbol{F}}\boldsymbol{P} = \partial_{\boldsymbol{F}}[\boldsymbol{F}\cdot\boldsymbol{S}] = \mathbb{I}\cdot\boldsymbol{S} + \boldsymbol{F}\cdot\mathbb{C}^{\mathrm{ep}}_{\mathrm{L}} : \partial_{\boldsymbol{F}}\boldsymbol{C}$$

where

$$\partial_{\boldsymbol{F}}\boldsymbol{C} = \boldsymbol{I}\underline{\otimes}\boldsymbol{F}^{\mathsf{T}} + \boldsymbol{F}^{\mathsf{T}}\overline{\otimes}\boldsymbol{I}, \qquad \mathbb{C}^{\mathrm{ep}}_{\mathrm{L}} = \mathbb{P}^{\mathsf{T}}_{\mathrm{L}} : \mathbb{D} : \mathbb{P}_{\mathrm{L}} + \boldsymbol{T} : \mathbb{L}_{\mathrm{L}}, \qquad \mathbb{L}_{\mathrm{L}} = 4\partial^2_{\boldsymbol{C}\boldsymbol{C}}\boldsymbol{E}.$$

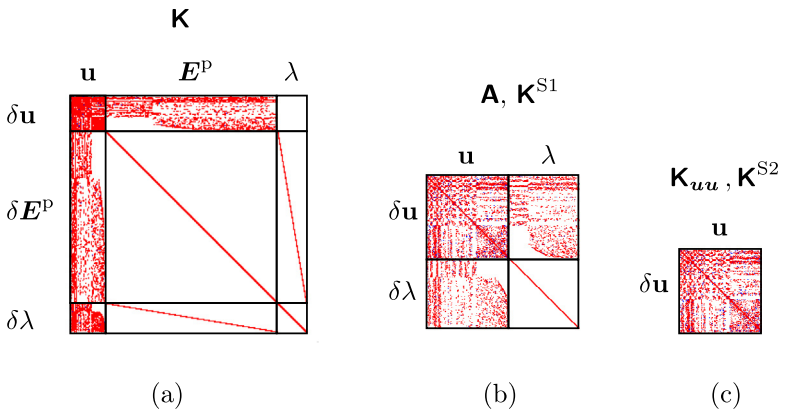

Fig. 2. Block matrices structure resulting from the multifield plasticity problem's nested solution strategy: (a) the full matrix K with diagonal block matrices $\mathsf{K}_{\boldsymbol{E}^{\mathrm{p}}\boldsymbol{E}^{\mathrm{p}}}$ ($\delta\mathbf{E}^{\mathrm{p}}\mathbf{E}^{\mathrm{p}}$ block) and $\mathsf{K}_{\lambda\lambda}$ ($\delta\lambda\lambda$ block), (b) the matrix K^{S1} as defined in (34) and (c) the structure of $\mathsf{K}_{\boldsymbol{uu}}$ and K^{S2} defined in Eq. (36). Note the sparsity pattern of the blocks associated with the plastic deformation fields $\boldsymbol{E}^{\mathrm{p}}$ and λ arising from their approximation in L^2 space.

Hence,

$$\partial_{\boldsymbol{F}}\boldsymbol{P} = \mathbb{C}^{\mathrm{ep}}_{\mathrm{N}}\,, \qquad \partial_{\boldsymbol{E}^{\mathrm{p}}}\boldsymbol{P} = -\mathbb{P} : \mathbb{D}\,,$$

$$\partial_{\boldsymbol{T}}\boldsymbol{N}^{\mathrm{p}} = \frac{1}{\|\boldsymbol{T}+\boldsymbol{B}\|_{\mathbb{I}^{\mathrm{dev}}}}\left[\mathbb{I}^{\mathrm{dev}} - \boldsymbol{N}^{\mathrm{p}}\otimes\boldsymbol{N}^{\mathrm{p}}\right], \qquad \partial_{\boldsymbol{F}}\boldsymbol{N}^{\mathrm{p}} = \partial_{\boldsymbol{T}}\boldsymbol{N}^{\mathrm{p}} : \mathbb{D} : \underbrace{\frac{1}{2}\mathbb{P}_{\mathrm{L}} : \partial_{\boldsymbol{F}}\boldsymbol{C}}_{\partial_{\boldsymbol{F}}\boldsymbol{T}}\,,$$

$$\partial_{\boldsymbol{F}}\boldsymbol{N} = -\partial_{\boldsymbol{T}}\boldsymbol{N} : \mathbb{D}\,,$$

$$\partial_{f}c = -\left[1 + \frac{c_{\lambda}\mathring{\lambda} + f}{|c_{\lambda}\mathring{\lambda} + f|}\right], \qquad \partial_{\boldsymbol{T}}f = \frac{\mathbb{I}^{\mathrm{dev}} : [\boldsymbol{T}+\boldsymbol{B}]}{\|\boldsymbol{T}+\boldsymbol{B}\|_{\mathbb{I}^{\mathrm{dev}}}}\,,$$

$$\partial_{\boldsymbol{F}}c = \partial_{f}c\,\partial_{\boldsymbol{T}}f : \mathbb{D} : \frac{1}{2}\mathbb{P}_{\mathrm{L}} : \partial_{\boldsymbol{F}}\boldsymbol{C} = \partial_{f}c\,\partial_{\boldsymbol{T}}f : \partial_{\boldsymbol{F}}\boldsymbol{T}\,, \qquad \partial_{\boldsymbol{E}^{\mathrm{p}}}c = -\partial_{f}c\,\partial_{\boldsymbol{T}}f : \mathbb{D}\,,$$

$$\partial_{\mathring{\lambda}}c = \left[c_{\lambda} + \frac{c_{\lambda}\mathring{\lambda} + f}{|c_{\lambda}\mathring{\lambda} + f|}\right].$$

5.4. Block matrix solver

A naive application of the Newton–Raphson approximation to the fully discrete global problem (28) leads to a proliferation of the number of degrees of freedom. The size of the resulting tangent matrix K is large in comparison to classical single-field approaches for plasticity where there is only one unknown global field, i.e., the displacement $\boldsymbol{u}$. An efficient solution procedure is therefore required for the multifield formulation to be competitive. As now detailed, this is achieved by exploiting the structure of the tangent matrix, as depicted in Fig. 2. More specifically, the linear system resulting from the Newton–Raphson approximation (28) is solved approximately in a nested manner using block matrix solvers available in PETSc [50] via the *FieldSplit* functionality. This allows the solution of the fully discrete global problem to be obtained from the solution of a series of significantly smaller systems of equations.

An alternative approach to reduce the size of the global problem is to statically condense out the plastic strain and consistency parameter fields during assembly, in a similar manner to [21] for a formulation of multifield plasticity at small strains. However, this requires substantial computational effort and is cumbersome for nonlinear problems such as finite strain plasticity, since it requires the calculation of the (consistent) tangent matrix for the statically condensed system.

By construction, the PETSc *FieldSplit* solver [50] assumes a two-by-two block matrix structure. Given this structure, the abstract algebraic expressions for the solution of an arbitrary two-by-two block matrix system following from a chain of multiplications of simpler matrices can be obtained from the procedure presented in Box 5.4.1 [see e.g., 51].

Box. 5.4.1: *FieldSplit* solution procedure for an arbitrary two-by-two block matrix system

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{S}^{-1} & -\mathbf{S}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{S}^{-1} & \mathbf{A}_{22}^{-1} + \mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{S}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}, \tag{30}$$

where $\mathbf{S}$ is the Schur complement of the block matrix $\mathbf{A}_{22}$ defined by $\mathbf{S} = \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}$. The expression (30) can be decomposed in terms of a chain of simpler matrix multiplications as follows:

$$\begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22}^{-1} \end{bmatrix} \overbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{A}_{21} & \mathbf{I} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{S}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix} \overbrace{\begin{bmatrix} \mathbf{I} & -\mathbf{A}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}}^{[\mathbf{a}_1^{[1]}\ \mathbf{a}_2^{[1]}]^{\mathrm{T}}}}^{[\mathbf{a}_1^{[2]}\ \mathbf{a}_2^{[2]}]^{\mathrm{T}}}}^{[\mathbf{a}_1^{[3]}\ \mathbf{a}_2^{[3]}]^{\mathrm{T}}}}^{[\mathbf{a}_1^{[4]}\ \mathbf{a}_2^{[4]}]^{\mathrm{T}}}}. \tag{31}$$

The solution process presented in Eq. (31) should be understood as a chain multiplication starting from the rightmost matrix–vector multiplication $[\mathbf{a}_1^{[1]}\ \mathbf{a}_2^{[1]}]^{\mathrm{T}}$ followed by the next matrix multiplication $[\mathbf{a}_1^{[2]}\ \mathbf{a}_2^{[2]}]^{\mathrm{T}}$ and so on. Henceforth, block-vectors annotated with a superscript $[n]$ refer to the result of the nth multiplication of the scheme (31). It is important to note that the action of the inverse matrices $\mathbf{S}^{-1}$ and $\mathbf{A}_{22}^{-1}$ with the vectors resulting from prior stages of the chain can be evaluated via the (possibly approximate) solution of a linear system of equations with size equal to the size of the matrix under consideration. Thus, rather than explicitly evaluating $\mathbf{S}^{-1}$ and $\mathbf{A}_{22}^{-1}$, one solves the sub-problems $\mathbf{A}_{22}\mathbf{a}_2^{[1]} = \mathbf{f}_2$, $\mathbf{S}\mathbf{u}_1^{[3]} = \mathbf{u}_1^{[2]}$ and $\mathbf{A}_{22}\mathbf{a}_2 = \mathbf{a}_2^{[4]}$ as part of the 1st, 3rd and 5th multiplications, respectively, in Eq. (31). Various approaches can be chosen to solve these sub-problems. The choice of the generalised minimal residual (GMRES) method with either block Jacobi (BJacobi) or algebraic multigrid (GAMG) preconditioners is assessed in Section 6.4. A direct solver could also be employed.

The block structure of $\mathbf{K}$ is depicted in Fig. 2(a). The block diagonal structure of $\mathbf{K}_{\boldsymbol{E}^{\mathrm{p}}\boldsymbol{E}^{\mathrm{p}}}$ and $\mathbf{K}_{\lambda\lambda}$ is an emerging property of functional setting whereby $\boldsymbol{E}^{\mathrm{p}}$, $\delta\boldsymbol{E}^{\mathrm{p}} \in L^2(\Omega)$ and λ, $\delta\lambda \in L^2(\Omega)$. This property can be exploited in the solution scheme (31).

The application of the general procedure for a two-by-two block matrix system presented in Box 5.4.1 to the tangent matrix for the multifield problem (29) follows as the composition of two nested *FieldSplit* solution schemes. This is done by considering two two-by-two block systems denoted as the *outer* and *inner* problems.

Outer problem

The outer problem involves the *FieldSplit* solution of the system (29) by partitioning (without the need to explicitly move columns and rows) the matrix $\mathbf{K}$ as

$$\mathbf{K}\Delta\mathbf{d} = -\mathbf{R} \implies \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{K}_{\boldsymbol{E}^{\mathrm{p}}\boldsymbol{E}^{\mathrm{p}}} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{s} \\ \Delta\mathbf{E}^{\mathrm{p}} \end{bmatrix} = -\begin{bmatrix} \mathbf{r}_{\mathrm{s}} \\ \mathbf{R}_{\boldsymbol{E}^{\mathrm{p}}} \end{bmatrix}, \tag{32}$$

where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\Delta\mathbf{s}$ and $\mathbf{r}_s$ are defined by

$$\mathbf{A} := \begin{bmatrix} \mathbf{K}_{uu} & \mathbf{0} \\ \mathbf{K}_{\lambda u} & \mathbf{K}_{\lambda\lambda} \end{bmatrix}, \quad \mathbf{B} := \begin{bmatrix} \mathbf{K}_{u E^{\mathrm{p}}} \\ \mathbf{K}_{\lambda E^{\mathrm{p}}} \end{bmatrix}, \quad \mathbf{C} := \begin{bmatrix} \mathbf{K}_{E^{\mathrm{p}} u} \\ \mathbf{K}_{E^{\mathrm{p}} \lambda} \end{bmatrix}^{\mathrm{T}}, \quad \Delta\mathbf{s} := \begin{bmatrix} \Delta\mathbf{u} \\ \Delta\lambda \end{bmatrix}, \quad \mathbf{r}_s := \begin{bmatrix} \mathbf{R}_u \\ \mathbf{R}_\lambda \end{bmatrix}. \tag{33}$$

The block matrices of the outer problem are related to the *FieldSplit* scheme in 30 as follows: $\mathbf{S} = \mathbf{K}/\mathbf{K}_{E^{\mathrm{p}}E^{\mathrm{p}}}$, $\mathbf{A}_{12} = \mathbf{B}$, $\mathbf{A}_{21} = \mathbf{C}$ and $\mathbf{A}_{22} = \mathbf{K}_{E^{\mathrm{p}}E^{\mathrm{p}}}$, where $\mathbf{K}/\mathbf{K}_{E^{\mathrm{p}}E^{\mathrm{p}}}$ is the Schur complement of $\mathbf{K}_{E^{\mathrm{p}}E^{\mathrm{p}}}$ defined by

$$\begin{aligned} \mathbf{K}/\mathbf{K}_{E^{\mathrm{p}}E^{\mathrm{p}}} = \mathbf{K}^{\mathrm{S1}} &:= \begin{bmatrix} \mathbf{K}_{uu} - \mathbf{K}_{u E^{\mathrm{p}}} \mathbf{K}^{-1}_{E^{\mathrm{p}}E^{\mathrm{p}}} \mathbf{K}_{E^{\mathrm{p}} u} & -\mathbf{K}_{u E^{\mathrm{p}}} \mathbf{K}^{-1}_{E^{\mathrm{p}}E^{\mathrm{p}}} \mathbf{K}_{E^{\mathrm{p}}\lambda} \\ \mathbf{K}_{\lambda u} - \mathbf{K}_{\lambda E^{\mathrm{p}}} \mathbf{K}^{-1}_{E^{\mathrm{p}}E^{\mathrm{p}}} \mathbf{K}_{E^{\mathrm{p}} u} & \mathbf{K}_{\lambda\lambda} - \mathbf{K}_{\lambda E^{\mathrm{p}}} \mathbf{K}^{-1}_{E^{\mathrm{p}}E^{\mathrm{p}}} \mathbf{K}_{E^{\mathrm{p}} u} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{S}_{uu} & \mathbf{S}_{u\lambda} \\ \mathbf{S}_{\lambda u} & \mathbf{S}_{\lambda\lambda} \end{bmatrix}. \end{aligned} \tag{34}$$

The matrix $\mathbf{K}^{\mathrm{S1}}$ has the same structure as $\mathbf{A}$ as shown in Fig. 2(b). Thus, the block diagonal structure of the matrix $\mathbf{S}_{\lambda\lambda}$ can be exploited when solving $\mathbf{K}^{\mathrm{S1}}\Delta\mathbf{s}^{[3]} = \Delta\mathbf{s}^{[2]}$ using the scheme (31). This leads to the definition of the *inner problem*.

Inner problem

The inner problem is to find the solution of

$$\begin{bmatrix} \mathbf{S}_{uu} & \mathbf{S}_{u\lambda} \\ \mathbf{S}_{\lambda u} & \mathbf{S}_{\lambda\lambda} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u}^{[3]} \\ \Delta\lambda^{[3]} \end{bmatrix} = \begin{bmatrix} \Delta\mathbf{u}^{[2]} \\ \Delta\lambda^{[2]} \end{bmatrix}. \tag{35}$$

The *FieldSplit* solver is again invoked. In a similar manner to the outer problem, a Schur complement is used to solve (35). This entails the solution of three smaller sub-problems involving the matrices $\mathbf{S}_{\lambda\lambda}$ and $\mathbf{K}^{\mathrm{S1}}/\mathbf{S}_{\lambda\lambda}$ defined by

$$\mathbf{K}^{\mathrm{S1}}/\mathbf{S}_{\lambda\lambda} = \mathbf{K}^{\mathrm{S2}} := \mathbf{S}_{uu} - \mathbf{S}_{u\lambda}\mathbf{S}^{-1}_{\lambda\lambda}\mathbf{S}_{\lambda u}. \tag{36}$$

Note that the matrix $\mathbf{K}^{\mathrm{S2}}$ has the same structure as $\mathbf{K}_{uu}$ as depicted in Fig. 2(c). Thus it is of the same size and structure as the tangent matrix for the classical elastoplasticity problem.

The key matrices for the two-level nested solution process employed are summarised in Box 5.4.2. Further details of the implementation in PETSc are provided in the Appendix.

Box. 5.4.2: Matrices utilised for the solution of the two-level nested scheme

Outer problem:

- $\mathbf{K}$: solve associated problem using Schur complement *FieldSplit* scheme (Box 5.4.1)
- $\mathbf{K}^{\mathrm{S1}}$: the Schur complement of the outer problem, the solution of the associated problem is the subject of the inner problem
- $\mathbf{K}_{E^{\mathrm{p}}E^{\mathrm{p}}}$: solve associated problem using GMRES with either GAMG or BJacobi preconditioner

Inner problem:

- $\mathbf{K}^{\mathrm{S1}}$: solve associated problem using Schur complement *FieldSplit* scheme (Box 5.4.1)
- $\mathbf{K}^{\mathrm{S2}}$: solve associated problem using GMRES with either GAMG or BJacobi preconditioner
- $\mathbf{S}_{\lambda\lambda}$: solve associated problem using GMRES with either GAMG or BJacobi preconditioner

5.5. Scaling and convergence criteria

The discretised form of the residual Eqs. (25)–(27) have units of work. The three global residual vectors are scaled by a physical parameter with units of stress (i.e., the Young's modulus, the yield stress and the hardening parameter, respectively) to convert them to units of length with magnitudes proportional to the characteristic size of the body L. As a consequence, the stabilisation parameter is chosen as $c_\lambda = 1$.

The Newton–Raphson scheme is deemed converged if the L^2 norm of the residual vector satisfies the following stopping criterion:

$$\|\mathbf{R}\| < \min\left(\frac{\mathrm{tol}^{\mathrm{a}}}{L},\ \mathrm{tol}^{\mathrm{r}}\|\mathbf{R}^0\|\right) = \min\left(\overline{\mathrm{tol}}^{\mathrm{a}},\ \overline{\mathrm{tol}}^{\mathrm{r}}\right),$$

where tol^{a} and tol^{r} are the absolute and relative tolerances, respectively, and $\mathbf{R}^0$ is the residual corresponding to the initial Newton iteration.

5.6. Influence of decomposition of tensor of elastic moduli on convergence behaviour

The robustness of the solution method is improved by following Remark 3. The incompressibility constraint for plastic flow (1) states that $\text{tr}[\boldsymbol{E}^{\text{p}}] = 0$. Hence, from Eq. (4), $\text{tr}[\dot{\boldsymbol{E}}^{\text{p}}] = \text{tr}[\boldsymbol{N}^{\text{p}}] = 0$. For pressure-independent yield functions, as considered here, the following equality holds $f(\boldsymbol{T}, \boldsymbol{E}^{\text{p}}, \lambda) = f(\boldsymbol{T}^{\mu}, \boldsymbol{E}^{\text{p}}, \lambda)$, where $\boldsymbol{T}^{\mu}$ was defined in Eq. (13). That is, using $\boldsymbol{T}^{\mu}$ to evaluate the residual Eqs. (26)–(27) does not change the results; however, it improves robustness. This improvement stems from the observation that during the Newton–Raphson process the computation of the global tangent matrix contains terms involving the sensitivity of the direction of plastic flow $\boldsymbol{N}^{\text{p}}$ relative to changes in the primary fields. From plastic incompressibility we expect $\boldsymbol{N}^{\text{p}}$ to be isochoric, but this need not necessarily be the case at non-converged Newton steps. Consequently, using $\boldsymbol{T}^{\mu}$ in (26)–(27) ensures that one satisfies a priori the incompressibility constraint, by forcing the non-equilibrated solution to remain on the constraint manifold. This improves robustness and does not have any effect on the physical correctness or accuracy of the solution.

5.7. Numerical integration in space and time

The non-smoothness of the Fischer–Burmeister complementarity function (8) was discussed in Remark 1. The numerical integration of the residual terms in $\mathbf{R}_{\lambda}$ requires careful consideration. Consider an element intersected by a plastic front, i.e., where part of the element is active and part inactive. The tangential derivative of the complementarity function is discontinuous at the plastic front. We address that issue by adopting a pragmatic approach, which provides a robust solution scheme, by applying a standard Gaussian quadrature rule order for tetrahedral elements [see e.g., 52]. The same integration rule is used to evaluate all terms in the left and the right-hand side of the linearised algebraic equations and has been shown to be robust. A backward Euler time integration scheme is employed for all examples.

6. Numerical examples

The multifield plasticity formulation is validated using a series of benchmark problems. It is shown that the formulation is as accurate as classical approaches using either a multiplicative decomposition of the deformation gradient or a logarithmic strain formulation. The performance of the approach is assessed, and features of the underlying theory are elucidated.

The implementation of the finite element model is performed in the open-source library MoFEM [53]. MoFEM is built upon the PETSc library [54]. PETSc provides a wide range of preconditioners, like *FieldSplit*, which allows for the hierarchical construction of block solvers. The preconditioner can be nested inside other preconditioners, such as multigrid or a higher level *FieldSplit*, with the construction of the hierarchy and other algorithmic choices as runtime options. Fundamental in *FieldSplit* preconditioning is the concept of extracting submatrices from a global matrix acting on the coupled system, making it ideally suited for multifield plasticity.

The order of the polynomial approximations for the displacement, plastic strain, and consistency parameter fields are chosen as 2, 1, and 1, respectively.

6.1. Beam with self-weight

As presented in [55], consider a beam with rectangular cross section subjected to loading in the form of self-weight with $\boldsymbol{b} = b\boldsymbol{e}_3$, where $b = -t\,50 \times 10^6\,\text{N/m}^3$ and $t \in [0, 1]$. The beam has a length of 1 m, a height of 0.1 m and a width of 0.04 m. The left surface of the beam is fully fixed. Symmetry boundary conditions are imposed on the right surface. The materials parameters are listed in Table 1. The beam is uniformly discretised into $30 \times 5 \times 8$ tetrahedral elements so as to match [55].

The vertical deflection of a point, located at the top of the right edge of the beam and denoted A, with increasing load parameter t is shown in Fig. 3. The distribution of plastic strain at the end of the analysis is non-symmetric due to the geometric non-linearities inherent in the model. The solution obtained using the multifield formulation

Table 1
Constitutive parameters for the problem of a beam subjected to self-weight.

Parameter	Symbol	Value
Young's modulus	E	$210 \times 10^9\,\mathrm{N/m^2}$
Poisson's ratio	ν	0.3
Yield stress	σ_y	$250 \times 10^6\,\mathrm{N/m^2}$
Hardening modulus	H	$1 \times 10^6\,\mathrm{N/m^2}$

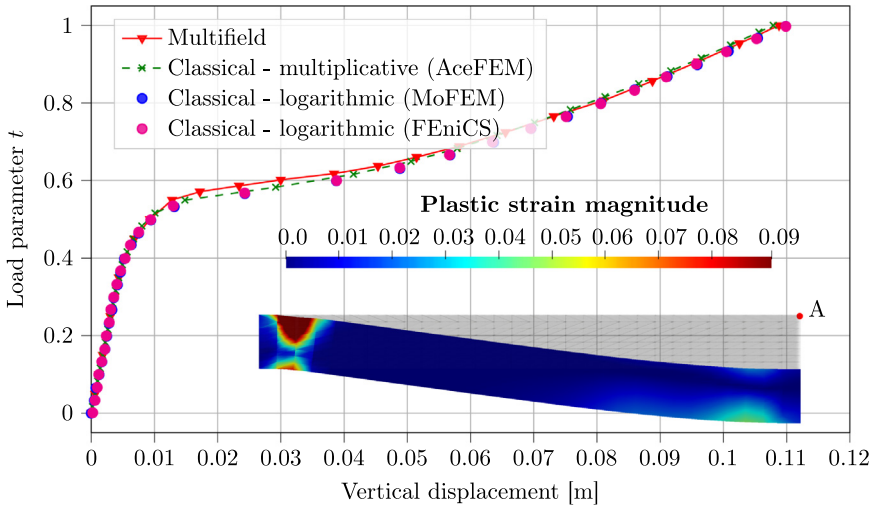

Fig. 3. Comparison between the multifield formulation and (i) a classical approach based on a multiplicative decomposition of the deformation gradient implemented in AceFEM and (ii,iii) a classical approach based on the logarithmic strain formulation and implemented in the libraries MoFEM and FEniCS, respectively, for the problem of a beam with self-weight. The vertical deflection of point A with increasing load and the distribution of the plastic strain magnitude at the final load step are shown.

is compared to (i) a classical approach based on a multiplicative decomposition of the deformation gradient implemented in the library AceFEM [56], (ii) the MoFEM library with a classical approach based on the logarithmic strain formulation implemented using the MFront material library [57] and (iii) the FEniCS library [58] with a logarithmic strain formulation, again provided by MFront [55]. The three sets of results are nearly indistinguishable except for a minor discrepancy in the hardening response post yield. The distribution of the magnitude of the plastic strain at the end of the analysis is also shown. The plastic hinge that forms in the vicinity of the fixed boundary is clearly visible.

6.2. Necking of a rod

This well-studied benchmark problem [see e.g., 7,11,59] concerns the necking of a circular rod of length 53.34 mm and radius 6.4135 mm subject to displacement-controlled loading. A geometrical imperfection, in the form a 1.8 % reduction in the central radius of the rod, is introduced to trigger necking. The applied displacement of 14 mm is imposed in equal load steps with the load parameter $t \in [0, 1]$. The linear isotropic hardening model in (3) is extended to a saturation-type nonlinear isotropic hardening response where $\beta = -H\alpha - [\sigma_y^\infty - \sigma_y^0][1 - \exp[-\omega\alpha]]$. The constitutive parameters and their definitions are listed in Table 2.

The symmetry of the problem allows for 1/8 of the domain to be simulated. The problem is discretised using 10 390 elements with the mesh finer along the first 8.98 mm of the rod.

The distribution of the plastic strain over the deformed configuration at various stages during the loading process is shown in Fig. 4. The necking of the central region of the rod, representative of the signature response for this problem, is clearly visible.

The evolution of the reaction force and the radial contraction due to the applied axial displacement are shown in Figs. 5(a) and (b), respectively, together with the results from Miehe et al. [7] obtained using a classical approach

Table 2
Constitutive parameters for the necking of a rod problem.

Parameter	Symbol	Value
Bulk modulus	κ	164.206 kN/mm^2
Shear modulus	μ	80.1938 kN/mm^2
Initial yield stress	σ_y^0	0.45 kN/mm^2
Infinite yield stress	σ_y^∞	0.715 kN/mm^2
Saturation parameter	ω	16.93

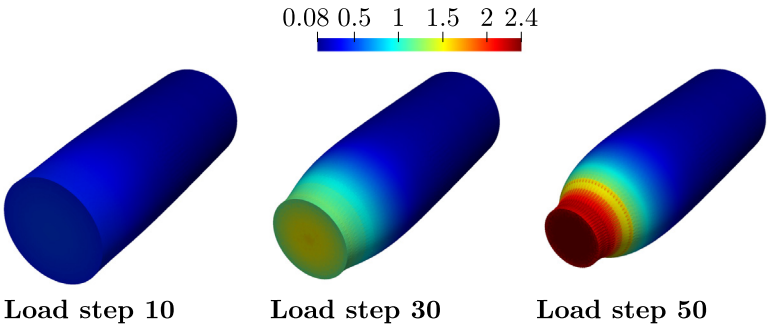

Fig. 4. The distribution of equivalent plastic strain at various load steps during the necking of a rod. Only half of the domain is shown.

with a logarithmic strain approximation. The multifield formulation is in good overall agreement with the classical approach. A minor difference in the radial contraction occurs towards the end of the simulation. This can be attributed to the differences in mesh density and the mesh distortion that occurs at these high levels of deformation.

The robustness and efficiency of the multifield formulation are assessed in Fig. 5 where the number of Newton iterations per load step is shown. The reaction force at each load step is also indicated. The discrepancy between the reference and the proposed model is attributed to differences in the element type and the spatial discretisation. The analysis is characterised by two distinct stages differentiated by the point at which the peak reaction force occurs. In the first stage, necking has not yet occurred. The number of Newton iterations required for convergence is generally 3 in this stage. At the onset of necking (load step 22) the number of Newton steps increases to 8 but remains below this thereafter and averages to 6. The convergence behaviour of the Newton scheme at selected load steps is shown in Fig. 6. An asymptotic quadratic rate of convergence is observed for all load steps, with the difference between load steps dependent on the behaviour during the initial iterations where the active set can change.

6.3. Squeezed tube

Considers a circular axisymmetric tube of length 40, inner radius 4 and thickness 1 length units, squeezed in the middle by two rigid cylindrical tools of radius 5, as depicted in Fig. 7. The problem is similar to that proposed by Seitz et al. [60], however we assume the contact to be frictionless as the purpose here is to assess the performance of the multifield algorithm and not the accuracy of the contact model. The tube is fully fixed on both ends. Due to symmetry, only 1/8 of the problem is considered with the appropriate boundary conditions for the displacements and contact constraints imposed. The cylinders are initially not in contact with the tube. They are then displaced inwards by 3 length units towards the centre of the tube, thereby initiating contact. Thereafter they are retracted in one load step.

A linear isotropic hardening response is assumed. The material parameters are $E = 206\,900$, $\nu = 0.29$, $\sigma_y^0 = 450$ and $H = 100$. The tube is meshed using 3447 tetrahedral elements.

Fig. 8 shows the accumulated plastic strain distribution superimposed on the deformed geometry for various tool displacements. The predicted deformation is in good agreement with Seitz et al. [60]. The displacement of the tool versus the resulting contact force is shown in Fig. 9 and compared to Seitz et al. who accounted for friction at the contact interface. Given this difference in the contact model, the results of the multifield model are deemed reasonable and the problem will now be used to assess the computational efficiency of the approach.

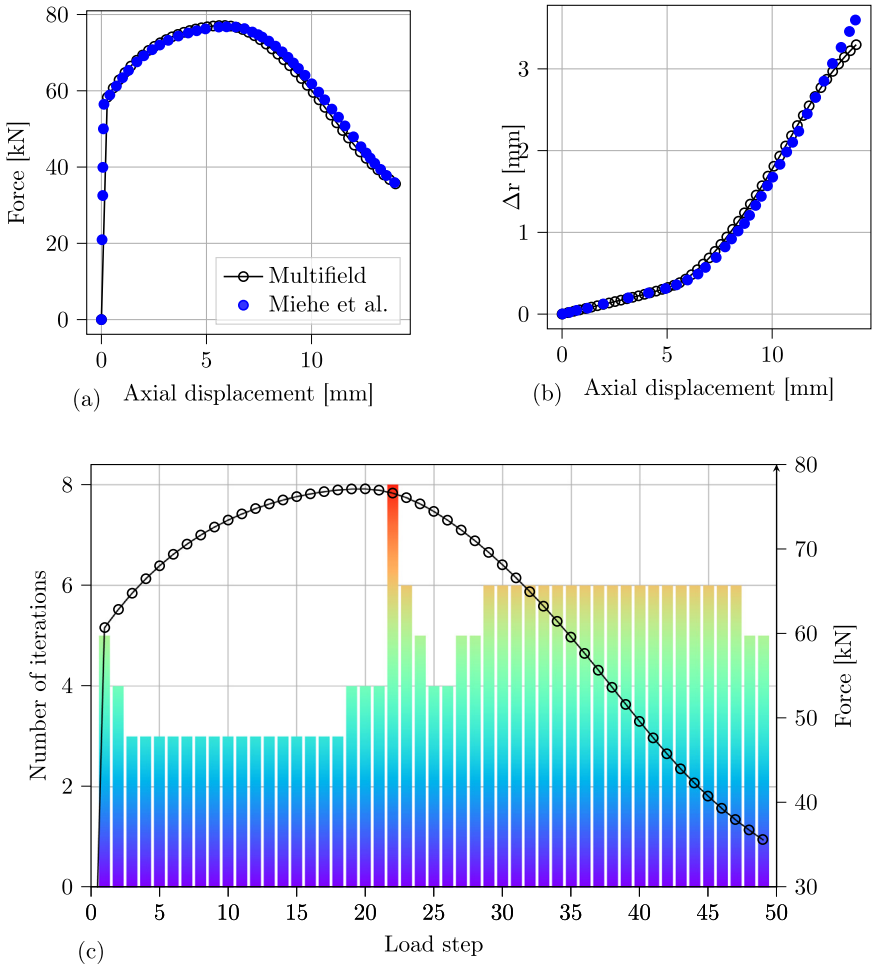

Fig. 5. Comparison of (a) the force versus axial displacement and (b) the radial contraction Δr versus axial displacement, obtained using the multifield approach and the classical approach with a logarithmic strain approximation by Miehe et al. [7]. In (c) the number of iterations per load step is shown together with the corresponding reaction force.

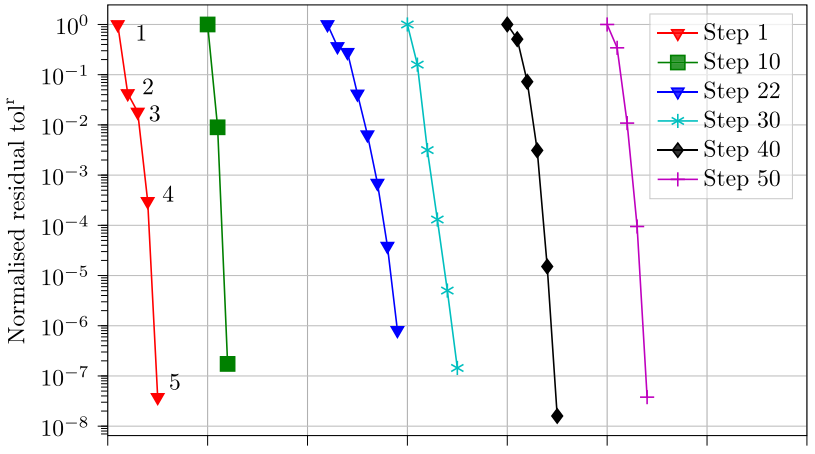

Fig. 6. Value of the residual at each Newton iteration for 6 selected load steps (1, 10, 22, 30 and 50) for the problem of necking of a rod.

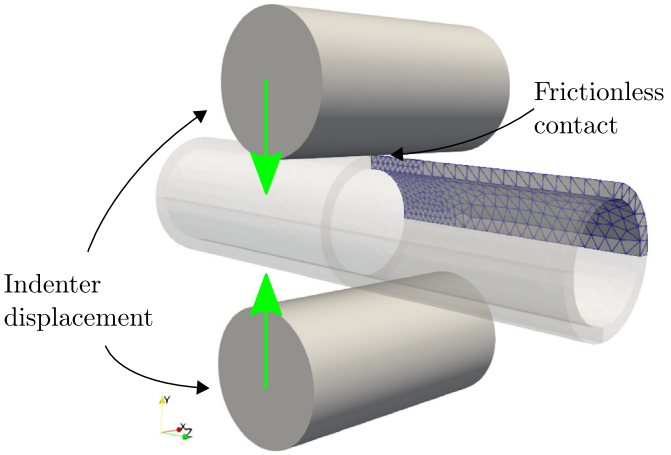

Fig. 7. The initial setup for the problem of a tube squeezed by rigid cylindrical tools.

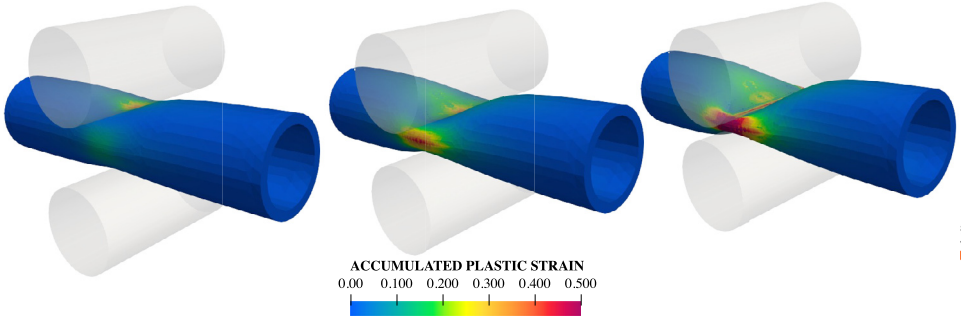

Fig. 8. Deformed configuration of the tube at different stages of squeezing with the accumulated plastic strain distribution superimposed.

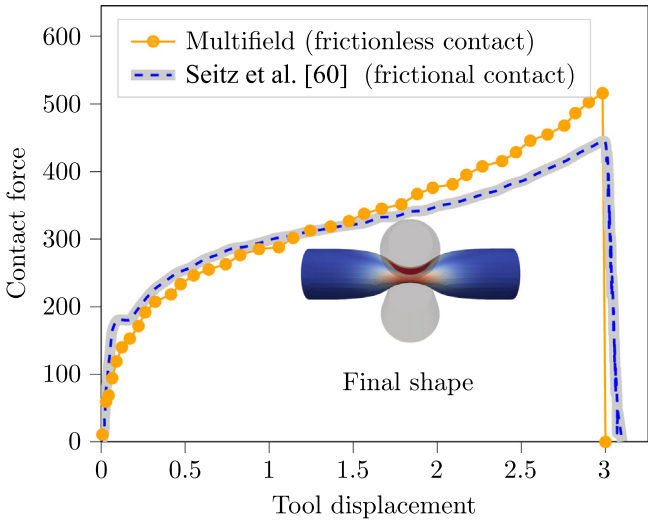

Fig. 9. The evolution of the total contact forces and final shape of the squeezed plastic tube.

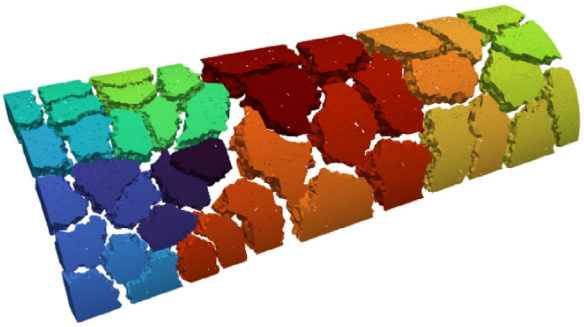

Fig. 10. 42 parallel partitions for the scalability study for a mesh of 315 125 tetrahedral elements.

6.4. Scalability of the block solver

6.4.1. Problem setup

The effectiveness and the scalability of the multifield formulation will now be assessed. To assess scalability, the squeezed tube domain from Section 6.3 is discretised using 315 125 tetrahedral elements. The total number of degrees of freedom (dofs) is 10 million. Analyses are performed using between 8 and 42 cores on a single node of a parallel computer system. The finite element mesh and parallel partitions for the 42-core simulation are shown in Fig. 10. Three solver configurations are considered with either a direct solver (MUMPS) used to solve the global problem (29) directly and without exploiting the block structure, or using the block matrix solver methodology described in Section 5.4 with either block Jacobi (BJacobi) or algebraic multigrid (GAMG) preconditioners employed for the outer and the inner problems (see Box 5.4.2). The number of GMRES iterations for each of the outer and inner problems was fixed to 3. Cumulative timings are evaluated for the first 20 load steps of the analysis indicated in Fig. 9 to ensure that a significant portion of the domain has undergone plastic deformation.

To better assess the performance of the block matrix solver, the runtime and scalability results are split into the two significant components, namely assembly and solving. The runtime and scalability for the assembly component are compared in Figs. 11 and 12, respectively. Linear (ideal) scaling of the assembly process is achieved for all cases. Assembly for the block matrix solver approach is always more time consuming than when compared to assembling the global system (29) alone. The choice of preconditioner used in the outer and inner problems influences the cumulative assembly time as a greater number of global Newton iterations are required for BJacobi than GAMG. Importantly however, the cost of the additional operations associated with the block matrix solver scales linearly and has a negligible effect when assembling larger problems, as seen in Fig. 12 where the timings for all approaches are converging. This matches the general expectation for a finite element implementation where the assembly time for a large sparse system is relatively small in comparison to the solver time.

Most of the computational effort is expended in the solution process. Here the block matrix solver approach demonstrates its superior performance, as shown in Fig. 13. For 42 cores, GAMG is almost 3 times faster, and BJacobi is almost 10 times faster, than MUMPS on the entire system. The block matrix solver also requires significantly less memory. For the 42-core case, the total virtual memory usage for MUMPS, BJacobi and GAMG was 191.2 GB, 44.9 GB and 72.2 GB, respectively.

The speedup comparison shown in Fig. 14 indicates that the overall efficiency of the block matrix solvers is approximately a factor of 1.5 below optimal for 42 cores. Note however that the direct solver (MUMPS) saturates fully at 32 cores whereafter the addition of more cores slows the analysis. The block matrix solver would still demonstrate speedup beyond 42 cores, albeit with diminishing returns as the work allocated to each core becomes too small to overcome the communication cost.

The numerical results presented in this section demonstrate that the proposed multifield approach for plasticity will allow for the solution of very large problems in a reasonable amount of time, utilising a large number of processors. The overhead due to the introduction of the additional fields is mitigated by applying efficient iterative solvers and exploiting the block matrix structure. Overall, the results show that tailored solvers offer a significant performance gain.

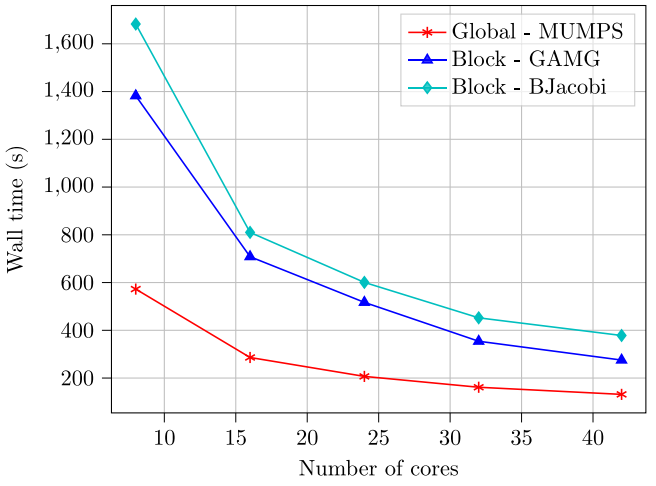

Fig. 11. Comparison of the cumulative runtime for the assembly component for the first load 20 steps. Global — MUMPS refers to the direct assembly of the global problem. Block refers to assembly using the block matrix solver methodology with either block Jacobi (BJacobi) or algebraic multigrid (GAMG) preconditioners employed for the outer and the inner problems.

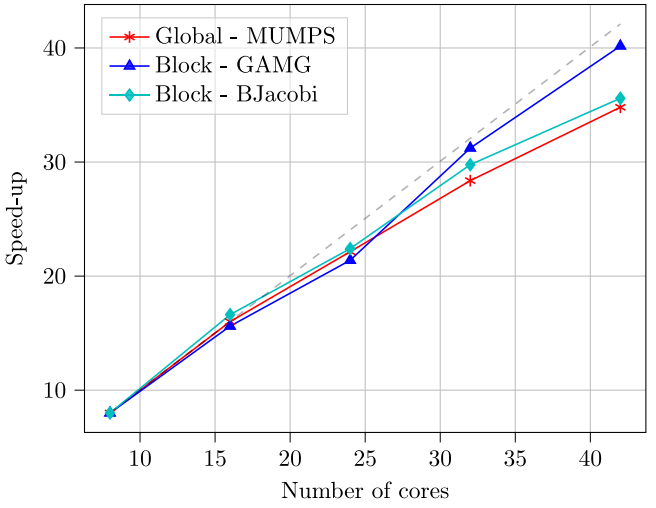

Fig. 12. Comparison of the assembly speedup for the first 20 load steps. The dashed line represents ideal speedup. Global — MUMPS refers to the direct assembly of the global problem. Block refers to assembly using the block matrix solver methodology with either block Jacobi (BJacobi) or algebraic multigrid (GAMG) preconditioners employed for the outer and the inner problems.

6.5. Comparison with classical computational plasticity

The multifield formulation circumvents the return mapping algorithm that is typically performed at the quadrature points of each finite element in classical computational plasticity [see e.g., 1,2, and the references therein]. This significantly reduces the complexity of the implementation at the cost of generating a significantly larger linear system of equations. In this section, a scalability study is performed to compare classical and multifield plasticity. The number of degrees of freedom in the multifield approach is approximately 7 times that of the classical approach.

To aid comparison, the classical scheme is implemented in the MoFEM library as a separate user module using the MFront library [61]. The constitutive model provided by MFront uses the same logarithmic strain framework for elastoplastic problems as implemented in the multifield framework. The MUMPS direct solver was used for both the classical scheme and for the solution of the linear problem associated with $\mathbf{K}^{S2}$ in the block matrix solver for the multifield approach — see Box 5.4.2 - as these problems are of identical size as indicated in Fig. 2 and

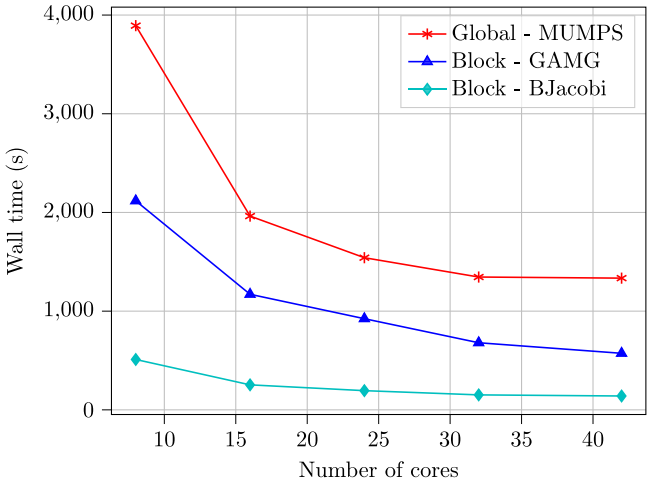

Fig. 13. Cumulative runtime for the solvers for the first 20 steps. Global — MUMPS refers to the direct solution of the global problem. Block refers to solution using the block matrix solver methodology with either block Jacobi (BJacobi) or algebraic multigrid (GAMG) preconditioners employed for the outer and the inner problems.

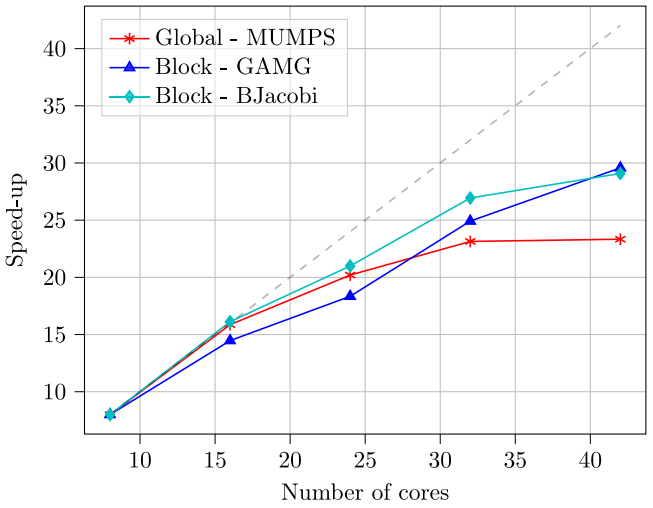

Fig. 14. Solver speedup for the first 20 steps. The dashed line represents ideal speedup. Global — MUMPS refers to the direct solution of the global problem. Block refers to solution using the block matrix solver methodology with either block Jacobi (BJacobi) or algebraic multigrid (GAMG) preconditioners employed for the outer and the inner problems.

therefore makes for a fair comparison of computational expense. The GMRES solver with BJacobi preconditioner was used for the remaining inner and outer computations in the block matrix solver.

The scalability tests are performed using 24 cores and varying the mesh density. The resulting plastic deformation obtained using the two approaches is identical. As indicated in Fig. 15, the performance of the multifield approach is less efficient for smaller problems. The performance gap between the approaches decreases as the problem size increases. At around 100 000 elements and above, both approaches perform similarly, i.e., the total execution time is almost identical. The multifield approach scales well and for large problems performs no worse than the classical methods for plasticity. When comparing the two approaches using CPU computing hardware, such performance is the optimal that could be expected from the multifield approach. That is, the total amount of time needed to resolve the remaining sub-problems outlined in Box. 5.4.2 is negligible compared to the solution of the $\mathbf{K}^{S2}$ sub-problem that is equal in size and sparsity with that of the classical approach. The negligible cost follows from the block diagonal structure of the sub-problems, as discussed in Section 5.4, with linear time complexity with respect to their

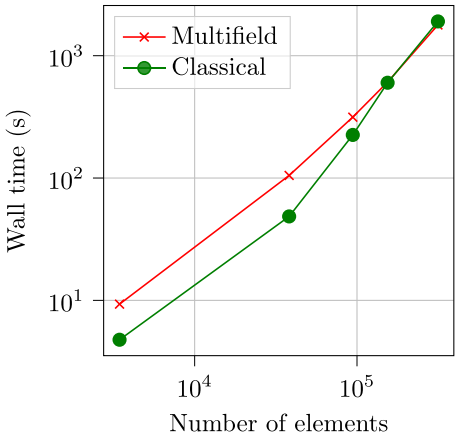

Fig. 15. Scalability comparison for the classical and multifield approaches based on the wall time, which represents the combined duration in seconds of the solver and assembly stages of the analysis.

size. A sparsity equivalent to the $\mathbf{K}^{S2}$ sub-problem will scale with higher than linear time complexity. As already discussed, the proposed approach is expected to perform better than the classical one when using next-generation architectures where the overhead generated by the memory required for the nested return mapping algorithms becomes a significant computational bottleneck.

7. Conclusion

A novel multifield formulation of plasticity at finite strains has been presented. The formulation has been validated using a series of benchmark problems. The competitive scalability of the block matrix solver has been demonstrated. Various choices of preconditioner for the inner and outer problems have been explored. The multifield approach has been shown to produce near-identical results as the classical formulation. In addition, the multifield formulation is no more expensive for large problems due to the ability of the block matrix solver to exploit the underlying structure of the various fields.

The present work will facilitate the efficient utilisation of computational elastoplasticity algorithms with implicit solvers in the next generation of computing hardware architectures. Implicit solution schemes are important for many crucial engineering applications, such as metal forming processes and structural stability problems. Critically, multifield approaches also provide a means for error control. The multifield formulation presented allows for future exploitation using fast matrix-free algorithms. The formation of the kernel for matrix-free methods enables greatly reduced memory bandwidth requirements at the cost of increasing the number of floating-point operations. This makes these methods well suited for accelerated hardware like GPUs and the utilisation of optimised sparse iterative solvers. This new performance model enables significantly faster and cheaper simulations at engineering tolerances. The classical approach, with its local integration of the elastoplastic model, makes it almost impossible to form an efficient kernel suitable for GPUs. To the best of the authors' knowledge, the implementation of elastoplasticity with implicit solvers for accelerated hardware remains an open challenge.

The proposed formulation is expected to be well suited for any flow relation with a smooth yield surface. Further research is required to extend the approach to non-smooth yield surfaces, such as for the Mohr–Coulomb constitutive model. Moreover, the formulation negates the need for solving for internal variables at the quadrature point level, as the plastic strain and the plastic multiplier are now unknown fields; these are solved at the global level making the method well suited for the development and assessment of *a posteriori* error indicators to guide mesh adaptivity. This will be explored in future work together with the extension to Arbitrary Lagrangian–Eulerian formulations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All code was developed using the open-source finite element library MoFEM.

Acknowledgements

The authors wish to thank the Engineering and Physical Sciences Research Council, United Kingdom grant EP/T008415/1 "Predictive Modelling for Incremental Cold Flow Forming: An integrated framework for fundamental understanding and process optimisation" for funding this work.

Appendix. Implementation of block matrix solver within PETSc

The inverses in the inner and outer levels of the block matrix solver are computed with either a direct solver or approximated using iterative methods via PETSc's *FieldSplit* class of preconditioners [62]. A standard parameter setup for the preconditioners and solvers is provided in Box A. The following conventions for PETSc run-time options are adopted. Each linear solver or sub-solver has two parts: the preconditioner (indicated by `-pc_type`) and the Krylov accelerator (indicated by `-ksp_type`). A single application of the preconditioner is denoted by `-ksp_type preonly`, while `-ksp_type gmres` denotes applying the preconditioner until a sufficient reduction in the residual norm or a certain number of iterations is achieved using the Generalised Minimal Residual method. For example, `-ksp_type gmres -pc_type bjacobi -ksp_max_it 3` indicates three iterations of Block-Jacobi. The solvers for sub-problems are prefixed (e.g., `_ep`, `_lambda`) to distinguish them.

Box. Appendix: PETSc preconditioners input parameters

```
# entire matrix K
-pc_type fieldsplit
-ksp_type preonly
-pc_fieldsplit_type schur
# for evaluation of inverse of block matrix K_{E^p E^p}
-fieldsplit_ep_ksp_type gmres
-fieldsplit_ep_ksp_max_it 3
-fieldsplit_ep_pc_type bjacobi
# for evaluation of inverse of block matrix K^{S1}
-fieldsplit_ulambda_ksp_type preonly
-fieldsplit_ulambda_pc_fieldsplit_type schur
-fieldsplit_ulambda_pc_type fieldsplit
# for evaluation of  inverse of block matrix S_{λλ}
-fieldsplit_ulambda_fieldsplit_lambda_ksp_type gmres
-fieldsplit_ulambda_fieldsplit_lambda_ksp_max_it 3
-fieldsplit_ulambda_fieldsplit_lmabda_pc_type bjacobi
# for evaluation of inverse of block matrix K^{S2}
-fieldsplit_ulambda_fieldsplit_u_ksp_type preonly
-fieldsplit_ulambda_fieldsplit_u_pc_type lu
-fieldsplit_ulambda_fieldsplit_u_pc_factor_mat_solver_type mumps
```

References

[1] J.C. Simo, T.J.R. Hughes, Computational Inelasticity, in: Volume 7 of Interdisciplinary Applied Mathematics, Springer-Verlag New York, Inc, 1998.

[2] E.A.d.S. Neto, D. Perić, D.R.J. Owen, Computational Methods for Plasticity: Theory and Applications, John Wiley & Sons, Ltd, 2008.

[3] J. Brown, Efficient nonlinear solvers for nodal high-order finite elements in 3D, J. Sci. Comput. 45 (1) (2010) 48–63.

[4] M. Kronbichler, K. Kormann, A generic interface for parallel cell-based finite element operator application, Comput. & Fluids 63 (2012) 135–147.

[5] B. Gmeiner, M. Huber, L. John, U. Rüde, B. Wohlmuth, A quantitative performance study for Stokes solvers at the extreme scale, J. Comput. Sci. 17 (2016) 509–521.

[6] D. Davydov, J.P. Pelteret, D. Arndt, M. Kronbichler, P. Steinmann, A matrix-free approach for finite-strain hyperelastic problems using geometric multigrid, Internat. J. Numer. Methods Engrg. 121 (13) (2020) 2874–2895.
[7] C. Miehe, N. Apel, M. Lambrecht, Anisotropic additive plasticity in the logarithmic strain space: Modular kinematic formulation and implementation based on incremental minimization principles for standard materials, Comput. Methods Appl. Mech. Engrg. 191 (47–48) (2002) 5383–5425.
[8] A. Cuitiño, M. Ortiz, A material-independent method for extending stress update algorithms from small-strain plasticity to finite plasticity with multiplicative kinematics, Eng. Comput. 9 (1992) 437–451.
[9] A.L. Eterovic, K.J. Bathe, A hyperelastic-based large strain elasto-plastic constitutive formulation with combined isotropic-kinematic hardening using the logarithmic stress and strain measures, Internat. J. Numer. Methods Engrg. 30 (6) (1990) 1099–1114.
[10] D. Perić, D.R. Owen, M.E. Honnor, A model for finite strain elasto-plasticity based on logarithmic strains: Computational issues, Comput. Methods Appl. Mech. Engrg. 94 (1) (1992) 35–61.
[11] J.C. Simo, Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory, Comput. Methods Appl. Mech. Engrg. 99 (1) (1992) 61–112.
[12] C. Miehe, A formulation of finite elastoplasticity based on dual co- and contra-variant eigenvector triads normalized with respect to a plastic metric, Comput. Methods Appl. Mech. Engrg. 159 (3–4) (1998) 223–260.
[13] M. Schmidt-Baldassari, Numerical concepts for rate-independent single crystal plasticity, Comput. Methods Appl. Mech. Engrg. 192 (11–12) (2003) 1261–1280.
[14] G. Scalet, F. Auricchio, Computational methods for elastoplasticity: An overview of conventional and less-conventional approaches, in: Archives of Computational Methods in Engineering - State of the Art Reviews, Vol. 25, Springer Netherlands, 2018, pp. 545–589.
[15] T. Kaiser, A. Menzel, An incompatibility tensor-based gradient plasticity formulation-Theory and numerics, Comput. Methods Appl. Mech. Engrg. 345 (2019) 671–700.
[16] T. Liebe, P. Steinmann, Theory and numerics of a thermodynamically consistent framework for geometrically linear gradient plasticity, Internat. J. Numer. Methods Engrg. 51 (12) (2001) 1437–1467.
[17] T. Liebe, A. Menzel, P. Steinmann, Theory and numerics of geometrically non-linear gradient plasticity, Internat. J. Engrg. Sci. 41 (13–14) (2003) 1603–1629.
[18] P.M. Pinsky, A finite element formulation for elastoplasticity based on a three-field variational equation, Comput. Methods Appl. Mech. Engrg. 61 (1) (1987) 41–60.
[19] J.C. Simo, J.G. Kennedy, R.L. Taylor, Complementary mixed finite element formulations for elastoplasticity, Comput. Methods Appl. Mech. Engrg. 74 (2) (1989) 177–206.
[20] B. Schröder, D. Kuhl, Small strain plasticity: classical versus multifield formulation, Arch. Appl. Mech. 85 (8) (2015) 1127–1145.
[21] B. Schröder, D. Kuhl, Static condensation within the context of multifield elastoplasticity, PAMM 15 (1) (2015) 423–424.
[22] N. Fleck, G. Muller, M. Ashby, J. Hutchinson, Strain gradient plasticity: Theory and experiment, Acta Metall. Mater. 42 (2) (1994) 475–487.
[23] J.S. Stölken, A.G. Evans, A microbend test method for measuring the plasticity length scale, Acta Mater. 46 (14) (1998) 5109–5115.
[24] M.D. Uchic, D.M. Dimiduk, J.N. Florando, W.D. Nix, Sample dimensions influence strength and crystal plasticity, Science 305 (5686) (2004) 986–989.
[25] E.C. Aifantis, On the microstructural origin of certain inelastic models, J. Eng. Mater. Technol. 106 (4) (1984) 326–330.
[26] E.C. Aifantis, The physics of plastic deformation, Int. J. Plast. 3 (3) (1987) 211–247.
[27] A. Menzel, P. Steinmann, On the continuum formulation of higher gradient plasticity for single and polycrystals, J. Mech. Phys. Solids 48 (8) (2000) 1777–1796.
[28] M.E. Gurtin, A gradient theory of single-crystal viscoplasticity that accounts for geometrically necessary dislocations, J. Mech. Phys. Solids 50 (1) (2002) 5–32.
[29] P. Gudmundson, A unified treatment of strain gradient plasticity, J. Mech. Phys. Solids 52 (6) (2004) 1379–1406.
[30] M.E. Gurtin, L. Anand, A theory of strain-gradient plasticity for isotropic, plastically irrotational materials. Part I: Small deformations, J. Mech. Phys. Solids 53 (7) (2005) 624–1649.
[31] M.E. Gurtin, L. Anand, A theory of strain-gradient plasticity for isotropic, plastically irrotational materials. Part II: Finite deformations, Int. J. Plast. 21 (12) (2005) 2297–2318.
[32] A.G. Evans, J.W. Hutchinson, A critical assessment of theories of strain gradient plasticity, Acta Mater. 57 (5) (2009) 1675–1688.
[33] S. Forest, Micromorphic approach for gradient elasticity, viscoplasticity, and damage, J. Eng. Mech. 135 (3) (2009) 117–131.
[34] B.D. Reddy, The role of dissipation and defect energy in variational formulations of problems in strain-gradient plasticity. Part 1: polycrystalline plasticity, Contin. Mech. Thermodyn. 23 (6) (2011) 551–572.
[35] B.D. Reddy, The role of dissipation and defect energy in variational formulations of problems in strain-gradient plasticity. Part 2: single-crystal plasticity, Contin. Mech. Thermodyn. 23 (6) (2011) 551–572.
[36] C. Miehe, Variational gradient plasticity at finite strains. Part I: Mixed potentials for the evolution and update problems of gradient-extended dissipative solids, Comput. Methods Appl. Mech. Engrg. 268 (2014) 677–703.
[37] R. de Borst, H. Muhlhaus, Gradient-dependent plasticity: Formulation and algorithmic aspects, Internat. J. Numer. Methods Engrg. 35 (1992) 521–539.
[38] R. de Borst, L.J. Sluys, H.-B. Muhlhaus, J. Pamin, Fundamental issues in Finite Element analyses of localization of deformation, Eng. Comput. 10 (1993) 99–121.
[39] T. Svedberg, K. Runesson, An algorithm for gradient-regularized plasticity coupled to damage based on a dual mixed FE-formulation, Comput. Methods Appl. Mech. Engrg. 161 (1–2) (1998) 49–65.
[40] L.P. Evers, W.A.M. Brekelmans, M.G.D. Geers, Non-local crystal plasticity model with intrinsic SSD and GND effects, J. Mech. Phys. Solids 52 (10) (2004) 2379–2401.

[41] J.K. Djoko, F. Ebobisse, A.T. McBride, B.D. Reddy, A discontinuous Galerkin formulation for classical and gradient plasticity, Part 2: Algorithms and numerical analysis, Comput. Methods Appl. Mech. Engrg. 197 (1–4) (2007) 1–21.
[42] P. Fredriksson, P. Gudmundson, L.P. Mikkelsen, Finite element implementation and numerical issues of strain gradient plasticity with application to metal matrix composites, Int. J. Solids Struct. 46 (22–23) (2009) 3977–3987.
[43] S. Bargmann, B.D. Reddy, Modeling of polycrystals using a gradient crystal plasticity theory that includes dissipative micro-stresses, Eur. J. Mech. A/Solids 30 (5) (2011) 719–730.
[44] B.D. Reddy, C. Wieners, B. Wohlmuth, Finite element analysis and algorithms for single-crystal strain-gradient, Internat. J. Numer. Methods Engrg. 90 (2012) 784–804.
[45] C.F. Niordson, J.W. Kysar, Computational strain gradient crystal plasticity, J. Mech. Phys. Solids 62 (1) (2014) 31–47.
[46] K.L. Nielsen, C.F. Niordson, A numerical basis for strain-gradient plasticity theory: Rate-independent and rate-dependent formulations, J. Mech. Phys. Solids 63 (1) (2014) 113–127.
[47] A. Panteghini, L. Bardella, On the finite element implementation of higher-order gradient plasticity, with focus on theories based on plastic distortion incompatibility, Comput. Methods Appl. Mech. Engrg. 310 (2016) 840–865.
[48] B.D. Coleman, W. Noll, The thermodynamics of elastic materials with heat conduction and viscosity, Arch. Ration. Mech. Anal. 13 (1) (1963) 167–178.
[49] C. Frederick, P. Armstrong, A mathematical representation of the multiaxial Bauschinger effect, Mater. High Temp. 24 (1) (2007) 1–26.
[50] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F. Smith, H. Zhang, PETSc users manual, Technical report, Argonne National Laboratory, 2014, https://www.anl.gov.
[51] J.R. Westlake, A Handbook of Numerical Matrix Inversion and Solution of Linear Equations, John Wiley & Sons, New York - London - Sydney, 1968.
[52] P. Keast, Moderate-degree tetrahedral quadrature formulas, Comput. Methods Appl. Mech. Engrg. 55 (3) (1986) 339–348.
[53] L. Kaczmarczyk, Z. Ullah, K. Lewandowski, X. Meng, X.-Y. Zhou, I. Athanasiadis, H. Nguyen, C.-A. Chalons-Mouriesse, E.J. Richardson, E. Muir, A.G. Shvarts, M. Wakeni, C.J. Pearce, MoFEM: An open source, parallel finite element library, J. Open Source Softw. 5 (45) (2020) 1441.
[54] S. Balay, W.D. Gropp, L.C. McInnes, B.F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A.M. Bruaset, H.P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202.
[55] Finite-strain elastoplasticity within the logarithmic strain framework. URL https://thelfer.github.io/mgis/web/mgis_fenics_finite_strain_elastoplasticity.html.
[56] J. Korelc, Multi-language and multi-environment generation of nonlinear finite element codes, Eng. Comput. 18 (4) (2002) 312–327.
[57] T. Helfer, B. Michel, J.M. Proix, M. Salvo, J. Sercombe, M. Casella, Introducing the open-source mfront code generator: Application to mechanical behaviours and material knowledge management within the PLEIADES fuel element modelling platform, Comput. Math. Appl. 70 (5) (2015) 994–1023.
[58] A. Logg, K.-A. Mardal, G.N. Wells, Automated Solution of Differential Equations by the Finite Element Method, Springer, 2012.
[59] P. Papadopoulos, J. Lu, A general framework for the numerical solution of problems in finite elasto-plasticity, Comput. Methods Appl. Mech. Engrg. 159 (1–2) (1998) 1–18.
[60] A. Seitz, A. Popp, W.A. Wall, A semi-smooth Newton method for orthotropic plasticity and frictional contact at finite strains, Comput. Methods Appl. Mech. Engrg. 285 (2015) 228–254.
[61] T. Helfer, J. Bleyer, T. Frondelius, I. Yashchuk, T. Nagel, D. Naumov, The MFront generic interface support project, J. Open Source Softw. 5 (48) (2020) 2003.
[62] J. Brown, M.G. Knepley, D.A. May, L.C. McInnes, B. Smith, Composable linear solvers for multiphysics, in: Proceedings - 2012 11th International Symposium on Parallel and Distributed Computing, ISPDC 2012, 2012, pp. 55–62.